FORECAST COMBINATION

IN REVENUE MANAGEMENT DEMAND FORECASTING

SILVIA RIEDEL

A thesis submitted in partial fulfilment of the requirements of

Bournemouth University for the degree of Master of Philosophy

August 2007

Bournemouth University in collaboration with Lufthansa Systems Berlin GmbH

Copyright Statement

PhD thesis Silvia Riedel

Forecast Combination in Revenue Management Demand Forecasting

Abstract

The domain of multi level forecast combination is a challenging new domain containing a large potential for forecast improvements. This thesis presents a theoretical and experimental analysis of different types of forecast diversification on forecast error covariances and resulting combined forecast quality. Three types of diversification are used: (a) diversification concerning the level of learning (b) diversification of predefined parameter values and (c) the use of different forecast models.

The diversification is carried out on forecasts of seasonal factor predictions in Revenue Management for Airlines. After decomposing the data and generating diversified forecasts a (multi step) combination procedure is applied. We provide theoretical evidence of why and under which conditions multi step multi level forecast combination can be a powerful approach in order to build a high quality and adaptive forecast system. We theoretically and experimentally compare models differing with respect to the used decomposition, diversification as well as the applied combination models and structures.

After an introduction into the application of forecasting seasonal behaviour in Revenue Management, a literature review of the theory of forecast combination is provided. In order to get a clearer idea of under which condition combination works, we then investigate aspects of forecast diversity and forecast diversification. The diversity of forecast errors in terms of error covariances can be expressed in a decomposed manner in relation to different independent error components. This type of decomposed analysis has the advantage that it allows conclusions concerning the potential of the diversified forecasts for future combination. We carry out such an analysis of effects of different types of diversification on error components corresponding to the bias-variance-Bayes decomposition proposed by James and

Hastie [James 96].

Different approaches of how to include information from different levels into forecasting are also discussed in the thesis. The improvements achieved with multi level forecast combination prove that theoretical analysis is extremely important in this relatively new field. The bias-variance-Bayes decomposition is extended to the multi level case. An analysis of the effects of including forecasts with parameters learned at different levels on the bias and variance error components show that forecast combination is the best choice in comparison to some other discussed alternatives. The proposed approach represents a completely automatic procedure. It realises changes in the error components which are not only advantageous at the low level, but have also a stabilising effect on aggregates of low level forecasts to the higher level. We also identify cases in which multi level forecast combination should ideally be connected with the use of different function spaces and/or thick modelling related to certain parameter values or preprocessing procedures.

In order to avoid problems occurring for large sets of highly correlated forecasts when considering covariance information, we investigated the potential of pooling and trimming for our case. We estimate the expected behaviour of our diversified forecasts in purely error variance based pooling represented by a common approach of Aiolfi and Timmermann [Aiolfi 04] and analyse effects of different kinds of covariances on the accuracy of the combined forecast. We show that a significant loss in the expected forecast accuracy may ensue because of typical inhomogeneities in the covariance matrix for the analysed case.

If covariance information is available in a sufficiently high quality, it is possible to run a clustering directly based on covariance information. We discuss how to carry out a clustering in that case. We also consider a case (quite common in our application) when covariance information may not be available and propose a novel simplified representation of the covariance matrix which represents the distance in the forecast generation space and is only based on knowledge about the forecast generation process. A new pooling approach is proposed that avoids

inhomogeneities in the covariance matrix by considering the information contained in the simplified covariance representation. One of the main advantages of the proposed approach is that the covariance matrix does not have to be calculated. We compared the results of our approach with the approach of Aiolfi and Timmermann and explained the reasons for significant improvement. Another advantage of our approach is that it leads to the generation of novel multi step, multi level forecast generation structures that carry out the combination in different steps of pooling.

Finally, we describe different evolutionary approaches in order to generate combination structures automatically. We investigate very flexible approaches as well as approaches that avoid the expected inhomogeneities in the error covariance matrix based on our theoretical findings.

The theoretical analysis is supported by experimental results. We could achieve an improvement of forecast quality up to 11 percent for the practical application of demand forecasting in Revenue Management compared to the current optimised forecasting system.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Acknowledgements

During the four year period I have been in contact with many people that have contributed to my work in a positive way.

First of all I would like to thank my supervisor Bogdan Gabrys. Thanks a lot for letting me profit from your knowledge, for intensive discussions and very helpful explanations and proofreading. Thanks for believing in me, for constant motivation and for being at the same time a nice and critical guy. Many thanks also for the flexibility and respect for busy periods at LSB. Without this flexibility I would never have been able to complete this PhD as a part time external student. Thanks also to Christiane Lemke for proofreading and testing.

I would also like to thank my colleagues at Lufthansa Systems Berlin for making this PhD possible. I thank Michal Lukaschewitsch for being my second supervisor and critically controlling the progress of the project from the application side. Thanks to my superiors Thomas Bueermann and Frank Masurat for the permission to work part time and for supporting me with a project covering travel and hotel costs as well as conference visits. Thanks also to Stefan Pölt from Lufthansa for allowing me to use Lufthansa data for the experiments.

Finally, I would like to thank my parents, Gisela and Roland Riedel, and my brother, Gerald, for their constant support and understanding.

Declaration

The work contained in this thesis is the result of my own investigations and has not been accepted nor concurrently submitted in candidature for any other award.

Overview of Original Contributions

Before starting with an introduction to the problem in the following sections, this section provides a brief summary of the major original findings arising from the thesis. The study has been summarised in a number of peer reviewed publications [Riedel 03][Riedel 04][Riedel 05a][Riedel 05b] [Riedel 07a](and [Riedel 07b] submitted) encompassing both theoretical and experimental material realising the project goals.

Experimental analysis of forecast combination in Revenue Management seasonal demand forecasting

The first contribution is concerned with an analysis of the potential of known linear and nonlinear combination models for the application to seasonal forecasting in Revenue Management for Airlines. Different known combination models described in Chapter 3 are applied to demand forecasts generated for a sample of 20 origin destination itinerary pairs of a major European Carrier. The combination is carried out on total demand predictions (Section 3.4) as well as on decomposed predictions in relation to the seasonal demand component (Section 4.6).

Discussion of the effects of diversification of different types of parameters in relation to the bias- variance- Bayes error decomposition

A novel summary of effects of diversification of different types of parameters is provided in Section 4.3. The analysis is based on the error bias- variance- Bayes decomposition proposed by James and Hastie [James 96]. The analysis of the effects of diversification of different types of parameters on different error components is provided. The results of this analysis allow to make conclusions for the combination of forecasts diversified by these types of parameters.

Analysis of multi level forecast combination in relation to the bias- variance- Bayes error decomposition

Multi level forecasting is based on the idea of learning information at different levels of data aggregation. Different approaches have been described in the literature [Fliedner 01] in order to determine the ideal level and to distribute the learned

information to other levels. We analyse the approaches of using the information learned at different levels and to use forecast combination approaches for a fusion of the learned behaviour. We carry out an investigation of multi level forecast combination in relation to the forecast error bias- variance- Bayes decomposition [James 96] in Chapter 5. We provide the extension of this decomposition for the multi level case.

Comparison of multi level forecast combination with other approaches using multi level information

The analysis of the decomposition of forecast errors when combining forecasts generated at different levels allows a comparison with alternative approaches of including information available at different levels. In Chapter 5 we analyse different cases of typical situations occurring at different levels concerning,e.g., noise at the low level of data aggregation and special behaviour in comparison to the higher level. We show that in many cases forecast combination can be used in order to take advantage of the potential of information provided at the different levels, but we also identify cases in which the pure multi level approach would not result in large forecast improvements. In order to solve this problem we identify alternative types of diversification which are able to handle such cases.

Analysis of effects on error covariances when different types of diversification are used at the same time

The results of the analysis of multi level forecast combination motivate a theoretical analysis of effects of forecast diversification on error covariances. We have carried out this analysis for the special case of forecasts that have been diversified by three different methods: with parameters learned at different levels, by thick modelling and with the use of different function spaces. In Chapter 6 we provide a novel view of effects of these methods of diversification on the decomposed error components. We express the "diversity" of different forecasts in relation to different error components and propose a measure in order to quantify it.

Analysis of effects of error variance based pooling in case of multi level

forecast combination

We also analyse what effects different kinds of covariances can have on the quality of purely error variance based pooling as proposed by Aiolfi and Timmermann [Aiolfi 04]. We could observe that if only error variance pooling is used for multi- level forecasts there is a loss in expected forecast accuracy because of typical inhomogeneities in the covariance matrix which frequently occur. If covariance information is available in a sufficiently high quality, it is possible to take it into account during the pooling process.

Proposition of a simplified covariance representation that can be used for pooling

In Section 6.4 we study the difficult case in which covariance information cannot be measured properly and propose a novel simplified representation of the covariance matrix which is only based on knowledge about the forecast generation process. We propose a new pooling approach that avoids inhomogeneities in the covariance matrix by considering the information contained in the simplified covariance representation and compare it with the approach of Aiolfi and Timmermann [Aiolfi 04]. In Section 6.5 we lead with a novel discussion of how to use covariance information if available in a reliable or less reliable quality. Based on this analysis we propose different options of how to include this information into a pooling procedure.

Evolution of multi step multi level combination structures

Novel aspects of Chapter 7 concern the generation of multi step multi level combination structures defined as optimisation problems that can be solved by evolutionary computation. We propose and analyse different approaches and constraints informed by to the theoretical findings provided in the previous chapters, which allow to explain differences in the results obtained in experiments. We obtain systems which are able to evolve well performing multi level combination structures automatically.

Additional Benefits

In addition to the theoretical and experimental contributions described in this thesis the knowledge gained about forecast combination could be used in different areas and has already influenced the implementation of recent components in the Revenue Management product *ProfitLine.Yield/O&D*. So different large and medium size airlines already profit from forecast improvements achieved with a sophisticated fusion of time series and passenger name record based noshow forecasts. New models to predict market and price sensitive demand for airlines developed for *ProfitLine.Yield/O&D* and *ProfitLine.Yield/Rembrandt* are based on forecast fusion approaches as well.

Overview of Mostly Used Variables and Indices

Variables

x	input data
y	target data
$\widehat{\cdot}$	estimation/prediction
ϵ	random noise
E	average value
δ^2	(error) variance (component)
ρ	covariance
ϱ	correlation
ϕ	parameter
w	linear combination weight
e	forecast error
Σ	covariance matrix
η	unit vector
rk	forecast rank
ζ	fitness

Functions

f	functional relationship between input data and target to be predicted
h	a function from function space $\mathcal{H}$ used in order to approximate f
F	combination function
G	subfunction in combination function

Indices

t	unspecified time period
t_d	departure date
t_p	process date
$t_{d,\tau}$	departure date d measured at a certain time τ prior to the departure
c	data component
i	level, subspace of the input space
m	index in an ordered set of forecasts (used as input in a forecast combination)
s	position in the forecast generation space
k	function space type
γ	step in a combination structure
α	parameter values used for thick modelling
$comb$	combined forecast
n	dimensions of a function space
$\widetilde{n}$	dimensions of a parameter vector
e	total error
h	error bias component
ϕ	error variance component
y	error Bayes component (random noise)

Position of Indices

${}^{m}\widehat{y}^{c}_{ti}$	forecast for component c at time t and level i generated with method m
$\mathcal{H}_{k\alpha}$	function space/method based on type of function and fixed parameter values
$h_{k\alpha}(x, \widehat{\phi}_i)$	function from $\mathcal{H}_{k\alpha}$ with parameters ϕ estimated on level i
${}^{\mathcal{H}_{k\alpha}i}\delta^2_e$	forecast error component by the used function space, the level of learning and the error component

1. INTRODUCTION

There are clear and obvious advantages in combining forecasts, both to better understand the generating mechanism of the series and also to pragmatically achieve better forecasts. (Granger and Ramanathan, [Granger 84])

1.1 Introduction to Revenue Management

This PhD is a cooperation project with Lufthansa Systems Berlin GbmH and related to the industrial application of Revenue Management forecasting for airlines. In order to motivate the theoretical relevance of the line of research followed in the PhD, we will start with a short introduction into Revenue Management and issues occurring in Revenue Management demand forecasting.

The product of the airline industry are seats on airplanes offered with different booking conditions and for different levels of comfort. To maximise revenue, priority is given to high revenue booking classes. Capacity must be protected for high revenue passengers usually arriving shortly before a plane's departure. Based on the size of the protected capacity, the capacity of low revenue classes needed to fill up the aircraft can be determined. Therefore, the central question of revenue management is: How much of the overall capacity should be made available for low-yield customers? Or in other words: How much space should be reserved for the high-yield segment?

To answer this question, the following technical components are used: a) an inventory to control capacity; b) a forecasting for assessing the demand in advance; and c) an optimisation to maximise the revenue by capacity control.

While the focus of this thesis is placed on forecasting of the demand, more detailed information about all revenue management components can be found in [McGill 99] [Talluri 04][Weatherford 92][Cross 97][Zaki 00][Pak 02].

Effects of Revenue Management on the revenue of an airline can be illustrated with the following example. Figures 1 and 2 show the booking process for two flights, a high demand flight and a low demand flight, with and without Revenue Management.

Generally, the low yield passengers book earlier than the high yield passengers. If they have the choice they book the high demand flight. Without Revenue Management the high demand flight is already nearly fully booked a long time prior to departure. There is no capacity remaining for later booking passengers booking in high yield fareclasses, which means that these bookings must be turned away. The result is a high demand flight filled with low yield passengers, which is bad, and a low demand flight flying with a lot of empty seats, which is even worse.

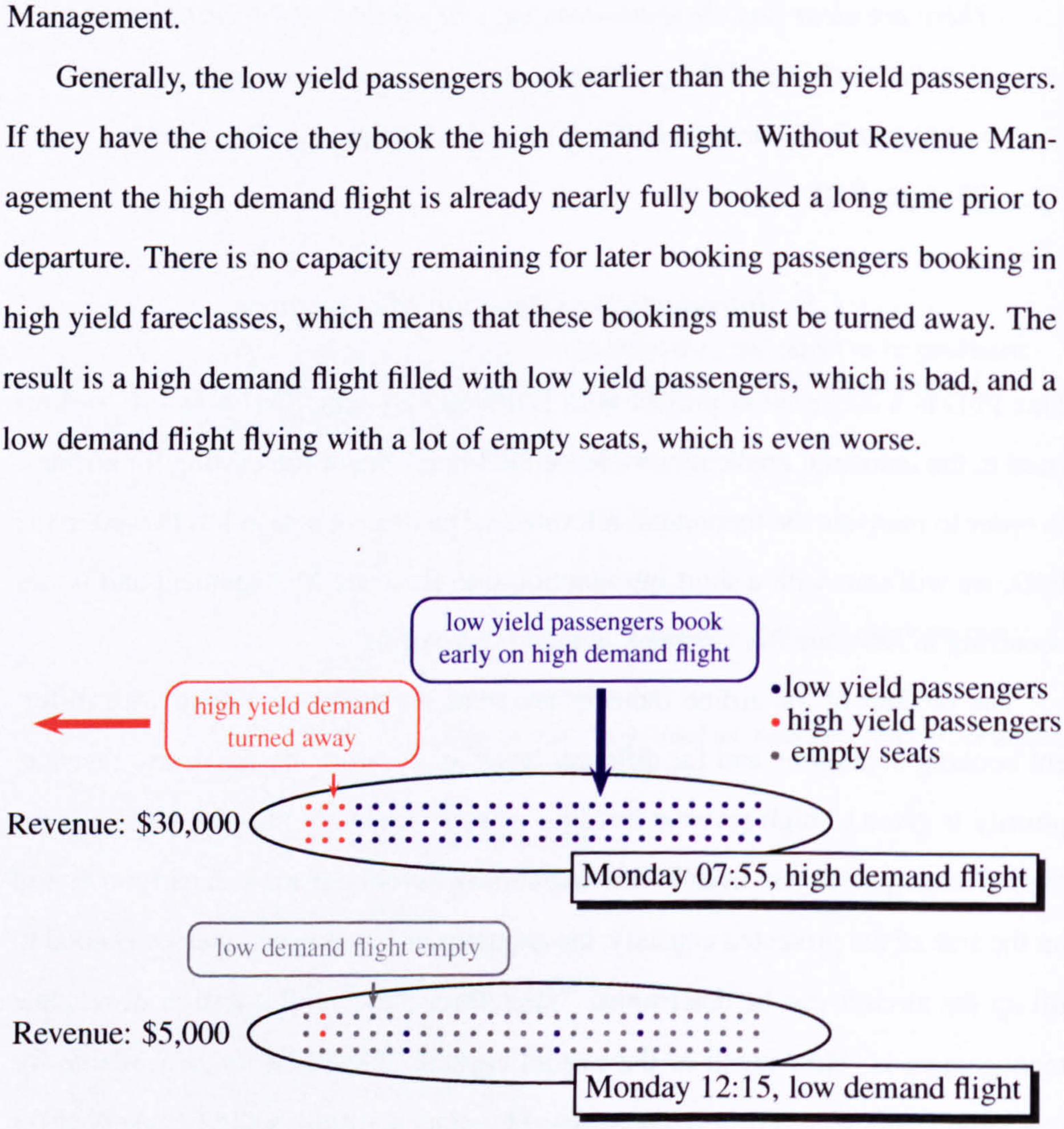

Fig. 1: An example of two typical flights with booking behaviour without Revenue Management system.

With Revenue Management system in place the high yield demand is assessed

in advance (as well as the low yield demand). This allows the blocking of seats in the high demand flight for the later arriving high yield customers. The early booking low yield passengers cannot book the high demand flight any more and partly move to the low demand flight. The result is a high demand flight filled with mostly high yield passengers and a low demand flight flying with low yield passengers, which brings an acceptable revenue for both flights.

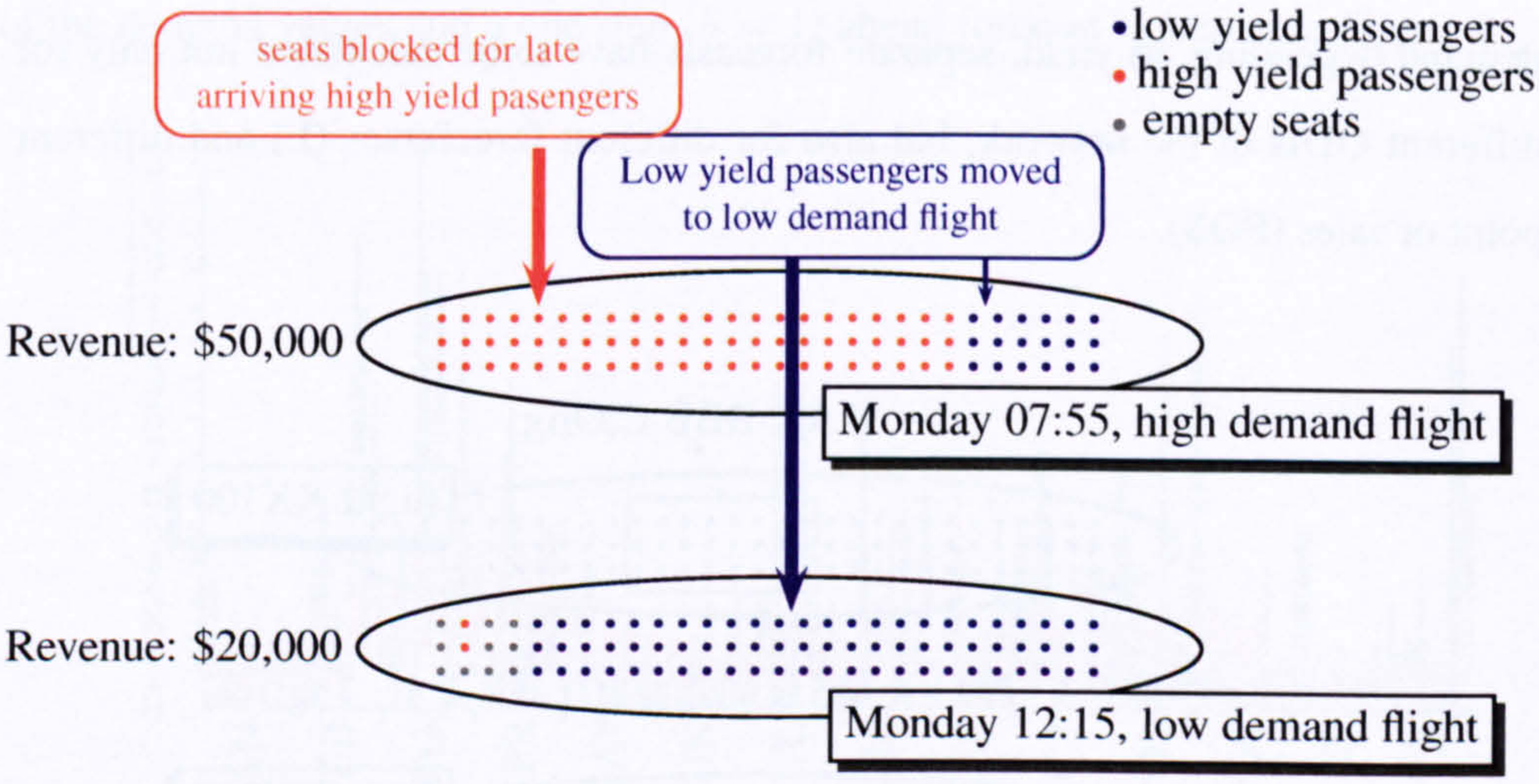

Fig. 2: An example of two typical flights with booking behaviour with Revenue Management system.

1.2 Demand Forecasting in Revenue Management

1.2.1 Segment versus O&D Forecasting

As traditional airlines (in contrast to some lowcost airlines) allow bookings not only for single flights, but for whole trips, it is a crucial Revenue Management system task not only to control the different types of demand concerning yield, but also to take into account network effects.

As a result, it has to be decided, for instance, if a local passenger should be accepted for a national flight or if it is advantageous to wait for the passenger

using this flight as an inbound flight to a high yield intercontinental flight. Such passengers would only be the best choice if not enough passengers are expected to take the intercontinental flight, because two local passengers generate in total more revenue than one connecting passenger.

To handle such effects, larger airlines have started using prediction systems which do not predict the demand per scheduled flight (segment), but per origin destination pair (O&D). Figure 3 shows an example for an ODI (origin destination itinerary) represented by different segments. As the optimisation controls the demand depending on yield, separate forecasts have to be calculated not only for different ODIs of the network, but also for different fareclasses (F) and different point of sales (POS).

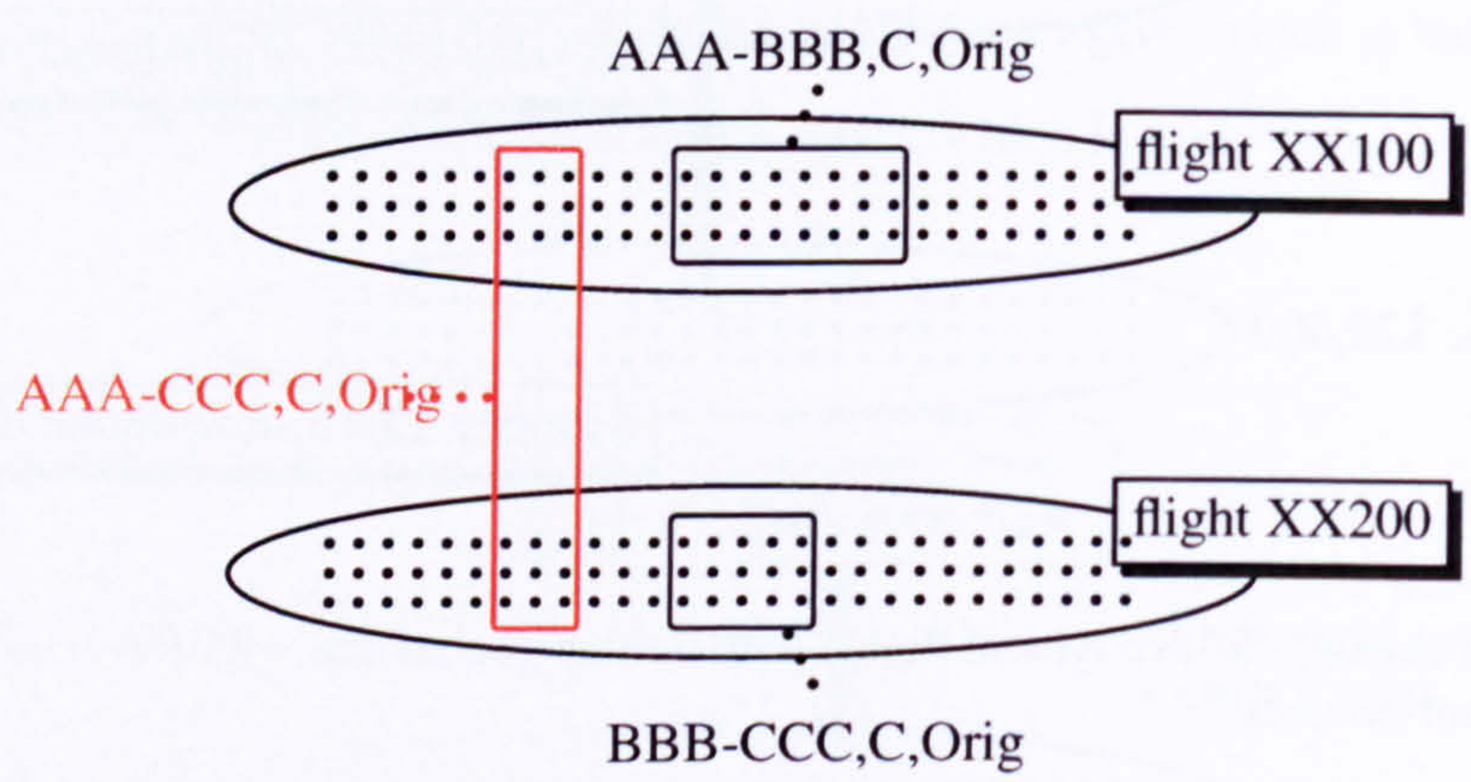

Fig. 3: Segment versus O&D view. The example shows two flights, a national flight AAA-BBB with flight number XX100 and a second intercontinental flight BBB-CCC with flight number XX200. The figure shows the demand in fareclass C (typical business passengers) and point of sale Orig (Country of Origin). Three ODIs are illustared, the two ODIs representing bookings without connection as well as the connection ODI for both flights.

1.2.2 Issues of O&D Forecasting

The Issue of a Large Number of Small Numbers Predictions

Demand at such a fine level of forecasting (i.e. ODI F POS) can be modelled as a time series, e.g. per departure date. Formally, one can say that we have a time series (y_{t_d}), $t_d = 1..t_p$ given denoting historical total demand for departure date t_d. The last date t_p represents the current process date. The general problem is to forecast the demand for future departure dates (y_{t_p+h}), $h \in \mathcal{N} \geq 1$. An example of the demand values and a one step ($h = 1$) ahead forecast is shown in Figure 4.

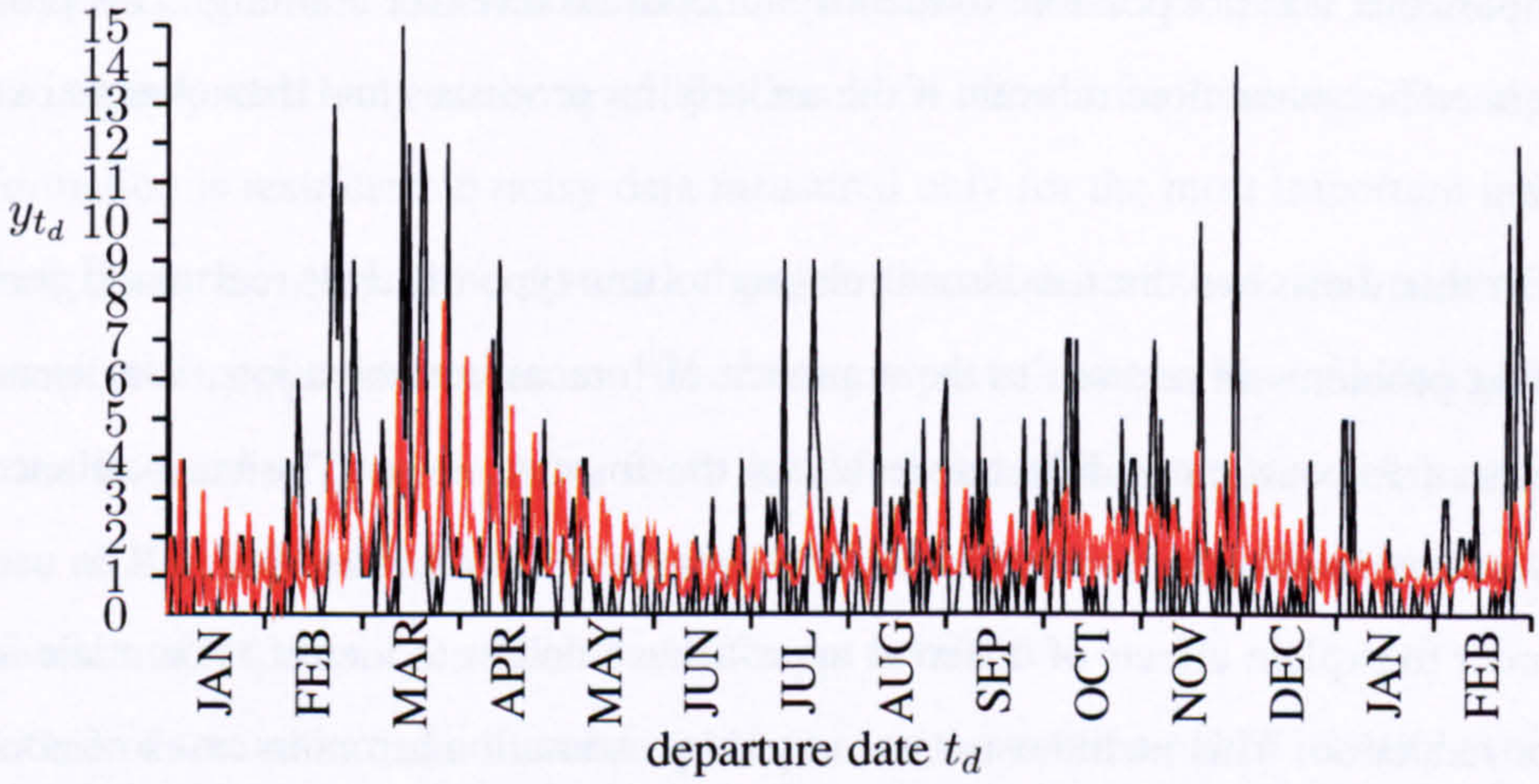

Fig. 4: Example of the demand values per departure date (black line) with one step (h=1) ahead forecasts (orange/light line).

Issues resulting from predicting small numbers at a very fine level are also quite common in other applications [Armstrong 01][Fliedner 01]. On this level, the data is extremely noisy and exhibits frequently multiple structural breaks. In our application, these structural breaks in the time series data reflect the changes in booking behaviour caused by seasonal changes, special events, such as holidays or fairs, changes in the flight schedules of both the airlines for which the predictions are made and the competitors, or changes of the political or cultural situation of a country. All these changes have to be handled in the forecasting process.

The reaction to large noise components and in consequence structurally poor

forecasts at the fine level of forecasting is often the decision to learn structural information or causal effects at higher levels meaning learning based on aggregates of the target data. So it is for instance possible to learn seasonal factors on the O&D level and to apply the learned factors for all fareclasses and point of sales. This decreases noise but leads to an information loss related to effects which occur only at the fine level.

The choice of the level of learning often results from a data analysis. However, even if the data analysis has been performed well, it is likely that the real relationship between given inputs at different levels and the values to predict is so complex that it is not possible to identify an optimal level for learning. This problem becomes even more relevant if the underlying processes and data change over time.

In this thesis we discuss issues relating to this type of hard, real world forecasting problems in relation to the approach of forecast combination. We discuss effects of forecasting at different levels on the forecast error. The bias-variance-Bayes error decomposition proposed by James and Hastie [James 96] will be used in order to explain effects of different approaches in order to identify potentials for error reduction. This includes issues caused by estimation errors in cases of noisy training data as well as the difficult task of using information available at different levels.

The Issue of Adaptation

Due to its broad applicability forecasting time series is a very well researched and discussed topic (good introductions to the topic are provided in [Armstrong 01] and [Brockwell 87]). Unfortunately, only a few methods could generate well performing forecasts for our application because of the already mentioned issues of noisy and quickly changing data on the very detailed level of forecasting. The world is changing so quickly that in general only a small number of historical data can be reliably used for predictions. Simple and robust models, such as sim-

ple average, different versions of exponential smoothing [Brown 63] or regression models [Granger 86], provide significantly better results than more sophisticated methods [Brockwell 87]. The reason for the better performance of simple models lays in their ability to make adequate forecasts even on a small number of very noisy historical data and their ability to adapt more quickly to new situations. We will present more references to the literature as well as applied approaches for our application in Section 2.

A typical approach to building a forecasting model consists of a phase of data analysis, determination of appropriate levels and preprocessing, model creation, parameter calibration and validation of the forecast model. For future forecasting, data is interpreted only at the level that has been chosen for learning. The input information is restricted to noisy data measured only for the most important influencing features. And if the demand changes, the chosen methods and parameter settings are not optimal any more. All of these aspects lead to a loss of information for the forecasting process. After some time, forecast quality tends to decrease because of a lack of adaptation concerning not only the chosen models, but also the relevance of information available at different levels.

One of the main tasks in order to adapt to new situations is to identify which parts of the demand depend on which input variables. That is the reason why decomposition strategies are used to split the demand into different components which may each depend on different input variables and therefore need to be predicted separately. Decomposition allows the prediction of demand changes separately, which are commonly overlapping and may be hard to identify. This enables the application of less complex and therefore more stable forecast models. It also allows: a) the determination of the efficiency of different inputs and different models per component; b) the selection of appropriate preprocessing; and c) the determination of appropriate levels for history representation and forecasting depending on the different stability of the components.

All of the decisions just mentioned (like the choice of preprocessing, levels of

learning or parametrisation) can become suboptimal in case of a changing situation. They also represent a restriction of the forecasting process in terms of a restriction of used input information and predefined decisions concerning, e.g., the applied models and therefore an information loss. If, e.g., relevant information changes to a level that is not considered in the learning process, we will observe a decreasing forecast accuracy. We therefore investigate options of how to automatically adapt these type of choices to new situations and how to use information available in relation to, e.g., different levels or parameter values.

We follow the general idea of a) using different methods, levels and parameter values in order to ensure that all information is theoretically available; and b) applying an automatic and adaptive fusion process that identifies the relevant information and generates a final prediction. Forecast combination approaches represent such a type of processes.

1.3 Combination of Forecasts

1.3.1 Information Fusion

Fusion of distinct information can be carried out on many different levels from pure data to the decisions of individual experts operating on different parts of the available information [Hall 92][Bezdek 99][Keller 97][de Menezes 00]. It turned out that even if applied on the same task using the same data, a joint decision of combined forecasts is potentially more effective than any one individual. The different levels of abstraction at which information fusion can be carried out are closely connected with the flow of a forecasting process: data level fusion, feature level fusion, and decision fusion [Bezdek 99].

Data fusion

Data fusion is a fusion at the basic level of data sensing [Pedrycz 98]. It has been used for instance to resolve the occlusion problem in vision systems [Bezdek 99]

and for improved object detection by overlapping many partially discriminative projections [Hathaway 96].

Feature fusion

There is little evidence of the feature fusion in the literature. Fusion on this level is considered more general compared to the data fusion and often resembles forecast fusion techniques. An example of feature fusion has been shown by Keller and Gader [Keller 97] where the data features extracted from Geo-Centers GPR system have been combined by a fuzzy rule incorporating some shape characteristics of the raw data.

Decision fusion

Decision fusion relates in general to combining information partially or fully processed by forecast or classifier models and therefore is perceived to be the most general [Bezdek 99]. The major motivation driving decision fusion is that different models learn from the data imperfectly, and because they are different, it is likely that their imperfections result in different forecast errors. Individual errors made by some forecast models for some input data could be compensated by other models performing well for that particular data. This thesis is related to decision fusion in terms of forecast combination.

1.3.2 Forecast Combination

Forecast combination approaches are today a scientifically acknowledged procedure [Clemen 89][de Menezes 00][Timmermann 05] to model complex functional relationships by producing not one optimal forecast $\widehat{y}$, but a number of forecasts $\{^{m}\widehat{y}\}$ and combining them for the final prediction $^{comb}\widehat{y} \in \mathcal{R}$. The existing combination approaches differ in the description of the functional relationship $f : \mathcal{R}^m \rightarrow \mathcal{R}$ which represents the fusion process. An overview of the development in this field as well as the most common models and their relation will be

presented in Chapter 3.

There are two common groups of combination models. In linear combination models the relationship is defined as a simple weighted sum of the individual forecasts:

$$^{comb}\widehat{y} = \sum_m w_m {}^m\widehat{y} \tag{1.1}$$

with combination weights $w_m \in \mathcal{R}$. Beside the *simple average model* [Bates 69], which gives the same weight to all individual forecasts, there are two common groups of linear combination models, in which individual forecast performance is taken into account to calculate the weights. While *rank based models* [Bunn 75] [Russell 87][Klapper 98b] describe forecast performance based on ranks of past performance without directly taking into account the statistical properties of forecast errors, *variance / covariance based models* [Bates 69] and *ordinary least squares regression based models* [Granger 84] use error variance and covariance information for calculation of the weights.

A more complex and flexible group of combination models are nonlinear combination models [Sharkey 96] [Genest 86][Jacobs 95][Xu 92]. In this group, mostly application specific, approaches differ in the selection of external input information as well as in the class of methods used. Typical nonlinear approaches include neural networks [Shi 99] and (fuzzy) expert systems [Fiordaliso 98].

1.4 Influences on Combination Efficiency

As there are different combination models available, we have to answer the question of how to choose appropriate sets of input forecasts and which combination model to apply under which conditions. Different approaches have been developed to explain the performance of the combined forecasts based on error variances and covariances of the individual forecasts. It has been shown theoretically and experimentally that the best results can be achieved if different individual forecasts are diverse in the sense that they are able to provide some kind of "diverse" knowledge

to a forecasts combination process. This diversity can be achieved by using

- different input information in terms of different available sources of information, different preprocessing or history pools;
- different functional or stochastic modelling approaches; or
- different parametrization of the models.

We study these influences for the case of the above mentioned forecasting problems that have to handle small numbers and very noisy data in a quickly changing environment. We discuss how we can measure diversity and under which conditions forecast combination provides improved results. We describe the diversity achieved by different types of forecast diversification in relation to different error components. In Chapter 4, for instance, we will see that the complexity of the applied forecast model can influence the error components in a different manner to the choice of diverse sets of data used for learning. The applied forecast diversification affects the covariances of the achieved set of predictions and with that the potential for forecast combination. The provided analysis of effects of diversification on various components of decomposed forecasting error enables an analysis of how we can *actively generate sets of divers forecasts.*

1.5 Aspects of Multi Level Forecasting

We consider cases in which each prediction represents the situation in concrete subspaces of the given target space. We illustrate our argumentation using an example of seasonal demand predictions for airlines. As we have already mentioned, these have to be generated for different origin destination itinerary pairs (ODI) as well as different fareclasses (F) and different point of sales (POS). This level of forecasting, which we also call the fine/low level, is very detailed (the seasonal behaviour for a given ODI F POS combination) and therefore characterised by small numbers and very noisy data. Therefore analysts also need aggregates of the generated low

level forecasts for decision making. Modern Graphical User Interfaces support this need. They offer the functionality of a data and forecast fusion to different higher levels, which represent in our example, for instance, the ODI level or even higher levels such as country or market pairs, as shown in Figure 5.

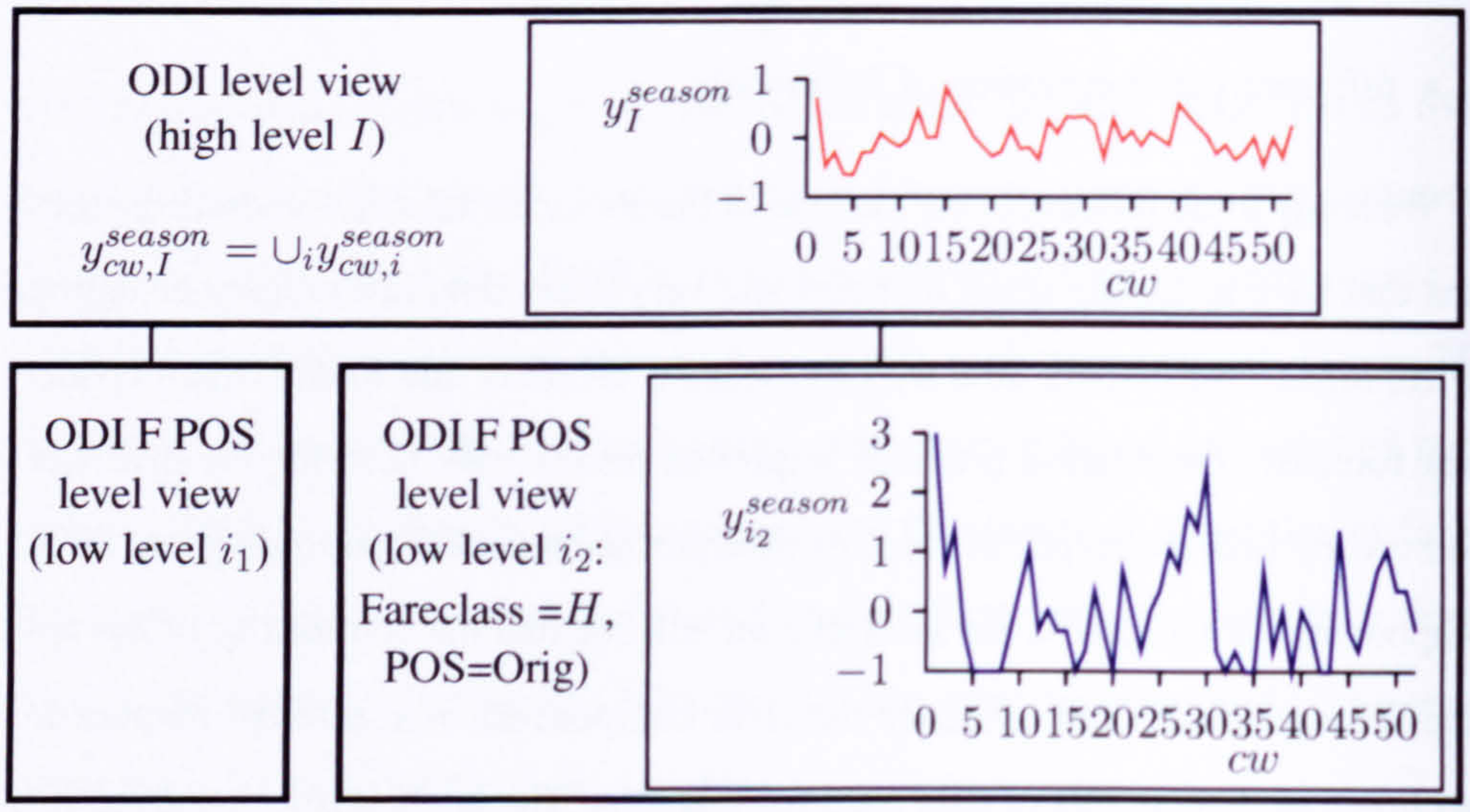

Fig. 5: A view of the low and the high level of measured historical seasonal behaviour. Seasonal factors y^{season} are shown per calendar week cw at a low level i_2 representing a special ODI Fareclass Point of Sale combination as well as at the high level I aggregate representing the whole ODI.

Large noise at the low levels often leads to the decision to learn structural information or causal effects based on aggregates of the input data or, in other words, to carry out an input data fusion with the objective of noise reduction. There is no obvious answer to the question about the adequate level for learning. Learning at different levels is related to different types of risk. If the level is chosen too fine, relevant structural information often can not be detected properly. If on the other hand the chosen level is too general, important characteristics related to special parts in the input space may be ignored. For our example this means that if we learn seasonal factors, for instance at the ODI level, we do not take into account seasonal effects in special fareclasses or point of sales properly. An introduction to such a type of problems as well as an overview of literature related to learning at

different levels and effects of forecast aggregation are provided in Chapter 5.

In practice, the problem to find the ideal level of learning is often resolved with trial and error approaches. The choice is made only on the basis of low level forecast errors. But if analysts make relevant decisions on the basis of a fusion of low level forecasts to a higher level, the need for high quality forecasts at higher levels should also be taken into account for the choice of the level of learning structural information.

In Chapter 5 we analyse effects of learning at two levels on the resulting forecast errors measured at these two levels. Choices that are purely made on forecast errors measured at the low level can be unfavourable with regard to the quality of the aggregated forecasts. We base our argumentations on the error bias, variance and Bayes decomposition proposed by James and Hastie. We provide this error decomposition for the multi-level case. This enables us: (a) to analyse effects of aggregation of forecasts generated with learning at the low level to the error components at the high level, and (b) to analyse the effects of using forecasts generated with learning at the high level to the error components at the low level.

As we will see the learning at both levels works well only in some cases, we also discuss the option of using forecast combination in order to make an automated choice or even to profit from knowledge at both levels. The positive effects of forecast combination in many applications have been explained in relation to different aspects and different decompositions of forecast errors and their correlation. We provide the analysis of the error components of combined multi-level forecasts at the low as well as at the high level. The analysis is based on the simplified version of the well known *optimal model* [Bates 69], the *optimal model with assumption of independence* [Granger 84], which takes into account the problem of high estimation errors of the inverse covariance matrix [Bunn 85] and is purely based on past error variances. We also discuss different cases of data configurations and the relation between the levels in order to show that forecasts combination works very well in cases where it represents an automatic choice of the appropriate

level as well as in cases where knowledge of both levels is relevant for learning. This includes a detailed discussion about what happens to the error components in different concrete situations illustrated using an artificial example. We will see that the approach of forecast combination allows not only an intelligent automatic choice of the superior level, if it exists, but also the generation of predictions that are more stable in terms of the quality of aggregates to higher levels and in case of changing environments. We will also show that a multi level forecast combination should ideally be connected with the use of different function spaces and/or diversification related to certain parameter values.

1.6 Generation of Multi Step Multi Level Combination Structures

A side effect of the multi level approach is that the number of forecasts to combine can get very large. It is often not possible to estimate covariances properly because of noisy training data or changing environments. Various studies [Russell 87][de Menezes 00] have shown that the resulting errors in the estimated covariance matrix can lead to large weight estimation errors for the optimal model especially for a large number of forecasts which in turn lead to unstable and poor combined forecasts. We therefore apply the approaches of pooling and trimming [Aiolfi 04] in order to handle that problem.

In experiments, which we have carried out in order to analyse the effects of different static and dynamic combination structures achieved by applying different kinds of pooling and trimming for the application of seasonal demand forecasting for airlines, we were surprised to see that the most promising structures seemed to have a tendency to cluster the input predictions depending on the type of diversifying procedure used. We could observe a clear tendency to combine first different forecasts generated at the same level but using different functional approaches and then to combine the forecasts representing different levels, or visa versa.

In Chapter 6 we provide a theoretical analysis which explains this behaviour. We start with an analysis of effects on covariances occurring for our special case

of combining forecasts that have been diversified by three different methods: with parameters learned at different levels, by fixed parameter value diversification and by the use of different function spaces. In order to explain differences in covariance values, we provide a novel view of effects of these methods of diversification on decomposed error components based on the bias- variance- Bayes error decomposition. We express the "diversity" of different forecasts in relation to different error components and propose a measure in order to quantify it.

We also analyse what effects different kinds of covariances can have on the quality of purely error variance based pooling. We refer to the approaches of Aiolfi and Timmermann [Aiolfi 04] who propose to pool forecasts based on the total error variances using k-means clustering. The results enable us to estimate the expected behaviour of our diversified forecasts. We will see that if only error variance pooling is used there is a loss in expected forecast accuracy because of typical inhomogeneities in the covariance matrix which frequently occur.

If covariance information is available in a sufficiently high quality, it is possible to take it into account during the pooling process. This means that we can run a clustering directly based on covariance information. We study the difficult case in which covariance information cannot be measured properly or is not calculated in case of applications with strong calculation time restrictions. Based on the determined effects of diversifying our forecasts in relation to different error components we propose a novel simplified representation of the covariance matrix which is only based on knowledge about the forecast generation process.

We propose a new pooling approach that avoids inhomogeneities in the covariance matrix by considering the information contained in the simplified covariance representation. We compare the results of our approach with the approach of Aiolfi and Timmermann and explain why it works better. We also mention that applying our approach again in the combination that combines the pools leads to the generation of multi step multi level forecast combination structures which carry out the combination in different steps of pooling and trimming. These multi step

multi level combination structures correspond to those which have generated significantly improved forecasts in our experimental work.

In Chapter 7 we finally describe different evolutionary approaches in order to evolve multi step multi level combination structures dynamically. We will see that evolving very flexible dynamic structures may lead to a problem of overfitting. We therefore discuss different options of how to restrict the search space. We will use our theoretical findings in order to define restrictions that avoid the generation of structures suffering from the covariance inhomogeneities mentioned above. Extensions of such evolutions allow the generation of stable and flexible multi level multi step combination structures containing good adaptive capabilities.

1.7 Organisation of the Thesis

The thesis is organised as follows:

After the introduction provided in this chapter we start with an introduction to the used notation as well as used forecasting approaches and methods for the application of Revenue Management forecasting in Chapter 2.

Then we provide a discussion and literature review concerning the topic of forecast combination in Chapter 3. Chapter 4 extends this analysis with a closer look at influences on the efficiency of forecast combination. We discuss the topic of forecast diversity in relation to: a) its impact on resulting forecast errors; b) the question of how we can quantify diversity; and c) options of how we can actively generate diverse forecasts. This chapter also provides discussions of which combination methods to use under which conditions and of the negative effects resulting from weight estimation errors.

We then discuss aspects of multi level learning in Chapter 5. After an introduction into the problem of choosing an appropriate level for learning we discuss the effects of such choices on different error components. We provide an extension of the error decomposition of James and Hastie to the multi level case and carry out an extensive analysis, answering the question of why the combination of predic-

tions using information learned at different levels constitutes a significantly better approach in comparison to using only the predictions generated at one of the levels or other multi level approaches.

Chapter 6 is then related to different questions of pooling. After a motivation why pooling is useful for our type of problem, we analyse the effects of the application of different types of diversification on forecasts error covariances and results accuracy if pure error variance based pooling is applied. We propose a simplified version of the covariance matrix and propose a new pooling approach that does not suffer from these type of problems. Finally, we discuss the dynamic generation of combination structures in Chapter 7 and finish with a summary, conclusions and potential for future work in Chapter 8.

Each chapter finishes with its own experimental section where we present the most relevant experimental results in order to motivate the ideas followed in the next chapters. Detailed results as well as a description of how to install the software used for the experiments are available in the appendix. The software is available on the CD accompanying the thesis.

2. INDIVIDUAL FORECAST GENERATION

2.1 Notation of a Forecasting Problem

2.1.1 Time Series

A large number of techniques for forecasting can be found in the literature [Armstrong 01][Brockwell 87] [Franses 63][Granger 86][Kennedy 92] [Masters 95][Elliott 07]. Parametric models assume that a relationship exists between given historical or currently available data and the data to forecast. The model describes how the data is expected to be composed as well as dependencies on given input data. We can, for instance, model a correlation over time or a linear dependency on another data set.

Models are normally built using sets of noisy data. Often it is of interest to see how series of such data develop over time. Time series define such series of data. In this thesis we use a common definition of time series similar to the one used by Brockwell and Davis in [Brockwell 87].

Definition 2.1 (Stochastic Process, Time Series): Let $t \in (1, \cdots, T) =: \mathcal{T} \subset \mathcal{N}$ be a countable index set. A stochastic process is a set $(y_t), t \in \mathcal{T}$ of random variables $y_t \in \mathcal{R}^n$. A stochastic process $(y_t), t \in \mathcal{T}, t = 1 \ldots T$ which is defined for the index set $\mathcal{T}$ of equidistant time intervals is called time series.

2.1.2 Causal Models

Causal models represent relationships between time series $x_t \in \mathcal{R}^n$ and $y_t \in \mathcal{R}$.We assume that x_t can be measured properly, that we have random noise in y_t and that an "ideal" functional relationship f exists in order to approximate y_t based on x_t.

We can represent the functional relationship between input vector $x_t \in \mathcal{R}^n$ and $y_t \in \mathcal{R}$ by the function f and a random noise term ϵ:

$$y_t = f(x_t) + \epsilon_{yt}, \tag{2.1}$$

with f the "true model" and ϵ Gaussian with $\epsilon_y \sim N(0, \delta^2_{\epsilon y})$ an independent residual component. The vector x may also contain past values or predictions of y as described in the model in [Timmermann 05].

A predefined class of functions $h : \mathcal{R}^n \times \Phi \rightarrow \mathcal{R}$ is used in order to approximate the relationship between x_t and y_t. We first define the function space comparable to the definition given in [Hansen 00]:

Definition 2.2 (Function Space): Let $x_t \in \mathcal{R}^n$ be a time series and $h : \mathcal{R}^n \times \Phi \rightarrow \mathcal{R}$ be a function with input x_t and let it depend on the parameters $\phi \in \Phi \subset \mathcal{R}^{\tilde{n}}$, then the function space of h is the linear space $\mathcal{H}$ consisting of all possible functions $h(;\phi)$ obtained by varying ϕ in the domain Φ.

In order to increase readability we will remove the parameter t in all following equations, so we write the true relationship as

$$y = f(x) + \epsilon_y. \tag{2.2}$$

We further assume that a best estimation of parameters ϕ exists in order to approximate f by $h(;\phi)$

$$f(x) \approx h(x, \phi) \tag{2.3}$$

and that we have a training set $(x, y)_{\mathcal{T}_h}$ of historical data which we use in order to estimate the parameter vector ϕ by $\widehat{\phi}$ so that $\widehat{y} = h(x, \widehat{\phi})$ represents our best estimation for the relationship between x and y. In the following we will always use the "hat" symbol in order to indicate estimations or predictions.

2.1.3 Decomposition

Data is often influenced by a whole set of factors which are assumed to be independent of each other. Some of the typical factors found in many forecasting applications are related to trends and seasonal effects. The approach of data decomposition is based on the idea of splitting the data y corresponding to these independent factors. The dependency on input data representing the impact can then be modelled for each factor separately. This approach is often advantageous [Armstrong 01] because it allows, for instance, the use of simpler models and parameter sets which are tuned to the characteristics of a specific component concerning, e.g., its structure, dependencies on input information, stability and noise level. With decomposed data it is also possible to satisfy different needs related to adaptation.

Corresponding to this approach we can write the target y depending on independent components y^c plus the noise term. For each component the functional relationship $y^c \approx f^c(x^c)$ is modelled separately. We can now use different function spaces h^c in order to approximate the different functions $f^c(x^c) \approx h^c(x^c, \phi^c)$.

We use a representation of y which assumes the target data to depend on one stable basic component c_0 as well as other components representing deviations from component c_0. A motivation for this approach will be given in section 2.2.3. We assume

$$y = y^{c_0}(1 + \sum_{c \neq c_0} y^c) + \epsilon_y. \tag{2.4}$$

We achieve a representation

$$y \approx h^{c_0}(x^{c_0}, \phi^{c_0}) \left(1 + \sum_{c \neq c_0} h^c(x^c, \phi^c)\right). \tag{2.5}$$

with $h^{c_0}(x^{c_0}, \phi^{c_0})$ representing a function that describes the behaviour of the stable component c_0 in absolute values and all other functions $h^c(x^c, \phi^c)$ estimating factors, such as seasonal factors or deviations based on special events.

2.2 Forecasting in Revenue Management

2.2.1 Demand Forecasting

As part of a modern Revenue Management system for airlines one of the critical tasks is to predict how many people would like to make a booking (if they were accepted). The target variable y_t in this case therefore represents the demand. The demand is related to different departures, so the time index $t = t_d$ represents in our application the departure date.

In Revenue Management applications the task is not to generate a single prediction, but a whole set in relation to the following properties:

- O&D - a pair of the airport of origin and the airport of destination, separated by

  ROUTING - an ordered set of airports of the itinerary, separated by

  ODO - a routing used on flights departing at specified time periods

  - F - a fareclass, which represents a fare structure connected with ticket rules and regulations

  - POS - a point of sale of the ticket, in our case separated only by "country of origin", "country of destination" and "others".

Figure 6 shows an example of demand at different departure dates for one ODO F POS combination. For exact definitions of these and other terms related to the airline industry please see Appendix A.

As mentioned before, the level on which the forecasts have to be generated is very detailed (i.e. demand per ODO F POS t_d), but analysts or related computer systems also use aggregates of the generated forecasts to higher levels (like traffic between countries). The aggregates are used for decision making or further calculation in various reports or in using a graphical user interface showing the expected situation at different levels.

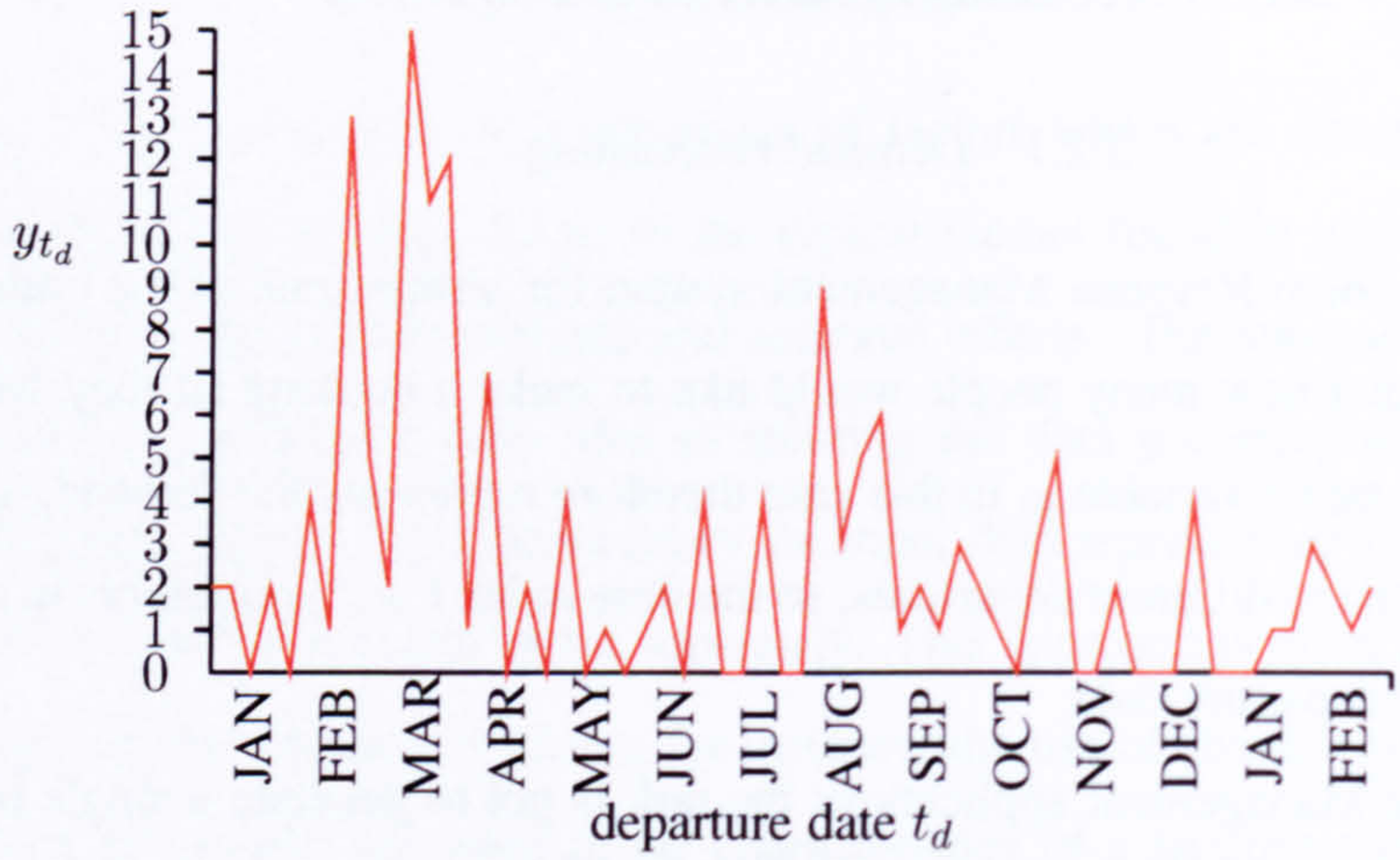

Fig. 6: Example for the time series of the demand at a given ODO DOW F POS combination.

2.2.2 Bookings versus Constrained and Unconstrained Demand

The most relevant information for demand prediction is the historical booking data as well as bookings that have already been made for a future flight for which the demand has to be predicted. However, there is a difficult problem occurring in all revenue management applications. The measured booking data is used to generate forecasts for the future demand. These forecasts serve as an input for the optimisation process which decides how many bookings will be accepted in the future in different fareclasses. As often not all bookings are accepted, the optimisation influences the number of bookings that will be observed in the future, which represent the input data for later forecasting. Figure 7 shows this spiral of influences.

The problem for the forecasting process is that the observed data does not represent the values which we would like to predict, i.e. the demand, which is the number of people who would like to make a booking. The bookings represent only that part of the demand which has been accepted. That is why the bookings are also called the *constrained demand*. The complete demand, also called *unconstrained demand*, cannot be measured and has to be approximated by an *unconstraining procedure*. The consequence is that for the fareclasses closed by the optimisation

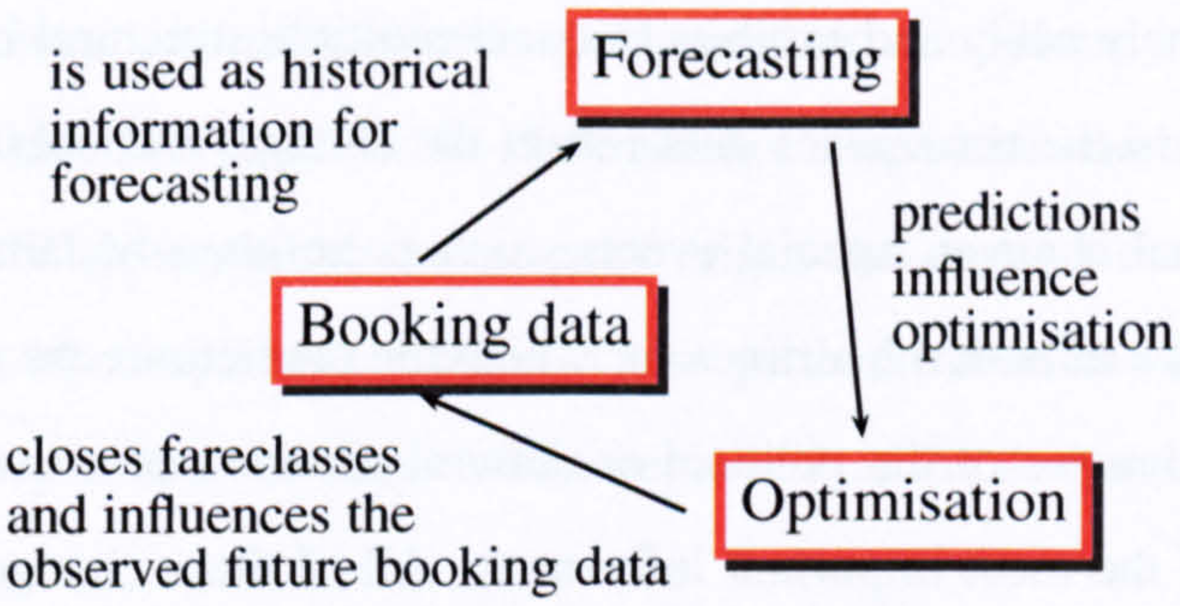

Fig. 7: The spiral of influences between bookings, forecasting and optimisation.

system we do not have any data given against which we can properly evaluate the generated unconstrained demand forecasts.

Frequently, demand forecasts can only be evaluated against data which is not real measurements, but approximations based on models which are comparable to those used to produce the forecasts.

2.2.3 Demand Components

Unfortunately, in practice only a few methods have been found to produce adequate forecasts for our application because of the structure and quality of the existing data [Talluri 04]. For the Revenue Management application the world is changing so quickly that in general only a small number of historical data is available and frequently a number of relevant values are missing. Multiple Lufthansa Systems Berlin internal studies on this topic have shown that for our data the simple and robust time series forecasting models, such as simple average, different versions of exponential smoothing [Brown 63] or regression models [Granger 86], are significantly better than a number of well known more sophisticated methods[Brockwell 87]. The reason for this lies in the simple methods' ability to make adequate forecasts even on a small number of historical data and their ability to adapt more quickly to new situations.

One of the common problems is that of predicting small numbers which result from the very fine level on which the forecasts have to be performed. On this level,

the data is extremely noisy and exhibits frequent multiple structural breaks. These structural breaks in the time series data reflect the changes in booking behaviour caused by seasonal changes, special events, such as holidays or fairs, changes in the flight schedules of both the airlines for which the predictions are made and the competitors, or changes of the political or cultural situation of a country. Figure 8 shows some of the most important influences. All of these changes have to be handled in the forecasting process and are the focus of adaptation mechanisms used within the forecasting system.

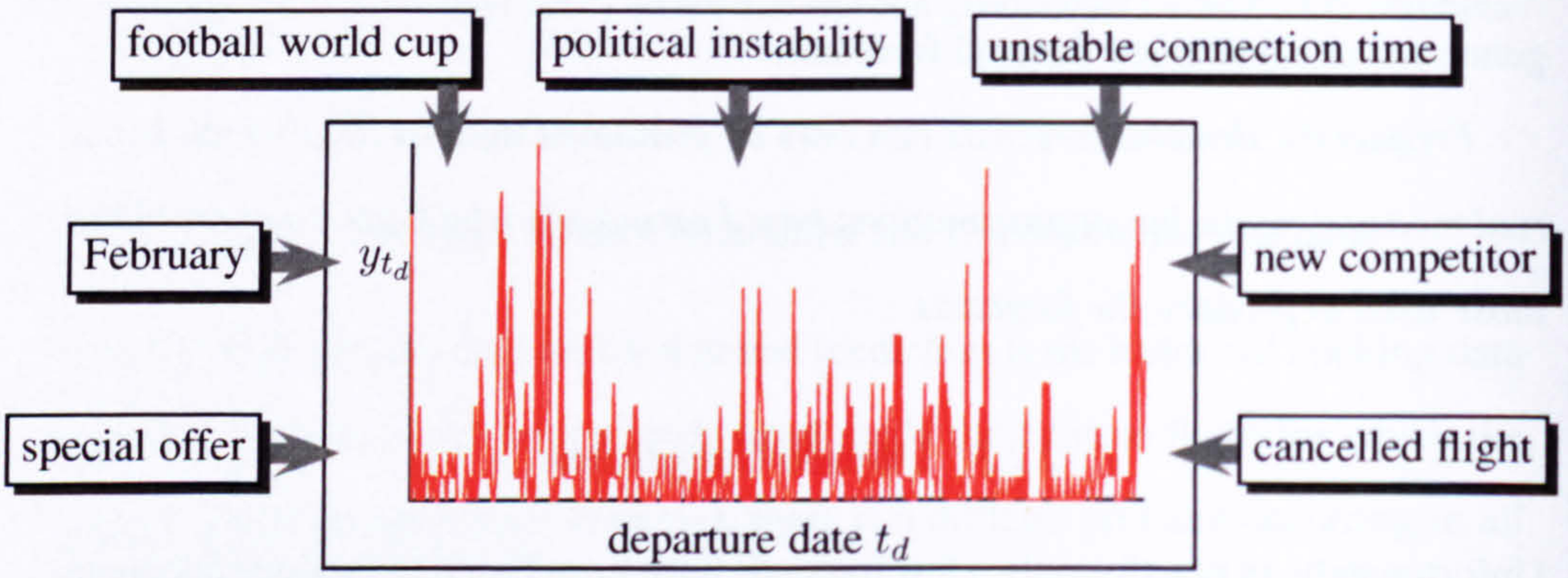

Fig. 8: Example of a demand curve together with potential influences.

The demand components related to the changes are based on abstract terms of attractiveness, attractiveness changes and short term influences. The decomposition model assumes that the changes requiring adaptation can be categorised into two major groups: *permanent changes*, such as market changes or long term schedule changes, and *short term changes*, such as seasonal behavior, events or schedule changes, only influencing some departures.

Attractiveness and Short Term Influences

The (unconstrained) demand at a given departure date depends on many factors. Our model assumes that some of them influence the structure and the amount of the demand in general and are relevant in a long term sense. The most important

of these influences are demographic and economic conditions of the origin and destination of the O&D, the DOW, the time slot (departure and arrival time), the reputation of the airline in the countries of origin and destination and the number and reputation of competitive airlines. These general influences define the attractiveness, that represents the stable world behaviour of the demand.

Definition 2.3 (Attractiveness): Let $t_d \in \mathcal{T}$ be a given departure date and $i \subset \mathcal{ODO} \times \mathcal{F} \times \mathcal{POS}$ indicate a subspace of routing, departure times, fareclass and point of sale. The attractiveness y^{attr} is a demand component that represents the expected unconstrained demand at the subspace i occurring for the departure date t_d if there would be no random noise, no flight specific behaviour and no quickly changing influences, such as season, events and short term schedule changes in the data.

Figure 9 shows an example of total demand y together with an estimation of the attractiveness $\widehat{y}^{attr}$.

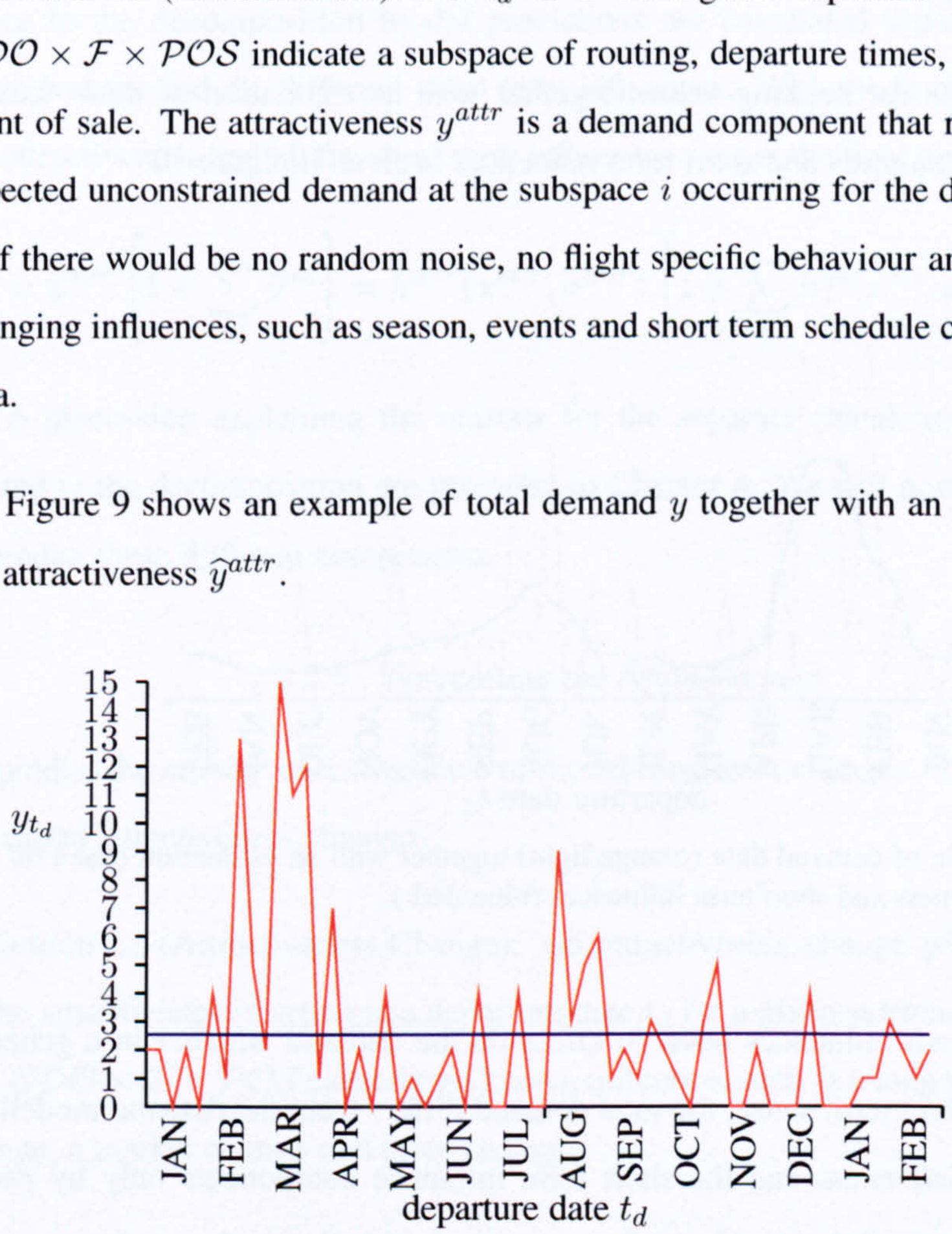

Fig. 9: Example of demand data (orange/light) together with an estimation of the attractiveness (blue/dark).

There are influences on the demand which have only short term effects. Most of them are not known. A short term influence is the known influence on the demand

occurring during a restricted time period and caused, e.g., by seasonal behaviour, events or short term schedule changes.

Definition 2.4 (Short Term Influence): A short term influence y^{sti} is a deviation of the unconstrained demand y from the attractiveness y^{attr} at a given departure t_d caused by a known influence, such as seasonal behaviour, events or short term schedule changes.

An example for booking values together with an estimation of these values based on attractiveness and short term influences is given in figure 10.

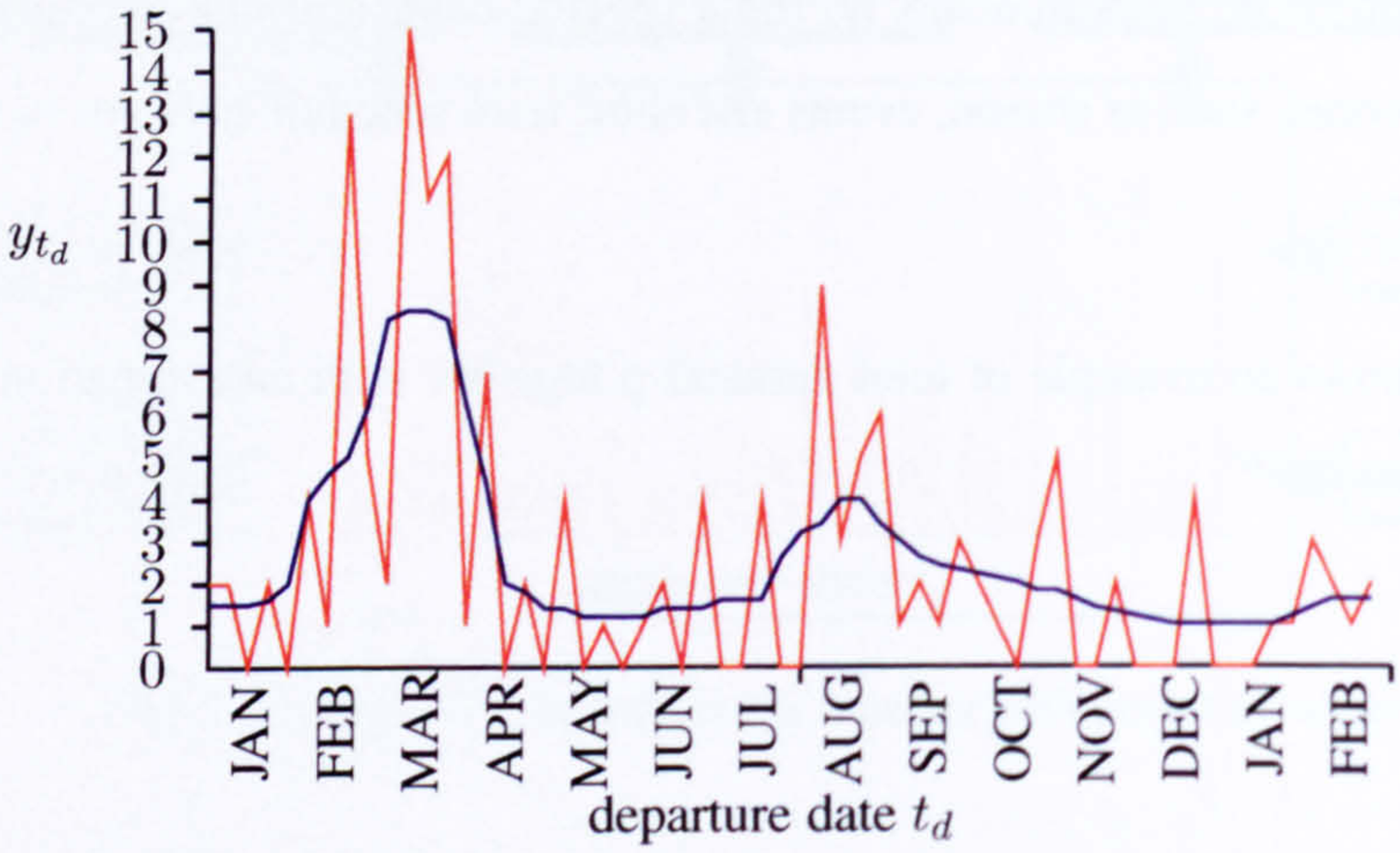

Fig. 10: Example of demand data (orange/light) together with an estimation based on attractiveness and short term influences (blue/dark).

As all known influences have an effect on the demand whether in a general sense or in a short term sense, the total demand differs from the demand modelled using the attractiveness and the short term influence components only by parts which cannot be explained and which are summarised in the random noise term ϵ_y. We can therefore write the demand model similar to (2.5) as

$$y = y^{attr}(1 + \sum_{sti} y^{sti}) + \epsilon_y, \tag{2.6}$$

sti representing an index over all given short term influences. The random noise term ϵ_y is also called "flight specific behaviour", because it can also be interpreted as an unknown influence occurring on specific flights.

2.2.4 The Process of Demand Forecasting

The model in equation (2.6) can be used to generate predictions. In correspondence to the decomposition model predictions are calculated separately for the attractiveness and the different short term influences. With given predictions for the attractiveness and all the short term influences we get the final prediction as:

$$\widehat{y} = \widehat{y}^{attr}\left[1 + \sum_{sti} \widehat{y}^{sti}\right] = h^{attr}(x^{attr}, \phi^{attr})\left[1 + \sum_{sti} h^{sti}(x^{sti}, \phi^{sti})\right]. \quad (2.7)$$

A discussion explaining the reasons for the separate calculation and details related to the decomposition are provided in Chapter 4. We will now discuss how to predict these different components.

2.2.5 Forecasting the Attractiveness

To predict the attractiveness we have to model long term changes. These changes are called attractiveness changes.

Definition 2.5 (Attractiveness Change): An attractiveness change y^{ac} is a change of the attractiveness starting at a departure date t_d for a given subspace $i \subset \mathcal{ODO} \times \mathcal{F} \times \mathcal{POS}$ caused by a known influence, such as a long term schedule change, a market change or a price change.

Based on this definition, the attractiveness for a future departure date t_d can be predicted with an estimation of the current attractiveness $y_{t_p}^{attr}$ with t_p denoting the process date and all expected attractiveness changes $\widehat{y}^{ac}$ expected between $t_p + 1$ and t_d by

$$\widehat{y}_{t_d}^{attr} = \widehat{y}_{t_p}^{attr} + \sum_{ac} \widehat{y}^{ac} \tag{2.8}$$

As the prediction of attractiveness changes is not the focus of this thesis and because of commercial sensitivity we will not go into detail concerning the prediction of attractiveness changes. Some details can be found in [Riedel 03]. Only test data without relevant attractiveness changes have been chosen for experiments so that in the following we will assume

$$\widehat{y}_{t_d}^{attr} = \widehat{y}_{t_p}^{attr} \tag{2.9}$$

for all future departure dates t_d.

The current attractiveness $y_{t_p}^{attr}$ is estimated based on the series of historical decomposed data that represent previous attractiveness estimations. Let us assume that we have T historical decomposed demand data y_t^{attr} given for a time period $t < t_p \in \mathcal{T}$ as well as historical attractiveness changes y^{ac}.

For each element of t we can calculate an approximation for $y_{t_p}^{attr}$ by

$${}^{t}\widehat{y}_{t_p}^{attr} = y_t^{attr} + \sum_{ac} \widehat{y}^{ac} \tag{2.10}$$

with ac containing all attractiveness changes between $t+1$ and t_p.

Calculating this approximation for different historical departures t leads to the generation of a time series (related to the time index t) containing different approximations for $\widehat{y}_{t_p}^{attr}$. This time series enables us not only to generate a prediction for the attractiveness with reduced approximation error, but also to determine unexpected small long term attractiveness changes corresponding to slow and regular changes of the attractiveness which can, e.g., be represented as a long term trend.

We can use different function spaces $h^{attr}(x^{attr}, \phi^{attr})$ in order to model $f^{attr}(x^{attr})$. The most successful approaches that have been found are very simple and stable approaches originating from the theory of time series forecasting like

the constant function

$$h_1^{attr}(x^{attr}, \phi^{attr}) = \phi_0^{attr}, \tag{2.11}$$

with learning ϕ_0^{attr} based on the series ${}^t\widehat{y}_{t_p}^{attr}$ corresponding to the methodology of simple exponential smoothing [Brown 63] or with the simple average

$$\phi_0^{attr} = \frac{1}{T}\sum_t {}^t\widehat{y}_{t_p}^{attr} \tag{2.12}$$

over the given set of T estimations based on historical data. We can also assume a linear trend

$$h_2^{attr}(x^{attr}, \phi^{attr}) = \phi_0^{attr} + \phi_1^{attr} * (t_d - t_p), \tag{2.13}$$

with parameters learned using the Brown method [Brown 63] or linear regression [Granger 86]. More sophisticated approaches like ARMA models [Brockwell 87] are possible as well, but have shown worse results because of the high noise in the data in connection with decomposition errors and short history pools caused by quickly changing environments.

2.2.6 Learning and Forecasting Short Term Influences

The currently modelled short term influences correspond to the (periodic) seasonal behaviour, special events (like fairs, conferences and holidays), short term schedule changes (sometimes caused by events), short term market changes and short term price changes.

As we have found that seasonal impacts in the demand are especially relevant for the forecast accuracy we will now describe the methods used to predict seasonal behaviour. All other impacts have been eliminated for our experiments in choosing a testset without relevant schedule changes , market changes or price changes. Relevant event periods have been excluded from the forecast evaluation as well as from the history pools.

Two general approaches are used for seasonal forecasting:

- the season is predicted based on the behaviour of the past years or
- the season is predicted based on the given booking data that have already occurred for a future departure.

The available input information x^{season} for seasonal predictions contains the calendar week cw to be predicted corresponding to the ISO standard as well as information about the current demand $y_{t_{d,\tau}}$ of the future departure t_d measured at time t_p and estimations of the attractiveness at the current moment $\widehat{y}^{attr}_{t_{d,\tau}}$ and at the departure $\widehat{y}^{attr}_{t_d}$ (both estimated based on historical departures). Historical seasonal factors y^{season}_{cw} are used as input information as well.

In the following subsections we refer always to seasonal predictions, we will not write the upper index "^{season}" in order to increase readability.

Predicting seasonal behaviour based on decomposed historical demand data

Let us assume we have weekly decomposed historical seasonal factors y_t given over several years. The data can be related to a special day of week or to aggregated demand of the whole week.

Figure 11 shows y_t depending on the calendar week cw together with two examples for learned seasonal factors based on this data. They are both based on estimations of the seasonal factors y_{cw}

$$\widehat{y}_{cw} = E(min(max(\frac{1}{2\phi_J + 1} \sum_{j=-\phi_J}^{\phi_J} [y_{cw+j}], \phi_{low}), \phi_{high})). \tag{2.14}$$

which are calculated per calendar week for each year. The estimates $\widehat{y}_{cw}$ are then averaged over the two years in order to represent an estimation for the total historical behaviour.

The two examples of learning the seasonal behaviour differ concerning the used parameters ϕ_J, ϕ_{low} and ϕ_{high}. Parameter ϕ_J represents the size of the neighbourhood of a calendar week that is taken into account for the estimation of the seasonal

behaviour. A bigger value means a noise reduction and the generation of smoother seasonal curves. But it also represents a restriction in modelling quick changes in the seasonal behaviour between neighboured weeks. The other two parameters ϕ_{low} and ϕ_{high} are also used for stabilisation purposes. They represent a lower and an upper limit to the expected seasonal factors. Strong restrictions again mean a noise reduction and allow, for instance, the avoidance of a "zero season" assumption in case of no historical bookings measured at the ODIFPOS level for a given calendar week, but represent also a restriction in flexibility of the learned seasonal factors. Improved versions of learning use simple exponential smoothing over the different years instead of taking the simple average in order to enforce the impact of newer data and consider also different impacts depending on the neighbourhood distance j.

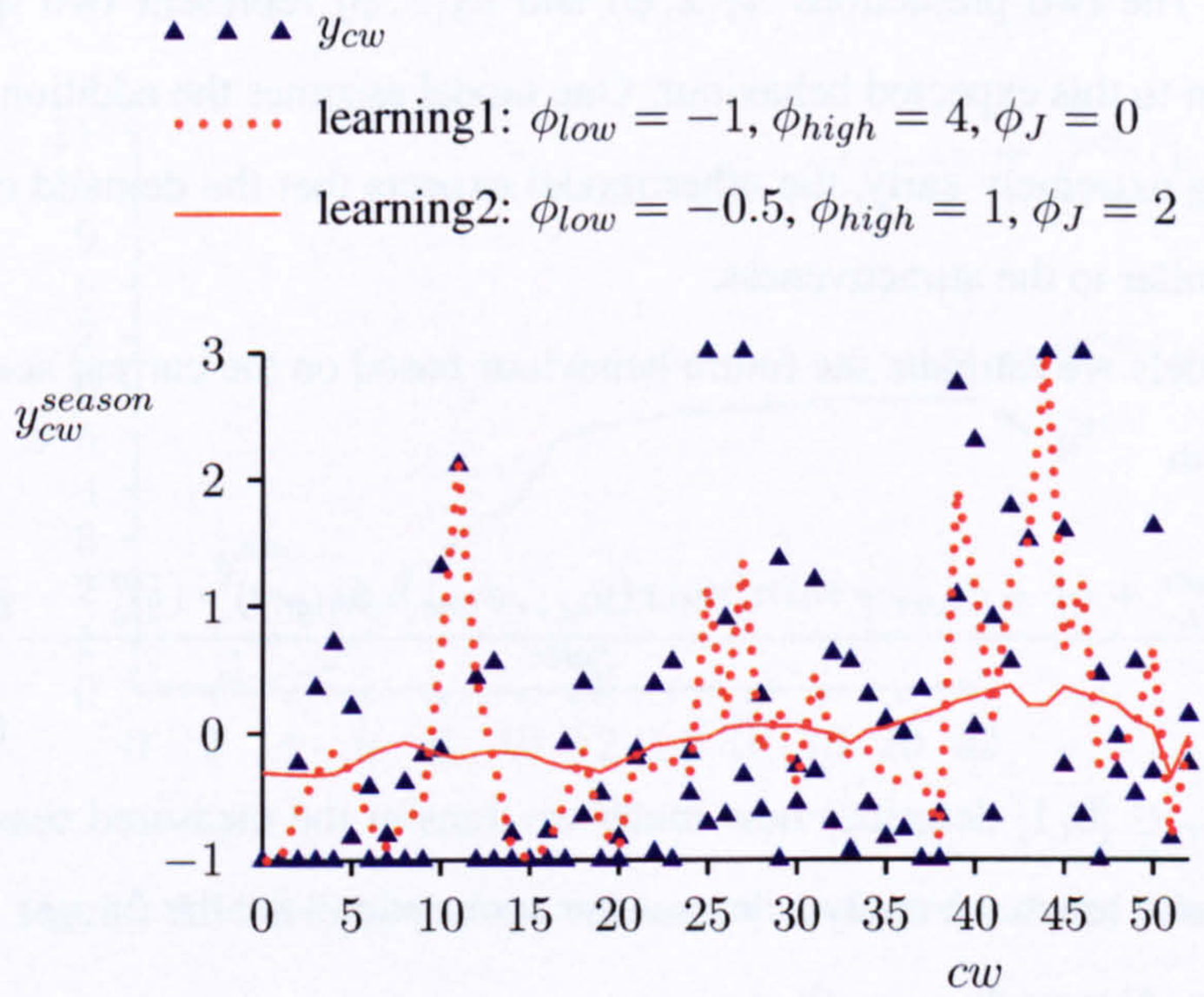

Fig. 11: Measured seasonal factors during 2 years with two learned curves $\widehat{y}_{cw}$. Learning 1 is carried out with the parameters that allow very high flexibility. Learning 2 is carried out with the parameters that generate a more stable curve.

The learned seasonal factors can be used in order to generate predictions for future seasonal behaviour. We have to consider already measured unconstrained

bookings $y^{unc}_{t_{d,\tau}}$ for a future departure t_d that should be predicted. We define

$$h_1(x,\phi)_{t_d} = \frac{y^{unc}_{t_{d,\tau}} + (1 + \widehat{y}_{cw}) * (\widehat{y}^{attr}_{t_d} - \widehat{y}^{attr}_{t_{d,\tau}})}{\widehat{y}^{attr}_{t_d}}. \tag{2.15}$$

Predicting seasonal behaviour based on current booking data

Seasonal effects may not only be predicted based on the past years observations. Current booking data gives additional indicators about seasonal behaviour as well, especially a short time prior to departure. Two models are used in order to predict the season based on current demand. It has been observed that the seasonal behaviour affects not only the additional demand caused by the season at the departure, but also the time when the demand occurs. So we could, for instance, observe a clear tendency that the demand of the Economy compartment occurs earlier in high seasons. The two predictions $h_2(x,\phi)$ and $h_3(x,\phi)$ represent two special cases in relation to this expected behaviour. One model assumes the additional demand occurring extremely early, the other model expects that the demand occurs in a manner similar to the attractiveness.

In both models we estimate the future behaviour based on the current seasonal impact $y_{t_{d,\tau}}$ with

$$h(x,\phi)_{t_d} = \frac{y^{unc}_{t_{d,\tau}} + (1 + \phi_{corr} * min(max(y_{t_{d,\tau}}, \phi_{low}), \phi_{high})) * (\widehat{y}^{attr}_{t_d} - \widehat{y}^{attr}_{t_{d,\tau}})}{\widehat{y}^{attr}_{t_d}} \tag{2.16}$$

Parameter $\Phi_{corr} \in [0,1]$ describes how much we transfer the measured season to the future and how much we apply a "no season assumption" for the future.

Model $h_2(x,\phi)$ uses $\Phi_{corr} = 0$:

$$h_2(x,\phi)_{t_d} = \frac{y^{unc}_{t_{d,\tau}} + (\widehat{y}^{attr}_{t_d} - \widehat{y}^{attr}_{t_{d,\tau}})}{\widehat{y}^{attr}_{t_d}}. \tag{2.17}$$

This means that it is expected that the complete additional or missing demand has already occurred.

The third model expects a seasonal factor for the future demand that corre-

sponds to a stabilised version of the currently observed seasonal factor $y_{t_{d,\tau}}$, we set $\Phi_{corr} = 1$ and get

$$h_3(x,\phi)_{t_d} = \frac{y^{unc}_{t_{d,\tau}} + (1 + min(max(y_{t_{d,\tau}}, \phi_{low}), \phi_{high})) * (\widehat{y}^{attr}_{t_d} - \widehat{y}^{attr}_{t_{d,\tau}})}{\widehat{y}^{attr}_{t_d}} \tag{2.18}$$

Figure 12 shows an example in order to illustrate the idea of the second and the third model. The blue/dark lines show the current unconstrained booking values y^{unc} together with an estimation of the attractiveness $\widehat{y}^{attr}$. The difference between the two is used in an additive or multiplicative manner in order to estimate the seasonal behaviour in future dcps τ. The resulting total forecasts are shown in orange/light lines. It can be seen that the additive adaptation $h_2(x,\phi))$ corresponds to the application of a constant offset to the attractiveness estimation, the multiplicative adaptation $h_3(x,\phi))$ stretches the future reference values.

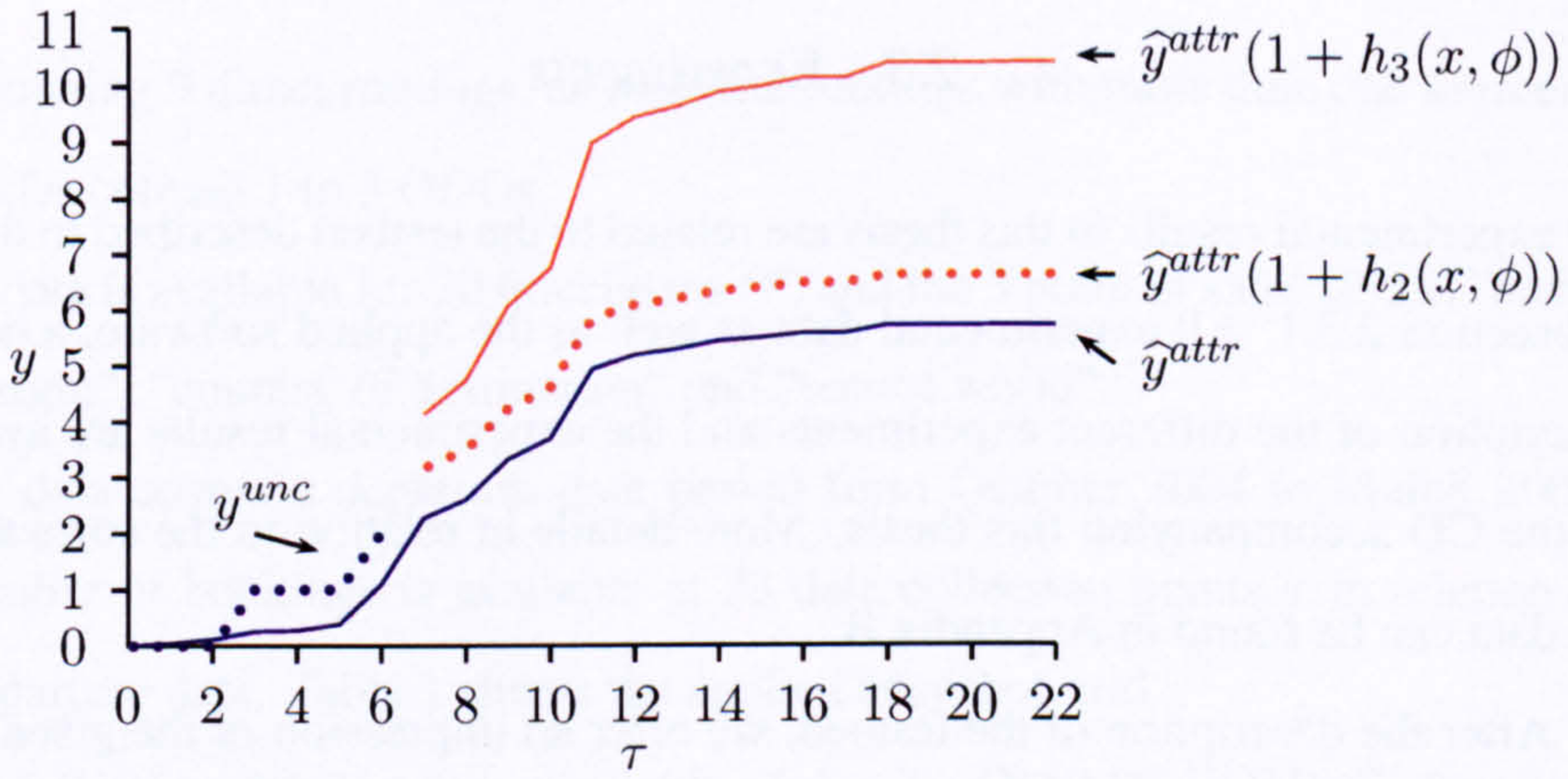

Fig. 12: Additive and multiplicative interpretation of seasonal behaviour.

The current combination of the predictions

The current version of the Revenue Management product *ProfitLine.Yield/O&D* uses a combination

$$h(x,\phi) = \sum_{i=1}^{3} G_i(.) * h_i(x,\phi) \tag{2.19}$$

in order to generate a final prediction for the expected seasonal behaviour.

The weight functions $G_1(.)$ to $G_3(.)$ each return values between 0 and 1 and fulfil

$$\sum_{i=1}^{3} G_i(.) = 1 \tag{2.20}$$

for all possible input configurations. They depend on different input values in a nonlinear manner and are fixed in the sense that they do not contain learned parameters. Because of commercial aspects neither the concrete functions nor their inputs are provided in this thesis, but we can state that the functions have been the result of an extensive data analysis and that they have been constantly tuned during the past years.

2.3 Experiments

All experimental results in this thesis are related to the testbed described in the next Subsection 2.3.1. All experimental data as well as the applied software, a detailed description of the different experiments and the experimental results are available on the CD accompanying this thesis. More details in relation to the software and the data can be found in Appendix B.

After the description of the testbed, we offer an impression of the given booking and availability data by providing some statistical properties in Subsection 2.3.2. In Subsection 2.3.3 different individual forecast methods are experimentally compared.

2.3.1 Testbed Description

The chosen testbed includes data of **10 representative O&Ds** consisting of

- 2 transatlantic O&Ds from Europe to America

τ	0	1	2	3	4	5	6	7	8	9	10	11
$t_d - t_p$	350	182	140	126	98	70	56	49	42	35	28	21
τ	12	13	14	15	16	17	18	19	20	21	22	
$t_d - t_p$	14	12	10	8	6	5	4	3	2	1	0	

Tab. 1: DCP Grid: the table shows at which days prior to departure $t_d - t_p$, with t_d the departure date and t_p the process date, new booking and availability information is available and new forecasts are calculated. Each of these "data collection points" (dcp) are described by an index τ with $\tau = 0$ the earliest time of forecasting about one year prior to departure and $\tau = 22$ the day of the departure.

- 1 intercontinental O&D from Europe to Asia
- 1 intercontinental O&D from Europe to Africa
- 1 intercontinental O&D from Asia to America
- 5 European O&Ds

and containing 9 direct routings, as well as 2 routings with more than one segment. The O&Ds contain 1 to 3 ODOs.

All data is available for 20 fareclasses (F) and the 3 point of sales (POS) "country of origin", "country of destination" and "rest of world".

The data covers a departure date period form October 2004 to March 2007. The number of bookings is available at 23 data collection points τ in relation to each departure date. Table 1 shows the applied snapshot grid.

The following data has been available per level i=ODO DOW F POS, departure date t_d and days prior to departure τ:

- the number of individual bookings $b_{i,t_{d,\tau}}$ and
- the availability information $av_{i,t_{d,\tau}}$ with $av_{i,t_{d,\tau}} = 1$ if the booking class has been closed at time $t_{d,\tau}$ and $av_{i,t_{d,\tau}} = 0$ otherwise.

For confidentiality reasons, the data is presented in a disguised form. Different O&Ds and ODOs as well as fareclasses are represented by an artificial indicator. In

POS	F	DOW	ODO	DW	*DCP_0*	...	*DCP_22*
2	9	1	1	1	0		3
2	9	1	2	1	2		8

Tab. 2: Example for the structure of the provided booking data. The first 5 columns contain the description of Point of Sale, Fareclass, Day of Week, ODO and Departure Week. The following columns contain the number of total bookings for each data collection point (dcp), so that the last column contains the number of bookings at the day of departure.

order to enable history pooling per day of the week, the departure date information is provided as a pair of departure week and day of week $t_d = (dw_d, dow_d)$. The first three fareclasses 0 to 2 represent the First Class Compartment, the following five fareclasses belong to the Business Compartment and all other fareclasses belong to the Economy compartment. The fareclasses are ordered corresponding to their nesting [McGill 99], which can be interpreted as if they were ordered in relation to the quality of the corresponding product (from more expensive and more flexible products to cheaper and less flexible products).

Table 2 shows an example of the representation of the data. The complete data tables are available on the CD accompanying this PhD thesis.

Some further characteristics of the data:

- Days without any values indicated in the files have not been valid departures (no flight on this day).
- The booking data is so called gross bookings. This means that even if some of these bookings have been subsequently cancelled, they are counted in the booking curves without considering this fact.
- The O&Ds have been chosen in a manner that there have been no significant schedule or market changes in order to simplify the experiments.

2.3.2 Statistical Properties

The software allows a determination and visualisation of some common statistical properties of the data in relation to all dimensions of the data (like Fareclass, Point of Sale, Day of Week, Departure Week and so on). Experiment 1 (see Appendix B.6.1) contains an interface of data loading. Then it is described how the data can be visualised and basic statistical properties can be determined. These properties contain the sum, average value, standard deviation, number of missing data as well as the number of zeros in relation to each value of each data dimension.

The following Figures 13 to 16 show some of the most relevant distributions of the input data:

- averaged number of bookings per
  - departure week dw at the time of departure,
  - data collection point τ,
  - fareclass F at the time of departure,
  - point of sale POS at the time of departure and
- averaged availability information per fareclass F (over all τ)

Detailed information about the statistical distributions are provided via the experimental results of Experiment 1.

The Figures show that the number of bookings is very low, which illustrates the fact that we have the problem of small number predictions. In contrast to a complete Revenue Management system, most of our O&Ds correspond to direct routings, the average number of bookings in a complete system would still be much lower. Most of the passengers book in the Economy compartment. This can be clearly seen in Figure 15. The Figure also shows that we do not have a balanced distribution of demand in different fareclasses. Figure 16 shows the tendency that the average availability decreases for higher fareclasses (per compartment). This effect can be explained with the strategy of closing cheaper fareclasses first.

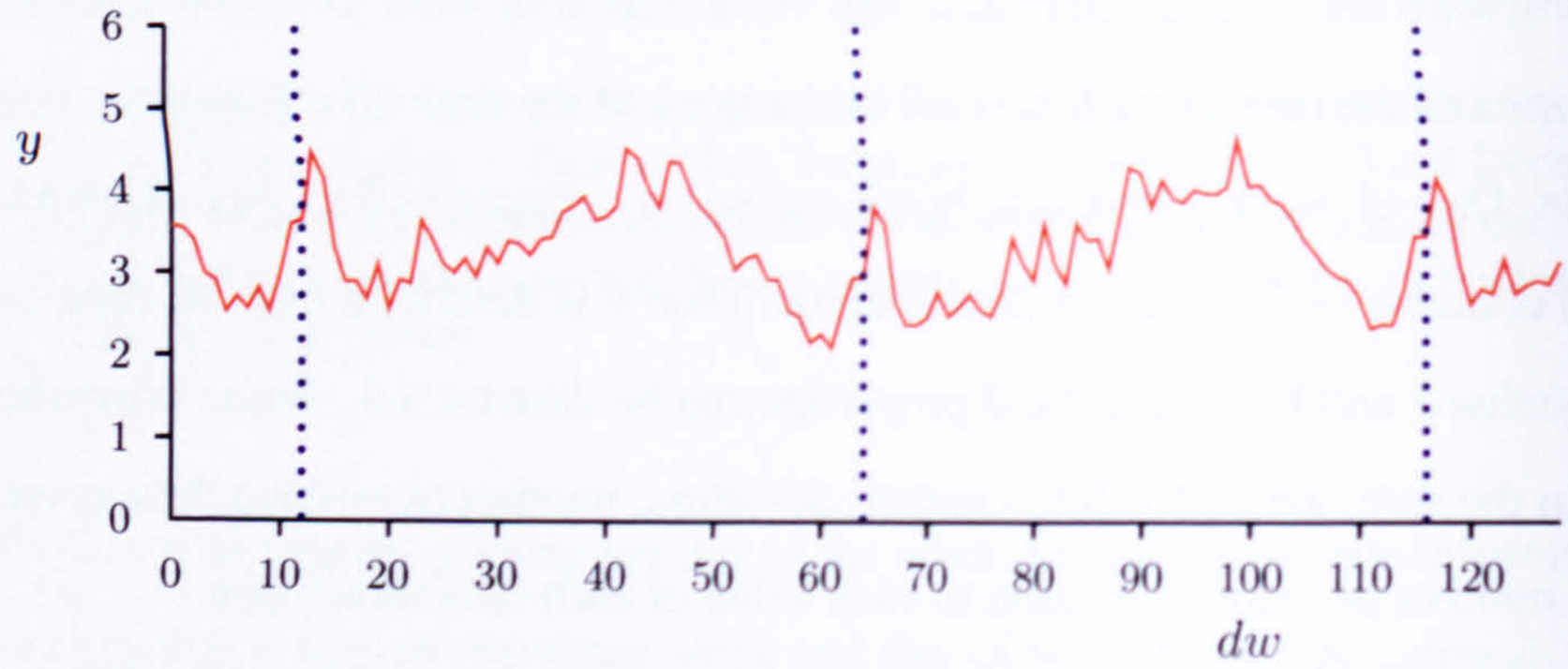

Fig. 13: Average bookings per departure week dw. The dotted lines indicate the yearly cycles.

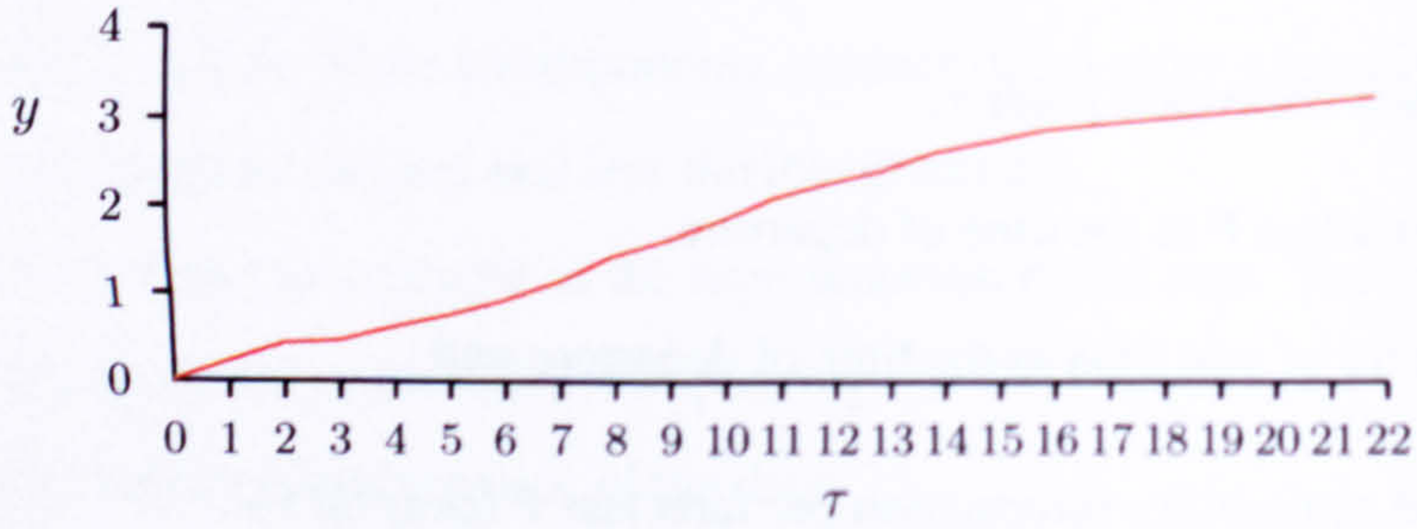

Fig. 14: Average bookings per data collection point τ.

2.3.3 Individual Forecast Performance

A pool of promising individual forecast methods has already been available at Lufthansa Systems as part of the Forecasting Kernel. It contains different methods for prediction of the attractiveness as well as for prediction of seasonal effects. Different methods to adapt to flight specific behaviour based on incoming bookings are available as well. The six most promising methods are described in Table 3. For details related to the methods see Sections 2.2.5 and 2.2.6.

Figure 17 shows an example of real data at the ODIFPOS level together with predictions ${}^{0}\widehat{y}$ to ${}^{6}\widehat{y}$ calculated at time $\tau = 5$ (70 days prior to departure).

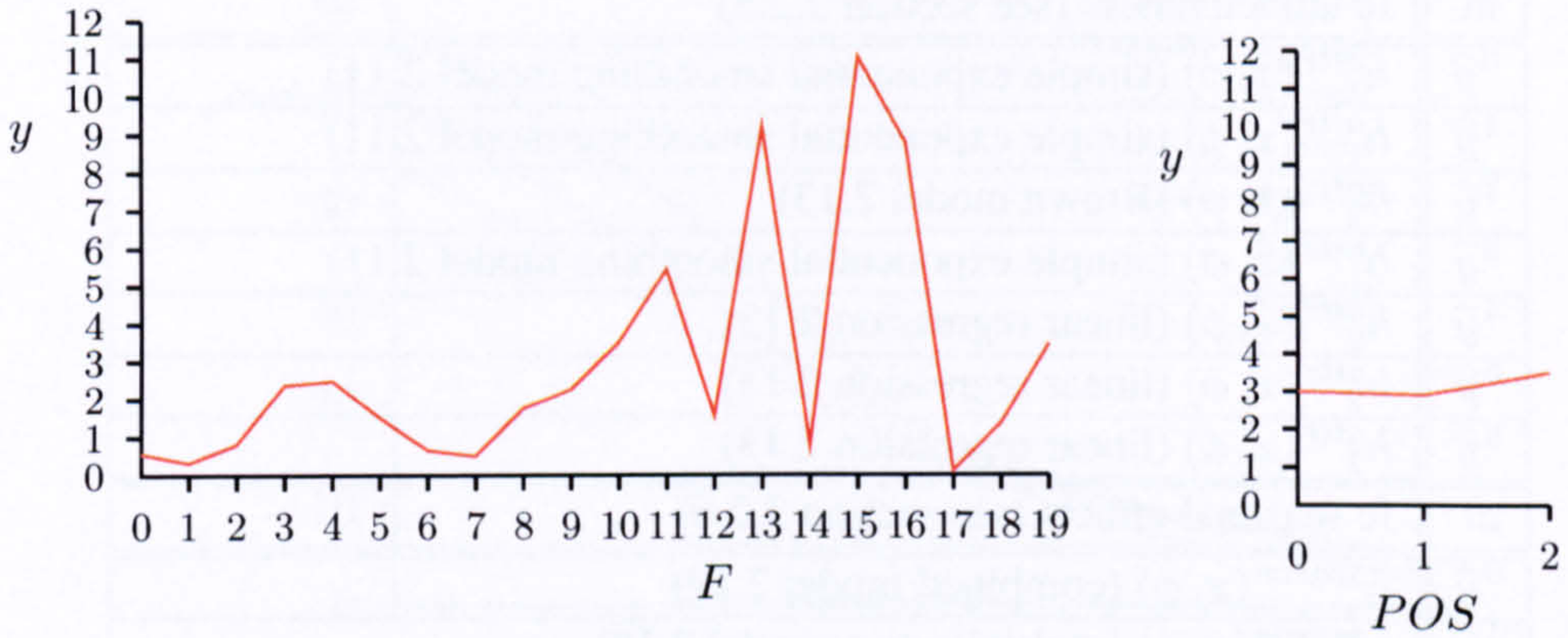

Fig. 15: Average bookings per fareclass F and point of sale POS.

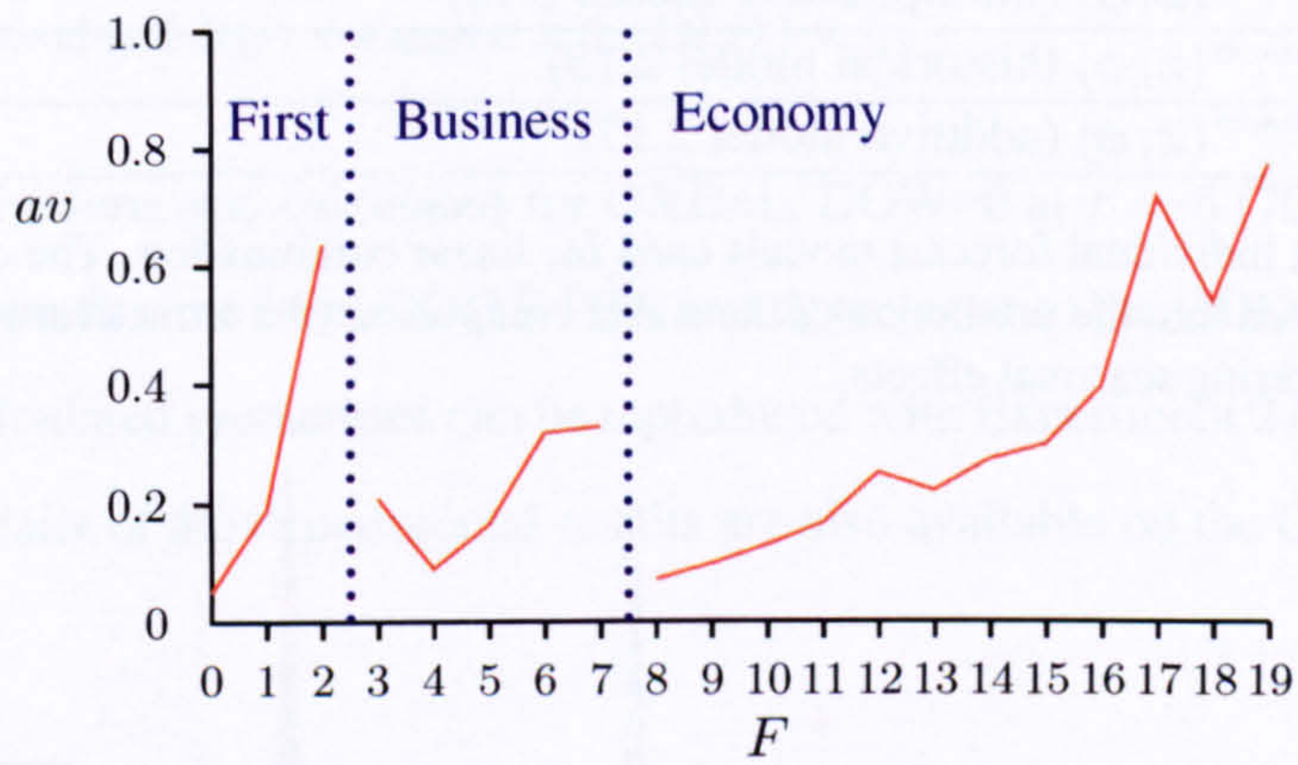

Fig. 16: Averaged availability (0=open, 1=closed) per fareclass: The figure shows quite well the tendency that within compartments cheaper fareclasses are closed before more expensive fareclasses. The dotted lines indicate the different compartments.

After having produced the individual forecasts, the forecast errors have been analysed. Tables 4 and 5 and Figure 18 illustrate the errors predicting the final dcp $\tau = 22$ from each dcp τ (x axis) on the fine level (ODOFPOS) and the high level (ODO).

It can be seen that method 0 is the best performing method. In the following chapters we will refer to these forecasts as "best individual forecast ${}^0\widehat{y}$" and use it as a baseline for the evaluation of combined forecast quality.

As error covariance values have a relevance for combination (this will be discussed in the following chapters), Tables 2.3.3 show examples of error covariance

m	fc attractiveness (see section 2.2.5)
${}^{0}\widehat{y}$	$h_1^{attr}(x,\phi)$ (simple exponential smoothing model 2.11)
${}^{1}\widehat{y}$	$h_1^{attr}(x,\phi)$ (simple exponential smoothing model 2.11)
${}^{2}\widehat{y}$	$h_2^{attr}(x,\phi)$ (Brown model 2.13)
${}^{3}\widehat{y}$	$h_1^{attr}(x,\phi)$ (simple exponential smoothing model 2.11)
${}^{4}\widehat{y}$	$h_2^{attr}(x,\phi)$ (linear regression 2.13)
${}^{5}\widehat{y}$	$h_2^{attr}(x,\phi)$ (linear regression 2.13)
${}^{6}\widehat{y}$	$h_2^{attr}(x,\phi)$ (linear regression 2.13)

m	fc seasonal effects (see section 2.2.6)
${}^{0}\widehat{y}$	$h^{season}(x,\phi)$ (combined model 2.19)
${}^{1}\widehat{y}$	$h_3^{season}(x,\phi)$ (multiplicative model 2.18)
${}^{2}\widehat{y}$	$h_3^{season}(x,\phi)$ (multiplicative model 2.18)
${}^{3}\widehat{y}$	$h_2^{season}(x,\phi)$ (additive model 2.17)
${}^{4}\widehat{y}$	$h_3^{season}(x,\phi)$ (multiplicative model 2.18)
${}^{5}\widehat{y}$	$h_1^{season}(x,\phi)$ (historical model 2.15)
${}^{6}\widehat{y}$	$h_2^{season}(x,\phi)$ (additive model 2.17)

Tab. 3: Different individual forecast models used for linear combination. The description is separated into the prediction of the stable component (the attractiveness) and the parts covering seasonal effects.

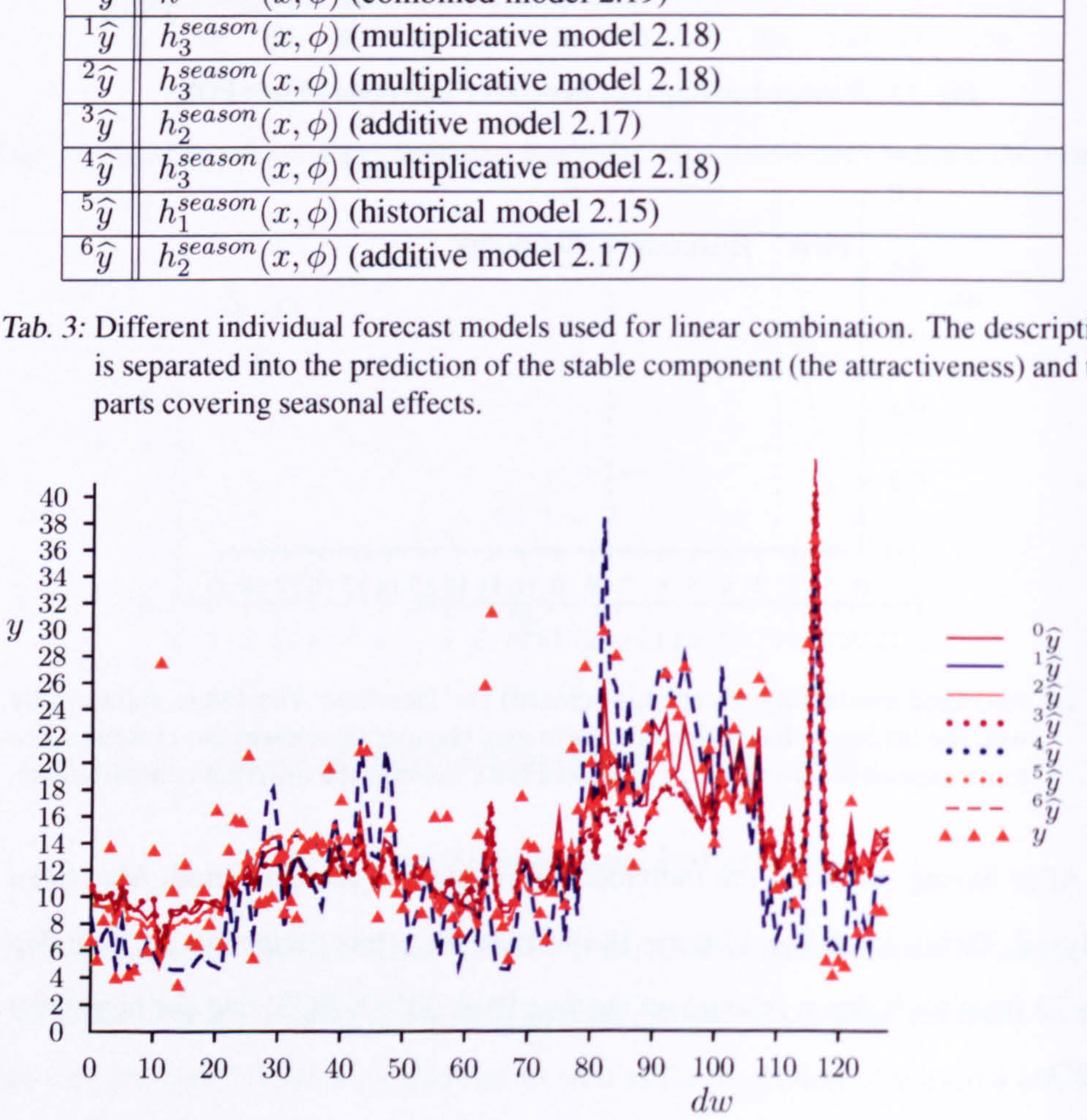

Fig. 17: Forecasts ${}^{0}\widehat{y}$ to ${}^{6}\widehat{y}$ generated for O&D=0, ODO=0, DOW=all (sum), Fareclass=16, POS=0, $\tau = 6$ together with the unconstrained demand y. The x-axis represents different departure weeks. The y- axis represents the demand.

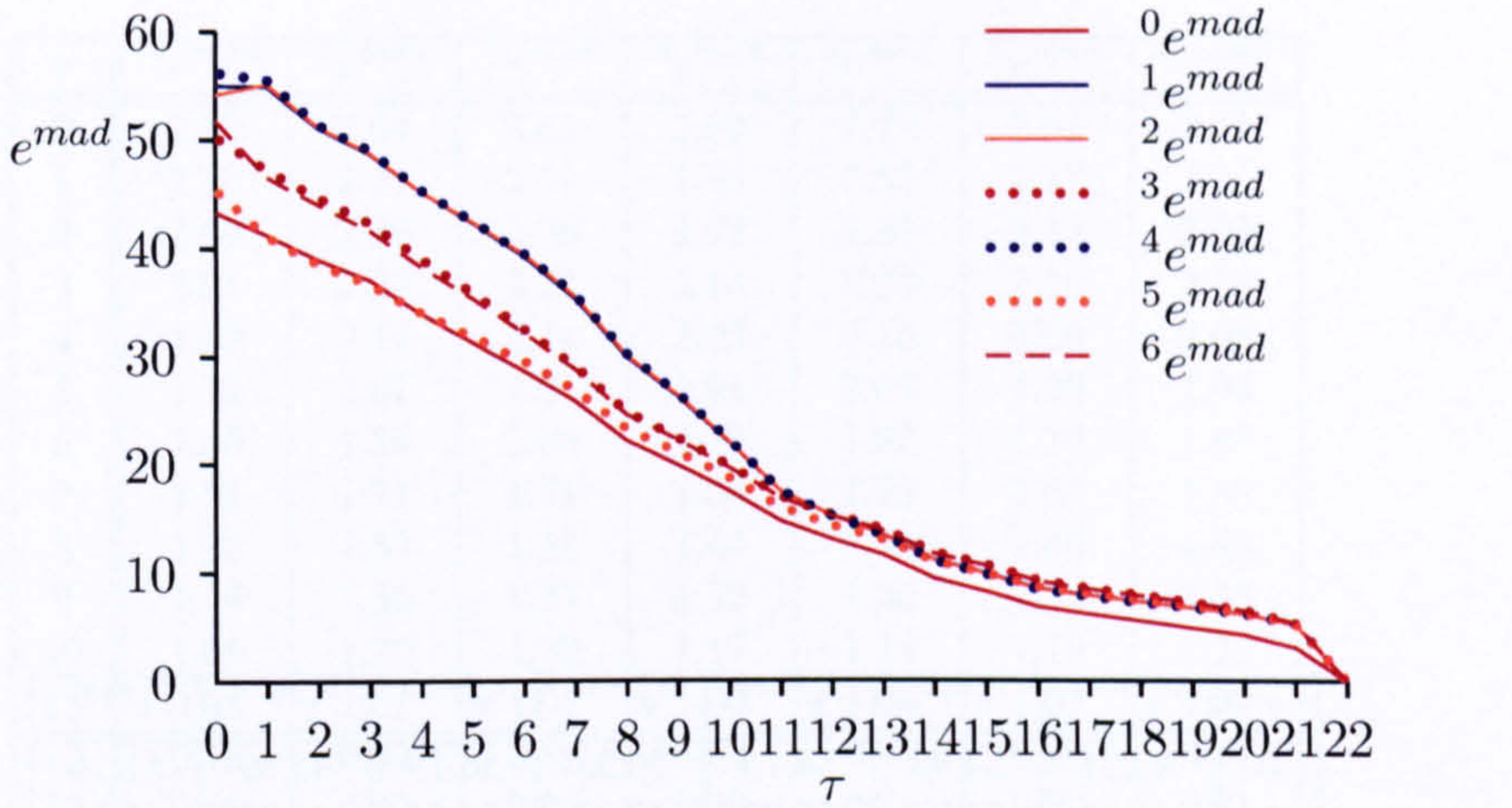

Fig. 18: Graphical representation of the mean absolute error e^{mad} per individual forecast method and dcp τ measured at the ODO level.

values of the forecasts calculated for O&D=0, DOW=0 at $\tau = 5$ (70 days prior to departure) on the fine level ODO F POS and aggregated to the ODO level.

The calculated predictions can be reproduced with Experiment 2 (see Appendix B.6.2). Details of the experimental results are also available on the CD.

τ	${}^{0}e^{mad}$	${}^{1}e^{mad}$	${}^{2}e^{mad}$	${}^{3}e^{mad}$	${}^{4}e^{mad}$	${}^{5}e^{mad}$	${}^{6}e^{mad}$
0	43.31	54.98	54.13	50.23	56.19	45.39	51.76
1	41.19	54.87	55.07	47.20	55.51	41.06	46.52
2	38.88	51.01	51.10	44.61	51.33	38.45	43.96
3	36.99	48.62	48.68	42.27	48.96	36.60	41.57
4	33.90	45.11	45.13	38.95	45.28	34.00	38.20
5	31.25	42.29	42.36	35.82	42.43	31.80	35.33
6	28.61	39.20	39.36	32.49	39.41	29.50	32.13
7	25.86	35.31	35.48	29.05	35.62	26.89	28.77
8	22.19	29.94	30.07	24.86	30.21	23.28	24.61
9	19.90	26.30	26.40	22.40	26.55	21.09	22.25
10	17.48	22.02	22.08	19.82	22.21	18.70	19.65
11	14.74	17.48	17.48	16.99	17.45	16.00	16.86
12	13.04	15.23	15.21	15.28	15.17	14.30	15.18
13	11.61	13.46	13.43	13.94	13.38	13.00	13.85
14	9.41	11.12	11.08	11.79	11.01	10.96	11.70
15	8.20	9.88	9.83	10.68	9.79	9.88	10.60
16	6.84	8.50	8.46	9.24	8.39	8.62	9.18
17	6.16	7.86	7.83	8.51	7.75	7.98	8.45
18	5.53	7.31	7.28	7.89	7.20	7.40	7.84
19	4.87	6.74	6.72	7.20	6.74	6.86	7.20
20	4.26	6.17	6.15	6.50	6.17	6.24	6.50
21	3.11	5.21	5.20	5.34	5.23	5.24	5.36
22	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Tab. 4: Mean absolute error e^{mad} per individual forecast method and dcp τ measured at the high level ODO.

τ	${}^0e^{mad}$	${}^1e^{mad}$	${}^2e^{mad}$	${}^3e^{mad}$	${}^4e^{mad}$	${}^5e^{mad}$	${}^6e^{mad}$
0	2.50	2.64	2.65	2.62	2.72	2.61	2.71
1	2.20	2.51	2.51	2.33	2.52	2.27	2.34
2	2.09	2.36	2.36	2.22	2.37	2.17	2.23
3	2.01	2.27	2.27	2.14	2.29	2.10	2.16
4	1.89	2.14	2.14	2.03	2.16	2.00	2.05
5	1.78	2.01	2.01	1.91	2.05	1.90	1.94
6	1.66	1.88	1.88	1.79	1.92	1.79	1.83
7	1.51	1.71	1.71	1.64	1.75	1.65	1.68
8	1.32	1.50	1.51	1.44	1.54	1.46	1.49
9	1.19	1.36	1.37	1.32	1.40	1.34	1.35
10	1.05	1.20	1.20	1.17	1.23	1.19	1.20
11	0.88	1.01	1.01	1.00	1.04	1.01	1.02
12	0.78	0.89	0.89	0.89	0.91	0.90	0.91
13	0.68	0.80	0.80	0.79	0.81	0.80	0.81
14	0.55	0.66	0.66	0.66	0.67	0.67	0.67
15	0.48	0.58	0.58	0.59	0.59	0.59	0.60
16	0.39	0.50	0.50	0.50	0.50	0.50	0.51
17	0.35	0.45	0.45	0.46	0.45	0.46	0.46
18	0.30	0.41	0.41	0.41	0.41	0.41	0.41
19	0.25	0.36	0.36	0.36	0.36	0.36	0.37
20	0.20	0.31	0.31	0.31	0.31	0.31	0.31
21	0.12	0.22	0.22	0.22	0.23	0.23	0.23
22	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Tab. 5: Mean absolute error e^{mad} per individual forecast method and dcp τ measured per ODO F POS.

	0	1	2	3	4	5	6
0	**2.47**	2.55	2.33	2.55	2.38	2.65	2.66
1	2.55	**2.80**	2.59	2.81	2.57	2.91	2.93
2	2.33	2.59	**2.58**	2.57	2.41	2.62	2.60
3	2.55	2.81	2.57	**2.82**	2.48	2.94	2.95
4	2.38	2.57	2.41	2.48	**3.02**	2.67	2.59
5	2.65	2.91	2.62	2.94	2.67	**3.07**	3.10
6	2.66	2.93	2.60	2.95	2.59	3.10	**3.13**

	0	1	2	3	4	5	6
0	**0.16**	0.17	0.17	0.17	0.12	0.16	0.16
1	0.17	**0.18**	0.19	0.18	0.13	0.18	0.18
2	0.17	0.19	**0.19**	0.18	0.13	0.19	0.18
3	0.17	0.18	0.18	**0.20**	0.09	0.18	0.19
4	0.12	0.13	0.13	0.09	**0.34**	0.13	0.09
5	0.16	0.18	0.19	0.18	0.13	**0.18**	0.18
6	0.16	0.18	0.18	0.19	0.09	0.18	**0.19**

Tab. 6: Error covariances for O&D=0, ODO=0, DOW=4. The upper table shows the covariances at the low level for fareclass=13 and POS=0, the table below shows the error covariances corresponding to forecasts aggregated over all farclasses and point of sales.

3. FORECAST COMBINATION MODELS

3.1 Introduction to Forecast Combination

> *Combining forecasts is a well-established procedure for improving forecast accuracy which takes advantage of the availability of both multiple information and computing resources of data-intensive forecasting.* (Bunn, [Bunn 89])

The general idea of forecast combination is quite simple. In order to profit from the information of different forecast models, not a single prediction is produced, but a whole set of forecasts which are then aggregated in a second step.

The superiority of this approach has been proved theoretically and experimentally for a lot of applications. To cite just one of the most common examples: Makridakis et al [Makridakis 82] carried out an extended study to compare forecast quality of different forecast methods including two different approaches of forecast combination. The study showed clearly that related to forecasts made for about 1000 time series the combining approaches outperformed on average the individual forecast models. Other studies [Makridakis 93][Russell 87] were carried out with the same results so that the combination of forecasts became a scientifically acknowledged procedure.

In this section we describe what combination of forecasts means and have a short discussion why it works. Different approaches to forecast combination are then presented in more detail in the following sections.

What is forecast combination?

Forecast combination is a procedure of generating one (combined) forecast

based on different individual forecasts and potentially additional information. It can be seen as a fusion procedure, represented by a function F, which receives as inputs a set of M individual forecasts $\{{}^m\widehat{y}\}$ and returns a combined forecast ${}^{comb}\widehat{y}$ (see Figure 3.1).

Definition 3.1 (Combination Function): Let a level i of forecasting be given as well as a set of predictions $\{{}^m\widehat{y}\}$ for a future time index t. A combination function F is a function $F : \mathcal{R}^M \rightarrow \mathcal{R}$ that calculates a combined forecast ${}^{comb}\widehat{y} = F(\{{}^m\widehat{y}\})$ based on the given input forecasts and potentially additional information.

In the following we will always indicate additional information about forecasts or their generation as a left upper index (like the index "m" or "comb").

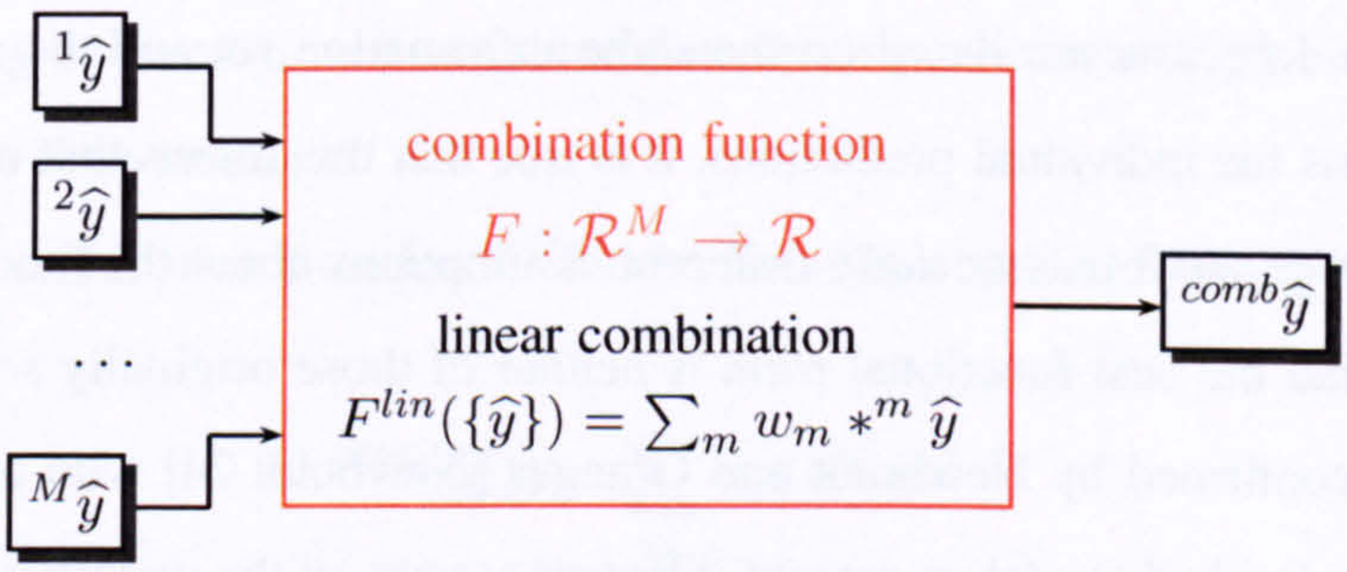

Fig. 19: Forecast combination as a black box

The task of the different combination approaches is to describe the functional relationship which represents the fusion.

Why is this simple idea working so well?

In the beginning of the discussion related to combination approaches different authors argued that if forecast combination works, this simply shows that the individual models representing the input for the combination process are not correct. If it is possible to generate a combined forecast which in the end represents nothing more than a relationship between different inputs (those of the individual forecasts) and one output (which is the combined forecast) and this output is better than each

individual forecast, it would have been possible to model that relationship directly in one forecast model. This in turn proves that the relationship modelled in the individual forecasts is not optimal and there is no need for forecast combination. So why does forecast combination produce good results?

Bates and Granger stated in 1969 [Bates 69] that combination works well because different forecasts consider different independent information of two kinds: one forecast might be based on variables or information that another forecast has not considered or the forecasts make different assumptions about the functional form of the relationship between the variables. It can also be that there is a non-stationarity in the parameters of the model which can be resolved by including forecasts based on different parameter sets into a combination process.

Granger and Ramanathan [Granger 84] discussed the second point and argued that if two forecasts are based on the same information set and the combination outperforms the individual predictions, it is true that this means that neither is optimal. If ,e.g., the forecasts make different assumptions about the functional form, it shows that the best functional form is neither of those originally selected. This has been confirmed by Newboldt and Granger [Newboldt 74] who observed that different individual models represent different aspects of the underlying stochastic process and one can never be certain that a particular model is the most appropriate.

Winkler and Makridakis [Winkler 83] summarised in 1983:

> *The traditional approach of forecasting involves choosing the forecasting method judged most appropriate of the available methods and applying it to some specific situations. The rationale behind such an approach is the notion that a "best" method exists and can be identified. An alternative to the traditional approach is to aggregate information from different forecasting methods by aggregating forecasts. This eliminated the problem of having to select a single method and rely exclusively on its forecasts.*

More concrete and scientifically reasoned arguments for the usefulness of fore-

cast combination will be provided during the analysis in the following chapters.

3.2 Linear Combination Models

The simplest, but also the most common are the linear combination models. The reason to use linear combination models lies in the simplicity of these models as well as in their robustness. In linear combination models the combined forecast is defined as a weighted sum of different given individual forecasts. *This means that the models expect a stable relationship between the individual models which does not depend on time or other influences and can therefore be determined based on historical forecast performance.*

Definition 3.2 (Linear Combination Function, Linear Combining Weight): Let a level i of forecasting be given as well as a set of time series predictions $\{{}^m\widehat{y}\}, m \in \mathcal{M} \subset \mathcal{N}$ for a future time index t.

A linear combination function F^{lin} calculates the combined forecast ${}^{comb}\widehat{y} = F^{lin}(\{\widehat{y}\})$ by

$$F^{lin}(\{\widehat{y}\}) = \sum_m w_m *^m \widehat{y}. \tag{3.1}$$

The parameters $w_m \in \mathcal{R}\ \forall m \in \mathcal{M}$ are called linear combination weights.

Different linear combination models differ in the manner of how to estimate the optimal combining weights w_m based on historical forecast performance.

In a lot of combination models the values or the sum of the combining weights are restricted. Some models restrict the sum of the combining weights to

$$\sum_{m=1}^{M} w_m = 1. \tag{3.2}$$

The advantage of this restriction is that if the individual forecasts are unbiased, this restriction asserts that the combined forecast is unbiased, too.

Other models restrict each weight to

$$0 \leq w_m \leq 1 \quad \forall m \in \mathcal{M} \tag{3.3}$$

for stabilisation purposes.

The following subsection provides a short overview of how the theory of linear combination models has developed. We will also provide references to the most important papers related to linear combination models. Subsection 3.2.2 gives an overview of different approaches to determine combining weights. Then the most common linear combination models are subsequently discussed in more detail in the subsections 3.2.3 to 3.2.7. The description of the models finishes with a comparison of the different models in subsection 3.2.8.

3.2.1 Historical Development

During the last forty years a number of studies related to combination methods have been carried out. According to Stigler (1974) the idea goes back, in the context of estimation, at least to Laplace. The seminal work directly related to linear combination models was presented by Bates and Granger in 1969 [Bates 69]. In this paper the authors propose some of the most common linear combination models and prove experimentally that combination models may be used to increase forecast quality.

A very good review of the most important linear combination methods was published by Clemen in 1989 [Clemen 89]. Menezes, Bunn and Taylor [de Menezes 00] review the most important papers from the perspective of the choice of the appropriate model. This review is also useful because it contains not only references to more recent papers, but also describes the most common models in a short and consistent notation. A good overview also concerning newer findings has been published by Timmermann in [Timmermann 06]. Good practical guidelines for the use of forecast combination are provided in [Armstrong 01].

Here are some of the most important papers related to the linear combination of forecasts:

1969	Bates and Granger [Bates 69] published their seminal paper about forecast combination, in which the most common combination models are proposed.
1974	Newboldt and Granger [Newboldt 74] analysed combinations of different time series forecasts for 80 time series using different estimates of the weights. They concluded that methods assuming independence between the individual forecast errors perform better than the optimal model proposed by Bates and Granger. They suggested to use a small number of forecasts.
1982	Makridakis et al. [Makridakis 82] carried out a general forecast competition of 1001 time series (later known as M- competition). They used two combinations of six forecasts, the simple average and the optimal model. A surprising result was that simple average combinations produced better results than error (co)variance based combinations.
1983	Winkler and Makridakis [Winkler 83] used the 1001 time series of the M- competition to compare the different models proposed by Bates and Granger. The results confirmed the results of Newboldt and Granger. But this time weighted average combinations outperformed simple average combinations.
1984	Granger and Ramanathan [Granger 84] proposed the combination of forecasts as an unlimited least squares regression with an intercept. They showed that if predictions are biased, unlimited regression models are superior to the optimal method.

1985	In a theoretical and simulation study Bunn [Bunn 85] evaluated the quality of combination methods dependant on three statistical values: the variance, the correlation coefficient and the length of the time series. The outcome was a theoretical explanation for the different performance of the models under different circumstances as well as proofs based on experiments with artificial and real data.
1987	Russell and Adam [Russell 87] proposed different rank based combination models and ran experiments with a dynamic selection of the forecasts to be used for combination. They found out that rank based models may perform well and that an intelligent choice of forecasts may be beneficial compared to combinations using a bigger set of forecast models.
1989	Flores and White [Flores 89] evaluated subjective against objective combinations of predictions. Their experiment covered 93 students as predictors and two different kinds of time series. They agreed with Newboldt and Granger and proposed not to combine more then four different predictions. Clemen [Clemen 89] has evaluated in his study about 209 articles related to the combination of forecasts and asked the question why the simple average performs so well in a lot of situations and under which conditions other methods perform better.
1990	Schmittlein et al. [Schmittlein 90] discussed potential methods for the switching between different methods of combination. Holden [Holden 90] proposed regression based combinations with an included intercept but weights summing up to 1.

1992	The empirical work of Gunther [Gunter 92] and Aksu and Gunther [Aksu 92] compared the quality of different least squares methods of combination and the simple average.They found out that the simple average and regression using weights restricted to be non-negative performed better than the unrestricted regression models.
1993	Makridakis et al [Makridakis 93] carried out the M2- competition. The objective of this competition was the measurement of the quality of ten forecasts, five of them made by human experts. They found out that approaches using forecast combination performed very well compared to other approaches.
1994	Deutsch et al. [Deutsch 94] introduced combination methods with changing weights which are calculated by switching regression models.
	MacDonald and Marsh [MacDonald 94] reported on experiments in which they used OLS regression as the method of combination for the prediction of exchange rates because of the presence of Bias in the single predictions. The superiority of the regression method has been confirmed in a number of following papers. But papers also exist which oppose this view with empirical proofs for the superiority of the optimal method over the OLS-regression. For details and references related to this discussion see, e.g., [de Menezes 00].

1998	Klapper [Klapper 98b] proposed extensions of rank models. He outperformed the models proposed by Russell and Adam by using second or higher power rank information. He also proposed multivariate versions of the models. Fischer and Harvey [Fischer 99] discussed under which conditions subjective combination may outperform objective combination of forecasts. Their paper also contained a good overview of literature related to judgemental combination of forecasts.
2000	Menezes, Bunn and Taylor [de Menezes 00] summarised guidelines for the choice of the appropriate linear combination model depending on statistical properties of forecast errors. Hansen discussed in his PhD Thesis [Hansen 00] the topic of forecast combination in relation to different bias- variance forecast error decompositions.
2001	Armstrong provided practical guidelines for the use of forecast combination in [Armstrong 01].
2004	Granger and Jeon introduced "thick modelling" in [Granger 04]. Aioflfi and Timmermann [Aiolfi 04] analysed forecast combination in relation to different error variance based approaches of pooling. Yang [Yang 04] studied some methods of combining procedures for forecasting a continuous random variable. Statistical risk bounds under the square error loss are obtained under distributional assumptions on the future given the current outside information and the past observations.
2005	Elliott and Timmermann [Elliott 05] compare several time varying and static forecast combination models.

2006	Timmermann [Timmermann 05][Timmermann 06] summarised newer findings in forecast combination and provided a consistent mathematical description.
2007	Sancetta [Sancetta 07] proposes online forecast combination for dependent heterogeneous data. The algorithm is an extension of Yang [Yang 04]. It holds for more general data series (e.g. the moment generating function does not need to exist) and a wide variety of loss functions are allowed.

3.2.2 Overview of Linear Combination Models

The most common models described below differ concerning the following points:

- the performance of the individual models is taken into account or not
- the correlation of the individual models is taken into account or not
- the manner in which the performance of an individual forecast is evaluated in comparison to other individual forecasts
- the weights are restricted to sum up to 1 or not
- the weights are restricted to a given interval like [0, 1] or not
- there is a constant term included in the combination or not

The simplest model is the simple average model (see subsection 3.2.3), which gives the same weight to all individual forecasts. As they are constant, the weights are highly restricted and the individual forecast performance or correlation is not taken into account.

There are two common groups of models which take the individual performance into account: rank based models, which are described in subsection 3.2.4, and error variance / covariance based models, which we discuss in subsection 3.2.5. They differ in the manner of how forecast performance is represented. While rank based models describe forecast performance based on ranks of past performance without interpreting statistical properties of forecast errors, the variance /

covariance based models use error variance and covariance information to represent forecast performance. Finally, we have the group of regression based models (described in subsection 3.2.6), in which forecast combination is modelled as an ordinary least squares regression problem and which is strongly related to variance / covariance based models.

In the following subsections the most important models are described in detail. To indicate which model has been used to calculate combining weights, an abbreviation of the model is used as an upper index. The abbreviations used here for the different models are given after the name of each model.

3.2.3 The Average Model

The average model [Bates 69] is a very robust model which is often the first choice in practical applications because of its simplicity. In this model each prediction gets the same weight. It is

$$w_m^{av} := \frac{1}{M},\ m \in \mathcal{M}. \tag{3.4}$$

The model performs very well in a lot of practical applications. For a discussion why this is the case see Section 4.4.2.

3.2.4 Rank- Based Models

Rank- based models [Bunn 75][Russell 87] determine the weights depending on the ranks of past performance of the individual forecasts. The general idea is to give higher weights to models which have performed well and lower weights to poor models. As a basis for the decision, which forecast is expected to be good and which to be bad, the term of the rank rk of forecasts is defined.

Definition 3.3 (Rank of Forecasts): Let a time series y be given for a historical time period. Let $\{{}^m\widehat{y}\}$, $m \in \mathcal{M}$ be a set of M forecast series predicting y. Let ${}^m e$

be the measured squared error ${}^{m}e = (y - {}^{m}\widehat{y})^2$ of the forecasts for a given time index t. Then the rank function is the function $rk : \mathcal{R} \rightarrow [1 \ldots M] \subset \mathcal{N}$ which gives an indicator value of '1' to the best model, '2' to the second best, '3' to the third best and so on. This means that the rank function fulfils

$$rk({}^{m_1}\widehat{y}) < rk({}^{m_2}\widehat{y}) \Leftrightarrow {}^{m_1}e <^{m_2} e \; \forall m_1, m_2 \in \mathcal{M}. \tag{3.5}$$

The Outperformance Model (outp)

In the outperformance model proposed by Bunn in 1975 [Bunn 75] each individual weight is interpreted as a probability that the corresponding individual prediction will perform the best in the future. The probability is estimated as percentage of times, where the individual prediction has performed best in the past.

$$w_m^{outp} := \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \begin{cases} 1 & : \quad rk({}^{m}e_t) = 1 \\ 0 & : \quad otherwise \end{cases} \tag{3.6}$$

The outperformance model is a simple, robust, intuitive model which gives good results even for short historical data. It is also possible to easily incorporate expert knowledge into the weights.

Generalised Rank Based Models (rk, rk$< j >$)

Generalised rank based models use not only the information about which model has performed best, but also the information about the other ranks. They were proposed by Russell and Adam [Russell 87] in 1987 (in their paper referred to as model CCIV3). The weights are defined as

$$w_m^{rk} := \frac{\sum_{t \in \mathcal{T}} (M + 1 - rk({}^{m}e_t))}{\sum_m \sum_{t \in \mathcal{T}} (M + 1 - rk({}^{m}e_t))} \tag{3.7}$$

which simply means that for each historical time interval an influence of M is given to the best model with decreasing values for the lower ranked models and ending with 1 given to the worst model. The influence values are then added up over all time intervals and scaled so that they sum up to 1.

Versions using second or even higher order rank information also exist.

For these models, it is

$$w_m^{rk<j>} := \frac{\sum_{t \in \mathcal{T}} (M + 1 - rk({}^m e_t))^j}{\sum_m \sum_{t \in \mathcal{T}} (M + 1 - rk({}^m e_t))^j} \tag{3.8}$$

with $j \in \mathcal{N}$. The term '$< j >$' in the title of the model stands for the value of j. In experiments carried out by Klapper [Klapper 98b] the versions $rk2$ and $rk4$ outperformed the basic model of Russell and Adam.

3.2.5 Variance/Covariance- Based Models

Variance/ covariance-based models calculate the weights based on a given variance or covariance structure of the forecast errors of the individual predictions. The general idea is that forecasts with a low error variance should get a higher combining weight. The simplest and robust approach calculates the weights directly on the basis of the error variances. A well known extension is the optimal model which also takes into account that the individual forecasts may be correlated. For an extreme example, suppose that we have three methods and that the correlations among their forecast errors are zero for method 1 and 2 , zero for methods 1 and 3, and one for methods 2 and 3. In this case, the forecasts provided by methods 2 and 3 are redundant and should not each be given the same weight as that given to the first method.The weights assigned to the different forecasting methods, then, should be related to the covariance matrix of forecast errors.

A large variety of extensions handling for instance bias and skewness effects also exist [de Menezes 00][Genest 86].

The Variance Model (var)

In the variance model the weights are based on the inverses of variances of the individual forecast errors. The model has been proposed by Bates and Granger [Bates 69] for two individual forecasts and generalised and studied in more detail by Granger and Ramanathan [Granger 84].

The weights are given as

$$w_m^{var} := \frac{\frac{1}{{}^m\delta^2}}{\sum_m \frac{1}{{}^m\delta^2}}. \tag{3.9}$$

where ${}^m\delta^2$ represents the error variance of forecast method m,

$${}^m\delta^2 := \frac{1}{|T|}\sum_{t\in T}({}^m\widehat{y} - y)^2. \tag{3.10}$$

The weights are based on the inverse of the error variance which means that forecast models performing well get a higher weight. The values are forced to sum up to 1 through dividing them by the sum of the inverses of error variances of all methods included in the combination.

The Optimal Model (opt)

The optimal model has been proposed in the seminal paper of Bates and Granger [Bates 69]. In that model the weights are calculated so that the variance of the error of the combined forecast is minimised. This is done under the condition that each individual prediction has no bias. The model takes into account that correlations among the errors of the forecasts may exist. That is why not only error variance information of the individual forecasts but also their covariances are included in the model.

The combining weights are calculated as:

$$w_m^{opt} := \left[\frac{\Sigma^{-1}\eta}{\eta^T\Sigma^{-1}\eta}\right]_m, \tag{3.11}$$

where $\eta = [1]^M$ represents the $[M * 1]$ unit vector and $\Sigma \in \mathcal{R}^{M\times M}$ is the covariance matrix of the forecast errors containing covariances

$$^{m_1,m_2}\rho := \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}}[(^{m_1}\hat{y} - y)(^{m_2}\hat{y} - y)],\ \forall m_1, m_2 \in \mathcal{M}. \tag{3.12}$$

Granger and Ramanathan (1984) [Granger 84] showed in 1984 that the method is equivalent to a least squares regression, in which the constant is suppressed and the sum of the weights is restricted to 1. The difficulty of the approach is that ρ has to be known to calculate the weights.

In practice the matrix ρ is often not stationary, so it has to be estimated on a regular basis using a restricted historical time period $\mathcal{T}$.

The motivation for the approach is given by Bates and Granger using an example of two forecast models $^{m_1}\hat{y}$ and $^{m_2}\hat{y}$. If we assume that the performance of the two models is consistent over time with error variance

$$^{m_1}\delta^2 := \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}}(^{m_1}\hat{y} - y)^2, \tag{3.13}$$

with $^{m_2}\delta^2$ analogous for all time periods $\mathcal{T}$, and the covariance $^{m_1,m_2}\rho$ as defined in (3.12), the error variance $^{comb}\delta^2 \in \mathcal{R}$ of the unbiased combined forecast corresponding to (3.1) with restriction (3.2)

$$^{comb}\delta^2 := \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}}(^{comb}\hat{y} - y)^2 \tag{3.14}$$

can be calculated as

$$^{comb}\delta^2 = w_{m_1}^2 *^{m_1} \delta^2 + w_{m_2}^2 *^{m_2} \delta^2 + 2 *^{m_1,m_2} \rho * w_{m_1} w_{m_2}. \tag{3.15}$$

The weight w_{m_2} can be substituted from (3.2), which leads to

$$^{comb}\delta^2 = w_{m_1}^2 *^{m_1} \delta^2 + (1 - w_{m_1})^2 *^{m_2} \delta^2 + 2 *^{m_1,m_2} \rho * w_{m_1}(1 - w_{m_1}). \quad (3.16)$$

We get the minimum of $^{comb}\delta^2$ by differentiating with respect to w_{m_1} and equating to zero. The minimum of $^{comb}\delta^2$ occurs for

$$w_{m_1} = \frac{^{m_2}\delta^2 - ^{m_1,m_2}\rho}{^{m_1}\delta^2 + ^{m_2}\delta^2 - 2 *^{m_1,m_2} \rho}. \quad (3.17)$$

This corresponds to equation (3.11) for the case of two forecast models. In this case the covariance matrix ρ corresponds to

$$\Sigma = \begin{pmatrix} ^{m_1}\delta^2 & ^{m_1,m_2}\rho \\ ^{m_1,m_2}\rho & ^{m_2}\delta^2 \end{pmatrix}. \quad (3.18)$$

and the inverse is

$$\Sigma^{-1} = \frac{1}{^{m_1,m_2}\rho^2 - ^{m_1}\delta^2 *^{m_2}\delta^2} * \begin{pmatrix} -^{m_2}\delta^2 & ^{m_1,m_2}\rho \\ ^{m_1,m_2}\rho & -^{m_1}\delta^2 \end{pmatrix}. \quad (3.19)$$

The application of this inverse matrix in (3.11) leads to weights as described in (3.17).

The Optimal Model with restricted Weights (optrw)

In the optimal model with restricted weights equation (3.11) has the additional restriction that no individual weight is allowed to be outside the interval [0,1]. The model was also proposed by Granger and Ramanathan [Granger 84] in 1984. It showed results that are much more stable in relation to small data changes. An explanation for this behaviour will be given later in Section 4.4. The inconvenience is that the calculation of the weights is not as straightforward as for the optimal model.

3.2.6 OLS- Regression Models

The Regression Model (ols)

In combinations with regression models the individual predictions are regressors in an ordinary least squares regression (OLS) with use of a constant.

Equation (3.1) is extended to

$$F^{ols}(\{\widehat{y}\}) := \sum_m w_m^{ols} *^m \widehat{y} + w_{M+1}^{ols}, \tag{3.20}$$

with $w_m^{ols} \in \mathcal{R}\ \forall m \in \mathcal{M} = [1, \ldots, M]$, $w_{M+1} \in \mathcal{R}$ a parameter that represents a constant term.

Granger and Ramanathan [Granger 84] argue that this method is superior to the optimal methods, because an unbiased prediction is produced, even if the single predictions contain a systematic error. They proved that the optimal model is nothing more than an OLS regression without use of a constant. From a theoretical point of view there is no reason to expect that the individual forecasts must be unbiased. The authors propose the use of an OLS regression containing a constant term, because it represents an extension of the optimal model which combines biased forecasts in an optimal manner.

The Regression Model with restricted Weights (olsrw)

The outcome of the experiments of Granger and Ramanathan [Granger 84] has also been a regression model with restricted weights corresponding to the regression model, but containing the restriction described in (3.2), which means that the combining weights must sum up to 1.

3.2.7 Other Models

Since the beginnings in 1969 other linear combination models have been proposed [de Menezes 00][Littlestone 92][Flores 89], but have rarely been applied in practi-

cal applications. The main reasons for this are a higher computational complexity as well as instabilities appearing for the more complex sophisticated models (we will come back to this point in Section 4.4). It is also much more difficult to interpret the results of these more sophisticated models.

Models taking into account the distributional properties

Some of the newer approaches are not only based on the pure accuracy perspective, but are also taking into account distributional properties like error variance, distribution asymmetry and serial correlation. For examples see [de Menezes 00].

Models based on Bayesian Probabilities or quasi- Bayes Probabilities

Bunn proposed different approaches to calculate linear combination weights based on Bayesian probabilities or quasi- Bayes probabilities for the first time in 1975 [Bunn 75], other publications followed in 1985 [Bunn 85] and 1989 [Bunn 89]. Some of the models can be interpreted as generalisations of the outperformance model. But as these models were usually worse in comparison to the common linear combination models, they are not further considered here.

Multivariate Approaches

Different authors tried to extend combining approaches to multivariate combining techniques (see ,e.g., Klapper [Klapper 98b] for rank based models). The idea of multivariate approaches is that information about the quality of future forecasts may be hidden in the past performance of other variables, so that rank information or correlation aspects of other forecasts are taken into account.

Classification Models

A wide range of combination models exists which are related to classification problems. The most common is weighted majority voting [Littlestone 92]. An overview

of methods for the combination of classifiers is provided by Ruta and Gabrys in [Ruta 00].

Judgemental Forecasting

A lot of authors discussed questions related to the combination of judgemental forecasts (forecasts produced by human experts) with system based forecasts or judgemental combinations (combination of system forecasts carried out by human experts). For a comparison between judgemental or subjective combinations and objective combinations see, e.g., Flores and White's paper of 1989 [Flores 89] in which competitive experiments are described. One of the most important questions here is under which conditions experts are able to beat pure system based forecasts. What kind of feedback do experts need in order to improve their forecasting or combination abilities? Approaches of automatic corrections of judgemental forecasts also exist. A good overview to the literature related to this topic until 1999 can be found in [Fischer 99].

3.2.8 Relations between the Linear Combination Models

All of the linear combination models seem to be quite different at the first view. Nevertheless, they have a lot of common characteristics. Moreover, they can be interpreted as two groups of models each representing a hierarchical structure in the sense of one method being a generalisation of another method. By generalisation we mean here that one or more restrictions are relaxed or completely removed.

In 1984, Granger and Ramanathan [Granger 84] showed that the optimal method is equivalent to a least squares regression, in which the constant is suppressed and the sum of the weights is restricted to 1. This knowledge allows us to compare the variance / covariance models to the regression based models and to interpret all of them in an hierarchical structure, beginning with the simple average model containing all possible restrictions, to the ols regression as the most flexible one containing no restrictions any more.

Figure 20 shows the hierarchical structure of the group of variance / covariance-based and regression-based models. Each node contains one model, the arrows between the models represent a generalisation direction.

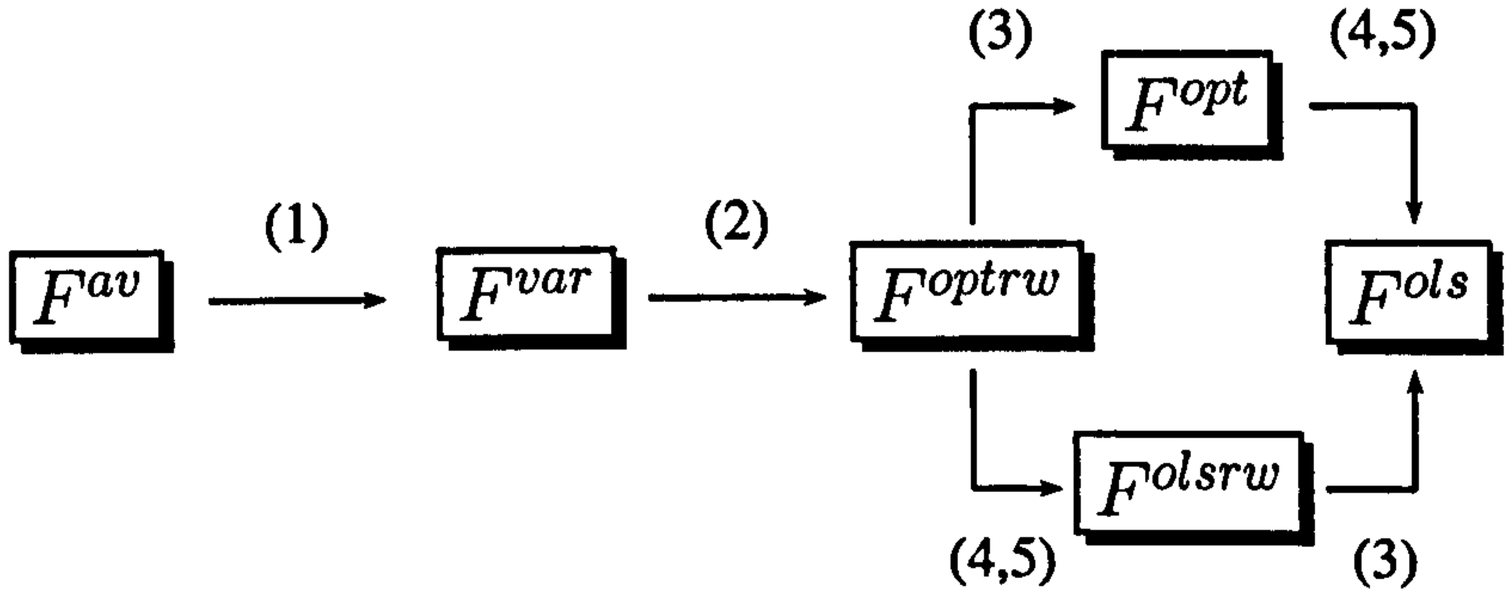

Fig. 20: The group of variance / covariance-based and regression-based models shown as hierarchical structure. The nodes represent the combining models. The arrows represent generalisations achieved by relaxing one or more restrictions.
1: the error variance is expected to be equal for each individual forecast model, 2: the covariance is expected to be zero between each pair of individual forecasts, 3: the combining weights are restricted to the interval $[0, 1]$, 4: the weights are restricted to sum up to 1, 5: the constant term is suppressed

The hierarchy of the other group of models, the rank-based models, is shown in figure 21.

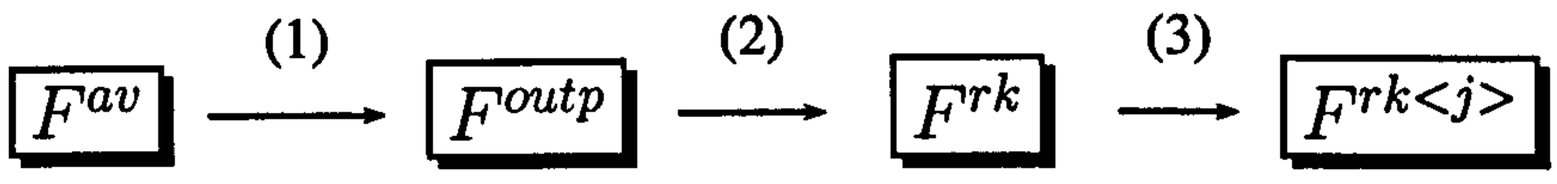

Fig. 21: The group of rank-based models shown as an hierarchical structure. The nodes represent the combination models. The arrows represent generalisations achieved by relaxing one or more restrictions.
1: the performance is expected to be equal for each individual forecast model, 2: only the best rank is taken into account, 3: the parameter j is restricted to $j = 1$

A discussion about advantages and disadvantages of the models as well as questions about the choice of a model in different situations will be provided in the next chapter.

3.3 Nonlinear Combination Models

Nonlinear combination models represent the general class of combination model without any restrictions on the combination function F. As the general definition of a nonlinear combination model is quite flexible, most of the approaches found in the literature are specialised to a specific application, to special individual forecast methods or to special classes of functions used in the combination.

The following subsection 3.3.1 gives a short overview of examples of nonlinear combination models published in the last ten years. Then we present three special cases of nonlinear combination approaches. In subsection 3.3.2 we present an extension of the linear combination models that has resulted in promising results for our application. Instead of having fixed weights we use dynamic weights depending on the predicted numbers. Subsection 3.3.3 discusses the option of a transformation of a linear fusion process into another space. Subsection 3.3.4 then gives an overview of the most often discussed nonlinear combination models which combine forecasts produced by general approximators like neural networks.

3.3.1 Historical Development

A good review of combination of artificial neural networks is given in Sharkey [Sharkey 96]. Genest & Zideck [Genest 86], Jacobs [Jacobs 95] and Xu et al. [Xu 92] summarise the combination models mostly used to combine neural networks.

1965	One of the first papers related to ANN combination is published by Nilsson in 1965 [Nilsson 96].
1990	Schapire [Schapire 90] proposes the boosting algorithm.
	Hansen and Salomon [Hansen 90] run experiments using neural network ensembles.

1992	Wolpert proposes a neural network to combine the outcomes of diversified neural networks. He introduces the term of "stacked generalisation" which is later used by other authors too.
	Geman et al. [Geman 92] discusses the topic of ANN error decomposition into bias and variance terms.
1993	Perrone and Cooper [Perrone 93] discuss the selection of nets for effective combination and suggests not to include nets exhibiting a high degree of correlation.
1994	Rogova [Rogova 94] proposes Dempster- Shafer belief- based methods.
1995	Krogh and Vedesby [Krogh 95] discuss the approach of cross- validation and provide an account of bias and variance terms in an ANN ensemble.
	Maclin and Shavlik [Maclin 95] discuss ways of how to generate diverse neural network ensembles using different network initialisations.

1996	Breiman [Breimann 96] proposes the method of bagging. Hashem [Hashem 96] discusses effects of collinearity for combination of ANNs. Tumer and Gosh [Tumer 96] propose to create diverse neural networks by injecting noise into the data, by using different pruning methods or by using different nonlinear transformations. A. Sharkey [Sharkey 96] writes a review paper on combining artificial neural nets as an introduction to a special issue of the *Connection Science* journal. Raviv and Intrator [Raviv 96] summarise and discuss in the same journal methods to create diverse neural nets in altering the training data. Rosen [Rosen 96] discusses options of how to create decorrelated neural networks.
1998	Liu [Liu 98] extends to theory of [Rosen 96] and proposes *negative correlation learning.*
1999	Opitz and Shavlik [Opitz 99b] present a genetic algorithm to create diverse sets of neural networks. Another proposition of how to combine forecasts using a neural network is given by Shan et al. [Shi 99]. Opitz and Maclin [Opitz 99a] carry out an empirical study to compare different ensemble methods based on bagging and boosting used for neural networks and decision trees.

2001	Zhou et al. [Zhou 01] propose to combine well selected subsets of a set of given neural networks. The subset and the corresponding weights are initialised and chosen using evolutionary strategies. Burgess proposes a population based algorithm to perform joint optimisation of a portfolio of models in [Dunis 01].
2005	He and Xu [He 05] propose a new nonlinear combination method using self- organising algorithms. Brown et al. summarise and extend the theoretical background related to negative correlation learning [Brown 05a]. They investigate the issue of how to explicitly manage the correlations of an ensemble of regression estimators [Brown 05b]. They also provide an experimental comparison with other ensemble learning techniques like bagging, boosting mixture of experts and Gaussian processes.
2007	Ozun and Cifter [Ozun 07] apply neural networks trained with a genetic algorithm in order to combine financial forecasts. Guidolin and Timmermann [Guidolin 07] present a flexible forecast combination approach considering regime switches.

3.3.2 A Dynamic Representation of Linear Combination Weights

In this section we focus on a special type of nonlinearity in combination models. In a first step towards nonlinear combination we extend the linear combination models by modelling adaptive weights depending on the predicted values. Equation (3.1) is extended to

$$F^{dyn}(\{\widehat{y}\}) = \sum_m G_m(\{\widehat{y}\}) *^m \widehat{y}, \tag{3.21}$$

with a given class of functions $G_m : \mathcal{R}^M \rightarrow \mathcal{R}\ \forall m \in \mathcal{M}$.

This approach makes sense for applications in which the expected performance of different models depends on the predicted numbers. The functions G_m can for instance represent a rule-based system which selects the weights depending on the predicted situation. It is also possible to incorporate additional knowledge into the functions G_m.

This approach of forecast combination is used in the current *ProfitLine.O&D* system for the combination of seasonal predictions (see Section 2.2.6). The model $h_3(;)$ presented in equation (2.18) works very well for high seasons, but produces unstable results for low seasons because of the small numbers to be predicted. We could therefore achieve highly improved results in comparison to pure linear combinations for our application taking the predicted values as well as other additional information into account. The functions G_m realises a smooth switch between a set of weights representing the performance of the different methods for low values and a set of weights representing the performance for high values. The switch is modelled as an extended sigmoid function. More details related to the functions G_m cannot be provided here because of commercial aspects.

Another example for a dynamic representation of linear combination weights can be found in the paper of Guidolin and Timmermann [Guidolin 07] who use a multivariate regime switching process to capture the existence of common, discrete factors driving both the stochastic process of the variable of interest and a related market variable.

3.3.3 Linear Combination of Transformed Forecast Values

Another special type of functional approaches represents the approach of linear combination of transformed forecast values. This approach is based on a linear combination, but the combination includes a preprocessing and a postprocessing of the predicted individual and combined forecasts.

Function $F^{lintrans}$ is represented as

$$F^{lintrans}(\{\widehat{y}\}) = G(\sum_{\tilde{m}} w_{\tilde{m}} * G_{\tilde{m}}(\{\widehat{y}\})). \tag{3.22}$$

The functions $G_{\tilde{m}}$ represent a transformation of the predicted values into another space, which can also be characterised by a different dimensionality. The transformed predictions are then linearly combined. The function G finally transforms the result back into the original space.

Merz and Pazzani used this approach very successfully in eliminating two of the most relevant risks of linear combination models by the transformation: a) a too large number of forecasts and b) the correlation between forecast errors. They used principal component analysis in order to generate a smaller number of independent predictions. Details can be found in [Merz 97]. In the case of the principal component regression applied by Merz and Pazzani the function $G_{\tilde{m}}$ as well as function G represent a weighted sum of the inputs, so that their algorithm realises a linear combination of the input predictions.

3.3.4 Using General Approximators

General approximators like mixtures of Gaussians [Ghahramani 94][Nowlan 91] or others can model a nonlinear application-specific behaviour. The functions F can represent any function space known as general approximator. The target is to model

$$y \approx F^{appr}(\{\widehat{y}\}) \tag{3.23}$$

in an optimal manner on the basis of training data measured for a historical time period $t \in \mathcal{T}$. The task of the combination process consists of determining the parameters of the function F. For some classes of functions, specific methods are known for how to determine the parameters based on given data samples. So we can, e.g., use the Expectation- Maximisation algorithm [Dempster 77] as a general method of finding the maximum likelihood estimates of the parameters of the underlying distribution in the case of Gaussian Mixture models. If such a method

is not known, evolutionary strategies can be used to determine optimal parameter settings [Zhou 01].

Neural Network Ensembles

> *There have been proved practical advantages in either decomposing a task into subtasks or combining several different solutions to the same task; the most significant one for the present purpose being that of improved performance.* (A. J. Sharkey, [Sharkey 96])

Neural networks represent well known general approximators.

Combination models which use neural networks are proposed, e.g., by Shi [Shi 99]. Neural network combination models are able to learn real nonlinear dependencies of the target on the predicted values. The combination function F^{neuron} of a typical neural network neuron is given by

$$F^{neuron}(\{\widehat{y}\}) = G(\sum_m w_m *^m \widehat{y}), \tag{3.24}$$

using a given function G (e.g. the sigmoid function) and learning the parameters w_m.

Most of the literature concerning neural networks and forecast combination is related to the question of how to combine predictions which *have been generated* using neural networks. Two general approaches exist to combining artificial neural networks (ANNs). The first approach is an ensemble-based approach, in which different neural nets are trained on what is essentially the same task, and then the outputs are combined [Krogh 95][Sharkey 96][Breimann 96][Schapire 90] [Freund 96][Druckner 94] [Tumer 96][Sharkey 95][Hansen 90] [Maclin 95][Rogova 94][Xu 92].

Many of these papers show that neural network ensembles can be very effective. Neural network combinations used in experiments carried out by Ruta and Gabrys [Ruta 07] in 2006, for instance, have been evaluated within the NISIS2006 competition. They showed the best predictive performance among 12 competitive

models for prediction of different univariate and multivariate time series.

The combination is typically carried out at the decision level, meaning a combination of the forecasting results. But it is also possible to combine at the model level, what has been done by Gabrys [Gabrys 02][Gabrys 03] for combination of neuro- fuzzy classifiers. An advantage of combining at the model level is the fact that such type of combination offers model transparency in terms of a single resulting classification model.

The other approach of combining ANNs is a modular approach. Here a problem is decomposed into different subtasks by application based decomposition or automatic decomposition [Sharkey 96][Jordan 95][Waibel 89]. As we cover the forecast combination here without putting too much emphasis on issues related closely to neural networks, we will focus on the ensemble based approach.

One of the most common approaches had been to generate a population of neural nets using different initialisations of the weights and then to chose the best one. But a number of studies have proven that often neural network ensembles using the results of more than one ANN can outperform the results of the best network. There are two main issues related to neural network ensembles:

- the creation or selection of neural nets to be combined [Breimann 96] [Hansen 90][Freund 96][Druckner 94] and
- the methods of combining them (including those presented in the previous section, but also others specialised for neural networks [Rogova 94] [Genest 86],[Jacobs 95][Xu 92]).

The principle efforts are related to creating neural networks which are diverse in order to provide different information to the combination process. These topics are discussed in the next chapter.

The studies of how to combine neural networks have also been focused on the analysis of the effects of the resulting decomposed forecast errors. Theil [Theil 91] showed that the errors can be decomposed into bias and variance terms, for neural networks the bias meaning the ability to generalise correctly on the given training

set and the variance indicating how much the result is sensitive to the given training set. These topics are discussed in the next chapter as well.

One issue related to general neural network combination is the fact that there is no understandable representation of the learned structures. Therefore, neuro-fuzzy approaches represent a very useful option for combination [Jang 93][Gabrys 03]. These approaches have the advantage that the fuzzy component provides an interpretable representation of the learned weights

For comparison purposes, we have included in our experiments the neuro-fuzzy approach ANFIS proposed by Jang in 1993 (for details see [Jang 93]). The acronym ANFIS derives its name from adaptive neuro-fuzzy inference system. Using a given input/output data set, ANFIS constructs a fuzzy inference system (FIS) whose membership function parameters are tuned (adjusted) using either a back-propagation algorithm alone or in combination with a least squares type of method. A network-type structure similar to that of a neural network, which maps inputs through input membership functions and associated parameters, and then through output membership functions and associated parameters to outputs, can be used to interpret the input/output mapping. The parameters associated with the membership functions change through the learning process. The computation of these parameters (or their adjustment) is facilitated by a gradient vector. This gradient vector provides a measure of how well the fuzzy inference system is modelling the input/output data for a given set of parameters. When the gradient vector is obtained, any of several optimisation routines can be applied in order to adjust the parameters to reduce some error measure. This error measure is usually defined by the sum of the squared difference between actual and desired outputs. ANFIS uses either back propagation or a combination of least squares estimation and backpropagation for membership function parameter estimation.

3.4 Experiments

3.4.1 Description of Experiments

The experiments have been carried out with the objective to apply linear and non-linear combination techniques to the different Revenue Management demand forecasts described in Table 3 of Section 2.3.3.

The experiments have been organized by following these steps:

- definition of the forecast pool (see Table 3)
- definition of the history pool (see Section 2.3.1, the years 2001 to 2003 have been used as history pool)
- calculation of the individual forecasts (see Figure 17, an analysis of some characteristics of the forecast errors is provided in Figure 18)
- calculation of combinations using the combination models F^{av}, F^{outp}, F^{var}, F^{opt}, F^{ols}, F^{dyn}, F^{appr} (as described in Sections 3.2 and 3.3)
- analysis of the results (will be provided in Section 3.4.2)
- analysis of the achieved combining weights (will be provided in Section 3.4.3)

The experiments can be reproduced with the software as described in experiments 3 (see Appendix B.6.3). The software also allows different modifications of the experiments in order to carry out the statistical analysis of dependencies of the achieved combination weights presented in Section 3.4.3.

As the functions $G_m(.)$ used for approach F^{dyn} (see Section 3.3.2) have been chosen in a very similar way to those applied in the current system, details related to these functions cannot be provided in this thesis because of commercial sensitivity. We can only mention here that different sigmoid functions are used in order to model the strength and weaknesses of different models in different seasons.

The experiments concerning the approach F^{appr} have been carried out within an integrated C++/ Matlab framework. The Matlab version of ANFIS [Jang 93]

has been used in order to train neural nets and to carry out the combination. The results provided on the CD represent the best results achieved after experimenting with structures of varying complexity.

3.4.2 Experimental Results

Table 7 and 8 show the errors of the combined forecasts as relative improvement in relation to the best individual forecast ${}^0\widehat{y}$ (see 2.3.3) at the low and the high level. A graphical representation of the combined errors calculated at the high level (ODO) is shown in Figure 22.

τ	F^{av}	F^{outp}	F^{var}	F^{opt}	F^{ols}	F^{dyn}	F^{appr}
0	-0.03	0.01	-0.02	-0.32	-18.19	-0.21	-1.56
1	-0.03	-0.02	-0.02	-0.24	-5.46	-0.13	-0.61
2	-0.03	-0.02	-0.02	-0.22	-6.36	-0.11	-0.40
3	-0.03	-0.02	-0.02	-0.22	-6.47	-0.13	-0.26
4	-0.04	-0.03	-0.03	-0.23	-3.65	-0.14	-0.18
5	-0.05	-0.03	-0.04	-0.21	-3.24	-0.12	-0.13
6	-0.06	-0.04	-0.05	-0.22	-2.22	-0.15	-0.13
7	-0.06	-0.04	-0.05	-0.22	-2.32	-0.25	-0.12
8	-0.07	-0.05	-0.06	-0.27	-1.96	-0.36	-0.26
9	-0.08	-0.05	-0.07	-0.25	-1.82	-0.33	-0.34
10	-0.09	-0.05	-0.07	-0.25	-2.01	-0.44	-0.49
11	-0.10	-0.05	-0.08	-0.23	-1.79	-0.43	-0.76
12	-0.11	-0.06	-0.09	-0.22	-1.48	-0.41	-0.84
13	-0.13	-0.06	-0.09	-0.22	-1.47	-0.42	-0.75
14	-0.16	-0.06	-0.11	-0.21	-1.38	-0.83	-0.87
15	-0.18	-0.07	-0.11	-0.20	-1.41	-0.87	-0.91
16	-0.22	-0.07	-0.12	-0.20	-1.57	-1.21	-0.97
17	-0.25	-0.08	-0.12	-0.19	-1.61	-1.10	-0.89
18	-0.29	-0.09	-0.13	-0.21	-1.67	-0.80	-0.72
19	-0.35	-0.10	-0.13	-0.23	-1.73	-0.56	-0.52
20	-0.45	-0.12	-0.14	-0.32	-1.90	-0.95	-0.50
21	-0.78	-0.19	-0.14	-1.11	-2.32	-4.83	-4.03
22	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Tab. 7: Relative improvement using forecast combination in comparison to the best individual forecast ${}^0\widehat{y}$ (out of sample results) calculated at level ODO F POS.

Depending on the combination model, the combined forecasts are more or less worse than the best individual forecast. It was not possible to achieve any improvement by the application of linear combination models to the indicated set of individual forecasts at the low level. Only small improvements could be observed

τ	F^{av}	F^{outp}	F^{var}	F^{opt}	F^{ols}	F^{dyn}	F^{appr}
0	-0.11	-0.08	-0.10	-0.51	-22.03	-0.14	-1.23
1	0.02	0.02	0.05	-0.07	-7.47	-0.13	-0.81
2	0.04	0.05	0.07	-0.05	-11.43	-0.13	-0.49
3	0.04	0.05	0.07	-0.07	-12.76	-0.13	-0.49
4	0.03	0.03	0.05	-0.15	-5.10	-0.14	-0.48
5	0.01	0.02	0.03	-0.13	-4.88	-0.15	-0.27
6	-0.02	-0.01	0.01	-0.15	-2.53	-0.15	-0.17
7	-0.03	-0.01	0.00	-0.15	-2.94	-0.25	-0.12
8	-0.03	-0.01	-0.01	-0.30	-2.18	-0.34	-0.20
9	-0.03	-0.01	-0.01	-0.26	-2.20	-0.27	-0.29
10	-0.03	0.00	-0.01	-0.28	-2.90	-0.59	-0.34
11	-0.04	0.00	-0.02	-0.21	-2.33	-0.58	-0.83
12	-0.05	0.00	-0.03	-0.18	-1.37	-0.54	-0.82
13	-0.06	-0.01	-0.04	-0.15	-1.41	-0.50	-0.71
14	-0.10	-0.02	-0.06	-0.15	-1.09	-0.69	-0.69
15	-0.14	-0.03	-0.08	-0.14	-1.21	-0.86	-1.07
16	-0.18	-0.04	-0.09	-0.10	-1.34	-1.03	-0.93
17	-0.21	-0.05	-0.09	-0.08	-1.52	-1.64	-0.81
18	-0.25	-0.06	-0.09	-0.09	-1.56	-1.83	-0.79
19	-0.31	-0.07	-0.10	-0.10	-1.56	-0.55	-0.67
20	-0.37	-0.07	-0.07	-0.13	-1.62	-0.81	-0.54
21	-0.56	-0.08	0.00	-1.14	-1.91	-3.92	-2.83
22	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Tab. 8: Relative improvement using forecast combination in comparison to the best individual forecast ${}^{0}\widehat{y}$ (out of sample results) calculated at the high level (ODO).

at the high level.

3.4.3 Analysis of Forecast Errors and Linear Combination Weights

Experiment 3 (see Appendix B.6.3) also provides the necessary outputs for an analysis of determined combination weights. In addition to the output of the calculated weights, basic statistical properties like average value, standard deviation as well as minimum and maximum are determined corresponding to each representation of each calculation dimension (like each fareclass, each point of sale, each dcp and so on).

After having calculated and evaluated the combined forecasts, an extensive analysis of the combining weights and the forecast errors has been carried out in order to explain the results. Table 9 shows the average value and variance of the weight given to the best forecast model per linear combination model. It confirms

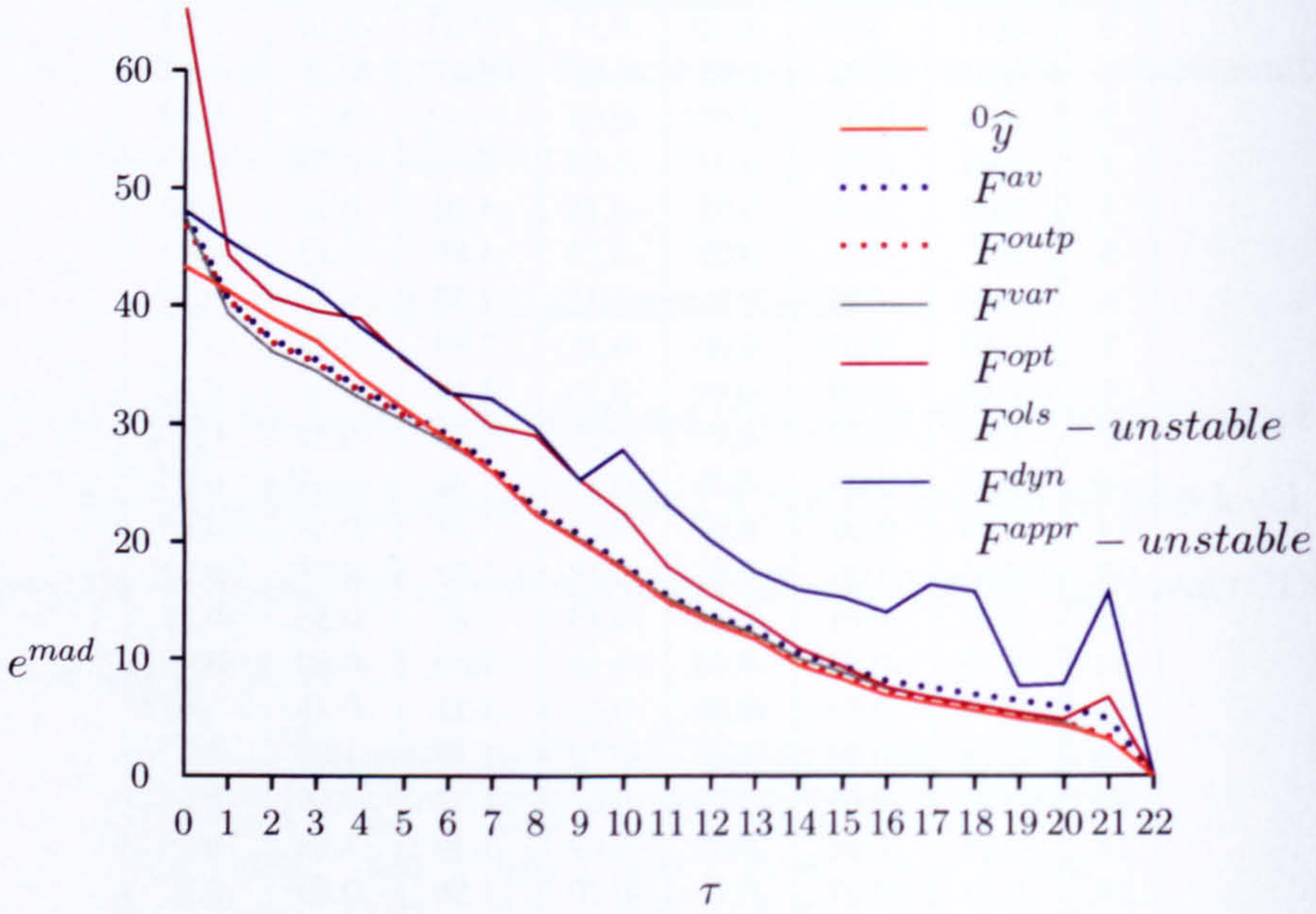

Fig. 22: Errors (mean absolute deviation) achieved using forecast combination in comparison to the best individual forecast $^0\widehat{y}$ at the high level ODO.

the expected behaviour based on the type of the models. The methods F^{av}, F^{outp} and F^{var} produce stable weights, which lead to good combined forecasts. The methods F^{opt} and F^{ols} have been completely unstable. This corresponds to the experiences found in the literature [Bunn 85] (we will discuss that in Section 4.4.1.). The more complex nonlinear models F^{dyn} and F^{appr} produce unstable results as well if applied on the noisy and highly correlated forecasts.

	F^{av}	F^{outp}	F^{var}	F^{opt}	F^{ols}
average	0.14	0.09	0.16	0.52	2.54
standard deviation	0	0.07	0.06	2.66	44.01

Tab. 9: Average and variance of the weight given to the best individual forecast method by different combination models for the example of OD0 0.

Additionally, an analysis of linear combination weights and combined forecast errors has been carried out in order to determine dependencies on different influencing features. This includes:

- an analysis of the correlation between the weights

- in relation to different combination models

- in relation to different fareclasses, day of weeks or point of sales

- the dependency of the weights and combined forecast errors on the average booking value
- the dependency of the weights and combined forecast errors on fareclass, day of week and point of sale
- the dependency of the weights and errors on the individual forecast error variances and covariances
- the dependency on variations using different history pools containing

  - different lengths of the learning period

  - different positions of the learning period

  - different approaches of how to move the learning period and to repeat learning

- the dependency on variations concerning the included input forecasts (combination based on different subsets)

The most relevant detected effects have been:

- Small booking values lead to more unstable (changing) weights.
- High variance in the booking data leads to more unstable (changing) weights.
- The weights achieved with the methods F^{outp} and F^{var} are highly correlated.
- The weights achieved with the methods F^{opt} and F^{ols} are highly correlated as well.
- The dependencies on the characteristics of individual forecast errors correspond to the dependencies described in the literature [de Menezes 00] (dependencies on error covariances will be discussed in the next chapter in Section 4.1.2).

- The quality of the combined forecast depends on systematic forecast errors. Significant systematic errors lead to especially bad combined forecasts and especially unstable combination weights.
- A smaller number of individual forecasts may provide slightly better combination results for models F^{opt} and F^{ols}, as long as there is at least one high quality individual forecast included.

3.4.4 Conclusions and Why it Did Not Work

The behaviour observed in our experiments corresponds to that observed in other experiments described in the literature [de Menezes 00][Bunn 85]. Nevertheless, it was not possible to clearly outperform the best individual forecast at both levels. One of the reasons is that there are two highly correlated individual models which outperform all the others. These models have already been frequently tuned and optimised. This leads to forecasts which are in total not better than the currently applied model.

Another reason for the small improvement can be found in the high covariance values between the individual forecasts. High noise terms in the data and forecast models which belong to the same group of models produce not very diverse results. We will discuss this in more detail in Section 4.1.2. The process of learning the combination weights over-interpretes small differences between the forecasts. It compensates small variations in big forecast errors by extreme combination weights. This leads to instabilities and, because of the over- interpretation of single historical data values, to weights which are not representative for the future. These effects could be observed especially if models have been used which use covariance information. This phenomenon will be discussed in more detail in Section 4.4.1.

During the analysis of the errors, after applying different sets of individual forecasts, it was observed that the combination works well for forecasts which differ either in the forecast of the attractiveness or in the forecast of short term

influences. Forecasts which differ in more than one of the parts do not form a suitable set for combination in a parallel fashion. The linear combination process has difficulties if it is to compensate more than one aspect of "diversity" between the individual forecasts. This seems logical since only one weight is associated with each individual forecast, and cannot simultaneously support the strength in one component of the forecast (e.g. the better forecast of the attractiveness) and dump the negative effects of another component (e.g. based on a bad seasonal forecast). We will therefore have a closer look at the relation between forecast combination and decomposition in Section 4.2.1.

A completely different aspect is that the quality of different seasonal forecasts depends highly on the size of the predicted numbers. Some methods are very good to predict high seasons and others are very good to predict low seasons. One reason for this effect is that high seasons bookings arrive earlier. As a larger amount of bookings can be observed in a high season, it is much easier for models interpreting current booking values to produce reliable predictions in that case. That is why nonlinear models using a dynamic representation of linear combination weights as described in Section 3.3.2 perform well for our application. As this type of combination exists already in the current system (see Section 2.2.6), it was not possible to significantly outperform this approach with alternative combination models as described in this chapter.

Summarising, the observations made by performing the experiments described in this chapter have led to the decision to study how "diversity" of individual forecasts can be defined, what kind of diversity is needed to obtain good combination results and how we can generate such diverse forecasts. All these questions are discussed in Chapter 4.

4. INFLUENCES ON COMBINATION EFFICIENCY

4.1 Diversity of Input Forecasts

One of the crucial issues relating to forecast combination is the task of choosing appropriate input forecasts that are to be combined. We will now concentrate on abilities of input forecasts to provide additional information to a combination process. If we use, e.g., a single prediction and duplicate this prediction ten times in order to generate a set of input forecasts, the combination of this set of identical forecasts will not lead to any improvements in comparison to the single forecast accuracy. This example shows that including forecasts into a combination process is only useful if there is some kind of additional information provided and the input predictions are diverse in a certain manner.

In this section we will therefore discuss the question of how we can determine the diversity of predictions. We will start with a brief overview of how diversity is defined in other domains in Section 4.1.1 and then discuss different characteristics of divers forecasts in Sections 4.1.2 and 4.1.3. The active generation of diverse forecasts using diversifying measures is later discussed in Section 4.2.

4.1.1 Diversity Measurements in other Domains

Diversity in Life Sciences

Biologists and ecologists defined their idea of diversity several decades ago. In biology diversity is used to measure how many populations of animals differ concerning a special behaviour. Rao [Rao 82] gives the following definition of diversity:

Let $(\mathcal{X}, \mathcal{B})$ be a measurable space, and let $\mathcal{P}$ be a convex set of probability measures defined on it. A Function H(.) mapping $\mathcal{P}$ onto the real line is said to be a measure of diversity if it satisfies the following conditions:

- *C1: $\mathcal{H}(P) \geq 0$, for any $P \in \mathcal{P}$ and $\mathcal{H}(P) = 0$ if $\mathcal{P}$ is degenerate.*
- *C2: $\mathcal{H}$ is a convex function of $\mathcal{P}$.*

Even if the task in biology seems to be quite different from that of evaluating the diversity of forecasts in order to get high quality combinations, there is a relation to our problem. It is interesting to see that a diversity measure for classifiers (the measure of disagreement, see [Kuncheva 01]) represents a special version of the measure proposed by Rao. For details related to the comparison between the measures see [Kuncheva 03].

Diversity of Classifiers

The most common measures of diversity for classifiers have been summarised and compared by L.I. Kuncheva and C.J. Whitaker [Kuncheva 01]. The authors have defined and compared a whole set of diversity measures related to the problem of classification. In [Kuncheva 01] Kuncheva and Whitaker summarise ten measures of diversity proposed in the literature, four pairwise and six non-pairwise measures. The analysis has been extended by Ruta and Gabrys [Ruta 00][Ruta 02].

4.1.2 Correlation as Diversity Indicator

Forecast Correlation is an indicator that can be used in order to describe the diversity of forecasts. If forecasts are highly correlated, it means that there is a lot of information that they have in common. If we duplicate a given forecast and include it into a forecast combination process many times, we cannot expect a large improvement over such individual forecast. In this case the forecasts are highly positively correlated. If on the other hand forecasts are independent or even

negatively correlated this means that errors can compensate each other during the combination.

Generalising equation (3.15) to a larger number of predictions without systematic error leads to a general error representation of

$$^{comb}\delta^2 = \sum_{m_1,m_2} w_{m_1} w_{m_2} (^{m_1,m_2}\rho) \tag{4.1}$$

with $m_1, m_2 \in \mathcal{M}$ indicating all pairs of input forecasts. The diagonal line of the covariance matrix contains the error variances of the input forecasts. In case of completely independent input forecasts, this means that the resulting error variance is only determined by the error variance of the input forecasts

$$^{comb}\delta^2 = \sum_{m} w_m^2 *^m \delta^2. \tag{4.2}$$

If we look at (4.1) including the resulting elements representing each covariance between a pair of forecasts, we get

$$^{comb}\delta^2 = \sum_{m} w_m^2 *^m \delta^2 + \sum_{m_1 \neq m_2} w_{m_1} w_{m_2} (^{m_1,m_2}\rho). \tag{4.3}$$

This representation shows clearly that the total error strongly depends on the error covariances represented in the second summand. If $^{m_1,m_2}\rho$ contains positive values indicating a positive correlation between the input predictions, the resulting error is increased. If $^{m_1,m_2}\rho$ contains negative values, we can achieve even a better result than we would achieve with independent forecasts. Additional information about the impact of error variances and covariances can be found in [Bunn 85].

4.1.3 Diversity in Relation to Error Decomposition

The Impact of Error Components on a Forecast Combination

We have just seen that the correlation between forecasts is essential in order to describe the potential of forecast combination in relation to a given set of input

forecasts. This can help in order to qualify such a set, but what to do if we find that a given set of forecasts does only contain highly correlated forecasts?

The most promising approach in this case is to evaluate options to chose other input forecasts. This can be achieved by using other methods of forecasting, other parameter values or other training data. But what to change and would such a set perform better?

In (2.1) we have modelled our data affected by random noise ϵ_y which is interpreted as not predictable. This noise term exists therefore in each prediction irrespective of the model we use. It can be the case that this term is so large in comparison to the predictable part that chosen input forecasts are already perfect and nevertheless highly correlated because of this error component. In this case no modification of the set of input forecasts will help to decrease the forecast error.

This example shows that an analysis of the composition of forecast errors can help to decide if and what to change in order to generate a divers set of input predictions. If we can decompose forecast errors into independent components, we can analyse the correlation in relation to each of the components. This can help to identify promising modifications of the forecast generation process.

The Bias- Variance- Bayes Error Decomposition

Let e represent the error which will be generated in predicting y based on $h(x, \widehat{\phi})$ (out of sample predictions):

$$e = y - \widehat{y} = y - h(x, \widehat{\phi}) = f(x) - h(x, \widehat{\phi}) + \epsilon_y \tag{4.4}$$

Let us assume that we have found an estimator $h(x, \widehat{\phi})$ which generates predictions without a systematic error so that (e) can be represented as Gaussian with $e \sim N(0, \delta_e^2)$.

Then the total error variance term δ_e^2 can be decomposed into different components. While different error decompositions can be found in [Geman 92] and

[Hansen 00], we will refer here to the decomposition of James and Hastie [James 96]:

$$\delta_e^2 = \delta_h^2 + \delta_\phi^2 + \delta_y^2. \tag{4.5}$$

The first error component ϵ_h with variance δ_h^2 is called the bias. This error component is based on the fact that the class of functions $h(x;)$ may not include the function $f(x)$. As we have assumed that an ideal parameter set ϕ exists in order to estimate $f(x)$ based on $h(x;\phi)$, the bias term of the error is defined by $\epsilon_h = f(x) - h(x,\phi)$.

The second term ϵ_ϕ of the error with variance δ_ϕ^2 is the error variance component. This term is based on the fact that the parameters ϕ cannot be estimated perfectly because of noise in the training data, limited number of training samples, etc. The variance term of the error is defined by $\epsilon_\phi = h(x,\phi) - h(x,\widehat{\phi})$.

The third term ϵ_y with variance δ_y^2 represents the irreducible Bayes error component in y which can be reduced only if more information becomes available in x.

While the third part of the error cannot be reduced without including additional information (as it represents a random deviation which is not covered by f) the bias and variance term can be substantially influenced by the complexity of the function $h(x;)$. So for instance, in case of artificial neural networks (ANNs) used as our function $h(x;)$ it depends on the choice of the architecture of an ANN or the algorithm on how to determine the parameter vector ϕ based on the training data.

If the function space of $h(x;)$ is very complex, we can assume that it is able to cover $f(x)$ very well so that we have a small bias term. But it is also difficult to estimate a complex parameter set, we have a high risk of overfitting and a large variance term. If on the other hand we use a simple class of functions $h(x;)$ with a low dimensionality of the parameter vector ϕ, we will be able to estimate the parameters well based on the training data and so have a low variance term, but we will have difficulty to cover the complexity of $f(x)$ by $h(x;)$ so that we have an increased bias term. For additional references and a detailed discussion of these

topics see [Geman 92], [Hansen 00] or [James 96].

The problem to find a good trade-off between error bias and variance is called the bias-variance dilemma. Different alternatives [Geman 92] have been proposed in order to determine a good trade-off between bias and variance while learning the parameters in $h(;\phi)$ or choosing function classes $h(;)$ with an appropriate quality.

4.2 Diversifying Methods

In the previous section we discussed how diversity can be represented and how we can obtain indicators if a given set of input forecasts is sufficiently diverse. But what should one do if it is not? In this section we provide an overview of how the generation of input forecasts can be influenced in order to achieve a set of divers forecasts.

The topic of generating diverse forecasts has mostly been oriented towards the creation of diverse neural networks or decision trees, even if some of the proposed diversifying techniques do not depend on these approaches. In this thesis the focus is not put on input forecasts generated with ANNs or decision trees, that is why we will not discuss issues directly related to these type of input forecasts here. A good overview in relation to ANN specific diversification methods can be found in [Raviv 96]. The following basic ideas of general diversifying techniques are discussed in the literature:

- decompose data and/or predictions
- diversify the function space $\mathcal{H}$
- diversify the training data

In the following subsections some details are provided about the different diversification approaches.

4.2.1 Decomposition of Data and Predictions

Issues Resulting from not Working on Decomposed Data

High covariance between input prediction errors is sometimes caused by the fact that the input predictions represent data composed from components and all of the input predictions predict some of the components in a similar manner. The resulting forecast errors relating to these components are then highly correlated.

This can be demonstrated with the following example: Let us assume a time series $y = y^1 + y^2$ to be predicted with y^1 and y^2 independent components. Let us also assume that we have two predictions given: ${}^1\widehat{y} = \widehat{y}^1 + {}^1\widehat{y}^2$ and ${}^2\widehat{y} = \widehat{y}^1 + {}^2\widehat{y}^2$. Both predictions predict the first component in the same manner, while for the second component different approaches are used. Let us also assume error variances $\delta^2_{y^1} = 2$, $\delta^2_{{}^1y^2} = 0.1$ and $\delta^2_{{}^2y^2} = 0.3$ with $\delta^2_{{}^1y^2}$ and $\delta^2_{{}^2y^2}$ not correlated. The covariance between the two forecasts is $\rho = \delta^2_{y^1} = 2$ because the error made in component 1 exists in both forecasts, the errors of component 2 are not correlated and therefore do not effect the covariance. This means that we have highly correlated forecasts because of a similar prediction of component 1.

If covariance information is not considered, the effect of including such components into a combination is a shifting of the resulting weights in the direction of equal weights.

We will illustrate that using the same example: As long as the weights sum up to 1, the first component is not taken into account by the combination, and the ideal weights depend therefore only on the second component. We achieve $w_1^{ideal} = 0.75$ and w_2^{ideal}=0.25. The same results are provided by the optimal model. If we use only the variance based model, we achieve $w_1^{var} = \frac{\frac{1}{2.1}}{\frac{1}{2.1}+\frac{1}{2.3}} \approx 0.52$ and $w_2^{var} \approx 0.48$. It can be clearly seen that these weights are much more similar so that the advantages of the first model are not sufficiently considered.

If on the other hand we use covariance information for the combination of highly correlated forecasts, we potentially achieve high weight estimation errors

and instabilities based on small deviations in the estimated covariance values. This topic will be discussed in Section 4.4.

The issues that have just been described can be avoided by forecasting different components separately and combining different predictions in relation to each component. The input predictions relating to components where the forecasts really differ are then much more diverse. For components where the input predictions do not differ there is no need for forecast combination.

In our example, this approach would have the following effect: We would predict the two components in a decomposed manner. For the first component there is only one approach given so that there is no need for forecast combination. The prediction of the second component would be a combination of ${}^1\hat{y}^2$ and ${}^2\hat{y}^2$. We would achieve weights $w_1^{var} = \frac{\frac{1}{0.1}}{\frac{1}{0.1}+\frac{1}{0.3}} = 0.75$ and $w_2^{var} = 0.25$ which correspond in this example to the optimal weights.

Automatic Approaches

Some automatic approaches to decompose data into independent components are proposed in the literature. The idea of using these automatic decomposition methods for combination to reduce collinearity is discussed and followed consequently by Merz and Pazzani [Merz 97], who propose the approach of splitting the individual forecasts (not the data!) using principal component analysis as a part of their combination model. A linear combination is carried out on the transformed input forecast set as described in Section 3.3.3.

Approach followed for our Application

Based on measured high covariances of forecast errors if predicting the total demand directly, we have followed the strategy of decomposing demand into the different components as described in Section 2.2.3. The data is decomposed corresponding to estimations of the different components and their confidence. For each of the components forecasts are generated and combined separately. Finally the

combined predictions are aggregated to the final prediction. Figure 23 illustrates this approach.

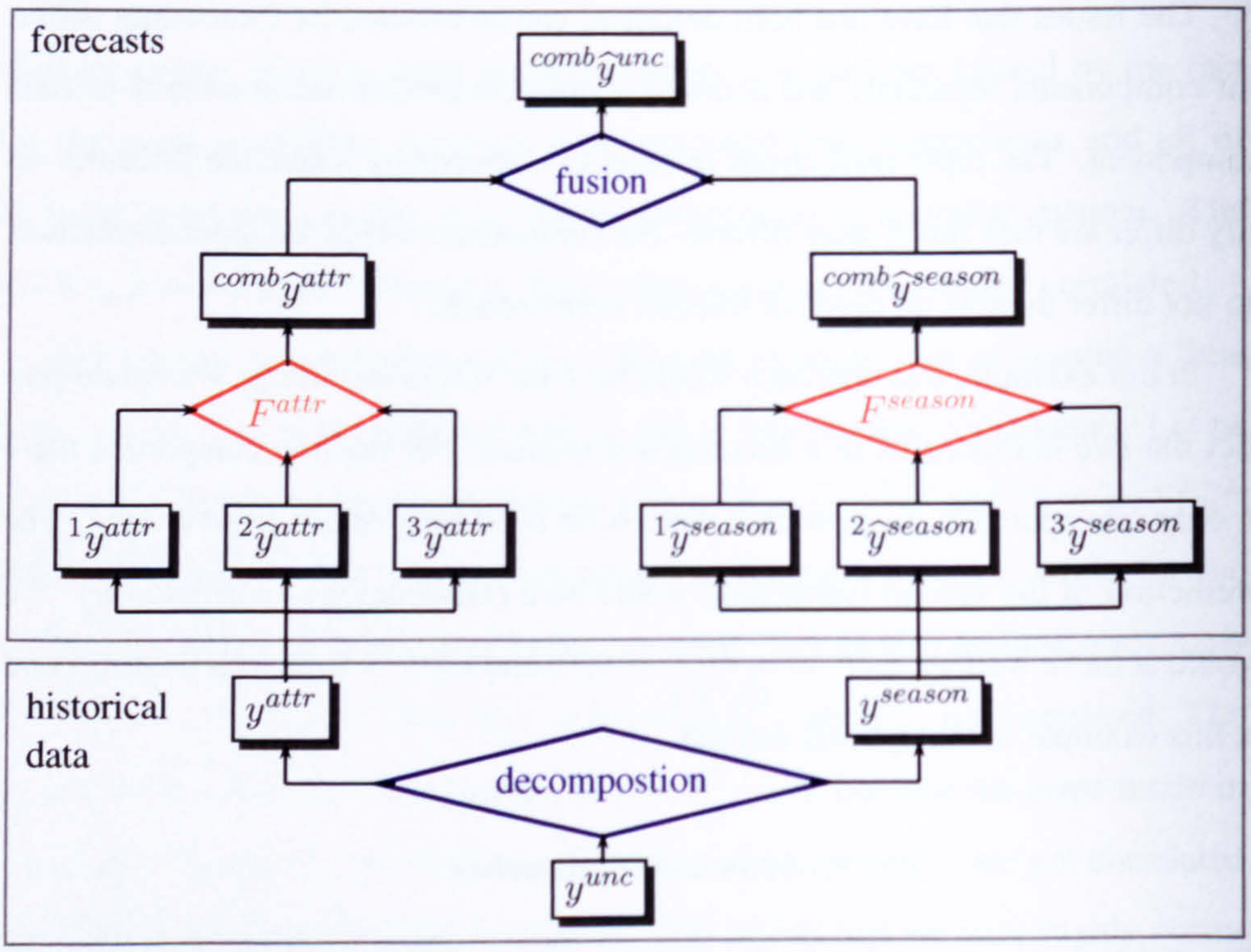

Fig. 23: General decomposition approach followed for the Revenue Management application.

4.2.2 Diversification of the Function Space

A different option in order to diversify forecasts is the use of different function spaces. The potential for diversification based on the generation of structures with varying complexity can be observed by analysing the effects on the different error components. Changing the complexity means a potential shift from the error bias components to the error variance component or visa versa. In Section 4.1.3 we have already argued that this can have an impact on the combination.

One option using forecast combination is based on the idea of combining different individual forecasts with strongly restricted function spaces $\mathcal{H}_k$. Let us assume that we have a set of functions $h_k : \mathcal{R}^n \times \Phi_k \rightarrow \mathcal{R}$ available. These forecasts gen-

erate errors with large bias terms and reduced variance terms compared to more complex function spaces. The idea is then to increase the complexity and therefore reduce the bias term during the fusion process. This can be achieved if the used function spaces $\mathcal{H}_k$ generate bias error terms which are not highly correlated. *So in such a case combination can be viewed as an option to model complex functional relationships on the basis of different less complex approaches.*

Another option is the use of function spaces with different complexities. Forecast combination can be seen as an option in order to find a good trade-off between error bias and error variance term by finding the best combination between approaches of different complexity. This avoids problems of using function spaces which are generally too complex or not complex enough. For a discussion related to these topics see [Geman 92] and [Hansen 00].

Function spaces with different complexities can be achieved for instance by varying the structure of a neural network or varying the algorithm employed in case of time series predictions.

On the other hand similar effects in terms of diversity inducement could be achieved by varying models parameter values like in the case of thick modelling. "Thick modelling" has been first proposed by Granger and Jeon in 2003 [Granger 04]. The general idea is to use different values for a given parameter instead of trying to determine the "optimal" value for that parameter and then to combine the generated predictions. Granger and Jeon describe "thick modelling" as:" *modelling [that] consists of using many alternative specifications of similar quality, using each to produce the output to require for the purpose of the modelling exercise,..., and then to combine or synthesise the results.*"

They motivate the approach of thick modelling by stating: " *Asymptotically, there will be a basic model, using some criterion, and it will be the true model if it is in the set considered. In this case, and only then, is the strategy of using the best model necessarily the superior strategy rather than using a thick modelling approach. As we are rarely in an asymptotic situation in macroeconomics, for*

instance, the more pragmatic approach seems to be superior. An advantage of thick modelling is that one no longer needs to worry about difficult decisions between close alternatives or between deciding the outcome of a test that is not decisive.".

Aiolfi and Favero [Aiolfi 05] state: *"If the process is sufficiently complex, then the reduction strategy can lead to a model which is more weakly correlated with the true model than the combination of different models."*

The advantage of fixing certain parameters is the generation of a less complex vector of remaining parameters which have to be determined during the learning process and so the reduction of the error variance term. Thick modelling can have a variance stabilising effect which is paid for with an increased error bias component. The advantage is that it can be expected that this increased bias can be eliminated by forecast combination because this part of the bias is not due to a model which is too poor but due to diverse restrictions on the function space $\mathcal{H}$.

The general idea of thick modelling consists not only of generating predictions based on different parameter settings but of all kinds of model generation choices including the use of different function spaces. In this thesis we will refer to the term especially in relation to the choice of different parameter values.

4.2.3 Diversification of the Training Data

It is also possible to diversify not the function space, but the data used for training. The following types of diversification of training data have been discussed in the literature.

Using Different Preprocessing

The possibilities of how to change data using different preprocessing are immense. The most common approaches are the extraction of different feature sets from the raw data or the data is differently distorted, e.g., by noise injection (see e.g. Raviv and Intrator [Raviv 96]) or by using nonlinear transformations (see Sharkey [Sharkey 95]).

Using Different Data Sources

Another option to diversify input information is to use data coming from different data sources. It depends on the application if there is the possibility to get such different kinds of data.

Generation of Disjoint Training Sets

Sampling data is a technique to generate different subsets of training data. Different resampling techniques have been developed in order to generate new subsets. The most common is bootstrapping [Schapire 90]. Other authors like A. Sharkey [Sharkey 96] propose methods of generating disjoint or mutually exclusive data sets.

In *random sampling with replacement* the training data set is randomly selected [Schapire 90][Breimann 96]. The subsets do not need to be disjunct, which means that we create a number of different, but overlapping data sets. They may also contain repeats. This resampling technique is used in a popular ensemble creation technique called bagging [Schapire 90]. Bagging has been proposed by Breiman [Breimann 96] and is based on the idea of bootstrapping [Schapire 90]. It uses a weighted majority vote to combine different individual forecast or classification results. In bagging the training set is randomly perturbed by sampling with replacement. The perturbed data may contain repeats, bagging creates a number of different, but overlapping data sets.

In *biased sampling with replacement* the data subsets used for training are influenced by results of previous training. Training data is adaptively resampled. So we can for instance learn problematic cases with an higher impact. This resampling technique is used in another very popular ensemble creation algorithm called Adaboost proposed by Freund and Shapire in 1996 [Freund 96].

Versions handling unbalanced data sets exist [Provost 00][Chawla 04] [Weiss 04][Batista 04][Kotsiantis 06] as well. For classification, a common problem is that classes may occur with unequal frequency. This causes biased estima-

tion [Kotsiantis 03] and suboptimal classification performance [Chawla 04]. One approach to handle that problem is the idea of applying over-sampling for rarely occuring classes and under-sampling for often occuring classes in order to generate balanced training data sets.

4.2.4 Summary

Summarising one can say that there is a strong relation of combination performance to the structure and correlation of individual forecast errors. Independent forecast errors can be achieved using 'divers' individual forecasts, which can be generated using

- different available sources of information,
- different preprocessing,
- different history pools,
- different functional or stochastic approaches or
- different parametrisation.

Very often these diversification procedures are applied in very random manner without a clear understanding of their effect on the combination error. In order to address these issues, an analysis of the different types of diversification in relation to the forecast error components corresponding to the decomposition of James and Hastie [James 96] will be performed in the next section. The purpose of this was to find a way of generating a well performing set of diversified forecasts in a controlled manner.

4.3 Effects of Diversification on the Error Components

In Subsection 4.2.2 we have argued that the complexity of the function space strongly effects the error components. An increase of complexity allows a reduction of the error bias component but also increases the risk of a high error variance component.

Unfortunately, in cases of small number predictions of very noisy data there is the risk that even with a strongly restricted function space we achieve high error variance terms because of the level of noise. For function spaces with limited complexity we can observe a shifting from the error bias to to the error variance term with increasing complexity. Nevertheless, it is possible that the total error does not change much until a certain complexity is reached for which the learning process gets more and more unstable. In this case diversification can help to reduce one or both of the error components. So we can, e.g., choose different function spaces with low error variance terms and expect a reduction of the resulting high error bias terms by the combination.

As an alternative to the choice of completely different functional approaches representing different complexity for combination in an uncontrolled manner, diversity can be reached by the choice of a common function space diversified by different fixed parameter values. We have already mentioned the approach of thick modelling as an option to reduce the complexity of a function space. The advantage of this approach is that we can control which error components will be affected and we can estimate the resulting effects on the generated covariances.

If we want to decide which type of parameters to choose for thick modelling, it is useful to get an impression of the covariances of the diversified predictions. We will therefore take a closer look at the effects of different types of parameter values on the error components. We will see that depending on the chosen type of parameter value the error components are affected in a different manner. This knowledge can be used in order to generate diversified sets of forecasts in relation to error components containing a potential for error reduction. We will also use this knowledge in later sections in order to discuss certain aspects of forecast pooling in Chapter 6.

Setting certain parameters as fixed values instead of learning them automatically represents a) a reduction of the function space if the parameter concerns the functional relationship or b) a diversification in relation to the training data if the

parameter effects learning or preprocessing.

We will now take a look at three different types of parameters with each of them representing a different type of diversification.

4.3.1 Parameters Affecting the Data Selected for Learning

Let us first discuss parameters that control the use of data for learning. Such parameters could, e.g., represent a historical period used for learning or a criterion allowing a random input data sub-selection like the one applied in bagging [Breimann 96]. This case represents the typical approach of thick modelling.

The function space is not affected by such types of parameters. The concrete parameter values ϕ_α just affect the error variance component ${}^{\mathcal{H}_\alpha}\delta^2_\phi$. This type of diversification is ideally used in connection with rather complex function spaces containing a low error bias term δ^2_ϕ. As the learning is based on different data we can expect a low correlation among the different error variance terms.

In case of a random selection of sufficiently large subsets we cannot expect any selection working significantly better than any other selection. We can therefore approximate the error variance terms by a single value ${}^{\mathcal{H}_\alpha}\delta^2_\phi \approx \delta^2_\phi \ \forall\alpha$. We can also expect similar covariances ${}^{\mathcal{H}_{\alpha 1},\mathcal{H}_{\alpha 2}}\rho_\phi \approx \rho_\phi \ \forall\alpha_1,\alpha_2$. The resulting covariance matrix can then be approximated with

$$\Sigma_e \approx \begin{pmatrix} \delta^2_e & \rho_e \\ & \ddots \\ \rho_e & \delta^2_e \end{pmatrix} \tag{4.6}$$

with $\delta^2_e = \delta^2_h + \delta^2_\phi + \delta^2_y$ the total error and covariances $\rho_e = \delta^2_h + \rho_\phi + \delta^2_y$.

Starting from (4.3) and taking into account the fact that we achieve equal weights

because of equal error variances and covariances we get a combined error of

$$
\begin{aligned}
{}^{comb}\delta^2 &= \sum_m w_m^2 *^m \delta_e^2 + \sum_{m_1 \neq m_2} w_{m_1} w_{m_2} ({}^{m_1,m_2}\rho_e) && (4.7)\\
&= \frac{1}{M}\delta_e^2 + \frac{M-1}{M}\rho_e && (4.8)\\
&= \delta_h^2 + \left[\frac{1}{M}\delta_\phi^2 + \frac{M-1}{M}\rho_\phi\right] + \delta_y^2, && (4.9)
\end{aligned}
$$

which decreases with a larger number of forecasts M. This shows that with respect to the correlation ρ_ϕ we can achieve improvement which is dependant on the number of forecasts to combine. We can also state that each of the forecasts contains the same amount of independent information and therefore none of them should be removed from the combination.

4.3.2 Parameters Affecting the Function Space without Influencing the Complexity

A similar effect that we have just observed for the error variance component can be observed for the error bias components if we diversify the function space without changing its complexity. This idea corresponds to the original idea of forecast combination: to use different simple forecast approaches and to generate the complex relationship between the given inputs and the target values by the fusion of these simple approaches. In the ideal case all of the predictions are characterised by low error variance terms. Depending on the correlation of the error bias terms of the predictions it is then possible to generate a more or less significant error reduction of the error bias term.

A typical behaviour if this type of diversification is generated by the choice of function parameters is shown in Figure 24. It can be seen that different chosen parameter values generate estimations of similar quality concerning the bias term, but for extreme values of the parameters the bias error component increases.

In the example shown in Figure 25 we assume that we have to approximate

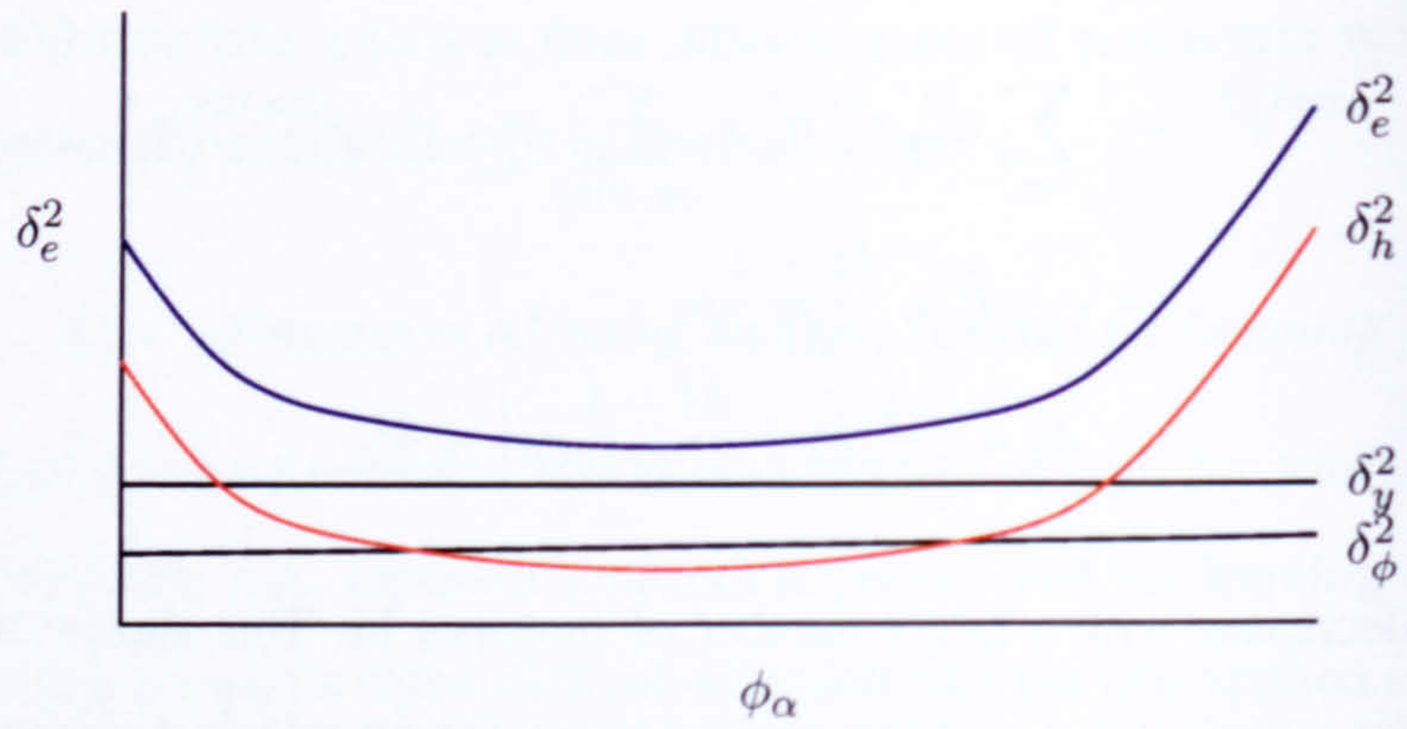

Fig. 24: Typical behaviour of error components in case of a parameter value affecting the error bias component. Extreme values cause an increasing error bias component. The error variance component is only slightly affected.

the polynomial $y = x^2 + 2 * x$ and we use a quadratic polynomial $h(x, \phi) = \alpha * x^2 + \phi_0 * x + \phi_1$ as function space with α diversified by thick modelling. Then we will observe an error increase for very low or very high predefined values of α. The error bias component is shown in Figure 26.

As the resulting functions are very similar we can generally expect high correlation between the different error variance terms. As the other error components are highly correlated as well it makes sense to exclude the extreme values from the combination process as they do not contain sufficiently unique information in order to justify an inclusion even with the higher total error.

4.3.3 Parameters Affecting the Complexity of the Function Space

A parameter that affects the adaptation capabilities often affects both, the error bias as well as the error variance component. Examples for such type of parameters in relation to the revenue management application and learning as described in Section 2.2.6 are

- the smoothing factor used for exponential smoothing ϕ_{sm}
- restrictions related to learned seasonal factors ϕ_{low},ϕ_{high} or

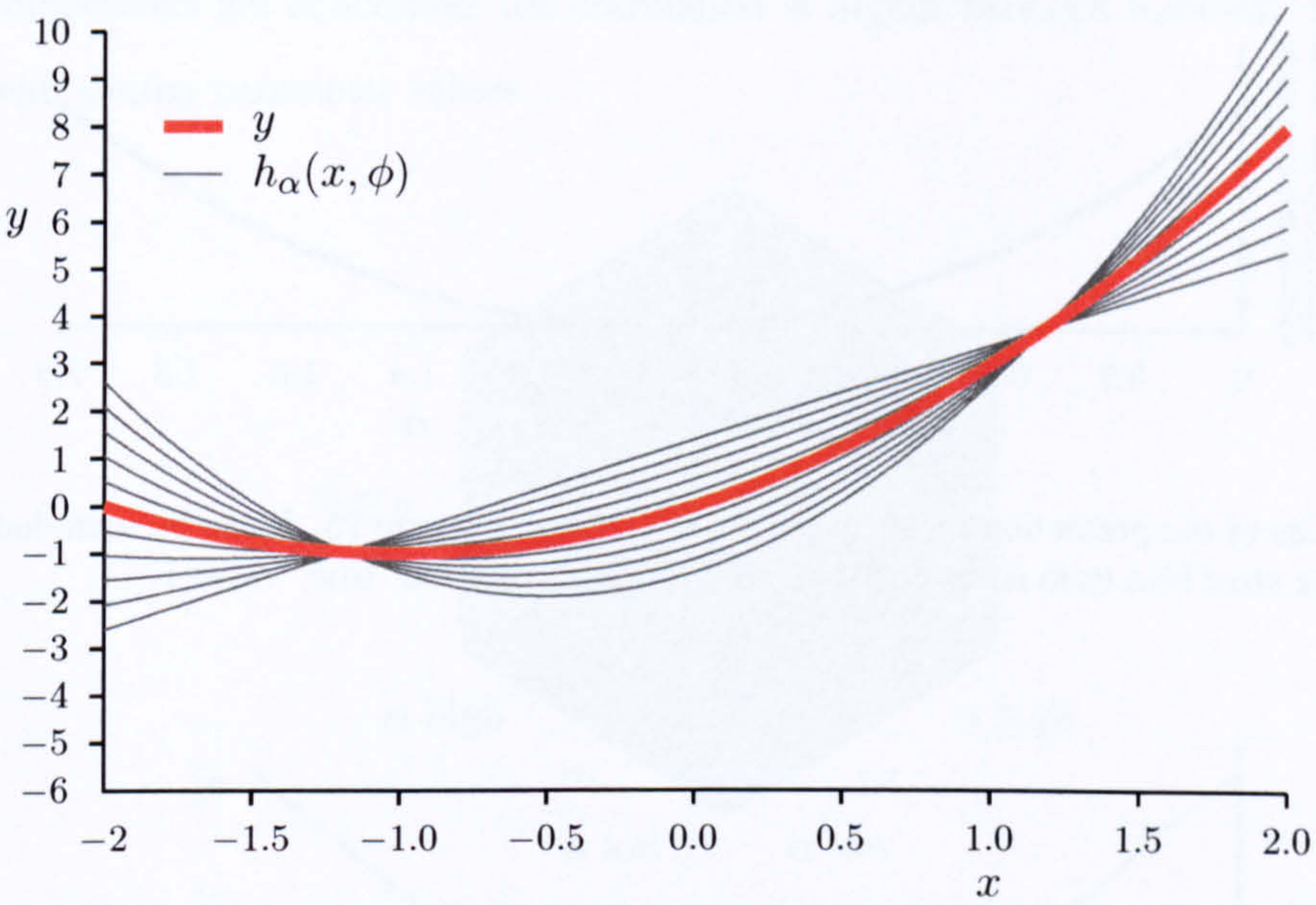

Fig. 25: Example of function $y = x^2 + 2 * x$ with optimal predictions generated using function space $h(x, \phi) = \alpha * x^2 + \phi_0 * x + \phi_1$. The parameter α is diversified, we use values 0, 0.2, 0.4 . . . to 2. The optimal parameters ϕ_0 and ϕ_1 are determined for each prediction in a manner that the quadratic deviation from y is minimised.

- the strength of smoothing of seasonal curves over neighboured weeks ϕ_J.

A typical behaviour looks like that illustrated in Figure 27.

This can be interpreted as follows: Stronger adaptation means higher belief in the data. This means an increase of the estimation error caused by noise represented by the error variance component. At the same time the increase in flexibility causes a decrease of the error bias component.

The figure indicates that extreme values of the parameter lead to higher total forecast errors. We can expect a level of low complexity represented by a parameter value ϕ_α and error ${}^{\mathcal{H}_\alpha}\delta_e^2 = {}^{\mathcal{H}_\alpha}\ \delta_h^2 + {}^{\mathcal{H}_\alpha}\ \delta_\phi^2 + \delta_y^2$ below which the error variance cannot decrease any more. A further reduction to a parameter value $\phi_{\tilde{\alpha}}$ only leads to an increase of the error bias component. As we only increase an already existing error with further reduction of complexity, this increased bias component is highly correlated with ${}^{\mathcal{H}_\alpha}\delta_h^2$. The error variance cannot be reduced any more and is also highly correlated with ${}^{\mathcal{H}_\alpha}\delta_\phi^2$. The resulting error for a parameter $\phi_{\tilde{\alpha}}$ can therefore

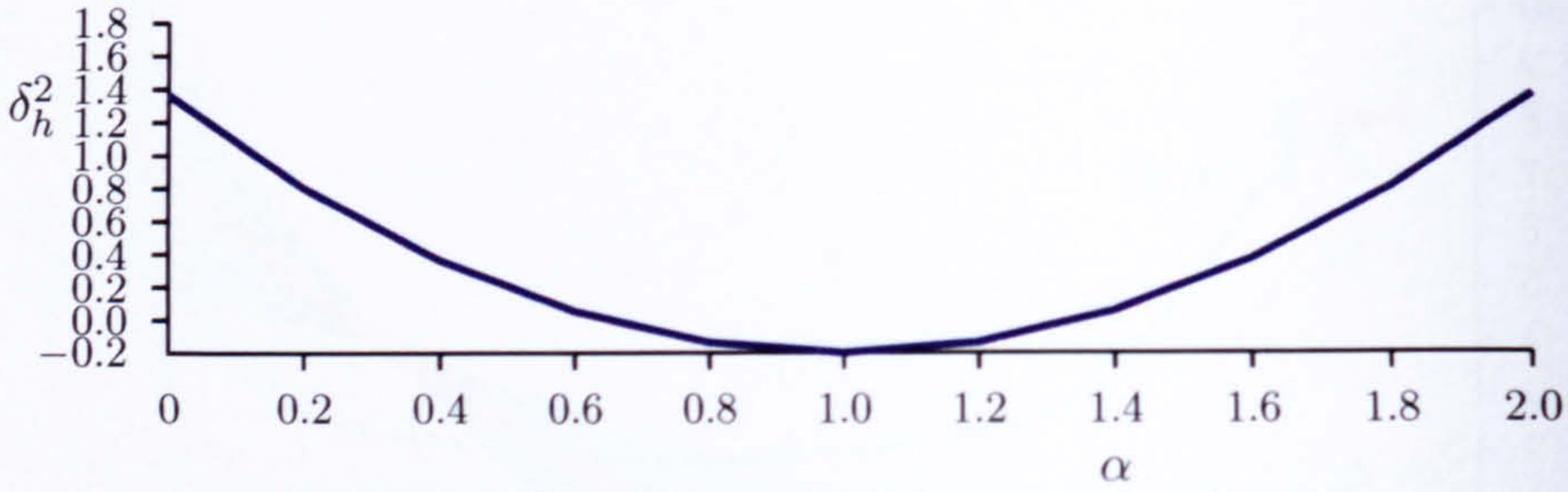

Fig. 26: Bias of the prediction for the example described in Figure 25. It can be seen that the error bias term is lower for parameter values near the "true" value 1.

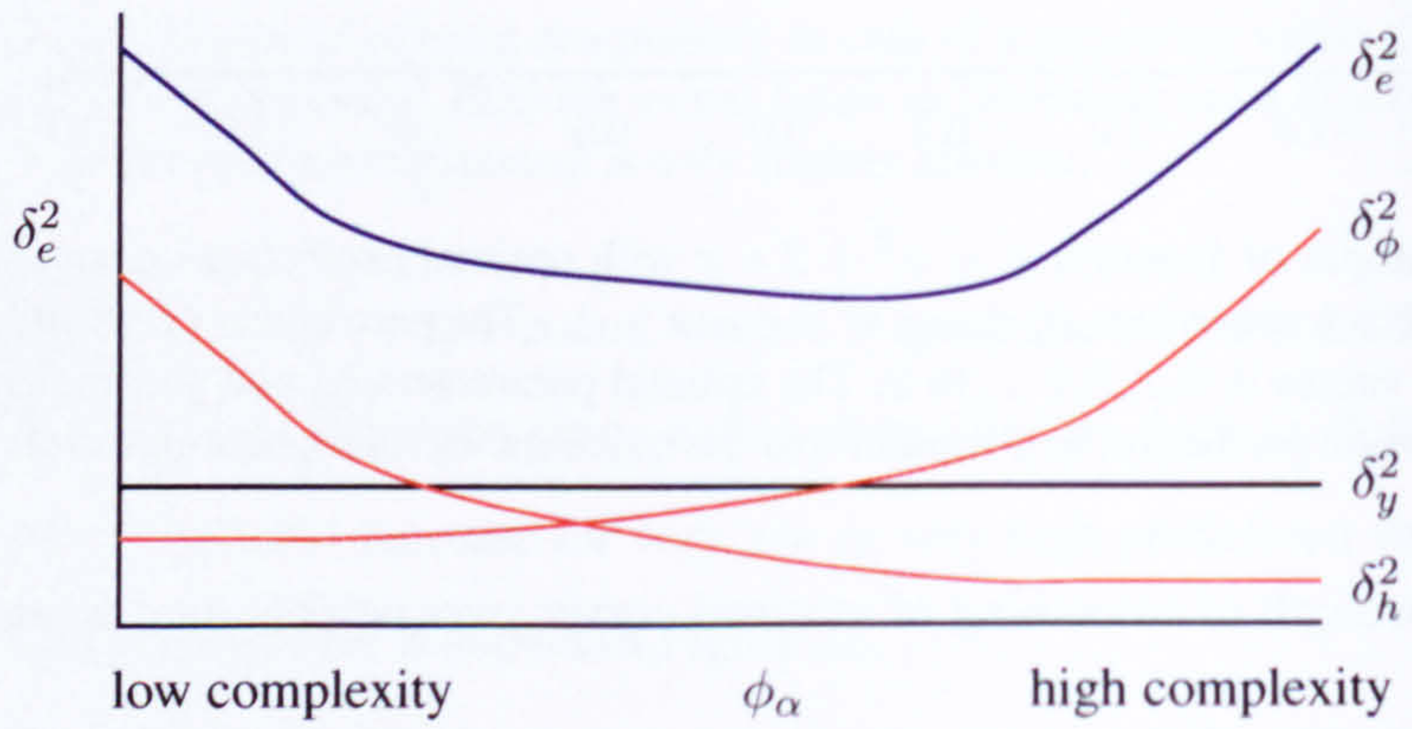

Fig. 27: Typical behaviour of error components in case of a parameter value effecting the complexity of the function space. With increasing complexity we can observe an increase error variance component and a decreasing error bias component.

be approximated by

$$^{\mathcal{H}_{\tilde{\alpha}}}\delta_e^2 \approx \lambda *^{\mathcal{H}_\alpha} \delta_h^2 +^{\mathcal{H}_\alpha} \delta_\phi^2 + \delta_y^2 \tag{4.10}$$

with a factor $\lambda > 1$ and covariance

$$^{\mathcal{H}_\alpha,\mathcal{H}_{\tilde{\alpha}}}\rho_h \approx^{\mathcal{H}_\alpha} \delta_h^2. \tag{4.11}$$

This behaviour also allows assumptions about the covariances, the content of unique information and the potential for combination. Figure 28 illustrates typical covariances of the forecast errors. It can be seen very well that, when both error

components are concerned, the correlation is higher between forecasts generated with similar parameter values.

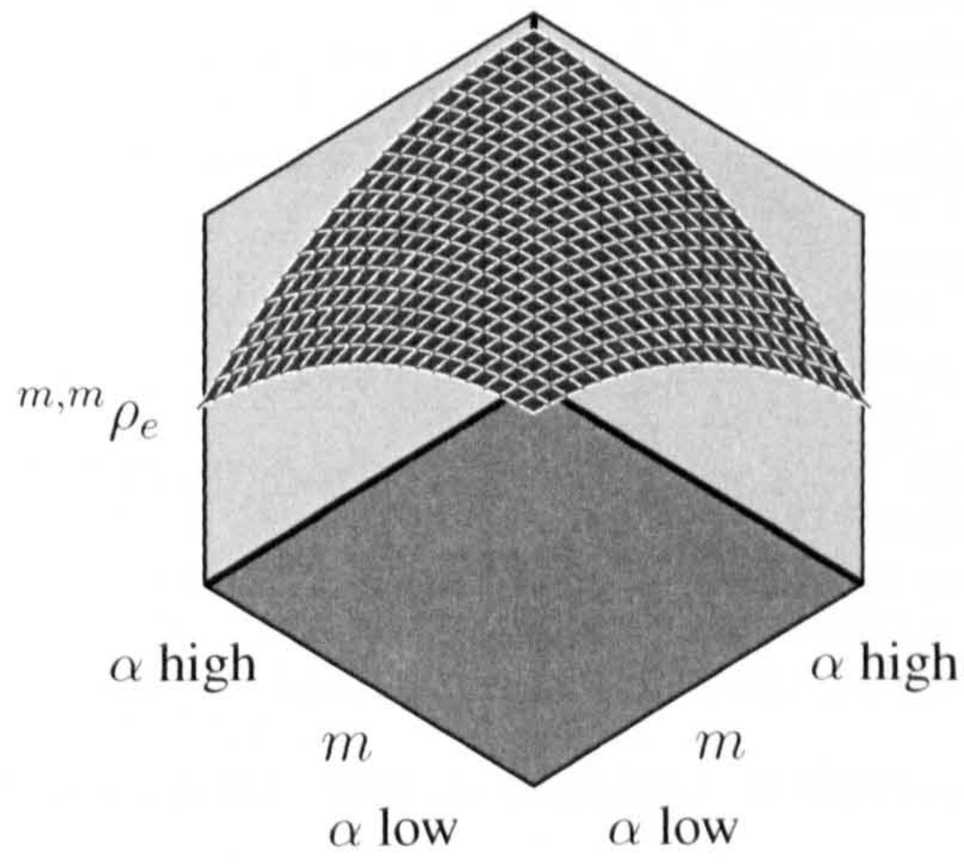

Fig. 28: Typical behaviour of covariances of forecasts diversified by parameter values effecting the complexity of the function space. The index m represents the index of the input forecasts diversified by a parameter α, the z-axis contains the error covariance values.

The fact that the forecasts generated with extreme parameter values are not highly correlated with forecasts representing the other extreme does not mean that these predictions contain any unique information. This cannot be seen by simply looking at the covariances, higher order statistics would be needed in order to determine the level of such information. But if we believe approximations (4.10) and (4.11), we can conclude that the extreme values do not contain any additional information compared to the more stable neighboured values. This means that it is worth removing these values from a fusion process. For this type of diversification this can be done purely on the basis of the total error variances in excluding the worst predictions from the combination. We will come back to the topic of trimming in Chapter 6 and Chapter 7 and see that this does not always hold for other types of diversification.

4.4 The Issue of Weight Estimation Errors

If we generate a set of more or less divers forecasts, we have to answer the question of which combination model to use. As we have seen in the previous chapter, some linear combination models can be seen as generalisations of other models, so why not just take the most general one?

It was a surprising result not only for Bates and Granger who introduced the optimal model [Bates 69], but also in a lot of following studies [Granger 84] [Makridakis 82], that the optimal model, which has been proven to be optimal in theory for unbiased forecasts, seems not to be optimal in practice in terms of combined forecast errors. Even worse, sometimes it performs quite badly. In contrast to these results, the simple average model, which seems to be a poor model in theory, performs very well for a lot of applications. These two phenomena are discussed in the two following subsections.

4.4.1 Why does Optimal Model sometimes perform so badly?

Even if the optimal model is producing weights which are optimal in theory, in practical experiments it is often beaten by most of the other models (see, e.g., [Bates 69],[Granger 84] and [Makridakis 82] to cite just the most popular experiments). The reason is extensively discussed in the literature. One of the most believable explanation is given by Bunn [Bunn 85]. His study covers theoretical as well as practical aspects. A theoretical reason lies in the behaviour of the optimal method for highly correlated forecasts. Bunn showed theoretically for the case of two forecasts to be combined that the generated combinations are no longer convex, with the consequence that the generated weights still sum up to one, but are of opposite sign. Large positive and negative weights can occur which can lead to numerical instabilities in practical applications. The fact that the inverse of the covariance matrix is quite sensitive to small changes in the covariance matrix and that it is generally known only as an approximation may lead to instabilities as well.

These instabilities are especially relevant if the covariance matrix differs only

slightly from a singular matrix. This happens, e.g., in the case of many predictions with about the same error variances and high covariances. In this case the optimal solution strongly depends on errors in the weight estimation. If for instance two similar forecasts are combined, it does not matter if we apply the weights $w_1 = 0.5$ and $w_2 = 0.5$ or $w_1 = -1000$ and $w_2 = 1001$, but slight deviations in the estimated covariances could suggest that the second solution is the better one.

Bunn showed these effects in practical applications by introducing outliers to artificially generated data. He found out that the combinations based on the data disturbed by outliers were much worse than the combination produced by the optimal model on the original data.

4.4.2 Why does Simple Average perform so well?

In an extensive study of the accuracy of forecasting methods, Makridakis et al ([Makridakis 82]) found that a simple average of forecasts from six methods outperformed virtually all individual methods as well as a weighted average of forecasts with weights calculated by the optimal model. Further experiments using the same data set and more models to calculate the weighted average did not confirm the superiority of the simple average (see [Winkler 83]).

Bunn [Bunn 85] proved theoretically and by experiments that if the quality of the individual forecasts is similar in terms of error variance, the ideal weights depend much more on the error variances than on the correlation. That is one of the reasons why the simple average is performing so well in a lot of applications.

Timmermann [Timmermann 05] has derived the loss in the quality of the combined predictions between the optimal model and the optimal model with assumption of independence using an example of two forecasts. Let us assume that we have two forecasts ${}^1\widehat{y}$ and ${}^2\widehat{y}$ generating total error variances ${}^1\delta^2$ and ${}^2\delta^2$ and an error covariance ρ. Then the relative error increase in using the variance based

model compared to the optimal model is

$$l = \left(\frac{1}{1 - \left(\frac{\rho}{{}^1\delta * {}^2\delta} \right)^2} \right) \left(1 - \left(\frac{2\rho}{{}^1\delta^2 + {}^2\delta^2} \right)^2 \right). \tag{4.12}$$

A graphical representation of (4.12) can be seen in Figure 29.

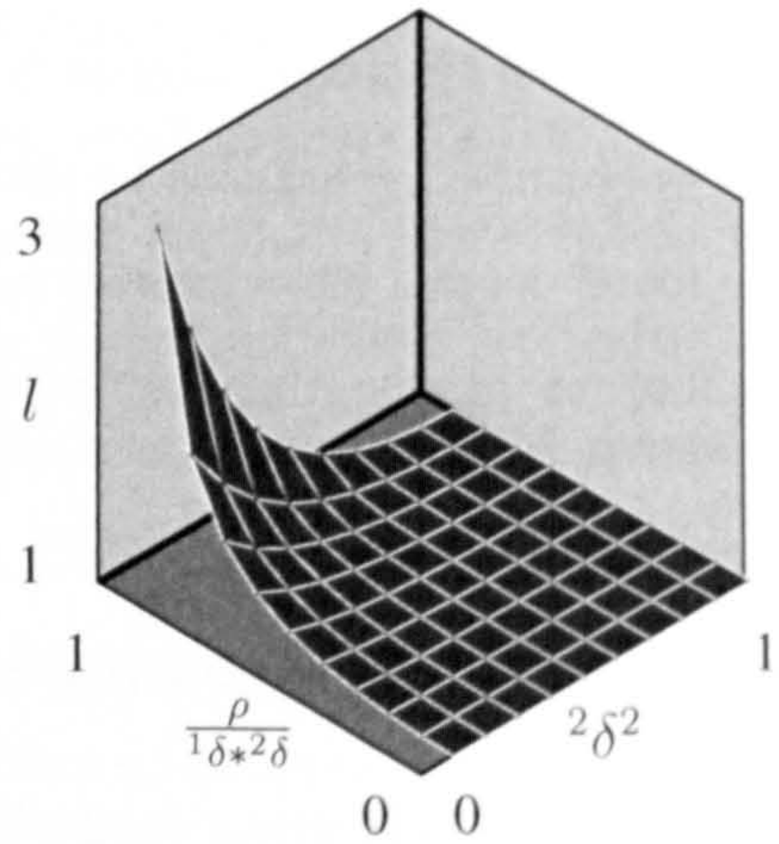

Fig. 29: Graphical representation of equation 4.12. We assume ${}^1\delta^2 = 1$, ${}^2\delta^2$ is shown on the x-axis, the correlation $\frac{\rho}{{}^1\delta * {}^2\delta}$ on the y-axis and the resulting error increase l compared to a combination taking ρ into account on the z-axis.

Figure 29 illustrates on the example of two forecasts that it is risky to combine forecasts for which the errors differ significantly without taking into account covariance information. The biggest losses occur for small values of variance ratio (meaning big differences in the forecast errors) in connection with high correlation values. If on the other hand there are no big differences between the error variances, the covariance is not really relevant for the determination of the weights. In this case, the simple average method will therefore perform well, it is a stable method that does not suffer from instabilities described in the previous subsection and that does not cause big losses because of not using covariance values.

Unfortunately, the fact that covariances do not matter in cases of similar error variances holds only for the case of two forecasts. We will discuss the more general case of more than two forecasts in Chapter 6.

4.5 Guidelines for the Use of Linear Combination Models

The choice of the number of forecasts to combine and the history pool to choose for learning are other issues which are investigated in the literature [Russell 87] [Winkler 83]. We will now discuss how to choose the combination model based on different statistical properties of the individual forecast errors.

4.5.1 The Choice of the Number of Forecasts to Combine

Different opinions can be found in the literature about the question how many forecasts to combine and how to choose the appropriate models to combine.

The preferences about the number of forecasts to combine differ from "not more than 3" (see e.g. [Newboldt 74]) to "as many as possible" (see e.g. [Granger 98]).

As ones of the first, Russell and Adam [Russell 87] discussed the question of how many forecasts to combine. They proposed to determine a relatively small number of forecasts to combine (they used 5) and then to choose the models to combine selectively depending on the ranks of the individual models. The advantage of this approach is that it is able to adapt to changes of the performance of the individual models.

In 1983 Makridakis and Winkler [Winkler 83] found that the accuracy of combined forecasts was little influenced by the specific methods included in the combination. Furthermore, it was shown that accuracy increased with increases in the number of methods being combined, although a degree of saturation was reached after about four or five methods. Finally they observed that the variability of accuracy among different combinations decreased as the number of methods included in the combination increased.

In 1989, Armstrong [Armstrong 89] proposed to combine only "sensible models" which predict reasonable results. It can easily happen that models are too complex and generate implausible predictions which lay outside of the expected range of the target variable. Armstrong and others argue that the inclusion of forecasts

that add only marginal information should be dropped in order to avoid increased parameter estimation errors. Instead of combining all forecasts, it is therefore often advantageous to discard the models with the worst performance (trimming). These results have been confirmed in newer studies for instance of Granger and Jeon [Granger 04] in 2004 or in the context of multiple classifier systems by Ruta and Gabrys [Ruta 05] in 2005.

4.5.2 The Choice of the History Pool

The discussion about the appropriate history pool (the historical data which is used to determine the combining weights) goes back to the seminal paper of Bates and Granger [Bates 69]. They studied various variations of combination models handling the fact that the performance of the individual forecasts may change over time.

Including also the results of other studies [Makridakis 82][Makridakis 93] we can say that the performance of real application forecasts changes over time indeed and that approaches which are able to take this into account generally perform better. A common approach to enable the combining process to adapt to new situations is to restrict the history pool and not to take into account very old data. Approaches which have been proven to be even better experimentally use older data, too, but give more weight to recent forecast errors than to those of the past which allows the weights to adapt quickly to new situations without ignoring the older information.

Bates and Granger stated that the methods of weighting or choosing historical data should be designed in a manner that if the individual forecast performance is stationary, the weights should quickly approach the optimum value and vary only a little from this value. They proposed to use exponentially smoothed variance values measured each on a restricted past period and got very good results with this approach. They surprisingly achieved bad results by approximating the error variance using always only the newest historical error value and then smoothing these values over time.

4.5.3 The Choice of the Combination Method based on Other Statistical Properties

Under stable conditions with 'good' data a lot of studies have shown that the relative performance of combined predictions depends mostly on the following factors: the variance of the forecast errors, the correlation between the forecast errors and the set of data which has been chosen for training.

Under the criterion of minimisation of the variance of the forecast errors and based on experiments carried out by Schmittlein et al. [Schmittlein 90] with the objective to use automatic switches between combination models, Menezes et al. [de Menezes 00] propose the following practical guidelines for the combination of predictions:

- For small data samples use the outperformance model, because it is a simple and stable model which can profit from the differences in the variances of the forecast errors.
- For medium data samples with low correlation the optimal model with assumption of independence should be used.
- For large data samples an optimal model or a restricted regression model will perform very well.

Generally, Menezes et al. indicate that if the forecast error variances are similar and no or only a small positive correlation exists, the simple average should be used.

Bunn [Bunn 85] even suggests that the optimal model should not only be used if a large history pool of data is available, but also if the pattern of forecast errors is unbiased, normally distributed and stationary over time.

Similar results have been reported by Klapper [Klapper 98a]. The author proved experimentally that the performance of the different combination models depends highly on the variance-covariance structure of the individual forecast errors and that the rank based and variance based models beat the covariance based mod-

els for forecasts having a low correlation. He suggests only to use the optimal models if the forecasts are highly correlated. He confirms the idea of Granger and Ramanathan [Granger 84] that the optimal model is often unstable because the weights are not restricted to be larger than 0 and the covariance matrix may not be estimated accurately enough. Therefore it sometimes produces extreme positive and negative weights that come up with nonsense combined values (see also Bunn in the previous subsection).

Beside the variance of the forecast errors, the distribution of the errors of the individual forecasts should be considered when a combination model is chosen. Different distributions imply different risks. Menezes et al. [de Menezes 00] point out two facts which are especially interesting from the practical point of view:

- If different combination models lead to different distributions of the forecast error, then the position with regard to the risk of each individual forecast is an additional factor when choosing the combination method.
- If the combination of different individual forecasts leads to different distributions of the forecast error, then the choice of the predictions to combine is especially important.

In the context of asymmetrically distributed forecast errors the criterion of the skewness of the distributions lead to the following guidelines:

- For small data samples use the outperformance model.
- For medium and large data samples use the optimal model with assumption of independence if there is only a small positive correlation, else the optimal method with restricted weights is suggested.

The authors also give the following advice:

- If individual forecasts are chosen for combination, different distributions of the forecast errors should be considered.

- As many predictions as possible should be included for combination. The use of a larger number of predictions may not give additional information in terms of error variance, but it can improve the distribution of the error of the combined forecast, thus reducing the risk.
- For the analysis of the results not only the mean forecast errors should be analysed, statistical measurements which indicate asymmetric behaviour in the forecast errors, such as the median, should be used as well.
- If a simple average is chosen for combination, it should be clear that a skewness in the predictions will also remain in the combined forecast.

Even if these guidelines propose a development from less complex (outperformance model) to more complex (optimal model) models, they differ from the guidelines proposed on the basis of error variance terms. The differences exist not only in the propositions concerning the use of the simple average, but also in the number of predictions which should be used.

Only a small number of authors studied the autocorrelation of the forecast errors of the combined forecast.The summarising studies of Menezes et al.
[de Menezes 00] show

- that autocorrelation in the individual forecasts can only partly be reduced, sometimes it is even enforced and
- that different combination methods produce different autocorrelation behaviour in the combined forecast.

Based on the results of the study, the following guidelines are given:

- For small data samples use the simple average.
- For medium or large data samples use the optimal method, for which independence is expected if the cross correlation is small. In the other case, the weights should be restricted or a regression-based approach with restricted weights should be used.

- If an autocorrelation can be determined in the combined forecast errors, the complete forecasting approach should be revised.

4.6 Experiments

4.6.1 Description of Experiments

We have carried out experiments in relation to decomposed data and diversification procedures. A data analysis of decomposed forecast errors indicated that the predictors of the seasonal behaviour are much more diverse than the predictors of the attractiveness. That is the reason why the experimental analysis is focused on diversification of the seasonal predictions. For prediction of the attractiveness component we have always used the best performing model which is the model ${}^1h(x,\phi)$ (simple exponential smoothing).

Two types of diversification have been applied. The function space has been diversified with the models $h_1^{season}(x,\phi)$ and $h_3^{season}(x,\phi)$. Diversified parameters applied for the calculation of seasonal factors: ϕ_{low} and ϕ_{high} (lower and upper limit of expected seasonal behaviour). In order to generate sets of range limits which are not completely unbalanced, the initial parameters chosen for $\phi_{low} = -0.3$, and $\phi_{high} = 2$ have been dumped with different factors between 0 and 1. The application of factor 0 in model $h_3^{season}(x,\phi)$ leads to model $h_2^{season}(x,\phi)$. This can be seen if we compare (2.17) with (2.18) using $\phi_{low} = 0$ and $\phi_{high} = 0$. That is the reason why model $h_2^{season}(x,\phi)$ has not been included directly into the diversification process.

The results can be reproduced with experiments 4 (see Appendix B.6.4).

4.6.2 Experimental Results

Table 10 shows the errors of the forecasts containing combined seasonal predictions as relative improvement in relation to the best individual forecast ${}^0\widehat{y}$ measured at the low level (ODO F POS). In order to consider all effects contained in the data,

the improvement has been measured for the generated total forecasts containing the attractiveness component, not only for the seasonal component. A graphical representation of the absolute total errors at the high level (ODO) is shown in Figure 30.

τ	F^{av}	F^{outp}	F^{var}	F^{opt}	F^{ols}	F^{dyn}	F^{appr}
0	-0.02	0.00	-0.01	-0.01	0.01	0.01	-0.05
1	-0.01	0.00	0.00	-0.04	-0.13	0.02	-0.03
2	0.00	0.01	0.00	-0.02	-0.32	0.05	0.04
3	0.00	0.01	0.00	-0.02	-0.20	0.05	0.04
4	0.00	0.01	0.00	-0.02	-0.27	0.04	0.03
5	0.00	0.01	0.00	-0.02	-0.18	0.04	0.03
6	0.00	0.01	0.00	-0.02	-0.22	0.03	0.02
7	0.00	0.01	0.00	-0.02	-0.26	0.03	0.02
8	0.00	0.01	0.00	-0.01	-0.22	0.03	0.02
9	0.00	0.01	0.00	-0.01	-0.17	0.01	0.02
10	0.00	0.01	0.00	-0.01	-0.19	0.01	0.01
11	0.00	0.01	0.01	-0.01	-0.19	0.01	0.01
12	0.00	0.01	0.01	-0.01	-0.23	0.01	0.01
13	0.01	0.01	0.01	-0.01	-0.21	0.01	0.01
14	0.01	0.02	0.02	0.00	-0.19	0.03	0.02
15	0.01	0.02	0.02	0.00	-0.21	0.03	0.02
16	0.02	0.03	0.03	0.01	-0.22	0.04	0.02
17	0.02	0.03	0.03	0.01	-0.23	0.04	0.02
18	0.03	0.03	0.04	0.02	-0.26	0.06	0.02
19	0.03	0.04	0.05	0.04	-0.23	0.06	0.02
20	0.04	0.05	0.06	0.01	-0.30	0.06	0.02
21	0.09	0.10	0.11	-0.64	-0.38	0.07	0.05
22	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Tab. 10: Relative improvement using forecast combination of diversified seasonal predictions in comparison to the best individual forecast ${}^0\widehat{y}$. The columns represent the results achieved with different combination models. Positive numbers mean than an improvement compared to the best individual forecast could be achieved (for instance 0.01 means an error reduction of 1%), negative values indicate that the best individual forecast could not be improved.

We can see that now we are able to slightly improve on the best individual forecast. An improvement of up to 3 to 5% could be achieved in early dcps at the high level. Combination models F^{outp}, F^{dyn} and F^{appr} beat the best individual forecast. The nonlinear models perform well too, the best results in early dcps have been achieved with the nonlinear models. These results could still be slightly improved by applying these combination models on selected subsets of the input forecasts.

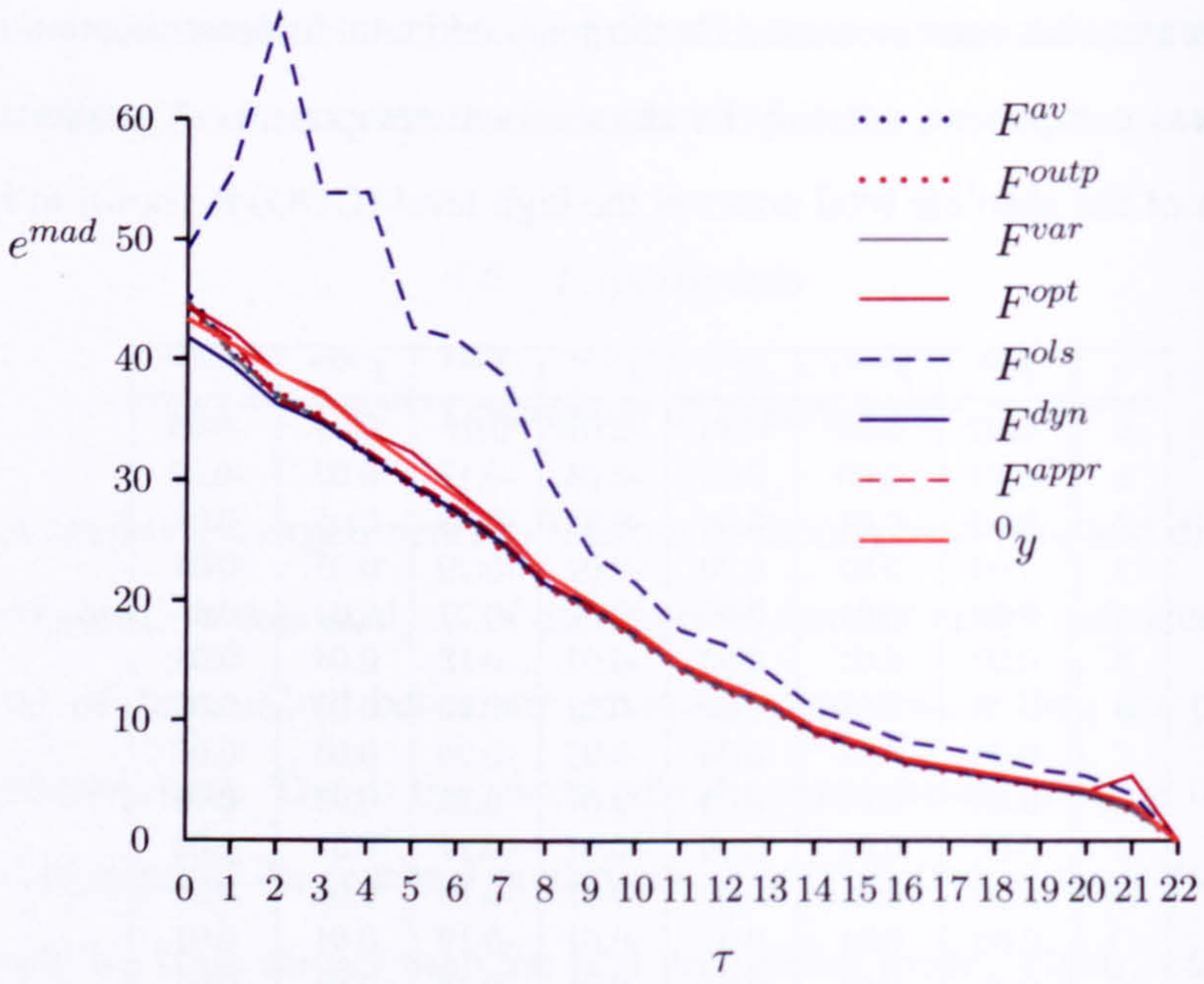

Fig. 30: Errors (mean absolute deviation measured at the ODO level) achieved using forecast combination of diversified seasonal predictions in comparison to the best individual forecast ${}^0\widehat{y}$ (see 2.3.3).

The fact that we achieve an improvement of *only* 3 to 5% can again be explained by the covariances of the diversified forecasts. Figure 31 shows an example for error covariances of diversified seasonal factors. It can be seen that the correlation between the predictions is still high. The example also shows that we can see relevant differences in the structure corresponding to the different types of diversification.

4.6.3 Analysis of Decomposed Forecast Errors

An analysis of the decomposed errors in relation to the error decomposition proves that the high covariance values are mostly based on high error variance and error Bayes terms. Noise in the data lead to the wrong estimations of historical or current seasonal factors. And as models $h_2^{season}(x, \phi)$ and $h_3^{season}(x, \phi)$ operate on the same input data, the resulting error variance components are additionally highly correlated.

Large error variance terms occur even for parameter values generating very

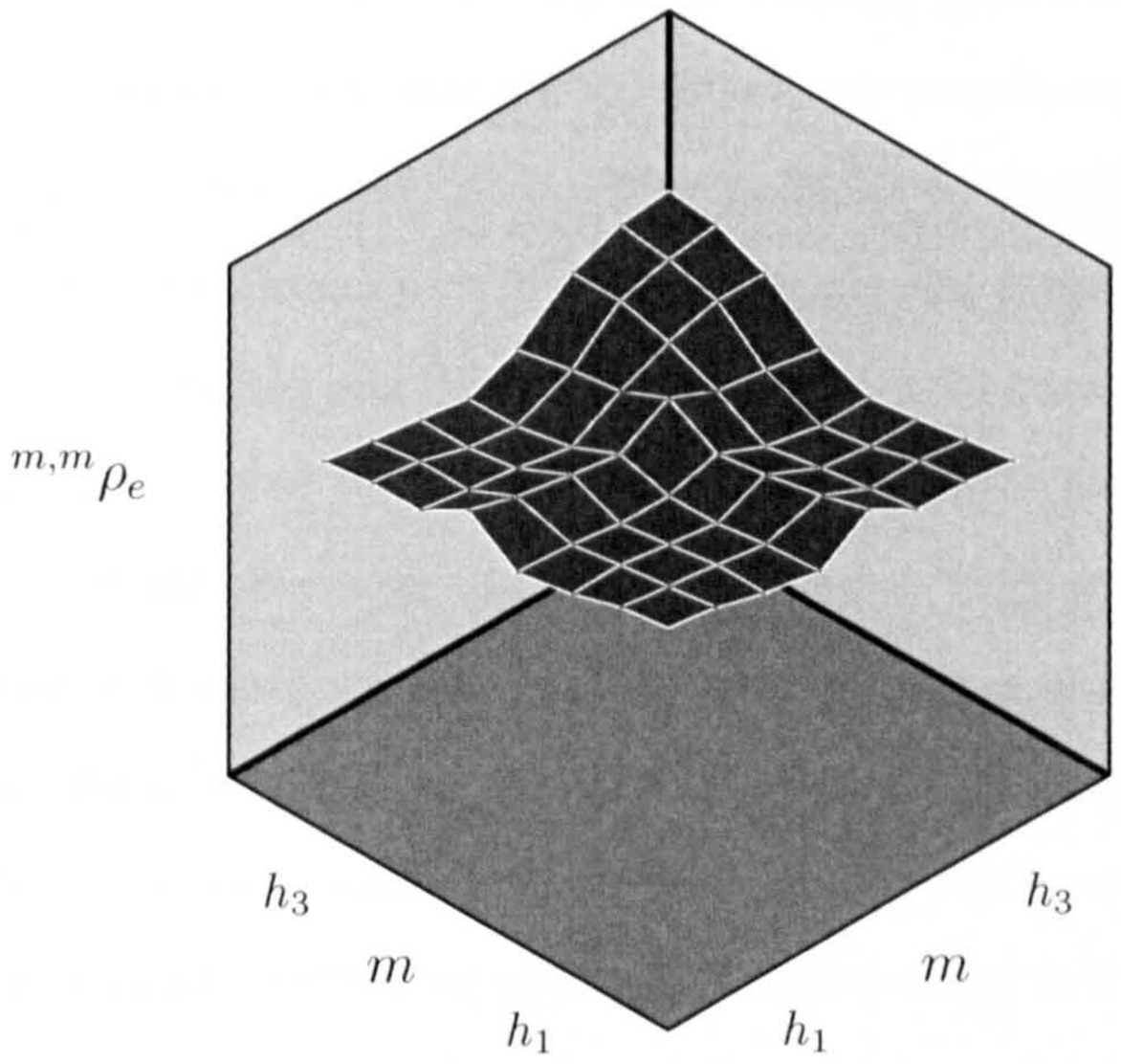

Fig. 31: Example of typical behaviour of covariances of forecasts diversified by more than one type of diversification. The index m represents the index of the input forecasts diversified by parameter ϕ_{low} and ϕ_{low} and by the use of function spaces $h_1^{season}(x, \phi)$ and $h_3^{season}(x, \phi)$. The z-axis contains the error covariance values. The four parts representing the different combination of function spaces can be distinguished very well.

restricted function spaces. That is the reason why the applied parameter diversification could not sufficiently reduce the covariances between the predictions. In contrast to the error variance terms, the error bias terms are lower and more diverse. These error terms could theoretically eliminate each other during the combination, but the high error variance and error Bayes terms lead to high weight estimation errors as described in Section 4.4.

4.6.4 Conclusions

The experiments show that data decomposition and diversification is beneficial for forecast combination. We could achieve better results with combination of not decomposed and diversified data. Especially the nonlinear combination models

perform much better.

Nevertheless, it was only possible to slightly outperform the best individual prediction. An analysis of error covariances of the seasonal predictions shows that even if we have improved the situation with the decomposition and the diversification, we still have highly correlated forecasts containing highly correlated error variance components even for the more robust models and parameter settings.

Therefore, we have to search for alternatives of how to generate predictions which are diverse in relation to the error variance component. We have seen that this objective could be achieved by using different types of data for learning. We will therefore enter into a discussion about learning at different levels in the next chapter.

5. COMBINATION OF FORECASTS GENERATED WITH MULTI LEVEL LEARNING

In this chapter we will discuss issues related to real world hard forecasting problems like our application which are characterised by large noise terms in the training data, frequently occurring structural breaks and quickly changing environments. We will address real world applications in which not a single prediction has to be generated, but a lot of predictions representing the situation in concrete subspaces of a given input spaces. If we have to generate predictions for seasonal effects of airline demand, we need to do this for different origin destination pairs as well as different fareclasses. The level on which the predictions have to be generated is often very detailed (like the seasonal behaviour for a given origin-destination airport pair and a given fareclass), but analysts or related computer systems also use aggregates of the generated forecasts to higher levels (like the seasonal behaviour related to the trafic between countries). The aggregates are used for decision making or further calculation, e.g., in terms of reports or in using a graphical user interface showing the expected situation at different levels.

The reaction to large noise components and in consequence structurally poor forecasts at the fine level of forecasting is often the decision to learn structural information or causal effects at higher levels meaning learning based on aggregates of the target data. This decreases noise but leads to an information loss related to effects which occur only at the fine level.

The question of which level to choose for learning is not obvious. The topic is discussed in the literature as "hierarchical forecasting". Common strategies of defining hierarchies or families of levels and working with aggregates or splits of

forecasts have been summarised by Flieder [Fliedner 01] in 2001.

We will discuss this topic on the example of two levels and see that choices which are purely based on the total error variance at the fine level of forecasting do not need necessarily be the optimal choice with regard to aggregates to higher levels. We will also address the question of information loss if only a single level is chosen for learning and discuss different options of how to incorporate multi level information. We will motivate why we think that forecast combination is a very promising approach in order to deal with this problem and discuss special questions of combining forecasts generated at different levels.

While typically forecasts are combined in a flat manner as denoted by equation (3.1), in this chapter we will lead an error component based discussion for the case of multi level forecasting. We discuss the topic of what happens if ϕ is learned at different levels and what alternatives exist in order to incorporate multi level information. This includes a discussion of the effects of forecast combination on the error bias and error variance component for different cases at the low and the high level.

5.1 Multi Level Forecasting

5.1.1 The Problem of Determining Appropriate Levels

In real world forecasting problems we often do not have to predict future values of only a single time series but the situation in an application defined input space. This is realised by splitting the input space into subspaces and generating time series predictions related to each subspace. In addition, the generated predictions are often visualised and used not only on the subspace level (which we will also call the fine/low level) but as an aggregate representing the expected future situation at the level of the total input space (the high level) as shown in Figure 32.

Let us take our application of seasonal booking behaviour predictions as an example. As the number of potential ODIFPOS combinations to analyse is very

big, graphical user interfaces offer the possibility to analysts to look at the data not only at the ODIFPOS level, but at aggregates of the booking data representing the ODI level including sums of bookings over all fareclasses and point of sales or even at higher levels in order to keep the overview and get an impression of the overall seasonal behaviour. Figure 32 shows an example taken at the ODI and the ODIFPOS level. It can be seen that this higher level is characterised by a much lower noise because of larger booking values.

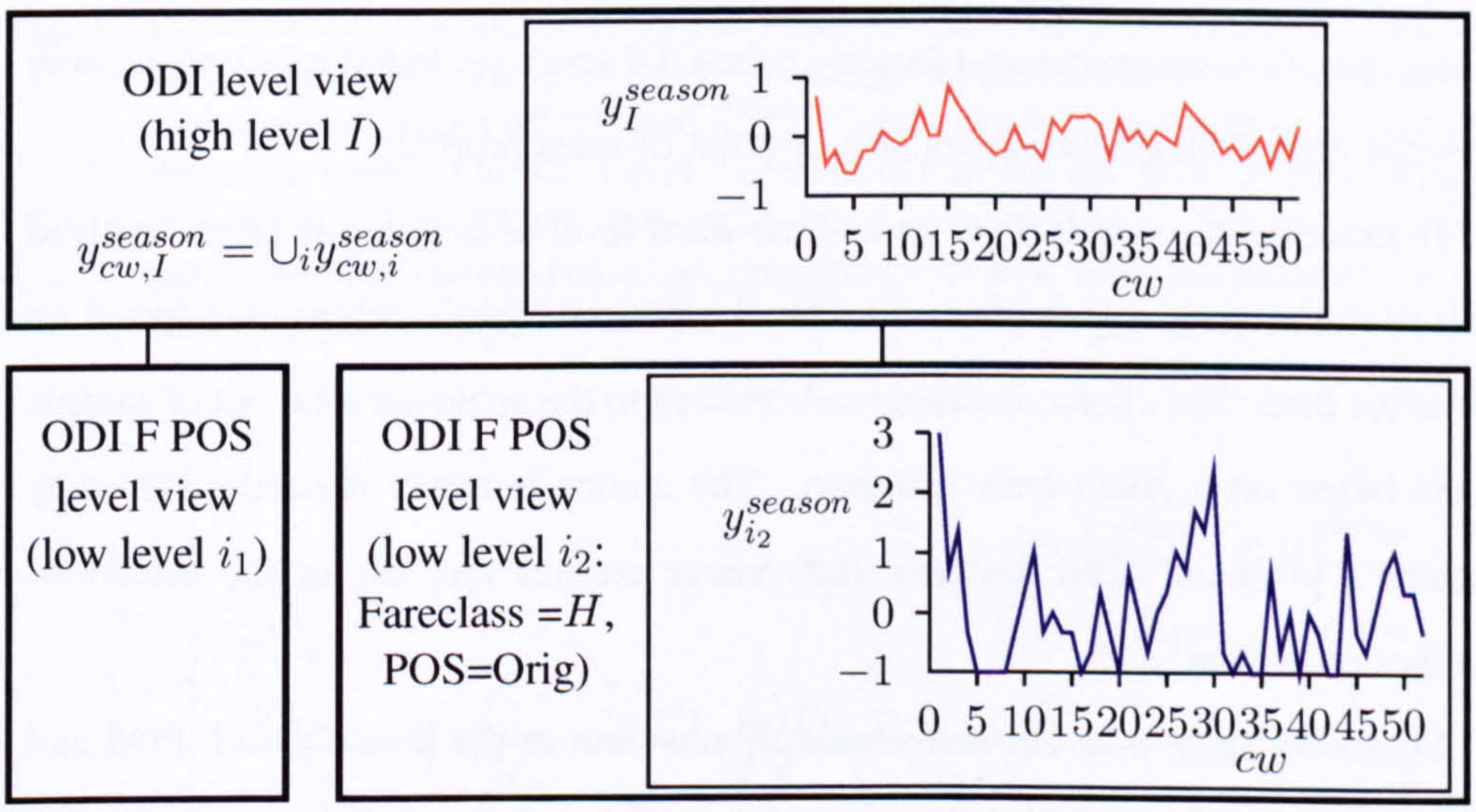

Fig. 32: Seasonal factors measured at the ODI and the ODIFPOS level.

While we assume the fine level of forecasting to be defined by the application, a crucial problem is to determine appropriate levels/subspaces on which the models are calibrated or structural characteristics are learned. So we would have to decide for our example if we learn seasonal factors at the aggregated ODI level or the fine ODIFPOS level.

The choice of the different levels for learning is related to different types of risk. If a level is chosen too fine, there is a high risk of undesirable large noise terms in the training data.

In Figure 11 shown in Section 2.2.6 we presented two learned representations of the seasonal behaviour. As we have seen in that section, both learned curves

have problems to model the seasonal factors properly. With learning method 1 we achieve unstable forecasts because of the high noise in the training data. Learning approach 2 has limited complexity and is too poor to model the true seasonal behaviour. Because of the large noise relevant structural information could not be detected any more. This means that we have a high error bias term as well as relevant parts in the error variance component.

If on the other hand the chosen level is too general, important characteristics related to special parts in the input space may be ignored. For our example this means that if we learn seasonal factors at the ODI level we do not take into account seasonal effects in special fareclasses or point of sales properly.

In practice, the problem of finding the ideal level of learning is often resolved with trial and error approaches. The level of calculation is determined based on static test data. The choice is often made related to the achieved total out of sample errors based on a given error criterion. The choice is rarely revised. This may become a problem when the error differences are not very big or the behaviour changes over time.

Figure 33 shows the learned seasonal behaviour at the level ODO F POS and the level ODO COMP POS together with the achieved seasonal factors in the following year. It can be seen that the low level seasonal curve shows different characteristics in comparison to the high level curve. While the low level curve matches much better in weeks 40-52, the high level curve fits slightly better in the middle of the year, so that it cannot be said that the low level curve is the better one in general. This can be seen more clearly in Figure 34, which shows the error which would have been made using the different curves to predict the following year at the fine level. It can also be seen that we have errors which are not strongly correlated which indicates a potential for forecast combination.

Another issue in choosing the level for learning based on out of sample errors achieved at the forecast level is that this choice could be unfavourable with regard to the aggregated forecasts representing the situation at the higher level. If relevant

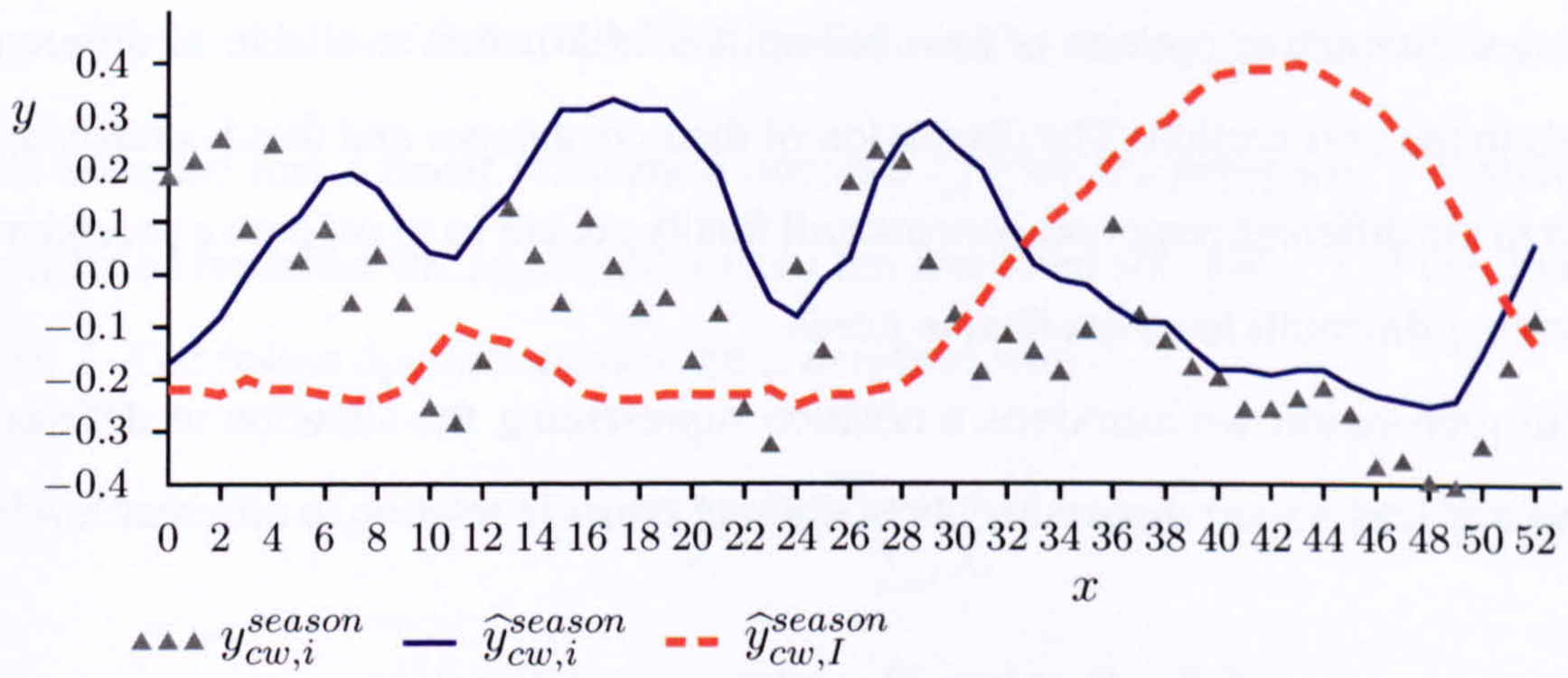

Fig. 33: Out of sample seasonal behaviour together with seasonal factors learned at the different levels. The example represents data generated for ODO=19, Fareclass=16. The seasonal factors $\widehat{y}_{cw,i}^{season}$ and $\widehat{y}_{cw,I}^{season}$ have been learned based on the data of departure weeks 0 to 52. Level i represents learning per ODO F POS, level I learning per ODO COMP POS (with data aggregated over fareclasses per compartment). The learned factors are compared with low level data measured in the following year in departure weeks 53 to 105.

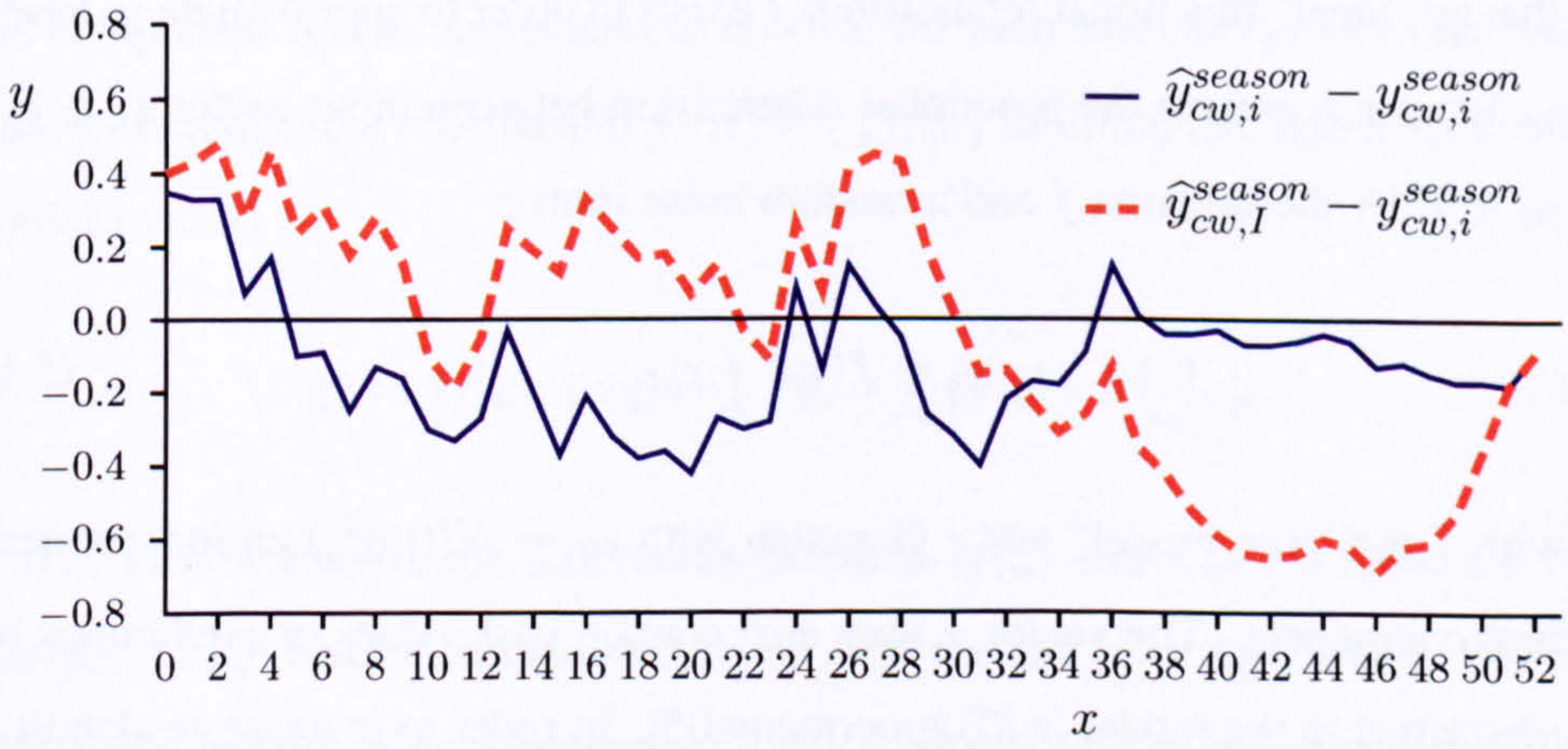

Fig. 34: Errors generated with the predictions shown in Figure 33. It can be seen that the errors are not strongly correlated.

decisions are made by analysts based on aggregates of the low level forecasts to the higher level, this fact should be taken into account for the choice of the level of learning structural information.

We will analyse the different impacts of different error components on the quality and stability of the forecasts at the different levels. We will start by discussing

different alternative options of how to use the information available at different levels in the next section. The discussion of their advantages and disadvantages related to the different error components will finally enable us to propose a procedure satisfying the multi level qualitative needs.

In preparation we introduce a notation representing the situation at different levels and lead a short discussion about optimal errors in relation to different levels.

5.2 Problem Description and Notation

5.2.1 Notation of Multi-Level Time Series

As in the previous chapters we discuss causal models representing relationships between time series $x_t \in \mathcal{R}^n$ and $y_t \in \mathcal{R}$, with t representing a time index. We further assume that x_t can be measured properly, that we have random noise in y_t and that an "ideal" functional relationship f exists in order to approximate y_t based on x_t. We can represent the functional relationship between input vector $x_t \in \mathcal{R}^n$ and $y_t \in \mathcal{R}$ by the function f and a random noise term ϵ:

$$y_t = f(x_t) + \epsilon_{yt}, \tag{5.1}$$

with f the "true model" and ϵ Gaussian with $\epsilon_y \sim N(0, \delta^2_{\epsilon y})$ an independent residual component. The vector x may also contain past values or predictions of y as described in the model in [Timmermann 05]. In order to increase readability, we will remove the parameter t in all following equations.

Let us now assume that we do not have to predict a single time series but a whole set representing different subspaces of an input space. We will use the index i in order to indicate any given subspace (the fine/low level) for which we have to generate predictions:

$$y_i = f_i(x) + \epsilon_{yi}. \tag{5.2}$$

Let further index I indicate values or measurements concerning a high level view.

5.2.2 The Relation Between y_i and y_I

It is assumed that a linear unification operator $\bigcup$ over the subspaces i is defined in order to represent the aggregation from the low level subspaces i to the higher level I. The fusion operator carries out a weighted sum

$$z_I = \bigcup z_i = \frac{\sum_i \lambda_i * z_i}{\sum_i \lambda_i} \tag{5.3}$$

over any data z_i measured at the different low levels (which could, e.g., be $f_i(x)$ or y_i). The parameters $\lambda_i \in \mathcal{R}$ are indicators for the relevance or size of subspace i as part of I.

Let us assume we have given impact parameters λ_i. Then we get a high level representation of y following (5.3) with $y_I = \bigcup y_i$ (high level targets are aggregates of the low level targets). As the noise component at the low level is white noise, this component is also aggregated to noise at the high level as $\epsilon_{yI} = \bigcup \epsilon_{yi}$ which leads to a predictable relationship $f_I(x) = \bigcup f_i(x)$ fulfilling the high level relation similar to (2.1)

$$y_I = f_I(x) + \epsilon_{yI} = \bigcup y_i = \bigcup f_i(x) + \bigcup \epsilon_{yi}. \tag{5.4}$$

Let us now analyse the differences between $f_I(x)$ and $f_i(x)$ as these are very relevant if high level information is to be used for low level forecasting. We can expect that big differences between f_I and f_i would lead to big errors at the low level if we replace estimates of f_i by estimates of f_I. We define ϵ_{fi} as

$$\epsilon_{fi} = f_I(x) - f_i(x). \tag{5.5}$$

Combining (5.3),(5.4) and (5.5) it follows from

$$\begin{aligned} y_I &= \bigcup f_i(x) + \bigcup \epsilon_{yi} = \bigcup (f_I(x) - \epsilon_{fi}) + \bigcup \epsilon_{yi} \\ &= f_I(x) - \bigcup \epsilon_{fi} + \epsilon_{yI} \end{aligned} \tag{5.6}$$

that ϵ_{fi} has the nice characteristics of reducing to 0 if aggregated at the high level:

$$\bigcup \epsilon_{fi} = 0. \tag{5.7}$$

5.2.3 Predicting y_i

A predefined set of functions $h_k : \mathcal{R}^n \times \Phi \rightarrow \mathcal{R}$ is used in order to approximate the relationship between x and y_i. We assume function spaces $\mathcal{H}_k$ given as defined in Section 2.1.2. The index k represents different function spaces defined for diversification purposes as described in Section 4.2.2.

We also assume that a best estimation of parameters ϕ_i exists at each of the levels in order to approximate f_i by $h_k(;\phi_i)$

$$f_i(x) \approx h_k(x, \phi_i) \tag{5.8}$$

and that the underlying distance norm is linear in a manner that for any two functions $f_1(x) : \mathcal{R}^n \rightarrow \mathcal{R}$ and $f_2(x) : \mathcal{R}^n \rightarrow \mathcal{R}$ with $h(;\phi_1)$ representing the best approximation for $f_1(x)$ and $h(;\phi_2)$ the best approximation for $f_2(x)$, the best approximation for $\lambda_1 * f_1(x) + \lambda_2 * f_2(x)$ is given by $\lambda_1 * h(x, \phi_1) + \lambda_2 * h(x, \phi_2)$ for any $\lambda_1, \lambda_2 \in \mathcal{R}$.

Let us now assume that we estimate ϕ_i with $\widehat{\phi}_i$ with i representing here the level on which we have determined $\widehat{\phi}$. Let ${}^{\mathcal{H}_k I}e_i$ represent the out of sample error measured at level i which will be generated by predicting y_i based on the function space and level of learning as indicated in the left upper index. In the given example we have estimated y_i with $h_k(x, \widehat{\phi}_I)$ meaning that we have used function space $\mathcal{H}_k$ and determined $\widehat{\phi}$ at the high level I:

$${}^{\mathcal{H}_k I}e_i = y_i - {}^{\mathcal{H}_k I}\widehat{y}_i = y_i - h_k(x, \widehat{\phi}_I) = f_i(x) - h_k(x, \widehat{\phi}_I) + \epsilon_{yi} \tag{5.9}$$

In order to increase readability in the following, we will write the left upper index only if the correponding information is relevant. This means that we will

indicate the level of learning only if it differs from the level of measurement, so we mean ${}^{\mathcal{H}_k}e_i = {}^{\mathcal{H}_k i}\ e_i$ and ${}^{\mathcal{H}_k}e_I = {}^{\mathcal{H}_k I}\ e_I$. Low level aggregates to the high level are indicated with $\cup$, so ${}^{\mathcal{H}_k \cup}e_I$ means the error measured at the high level I (level of measurement always indicated as the right lower index) and achieved by use of function space $\mathcal{H}_k$ and aggregating low level forecasts to the high level.

Corresponding to 4.5 the total error variance term δ_{ei}^2 can be decomposed into

$$ {}^{\mathcal{H}_k I}\delta_{ei}^2 = {}^{\mathcal{H}_k}\ \delta_{hi}^2 + {}^{\mathcal{H}_k I}\ \delta_{\phi i}^2 + \delta_{yi}^2. \tag{5.10}$$

The right lower index again represents the error component as well as the level of error determination. The left upper index again provides the information about the forecast generation including the function space as well as the level of learning. As the bias component does not depend on the level of learning, this information is not provided for δ_{hi}^2. The Bayes component δ_{yi}^2 does not depend on the forecast generation at all as long as the input information does not change, so we do not have to provide information about the function space as well as the level of learning for this component.

5.2.4 Properties of the Error Components in Relation to Forecast Aggregation

Of course the main objective is to achieve good predictions at the level of forecasting, i.e. the low level, which means a minimisation of δ_{ei}^2. However, as in a lot of applications the generated forecasts are (also) used on an aggregated level, it is also worth to analyse the error ${}^{\cup}\delta_{eI}^2$. If we can find a good trade-off between the errors at different subspaces which generate more stable predictions meaning lower errors at the high level, this is certainly advantageous.

Different components of the error are related to different stability if they are aggregated. The stability depends on the correlation of an error component between different subspaces. If an error component is positively correlated between subspaces, we have to expect an error accumulation effect. If on the other hand

we have no or even a negative error correlation, these errors will compensate each other well.

The error variance component is a critical component for aggregation. The values y_i are often very noisy and the noise is often highly correlated between the different subspaces. Similar deviations in the target values of the training set contain the risk of generating highly correlated residuals ϵ_{yi}. It is therefore possible that the correlated residuals in the training set lead also to unstable (large) and highly positively correlated terms $\delta^2_{\phi i}$ and therefore to very big terms ${}^{\cup}\delta^2_{\phi I}$.

The situation is different for the bias term. Because of the linearity that we have assumed for the distance norm we also know that

$$\bigcup h(x, \phi_i) = h(x, \phi_I) \tag{5.11}$$

is true. It follows that

$$\bigcup \epsilon_{hi} = \epsilon_{hI} \tag{5.12}$$

because of $f_I(x) = \bigcup f_i(x)$, $\bigcup h(x, \phi_i) = h(x, \phi_I)$ and the definition of the bias term at both levels: $f_i(x) = h(x, \phi_i) + \epsilon_{hi}$ and $f_I(x) = h(x, \phi_I) + \epsilon_{hI}$. This means that all kinds of low level problems in case of more complex functions $f_i(x)$ at the low level compared to $f_I(x)$ compensate each other during the aggregation. If on the other hand $f_I(x)$ is more complex in comparison to the different subspaces $f_i(x)$, this means that we have correlations between the subspaces $f_i(x)$ which are not extremely big. In this case, we have only a few compensation effects of the error bias component during the aggregation, but probably lower bias values δ^2_{hi} because of the lower complexity of $f_i(x)$.

5.2.5 An Artificial Example

In O&D Revenue Management Systems [McGill 99] [Talluri 04][Weatherford 92] [Cross 97][Zaki 00][Pak 02][Neuling 04] seasonal predictions have to be carried out at a very fine level where the behaviour changes very quickly so that it is not

level	λ_i	$f_i(x)$	$\delta^2_{\epsilon yi}$
i_1	0.6	$sin((x-12)/(9))$	0.8
i_2	0.2	$-sin((x-12)/(9))$	2
i_3	0.2	$sin((x-12)/(9))$	2
I	-	$0.6 * sin((x-12)/(9))$	0.64

Tab. 11: Characteristics of the example data

possible to take a large number of historical data into account. As we have mentioned in Chapter 2, predictions have to be generated not only for different flights or origin-destination-itinerary pairs (the so called ODIs), but also separately for different fareclasses (F) representing different prices and booking restrictions as well as different point of sales (POS).

Let us assume we have to model a seasonal dependency of the booking behaviour on the calendar week in terms of seasonal factors $y^{season}_{cw\,i}$ at the low level i representing an ODO, DOW, F, POS combination as presented in Section 2.2.6.

As the "true relationship" $y^{season}_i = f(cw) + \epsilon_{yi}$ is not known, we introduce artificial ones in order to be able to illustrate certain behaviour of different error components. We use three subspaces i_1 to i_3 and assume seasonal dependencies $f_{i1}(x)$ to $f_{i3}(x)$ with x representing the calendar week as well as noise as described in Table 11. Figure 35 shows the assumed functions $f_i(x)$ at the different levels together with the noisy training target values assumed for two years of training data.

Two different methods of determining/learning the parameters are defined comparable to equation (2.14). They are both based on a function $h(x, \phi)$

$$h(x, \widehat{\phi}) = E(min(max(\frac{1}{2\phi_J + 1} \sum_{j=-\phi_J}^{\phi_J} [y_{cw+j}], \phi_{low}), \phi_{high})) \qquad (5.13)$$

as defined in (2.14) provided in Section 2.2.6. Because of restrictions to the possibly learned parameter sets they describe function spaces of varying complexity at the ODIFPOS level.

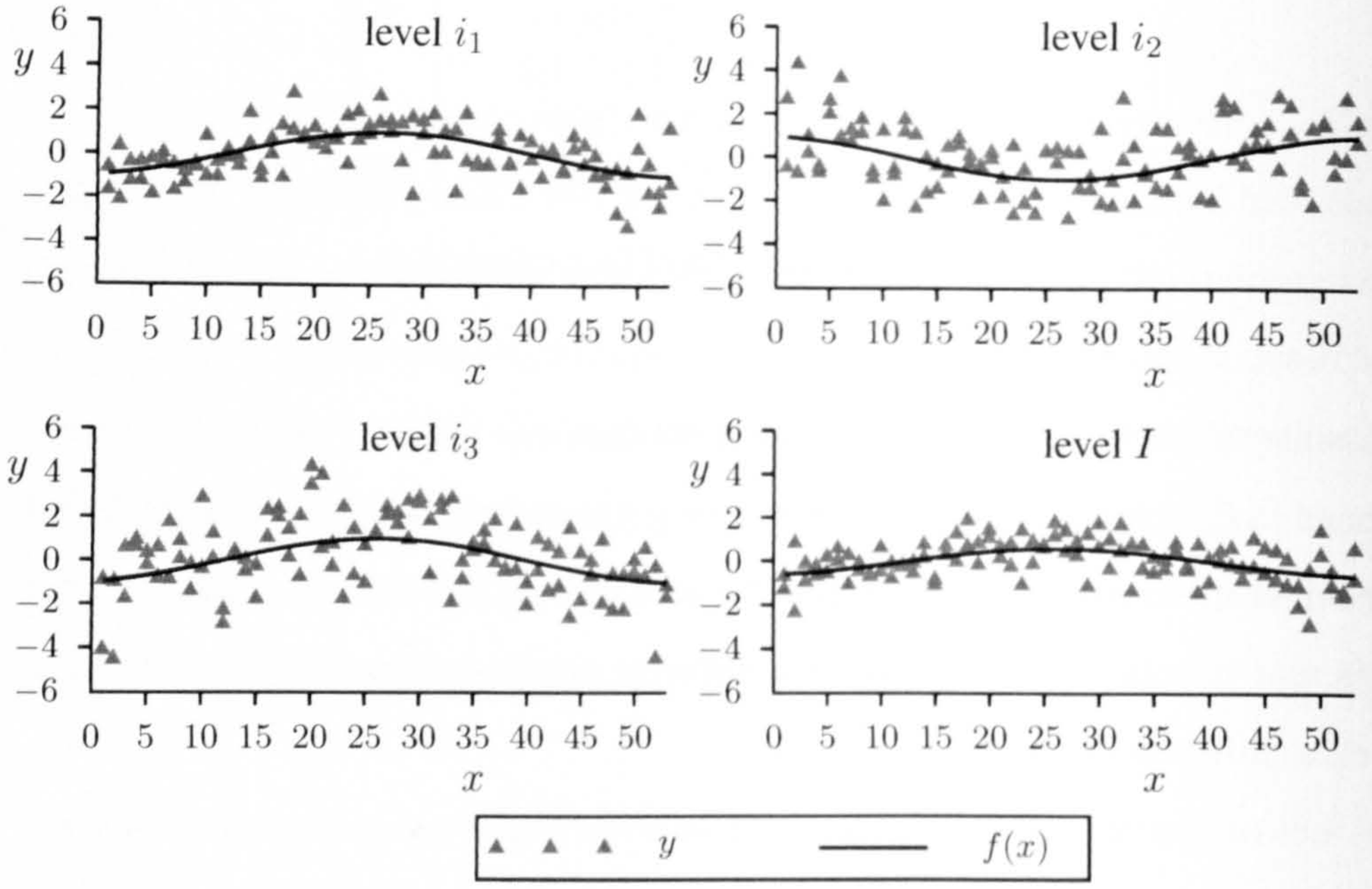

Fig. 35: Artificial Data generated for subspaces i_1 to i_3 and aggregated to the high level I.

The first learning approach generating $h_1(x, \phi)$ represents a very complex function space $\mathcal{H}_1$. Each seasonal factor is only restricted to the low limit of -1 which is determined by the application (a seasonal reduction of demand of more than 100% is not possible). We use parameters $\phi_{low} = -1$, $\phi_{high} = 4$ and $\phi_J = 0$. The seasonal factors are learned based on historical data by the best in sample estimation corresponding to a MSE error minimisation criterion which leads to a simple average of the data related to the corresponding calendar week

$$h_1(x, \widehat{\phi}) = E(min(max(y_{cw}, -1), 4)). \tag{5.14}$$

The second learning approach reduces the function space by two kinds of restrictions - limits to the generated seasonal factors as well as possible differences between neighboured seasonal factors obtained by smoothing the data. For the detection of each seasonal factor neighboured values are taken into account. Additionally, a lower and an upper limit of -0.5 and 0.6 for the expected seasonal deviation are used for stabilisation purposes in order to avoid, for instance, a zero season

level	h	$\delta^2_{\epsilon hi}$	$\delta^2_{\epsilon\phi i}$	${}^{\phi I}\delta^2_{\epsilon\phi i}$	${}^{comb}\delta^2_{\epsilon\phi i}$	$\delta^2_{\epsilon ei}$	${}^{\phi I}\delta^2_{\epsilon ei}$	${}^{comb}\delta^2_{\epsilon ei}$	w_i
i_1	1	0.00	0.45	0.33	0.33	1.25	1.13	1.13	0.42
i_2	1	0.00	0.88	1.59	0.72	2.88	3.59	2.72	0.64
i_3	1	0.00	1.13	0.33	0.33	3.13	2.33	2.33	0.23
I	1	0.00	0.28	0.28	0.28	0.92	0.92	0.92	-
i_1	2	0.06	0.04	0.05	0.04	0.90	0.91	0.90	0.54
i_2	2	0.06	0.05	1.09	0.09	2.11	3.15	2.15	0.91
i_3	2	0.06	0.07	0.05	0.05	2.13	2.11	2.11	0.46
I	2	0.02	0.02	0.01	0.01	0.68	0.66	0.66	-

Tab. 12: Error components of the forecast results

assumption in case of no historical bookings measured at the ODIFPOS level for a given calendar week.

$$h_2(x,\widehat{\phi}) = E(min(max(\frac{1}{5}\sum_{j=-2}^{2}[y_{cw+j}], -0.5), 0.6)). \tag{5.15}$$

The artificial example allows us to have a separate look at the different error components. Table 12 shows the results of different error components generated with learning method 1 and 2 as described in equations (5.14) and (5.15). The high level I contains the corresponding errors of the aggregated predictions. The bias, variance and total error of the pure low level predictions (and corresponding aggregates to the higher level) can be seen for the different subspaces and the two learning methods in columns 3, 4 and 7. It can be clearly seen that learning method 2 generates better results, even if it contains a bias component larger than zero. Learning method 1 is less stable and contains much larger parts in the variance component. We will discuss the other columns in later sections.

The bias component generated with learning method 2 can be seen in Figure 36 using the example of subspace i_1 together with the function $f_{i1}(x)$ and the prediction (with deviation from f because of bias plus variance error terms). The bias contains restrictions in the case of very strong seasonal effects because of the used limits of $[-0.5, 0.6]$ as well as minimal deviations because of the smoothing.

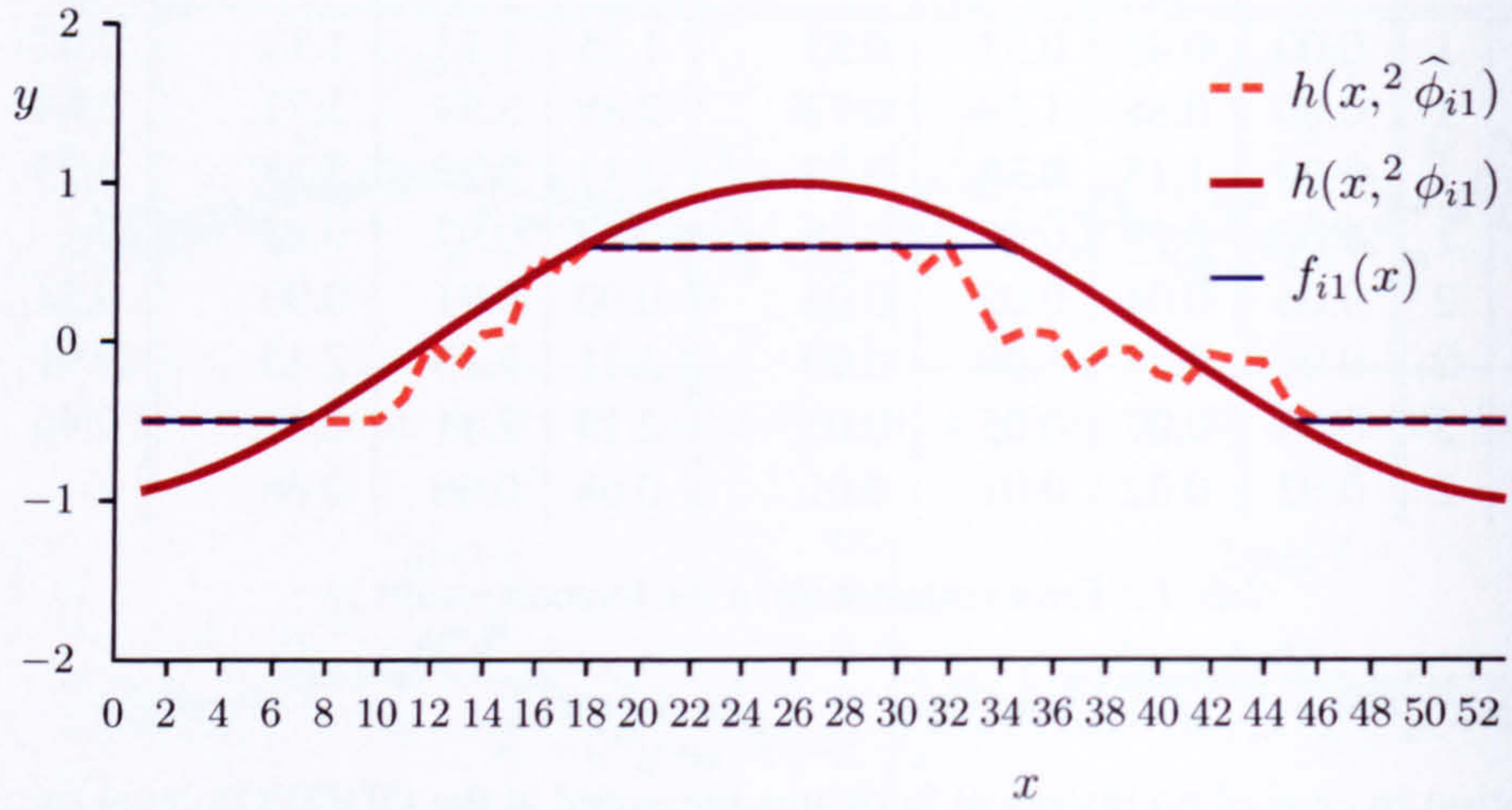

Fig. 36: Function $f_{i1}(x)$ together with the optimal and the generated prediction $h(x,^2\widehat{\phi_{i1}})$.

5.3 Alternative Options in Order to Incorporate Multi Level Information

5.3.1 Building one "Super Model"

The idea of this approach is to build one model which includes all available information, the "super model" (sm). We increase the search space and generate functions $h_{sm}(;)$ at the low level learned based on all information including higher level training data:

$$^{\mathcal{H}_{sm}}\widehat{y_i} = h_{sm}(x, \widehat{\phi}_{sm}) \tag{5.16}$$

with parameter vector ϕ_{sm} to estimate on the basis of training data at both levels.

Using the bias- variance decomposition (4.5) we get for this model

$$^{\mathcal{H}_{sm}}\delta^2_{ei} = ^{\mathcal{H}_{sm}}\delta^2_{hi} + ^{\mathcal{H}_{sm}}\delta^2_{\phi i} + \delta^2_{yi}. \tag{5.17}$$

This approach shows the clear advantage that all available information can be used in one model, which enables us to find the real relationships between the inputs. If our training data is stable enough to determine ϕ_{sm} well for appropriate functions $h(;)_{sm}$, this is the ideal choice. If we have increased the function space in $\mathcal{H}_{sm}$ compared to a function space $\mathcal{H}$ considering only a single level, we can expect

a lower error bias component ${}^{\mathcal{H}_{sm}}\delta^2_{hi} <^{\mathcal{H}} \delta^2_{hi}$. But with noisy training data we also risk a much higher error variance component ${}^{\mathcal{H}_{sm}}\delta^2_{\phi i} >^{\mathcal{H}} \delta^2_{\phi i}$ with a more complex model. This depends on how complex the different targets are connected in $h(;)_{sm}$ during the training process. If we have complex relations between the targets at the different levels in order to estimate a high dimensional parameter vector ϕ_{sm}, we risk high variance terms. If h_{sm} is less complex, e.g. a linear combination of less complex functions each depending only on y_i or y_I, we can achieve stable results, but results which also could have been achieved with a decomposed causal model followed by forecast combination.

Including higher level input information and using a higher dimensional vector ϕ_{sm} compared to $\phi_{\mathcal{H}}$ means a shift from the error bias component to the error variance component which is only beneficial if the noise at the low level is not so high that the parameters of the complex function can be estimated well enough so that we have

$$ {}^{\mathcal{H}}\delta^2_{hi} - {}^{\mathcal{H}_{sm}}\delta^2_{hi} >^{\mathcal{H}_{sm}} \delta^2_{\phi i} - {}^{\mathcal{H}}\delta^2_{\phi i}. \tag{5.18} $$

As instabilities in the estimation of ϕ_{sm} can lead to very large variance terms ${}^{\mathcal{H}_{sm}}\delta^2_{\phi i}$, following the argumentation in Section 5.2.4 we cannot even exclude that ${}^{\mathcal{H}_{sm}\cup}\delta^2_{eI} >^{\mathcal{H}I} \delta^2_{eI}$. This means that the low level error instabilities show a negative effect also at the high level and we achieve high level predictions which are worse compared to the predictions generated directly in I.

A similar argumentation can be used with a "super model" which includes the information of one or more neighboured subspaces.

5.3.2 Extending the History Pool

If the error term $\delta^2_{\phi i}$ is large, one available option is to increase the history pool for determining parameters ϕ also based on elements $(x_j, y_j)_T$ of the training set related to other parts of the input space $j \neq i$. As we have a bigger history pool, probably the estimated parameters would be more stable, so we have a reduced error variance component. But again we buy this improvement by a decrease of

adaptation to the special behaviour in i. As the function h is learned not only on training data of i, special features of $f_i(x)$ are contained only in parts of the data and will only be poorly modelled in h. In addition, special features contained in the relationship $f_j(x)$ in other subspaces are misleadingly modelled by h.

Nevertheless, for cases with only small differences between $f_i(x)$ and $f_I(x)$ or at least one or more other subspaces $f_j(x)$, small training sets $(x_i, y_i)_T$ and resulting big error variance components $\delta^2_{\phi i}$, the extension of the history pool might be beneficial.

5.3.3 Combining Forecasts Generated at Different Levels

In Chapter 3 and 4 it was already mentioned that combining techniques can be used in order to build complex functional approaches based on less complex ones in realising a reduction of the error bias component. It can also be used in order to decrease the error variance component by following a thick modelling approach related to the setting of certain parameter values or to preprocessing. A similar situation compared to these tasks can be expected related to the choice of the forecast level. Each forecast level contains information based on which functional relationships, ideal parameter settings, etc. can be determined, but it is likely that none of the models is optimal since it does not take into account all the available information. Low level forecasts potentially miss general structure information. High level forecasts do not take into account the special characteristics related to the concrete part of the input space, or the representation of these characteristics is contained in the forecast model in a completely different manner than having built the model directly on the finer level. That is why it makes sense to study the approach of forecast combination as an option in order to incorporate the knowledge at the different levels and to anaylse the effects on different error components at different levels.

5.4 Effects of Learning at Different Levels on the Error Components

We will now analyse effects of learning at the different levels on the error components. The analysis is not only focused on the low level results, we are also interested in generating high quality forecasts at the high level. This can be achieved by learning directly at the high level or by aggregating low level predictions.

5.4.1 Learning h at the Low Level

Corresponding to (4.5) the error achieved if we learn at the low level can be decomposed into

$$\delta_{ei}^2 = \delta_{hi}^2 + \delta_{\phi i}^2 + \delta_{yi}^2 \tag{5.19}$$

Let us now consider the aggregated pure low level predictions

$$^{\cup}\widehat{y}_I = \bigcup \widehat{y}_i = \bigcup h(x, \widehat{\phi}_i). \tag{5.20}$$

The aggregation leads to errors at the high level of

$$\begin{aligned} y_I -^{\cup} \widehat{y}_I &= y_I - \bigcup h(x, \widehat{\phi}_i) \\ &= y_I - \bigcup (y_i - \epsilon_{hi} - \epsilon_{\phi i} - \epsilon_{yi}) \\ &= y_I - (\bigcup y_i - \bigcup (\epsilon_{hi} + \epsilon_{\phi i}) - \bigcup \epsilon_{yi}) \\ &= \epsilon_{hI} + \bigcup \epsilon_{\phi i} + \epsilon_{yI}. \end{aligned} \tag{5.21}$$

As the bias-variance-bayes decomposition holds for the high level and we have already identified ϵ_{hI} as elements of the error bias component and ϵ_{yI} as the Bayes we know that the elements $\bigcup \epsilon_{\phi i}$ represent the error variance component and are so independent of the other parts of the error. We get total error variances

$$^{\cup}\delta_{eI}^2 = \delta_{hI}^2 +^{\cup} \delta_{\phi I}^2 + \delta_{yI}^2. \tag{5.22}$$

5.4.2 Learning h at the High Level

The alternative is to learn at the high level and to use the learned parameters for low level forecasts: ${}^{I}\widehat{y_i} = h(x, \widehat{\phi_I})$.

We will now analyse the composition of the resulting low level error. Combining (5.2), (5.5) and (4.5) we get

$$\begin{aligned} {}^{I}e_i &= y_i - {}^{I}\widehat{y_i} \\ &= f_i(x) + \epsilon_{yi} - h(x, \widehat{\phi_I}) \\ &= f_I(x) - \epsilon_{fi} + \epsilon_{yi} - h(x, \widehat{\phi_I}) \\ &= f_I(x) - \epsilon_{fi} + \epsilon_{yi} - (f_I(x) - \epsilon_{hI} - \epsilon_{\phi I}) \\ &= -\epsilon_{fi} + \epsilon_{hI} + \epsilon_{\phi I} + \epsilon_{yi}. \end{aligned} \tag{5.23}$$

We know that ϵ_{hI} and $\epsilon_{\phi I}$ are independent and that ϵ_{yi} is pure random noise. In this case we can represent the error as

$${}^{\phi I}\delta^2_{ei} = [\delta^2_{fi} + 2 * Cov(\epsilon_{hI}, \epsilon_{fi}) + 2 * Cov(\epsilon_{\phi I}, \epsilon_{fi})] + \delta^2_{hI} + \delta^2_{\phi I} + \delta^2_{yi}. \tag{5.24}$$

Let us now relate the above to the bias- variance- Bayes decomposition.

The series ϵ_{fi} can again be decomposed in relation to the best approximation $h(x; \phi_{\epsilon fi}) \in \mathcal{H}$:

$$\epsilon_{fi} = h(x; \phi_{\epsilon fi}) + \epsilon_{hfi} \tag{5.25}$$

and it follows that

$${}^{I}e_i = -h(x; \phi_{\epsilon fi}) - \epsilon_{hfi} + \epsilon_{hI} + \epsilon_{\phi I} + \epsilon_{yi}. \tag{5.26}$$

The elements ϵ_{hfi} and ϵ_{hI} belong to the bias term. Because of the linearity assumption of the approximation we know that

$$\delta^2_{hi} = \delta^2_{hfi} + \delta^2_{hI}. \tag{5.27}$$

We can therefore also represent the error as

$$^{I}\delta_{ei}^2 = \delta_{hi}^2 + [\delta_{\phi I}^2 + \delta_{hfi}^2 + 2 * Cov(\epsilon_{\phi I}, h(x; \phi_{\epsilon fi})] + \delta_{yi}^2 \qquad (5.28)$$

where δ_{hi}^2 belongs to the bias component, $^{I}\delta_{\phi i}^2 = \delta_{\phi I}^2 + \delta_{hfi}^2 + 2 * Cov(\epsilon_{\phi I}, h(x; \phi_{\epsilon fi})$ to the variance component and δ_{yi}^2 to the residuals.

We see that learning at the high level outperforms learning at the low level if

$$\delta_{\phi I}^2 + \delta_{hfi}^2 + 2 * Cov(\epsilon_{\phi I}, h(x; \phi_{\epsilon fi}) < \delta_{\phi i}^2. \qquad (5.29)$$

It strongly depends on the variance of ϵ_{fi} if this relation is true, we will discuss that in more detail for different cases in the next section. While in some cases clear tendencies can be detected, the question is what level to choose for learning if the error variances are about the same:

$$\delta_{\phi I}^2 + \delta_{hfi}^2 + 2 * Cov(\epsilon_{\phi I}, h(x; \phi_{\epsilon fi}) \approx \delta_{\phi i}^2. \qquad (5.30)$$

As this decision has no relevant impact on the measured low level forecast quality, the decision should be made in relation to the high level quality as well as stability assumptions in case of changing environments.

Because of

$$\bigcup {}^{I}\widehat{y_i} = \bigcup h(x, \widehat{\phi_I}) = h(x, \widehat{\phi_I}) \qquad (5.31)$$

we know that $\bigcup {}^{I}\widehat{y_i} = \widehat{y_I}$. We profit from equation (5.7) with

$$\bigcup(-h(x; \phi_{\epsilon fi}) - \epsilon_{hfi}) = 0. \qquad (5.32)$$

This indicates that in contrast to pure low level predictions we have an effect of error elimination of a part of the error variance component if the errors are aggregated to the high level. This can also have a stabilising effect in case of a changing environment when the situation does not change at the high level, i.e. shifts be-

tween the different subspaces. That is why we should always choose the higher level in these cases.

5.4.3 Using Forecast Combination

As we have already mentioned, the objective is to make choices concerning the level(s) for learning which manipulate the resulting errors concerning their correlation in a controlled manner. We have already seen that the choice of both levels for learning works well for some cases and not so well for others. The decision for one of the two approaches is difficult because the decision criterion should not depend on the pure error values at the low level. These do not take into account error variance correlations and stability effects in case of a changing environment. If we can manipulate the correlations of error variances in a manner that this is advantageous for the aggregation, this should be taken into account for the choice of the level. On the other hand, we want predictions at the fine level which do not only have a small error, but which also sufficiently and clearly show special characteristics (features) of a given subspace if this is possible. If the data is additionally very noisy, the errors can not be detected properly and, as the true function is not known, a decomposition of the error is not possible.

That is why an automated process is needed in order to make a qualified choice. Additionally it is advantageous to take not only one level into account, but to use the information present at both levels in order to generate good predictions. We need a flexible decision strategy in order to generate errors at the low level which are better or at least not much worse compared to the best choice of learning at the low or the high level, and at the same time to profit from similarities between the subspaces and levels in order to generate lower high level error variance terms. The decision process should be an automatic process which does not need to know details related to error decompositions.

5.4.4 Impacts of Forecast Combination on Low Level Forecasts

Using linear forecast combination on forecasts generated at the low and at the high level generates combined forecasts

$$\widehat{y_i} = w_i * h(x, \widehat{\phi_i}) + (1 - w_i) * h(x, \widehat{\phi_I}) \tag{5.33}$$

and errors

$$\begin{aligned}
^{comb}e_i &= w_i * h(x, \widehat{\phi_i}) + (1 - w_i) * h(x, \widehat{\phi_I}) - y_i \\
&= w_i * (h(x, \widehat{\phi_i}) - y_i) + (1 - w_i) * (h(x, \widehat{\phi_I}) - y_i) \\
&= w_i * (\epsilon_{hi} + \epsilon_{\phi i}) + (1 - w_i) * (-h(x, \phi_{\epsilon fi}) - \epsilon_{hfi} + \epsilon_{hI} + \epsilon_{\phi I}) + \epsilon_{yi} \\
&= \epsilon_{hi} + [w_i * \epsilon_{\phi i} + (1 - w_i) * (-h(x, \phi_{\epsilon fi}) + \epsilon_{\phi I})] + \epsilon_{yi}.
\end{aligned} \tag{5.34}$$

Under the assumption of independence this leads to

$$\begin{aligned}
^{comb}\delta^2_{ei} \sim\ & \delta^2_{hi} + w_i^2 * \delta^2_{\phi i} + (1 - w_i)^2 * \\
& (\delta^2_{\phi I} + \delta^2_{hfi} + 2 * Cov(\epsilon_{\phi I}, h(x; \phi_{\epsilon fi})) + \delta^2_{yi}.
\end{aligned} \tag{5.35}$$

More realistically, we have to expect covariances between the different error variance components. The difference between pure low level and pure high level forecasts is determined by the error variance component which can be approximated by

$$\begin{aligned}
^{comb}\delta^2_{\phi i} \sim\ & w_i^2 * \delta^2_{\phi i} + (1 - w_i)^2 * \\
& (\delta^2_{\phi I} + \delta^2_{hfi} + 2 * Cov(\epsilon_{\phi I}, h(x; \phi_{\epsilon fi})).
\end{aligned} \tag{5.36}$$

We will discuss what this means for different cases in Section 5.5. We will see that the weights are determined in a manner that for cases where the results generated at one level clearly outperform the other, the combination represents an automated choice of that level. For cases where both levels contain relevant

information, the fusion process can even outperform the quality achieved at both levels.

This can be seen for our artificial example by comparing columns 3,4 and 5 in Table 12. For subspaces i_1 and i_3 the error variance of the combined forecast is very close to the best single level results. For subspace i_3 we can even outperform the results achieved at the low and the high level.

5.4.5 Impacts of Forecast Combination to Aggregated Low Level Forecasts

Forecast combination can be beneficial in order to increase the forecast quality at the low level. But the potential is still bigger if the forecasts are aggregated to the higher level as we show now in comparing combined aggregates with pure low level aggregates.

If we look at the aggregate of the combined predictions we get

$$\begin{aligned} y_I - {}^{comb\cup}\widehat{y}_I &= y_I - \bigcup(w_i * h(x,\widehat{\phi}_i) + (1-w_i) * h(x,\widehat{\phi_I})) \\ &= y_I - \bigcup(\epsilon_{hi} + [w_i * \epsilon_{\phi i} + (1-w_i) * (-h(x,\phi_{\epsilon fi}) + \epsilon_{\phi I})] + \epsilon_{yi}) \\ &= \epsilon_{hI} + \bigcup[w_i * \epsilon_{\phi i} + (1-w_i) * (-h(x,\phi_{\epsilon fi}) + \epsilon_{\phi I})] + \epsilon_{yI} \\ &= \epsilon_{hI} + \bigcup[w_i * \epsilon_{\phi i}] + \bigcup[(1-w_i) * \epsilon_{\phi I}] \\ &\quad - \bigcup[(1-w_i) * h(x,\phi_{\epsilon fi})] + \epsilon_{yI}. \end{aligned} \tag{5.37}$$

We know that ϵ_{hI} represents with δ^2_{hI} the bias component, ϵ_{yI} is the Bayes, so it is clear that

$${}^{comb}\epsilon_{\phi\cup} = \bigcup[w_i * \epsilon_{\phi i}] + \bigcup[(1-w_i) * \epsilon_{\phi I}] - \bigcup[(1-w_i) * h(x,\phi_{\epsilon fi})] \tag{5.38}$$

represents the variance error component (with variance ${}^{comb\cup}\delta^2_{\phi I}$).

We can now write the error as

$${}^{comb\cup}\delta^2_{eI} = \delta^2_{hI} + {}^{comb\cup}\delta^2_{\phi I} + \delta^2_{yI}. \tag{5.39}$$

Comparing the resulting error with the aggregated pure low level errors given in equation (5.22) and the high level learning error at the high level we have to again compare only the error variance terms ${}^{comb\cup}\delta^2_{\phi I}$, ${}^{\cup}\delta^2_{\phi I}$ and $\delta^2_{\phi I}$.

Let us now have a look what happens to the different parts of equation (5.38) during the aggregation. Compensation effects depend on the correlation of the elements at the different subspaces.

The first part is an aggregate of the weighted low level variance term $\epsilon_{\phi i}$. As the low level parameter learning instabilities tend to be positively correlated, the component ${}^{\cup}\delta^2_{\phi I}$ can get very big and generate instabilities at the high level. This can only happen in the aggregation of the weighted elements if we have cases of large weights together with high terms $\epsilon_{\phi i}$. Compared to the pure low level forecast the forecast combination represents a reduction of this component which is especially important and positive if we have big terms $\epsilon_{\phi i}$.

The second part of equation (5.38) is an aggregate of weighted elements $\bigcup[(1-w_i) * \epsilon_{\phi I}]$. Because of

$$\bigcup[(1 - w_i) * \epsilon_{\phi I}] = \epsilon_{\phi I} * \bigcup(1 - w_i) \tag{5.40}$$

this part is stable and small in case of large weights (the interesting case containing potential stability problems) and small values of $\epsilon_{\phi I}$ in comparison to $\epsilon_{\phi i}$. In case of using only small weights, this means that we generate predictions which are similar to the pure high level predictions.

The third part $-\bigcup[(1-w_i)*h(x,\phi_{\epsilon fi})]$ is determined by the function $h(x,\phi_{\epsilon fi})$. Because $\bigcup h(x,\phi_{\epsilon fi}) = 0$ we can expect that the different elements of $h(x,\phi_{\epsilon fi})$ tend to be negatively correlated. It also follows

$$-\bigcup[(1 - w_i) * h(x, \phi_{\epsilon fi})] = \bigcup[w_i * h(x, \phi_{\epsilon fi})] \tag{5.41}$$

which means that we only achieve big values in cases where $h(x,\phi_{\epsilon fi})$ is relevant and w_i is large.

Comparing columns 3,4 and 5 in Table 12 for the high level aggregate I show these positive effects of the negative correlations for our artificial example. We can see that using forecast combination leads not only to better low level predictions, the aggregated combined predictions outperform the aggregated pure low level predictions and have the same quality as the forecasts generated directly at the high level.

We will now compare the effects of the different approaches in more detail for different cases in order to be able to make more specific statements about the expected forecast accuracy.

5.5 Discussion of Different Cases

5.5.1 Case1 (h is too complex to be learned properly even at the high level I)

In this case we will have a big variance term $\delta^2_{\phi I}$. The situation will probably be even worse at the low level. In any case the generated predictions will have a bad quality, but all of the other options discussed before will also have problems to reduce the error variance term. This case does not correspond to the general idea of including higher level information where the situation is more stable, we should use less complex functions h or include information generated at a higher level where the situation is more stable.

5.5.2 Case2 (h is not complex enough)

Geman et al [Geman 92] argue that if we have relevant bias problems, meaning high terms δ^2_{hi} and δ^2_{hI} in our predictions, it is not possible to solve these problems properly without including other functions in order to approximate f. Nevertheless, it can be that even with a very simple function h we get variance problems $\delta^2_{\phi i}$ if the training set in i is limited in sample size and characterised by high noise terms. If we get this problem, we can reduce at least this part of the forecast error with the forecast combination approach.

But if we also want to reduce the bias term we have no other choice than to increase the complexity in h, which is dangerous because of the potential variance problems or to include other functions $\tilde{h}$ which add additional information. If we also include predictions generated with $\tilde{h}$ into the combination process, we have a chance to generate more complex functions during the fusion process and so to reduce the bias term (see Section 4.2.2).

5.5.3 Case3 (i is representative for I)

This case means that the subspace i has a large impact λ_i in I. It follows that $\delta^2_{hi} \approx \delta^2_{hI}$, $\delta^2_{\phi i} \approx \delta^2_{\phi I}$ and δ^2_{fi} small in comparison to the other error components. The errors between the low and the high level forecasts are highly correlated and have a similar size so that we will probably achieve weights near 0.5.

In this situation the best approach would be to determine the model at the low level, but choosing the high level does not make a big difference. We will not achieve any improvements using forecast combination compared to pure low level or pure high level predictions, but we also do not have negative effects which we would have in following the approaches discussed in Sections 5.3.1 and 5.3.2.

This case is represented in our example at subspace i_1. Figure 37 shows clearly that the predictions generated by learning at the low and the high level are strongly correlated. We have achieved combination weights of 0.42 for learning method 1 and 0.54 for learning method 2. The error of the combined prediction is in both cases very close to the best choice.

5.5.4 Case4 (stable situation in i , but clear special characteristics in i)

In this case we can assume small components $\delta^2_{\phi i}$, $\delta^2_{\phi I}$ with δ^2_{fi} significant. Following the strategy of forecast combination we will get a large weight w_i because of the high error component δ^2_{fi} in the high level predictions (see (5.24)). This means that the fact, that the low level predictions should be used, can be represented by the weights very well. Also in this case it is not necessary to include

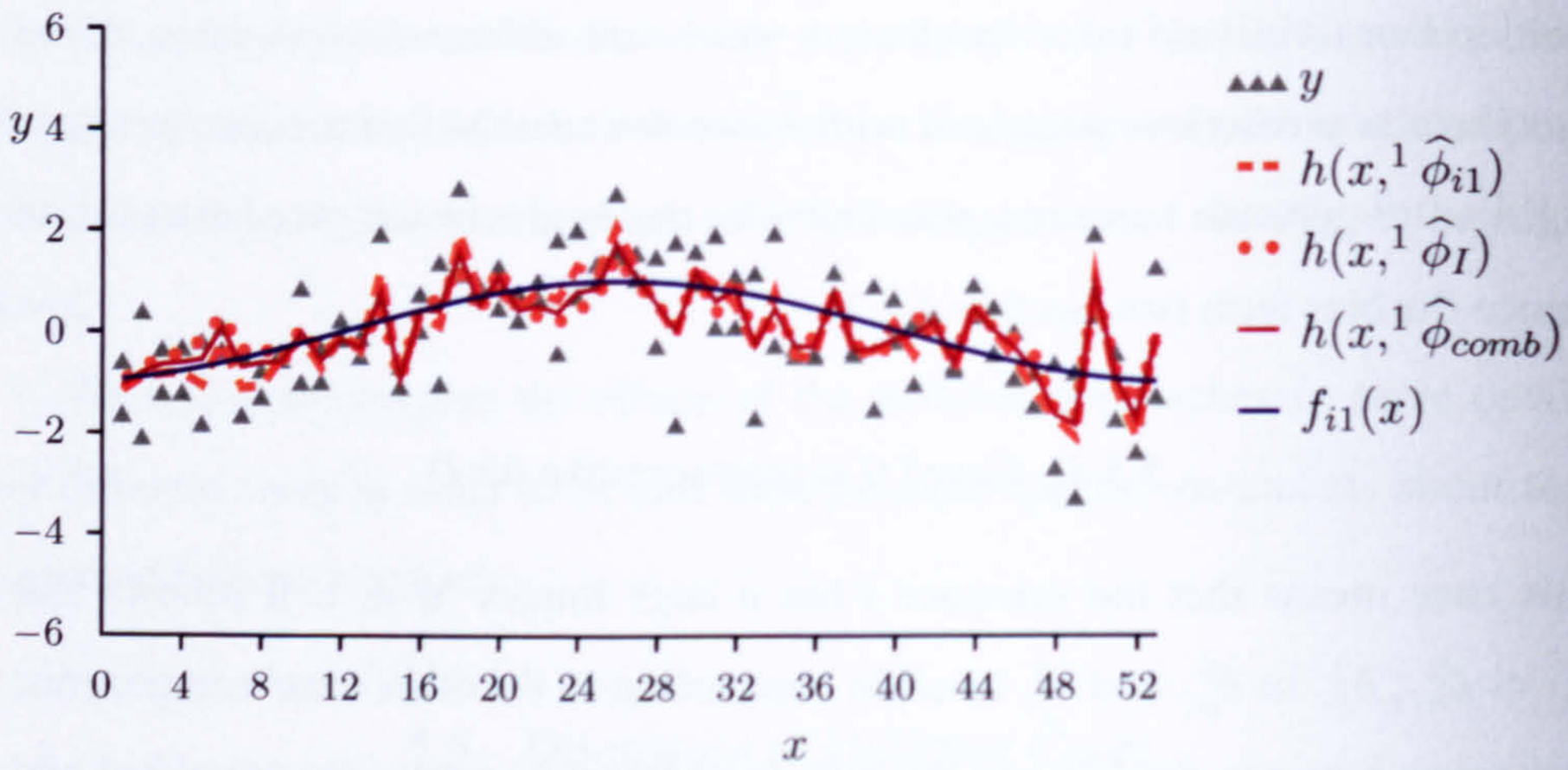

Fig. 37: Predictions for subspace i_1 generated with $h(x,^1\widehat{\phi}_{i1})$.

higher level knowledge, but taking into account the higher level predictions with a small weight can nevertheless have a stabilizing effect at the higher level. As $\delta^2_{\phi i}$ and $\delta^2_{\phi I}$ are small and the error variance term as described in (5.36) is therefore strongly influenced by δ^2_{hfi} we will have no problems during the aggregation (as argued in 5.4.5).

An example for this case exists in subspace i_2 of our example if learned with method 2. The low level forecast has been chosen with combination weight 0.91.

5.5.5 Case5 (h is too complex to be learned properly in i with δ^2_{fi} small)

In this case we have a very noisy training set with only few training data available in i. Learning only in i will lead us to overfitting and big variance terms $\delta^2_{\phi i}$. At the high level we have small values in all components of the error terms assuming that δ^2_{fi} is small.

In this case the high level predictions will provide good predictions. This can also be well represented by forecast combination weights. We will achieve a small weight w_i and therefore no instabilities during aggregation. Forecast combination

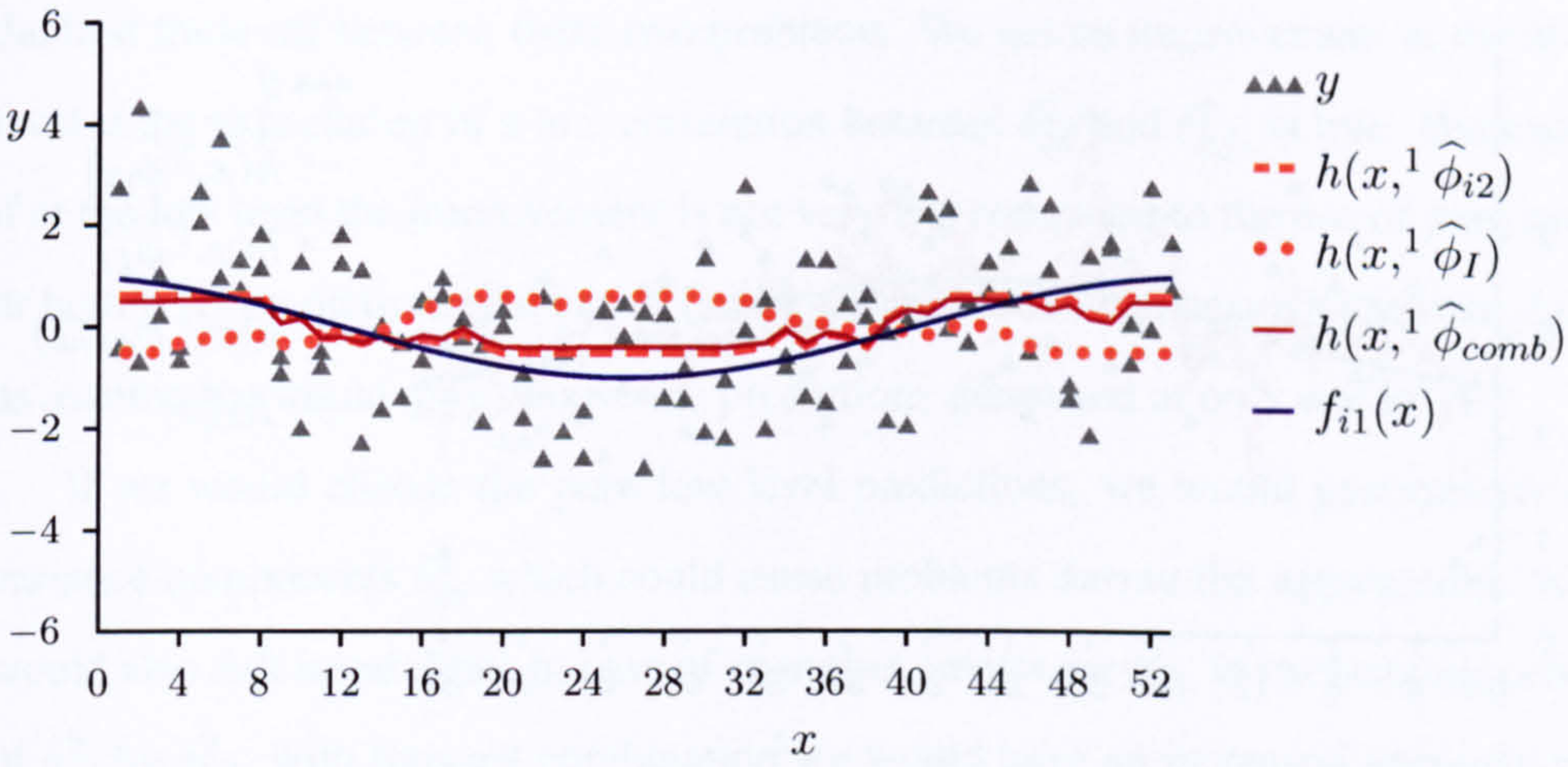

Fig. 38: Predictions for subspace i_2 generated with $h(x,^2\widehat{\phi}_{i2})$.

will not lead to improvements compared to the optimal choice of using only the high level predictions, but it can make this choice for us automatically. As the pure low level learning is unstable, building a super model as discussed in Section 5.3.1 would lead to unstable variance components as well. The extension of the history pool as described in 5.3.2 could be a solution, but would not have additional positive effects compared to pure high level predictions or the forecast combination approach.

This case is present in our example at subspace i_3. At this subspace the function $f_{i3}(x)$ is very close to $f_I(x)$. Figure 39 shows that the high level predictions outperform the low level predictions. This is reflected in the combination weights of 0.22 and 0.45. The combined results even outperform slightly the high level predictions.

The higher weight in the case of the second learning method is due to a large bias error term in comparison to the error variance term. This can be seen very well in Figure 40.

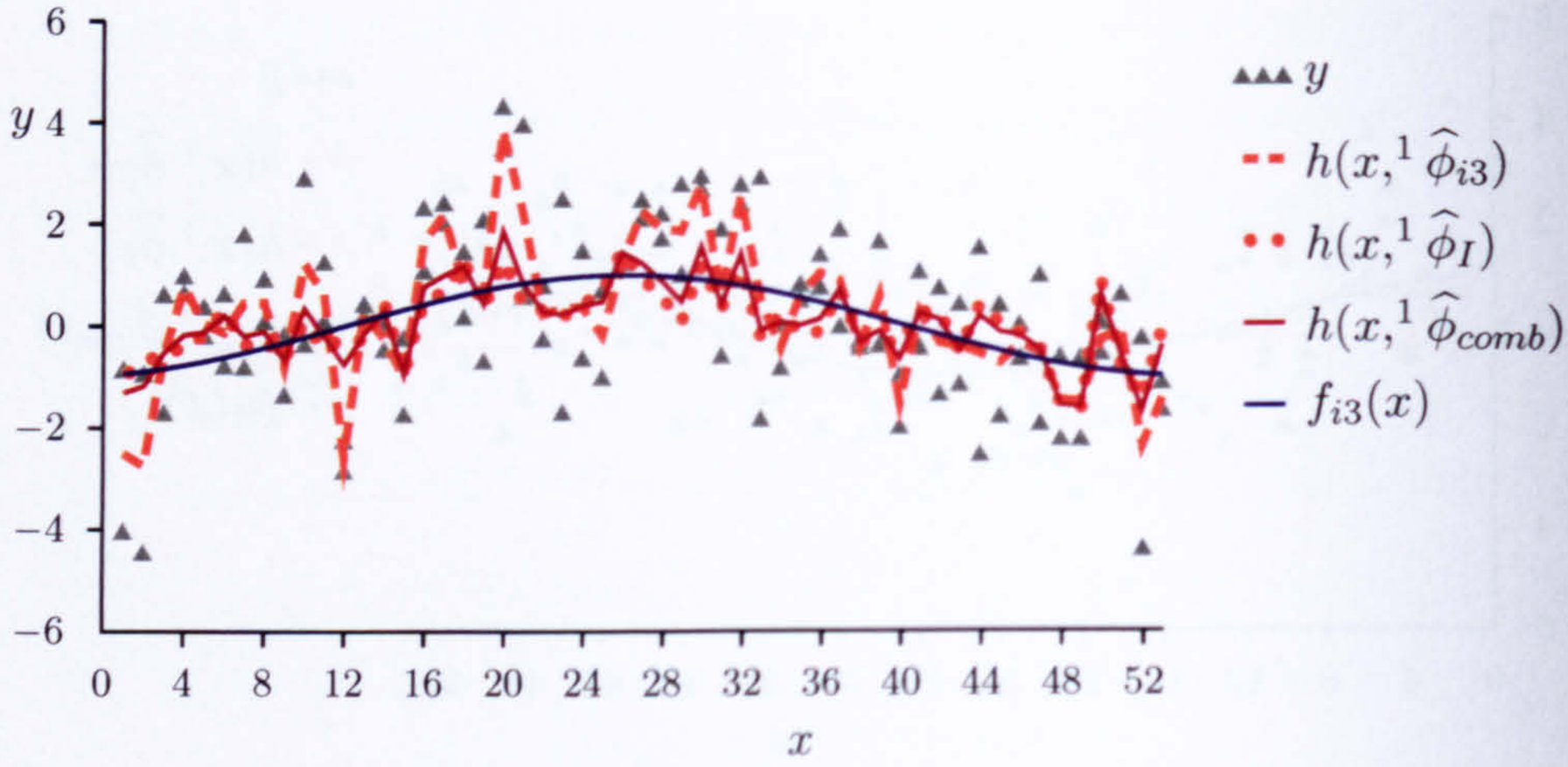

Fig. 39: Predictions for subspace i_3 generated with $h(x,^1\widehat{\phi}_{i3})$.

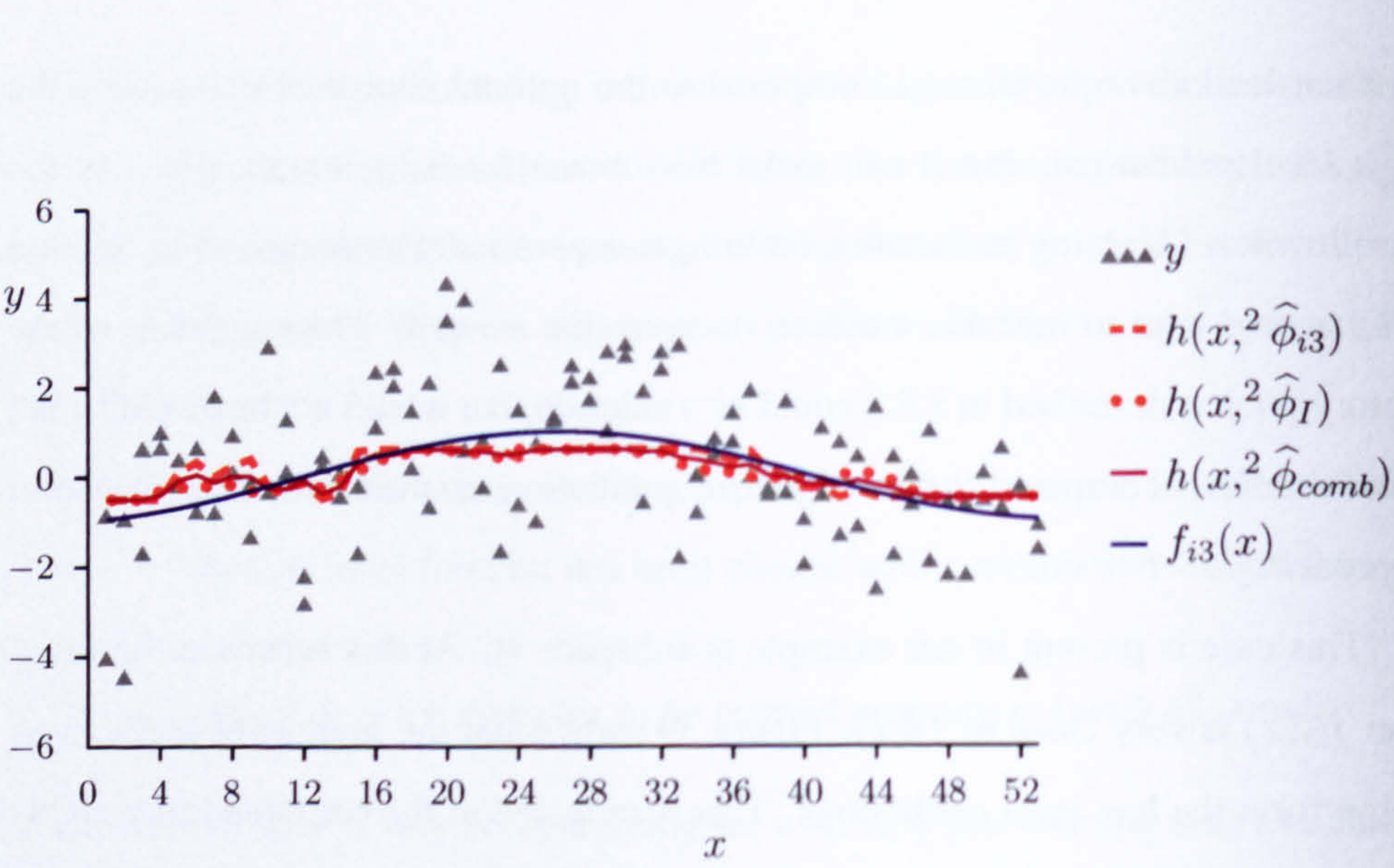

Fig. 40: Predictions for subspace i_3 generated with $h(x,^2\widehat{\phi}_{i3})$.

5.5.6 *Case6 (h is too complex to be learned properly in i with δ^2_{fi} relevant)*

This case represents the practically most relevant and also most interesting case. It means that $\delta^2_{\phi i}$ is big and we have also a big error term δ^2_{hfi}. Both predictions,

pure low level and pure high level predictions will not be very good, but there is a chance that the errors are not strongly correlated. Forecast combination finds for us the best trade-off between these two problems. We get an improvement at the low level if the expectation of a low correlation between $\delta^2_{\phi i}$ and δ^2_{hfi} is true. But even if at the low level the improvement is not very big compared to the use of pure low or high level predictions, the use of forecast combination can be advantageous. Let us assume we would only choose the predictions generated at only one level.

If we would choose the pure low level predictions, we would generate error variance components $\delta^2_{\phi i}$ which could cause problems during the aggregation. We would also risk instabilities in case of changing environments. In exchanging parts of $\delta^2_{\phi i}$ by δ^2_{hfi} with forecast combination we would have an increased aggregation stability (which we discussed in Section 5.4.5) as well as a higher stability if the situation changes.

If on the other hand we would choose the high level predictions, we would generate predictions which do not represent the special characteristics in the subspace i at all which is not good for analysts or other systems which work with the generated predictions.

The situation in subspace i_2 learned with method 1 in our example represents that case. The differences in the error variance term of the low and the high level learning can be clearly seen in Figure 41. While the function learned at the low level has very high random deviations from the true function based on the noisy target data, the function learned at the high level is much smoother but has a completely different trend. It can also be seen that the combined forecast represents a good trade-off between the two which on one hand has reduced noise and on the other hand approximates better the low level function $f_{i2}(x)$ than the function learned at the high level.

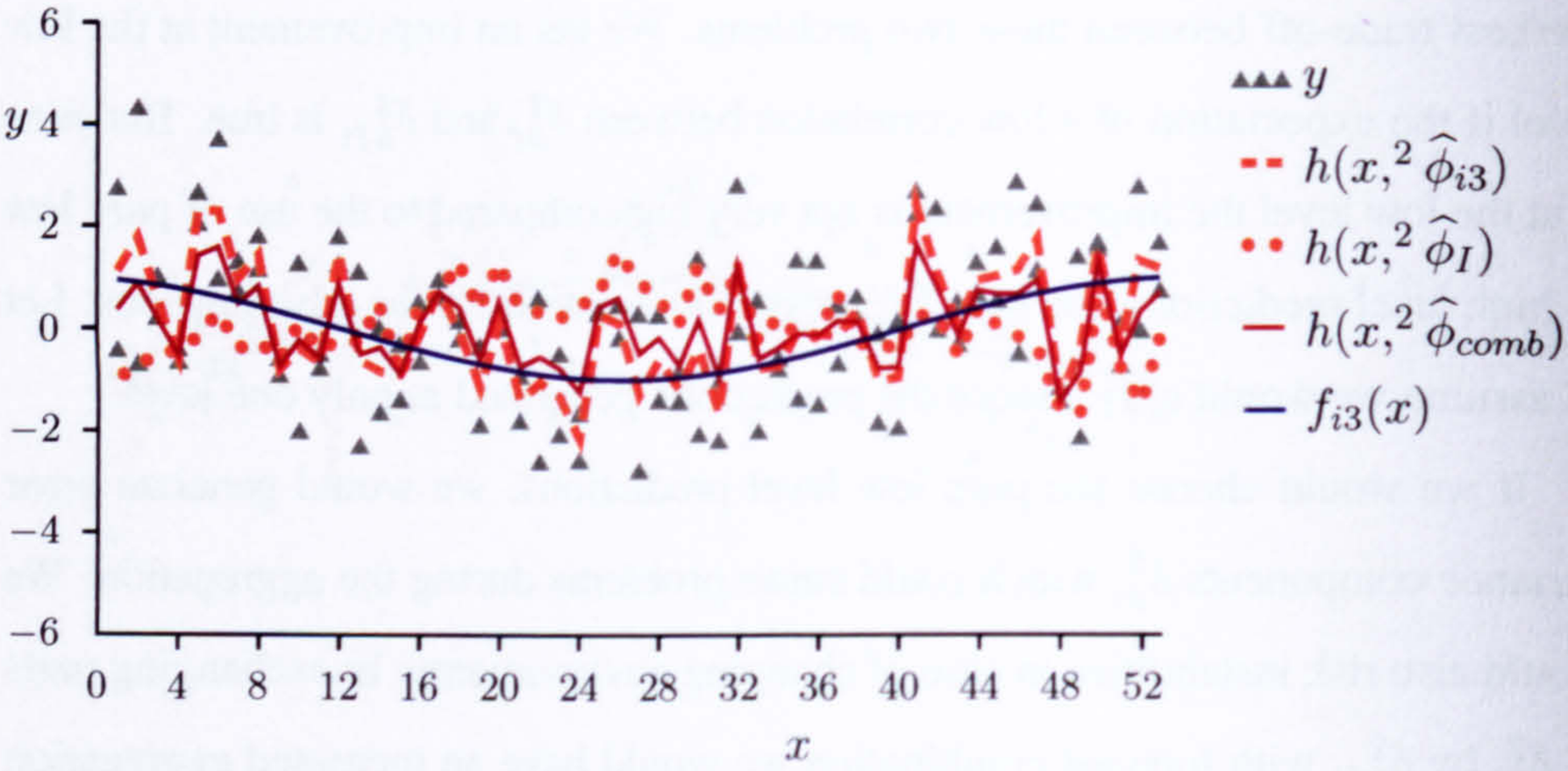

Fig. 41: Predictions for subspace i_2 generated with $h(x,^1\widehat{\phi}_{i2})$.

5.6 Summary

As it could be seen from the previous subsections using the approach of generating multi level forecasts and combining them seems advantageous in comparison to using pure low or high level forecasts or by following the strategy of the "super model" or extending the history pool. In most cases we will achieve an improved result at the low level. In cases where the low level forecast quality can only be slightly improved as compared to the best chosen individual low or high level forecast evaluated at the low level, the forecast combination process represents an automatic decision which level to choose. Additionally, in many cases we can also reach a modification in the correlation between error variance terms in a manner that the aggregate of low level forecasts gets a higher quality, which is especially important in systems where forecasts are generally aggregated in order to support decision making processes. It can be seen by analysing the different parts in equation (5.38).

We have already argued that the first component is an unstable component with

elements which tend to be positively correlated. We have also mentioned that in the aggregation of the weighted elements instabilities can only happen if we have cases of large weights together with high terms $\epsilon_{\phi i}$. The discussion of the different cases showed that this situation does not occur. In all cases where the elements $\epsilon_{\phi i}$ are big in comparison to $\epsilon_{\phi I}$ we do not get large combination weights w_i. We have shown that the only cases where we do not get a small weight are the cases 3 and 6 with weights around 0.5.

While the second part is stable in any case, the third part can again contain big values in cases where $h(x, \phi_{\epsilon f i})$ is relevant and w_i is large. Again we have only the cases 3 and 6 where this can happen.

We have seen that in case 3 it simply does not matter which level to choose because the low and high level are comparable and highly correlated. In case 6 we have high elements $\epsilon_{\phi i}$ as well as big terms $h(x, \phi_{\epsilon f i})$. The replacement of parts of the pure low level forecast error variance terms $\delta^2_{\phi \cup}$ into $\bigcup[w_i * h(x, \phi_{\epsilon f i})]$ is advantageous because of the negative correlation of the elements in $\bigcup[w_i * h(x, \phi_{\epsilon f i})]$.

Summarising we can say that in all cases where $h(;)$ is appropriate at the low or the high level (cases 3 to 6) forecast combination will generate very good results at the low as well as at the high level in comparison to pure low or high level predictions. In cases 4 and 5 forecast combination represents an automated choice of the right level. In case 6 we can even expect that the combined forecast outperforms the pure low or high level forecasts assuming the objective of generating good predictions for both levels.

The most problematic cases are the cases 1 and 2 where $h(;)$ is structurally too poor or too complex for both of the levels. We propose to follow the approach of "thick modelling" and the approach of using different function spaces as described in chapter 4 in addition to multi level combination, because these approaches offer additional benefits and enable us to find stable solutions in these problematic cases.

5.7 Experiments

5.7.1 Description of Experiments

Experiments have been carried out in order to analyse the effects of multi level learning on the prediction of the seasonal component of our application.

Table 13 shows an example for data diversified concerning three types of diversification: the level of learning, the used function space and thick modelling concerning the parameter of smoothing seasonal factors.

m	function space $\mathcal{H}$(see section 2.2.6)	level of learning	parameters $\phi_{low}, \phi_{high}, \phi_J$
${}^{0}\widehat{y}$	$h_1^{season}(x,\phi)$ (historical model 2.15)	ODO DOW F POS	$\phi_{low}=0, \phi_{high}=0$
${}^{1}\widehat{y}$	$h_1^{season}(x,\phi)$ (historical model 2.15)	ODO DOW F POS	$\phi_{low}=-0.33, \phi_{high}=1$
${}^{2}\widehat{y}$	$h_1^{season}(x,\phi)$ (historical model 2.15)	ODO DOW F POS	$\phi_{low}=-0.66, \phi_{high}=2$
${}^{3}\widehat{y}$	$h_1^{season}(x,\phi)$ (historical model 2.15)	ODO DOW F POS	$\phi_{low}=-1, \phi_{high}=3$
${}^{4}\widehat{y}$	$h_2^{season}(x,\phi)$ (additive model 2.17)	ODO DOW F POS	$\phi_{low}=0, \phi_{high}=0$
${}^{5}\widehat{y}$	$h_3^{season}(x,\phi)$ (multiplicative model 2.18)	ODO DOW F POS	$\phi_{low}=-0.33, \phi_{high}=1$
${}^{6}\widehat{y}$	$h_3^{season}(x,\phi)$ (multiplicative model 2.18)	ODO DOW F POS	$\phi_{low}=-0.66, \phi_{high}=2$
${}^{7}\widehat{y}$	$h_3^{season}(x,\phi)$ (multiplicative model 2.18)	ODO DOW F POS	$\phi_{low}=-1, \phi_{high}=3$
${}^{8}\widehat{y}$	$h_1^{season}(x,\phi)$ (historical model 2.15)	ODO DOW COMP POS	$\phi_{low}=0, \phi_{high}=0$
${}^{9}\widehat{y}$	$h_1^{season}(x,\phi)$ (historical model 2.15)	ODO DOW COMP POS	$\phi_{low}=-0.33, \phi_{high}=1$
${}^{10}\widehat{y}$	$h_1^{season}(x,\phi)$ (historical model 2.15)	ODO DOW COMP POS	$\phi_{low}=-0.66, \phi_{high}=2$
${}^{11}\widehat{y}$	$h_1^{season}(x,\phi)$ (historical model 2.15)	ODO DOW COMP POS	$\phi_{low}=-1, \phi_{high}=3$
${}^{12}\widehat{y}$	$h_2^{season}(x,\phi)$ (additive model 2.17)	ODO DOW COMP POS	$\phi_{low}=0, \phi_{high}=0$
${}^{13}\widehat{y}$	$h_3^{season}(x,\phi)$ (multiplicative model 2.18)	ODO DOW COMP POS	$\phi_{low}=-0.33, \phi_{high}=1$
${}^{14}\widehat{y}$	$h_3^{season}(x,\phi)$ (multiplicative model 2.18)	ODO DOW COMP POS	$\phi_{low}=-0.66, \phi_{high}=2$
${}^{15}\widehat{y}$	$h_3^{season}(x,\phi)$ (multiplicative model 2.18)	ODO DOW COMP POS	$\phi_{low}=-1, \phi_{high}=3$
${}^{16}\widehat{y}$	$h_1^{season}(x,\phi)$ (historical model 2.15)	ODO F POS	$\phi_{low}=0, \phi_{high}=0$
${}^{17}\widehat{y}$	$h_1^{season}(x,\phi)$ (historical model 2.15)	ODO F POS	$\phi_{low}=-0.33, \phi_{high}=1$
${}^{18}\widehat{y}$	$h_1^{season}(x,\phi)$ (historical model 2.15)	ODO F POS	$\phi_{low}=-0.66, \phi_{high}=2$
${}^{19}\widehat{y}$	$h_1^{season}(x,\phi)$ (historical model 2.15)	ODO F POS	$\phi_{low}=-1, \phi_{high}=3$
${}^{20}\widehat{y}$	$h_2^{season}(x,\phi)$ (additive model 2.17)	ODO F POS	$\phi_{low}=0, \phi_{high}=0$
${}^{21}\widehat{y}$	$h_3^{season}(x,\phi)$ (multiplicative model 2.18)	ODO F POS	$\phi_{low}=-0.33, \phi_{high}=1$
${}^{22}\widehat{y}$	$h_3^{season}(x,\phi)$ (multiplicative model 2.18)	ODO F POS	$\phi_{low}=-0.66, \phi_{high}=2$
${}^{23}\widehat{y}$	$h_3^{season}(x,\phi)$ (multiplicative model 2.18)	ODO F POS	$\phi_{low}=-1, \phi_{high}=3$
${}^{24}\widehat{y}$	$h_1^{season}(x,\phi)$ (historical model 2.15)	ODO COMP POS	$\phi_{low}=0, \phi_{high}=0$
${}^{25}\widehat{y}$	$h_1^{season}(x,\phi)$ (historical model 2.15)	ODO COMP POS	$\phi_{low}=-0.33, \phi_{high}=1$
${}^{26}\widehat{y}$	$h_1^{season}(x,\phi)$ (historical model 2.15)	ODO COMP POS	$\phi_{low}=-0.66, \phi_{high}=2$
${}^{27}\widehat{y}$	$h_1^{season}(x,\phi)$ (historical model 2.15)	ODO COMP POS	$\phi_{low}=-1, \phi_{high}=3$
${}^{28}\widehat{y}$	$h_2^{season}(x,\phi)$ (additive model 2.17)	ODO COMP POS	$\phi_{low}=0, \phi_{high}=0$
${}^{29}\widehat{y}$	$h_3^{season}(x,\phi)$ (multiplicative model 2.18)	ODO COMP POS	$\phi_{low}=-0.33, \phi_{high}=1$
${}^{30}\widehat{y}$	$h_3^{season}(x,\phi)$ (multiplicative model 2.18)	ODO COMP POS	$\phi_{low}=-0.66, \phi_{high}=2$
${}^{31}\widehat{y}$	$h_3^{season}(x,\phi)$ (multiplicative model 2.18)	ODO COMP POS	$\phi_{low}=-1, \phi_{high}=3$

Tab. 13: Set of forecasts diversified concerning the function space, level of learning and parameters used for thick modelling.

The diversification of the function space and parameter values has been applied as described for experiment 4 (see Section 4.6.1). This means that we have used diversified function spaces $h_1^{season}(x, \phi)$ and $h_3^{season}(x, \phi)$ as well as diversified parameter values ϕ_{low} and ϕ_{high} applied for the calculation of seasonal factors. The initial parameters $\phi_{low} = -1$, and $\phi_{high} = 3$ have been dumped with the same factors between 0 and 1 as carried out in experiment 4. The application of factor 0 in model $h_3^{season}(x, \phi)$ again leads to model $h_2^{season}(x, \phi)$ so that this model is also included into the fusion process. Additionally, the calculation of seasonal factors has been diversified concerning the level of learning. The determination of historical seasonal factors y_{cw} (2.14) has been diversified as well as the calculation of current seasonal behaviour $y_{t_{d,\tau}}$ (2.18). The level diversification is reached with a data decomposition carried out at the different levels mentioned in Table 13. The decomposed data is then applied in the learning and forecasting process using the diversified methods and parameter values. Note that in order to always generate forecasts adapted to the current booking behaviour it is not possible to diversify the values of $y_{t_{d,\tau}}^{unc}$, $\widehat{y}_{t_d}^{attr}$, $\widehat{y}_{t_{d,\tau}}^{attr}$ and $\widehat{y}_{t_d}^{attr}$ used in models $h_1^{season}(x, \phi)$ (2.15) and $h_3^{season}(x, \phi)$(2.18). The different levels of learning are only related to the applied seasonal factors, not to the values and forecasts of the current behaviour and the expected future attractiveness. The calculations have been carried out for one well performing linear model (F^{var}) as well as for one nonlinear model (F^{appr}). We have compared: a) the results achieved with a restricted set generated with trimming to the best performing 10 input forecasts with b) the results achieved with a restricted set generated with trimming to the best performing 5 input forecasts in each combination.

Details related to the experimental setup can be found in the appendix describing experiment 5 (B.6.5).

5.7.2 Experimental Results

Table 14 shows the errors of the forecasts containing combined seasonal predictions as relative improvement in relation to the best individual forecast ${}^0\widehat{y}$ at the low level of forecasting (ODO F POS) and the high level (ODO). A graphical representation of the absolute total errors at the high level is shown in Figure 42.

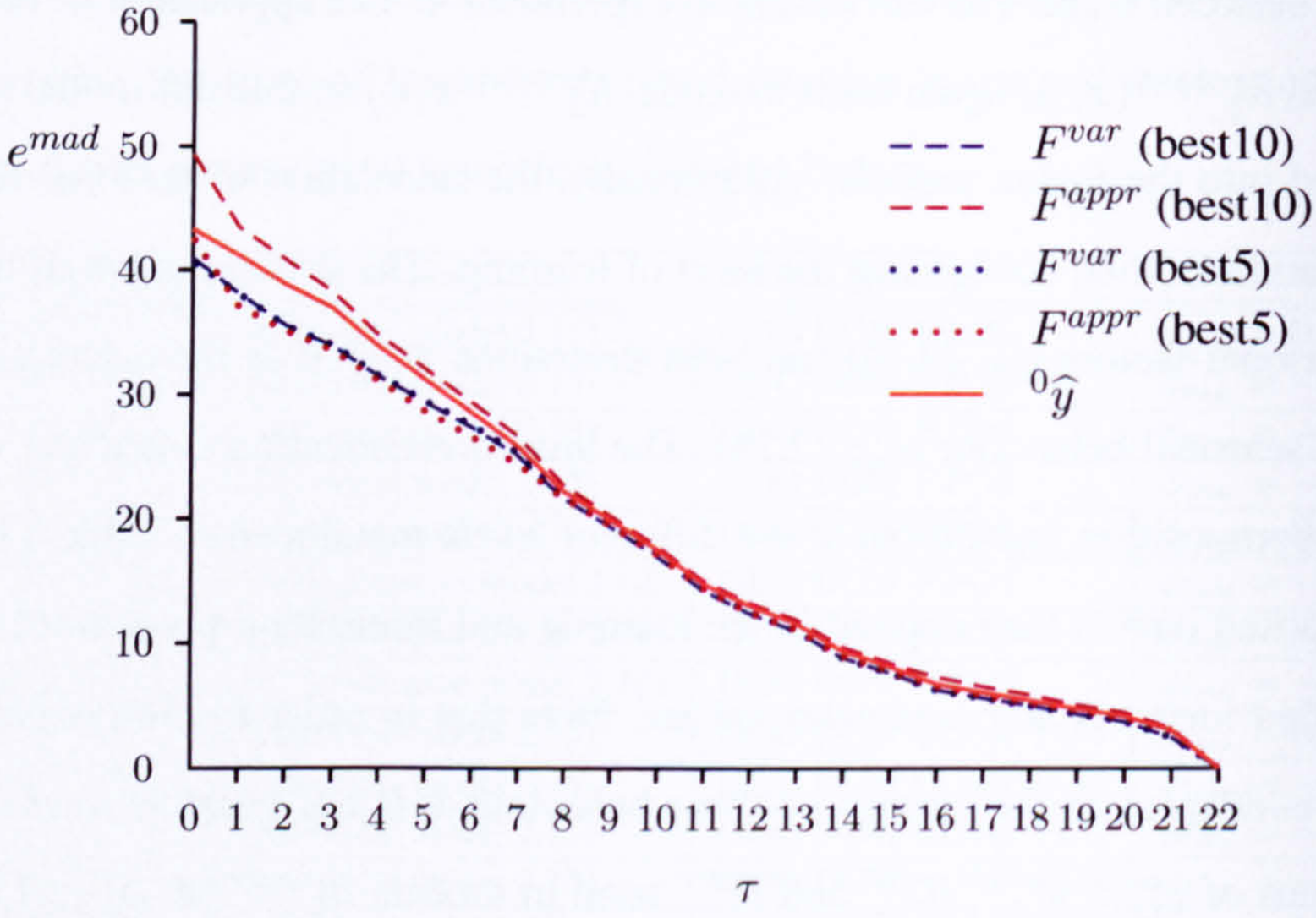

Fig. 42: Absolute errors (mad) achieved using forecast combination of diversified seasonal predictions in comparison to the best individual forecast ${}^0\widehat{y}$ at the high level (ODO).

The results show that the application of multi level forecast combination is a promising approach. We could achieve an improvement up to 3% at the low level and even up to 8% at the high level. The results achieved at the high level support the theoretical findings indicating that we generate an effect of elimination of lowly correlated errors if aggregating the combined predictions to the high level. Unfortunately, a more detailed analysis of the results has shown that the nonlinear models, with which we achieve the largest improvements at the high level, can also show unstable behaviour in single cases. This means that they do not represent a very secure alternative. The models are very sensible and insecure if applied on a larger number of forecasts. They also need significantly increased calculation time.

Figure 43 shows an example of error covariances generated for multi level

forecasts. It can be seen that now we find parts of the covariance matrix indicating a low correlation between the forecasts.

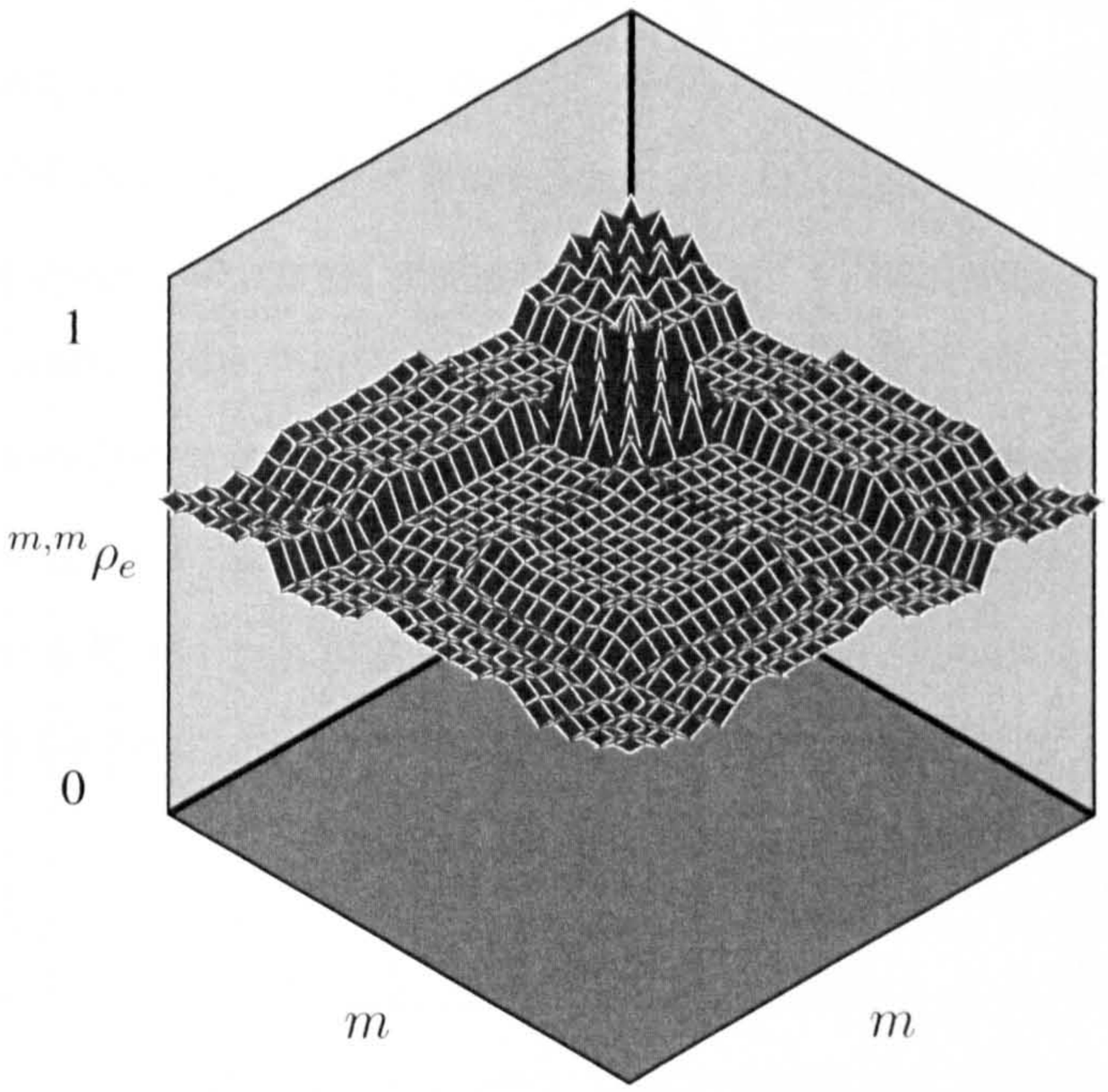

Fig. 43: Example of covariances achieved with multi level diversification.

Unfortunately, the combination process cannot optimally profit from these parts of the covariance matrix because of the unstable behaviour of the more sophisticated models. The results also clearly indicate that the combination performs better on a subset of input predictions. Especially the more complex nonlinear model gets unstable for a larger set of input predictions. The best results are achieved with the simple model which does not consider covariance information and which therefore does not profit from the unique information contained in some of the input forecasts in an optimal manner. The instabilities of the other models are caused by the large number of input forecasts generated with multi level diversification.

The results motivate a closer look at the effects of combining a large number of forecasts in the context of multi level diversification. We have seen that a large number of forecasts to be combined seems to be not optimal for the more sophis-

ticated combination models. On the other hand, we know that we loose relevant information by not using the forecasts with significantly different covariance values in a more explicit manner.

Figure 43 also shows that we can easily identify parts of the covariance matrix belonging to the different types of diversification that we have applied. We can very well distinguish parts affected by the different chosen parameter values as well as breaks representing the application of forecasts generated with different models and learned at different levels. This behaviour is very typical if different types of diversifications are used in order to generate input predictions for a combination process. In the next chapter we will therefore lead a discussion of how we can handle the problem of a large number of input forecasts and how we can profit from the special structure of the error covariance matrix.

τ	F_i^{var}(best10)	F_i^{appr}(best10)	F_i^{var}(best5)	F_i^{appr}(best5)
0	0.03	-0.22	0.04	0.02
1	0.03	-0.15	0.03	0.02
2	0.03	-0.12	0.03	0.03
3	0.03	-0.10	0.03	0.03
4	0.03	-0.08	0.03	0.04
5	0.03	-0.01	0.03	0.03
6	0.03	0.00	0.03	0.02
7	0.02	-0.02	0.02	0.02
8	0.02	-0.02	0.02	0.02
9	0.02	-0.02	0.02	0.02
10	0.02	-0.03	0.01	0.02
11	0.02	-0.03	0.01	0.03
12	0.02	-0.03	0.01	0.03
13	0.02	-0.03	0.02	0.02
14	0.03	-0.03	0.03	0.03
15	0.03	-0.02	0.03	0.02
16	0.04	-0.02	0.04	0.04
17	0.04	-0.03	0.05	0.05
18	0.05	-0.03	0.05	0.05
19	0.06	-0.09	0.06	0.07
20	0.07	-0.06	0.07	0.11
21	0.12	-0.05	0.12	0.20
22	0.00	0.00	0.00	0.00

τ	F_I^{var}(best10)	F_I^{appr}(best10)	F_I^{var}(best5)	F_I^{appr}(best5)
0	0.05	-0.17	0.06	-0.01
1	0.08	-0.08	0.07	0.07
2	0.08	-0.07	0.07	0.08
3	0.08	-0.05	0.06	0.08
4	0.07	-0.04	0.05	0.07
5	0.06	-0.04	0.03	0.07
6	0.04	-0.04	0.01	0.06
7	0.03	-0.04	0.00	0.05
8	0.03	-0.03	-0.01	0.02
9	0.03	-0.02	0.00	0.02
10	0.03	-0.02	-0.01	0.02
11	0.04	-0.02	0.00	0.02
12	0.05	-0.02	0.02	0.04
13	0.05	-0.03	0.03	0.04
14	0.06	-0.04	0.05	0.04
15	0.06	-0.04	0.05	0.03
16	0.07	-0.08	0.07	0.03
17	0.07	-0.08	0.07	0.03
18	0.08	-0.08	0.08	0.04
19	0.09	-0.05	0.10	0.04
20	0.11	-0.06	0.12	0.05
21	0.20	-0.03	0.20	0.02
22	0.00	0.00	0.00	0.00

Tab. 14: Relative improvement using forecast combination of diversified multi level predictions in comparison to the best individual forecast ${}^0\widehat{y}$.

6. POOLING FOR COMBINATION OF MULTI LEVEL FORECASTS

6.1 Reasons for Pooling

The advantages and disadvantages of different linear combination models have been extensively discussed in Chapter 3 and 4. The fact that the optimal model often performs worse than the simple average in practical applications has initiated a long discussion (see Section 4.4). Bunn [Bunn 85] explained the effects by high estimation errors of the weights based on the fact that the covariance matrix of the forecast errors is not exactly known. He showed that the estimation error increases in cases of short time series, time-varying forecast errors or other instabilities. Other aspects influencing the expected error reduction by forecast combination are the number of combined forecasts, the general level of different error components as well as the level and distribution of error variances and the correlation among different input forecasts. Too short time series, time-varying forecast errors or other instabilities can result in inaccurate estimations and changes in error variances and especially in error covariances. On the other hand, using the purely error variance based models or even the simple average result in suboptimal weights because they do not take the correlations between the forecast errors into account.

We are looking for an approach that generates similar results to the ideal model but does not need to calculate covariance information. The approach of pooling represents an interesting option in order to achieve that goal. In order to motivate this, we will shortly repeat and summarise three aspects of influences on combination efficiency that are relevant for pooling.

6.1.1 Combination influenced by the number of forecasts to combine

An increased number of forecasts to combine can lead to increased weight estimation errors.

This topic has been discussed in Section 4.5.1. We have seen that it can easily happen that models are too complex and generate implausible predictions which lay outside of the expected range of the target variable. The inclusion of forecasts that add only marginal information should be dropped in order to avoid increased parameter estimation errors. Instead of combining all forecasts, it is therefore often advantageous to discard the models with the worst performance (trimming).

6.1.2 Combination influenced by the level of total error variances

If forecast error terms are smaller, the optimal weights can be estimated more accurately.

A proper determination of the weights can be difficult for big values δ_ϵ^2 especially if the error variance component δ_ϕ^2 is big compared to the error bias component. Random impacts in the training data will influence the determination of the weights more than differences in the bias component. High impact of the chosen training set on the determined weights means unstable combination weights. Details related to this topic have been discussed in Section 4.4 and different Sections of Chapter 5.

6.1.3 Combination influenced by homogeneity of error variances and error correlation

Not only the general level of errors but the relation of error variance components and correlation among different forecast models is also very relevant in two important aspects.

Homogeneous covariances can lead to high estimation errors if the optimal model is used.

The first aspect is that small differences in covariances increase the risk of high estimation errors. Errors in the expected covariance matrix can have a bigger impact on the matrix inversion. This has been shown by Bunn in [Bunn 85] and it has already been discussed in Section 4.4.1.

The expected loss in combined forecast quality when using a more stable combination model (i.e. simple average) in comparison to the optimal model strongly depends on the homogeneity in the distribution and correlation of the error components

The second aspect is related to the fact that depending on the range of error variances and correlations among all forecasts the use of simpler and more stable combination models and avoiding the estimation of the whole covariance matrix can lead to different levels of loss in the combined forecast. Similar values in error variances and correlation among all forecasts lead to low losses if a simpler model is used. A motivation and discussion of these dependencies can be found in Section 4.4.2.

6.1.4 Why pooling ?

Based on the three previous subsections it can be said that in an effective combination one should use

1. a limited number of combined forecasts containing diverse information, but also containing
2. low total error variance terms and
3. homogenous error variance and correlation values in order to be able to avoid high weight estimation errors by using a simpler linear combination model without a high expected loss compared to the optimal model.

Unfortunately, in our case of combining multi level forecasts none of these criteria would normally be fulfilled. If we use more than one diversification criterion the number of generated forecasts is large. Large noise terms and small numbers

lead to high total error variance terms. And we will show in the next sections that if we use different diversification approaches we cannot expect homogeneous covariances.

The Idea of Pooling

The approach of combination by pooling realises a combination task related to a given set of input forecasts $F : (\{\widehat{y}\}) \rightarrow^{comb} \widehat{y}$ by splitting it into different subtasks ${}^{g}F : (c_g = \{\widehat{y}\}_g \subset \{\widehat{y}\}) \rightarrow^{g} \widehat{y}$ followed by a combination $\widetilde{F} : (\{{}^{g}\widehat{y}\}) \rightarrow^{comb} \widehat{y}$ that carries out the final combination. The sets c_g of input forecasts of the subtasks are called forecast pools or forecast clusters.

The Advantage

Ideally the subtasks ${}^{g}F$ each contain some of the advantageous characteristics mentioned above. For example let us assume that we have a clustering mechanism that groups the forecasts in relation to criterion 3 into clusters of a limited number of forecasts. The lower number of forecasts to be combined in each subtask leads to a potential decrease of the weight estimation errors in each combination because of criterion 1. In the first step we have the additional advantage of criterion 3 and can therefore use a more stable combination model (like F^{av} or F^{var}) for the combination. In the second combination we profit from criterion 2, as a first combination step has already been carried out. In many cases we can also expect lower differences in error variances and covariances after the first combination.

It is possible that the final combination $\widetilde{F}$ is again a combination using the approach of pooling. Following this idea we can generate complex multi step combination structures. Similar structures have been the subject of studies of Ruta and Gabrys [Ruta 05] in the context of classifier combination approaches.

6.2 Error variance based pooling

The difficulty of pooling is related to the question of how to generate the pools. As we have difficulties to properly estimate covariance information, the clustering can not be performed directly on the covariance matrix without taking these difficulties into account.

6.2.1 The pooling approaches of Aiolfi and Timmermann

Aiolfi and Timmermann [Aiolfi 04] studied different approaches of clustering connected with different combination models and trimming. They used quantiles and k-means clustering based on past forecast performance in order to find the optimal number of clusters and the optimal separation points between the forecast sets.

We refer here to the algorithm which they called CEW in [Aiolfi 04]. It generates a combined forecast ${}^{comb}\widehat{y}$ based on a set of input forecasts $\{\widehat{y}\}$ with an algorithm that can be summarised as follows:

Algorithm 1: $F^{cew}(\{\widehat{y}\}) \rightarrow^{comb} \widehat{y}$

1. order $\{\widehat{y}\} \rightarrow \{\widehat{y}_r\}$ depending on the ranks of forecast performance meaning the total error variances ${}^{r}\delta_e^2$
2. determine G clusters c_g,$g \in [0 \ldots G-1]$ by k-means clustering based on ${}^{r}\delta_e^2$
3. remove the last cluster containing the worst forecasts (trimming)
4. for each cluster $c_g, g = 0 \ldots G-2$ run a linear combination F^{av} in order to achieve ${}^{g}\widehat{y} = F^{av}(c_g)$
5. combine the results of the clusters to achieve the combined forecast ${}^{comb}\widehat{y} = F^{var}(\{{}^{g}\widehat{y}\})$ or ${}^{comb}\widehat{y} = F^{opt}(\{{}^{g}\widehat{y}\})$ after having potentially applied an additional trimming of $(\{{}^{g}\widehat{y}\})$

All approaches analysed by Aiolfi and Timmermann run a clustering which is purely based on information about error variance terms. Correlation information is interpreted as inaccurate and not taken into account at all.

nbr	level	ϕ_α	δ_h^2	δ_ϕ^2	δ_y^2	δ_e^2
00	i	0.0	0.25	0.85	0.3	1.4
01	i	0.025	0.25	0.81	0.3	1.36
02	i	0.05	0.25	0.74	0.3	1.29
03	i	0.075	0.24	0.67	0.3	1.21
04	i	0.1	0.22	0.66	0.3	1.18
05	i	0.125	0.23	0.73	0.3	1.26
06	i	0.15	0.25	0.74	0.3	1.29
07	i	0.175	0.25	0.76	0.3	1.31
08	i	0.2	0.25	0.83	0.3	1.38
09	I	0.0	0.23	0.67	0.3	1.20
10	I	0.025	0.22	0.67	0.3	1.19
11	I	0.05	0.20	0.68	0.3	1.18
12	I	0.075	0.20	0.54	0.3	1.04
13	I	0.1	0.20	0.54	0.3	1.04
14	I	0.125	0.20	0.58	0.3	1.08
15	I	0.15	0.20	0.68	0.3	1.18
16	I	0.175	0.20	0.69	0.3	1.19
17	I	0.2	0.24	0.66	0.3	1.20

Tab. 15: Example for a set of multi level forecasts generated over two levels i and I and with different values related to the parameter ϕ_α. The example gives in the first column a number, in the second and third column the level and parameter information. The following three columns represent the error bias component, error variance component, error Bayes component and the total error variance.

6.2.2 Example

For the illustration purposes, in this example we consider thick modelling related to one parameter and two levels of learning for the other parameters. The parameter ϕ_α which is handled using thick modelling is related to the error bias component as described in Section 4.3.2. If we learn the parameters ϕ, certain values of parameter ϕ_α generate forecasts with a higher error compared to other values (because of instabilities in the error variance term). Table 15 and Figure 44 show the error components and the total error depending on ϕ_α and the level of learning. The best results are achieved by learning at the high level for parameter values between 0.075 and 0.125.

The covariances are shown in Table 16. The different level in the numbers related to the separated parts can be clearly seen. The example shows the typical behaviour of our studied cases: we have a large set of input forecasts which

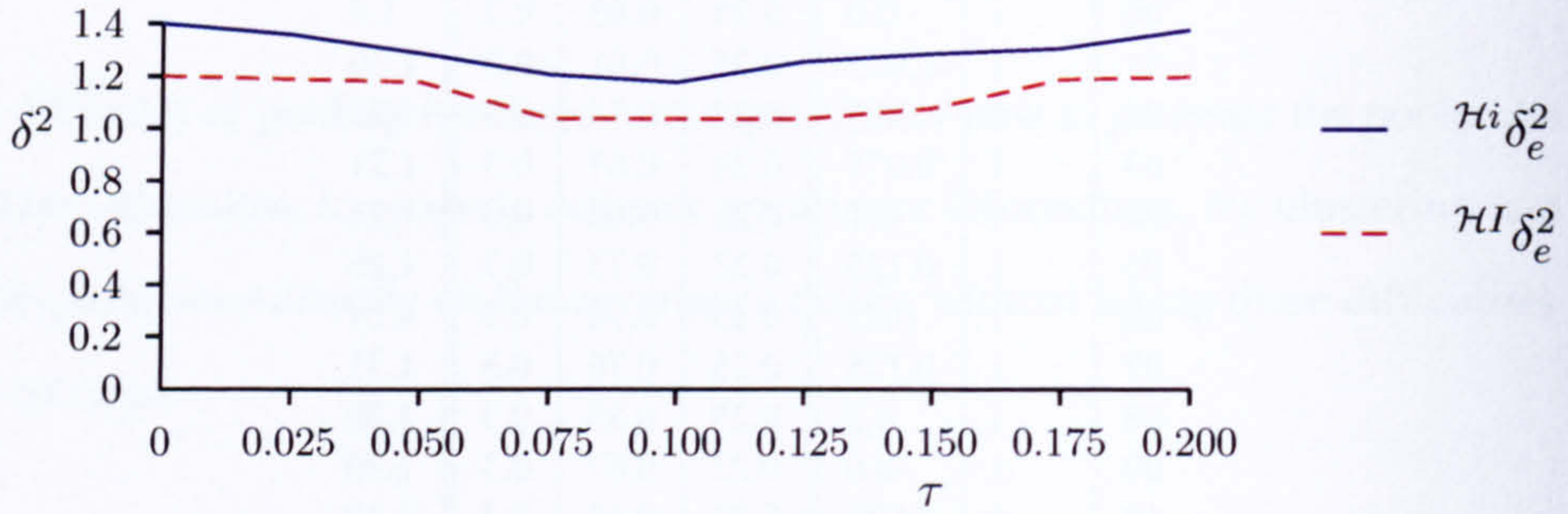

Fig. 44: Graphical representation of the errors given in the example shown in Table 15.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	**1.4**	1.1	1.09	1.09	1.08	1.07	1.09	1.1	1.1	0.72	0.56	0.54	0.53	0.53	0.55	0.56	0.56	0.56
1	1.1	**1.36**	1.06	1.05	1.03	1.03	1.05	1.07	1.08	0.58	0.7	0.6	0.58	0.56	0.55	0.56	0.57	0.57
2	1.09	1.06	**1.29**	1.08	1.06	1.05	1.05	1.08	1.08	0.54	0.57	0.68	0.57	0.56	0.55	0.55	0.55	0.54
3	1.09	1.05	1.08	**1.21**	1.03	1.03	1.02	1	0.98	0.54	0.55	0.55	0.66	0.55	0.55	0.53	0.55	0.57
4	1.08	1.03	1.06	1.03	**1.18**	1.06	1.06	1.06	1.13	0.57	0.56	0.53	0.55	0.71	0.55	0.52	0.54	0.55
5	1.07	1.03	1.05	1.02	1.06	**1.26**	1.14	1.08	1.1	0.53	0.55	0.55	0.56	0.58	0.72	0.57	0.54	0.54
6	1.09	1.05	1.05	1	1.06	1.14	**1.29**	0.15	1.16	0.52	0.53	0.53	0.53	0.55	0.57	0.69	0.55	0.53
7	1.1	1.07	1.08	0.98	1.06	1.08	0.15	**1.31**	1.3	0.53	0.53	0.54	0.54	0.56	0.58	0.6	0.69	0.56
8	1.1	1.08	1.08	1.01	1.13	1.1	1.16	1.3	**1.38**	0.5	0.52	0.52	0.54	0.55	0.55	0.58	0.6	0.67
9	0.72	0.58	0.54	0.54	0.57	0.53	0.52	0.53	0.5	**1.2**	1.09	1.09	0.89	0.87	0.86	1.07	1.07	1.07
10	0.56	0.7	0.57	0.55	0.56	0.55	0.53	0.53	0.52	1.09	**1.19**	1.08	0.9	0.91	0.88	1.08	1.07	1.07
11	0.54	0.6	0.68	0.55	0.53	0.55	0.53	0.54	0.52	1.09	1.08	**1.18**	0.91	0.92	0.9	1.09	1.08	1.07
12	0.53	0.58	0.57	0.66	0.55	0.56	0.53	0.54	0.54	0.89	0.9	0.91	**1.04**	0.92	0.95	0.92	0.9	0.9
13	0.53	0.56	0.56	0.55	0.71	0.58	0.55	0.56	0.55	0.87	0.91	0.92	0.92	**1.04**	0.94	0.93	0.9	0.9
14	0.55	0.55	0.55	0.55	0.55	0.72	0.57	0.58	0.55	0.86	0.88	0.9	0.95	0.94	**1.08**	0.93	0.91	0.9
15	0.56	0.56	0.55	0.53	0.52	0.57	0.69	0.6	0.58	1.07	1.08	1.09	0.92	0.93	0.93	**1.18**	1.08	1.07
16	0.56	0.57	0.55	0.55	0.54	0.54	0.55	0.69	0.6	1.07	1.07	1.08	0.9	0.9	0.91	1.08	**1.19**	1.08
17	0.56	0.57	0.54	0.57	0.55	0.54	0.53	0.56	0.67	1.07	1.07	1.07	0.9	0.9	0.9	1.07	1.08	**1.2**

Tab. 16: Covariance matrix of our example

are characterised by diverse total error variances and inhomogeneous covariances depending on how the forecasts have been generated.

We will now see why using a flat combination on such a set of input forecasts is risky and discuss the approach of pooling as a promising alternative. The example will later be used again in order to illustrate advantages and weaknesses of different pooling methods in relation to our forecasting problems.

The ranks and clusters for each prediction of our example related to algorithm 1 are shown in columns "r" and "g" of Table 45. We achieve a first cluster containing all high level predictions with most stable settings of the parameter ϕ_α. A second cluster contains other high level forecasts as well as two low level forecasts. The worst predictions are removed by trimming. Figure 46 shows the final combination

structure.

nbr	level	ϕ_α	δ_e^2	r	g
00	i	0.0	1.4	17	-
01	i	0.025	1.36	15	-
02	i	0.05	1.29	12	-
03	i	0.075	1.21	10	1
04	i	0.1	1.18	3	1
05	i	0.125	1.26	11	-
06	i	0.15	1.29	13	-
07	i	0.175	1.31	14	-
08	i	0.2	1.38	16	-
09	I	0.0	1.20	9	1
10	I	0.025	1.19	6	1
11	I	0.05	1.18	4	1
12	I	0.075	1.04	0	0
13	I	0.1	1.04	1	0
14	I	0.125	1.08	2	0
15	I	0.15	1.18	5	1
16	I	0.175	1.19	7	1
17	I	0.2	1.20	8	1

Fig. 45: Ranks and clusters for the example.

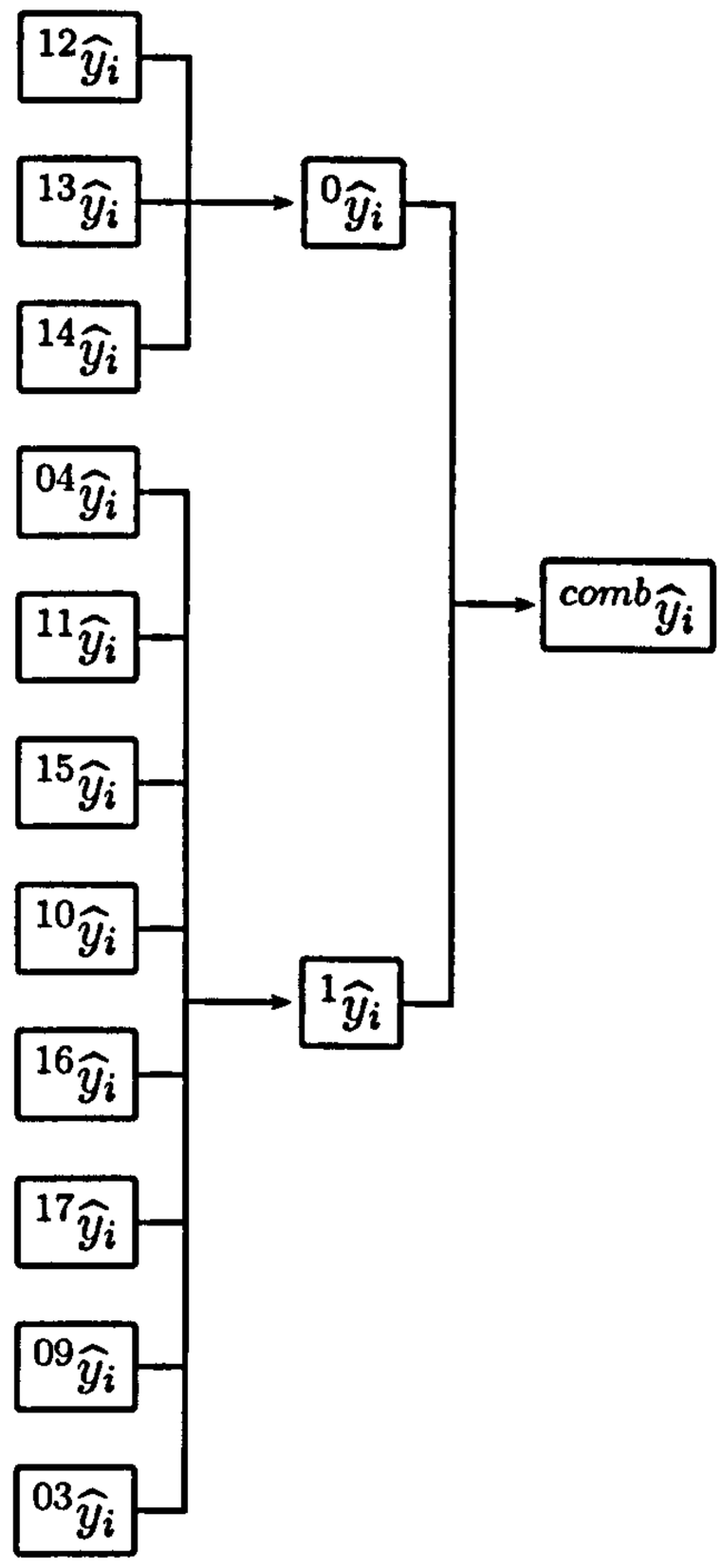

Fig. 46: Resulting combination structure for the example based on algorithm 1 proposed in [Aiolfi 04].

We will now analyse what effects the use of simpler combination models has on error variance when it is based on pools of forecasts in comparison to the optimal linear weights. We will analyse different cases of variance and covariance distribution in order to be able to evaluate the consequences for the combination of multi level forecasts.

6.2.3 Combining two forecasts

As we have already mentioned in Section 4.4.2 Timmermann [Timmermann 05] has derived the loss in the quality of the combined predictions between the optimal model and the optimal model with assumption of independence using an example of two forecasts.

Figure 29 illustrates on the example of two forecasts that it is risky to combine forecasts for which the errors differ significantly without taking into account covariance information. The biggest losses occur for small values of variance ratio (meaning big differences in the forecast errors) in connection with high correlation values. If we have, e.g., error variance ${}^2\delta^2 = 0.5 * {}^1\delta^2$, and error correlation 0.7, we already loose 10% of forecast accuracy according to equation (4.12) when using F^{var} instead of F^{opt}.

Therefore, we can state that forecasts with significantly different quality of errors should not be combined without taking into account the covariances.

For forecasts with about the same level of errors the combination of two forecasts is much more stable. As it can be seen in Figure 29, if the ratio of the variances of the two forecasts is near 1, the graph reaches a plateau where the covariance between the forecasts does not matter.

The approach of Aiolfi and Timmermann to combine only forecasts with similar error variances is therefore a good idea seen from the perspective of only two forecasts to be combined per pool. We will now analyse if this behaviour can be generalised.

6.2.4 The general case: combining more than two forecasts

Let us consider a more general case of a combination of more than two forecasts. Unfortunately, in this case it is not possible to state that for the combination of more than two forecasts the covariance does not matter if we combine forecasts with the same level of error variances. It is the homogeneity of covariance values that determines the potential loss of forecast quality. We will show using three

examples that criterion 3 of Section 6.1.4 is critical for the quality of the combined forecasts using the pooling approaches of Aiolfi and Timmermann.

Two examples of extreme cases concerning homogeneity

Let us assume that we have M forecasts and want to combine these given a covariance matrix Σ. The optimisation problem to solve is the determination of a vector of linear weights w fulfilling $min(w'\Sigma w)$ under the condition $w'\eta = 1$ where $\eta = (\{1\})^M$ represents the $[M * 1]$ unit vector. Generating the Lagrangian and solving the first order condition lead to the well known formula (3.11) provided in [Bates 69]

$$w = \left[\frac{\Sigma^{-1}\eta}{\eta^T\Sigma^{-1}\eta}\right]. \tag{6.1}$$

If we have optimal homogeneity meaning the same error variance δ^2 and correlation $\varrho = \frac{\rho}{\delta^2}$ for all forecasts to be combined, the inverse of the covariance matrix is given by

$$\Sigma^{-1} = \frac{1}{\delta^2(1-\varrho)} * (I - \frac{\varrho}{1+(M-1)\varrho}\eta\eta'). \tag{6.2}$$

Inserting this into (3.11) leads directly to $w = \left(\frac{1}{M}\right)\eta$. Details related to the proof of this fact can be found in [Timmermann 05].

If we have forecasts which all have about the same error variance and covariance level, then the resulting optimal weights are equal weights.

It follows that for totally homogenous covariances we have no loss in using the error variance based model or the simple average model compared to the optimal model.

Let us now assume that among the M forecasts we have $M-1$ identical ones (meaning $\varrho = 1$) and one forecast which is uncorrelated to all of the others. The relative error increase in using the simple average model can be described by

$$l = 2 * \frac{M + (M-2) * (M-1)}{M^2}. \tag{6.3}$$

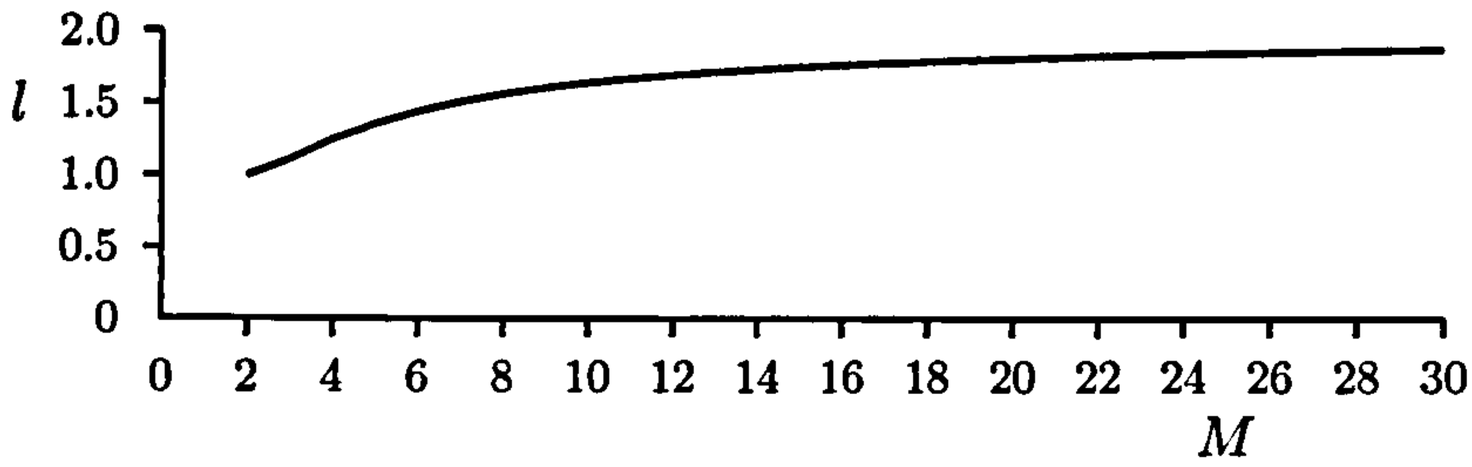

Fig. 47: Graphical representation of equation 6.3.

A graphical representation is shown in Figure 47. It means that if reliable information about correlation was available, the single uncorrelated forecast could be much more effectively used especially when there are many other correlated forecasts combined at the same time.

Assumption of two homogenous groups of forecasts

We assume again equal error variances and analyse covariance effects in order to see what can happen during the combination of one cluster related to the approach of Aiolfi and Timmermann. We will now analyse a special case concerning the structure of the coveriance matrix which, as we will see later, plays an important role in the case of multi level forecasting. We expect two sub-matrices with perfect homogeneity concerning covariances.

We assume

$$\Sigma = \left(\begin{array}{c|c} {}^{1}\Sigma & {}^{3}\Sigma \\ \hline {}^{3}\Sigma^{T} & {}^{2}\Sigma \end{array} \right) \tag{6.4}$$

with

$${}^{1}\Sigma \in \mathcal{R}^{M_1 \times M_1} = \begin{pmatrix} \delta^2 & {}^{1}\rho & \dots & {}^{1}\rho \\ {}^{1}\rho & \delta^2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & {}^{1}\rho \\ {}^{1}\rho & \dots & {}^{1}\rho & \delta^2 \end{pmatrix} \tag{6.5}$$

$$ {}^{2}\Sigma \in \mathcal{R}^{M_2 \times M_2} = \begin{pmatrix} \delta^2 & {}^{1}\rho & \dots & {}^{1}\rho \\ {}^{1}\rho & \delta^2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & {}^{1}\rho \\ {}^{1}\rho & \dots & {}^{1}\rho & \delta^2 \end{pmatrix} \tag{6.6} $$

both with optimal homogeneity, and

$$ {}^{3}\Sigma \in \mathcal{R}^{M_1 \times M_2} = \begin{pmatrix} {}^{2}\rho & {}^{3}\rho & \dots\dots & {}^{3}\rho \\ {}^{3}\rho & {}^{2}\rho & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ {}^{3}\rho & \dots & {}^{3}\rho & {}^{2}\rho & {}^{3}\rho & \dots & {}^{3}\rho \\ {}^{3}\rho & \dots\dots\dots\dots & \dots & {}^{3}\rho \\ \vdots & & & & \vdots \\ {}^{3}\rho & \dots\dots\dots\dots & \dots & {}^{3}\rho \end{pmatrix} \tag{6.7} $$

containing the value ${}^{2}\rho$ in M_3 "diagonal" elements[1] and the value

$$ {}^{3}\rho = {}^{1}\rho + {}^{2}\rho - \delta^2 > 0 \tag{6.8} $$

otherwise. The matrices ${}^{1}\Sigma$ and ${}^{2}\Sigma$ represent each a group of forecasts with equal variances and homogeneous covariances. They differ only in the size of the matrices. The matrix ${}^{3}\Sigma$ defines the relation between the two groups of forecasts in terms of covariances which is again expected to be homogeneous except the relation between special pairs of forecasts for which the relation is represented in the "diagonal" elements.

If we run the combination using the simple average model and achieve weights $w = \left(\frac{1}{M_1+M_2}\right)\eta$. We can estimate the total error variance of the combined forecast by (4.1)

$$ {}^{comb}\delta^2 = \sum_{m=1\dots 2M, n=1\dots 2M} w_m w_n ({}^{m,n}\rho) \tag{6.9} $$

[1] With "diagonal" we mean here that there is never more than one of these values per row and per column.

and achieve for the equal weights

$$\begin{aligned} {}^{comb}\delta^2 &= \frac{1}{(M_1+M_2)^2} * \\ &\quad [(M_1+M_2-2M_1M_2+2M_3)\delta^2 \\ &\quad +((M_1+M_2)^2-M_1-M_2-2M_3)({}^1\rho) \\ &\quad +2M_1M_2({}^2\rho))] \end{aligned} \tag{6.10}$$

Let us now assume we do not group all of the forecasts as it would be done by Aiolfi and Timmermann, but split them corresponding to the two groups of forecasts with homogeneous covariance values related to ${}^1\Sigma$ and ${}^2\Sigma$. For each of the group we have perfect homogeneity with respect to their covariance values, we can therefore use weights $w_1 = \left[\frac{1}{M_1}\right]_{M_1}$ and $w_2 = \left[\frac{1}{M_2}\right]_{M_2}$ without having to expect any loss compared to the use of the optimal model. We achieve ${}^1\delta^2 = \frac{1}{M_1}\delta^2 + \frac{M_1-1}{M_1}({}^1\rho)$ and ${}^2\delta^2 = \frac{1}{M_2}\delta^2 + \frac{M_2-1}{M_2}({}^1\rho)$. That leads using F^{var} to the total linear combination weights (including the two combinations) $w = \begin{pmatrix} [w_1]_{M_1} \\ [w_2]_{M_2} \end{pmatrix}$ with $w_1 = \frac{1}{M_1} * \frac{{}^2\delta^2}{{}^1\delta^2+{}^2\delta^2}$, $w_2 = \frac{1}{M_2} * \frac{{}^1\delta^2}{{}^1\delta^2+{}^2\delta^2}$ and together with (6.9) to

$$\begin{aligned} {}^{comb}\widetilde{\delta^2} &= (M_1 * w_1^2 + M_2 * w_2^2) * \delta^2 \\ &+ (w_1^2 * M_1 * (M_1-1) + w_2^2 * M_2 * (M_2-1))^1\rho \\ &+ 2M_3 w_1 w_2 ({}^2\rho) \\ &+ (2M_1M_2 - 2M_3) w_1 w_2 ({}^1\rho + {}^2\rho - \delta^2) \end{aligned}$$

The relative error increase

$$l = \frac{{}^{comb}\delta^2}{{}^{comb}\widetilde{\delta^2}} \tag{6.11}$$

depending on ${}^1\rho, {}^2\rho$ (restricted to values fulfilling (6.8)) and assuming $\delta^2 = 1$, $M_1 = 2$, $M_2 = 6$ and $M_3 = 0$ is visualised in Figure 48. We will later see that this corresponds to the case of our example from Section 6.2.2 (except for the scaling of the whole covariance matrix with the error variance). Figure 48 shows that relevant

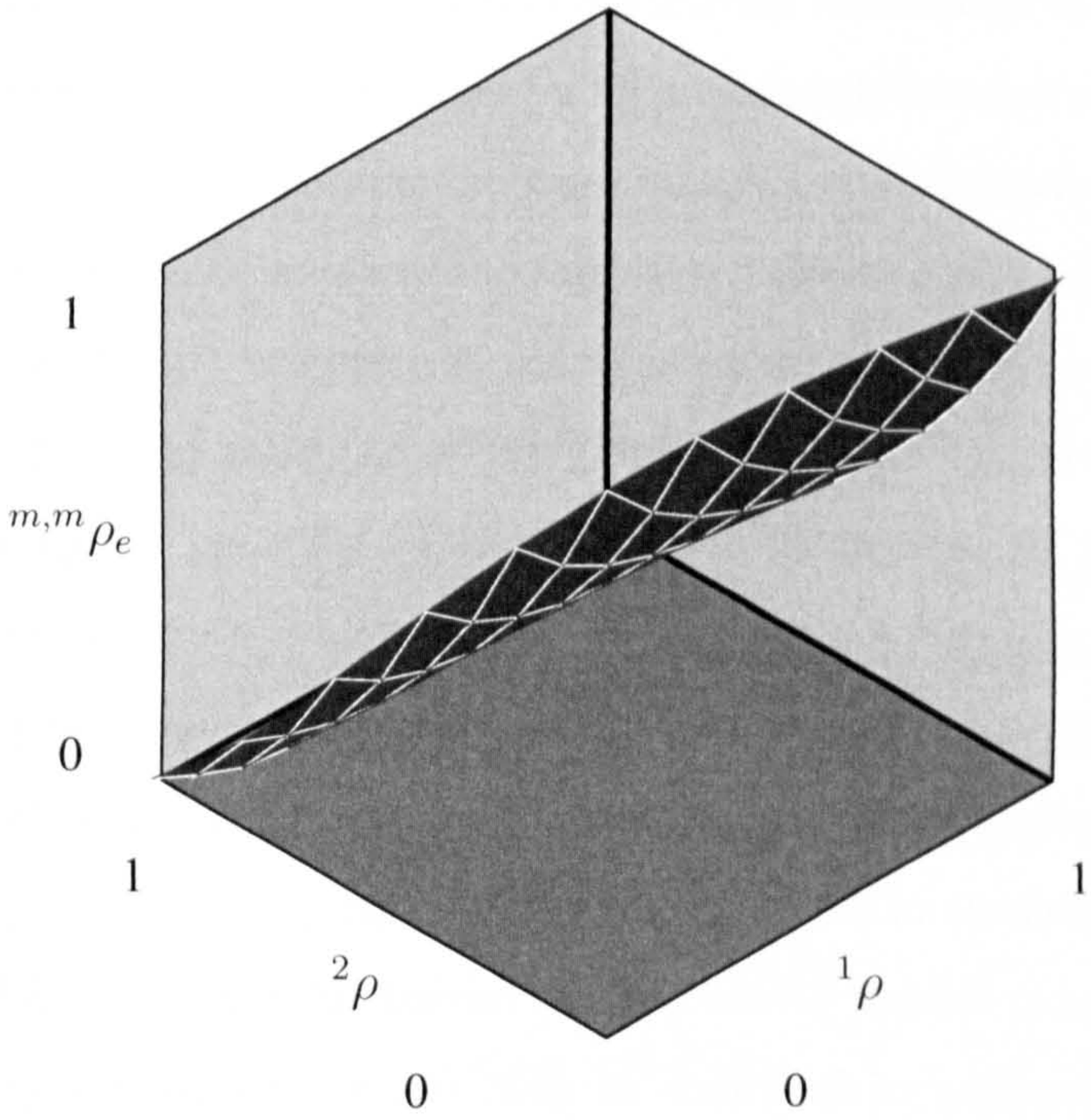

Fig. 48: Graphical representation of equation 6.11 assuming $\delta^2 = 1$, $M_1 = 2$, $M_2 = 6$ and $M_3 = 0$.

prediction accuracy loss (in the visualised example up to 25%) can ensue by using the approach of Aiolfi and Timmermann instead of using an additional splitting because of the inhomogeneities in the covariance matrix.

6.3 Issues of error variance based pooling for multi level forecasts

Let us assume we use the approach of Aiolfi and Timmermann for our previously defined set of individual forecasts (Figure 45) generated using different levels of learning as well as thick modelling concerning a set of parameters Φ^{α} and different function spaces $\mathcal{H}_k$. We achieve a large number of forecasts, so criterion 1 of Section 6.1.4 is not fulfilled. In [Riedel 05a] we have seen that forecasts generated with learning at the low level could be unstable (high error variance term because of extremely noisy training data), learning at the high level can lead to insufficiently adapted forecasts (high error variance term because of missing adaptation to special

features that can be observed at the low level). Both effects can lead to high total error variances for some of the generated forecasts. Suboptimal parameter values used for thick modelling or inadequate functions at one of the levels can have similar effects. Therefore, we have to expect that criterion 2 is also not fulfilled. Both problems can be solved with the approach of Aiolfi and Timmermann.

Let us now consider the third criterion, the homogeneity of the covariances. We have seen that for good results of the approach of Aiolfi and Timmermann this criterion is the critical one. We can get a better idea about expected covariances by analysing the effects of different kinds of forecast diversification (using different levels of learning, thick modelling and using different functions $h_k(;)$) on different independent components of the achieved forecast errors.

As we will see in Section 6.4 the analysis carried out in Section 6.3.1 leading to a better understanding of the effects of some forecast diversifying procedures on various error components will also enable us to take advantage of information contained in the covariance matrix without the need to calculate the covariance values themselves.

6.3.1 Impact of forecast diversification on the covariance matrix

Let us assume that we learn an optimal parameter set $\widehat{\phi}$ in i with a given function space in order to achieve $h(x, \widehat{\phi}) = h_{k,\phi_\alpha}(x, {}^{i}\widehat{\phi})$ described by the general type of function indicated by index k and the set of parameters ϕ_α used as constants in $h(;)$. We have seen in Section 4.1.3 that the total error variance ${}^{\mathcal{H}i}\delta_e^2$ can be decomposed into the independent components

$$ {}^{\mathcal{H}i}\delta_e^2 = \delta_h^2 + {}^{\mathcal{H}i}\delta_\phi^2 + \delta_y^2 \tag{6.12} $$

We have shown in [Riedel 05a] that learning at the high level I leads to a decomposition

$$ {}^{\mathcal{H}I}\delta_e^2 = \delta_h^2 + {}^{\mathcal{H}I}\delta_\phi^2 + \delta_y^2 \tag{6.13} $$

meaning that we have differences only in the error variance component.

We will now describe impacts of different kinds of forecast diversification on the covariance matrix by using the example of forecasts diversified by thick modelling together with forecasts with ϕ learned at different levels in one cluster.

The impact of diversification by using different levels of learning

Let us first assume two forecasts ${}^1\widehat{y} = h_{k,\phi_{a1}}(x, {}^i\widehat{\phi})$ and ${}^2\widehat{y} = h_{k,\phi_{a1}}(x, {}^I\widehat{\phi})$ where we have differences only in the level of learning using the same function space that we will call $\mathcal{H}_1$. As we want to analyse covariances of forecasts relating to the same pool corresponding to the pooling approach of Aiolfi and Timmermann, we also assume identical total error variances

$$ {}^{\mathcal{H}_1 i}\delta_e^2 = {}^{\mathcal{H}_1 I}\delta_e^2. \tag{6.14}$$

It can easily be seen that (5.19) and (6.13) represent decompositions for ${}^{\mathcal{H}_1 i}\delta_e^2$ and ${}^{\mathcal{H}_1 I}\delta_e^2$ and that these differ only in the error variance component. We have therefore

$$ {}^{\mathcal{H}_1 i}\delta_\phi^2 = {}^{\mathcal{H}_1 I}\delta_\phi^2. \tag{6.15}$$

The covariance ${}^{\mathcal{H}_1 i,\mathcal{H}_1 I}\rho_e$ between ${}^1\widehat{y_i}$ and ${}^2\widehat{y_i}$ can be achieved by comparing again equations (5.19) and (6.13). Because of identical bias component and Bayes component (they both do not depend on the level of learning) we have

$$ {}^{\mathcal{H}_1 i,h_1 I}\rho_e = \delta_{h_1}^2 + {}^{\mathcal{H}_1 i,\mathcal{H}_1 I}\rho_\phi + \delta_y^2. \tag{6.16}$$

The covariance ${}^{\mathcal{H}_1 i,\mathcal{H}_1 I}\rho_\phi$ of the error variance component is determined by the similarities between y_i and y_I (influenced among others by the aggregation parameter λ_i). If we assume significant differences between the levels we can also expect clearly distinctive values ${}^{\mathcal{H}_1 i,\mathcal{H}_1 I}\rho_\phi < {}^{\mathcal{H}_1 i}\delta_\phi^2$.

Equation (6.16) shows that differences between the forecasts ${}^1\widehat{y}$ and ${}^2\widehat{y}$ can

only be found in the error variance component. We can express the level of "diversity" $\Theta({}^1\widehat{y},{}^2\widehat{y})$ as the uncorrelated part of the total error variance in relation to the total error variance. Using (5.19) and (6.16) we get

$$\Theta({}^1\widehat{y},{}^2\widehat{y}) = 1 - \frac{{}^{\mathcal{H}_1 i,\mathcal{H}_1 I}\rho_e}{{}^{\mathcal{H}_1 i}\delta_e^2} = \frac{{}^{\mathcal{H}_1 i}\delta_\phi^2 \; {}^{\mathcal{H}_1 i,\mathcal{H}_1 I}\rho_\phi}{{}^{\mathcal{H}_1 i}\delta_e^2} \tag{6.17}$$

This representation has the advantage that it clearly shows that only the error variance component is responsible for any differences between the forecasts. If the levels are very different, we have a low covariance in the error variance component and with that very diverse predictions, which means a high improvement in forecast accuracy when the two forecasts are combined. The relation to the total error variance also indicates that large improvements in forecast accuracy can only be achieved if the error variance component is big in relation to the two other components.

The impact of diversification by using thick modelling

Let us now assume a third forecast ${}^3\widehat{y} = h_{k,\phi_{\alpha 2}}(x,{}^i\widehat{\phi})$ using another function space $\mathcal{H}_2$ and level i of learning. Let the parameter values be diversified as described in Section 4.3.2, this means that the parameter does not influence the complexity of the function space. As the differences between $\mathcal{H}_1$ and $\mathcal{H}_2$ are only caused by changes in predefined parameter values, we can assume similar complexity for learning meaning

$${}^{\mathcal{H}_1 i}\delta_\phi^2 = {}^{\mathcal{H}_2 i}\delta_\phi^2. \tag{6.18}$$

We want to study the effects when ${}^3\widehat{y}$ belongs to the same cluster as ${}^1\widehat{y}$ and ${}^2\widehat{y}$, we therefore assume identical total error variances

$${}^{\mathcal{H}_1 i}\delta_e^2 = {}^{\mathcal{H}_2 i}\delta_e^2. \tag{6.19}$$

As the Bayes component does not depend on $\mathcal{H}$, it follows from (5.19),(6.18) and

(6.19) that we have also identical bias $\delta^2_{h_1} = \delta^2_{h_2}$.

Let us now analyse the covariance ${}^{\mathcal{H}_1 i,\mathcal{H}_2 i}\rho_e$. As we have used different function spaces we can have relevant differences in the error bias component. On the other hand we have parameter settings learned at the same level using the same set of noisy input data so that we can have quite highly correlated error variance terms. We get

$$ {}^{\mathcal{H}_1 i,\mathcal{H}_2 i}\rho_e = \rho_{h_1,h_2} + {}^{\mathcal{H}_1 i,\mathcal{H}_2 i}\rho_\phi + \delta^2_y. \tag{6.20} $$

The differences in the error bias as well as in the error variance component are influenced by the differences between $\mathcal{H}_1$ and $\mathcal{H}_2$. If we assume that relevant differences between the function spaces exist, we can also expect clearly distinctive values $\rho_{h_1,h_2} < \delta^2_{h_1}$ for the error bias component. On the other hand, because of (6.18) and taking into account the fact that we learn using the same training data we have to expect a correlation factor near 1 in the error variance term. This corresponds to Granger and Jeon [Granger 04] who state that using the approach of thick modelling we often have the relevant differences in the error bias term in connection with an only slightly changing error variance term. We can therefore approximate ${}^{\mathcal{H}_1 i,\mathcal{H}_2 i}\rho_e$ by

$$ {}^{\mathcal{H}_1 i,\mathcal{H}_2 i}\rho_e \approx \rho_{h_1,h_2} + {}^{\mathcal{H}_1 i}\delta^2_\phi + \delta^2_y. \tag{6.21} $$

We see that using the different function spaces $\mathcal{H}_1$ and $\mathcal{H}_2$ leads to uncorrelated parts in the error bias component in opposition to the use of different levels for learning, where we have uncorrelated parts only in the error variance component.

We get the "diversity" of

$$ \Theta({}^1\widehat{y}, {}^3\widehat{y}) = 1 - \frac{{}^{\mathcal{H}_1 i,\mathcal{H}_2 i}\rho_e}{{}^{\mathcal{H}_1 i}\delta^2_e} \approx \frac{\delta^2_{h_1} - \rho_{h_1,h_2}}{{}^{\mathcal{H}_1 i}\delta^2_e} \tag{6.22} $$

We can clearly see that large improvements of the forecast accuracy in combining the forecasts can only be achieved if the bias error component is relevant

compared to the other two components and if the parameter change leads to relevant changes in the bias.

The resulting impact of diversification by thick modelling and different levels of learning

Let us now analyse the relation between forecast ${}^2\widehat{y}$ and forecast ${}^3\widehat{y}$. We already know that the forecasts belong to the same pool, because of (6.14) and (6.19) it follows that

$$^{\mathcal{H}_1 I}\delta_e^2 = ^{\mathcal{H}_2 i}\delta_e^2. \tag{6.23}$$

The error bias component is determined by the used functions, while the error variance component by both, the functions as well as the level of learning. We get

$$^{\mathcal{H}_1 I,\mathcal{H}_2 i}\rho_e = \rho_{h_1,h_2} + ^{\mathcal{H}_1 I,\mathcal{H}_2 i}\rho_\phi + \delta_y^2. \tag{6.24}$$

If we assume again that the difference between $\mathcal{H}_1$ and $\mathcal{H}_2$ does not have a big impact on the error variance component we can approximate

$$^{\mathcal{H}_1 I,\mathcal{H}_2 i}\rho_e \approx \rho_{h_1,h_2} + ^{\mathcal{H}_1 I,\mathcal{H}_1 i}\rho_\phi + \delta_y^2. \tag{6.25}$$

We see that now we have uncorrelated parts in the bias as well as in the error variance component. We can therefore expect a significantly lower total covariance. The relation to covariances $^{\mathcal{H}_1 i,\mathcal{H}_1 I}\rho_e$ and $^{\mathcal{H}_1 i,\mathcal{H}_2 i}\rho_e$ can be expressed by again using the diversity measure. We have

$$\begin{aligned}\Theta({}^2\widehat{y},{}^3\widehat{y}) &= 1 - \frac{^{\mathcal{H}_1 I,\mathcal{H}_2 i}\rho_e}{^{\mathcal{H}_1 i}\delta_e^2} \\ &\approx \frac{[\delta_{h_1}^2 - \rho_{h_1,h_2}] + [^{\mathcal{H}_1 i}\delta_\phi^2 - ^{\mathcal{H}_1 I,\mathcal{H}_1 i}\rho_\phi]}{^{\mathcal{H}_1 i}\delta_e^2}\end{aligned} \tag{6.26}$$

which leads with (6.17) and (6.22) to

$$\Theta({}^{2}\widehat{y},{}^{3}\widehat{y}) \approx \Theta({}^{1}\widehat{y},{}^{2}\widehat{y}) + \Theta({}^{1}\widehat{y},{}^{3}\widehat{y}). \tag{6.27}$$

and

$${}^{\mathcal{H}_1 I,\mathcal{H}_2 i}\rho_e \approx^{\mathcal{H}_1 i,\mathcal{H}_1 I} \rho_e +^{\mathcal{H}_1 i,\mathcal{H}_2 i} \rho_e -^{\mathcal{H}_1 i} \delta_e^2. \tag{6.28}$$

The achieved diversity shows that we risk to have *strong inhomogeneities in the covariance matrix* if we use more than one diversification criteria per pool. Equations (6.16) and (6.21) show that even the application of a single but changing diversification criterion like between forecast pairs $({}^{1}\widehat{y},{}^{2}\widehat{y})$ and $({}^{1}\widehat{y},{}^{3}\widehat{y})$ can lead to different impacts in relation to the error components and therefore generate significantly different covariance values. Even in the lucky case when the use of two diversification criteria leads to comparable covariances meaning that, e.g., ${}^{\mathcal{H}_1 i,\mathcal{H}_1 I}\rho_e \approx^{\mathcal{H}_1 i,\mathcal{H}_2 i} \rho_e$ and with that

$$\Theta({}^{1}\widehat{y},{}^{2}\widehat{y}) \approx \Theta({}^{1}\widehat{y},{}^{3}\widehat{y}), \tag{6.29}$$

we get the problem that automatically such pool contains at least one pair of forecasts generated using both diversification criteria which can lead to a significantly different covariance value. This can be seen on the example of ${}^{\mathcal{H}_1 I,\mathcal{H}_2 i}\rho_e$ and equation (6.27) leading to

$$\Theta({}^{2}\widehat{y},{}^{3}\widehat{y}) \approx 2 * \Theta({}^{1}\widehat{y},{}^{2}\widehat{y}) \tag{6.30}$$

under assumption (6.29).

Let us assume we have a larger set of forecasts $\{\widehat{y}\}$ with similar total error variance given and that this set represents the two diversification criteria as discussed in 6.3.1. Then we can write $\{\widehat{y}\}$ as $\{\widehat{y}\} = \{{}^{i}\widehat{y}\} \cup \{{}^{I}\widehat{y}\}$ representing different levels of learning with each pair of forecasts in ${}^{1}\{\widehat{y}\}$ or ${}^{2}\{\widehat{y}\}$ differing only by thick modelling. The covariance between each pair of forecasts in $\{{}^{i}\widehat{y}\}$ or in $\{{}^{I}\widehat{y}\}$ can be described similarly to equation (6.21) which means that we can expect homoge-

neous covariances in $\{^i\widehat{y}\}$ as well as in $\{^I\widehat{y}\}$. The covariance of any pair of forecasts representing different levels of learning can be estimated similarly to (6.28). This leads to a covariance structure similar to the covariance matrix analysed in Section 6.2.4. We have shown that combining pools of forecasts characterised by this type of covariance matrices can lead to big losses compared to the approach of applying an additional splitting corresponding to the clusters that relate only to one diversification criterion.

The problem of inhomogeneous covariance matrices is not only related to thick modelling and multi level learning. Similar inhomogeneous covariance matrices can occur if we combine, e.g., forecasts generated using different function spaces $\mathcal{H}_k$ in order to represent the option to use a more risky model together with learning at different levels. Summarising, we can state that if we pool the multi level forecasts using the method of Aiolfi and Timmermann, we risk large forecast accuracy losses in comparison to a forecast combination combining always forecasts that used only one diversification method.

Example

The relation of different forecasts to the level of learning and the value of ϕ_α are visualised in Figure 49.

Let us consider cluster 1 (the grey one) generated with F^{CEW}. The covariance matrix related to this cluster (containing the forecasts ordered by its number) is

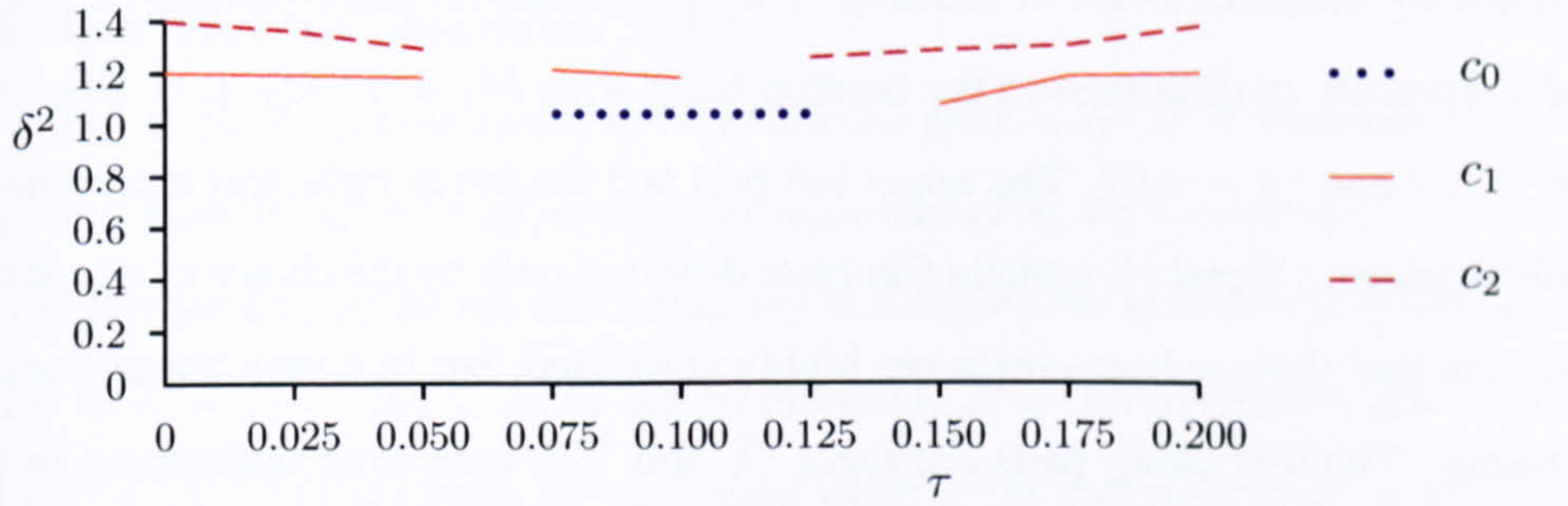

Fig. 49: Graphical representation of the errors given in the example shown in Table 15. Different line styles represent different clusters generated with F^{CEW}. The upper lines are the errors learned at the low level i, the lower lines represent errors learned at the high level.

given by

$$
{}^1\Sigma = \left(\begin{array}{cc|cccccc}
1.21 & 1.03 & 0.54 & 0.55 & 0.55 & 0.53 & 0.55 & 0.57 \\
1.03 & 1.18 & 0.57 & 0.56 & 0.53 & 0.52 & 0.54 & 0.55 \\
\hline
0.54 & 0.57 & 1.2 & 1.09 & 1.09 & 1.07 & 1.07 & 1.07 \\
0.55 & 0.56 & 1.09 & 1.19 & 1.08 & 1.08 & 1.07 & 1.07 \\
0.55 & 0.53 & 1.09 & 1.08 & 1.18 & 1.09 & 1.08 & 1.07 \\
0.53 & 0.52 & 1.07 & 1.08 & 1.09 & 1.18 & 1.08 & 1.07 \\
0.55 & 0.54 & 1.07 & 1.07 & 1.08 & 1.08 & 1.19 & 1.08 \\
0.57 & 0.55 & 1.07 & 1.07 & 1.07 & 1.07 & 1.08 & 1.2
\end{array}\right) \tag{6.31}
$$

$$
\approx \left(\begin{array}{cc|cccccc}
\mathbf{1.2} & 1.07 & 0.55 & 0.55 & 0.55 & 0.55 & 0.55 & 0.55 \\
1.07 & \mathbf{1.2} & 0.55 & 0.55 & 0.55 & 0.55 & 0.55 & 0.55 \\
\hline
0.55 & 0.55 & \mathbf{1.2} & 1.07 & 1.07 & 1.07 & 1.07 & 1.07 \\
0.55 & 0.55 & 1.07 & \mathbf{1.2} & 1.07 & 1.07 & 1.07 & 1.07 \\
0.55 & 0.55 & 1.07 & 1.07 & \mathbf{1.2} & 1.07 & 1.07 & 1.07 \\
0.55 & 0.55 & 1.07 & 1.07 & 1.07 & \mathbf{1.2} & 1.07 & 1.07 \\
0.55 & 0.55 & 1.07 & 1.07 & 1.07 & 1.07 & \mathbf{1.2} & 1.07 \\
0.55 & 0.55 & 1.07 & 1.07 & 1.07 & 1.07 & 1.07 & \mathbf{1.27}
\end{array}\right)
$$

It can be seen that the level of covariances differ in different parts of the matrix separating different levels of learning. The approximation corresponds to a matrix of a structure as discussed in the Section 6.2.4 with $M_1 = 2$, $M_2 = 6$, $M_3 = 0$, $\delta^2 = 1.2$ and ${}^1\rho = 1.07$. The upper left part and the lower right part representing substructures ${}^1\Sigma$ and ${}^2\Sigma$ contain forecasts differing only by the choice of ϕ_α. It can be seen that these substructures are highly correlated, but in a very homogeneous manner. The remaining parts represent ${}^3\Sigma$ and ${}^3\Sigma^T$ indicating differences in the level of learning. In this example there are forecasts where the parameter ϕ_α is identical meaning that all forecasts differing in the level differ also concerning the parameter value. That is why the "diagonal" elements containing a clearly higher covariance values compared to the other elements disappear in that substructure for our example.

Following (6.11) with this data leads to $l \approx 1.07$ which means that we loose about 7 percent of forecast accuracy if we combine the cluster without an additional splitting of the covariance matrix because of inhomogeneities based on the different error decomposition relating to the different diversification criteria.

6.4 Pooling based on the Distance in the Forecast Generation Space

We propose here an alternative pooling approach especially for multi level forecasts that takes into account some of the information that we have about the generation of the forecasts. If high quality covariance information is available, we can use it directly in order to generate clusters. We consider here the risk because of quickly changing environments, very noisy training data or frequently occurring structural breaks covariances may not be measured properly. An additional very relevant reason of not using covariance values directly is the increased calculation time that is needed in order to calculate the matrix and to carry out the clustering. Instead of using covariance values directly, we use the information which we have about the forecast generation process as an additional indicator in order to generate clusters which are characterisised by more homogeneous covariances.

6.4.1 Definition of the Forecast Generation Space

Definition (Forecast Generation Space)

Let $y_i \in \mathcal{R}^n$ be a time series to be predicted given a set of K function spaces $(\mathcal{H}_k)$ and a set $\Phi^\alpha \subset R^{m_\alpha}$ of parameters represented by thick modelling.

Let further $\mathcal{I} \subset \mathcal{N}$ be the set containing indices for the used levels of learning $\{i, I\}$, let $\mathcal{K} = \{1 \dots K\} \subset \mathcal{N}$ be the set of indices of all used function spaces $\mathcal{H}_k$ and $\mathcal{M}^\alpha \subset \mathcal{N}^{m_\alpha}$ be an index for each used value of the parameters $\phi^\alpha \in \Phi^\alpha$.

Then the forecast generation space $\mathcal{S} = \mathcal{I} \times \mathcal{K} \times \mathcal{M}^\alpha \subset \mathcal{N}^{m_\alpha+2}$ is a unique description of a forecast generation process for y_i concerning the used function space, predefined parameter values and the used level of learning.

In the following we will represent each forecast for y_i as ${}^s\widehat{y}_i$, $s \in \mathcal{S}$ in order to indicate details related to its generation.

For our example we have $\mathcal{I} = [0, 1]$ (index 0 representing i and index 1 representing I), $K = [0]$ (in this example we use only one function space) and $\mathcal{M}^\alpha = [0, \dots, 4]$ representing the index of the five used parameter values for ϕ_α. Table 45 contains in column "s" the description of each of the forecasts in space $\mathcal{S}$.

Let us now analyse the expected covariances between the errors of a pair of forecasts $({}^{s1}\widehat{y}_i, {}^{s2}\widehat{y}_i)$. We have seen in the previous example that each difference in each dimension can have different effects on the correlation between different error components. If $s1$ and $s2$ differ only in the setting of one parameter, we expect a different correlation compared to differences in more than one dimension in $\mathcal{S}$.

That is why we introduce a distance measure for elements in $\mathcal{S}$ in order to describe expected similarity in the error decomposition.

Definition (Distance in the Forecast Generation Space)

Let $s1, s2 \in \mathcal{S}$. Let further in the following D indicate an unspecified dimension in $\mathcal{S}$ and s_D the value of dimension D in any element $s \in \mathcal{S}$.

	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17
00	0	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2
01	1	0	1	1	1	1	1	1	1	2	1	2	2	2	2	2	2	2
02	1	1	0	1	1	1	1	1	1	2	2	1	2	2	2	2	2	2
03	1	1	1	0	1	1	1	1	1	2	2	2	1	2	2	2	2	2
04	1	1	1	1	0	1	1	1	1	2	2	2	2	1	2	2	2	2
05	1	1	1	1	1	0	1	1	1	2	2	2	2	2	1	2	2	2
06	1	1	1	1	1	1	0	1	1	2	2	2	2	2	2	1	2	2
07	1	1	1	1	1	1	1	0	1	2	2	2	2	2	2	2	1	2
08	1	1	1	1	1	1	1	1	0	2	2	2	2	2	2	2	2	1
09	1	2	2	2	2	2	2	2	2	0	1	1	1	1	1	1	1	1
10	2	1	2	2	2	2	2	2	2	1	0	1	1	1	1	1	1	1
11	2	2	1	2	2	2	2	2	2	1	1	0	1	1	1	1	1	1
12	2	2	2	1	2	2	2	2	2	1	1	1	0	1	1	1	1	1
13	2	2	2	2	1	2	2	2	2	1	1	1	1	0	1	1	1	1
14	2	2	2	2	2	1	2	2	2	1	1	1	1	1	0	1	1	1
15	2	2	2	2	2	2	1	2	2	1	1	1	1	1	1	0	1	1
16	2	2	2	2	2	2	2	1	2	1	1	1	1	1	1	1	0	1
17	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	0

Tab. 17: Distance matrix related to the example shown in Table 15 depending on the position in $\mathcal{S}$.

Then the distance $\Delta : \mathcal{S} \times \mathcal{S} \rightarrow \mathcal{N}$ *is defined as*

$$\Delta(s1, s2) = \sum_{D \in [0,\ldots,m_\alpha+2]} \begin{cases} 0: & s1_D = s2_D \\ 1: & otherwise \end{cases} \tag{6.32}$$

The distance expresses the number of dimensions of the forecast generation space in which the forecast generation differs. The matrix of pairwise distances represents a kind of simplified version of the covariance matrix which we can use for pooling. Table 17 shows the distance matrix related to our example given in Table 15.

6.4.2 The Clustering Algorithm

Seen from the covariance aspect, we should only cluster predictions with a pairwise distance of 1. Choosing elements including a distance larger than 1 would mean that we risk the inhomogeneities in the covariance matrix as we have described it in the preceding sections.

In contrast, pairwise distance of 1 for all elements per cluster means that the

elements differ only in exactly one dimension D which promises a more homogeneous covariance matrix. If the dimension D has only a range of 2, we do not risk any problems because of the covariance if the variance ratio is close to 1 (see Section 6.2.3). For larger ranges we can expect homogenous correlation between different error components at least in the case of thick modelling. As this again means that all elements $s_{\tilde{D}}, \tilde{D} \neq D$ of all other dimensions are constant per cluster, each cluster can be described by an element of the space S/D meaning space S reduced by dimension D.

The proposed clustering can therefore be interpreted as an aggregation of one dimension in the forecast generation space.

In order to avoid too many variations in the error variances in such sets of forecasts, we follow a trimming strategy and eliminate all those forecasts with relatively bad quality related to that dimension (e.g. we discard the obviously bad parameter values for a given model at a given level or predictions at completely unstable levels for a given model with given parameter settings, etc.).

The algorithm realising a multi level fusion F^{mlp} of a set of forecasts $\{{}^{s}\widehat{y}_i\}$ related to a given forecast generation space $\mathcal{S}$ into a set of forecasts $\{{}^{\tilde{s}}\widehat{y}_i\}$ representing the clusters can be summarised as follows.

Algorithm 2: $F^{mlp}(\{{}^{s}\widehat{y}_i\}, S) \rightarrow (\{{}^{\tilde{s}}\widehat{y}_i\}, \tilde{S})$

1. select a dimension D of S that should be aggregated
2. set $\tilde{S} = S/D$ as the forecast generation space of the cluster results
3. for each $\tilde{s} \in \tilde{S}$: cluster $c_{\tilde{s}} = \{{}^{s}\widehat{y}_i : s/s_D = \tilde{s}\}$
4. remove the worst forecasts using a trimming procedure per cluster $c_{\tilde{s}}$
5. for each cluster $c_{\tilde{s}}$ run a linear combination F in order to achieve the forecast ${}^{\tilde{s}}\widehat{y}_i = F(c_{\tilde{s}})$

Depending on the strength of the used trimming strategy the *simple average model* or the *optimal model with assumption of independence* can be chosen as

combination model F. For our application we have used a strong trimming. We have not accepted more than the best three forecasts per cluster. All forecasts of which the total error variance differed more than 5% of that of the best forecast of the cluster have been removed as well.

Example

Let us carry out the clustering related to our example shown in Table 15. We start with $\mathcal{S} = I \times K \times \mathcal{M}^{\alpha} = [0,1] \times [0] \times [0,\ldots,8]$ as described above.

Step 1 : We select $D = 2$ meaning that the aggregated dimension is $\mathcal{M}^{\alpha}$.

Step 2 : The resulting forecast generation space is $\widetilde{\mathcal{S}} = \mathcal{S}/\{\mathcal{M}^{\alpha}\} = I \times K$=$\{(0,0),(1,0)\}$.

Step 3 : We generate the cluster :

- $c_{(0,0)} = \{{}^{s}\widehat{y}_i : s/s_D = (0,0)\}$
  $= \{{}^{(0,0,0)}\widehat{y}_i, {}^{(0,0,1)}\widehat{y}_i, {}^{(0,0,2)}\widehat{y}_i, \ldots, {}^{(0,0,8)}\widehat{y}_i\}$
- $c_{(1,0)} = \{{}^{s}\widehat{y}_i : s/s_D = (1,0)\}$
  $= \{{}^{(1,0,0)}\widehat{y}_i, {}^{(1,0,1)}\widehat{y}_i, {}^{(1,0,2)}\widehat{y}_i, \ldots, {}^{(1,0,8)}\widehat{y}_i\}$

Step 4 : We trim the clusters by choosing the best predictions per cluster. Only up to three forecast are selected per cluster. All forecasts for which the total error variance differs by more than 5% in comparison to that of the best forecast of the cluster, are removed as well.

- $c_{(0,0)} = \{{}^{(0,0,3)}\widehat{y}_i, {}^{(0,0,4)}\widehat{y}_i\}$
- $c_{(1,0)} = \{{}^{(1,0,3)}\widehat{y}_i, {}^{(1,0,4)}\widehat{y}_i, {}^{(1,0,5)}\widehat{y}_i\}$

Step 5 : We run the combination for each cluster

- $c_{(0,0)} = \{{}^{(0,0,3)}\widehat{y}_i, {}^{(0,0,4)}\widehat{y}_i\} \rightarrow {}^{(0,0)}\widehat{y}_i$
- $c_{(1,0)} = \{{}^{(1,0,3)}\widehat{y}_i, {}^{(1,0,4)}\widehat{y}_i, {}^{(1,0,5)}\widehat{y}_i\} \rightarrow {}^{(1,0)}\widehat{y}_i$

The combination of the results can be carried out in a second step.

Based on (6.9) we achieve with this structure a total forecast error of 0.816 in comparison to a value of 0.877 achieved with the structure of Aiolfi and Timmermann shown in Figure 46. This means a reduction of 6.9% of the total forecast error.

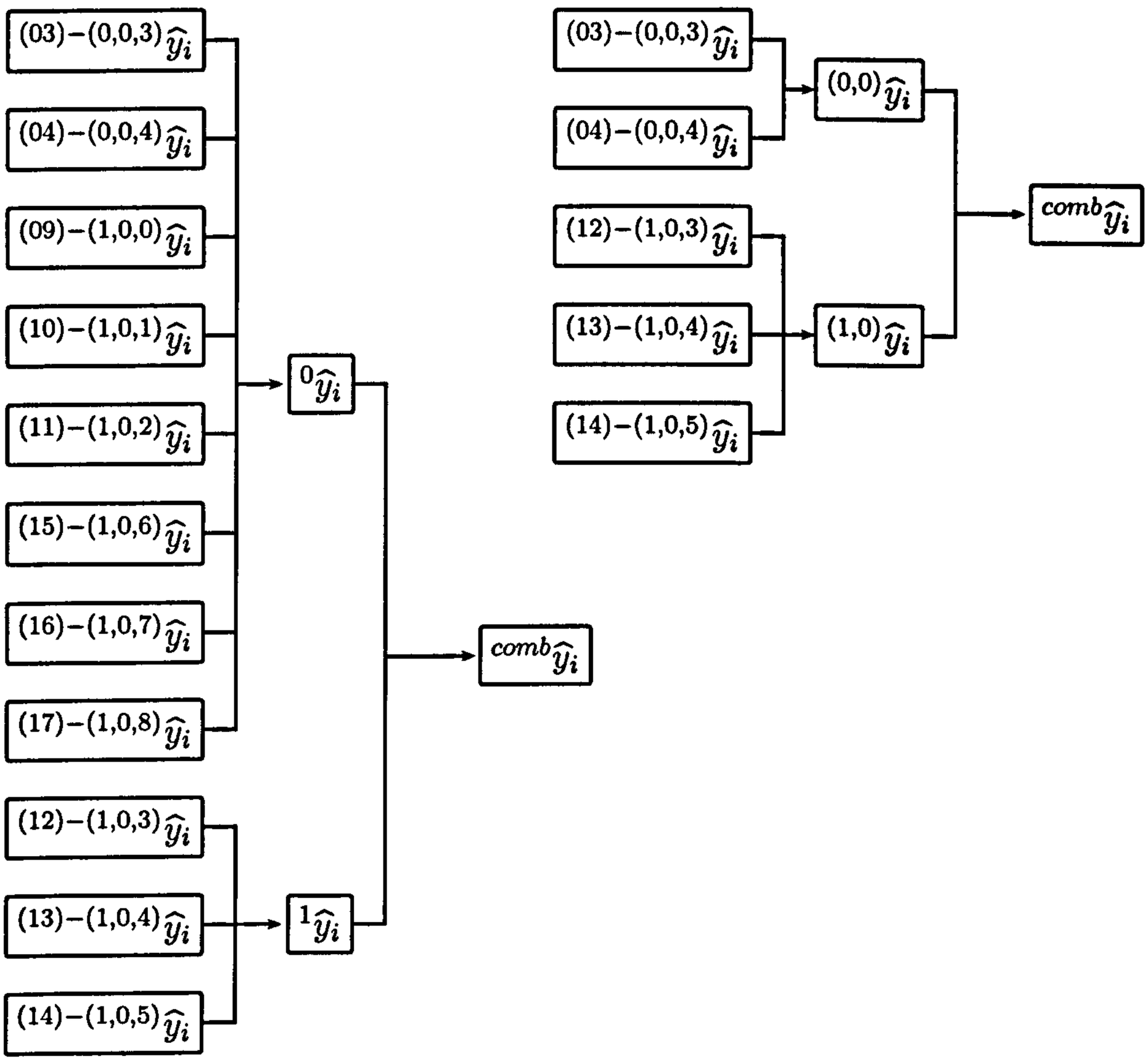

Fig. 50: Comparison of the achieved combination structures. The left structure is the combination structure achieved with the approach of Aiolfi and Timmermann for our example with the error variances given in Table 15 and covariances given in Table 16. The right structure is the structure achieved using the information about the forecast generation space. The input forecasts are described first by the number of the forecast in Table 16, then the position in the forecast generation space is provided as additional inormation (for instance $^{(12)-(1,0,3)}\widehat{y}_i$ means forecast number 12 with position (1,0,3) in the forecats generation space).

Figure 50 shows the achieved combination structure in comparison to the structure achieved with the algorithm of Aiolfi and Timmermann that we have already

shown in Figure 46. In order to be able to compare the structures we have labelled and ordered the forecasts per cluster depending on their position in the forecast generation space. It can be seen that the second cluster is identical. In the first cluster the two first forecasts are also contained in both versions of pooling, but the remaining forecasts have been removed by our method. We can see that this is beneficial by looking at the position in the forecast generation space of these forecasts. All of the removed forecasts have been generated at the high level. As they do not represent the best forecasts at this level and we have clearly better forecasts generated at this level included in the second pool, these forecasts do not contain relevant diverse information. Including them has the effect that the other two forecasts of the same pool get a lower total combination weight. As these two forecasts do contain diverse information because they have been generated at level i, these forecasts have not high enough influence in the structure of Aiolfi and Timmermann. So summarising once more, the information about the diversity cannot be achieved by considering only the total error variances. Considering the information about the forecast generation, on the other hand, enables us to make certain assumptions about potential diversity which as illustrated has led to the increased forecast accuracy.

6.4.3 Generation of multi step combination structures

As we have mentioned, the proposed pooling represents an aggregation of one dimension in the forecast generation space. The result of a pooling related to a dimension D is again a set of forecasts the generation of which can be defined by the forecast generation space S/D. If S/D contains only one element (meaning all exsting dimensions have already been aggregated or have the range 1), the pooling has generated a final result which can be used as the final combined forecasts. Otherwise, we can combine the remaining forecasts using a flat combination. But as the number of resulting forecasts can still be big, there is the other option to repeat the pooling approach based on S/D, a chosen dimension $\tilde{D} \neq D$, etc.

This idea leads to an approach of the successive generation of pools and so the generation of multi step combination structures. Each step leads to the reduction of one dimension of $\mathcal{S}$ so that the total number of steps is defined by the dimensionality of $\mathcal{S}$. The procedure of the generation of the structures can be described as follows.

Algorithm 3: $F^{mlps}(\{^s\widehat{y}_i\}, \mathcal{S}) \rightarrow^{comb} \widehat{y}_i$

1. set $\mathcal{S}^0 = \mathcal{S}$, q=0 (the step), $Y^0 = \{^s\widehat{y}_i\}$

2. while $q < m_\alpha + 2$ and $|\mathcal{S}^q| > 1$:
   $(\mathcal{S}^{q+1}, Y^{q+1}) = F^{mlp}(\mathcal{S}^q, Y^q), q = q + 1$

3. set $^{comb}\widehat{y}_i = Y^q$

Figure 51 shows an extract of the resulting structure for an example containing more than one function space ($K = 3$), two parameters controlled by thick modelling ($m_\alpha = 2$) and $\mathcal{S} = [0,1] \times [0,\ldots,3] \times [0,\ldots,10] \times [0,\ldots,8]$. In this example we have again used the trimming approach of selecting always the best three forecasts per pool.

6.5 Determining Pools based on the Estimated Covariance Matrix

We have seen that in each step algorithm F^{ml} needs the information which dimension D is used for the next step of pooling. In the example shown in Figure 51 we first combine dimension D_3, then dimension D_4 and so on.

The question of which dimension to choose next is a crucial task. If we assume covariance information as not reliable we do not have the needed information in order to make this decision on a theoretical basis. The best order can then only be determined on the basis of the resulting forecasts. One option is to carry out an experimental study during a phase of data analysis. We will discuss automatic and adaptive alternatives in the next chapter.

But what to do if good covariance estimates are available? In this section we

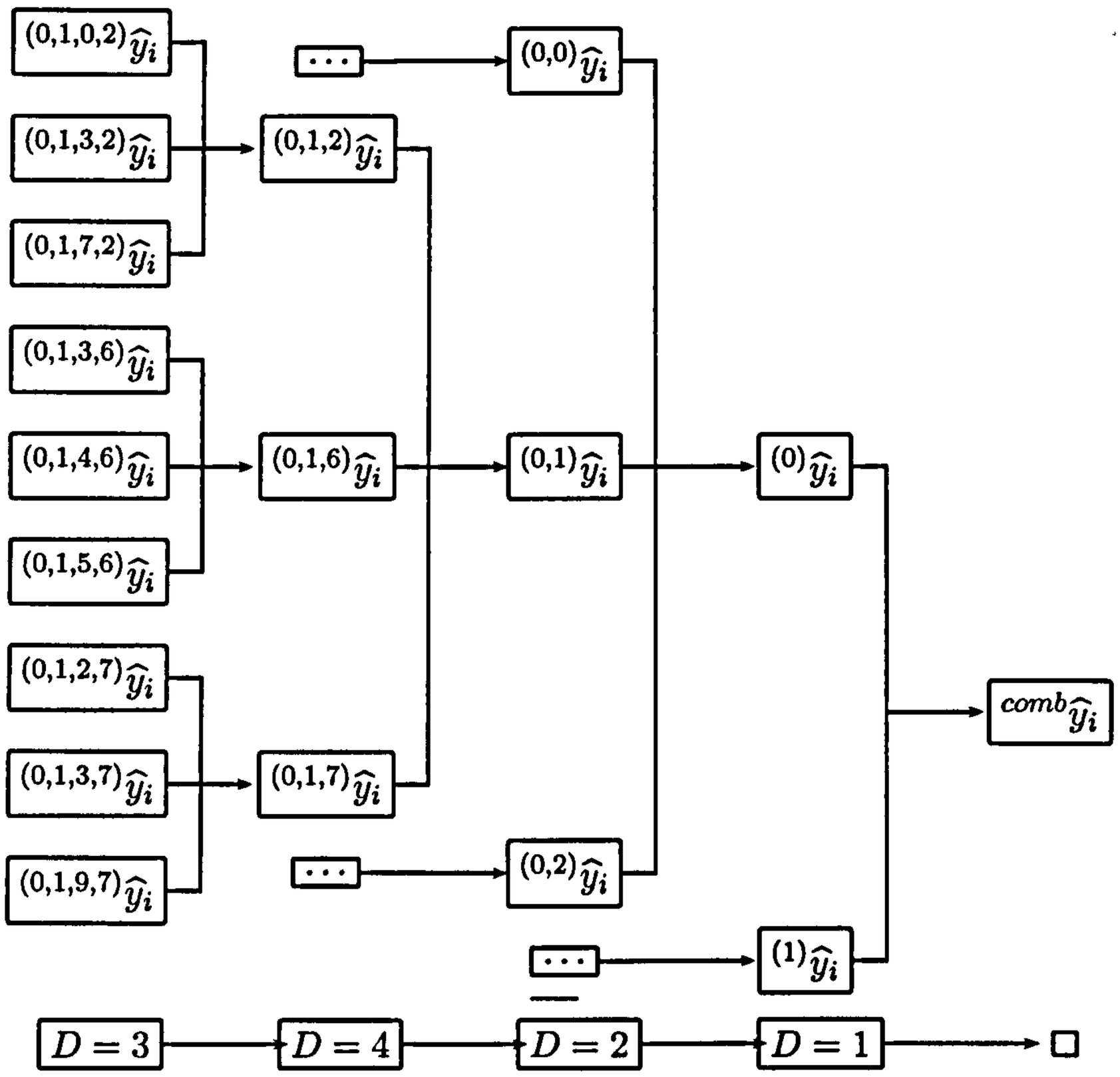

Fig. 51: Extract of a more complex combination structure with $\mathcal{S} = [0,1] \times [0,\ldots,3] \times [0,\ldots,10] \times [0,\ldots,8]$. Below the line it is indicated which dimension D has been chosen in each step.

will discuss different options of additionally using more or less reliable covariance information.

If reliable covariance information is available, it is possible to apply the optimal model directly. But also in this case we have to consider the risk of instabilities based on small deviations in the covariance matrix in case of a large number of forecasts containing sets of similarly correlated forecasts. As we have described in Section 4.4.1 such groups can lead to a covariance matrix that is similar to a singular matrix with a much lower rank. The resulting weights strongly depend on the small deviations in the covariance estimates and are often characterised by very large numbers of opposite sign.

It is therefore useful to apply pooling or trimming approaches even if only slightly disturbed covariance information is available. We will now discuss different options of how to do this.

6.5.1 Trimming: Selecting a Representative Set of Input Forecasts

The first option realises a flat combination after having carried out an intensive trimming. The idea here is to remove forecasts in a controlled manner in order to avoid inconsistencies in the covariance matrix and then carry out a flat combination on a much smaller set of representative and diverse input forecasts. We will discuss different options of how to select these forecasts in Section 7.1.3 of the next chapter.

6.5.2 Using Covariance Information for Pooling

If covariance information is reliable, we can use this information in order to generate pools based on this information instead or in addition to the information about the diversification process. Unfortunately, the covariance information allows a proper determination of the appropriate dimension for pooling on the basis of covariance homogeneity or resulting forecast performance only for one next step of pooling. We would need to have higher order statistics information in order to calculate the correlation of forecasts representing the results of the first step of pooling. This means that even with reliable covariance values there is the need for a) a very time expensive recalculation of covariances after each step of pooling or b) an estimation of covariances between the results of the first step of pooling.

6.5.3 Generating Pools based on Covariance Homogeneity

The definition of pools can be carried out in a manner that we generate pools which are as homogeneous as possible. This can be motivated as follows.

We have already argued that inhomogeneities in the covariance matrix lead to errors in the estimation of the weights if we apply a less complex linear combina-

tion model, such as F^{av} (the simple average model) or F^{var} (the optimal model with assumption of independence). In Section 6.2.4 we have shown that we can expect a significant loss in accuracy if we combine different groups of homogeneous forecasts without considering the differences in the covariances between the groups. We have also shown that combining first the homogeneous pools and then combining the results can help to decrease that loss. It is therefore advantageous to identify such pools of homogeneous forecasts. We can then combine the homogeneous pools with F^{av} in a first step without a significant loss in comparison to the optimal model. This reduces the complexity of the resulting covariance matrix, which then allows the use of a more complex model in order to combine the results. The question is now how to determine and evaluate the homogeneity of pools.

Using Common Distance based Clustering Algorithms

A first option is to apply known clustering algorithms working directly on the covariance matrix or on the matrix containing the pairwise diversity as defined in Section 6.3.1.

The objective would be to identify pools containing sets of forecasts that are highly correlated (not very diverse) among each other and diverse with other pools of forecasts.

The distance ${}^{m_1,m_2}\Delta$ of any two input forecasts ${}^{m_1}\widehat{y}$ and ${}^{m_2}\widehat{y}$ needed in order to apply common distance based clustering algorithms [Witten 05] can be defined as the not correlated part of the errors

$$ {}^{m_1,m_2}\Delta = 1 - {}^{m_1,m_2}\varrho_e. \tag{6.33}$$

Determining the Choice of a Dimension for Algorithm F^{ml}

Another option is to include also information about the diversification process in applying algorithm F^{ml} and using the covariance information only in order to determine a dimension used for the first or next step of pooling. This means that

we have already different alternatives for pools available (those defined by pooling corresponding to each not yet combined dimension D of the forecast generation space). The task is then to compare these alternatives corresponding to the homogeneity of the pools. We can use the same optimality criteria as used for clustering in order to evaluate the different alternatives.

6.5.4 Generating Pools based on Expected Forecast Performance

An alternative approach is to evaluate the pools directly on the expected forecast performance or on the expected loss if not applying the optimal model.

Let us assume we have again different alternatives for potential pools given and have to decide which one to choose. Then we can estimate the homogeneity of the covariances of the pools by comparing the expected result corresponding to the optimal model and the simple average model. This helps to evaluate the loss achieved by not using homogeneous covariances. We have seen that if the covariances of a pool are completely homogeneous, the optimal model and the simple average model generate the same weights and with that a similar quality of the resulting combined forecasts. Comparing the expected error variances of the resulting forecasts helps in order to estimate the differences without considering the instabilities. The expected combined forecast error variance can be estimated corresponding to equation (4.1). For a given pool of forecasts ${}^{c}\widehat{y}$ containing M_c forecasts we can estimate the loss ${}^{c}l$ based on this equation by

$$ {}^{c}l = \frac{\frac{1}{M_c^2}\sum_{m_1,m_2 \in M_c} ({}^{m_1,m_2}\rho)}{\sum_{m_1,m_2 \in M_c} w_{m_1}^{opt} w_{m_2}^{opt} ({}^{m_1,m_2}\rho)}. \tag{6.34} $$

with weights w^{opt} determined based on the covariance matrix corresponding to pool c with the optimal model.

The total loss ${}^{D}l$ corresponding to a chosen dimension D for pooling can then be described as the sum of the loss corresponding to each pool with

$$^{D}l = \sum_{c} \frac{\frac{1}{M_c^2} \sum_{m_1,m_2 \in M_c} (^{m_1,m_2}\rho)}{\sum_{m_1,m_2 \in M_c} w_{m_1}^{opt} w_{m_2}^{opt} (^{m_1,m_2}\rho)}. \tag{6.35}$$

The dimension D with the lowest loss is chosen as next dimension for pooling.

6.5.5 How to Estimate Covariances between Results of a First Step of Pooling

The algorithm of selecting a next dimension D for pooling is optimal only in relation to one single pooling step. The loss ^{D}l can be used in order to minimise the lost quality in relation to a first combination. But it is not sure that this decision is still optimal if we consider the following combination(s) of the resulting forecasts.

If we want to take further steps into account, it is necessary to estimate the resulting covariance matrix. The quality of the forecasts resulting from a first pooling and combination with a simple average model can be approximated following equation (4.1) by

$$^{c}\delta^2 = \frac{1}{M_c^2} \sum_{m_1,m_2 \in M_c} (^{m_1,m_2}\rho). \tag{6.36}$$

It is more difficult to estimate the covariances $^{c_1,c_2}\rho$ between the resulting pools. Higher order statistics would be necessary in order to enable an exact estimation of resulting covariances. We can demonstrate this with the following example:

Example

Let us assume we have 6 forecasts $^{m1}\widehat{y}$ to $^{m6}\widehat{y}$ given with covariance matrix

$$\Sigma = \left(\begin{array}{cccc|ccc} \mathbf{2} & 0.6 & 0.6 & & 0.2 & 0.2 & 0.2 \\ 0.6 & \mathbf{2} & 0.6 & & 0.2 & 0.2 & 0.2 \\ 0.6 & 0.6 & \mathbf{2} & & 0.2 & 0.2 & 0.2 \\ \hline 0.2 & 0.2 & 0.2 & & \mathbf{2} & 0.6 & 0.6 \\ 0.2 & 0.2 & 0.2 & & 0.6 & \mathbf{2} & 0.6 \\ 0.2 & 0.2 & 0.2 & & 0.6 & 0.6 & \mathbf{2} \end{array}\right) \tag{6.37}$$

and we generate pools ${}^{c1}\widehat{y} = F^{av}({}^{m1}\widehat{y}, {}^{m2}\widehat{y}, {}^{m3}\widehat{y})$ and ${}^{c2}\widehat{y} = F^{av}({}^{m4}\widehat{y}, {}^{m5}\widehat{y}, {}^{m6}\widehat{y})$. Equation (6.36) leads to ${}^{c1}\delta^2 = {}^{c2}\delta^2 = 1.06666$. We will now show two different error decompositions of ${}^{m1}\delta^2$ to ${}^{m6}\delta^2$ leading both to the indicated covariance matrix, but to different covariances between ${}^{c1}\widehat{y}$ and ${}^{c2}\widehat{y}$. A graphical representation of the two error decompositions can be seen in Figure 52.

In order to increase readability of the error components, the components are described corresponding to the forecasts in which they occur, e.g. component ${}^{456}\delta^2$ means an error component occurring in forecasts ${}^{m4}\widehat{y}, {}^{m5}\widehat{y}$ and ${}^{m6}\widehat{y}$. In the first error decomposition we assume independent error components

$$\begin{aligned}
{}^{m1}\delta^2 &= {}^{123456}\delta^2 + {}^{123}\delta^2 + {}^{1}\delta^2 \\
{}^{m2}\delta^2 &= {}^{123456}\delta^2 + {}^{123}\delta^2 + {}^{2}\delta^2 \\
{}^{m3}\delta^2 &= {}^{123456}\delta^2 + {}^{123}\delta^2 + {}^{3}\delta^2 \\
{}^{m4}\delta^2 &= {}^{123456}\delta^2 + {}^{456}\delta^2 + {}^{4}\delta^2 \\
{}^{m5}\delta^2 &= {}^{123456}\delta^2 + {}^{456}\delta^2 + {}^{5}\delta^2 \\
{}^{m6}\delta^2 &= {}^{123456}\delta^2 + {}^{456}\delta^2 + {}^{6}\delta^2
\end{aligned} \tag{6.38}$$

with ${}^{123456}\delta^2 = 0.2$ representing a common part existing in each forecast (like the error Bayes component), ${}^{123}\delta^2 = {}^{456}\delta^2 = 0.4$ representing common parts per pool and ${}^{1}\delta^2 = \ldots = {}^{6}\delta^2 = 1.4$ representing unique components in relation to each forecast. The combination leads to

$${}^{c1}\delta^2 = {}^{123456}\delta^2 + {}^{123}\delta^2 + \frac{1}{9} * ({}^{1}\delta^2 + {}^{2}\delta^2 + {}^{3}\delta^2) \tag{6.39}$$

$${}^{c2}\delta^2 = {}^{123456}\delta^2 + {}^{456}\delta^2 + \frac{1}{9} * ({}^{4}\delta^2 + {}^{5}\delta^2 + {}^{6}\delta^2) \tag{6.40}$$

Because of the independence of the components we achieve a covariance

$${}^{c1,c2}\rho = {}^{123456}\delta^2 = 0.2. \tag{6.41}$$

The second error decomposition assumes a more irregular distribution of the error components in relation to the pools. Now we assume an error decomposition of

$$
\begin{aligned}
{}^{m1}\delta^2 &= {}^{124}\delta^2 + {}^{125}\delta^2 + {}^{126}\delta^2 + {}^{13}\delta^2 + {}^{1}\delta^2 \qquad (6.42)\\
{}^{m2}\delta^2 &= {}^{124}\delta^2 + {}^{125}\delta^2 + {}^{126}\delta^2 + {}^{23}\delta^2 + {}^{2}\delta^2\\
{}^{m3}\delta^2 &= {}^{13}\delta^2 + {}^{23}\delta^2 + {}^{34}\delta^2 + {}^{35}\delta^2 + {}^{36}\delta^2 + {}^{3}\delta^2\\
{}^{m4}\delta^2 &= {}^{124}\delta^2 + {}^{456}\delta^2 + {}^{34}\delta^2 + {}^{4}\delta^2\\
{}^{m5}\delta^2 &= {}^{125}\delta^2 + {}^{456}\delta^2 + {}^{35}\delta^2 + {}^{5}\delta^2\\
{}^{m6}\delta^2 &= {}^{126}\delta^2 + {}^{456}\delta^2 + {}^{36}\delta^2 + {}^{6}\delta^2
\end{aligned}
$$

with ${}^{456}\delta^2 = {}^{13}\delta^2 = {}^{23}\delta^2 = 0.6$, ${}^{1}\delta^2 = {}^{2}\delta^2 = 0.8$, ${}^{4}\delta^2 = {}^{5}\delta^2 = {}^{6}\delta^2 = 1$ and all other components ${}^{*}\delta^2 = 0.2$. Corresponding to this decomposition we achieve

$$
\begin{aligned}
{}^{c1}\delta^2 &= \frac{4}{9}({}^{124}\delta^2 + {}^{125}\delta^2 + {}^{126}\delta^2 + {}^{13}\delta^2 + {}^{23}\delta^2)\\
&\quad + \frac{1}{9}({}^{34}\delta^2 + {}^{35}\delta^2 + {}^{36}\delta^2 + {}^{1}\delta^2 + {}^{2}\delta^2 + {}^{3}\delta^2)) \qquad (6.43)\\
{}^{c2}\delta^2 &= {}^{456}\delta^2 + \frac{1}{9}({}^{124}\delta^2 + {}^{125}\delta^2 + {}^{126}\delta^2\\
&\quad + {}^{34}\delta^2 + {}^{35}\delta^2 + {}^{36}\delta^2 + {}^{4}\delta^2 + {}^{5}\delta^2 + {}^{6}\delta^2)). \qquad (6.44)
\end{aligned}
$$

and covariance

$$
\begin{aligned}
{}^{c1,c2}\rho &= \frac{1}{9}({}^{124}\delta^2 + {}^{125}\delta^2 + {}^{126}\delta^2 + {}^{34}\delta^2 + {}^{35}\delta^2 + {}^{36}\delta^2)\\
&= \frac{6}{9} * 0.2 \approx 0.13. \qquad (6.45)
\end{aligned}
$$

Estimation of the Covariances

The example shows that it is not sufficient to have the covariances of the original input forecasts in order to estimate the covariances between the pools. But if we make certain assumptions about the relation between the error components we can

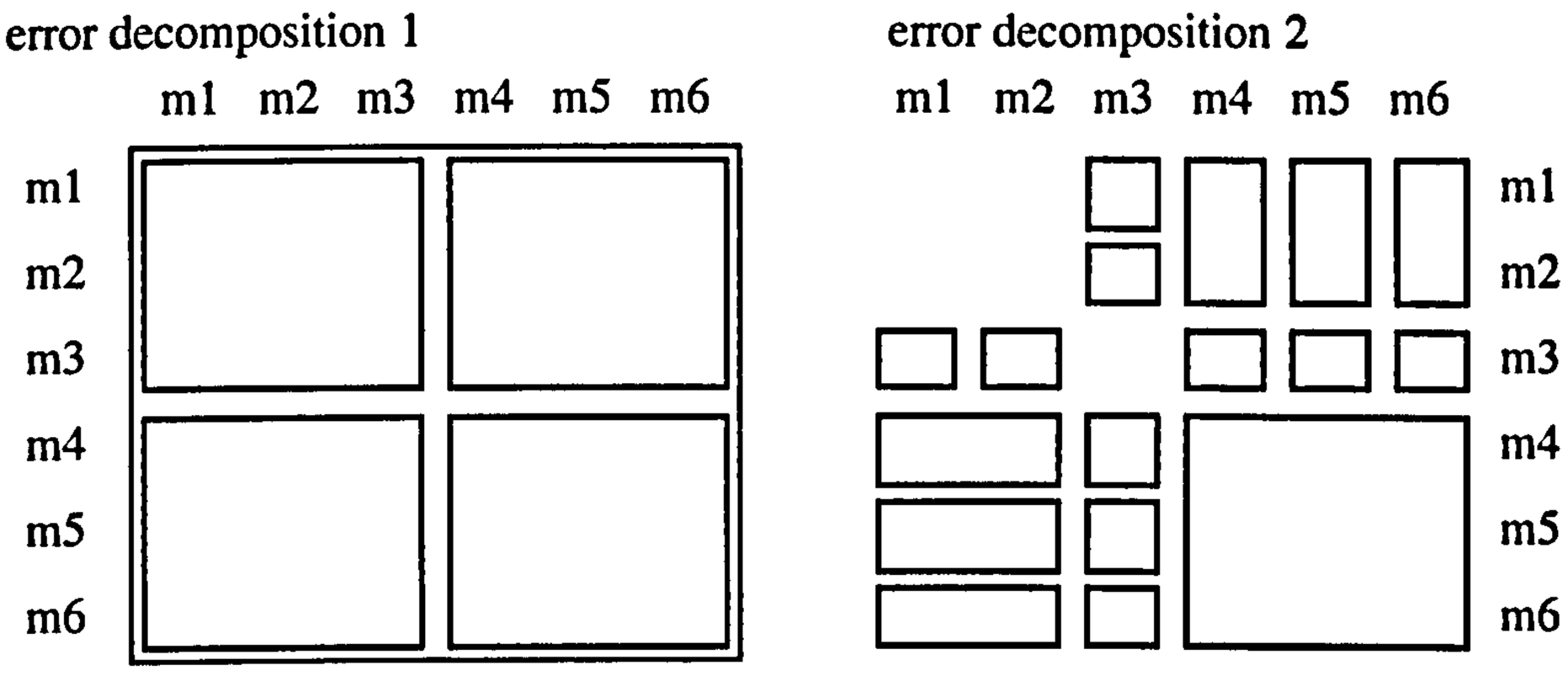

Fig. 52: Graphical representation of the two error decompositions. The frames indicate common error parts. Error components ${}^1\delta^2$ to ${}^6\delta^2$ which indicate unique parts of each of the forecasts in both decompositions are not contained in this visualisation.

produce an adequate estimation.

If we assume that we have generated the forecasts by at least two different types of diversification and different error components are concerned, we can assume that

- we have a large common part representing the error Bayes component δ_y^2
- in addition to the error Bayes component we have a common part per pool representing the error component that has not been diversified by this diversification ${}^c\delta_y^2$

All other error covariance parts can be interpreted as pairwise covariances. This means that all forecast errors can be decomposed into

$$ {}^m\delta_e^2 = \delta_y^2 + \delta_c^2 + \delta_m^2 \tag{6.46} $$

The first component is a component that all forecasts have in common including the error Bayes component. It can be estimated with

$$ \widehat{\delta}_y^2 = min(\Sigma) \tag{6.47} $$

The second component is a common component of the pool c. It can be estimated

with

$$\widehat{\delta}_c^2 = min({}^c\Sigma) - min(\Sigma) \tag{6.48}$$

with ${}^c\Sigma$ representing the covariance matrix corresponding to pool c. The remaining components δ_m^2 are error components representing a unique behaviour of each single forecasts. They are calculated with

$$\delta_m^2 =^m \delta_e^2 \quad - \quad \delta_y^2 - \delta_c^2. \tag{6.49}$$

The correlation of two forecasts of the same pool can be expressed by

$${}^{m1,m2}\rho_e^2 \quad = \quad \delta_y^2 + \delta_c^2 + \rho_{m1,m2} \tag{6.50}$$

with $\rho_{m1,m2}$ expressing the unique (not accounted for elsewhere) common parts between ${}^{m1}\delta_m^2$ and ${}^{m2}\delta_m^2$. Forecasts of different pools are correlated with

$${}^{m1,m2}\rho_e^2 \quad = \quad \delta_y^2 + \rho_{m1,m2}. \tag{6.51}$$

All elements $\rho_{m1,m2}$ are assumed to be independent of each other.

Corresponding to this decomposition, for two pools $c1, c2$ with size M_1, M_2 we achieve error covariances of

$$\begin{aligned} {}^{c1,c2}\rho_e &= \delta_y^2 + \frac{1}{M_1 M_2} \sum_{m_1 \in c_1, m_2 \in c2} \rho_{m1,m2} && (6.52) \\ &= min(\Sigma) + \frac{1}{M_1 M_2} \sum_{m_1 \in c_1, m_2 \in c2} \left[{}^{m1,m2}\rho - min(\Sigma)\right] \\ &= \frac{1}{M_1 M_2} \sum_{m_1 \in c_1, m_2 \in c2} \left[{}^{m1,m2}\rho\right] && (6.53) \end{aligned}$$

Equation (6.53) shows that the size of the assumed Bayes component does not occur in the final representation of the estimated covariances, which means that it does not matter if we interpret that part of the demand as common Bayes

component or as pairwise covariance parts. The estimation simply represents the average value of the covariances between the elements of different pools, it is a simple estimation that can easily be carried out on a given covariance matrix.

This approximation allows the calculation of complete resulting covariances of a step of pooling. Using again equation (6.36) on the resulting covariance matrix allows the estimation of the quality of the combined pools or even more than one further step of pooling. The choice of the dimension used in the next step of pooling can then be made directly on the basis of the estimated total error variance of the final combined forecast.

6.6 Trimming Versus Pooling

Trimming and Pooling are both approaches which can be used in order to generate a smaller set of predictions for a next step of forecast fusion. A question which one of the approaches to use is not easy to answer. Instead of averaging a set of forecasts representing a pool we can choose a single representative forecast in order to represent the pool. The proposed algorithms F^{cew} and F^{mlp} include steps of trimming, the relevance of these steps depends on the decision of how much to trim. Theoretically, it is possible to use trimming in algorithms F^{cew} and F^{mlp} in such an excessive manner that only one best forecast is remaining per pool.

In this section we compare the two approaches and discuss them in relation to the different types of diversification. We start with a short summary of advantages and risks of the two approaches and analyse then different diversification procedures in the bias- variance- Bayes error decomposition framework.

6.6.1 Advantages and Risks of Pooling and Trimming

The comparison of choosing the "best" forecast per pool versus a simple average or forecast error variance based combination is comparable with a discussion of under which conditions forecast combination can beat a best forecast [Timmermann 05].

Trimming contains the risk that we do not use potentially unique information

provided in the non selected forecasts. On the other hand, it is a stable procedure and we do not run the risk of weight estimation errors.

Pooling often leads to results which outperform the best forecast, but high weight estimation errors can also lead to unstable combined predictions. If we use the simple average combination, there is additionally the risk of over interpretation of forecasts with bad quality. Let us assume we have included M forecasts with different total error variance terms into a pool. Equation (4.1) can be represented as

$$^{comb}\delta^2 = \frac{(M-1)^2}{(M)^2}(^{comb,M-1}\delta^2) + \frac{1}{(M)^2}(^{m1}\delta^2 + 2 * \sum_{m \neq m1}(^{m,m1}\rho)) \quad (6.54)$$

in order to show the impact of any single input forecast m_1 with $^{comb,M-1}\delta^2$ the resulting forecast error achieved if not including m_1. We achieve $^{comb}\delta^2 <^{comb,M-1} \delta^2$ if

$$(2M-1)(^{comb,M-1}\delta^2) >^{m1} \delta^2 + 2 * \sum_{m \neq m1}(^{m,m1}\rho) \quad (6.55)$$

This representation shows that if a forecast with a larger forecast error contains unique information represented by low error covariance terms, it can be beneficial to keep this forecast in the pool. In case of no unique information we should trim this forecast.

6.6.2 Trimming versus Pooling in Connection with Thick Modelling

Let us now consider the case that we have diversified a parameter corresponding to the idea of thick modelling. In Section 4.3 we have already analysed effects of different types of parameters on the error components and achieved covariances. We will now use this information in order to analyse how to behave with regard to trimming for the different cases.

Parameters Affecting the Data Selected for Learning

In Section 4.3.1 we have argued that in the case of parameters effecting the data selected for learning the concrete parameter values ϕ_α just effect the error variance component ${}^{\mathcal{H}_\alpha}\delta^2_\phi$. We can expect a set of forecasts with about the same quality each containing unique information. In this case, we should not apply any trimming.

Parameters Affecting the Function Space without Changing the Complexity

Section 4.3.2 contains the discussion in relation to parameters affecting the function space. We have illustrated in Figure 24 that in this case we potentially also have choices of function spaces included which are suboptimal compared to others and do not contain relevant unique information. These extreme values should be trimmed. Figure 53 shows the proposed approach.

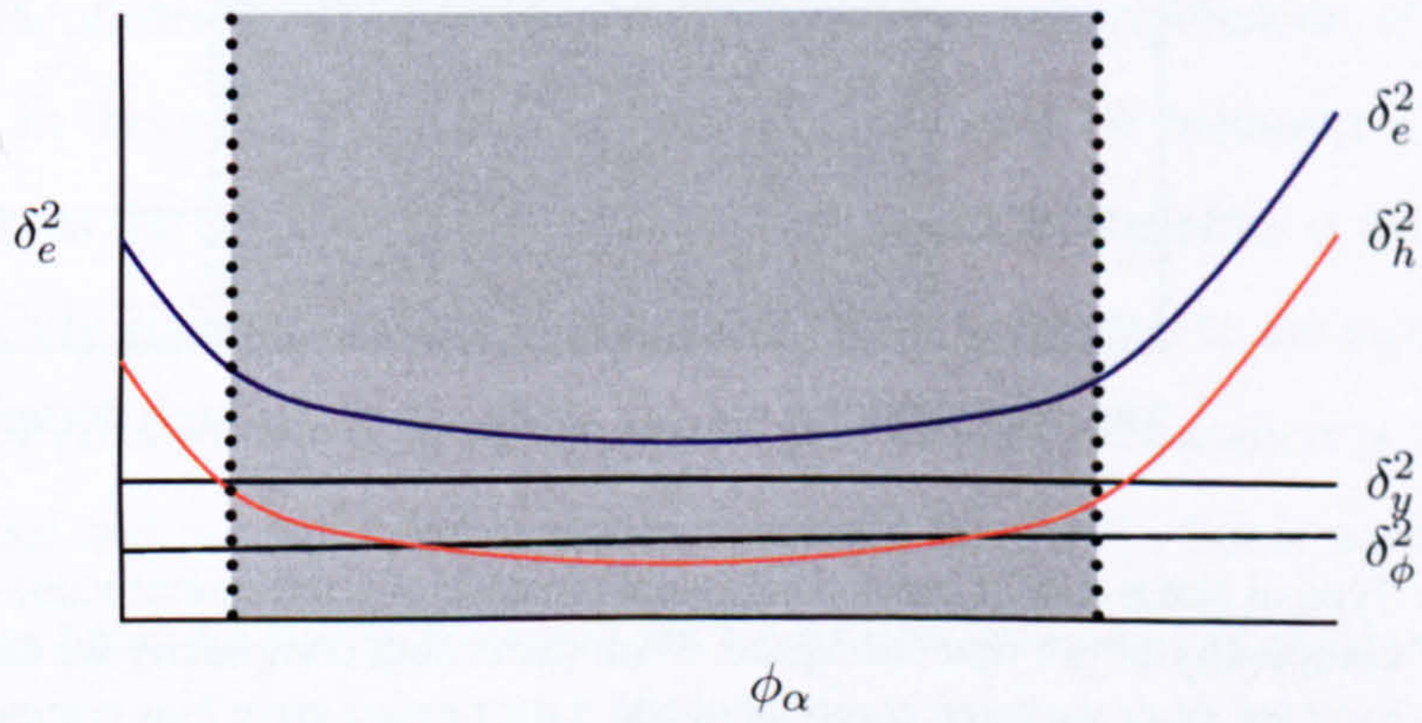

Fig. 53: Typical behaviour of error components in case of a parameter value affecting the error bias component. Extreme values cause an increasing error bias component. The error variance component is only slightly effected. As the extreme values cause forecasts which do not contain much unique information and are characterised by a high total forecast error, these forecasts should be trimmed in advance.

Parameters Affecting the Complexity of the Function Space

In Section 4.3.3 we have analysed parameters effecting the complexity of the function space and argued that both, error bias and error variance term are concerned.

Also in this case extreme parameter values lead to higher total forecast errors and we have concluded that the extreme values do not contain any additional information compared to the more stable neighboured values. Extreme values can therefore be trimmed directly based on total error variance information. It depends on the amount of decrease of the error bias component in comparison to the increase of the error variance component as illustrated in Figure 27 if this approach leads to trimming of only a small number of extreme values or if a large amount of parameter values can be trimmed. Figure 54 shows two examples of behaviour resulting in different total error curves and with that a different number of forecasts to be trimmed.

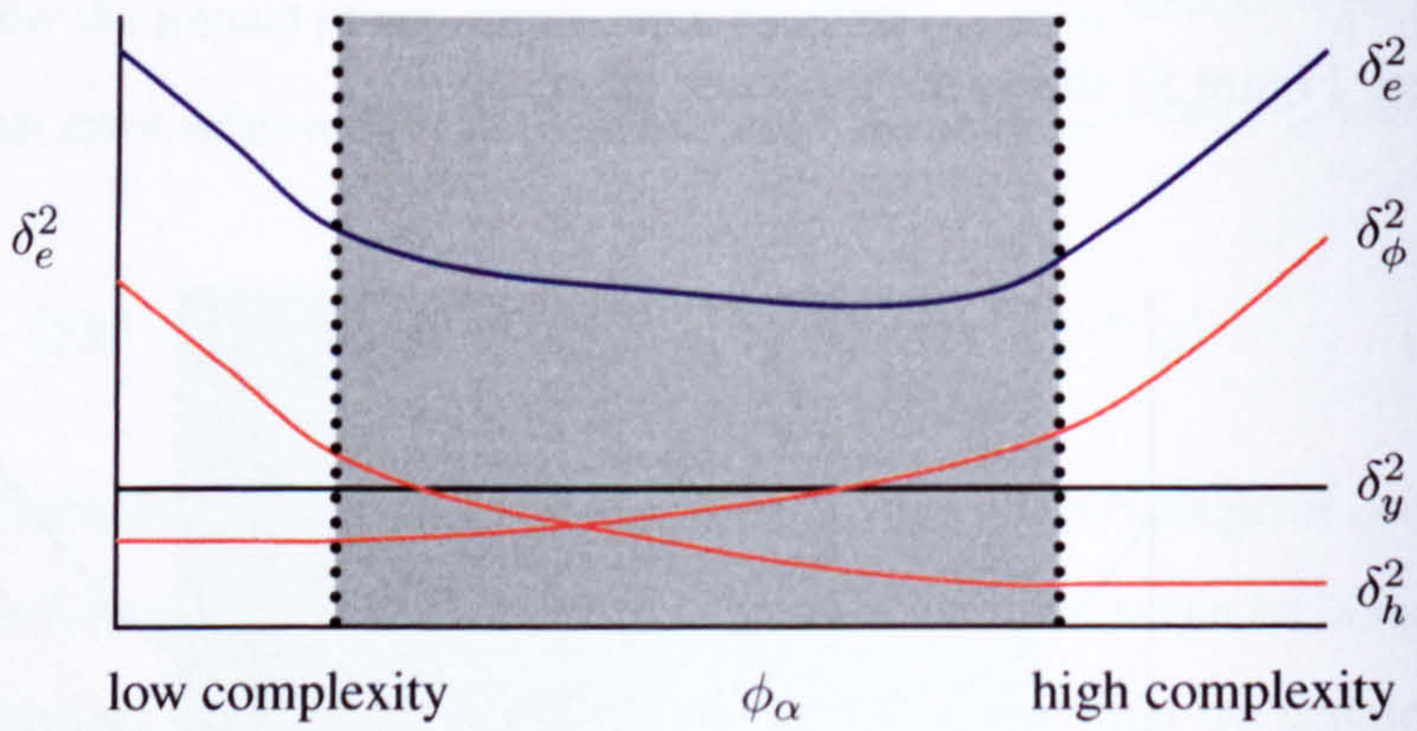

Fig. 54: Typical behaviour of error components in case of a parameter value effecting the complexity of the function space. With increasing complexity we can observe an increase error variance component and a decreasing error bias component.

Summary

Summarising it can be said that even if different types of parameters suggest different kinds of trimming, all kinds of proposed trimming can be covered by carrying out a total error variance based trimming.

6.6.3 Trimming versus Pooling in Connection with Multi Level Learning

The analysis provided in Section 5.5 for the different cases of behaviour at the different levels allows conclusions about the usefulness of trimming. While the situation in cases 3, 4 and 5 is clear and trimming would not be beneficial but would also not have any negative effect we would achieve negative effects with trimming in cases 1, 2 and 6. Especially case 6, which is the most interesting and common case, is critical for trimming. In this case the forecasts calculated at the different levels contain relevant diverse knowledge and should be included in a combination procedure even if they differ slightly concerning the total error variance term.

It can therefore be suggested for this type of diversification not to carry out any trimming or to apply only a very moderate trimming.

6.6.4 Trimming versus Pooling in Connection with Different Function Spaces

The most difficult decision about trimming is related to a diversification of function spaces. In this case, the benefit of trimming can only be evaluated if more knowledge about the diversity of the used function spaces is available. If the function spaces really generate diverse forecast error terms in relation to the error bias as well as the error variance term, the use of this diverse information is potentially beneficial and we will achieve better combined forecasts without trimming. If on the other hand the function spaces have common subspaces or even contain each other (e.g. using polynomials with different degrees), we can have cases with highly correlated forecast errors in which trimming is beneficial.

A theoretically funded decision about trimming in the case of using different function spaces can only be made on the basis of reliable covariance information.

An algorithm describing how to carry out trimming in this case will be provided in Section 7.1.3 of the next chapter.

6.7 Experiments

6.7.1 Description of Experiments

The experiments have been carried out in order to compare the different approaches of pooling. We compare the approach of Aiolfi and Timmermann as described in Section 6.2.1 with pooling based on the distance in the forecast generation space in connection with different types of trimming.

The applied set of input forecasts corresponds to the one used in Chapter 5 and described in Table 14. The forecast generation space is therefore composed of the following diversification dimensions:

- D1: parameter diversification of ϕ_{low} and ϕ_{high}
- D2: diversification of models $h_1^{season}(x, \phi)$ (historical model 2.15) and $h_3^{season}(x, \phi)$ (multiplicative model 2.18)
- D3: level diversification from fareclass to compartment
- D4: level diversification, aggregation over all days of the week

Table 18 summarises different experimentally compared pooling approaches. Only the best performing multi level pooling structures are contained in the table.

approach	description	see ...
FLAT	flat combination with only a weak trimming (10 best)	5.7.1
FLAT5	flat combination using always the best 5 forecasts	5.7.1
CEW	pooling approach of Aiolfi and Timmermann	6.2.1
MLP1	multi level pooling with order D1, D2, D3, D4	6.4
MLP2	multi level pooling with order D2, D1, D3, D4	6.4
MLP3	multi level pooling with order D3, D4, D1, D2	6.4
MLP4	multi level pooling with order D3, D4, D2, D1	6.4
MLP5	multi level pooling with order D1, D3, D4, D2	6.4
MLP6	multi level pooling with order D2, D3, D4, D1	6.4

Tab. 18: Set of forecasts diversified concerning the function space, level of learning and parameters used for thick modelling.

Details related to the experimental setup can be found in the Appendix describing experiment 6 (B.6.6). Details related to the experimental setup concerning the approach of Aiolfi and Timmermann are contained in the Appendix describing experiment 7 (B.6.7).

6.7.2 Experimental Results

Tables 19 and 20 show the errors of the forecasts containing combined seasonal predictions as relative improvement in relation to the best individual forecast ${}^0\widehat{y}$ at the low level of forecasting (ODO F POS) and at the high level (ODO). A graphical representation of the absolute total errors achieved with different structures at the high level is shown in Figure 55.

τ	FLAT	FLAT5	CEW	MLP1	MLP2	MLP3	MLP4	MLP5	MLP6
0	0.03	0.04	0.04	0.01	0.01	0.01	0.01	0.01	0.01
1	0.03	0.03	0.03	0.02	0.02	0.02	0.02	0.02	0.03
2	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03
3	0.03	0.03	0.03	0.02	0.03	0.03	0.03	0.02	0.03
4	0.03	0.03	0.03	0.02	0.03	0.03	0.03	0.02	0.03
5	0.03	0.02	0.03	0.02	0.03	0.02	0.03	0.02	0.03
6	0.03	0.02	0.03	0.02	0.03	0.02	0.02	0.02	0.03
7	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
8	0.02	0.01	0.02	0.01	0.02	0.02	0.02	0.01	0.02
9	0.02	0.02	0.02	0.01	0.01	0.01	0.01	0.01	0.01
10	0.01	0.00	0.02	0.01	0.01	0.01	0.01	0.01	0.01
11	0.02	0.01	0.02	0.01	0.01	0.01	0.01	0.01	0.01
12	0.02	0.01	0.02	0.01	0.01	0.01	0.01	0.01	0.01
13	0.02	0.01	0.02	0.01	0.01	0.01	0.01	0.01	0.01
14	0.03	0.03	0.03	0.02	0.02	0.02	0.02	0.02	0.02
15	0.03	0.03	0.03	0.02	0.02	0.02	0.02	0.02	0.02
16	0.04	0.04	0.04	0.03	0.03	0.03	0.03	0.03	0.03
17	0.05	0.05	0.04	0.04	0.04	0.04	0.04	0.04	0.04
18	0.05	0.05	0.05	0.04	0.04	0.04	0.04	0.04	0.04
19	0.06	0.06	0.06	0.05	0.06	0.05	0.05	0.05	0.05
20	0.07	0.08	0.06	0.07	0.07	0.07	0.07	0.07	0.07
21	0.12	0.12	0.10	0.12	0.12	0.12	0.12	0.12	0.12
22	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Tab. 19: Relative improvement using forecast combination of diversified multi level predictions in comparison to the best individual forecast ${}^0\widehat{y}$ measured at the low level (ODO F POS).

All pooling approaches clearly beat the simple flat combination with different

τ	FLAT	FLAT5	CEW	MLP1	MLP2	MLP3	MLP4	MLP5	MLP6
0	0.06	0.06	0.06	0.01	0.01	0.01	0.01	0.01	0.01
1	0.06	0.07	0.06	0.09	0.09	0.09	0.09	0.09	0.09
2	0.07	0.07	0.06	0.10	0.11	0.11	0.11	0.10	0.11
3	0.07	0.06	0.07	0.10	0.11	0.10	0.11	0.10	0.11
4	0.05	0.05	0.04	0.09	0.10	0.09	0.10	0.09	0.10
5	0.04	0.03	0.03	0.08	0.08	0.08	0.08	0.08	0.08
6	0.02	0.01	0.00	0.06	0.07	0.07	0.07	0.06	0.07
7	0.01	0.00	-0.02	0.05	0.05	0.05	0.05	0.05	0.05
8	0.01	-0.01	-0.03	0.04	0.04	0.04	0.04	0.04	0.04
9	0.01	0.00	-0.02	0.04	0.04	0.04	0.04	0.04	0.04
10	0.01	-0.01	-0.01	0.04	0.03	0.03	0.03	0.04	0.03
11	0.02	0.00	-0.01	0.04	0.03	0.03	0.03	0.04	0.03
12	0.03	0.02	0.00	0.04	0.04	0.04	0.04	0.04	0.04
13	0.03	0.03	0.01	0.05	0.05	0.05	0.05	0.05	0.05
14	0.05	0.05	0.03	0.05	0.05	0.05	0.05	0.05	0.05
15	0.05	0.05	0.04	0.05	0.05	0.05	0.05	0.05	0.05
16	0.06	0.07	0.06	0.06	0.06	0.06	0.06	0.06	0.06
17	0.07	0.07	0.05	0.07	0.07	0.07	0.07	0.07	0.07
18	0.07	0.08	0.07	0.07	0.07	0.07	0.07	0.07	0.07
19	0.09	0.10	0.09	0.09	0.09	0.09	0.09	0.09	0.09
20	0.11	0.12	0.08	0.11	0.11	0.11	0.11	0.11	0.11
21	0.20	0.20	0.17	0.20	0.20	0.20	0.20	0.20	0.20
22	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Tab. 20: Relative improvement using forecast combination of diversified multi level predictions in comparison to the best individual forecast ${}^0\widehat{y}$ measured at the high level (ODO).

strengths of trimming at the high level. While at the low level an improvement of up to 3% percent could be achieved in the early dcps, a *significant improvement of up to 11% could be measured at the high level* with the multi level structures. The approach of Aiolfi and Timmermann could generate improvements of up to 7%. The larger improvement at the higher level can be explained with the extremely large error Bayes component at the low level. The noise in the data is so large at the low level that any improvement in forecasting will always have stronger effects at higher levels.

The very best results have been achieved with structure MLP6. Corresponding to this structure forecasts generated by different function spaces are combined first. The different function spaces contain the most relevant differences in error variances so that the decrease of the total error variance achieved in the first step

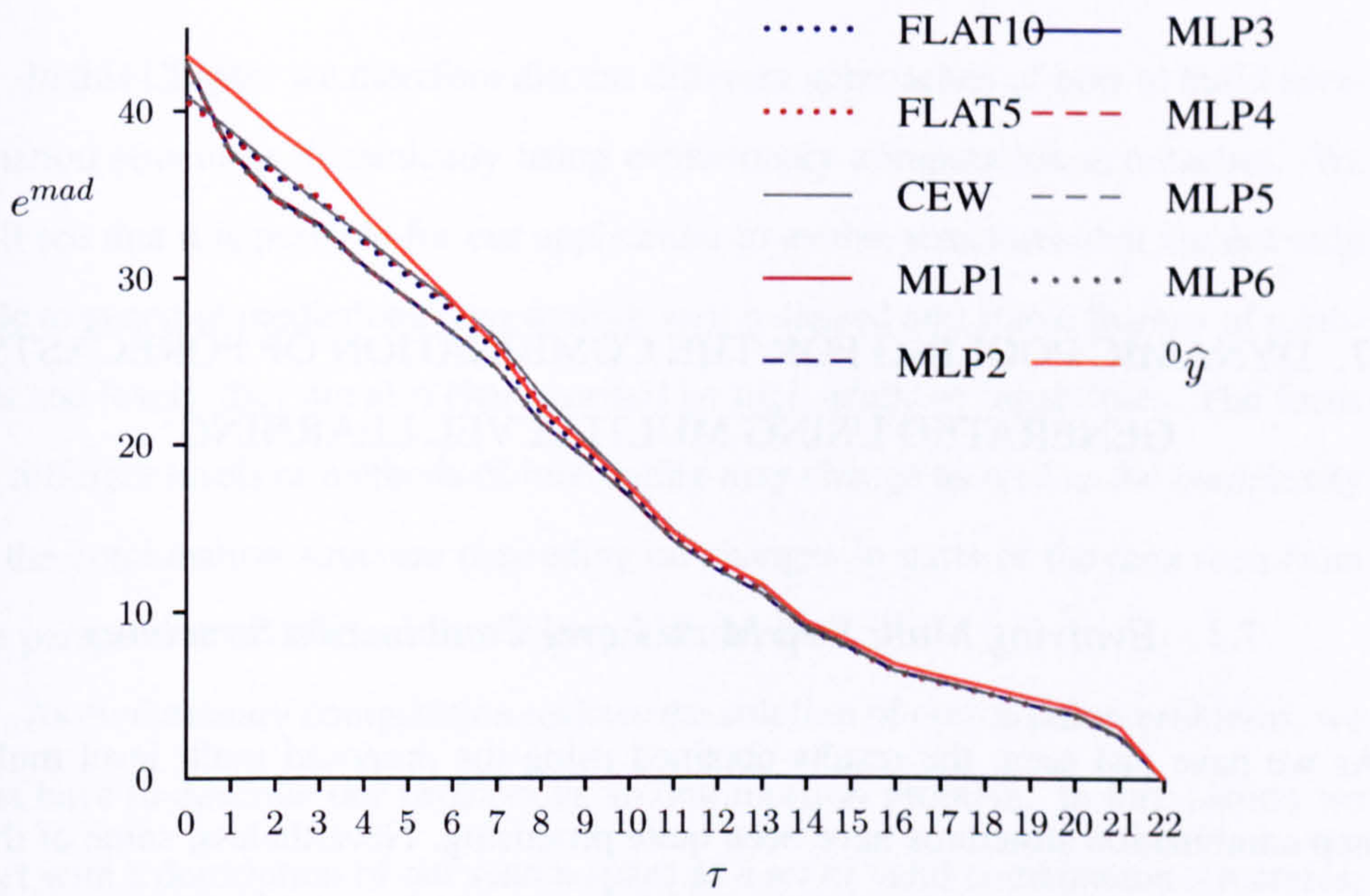

Fig. 55: Error variances achieved using forecast combination of diversified seasonal predictions in comparison to the best individual forecast $^0\widehat{y}$ measured at the high level (ODO).

is useful for the combination in later steps. However, it can be seen that the results achieved with the different structures of the "best 6" list are quite similar so that it is difficult to decide which one to chose. But it should also be mentioned that other multi level structures using a different order of the diversified dimensions than shown in this list generated less good results.

7. DYNAMIC POOLING FOR THE COMBINATION OF FORECASTS GENERATED USING MULTI LEVEL LEARNING

7.1 Evolving Multi Step Multi Level Combination Structures

As we have just seen, the results obtained using the proposed multi level multi step combination structures have been quite promising. Nevertheless, some of the structures produced better results than others. Even if the good results show that the use of multi step multi level structures may be a way to overcome the problems described in Section 6.1, the approach of using predefined structures needs a lot of expert knowledge in order to identify the most promising ones. This task is getting even harder by the fact that a lot of decisions, like the choice of parameter values used for trimming, have to be made in advance. Potential structures, once identified, have to be verified by experiments using trial and error principles. And as the fixed structures contain only limited adaptive capabilities, they would have to be rebuilt on a regular basis.

The best structures do not necessarily need to be the intuitive ones. Additionally, we prefer the generation of structures that work well in a changing environment.

All these reasons motivate the search for dynamic approaches generating and adapting structures automatically. Evolutionary computation offers common algorithms to solve such kind of problems. It simulates evolution in applying optimisation algorithms which iteratively improve the quality of solutions until an optimal, or at least high quality solution is found. As evolution continues over time the iterative process generates solutions which have proven to be flexible in a changing environment by having survived different generations.

In this Chapter we therefore discuss different approaches of how to build combination structures dynamically using evolutionary computation approaches. We will see that it is possible for our application to evolve structures that are not only able to generate predictions representing well balanced and stable fusions of methods and levels, they are also characterised by high adaptive capabilities. The focus on different levels or methods of forecasting may change as well as the complexity of the combination structure depending on changes in parts of the data seen from the perspective of different data aggregation levels.

As evolutionary computation realises the solution of optimisation problems, we first have to describe our problem as an optimisation problem. In this section we start with a description of our search space as a set of valid combination structures, then discuss different criteria to be optimised which are based on forecast quality of the resulting combined predictions and finally provide a discussion of how to generate a restricted set of input forecasts.

7.1.1 Description of the Search Space

As we want to learn combination structures, we first have to define what we understand by a combination structure in order to describe our search space.

Definition 7.1 (Combination Structure): A combination structure is a combination function $F : \mathcal{R}^M \rightarrow \mathcal{R}$ as defined in Section 3.1 that carries out a forecast combination ${}^{comb}\widehat{y} = F(\{\widehat{y}\})$ of a set of input predictions $\{\widehat{y}\}$ by a successive application of basic known linear or nonlinear combination functions using each subsets of $\{\widehat{y}\}$ and intermediate combination results as a set of input forecasts.

Each application of a basic combination function is related to a step in the fusion process. In each step γ the set of input forecasts is a subset of $\{\widehat{y}\} \cup \{{}^{1}\widehat{y}\} \cup \ldots \cup \{{}^{\gamma-1}\widehat{y}\}$, the generated fusion results represent set $\{{}^{\gamma}\widehat{y}\}$.

Figure 56 shows an example of a combination structure.

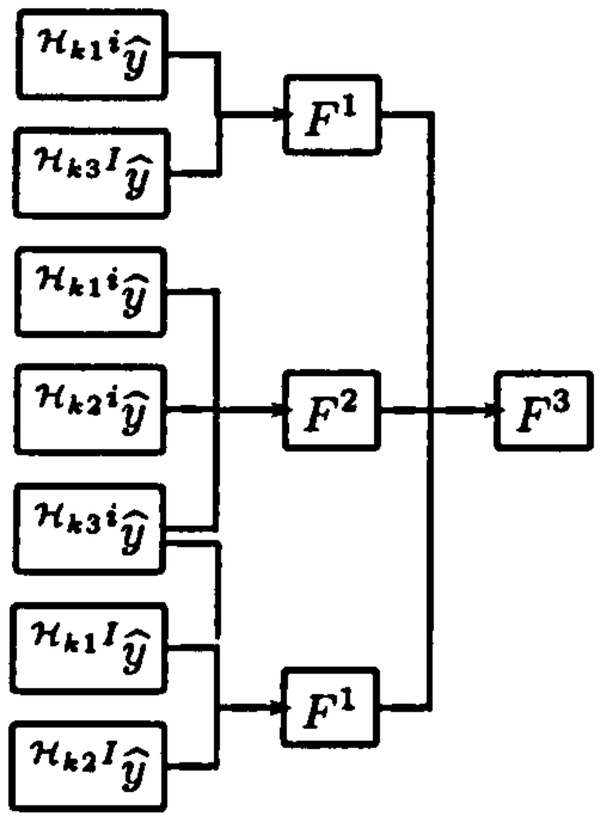

Fig. 56: An example of a combination structure. It combines multi level forecasts generated using three functional spaces $\mathcal{H}_{k1}$ to $\mathcal{H}_{k3}$ at two levels i and I. The different functions F^1 to F^3 represent three different combination methods. It can be seen that forecast ${}^{\mathcal{H}_{k3}i}\widehat{y}$ is used as input in two basic combination functions.

Differences in Comparison with Pooling

At the end of the last Chapter we already spoke about multi step combination structures in the context of pooling. It can be seen that the structure shown in Figure 56 cannot be generated with pooling, that is why we will shortly discuss the limits of structures generated by pooling in comparison to the general definition here.

Structures achieved by single step or multi step pooling correspond to the general definition, but contain two kinds of limitations. Pooling means grouping the input forecasts into clusters, which leads to disjunct sets of input forecasts. This means that in each step the resulting structures contain each input forecast only in one of the basic combinations. Additionally, the set of input forecasts in each step γ is restricted to $\{{}^{\gamma-1}\widehat{y}\}$ in comparison to set $\{\widehat{y}\}\cup\{{}^{1}\widehat{y}\}\cup\ldots\cup\{{}^{\gamma-1}\widehat{y}\}$ used in the definition above, which means that in each step we combine only the results of the preceding step without considering other predictions that are available in earlier steps.

Alternatives of Restrictions of the Search Space

Our search space corresponds to the space of combination structures that are limited by different input configurations and restrictions. Variations of search spaces

are based on different sets of input forecasts and different sets of basic combination functions. So it is possible to use only one given linear combination model, the most common approaches would be the use of F^{av}, F^{var} or F^{outp} which we have introduced in Section 3.2.2.

Additionally, it may be useful to restrict the search space in order to avoid too complex structures leading to overfitting. Those restrictions can be:

- a limitation of the (maximal) number of steps γ
- limitations to the number of input forecasts (total and/or in each basic combination)
- a limitation to disjunct sets of input forecasts (as carried out for pooling)
- a limitation to the set of input forecasts to $\{^{\gamma-1}\widehat{y}\}$ in each step γ (as carried out for pooling)
- limitations concerning the maximal variance ratio of the input forecasts (total and/or in each basic combination) correponding to the idea of trimming

The most restricted version starts with a fixed maximal number of disjunct input forecasts $M^{max} \in \mathcal{N}$ for each combination at step 1. It is assumed that the number of steps is limited to $\gamma^{max} = 2$ and that the second step consists of a combination combining all results of the first step. The set of applied basic combination models is restricted to one predefined model F.

7.1.2 Definition of the Optimum Criterium and Fitness

The most simple and intuitive criterion to optimise is defined by the accuracy of the resulting forecasts. We want to learn combination structures which generate high quality combined predictions measured on unseen data. The fitness is calculated as a mean absolute deviation value on the level of forecasting and is given as

$$\zeta^{mad} = E(|^{comb}e|) \tag{7.1}$$

or as the error variance

$$\zeta^{var} =^{comb} \delta_e^2 \tag{7.2}$$

measured over a given evaluation period.

Level of Error Measurement

As discussed in the previous Chapter, the main objective is to achieve good predictions at the low level of forecasting, which means a minimisation of ${}^{comb}\delta_{ei}^2$ and learn a separate structure for each subspace i, in our example each ODO-DOW-F-POS combination. But as the generated forecasts are also used on an aggregated level, it is also worth analysing the error ${}^{comb\cup}\delta_{eI}^2$.

In Section 5.4.5 we have seen that combining multi level predictions with weights purely based on errors measured at the low level can also have positive effects on the errors of the high level aggregates measured at the high level.

If on the other hand we would learn a combination structure for a low level subspace i and want to measure the fitness at a higher level I, which forecasts of the other subspaces should be used for aggregation? We have only the options (a) to use the same learned structure for all elements of lower level subspaces i of I or (b) to learn different structures as an integrated process.

Penalty Terms

Additional needs or constraints with regard to the resulting combination structures can be modelled as penalty terms of the fitness function. So it is, e.g., possible to generate unbiased results by adding a penalty term in relation to the systematic error. A fitness function avoiding systematic errors can be represented as

$$\zeta^{var,syst} =^{comb} \delta_e^2 + w * E({}^{comb}e) \tag{7.3}$$

where w represents a predefined weight that describes the relation between total error and systematic error relevance.

As we are interested in small and stable structures, penalty terms corresponding to the complexity of the structure, the independence of the included combination procedures or the number of multiple applications of input forecasts are possible as well. An unfavourable characteristic of unnecessarily complex structures is that different basic combinations use similar inputs. This can be avoided by including penalty terms representing measurements of diversity of the set of input forecasts of two combinations.

A first option is to use measures similar to the measures for the diversity of classifiers which have been described in Section 4.1.1. These measures do not take into account the correlation between forecasts.

We can use

$$\zeta^{var,div} =^{comb} \delta_e^2 + w * \sum_{j_1,j_2} \sum_{m} \lambda_{j_1,m} * \lambda_{j_2,m} \tag{7.4}$$

with j is an index over the basic combinations and $\lambda_{j,m} = 1$ if input forecast ${}^{m}\widehat{y} \in \{{}^{\gamma-1}\widehat{y}\}$ is included in the basic combination j at step γ and $\lambda_{j,m} = 0$ otherwise.

We can also include the correlation between each pair of input forecasts,

$$\zeta^{var,corr} =^{comb} \delta_e^2 + w * \sum_{j_1,j_2} \sum_{m_1,m_2} \lambda_{j_1,m_1} * \lambda_{j_2,m_2} * \varrho_{m_1,m_2} \tag{7.5}$$

with ϱ_{m_1,m_2} the correlation coefficient between forecast errors ${}^{m_1}e$ and ${}^{m_2}e$.

Alternatives to Forecast Error Based Fitness

We have just modelled the fitness function in a manner that it represents the quality of the resulting forecasts in terms of a mean absolute forecast deviation value and potentially including other information and/or penalty terms. This definition makes the evaluation of the fitness function expensive in terms of performance, because it may contain a new calculation of the weights or parameters of all com-

binations included in a combination structure as well as a determination of error and correlation terms on the testbed. In Section 6.5.5 we have described how the covariance between results of pooling can be estimated. A fitness function evaluating the forecast error resulting from a given combination structure can be replaced by an approximation of this error based on equation (4.1) in connection with the algorithm for covariance estimation presented in Section 6.5.5. This approach generates less reliable fitness values (estimations in comparison to real measurements). But as the combined forecasts do not have to be calculated, the evolution can be carried out with dramatically reduced calculation time.

Another issue related to pure total error based fitness is that aspects related to different levels in relation to the error components as discussed in Chapter 5 are not sufficiently taken into account. So it is possible that a structure in which we have removed all high level predictions produces good results over a certain period of time. But if suddenly demand is shifted between subspaces i, the structure is not optimal any more. As all high level predictions have been removed, an adaptation of combination weights enforcing the high level predictions is not possible. Alternative definitions of the fitness can be directly based on the variance or covariance matrix of the input predictions. Simplified versions similar to the one described in Chapter 6 or different approaches of variance/covariance estimation in a changing environment can increase the stability of the estimations.

7.1.3 Input forecast selection

We have already mentioned that the number of potential input forecasts can get very large, especially if generated by different types of diversification. That is why it is sometimes not useful to include all of them into a combination process. If we apply, e.g., the approach of thick modelling there is the question of how many different parameter values to start with. This decision is not only relevant for the generation of the structures, but determines also the performance of the input forecast generation process.

If information about forecast generation is not used, the input forecast selection can be interpreted as a mapping from the multidimensional forecasts generation space $\mathcal{S}$ into a series of selected inputs $({}^{m}\widehat{y})$, $m = (1, \ldots M) \subset \mathcal{M}$ the index set representing the selected predictions. The approach is illustrated in Figure 57.

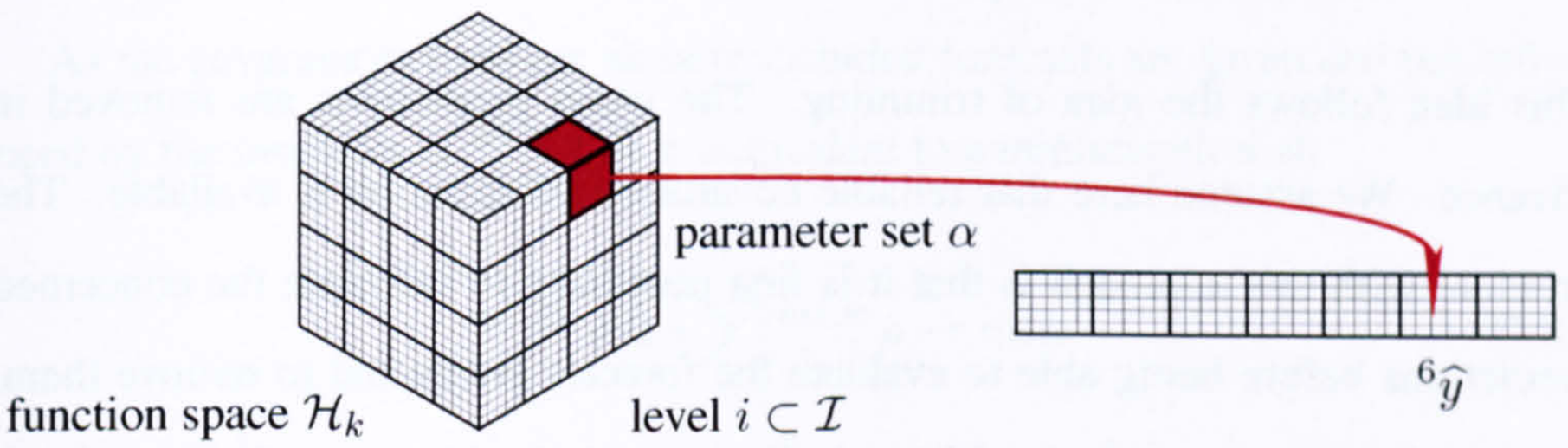

Fig. 57: Selection of the 6th input forecast. The multidimensional individual forecasts generation space $\mathcal{S}$ is, in this example, characterised by one dimension representing the function space $\mathcal{H}_k$, one dimension representing the level and one dimension representing parameter values used for thick modelling.

There are different options of how to handle this problem.

Random or Expert Selection

The first option is to choose a representative set randomly or by expert selection. This is the easiest option but carries the risk that relevant forecasts are not selected.

Selection Considering the Diversification Process

In this option we choose some predictions for each type/ combination of diversification. So we select, e.g., a few representative values of parameters applied for thick modelling. The set should by chosen small in a manner that it covers the complete forecast generation space well. It can be expected that one does not loose too much relevant information by including only those forecasts into the evolution, because forecasts differing in only one dimension, e.g. only by small parameter changes, are often highly correlated so that the information loss is not critical.

A selection can be performed by selecting the values separately for each dimension. Each value representing a subset of a dimension of $\mathcal{D}$ is selected a certain

number of times so that in total K values are selected. Then a series of the selected values is generated using a random ordering. This series describes then the value of the input forecasts to select concerning this dimension.

Selection Considering Error Variance / Covariance Information

This idea follows the idea of trimming. The worst predictions are removed in advance. We assume here that reliable covariance information is available. The problem with this approach is that it is first necessary to calculate the concerned predictions before being able to evaluate the forecast errors and to remove them. That is why this approach should be ideally connected with a pre-selection of calculated prediction during the diversification process.

As forecasts with a high total error can contain relevant and unique information, it is risky to apply trimming purely based on error variances. We have discussed this topic in relation to the different types of diversification in Section 6.6. A good and easy alternative is the application of trimming as proposed by Timmermann as part of a pooling procedure. First we carry out a trimming in relation to the whole set of input forecasts (corresponding to the algorithm of Timmermann we remove the worst pool). Later additional trimmings are carried out in relation to each pool.

The selection considering error covariances can be carried out as a process of successive insertion of forecasts or a process of starting with a complete set followed by covariance based trimming.

The first option is to start with a single forecast containing the best total error variance. We then successively add forecasts in a manner that as much as possible new information is provided in each step. Let us assume we have already selected a set of M forecasts $\{{}^{m}\widehat{y}\}$ and want to add a forecast ${}^{m1}\widehat{y}$. Following equation (4.1) we want to minimise

$$ {}^{comb}\delta^2 = \frac{1}{(M+1)^2}(\sum {}^{m,m1}\Sigma) \rightarrow min \tag{7.6}$$

representing the average of all elements of the new (extended) covariance matrix ${}^{m,m1}\Sigma$. This can be expressed with help of the previous covariance matrix ${}^{m}\Sigma$ by

$$ {}^{comb}\delta^2 = \frac{1}{(M+1)^2}(\sum {}^{m}\Sigma + {}^{m1}\delta^2 + 2 * \sum_{m}({}^{m,m1}\rho)) \rightarrow min \qquad (7.7) $$

As the covariances between already included forecasts are given and not influenced by the insertion of ${}^{m1}\widehat{y}$, this is equivalent to a minimisation of

$$ {}^{m1}\delta^2 + 2 * \sum_{m} {}^{m1,m}\rho \rightarrow min. \qquad (7.8) $$

Similarly we can describe a process of successive deletion of forecasts. In this case, we start with the complete set of forecasts and successively remove forecasts m_2 maximising

$$ {}^{m2}\delta^2 + 2 * \sum_{m} {}^{m2,m}\rho \rightarrow max. \qquad (7.9) $$

Evolving the Set of Input Forecasts

A completely different option is to start with a subset of input forecasts, but to extend or change this set during the evolution process. If it is learned that certain forecasts are especially relevant, it may be useful to include other forecasts which have similar characteristics. This approach has not yet been followed during the PhD and represents a promising extension of the current work.

7.2 Using Genetic Programming

We started with the most common and simple approaches which are genetic algorithms. But it became clear quite quickly that a fixed length bit-representation of the objects to evolve are not ideal in order to represent dynamic combination structures. Even if the number of input forecasts to the combination process is restricted, we could not avoid getting chromosomes with a complex structure of genes if the

size of potential steps is larger than two and more than one combination model may be used.

A more flexible representation which is perfectly fitting to the tree-like multi step combination structures is offered by the approach of genetic programming (see, e.g., [Koza 92] or [Negnevitsky 05]). A genetic program (GP) can be interpreted as a tree with ordered branches, in which each node represents the application of a primitive function on arguments passed to the node by the branches from the next lower level. The leaves represent basic arguments called terminals. The root node represents the application of the function generating the final result.

The process of the development to evolve combination structures using GPs includes the following steps (see [Negnevitsky 05]):

1. determine the set of terminals and select the set of primitive functions.

2. define a fitness function.

3. define an initial population.

4. define crossover and mutation operators.

The next subsections follow these steps.

7.2.1 Terminals and Primitive Functions

The terminals correspond to our chosen subset of the set of potential input forecasts $\{\widehat{y}\}$.

The set of primitive functions corresponds to the set $\{F\}$ of basic combination functions included into the evolution process. If we want to use only one predefined combination model, we have only one primitive function describing a basic combination.

Figure 58 shows an example for a genetic program which represents a combination structure containing more than 2 steps and more than one combination model.

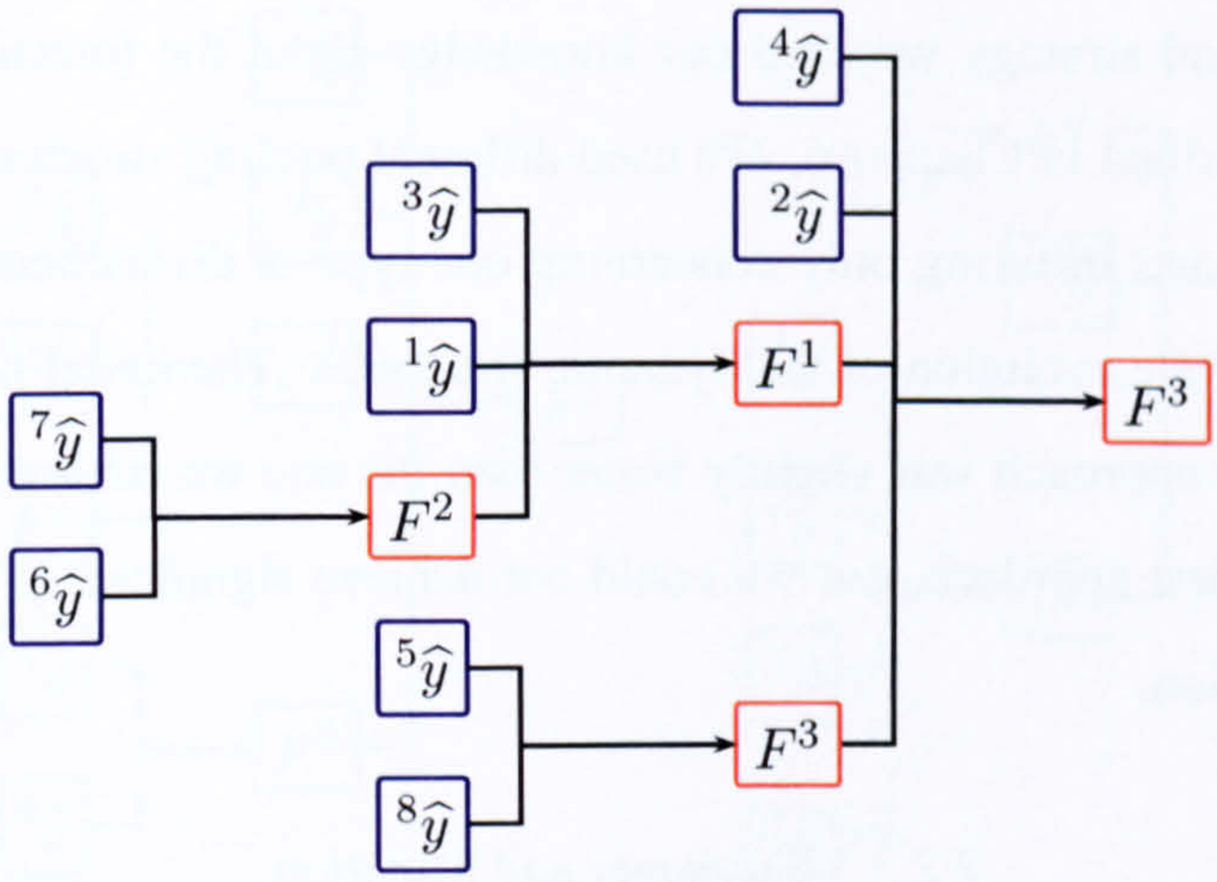

Fig. 58: Example of a genetic program with three different combination models F^1 to F^3 and selected input forecasts ${}^1\widehat{y}$ to ${}^8\widehat{y}$. The combination model is part of the description of the primitive functions. The terminals are shown in blue/dark, the primitive functions in orange/light.

7.2.2 Generation of an Initial Population

The population size is limited because of computational power and performance. As each member of the population represents a combination structure consisting of different combination procedures for which the combination weights (or other parameters if we have a nonlinear combination model) have to be learned for fitness evaluation, the population size should be as small as possible in order to be able to run the evolution quickly. On the other hand we have to assert that the space of potential solutions is well covered, at least in the domain where we can expect the optimal solution. That is the reason why it can be worth focusing on the determination of good initial populations.

We have followed two strategies which were both based on input forecast selections as described in 7.1.3.

In the first strategy we generated initial combination structures randomly only based on a few parameters, e.g. mean value and standard deviation given for the number of input forecasts for each combination procedure, the number of steps or the number of combination procedures to include per step.

In the second strategy we used our knowledge about the forecast generation process as described in Chapter 6. We used different pooling structures, each pool including forecasts differing only concerning one type of diversification as initial populations for the evolution of the dynamic structures. The initial fitness following this second approach was slightly better than the one we achieved on average following the first approach, but we could not achieve significantly better results after the evolution.

7.2.3 Crossover and Mutation

Here we can use the standard operators described e.g. in [Negnevitsky 05]. The crossover operator randomly exchanges subtrees of the two parents. For our combination structures this means that we exchange substructures or single combination procedures. For our problem the crossover operator has to be restricted in the sense that limitations of the maximal number of steps are not violated. Very stable versions of crossover allow only exchanges of subtrees representing the same step of combination. The process is shown in Figure 59 using a simple example.

The mutation operator randomly exchanges a terminal or a primitive function. Concerning the combination structures, mutation means that the combination methods are changed in the combination procedures or that input forecasts are randomly exchanged in the combination procedures of step 1 (including the possibility to add or to remove an input forecast). For an example see Figure 60.

7.2.4 Experiments

We have carried out a number of experiments in order to compare combinations based on dynamic structures of varying complexity. Table 21 summarises these structures. They differ concerning the restrictions of the search space (7.1.1) as well as concerning the definition of the fitness function (7.1.2) and the selection of input forecasts (7.1.3).

All experiments with random input forecast selection started with initial struc-

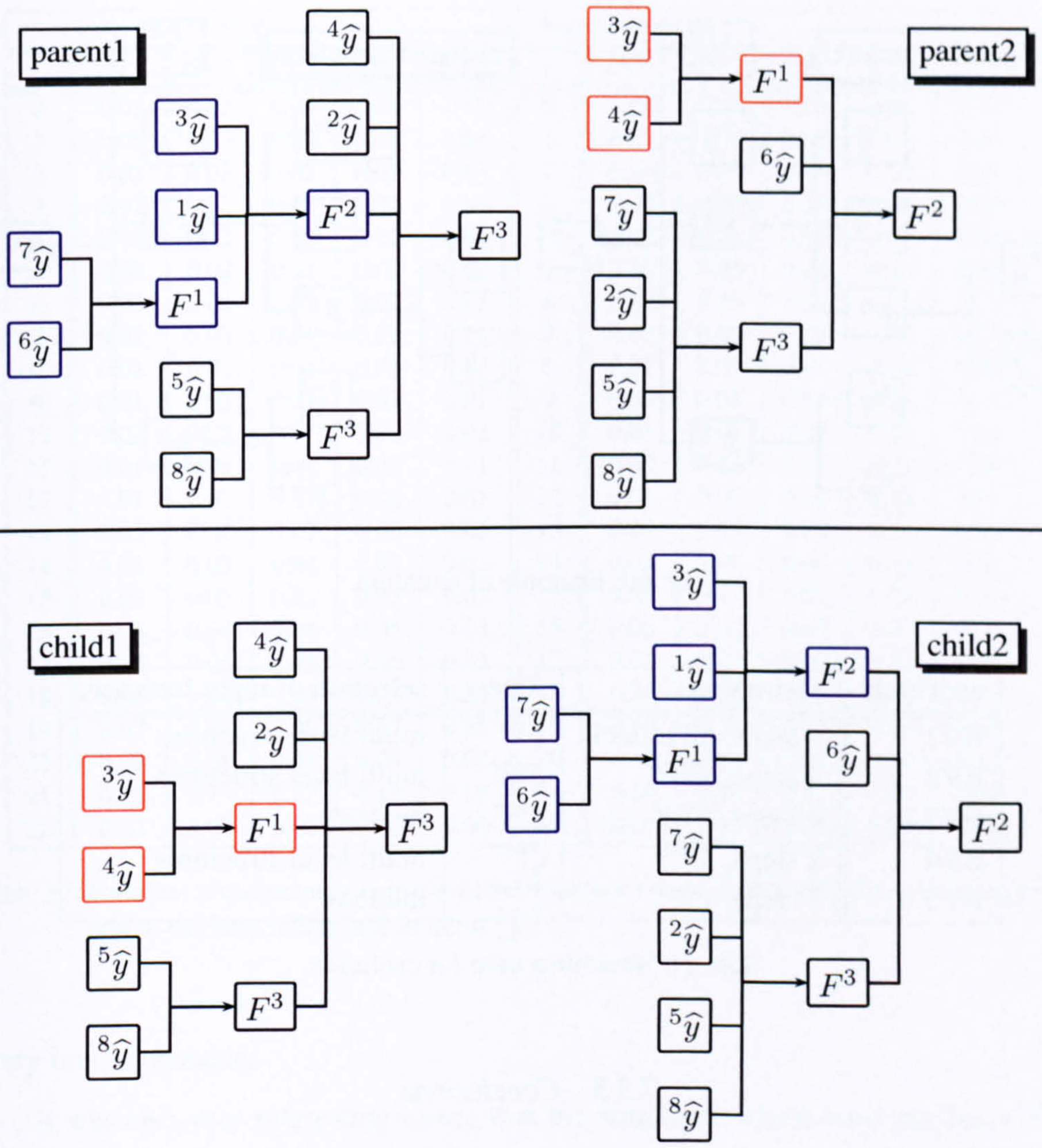

Fig. 59: Example of crossover.

tures containing two steps, a first step contained 5 combination procedures, the second step combines the results of the first step. We have used a mutation probability of 20% and a maximum number of crossover of 40.

Details related to the experimental setup can be found in the Appendix describing experiments 7 (B.6.7).

Table 22 shows the errors of the forecasts containing combined seasonal predictions as relative improvement in relation to the best individual forecast $^{0}\widehat{y}$ at the low level of forecasting (ODI F POS).

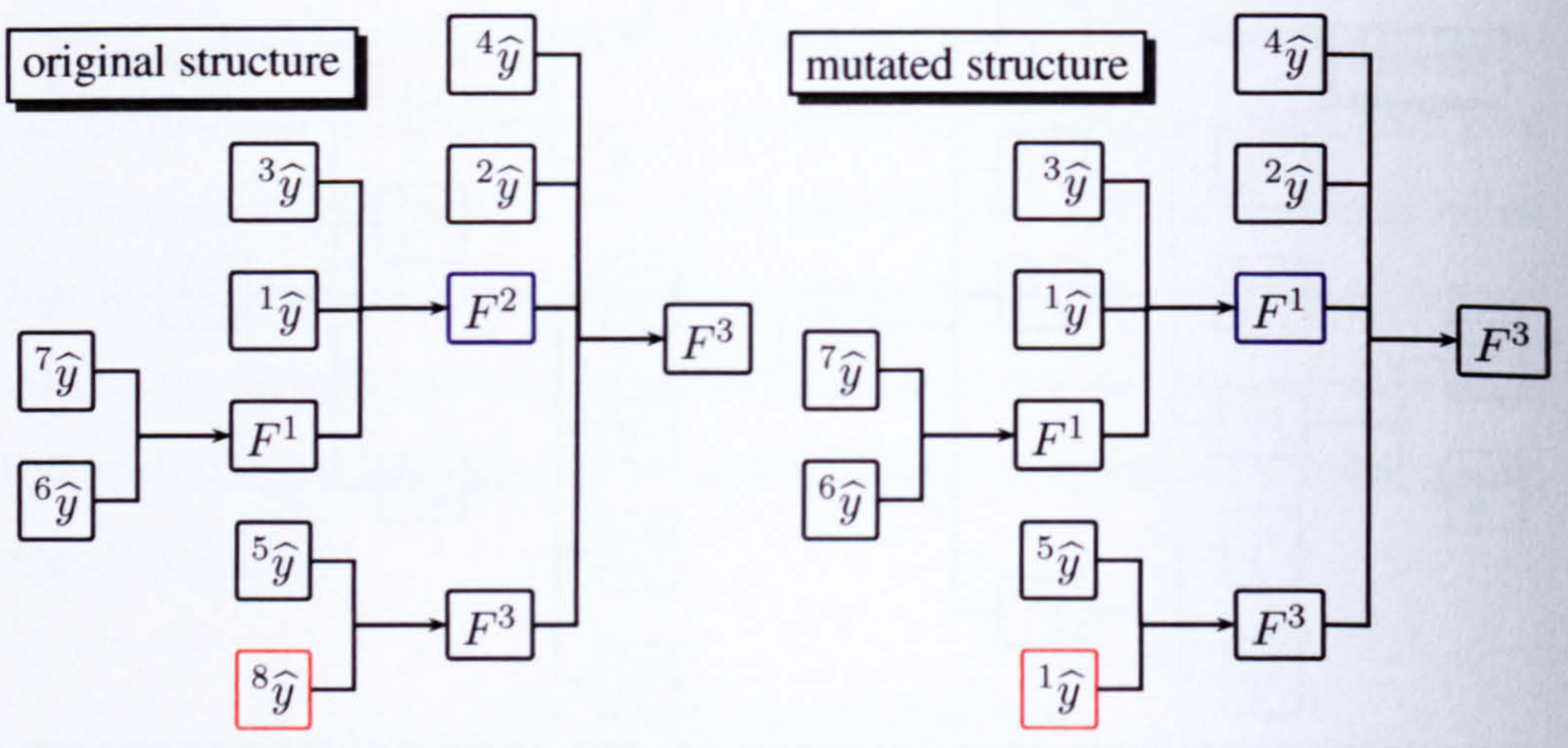

Fig. 60: Example of mutation.

approach	restrictions	fitness	selection of input forecasts
EV1	4 steps, all models	ζ^{var}	multi level structures
EV2	4 steps, F^{var}	ζ^{var}	multi level structures
EV3	4 steps, F^{var}	ζ^{var}	random
EV4	2 steps, F^{var}	ζ^{var}	multi level structures
EV5	2 steps, F^{var}	ζ^{var}	random

Tab. 21: Structures used for evolution.

7.2.5 Conclusions

The dynamic structures did not outperform the structures generated with the idea of pooling presented in the last Chapter. A detailed analysis has shown that the dynamic structures generate very diverse quality corresponding to the concrete constellation in different fareclasses and point of sales. Structures evolved using the whole set of combination models were surprisingly good in some cases, but others clearly showed problems caused by overfitting. This effect has been verified by a very simple analysis of the achieved improvements compared to the number of steps or the number of combination procedures. It clearly showed that the bigger structures achieved poor results because of missing generalisation capabilities. These structures were often learned in situations were the number of bookings was

low level ODO F POS						high level ODO					
τ	EV1	EV2	EV3	EV4	EV5	τ	EV1	EV2	EV3	EV4	EV5
0	0.03	0.03	0.01	0.02	-0.02	0	0.06	0.04	0.03	0.04	-0.10
1	0.02	0.03	0.02	0.03	0.04	1	0.07	0.08	0.07	0.08	0.06
2	0.02	0.03	0.02	0.02	0.03	2	0.09	0.08	0.08	0.09	0.07
3	0.02	0.02	0.02	0.02	0.03	3	0.09	0.08	0.08	0.08	0.07
4	0.02	0.02	0.02	0.02	0.02	4	0.07	0.07	0.06	0.07	0.06
5	0.02	0.02	0.01	0.02	0.02	5	0.06	0.05	0.04	0.05	0.02
6	0.02	0.02	0.01	0.02	0.02	6	0.03	0.03	0.02	0.03	0.02
7	0.02	0.02	0.01	0.02	0.01	7	0.02	0.02	0.02	0.03	0.01
8	0.02	0.02	0.01	0.02	0.02	8	0.01	0.01	0.01	0.02	0.00
9	0.01	0.02	0.01	0.02	0.01	9	0.00	0.01	0.00	0.01	-0.01
10	0.02	0.02	0.01	0.02	0.02	10	0.01	0.01	0.01	0.01	0.01
11	0.01	0.02	0.01	0.02	0.01	11	0.00	0.01	0.01	0.01	0.00
12	0.01	0.02	0.01	0.02	0.01	12	0.01	0.02	0.02	0.03	0.01
13	0.02	0.02	0.02	0.02	0.02	13	0.03	0.03	0.02	0.03	0.03
14	0.02	0.03	0.02	0.03	0.02	14	0.04	0.04	0.04	0.04	0.04
15	0.03	0.03	0.03	0.03	0.03	15	0.05	0.05	0.05	0.05	0.05
16	0.04	0.04	0.04	0.04	0.03	16	0.06	0.07	0.07	0.07	0.06
17	0.05	0.05	0.05	0.05	0.03	17	0.07	0.07	0.07	0.07	0.05
18	0.05	0.05	0.05	0.05	0.04	18	0.08	0.08	0.08	0.08	0.06
19	0.07	0.07	0.06	0.07	0.05	19	0.10	0.10	0.10	0.10	0.08
20	0.08	0.08	0.07	0.08	0.07	20	0.12	0.11	0.11	0.11	0.10
21	0.12	0.13	0.12	0.13	0.09	21	0.20	0.20	0.20	0.20	0.16
22	0.00	0.00	0.00	0.00	0.00	22	0.00	0.00	0.00	0.00	0.00

Tab. 22: Relative improvement using evolved forecast combination structures in comparison to the best individual forecast ${}^0\widehat{y}$.

very low or unstable.

It was also very interesting to see that the structures which have not been effected by overfitting showed the tendency to generate basic combinations which cluster the input predictions corresponding to their type of diversification. So we could observe a clear tendency to combine first different forecasts generated at the same level but using different functional approaches and then to combine the forecasts representing different levels or vice versa.

Exceptions could often be found in cases where forecasts differed significantly in error variance, but the good ones contained highly correlated errors. In these cases total improvement of the combination is low compared to the simple choice of one of the best single predictions.

Other exceptions could be found if function spaces of different complexity have

been used. Especially in cases of very small numbers we could often achieve structures clustering more stable forecasts of lower levels with more flexible forecasts from higher levels. Similar effects have been achieved if parameters that affect the complexity are controlled by thick modelling. This effect can be explained by analysing equation (6.11) for the case $M_1 = M_2$ meaning that we have two groups of homogeneous forecasts with the same size. In this case, a direct combination as well as combination per pool defined by the diversification generate equal weights for all concerned forecasts. The negative aspects of the inhomogeneities of the covariance matrix do not affect the generation of the weights in that case. It is therefore possible to combine all of the forecasts in one step, but an additional pooling corresponding to the two groups would not effect the resulting forecast accuracy.

In total, the achieved results strongly support our findings that forecasts differing concerning more than one diversification criterion should not be combined. In cases where such structures are evolved it is not a disadvantage if the fusion is separated corresponding to the types of diversification.

We will therefore search now for approaches that evolve structures that contain the additional restriction that only forecasts are combined which differ concerning not more that one type of diversification. As this restriction represents a clear limitation of the search space there is the potential to decrease the risk of overfitting by following this idea.

7.3 Considering the Covariance Homogeneity

In the previous Chapter we have provided a theoretical analysis of the behaviour of forecasts that have been diversified by three different methods: with parameters learned at different data aggregation levels, by thick modelling and by the use of different function spaces. We have also mentioned that a side effect of the application of different types of diversification is that the number of forecasts to combine can get very large and that the resulting errors in the estimated covariance matrix

can lead to high weight estimation errors. We have therefore analysed the approach of error variance based pooling as proposed by Aiolfi and Timmermann [Aiolfi 04] in order to handle that problem. We could show theoretically that we risk a significant loss in the expected forecast accuracy because of typical inhomogeneities in the covariance matrix for the analysed case. We have proposed a new pooling approach that avoids the covariance inhomogeneities in considering only information that is contained in a simplified covariance representation based on knowledge about the forecast generation process.

In this section we describe evolutionary approaches used in order to evolve the order of pooling of the dimensions. Algorithm F^{ml} needs in each step the information which dimension D is used for the next step of pooling. We will now describe different options of how we can define such a kind of evolution.

After a short motivation we propose alternatives of how to determine the order of diversification dimensions used for pooling in our algorithm. Determining that order based on error covariances contains the already discussed risk based on estimation errors. Evolving that order avoids the time and cost consuming determination of the best structures based on static test data and additionally allows the adaptation to changed situations. The main advantages compared to the completely dynamic structures discussed in the last section is calculation time and stability of the resulting structures.

7.3.1 Genes and Chromosomes

Let us assume we have a forecast generation space given by $\mathcal{S} = \mathcal{D}_1 \times \ldots \times \mathcal{D}_K$ with K the number of diversified dimensions as already described above. The generation of combination structures following algorithm F^{ml} is determined by a vector that indicates the order of the dimensions D to be used for pooling.

We define genes as $g \in \mathcal{N}$. They each represent an index k of a dimension of the forecast generation space. Chromosomes are defined as vectors of disjunct genes $cr \in \mathcal{N}^K$. The order of the genes in a chromosome describes the order

of dimensions used for pooling. The example for a chromosome $cr \in \mathcal{N}^4$ corresponding to the pooling described in Figure 51 is provided in Figure 62 as parent 1.

7.3.2 Crossover and Mutation

We have carried out experiments using two types of child generation.

The first type generates a child based on two parent elements. The crossover considers the position of the dimensions in the chromosomes of the two parents. The child is calculated using the following algorithm :

- initialise the child cr^{child} without any genes
- loop $k = 1$ to K
  - select randomly one of the parents cr^{p1}
  - if gene cr_k^{p1} is not yet contained in the child → add gene cr_k^{p1} to the child
  - if gene cr_k^{p2} of the other parent is not yet contained in the child → add gene cr_k^{p2} to the child

An example of two parents with a generated child is shown in Figure 61.

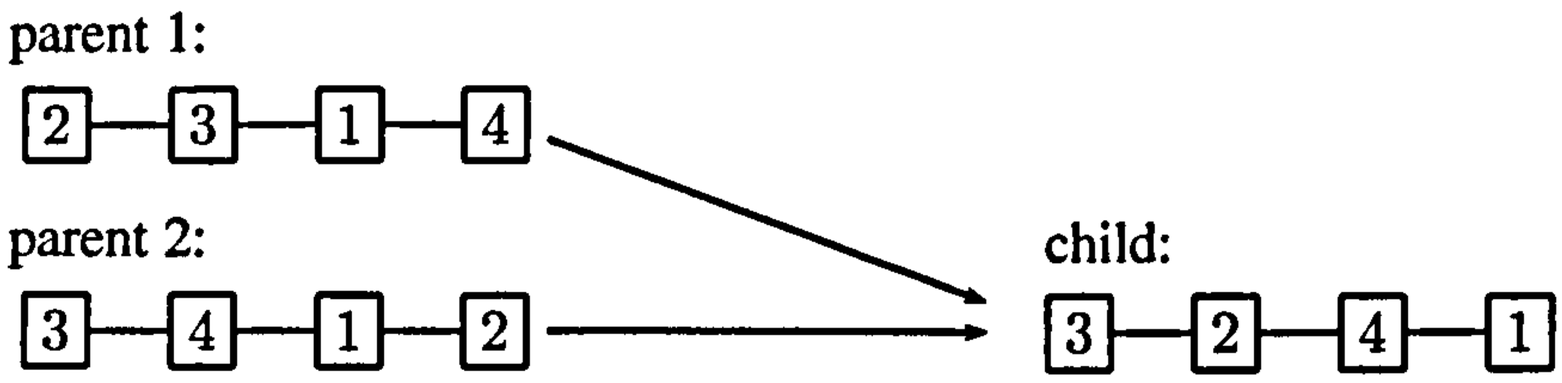

Fig. 61: Example of the first type of crossover.

The second crossover uses only one parent element. The child is generated by an exchange of any randomly selected gene with a neighboured gene. If we accept the child only if it performs better than its parent similar to Tabu Search, this type of evolution can be carried out with a very small population or even a single chromosome. Figure 62 provides an example of this type of crossover.

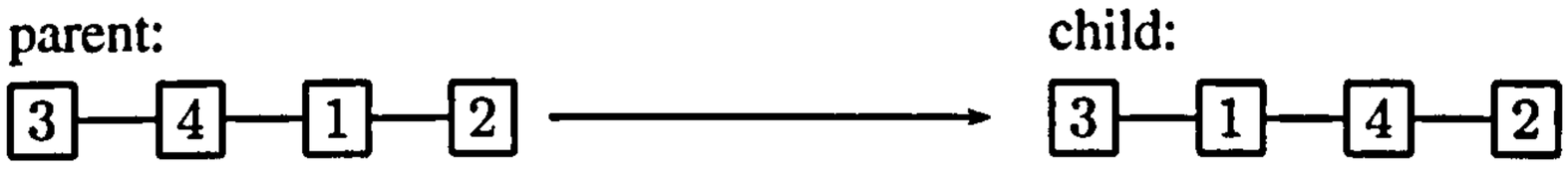

Fig. 62: Example of the second type of crossover.

The mutation has been used in order to adapt the trimming percentage. We have carried out a mutation in each fifth crossover. During the mutation the trimming percentage λ has been randomly modified up to 10 percent of the previous value. We have experimented with two types of representation of λ: a global representation with the same value used for all steps of pooling and a separate representation per combined pool.

7.3.3 Experiments

We have experimentally compared the described approaches of evolving the order of dimensions used for pooling. The experimental setup has been identical with the experiments described in Section 6.7.1 with the only difference that we have not calculated results for different predefined structures separately, but have evolved the order of pooling of the dimensions as well as the trimming percentage. Table 23 summarises the compared evolutions, Table 24 shows the relative improvement compared to the best individual forecast. Details for these experiments can be found in the Appendix in Section B.6.7.

approach	crossover	evolved trimming percentage
EV6	1	global
EV7	1	per combination
EV8	2	global
EV9	2	per combination

Tab. 23: Structures used for evolution.

7.3.4 Conclusions

It can be seen that evolving the order of dimensions allows the generation of structures which have about the same quality compared to the structures representing

low level ODO F POS					high level ODO				
τ	EV9	EV10	EV11	EV12	τ	EV6	EV7	EV8	EV9
0	0.00	0.00	0.00	0.00	0	-0.01	-0.01	-0.01	-0.01
1	0.02	0.02	0.02	0.02	1	0.09	0.09	0.09	0.09
2	0.02	0.02	0.02	0.02	2	0.11	0.11	0.11	0.11
3	0.02	0.02	0.02	0.02	3	0.10	0.10	0.10	0.10
4	0.02	0.02	0.02	0.02	4	0.09	0.09	0.09	0.09
5	0.02	0.02	0.02	0.02	5	0.08	0.08	0.08	0.08
6	0.02	0.02	0.02	0.02	6	0.06	0.06	0.06	0.06
7	0.02	0.02	0.02	0.02	7	0.05	0.05	0.05	0.05
8	0.01	0.01	0.01	0.01	8	0.03	0.03	0.03	0.03
9	0.01	0.01	0.01	0.01	9	0.03	0.03	0.03	0.03
10	0.01	0.01	0.01	0.01	10	0.03	0.03	0.03	0.03
11	0.01	0.01	0.01	0.01	11	0.03	0.03	0.03	0.03
12	0.01	0.01	0.01	0.01	12	0.04	0.04	0.04	0.04
13	0.01	0.01	0.01	0.01	13	0.05	0.05	0.05	0.05
14	0.02	0.02	0.02	0.02	14	0.05	0.05	0.05	0.05
15	0.02	0.02	0.02	0.02	15	0.05	0.05	0.05	0.05
16	0.03	0.03	0.03	0.03	16	0.06	0.06	0.06	0.06
17	0.03	0.03	0.03	0.03	17	0.07	0.07	0.07	0.07
18	0.04	0.04	0.04	0.04	18	0.07	0.07	0.07	0.07
19	0.05	0.05	0.05	0.05	19	0.09	0.09	0.09	0.09
20	0.06	0.06	0.06	0.06	20	0.11	0.11	0.11	0.11
21	0.11	0.11	0.11	0.11	21	0.20	0.20	0.20	0.20
22	0.00	0.00	0.00	0.00	22	0.00	0.00	0.00	0.00

Tab. 24: Relative improvement using evolved forecast combination structures in comparison to the best individual forecast ${}^{0}\widehat{y}$.

the best known order of dimensions. The evolutionary approach can therefore be evaluated as useful in order to determine the order of dimensions automatically. The experiments also prove that approaches which allow only the combination of forecasts that have been diversified only by one type of diversification perform better than approaches that do not contain this restriction.

Both types of crossover perform well. In many cases the solutions found by the four types of evolution represent the same order of pooling and differ only slightly concerning the trimming percentage.

8. SUMMARY AND POTENTIAL FOR FUTURE WORK

8.1 Justification for the Line of Research

The domain of multi level forecast combination is a challenging new domain containing a large potential for forecast improvements. This thesis presented a theoretical and experimental analysis of different types of forecast diversification on forecast error covariances and resulting combined forecast quality. We have seen that forecast diversification concerning the level of learning in connection with thick modelling and the use of different function spaces followed by a (multi step) combination procedure can be a powerful approach in order to build a high quality and adaptive forecast system. We have compared models differing concerning decomposition, diversification as well as concerning the applied combination models and structures.

After an introduction into the application as well as into the theory of forecast combination in the Chapters 2 and 3 we investigated aspects of diversity and diversification procedures in Chapter 4. This chapter also contains an analysis of effects of diversification in relation to different types of parameter values on error components corresponding to the bias-variance-Bayes decomposition.

Different approaches of how to include information from different levels into forecasting have been discussed in Chapter 5. The improvements achieved with multi level forecast combination prove that it is worth carrying out theoretical analysis in this relatively new field. We have provided the extension of the bias-variance-Bayes decomposition to the multi level case. An analysis of the effects of including forecasts with parameters learned at different levels on the bias and variance error components has shown that forecast combination is the best choice

in comparison to the other alternatives. The proposed approach represents a completely automatic procedure. It realises changes in the error components which are not only advantageous at the low level, but have also a stabilising effect on aggregates of low level forecasts to the higher level. We have also identified cases in which multi level forecast combination should ideally be connected with the use of different function spaces and/or thick modelling related to certain parameter values or preprocessing procedures.

We have provided an analysis of effects of such large sets of forecasts on covariance values in Chapter 6. We have seen within the bias-variance-Bayes decomposition framework that different kinds of diversification can have impacts on different error components. The "diversity" of a pair of forecasts has been quantified as the uncorrelated part of the total error variance in relation to the total error variance.

In order to avoid problems occurring for large sets of highly correlated forecasts if considering covariance information, we investigated the potential of pooling and trimming for our case. We estimated the expected behaviour of our diversified forecasts in purely error variance based pooling represented by a common approach of Aiolfi and Timmermann and analysed effects of different kinds of covariances on the accuracy of the combined forecast. We showed that a significant loss in the expected forecast accuracy may ensue because of typical inhomogeneities in the covariance matrix for the analysed case.

If covariance information is available in a sufficiently high quality, it is possible to run a clustering directly based on covariance information. We have discussed how to carry out a clustering in that case. We have also considered a case (quite common in our application) when covariance information may not be available and proposed a novel simplified representation of the covariance matrix which represents the distance in the forecast generation space and is only based on knowledge about the forecast generation process. A new pooling approach has been proposed that avoids inhomogeneities in the covariance matrix by considering the informa-

tion contained in the simplified covariance representation. One of the main advantages of the proposed approach is that the covariance matrix does not have to be calculated. We compared the results of our approach with the approach of Aiolfi and Timmermann and explained the reasons for significant improvement. Another advantage of our approach is that it leads to the generation of novel multi step multi level forecast generation structures that carry out the combination in different steps of pooling.

Finally, we described different evolutionary approaches in order to generate combination structures automatically in Chapter 7. We investigated completely flexible approaches as well as approaches that avoid the expected inhomogeneities in the error covariance matrix based on our theoretical findings. We also proposed a solution to the problem of determining the order of the dimensions used for pooling in our pooling algorithm using the simplified covariance representation.

The theoretical analysis is supported by our experimental results. We could achieve an improvement of forecast quality up to 11 percent for the practical application of demand forecasting in Revenue Management compared to the current optimised forecasting system.

8.2 Future Work

While forecast combination in general has been well studied [Timmermann 05], the research in relation to multi level forecast combination is in its beginnings. We still do not have clear understanding under which conditions to generate forecasts at which level. Further mathematical and experimental investigations will help to better understand the underlying mechanisms and to improve control of use at different levels.

Another new field that is worth further investigation is the domain of generating stable and powerful multi step combination structures. The recent work of Gabrys and Ruta show a potential of combining a large number of forecasts also in relation to fusion of classifiers [Ruta 05]. Their surprisingly good results in the NISIS

competition 2006 [Ruta 07], achieved with the application of a two step pooling of diversified neural networks for time series prediction with the pools also defined by the type of diversification in combination with trimming, show the potential of research in this domain for other applications.

Personally, I am very happy that a new PhD project in cooperation with Lufthansa Systems Berlin GmbH started in October 2006 with the objective to continue research in the domain of multi level forecast fusion. The existence of this project proves the practical relevance of the research carried out in this PhD as well as the stability of the existing cooperation between Bournemouth University and Lufthansa Systems Berlin.

The main two components in our application which decide about the quality of the final forecast are the accurate predictions of the demand based on the current and historical booking information combined with accurate predictions of cancellation rates. The main focus of the current analysis has been on the booking based forecasting and use of novel adaptable multi level forecast combination techniques for improving of the forecast quality. However, the prediction of cancellation rates which relates to the understanding and intelligent modelling of the customer behaviour has not been used extensively until now. In the new project substantial level of information stored in the airline Passenger Name Records (PNRs) will be exploited through the use of data mining approaches and new adaptable classification methods for modelling and understanding of various groups of customer behaviours and improvement of the cancellation forecasts.

Two general data sources determining potential types of models can be used for cancellation predictions: models based on PNR attribute information and models based only on information related to time and the expected number of bookings. Both types of models allow and are likely to benefit from multi level approaches.

The purely time based models depend on historical and current booking and cancellation numbers related to different times prior to the departure. Traditionally these are statistical time series approaches or causal models predicting the

absolute number of expected cancellations, the cancellation rate or probabilities of cancellation per booking. The issue here is the choice of the level (or following the already mentioned multi level approaches) of the determination of historical cancellation rates or probabilities, the adaptation to the special booking behaviour of the current departure to predict and the adaptation to different types of changes like seasonal effects, schedule changes and others. In addition there is the hard issue of pre-processing very small booking and cancellation numbers which leads to anomalous extreme cancellation rate predictions if they are not stabilised at the fine level.

The second class of models is based on exploitation of the PNR attribute information within a data mining frameworks which through various exploratory data analysis approaches would then result in generation of clustering, classification and predictive models used for identification and description of different customer behaviours and groupings with different propensities for cancellation in different circumstances.

One of the main aims and challenges of the new project is a development of an adaptable framework within which the times series based forecasts of the cancellation rates will be combined with cancellation forecasts based on the modelling of customer specific behaviour.

APPENDIX

A. DEFINITIONS RELATED TO AIRLINE REVENUE MANAGEMENT

A.1 Region

In this subsection we define locations like airports, cities and routings.

Definition A.1 (airport): The set $AP \subset \mathcal{R} \times \mathcal{R}$ is the set of airports. The airports $ap \in AP$ are described as a pair of their longitude and latitude. The ID of an airport ID(ap) is a unique three letter string,e.g. ID(ap)=FRA means the airport of Frankfurt/Main. The set of airports can be ordered by the longitude/latitude or by the ID.

Definition A.2 (city): A city (in the airline meaning) $ci \in CI$ is a set of airports $ci = (ap) \subset AP$. Every airport belongs to one and only one city. Cities have unique three letter IDs, too. Cities (in the airline meaning) handle the fact, that big cities (in the general meaning) can contain more than one airport.

Definition A.3 (country, global traffic area): Similar to the definitions of cities, countries $cou \in COU$, $cou = (ci) \subset CI$ are defined as sets of cities and global traffic areas $gta \in GTA$, $gta = (cou) \subset COU$ as sets of countries. Single difference: the IDs of countries and global traffic areas are unique two character IDs.

Definition A.4 (leg): A $leg \in LEG \subseteq AP \times AP$ is an ordered pair of airports. If $leg = (ap1, ap2), ap1, ap2 \in AP$,then ap1 is called the origin O(leg)=ap1, ap2 is called the destination D(leg) =ap2.

Legs are written by the IDs of the airports, too, i.e. leg(ap1, ap2)= leg (ID(ap1) ID(ap2)) or leg = ID(ap1) ID(ap2). The leg from Frankfurt Main to Berlin Tegel could, e.g., be written as leg(FRA TXL) or leg=FRA TXL.

Definition A.5 (routing): A routing $rou \in ROU$ is an ordered set of legs $rou = (leg_i) \in LEG, i = 1..n \in \mathcal{N}$, which satisfies the conditions

- $o(leg_i) = d(leg_{i-1}) \; \forall i = 2..n$ and
- there do not exist cycles, i.e. for any airport $ap \in AP$ with $ap \in leg_i$ and $ap \in leg_j$ with $i < j \rightarrow j = i + 1$.

The origin of a routing O(rou) is defined by the origin of its first leg and the destination D(rou) of a routing is defined by the destination of the last leg, i.e. it is $O(rou) = O(leg_1)$ and $D(rou) = D(leg_n)$.

There exist different notations for routings, like $rou(leg_1, leg_2, , leg_n)$ or $rou = leg_1, leg_2, , leg_n$. As legs are unique pairs of airports, routings can be described directly by the list of airports, too, i.e. $rou(leg_1, leg_2, , leg_n)$ can be written with the notation $rou(O(leg_1), D(leg_1), D(leg_2), , D(leg_n))$ or $rou = O(leg_1)D(leg_1)D(leg_2)D(leg_n)$ given the fact that $O(leg_2) = D(leg_1), , O(leg_n) = D(leg_{n-1})$.

In a lot of applications it is not relevant on which way to come from an airport to another airport, that is why routings with the same origin and the same destination are clustered to ODs.

Definition A.6 (OD): Given the set of existing airports AP, the set of existing routings ROU and two airports $ap1, ap2 \in AP$, an $od(ap1, ap2) \in OD$ is defined as the set of routings $(rou_i) \subset ROU, i = 1..n \in N$, where $O(rou_i) = ap1$ and $D(rou_i) = ap2 \; \forall i = 1..n$. The origin and destination of the routings is also called the origin and destination of the od, O(od) and D(od).

A.2 Time

Now some definitions concerning date and time are given. As everybody knows what is a day of week, some of the definitions may seem to be unnecessary, nevertheless they are given in order to clarify the notation used in other sections.

The sets D and TIME are used to define process-, departure- and arrival dates and times.

Definition A.7 (Date / Time): The set of dates $D \subset \mathcal{N} \times \mathcal{N} \times \mathcal{N}$ is the set of valid calendar dates. A date is defined as the triple $date := (day, month, year) \in D$. It is $day \in (1,..,31) \in \mathcal{N}$, $month \in (1,..,12) \in \mathcal{N}$ and $year \in \mathcal{N}$. The set of times $TIME \in \mathcal{N} \times \mathcal{N}$ is the set of valid minutes of a day given in hours and minutes, i.e. $time = (hour, minute) \in TIME$, $hour \in (0,..,23) \in \mathcal{N}$, $minute \in (0,..,59) \in \mathcal{N}$.

The notation of the dates is not standardised, all international formats to describe a date are possible. In this thesis the notation day.month.year is used. The notation of the time is not standardised either, in this thesis we use the notation hours:minutes. If dates and times are connected with locations, it must be defined whether the hour and minute information refers to the UTC (European Standard Time) or the LT (Local Time).

Definition A.8 (Day of Week): The set $DOW = (1,...,7) \in \mathcal{N}$ is the set of the existing days of week. In this notation dow=1 means "Monday", dow=2 means "Tuesday" etc. The day of week can be obtained as a function of a date $dow(date)$, $date \in D$.

Definition A.9 (Calendar Week): The set of calendar weeks $CW = (1,..,53) \in N$ is the set of ISO calendar weeks. A calendar week $cw \in CW$ can be obtained as a function of a date $cw(date)$, $date \in D$.

Given a calendar year $year \in \mathcal{N}$, a date can be obtained as a combination of the calendar week and a day of week $date(cw, dow) \in D$, $cw \in CW$ and $dow \in DOW$.

Another definition to handle relative dates to another date (in the next case it is the departure date) is the definition of snapshots and the snapshot grid.

Definition A.10 (Snapshot Grip/Snapshot/ DCP): A snapshot grid is a function $SG : \mathcal{DCP} \subset N \mapsto DAYSTODEP \subset N$ with $\mathcal{DCP} = (1, .., DCP)$ and $DAYSTODEP \in [0, 362]$. The function is strictly decreasing, i.e. $SG(dcp) > SG(dcp + 1)$, $dcp = 1..DCP - 1$ and it is $SG(DCP) = 0$. The elements $dcp \in DCP$ are called Data Collection Points. The elements $(dcp - 1) = (0, ..., DCP - 1)$ are called snapshots, too.

The meaning of the snapshot grid is to indicate days (relative to a given departure date) on which some actions (like producing forecasts for that departure) have to be done, e.g. $SG(1) = 350$, $SG(2) = 182$, etc. means that the first action concerning a departure has to be done 350 days before the departure, the second action 182 days before the departure, etc.

There are three points which are often discussed using snapshot grids:

- Should more than one snapshot grid be used or is it sufficient to have only one?
- How many snapshots should be used?
- How should the snapshots be selected?

The answers to the questions are correlated. There exist studies that recommend that in general it is sufficient to have no more than 17 snapshots, and the snapshots should be selected such that the mean value of bookings between two snapshots is constant. In reality snapshot grids depend more on the controlling process than on these suggestions.

A.3 Flight Schedules

In the last two subsections the basic definitions related to locations and points of time have been given. This allows now to describe flights, segments and ODIs.

Definition A.11 (Flight / Flight Schedule): A (planned/ realised) flight is (in the sense of this paper) an element of the set $FL = LEG \times D \times TIME$, i.e. a flight $fl \in FL$ is determined by a leg, a departure date and a departure time. A flight schedule $fs \in FS \subseteq FL$ is the set of currently planned flights given a special process date, i.e. $fs(pd) = (fl) \subset FL$ with $pd \in D$.

We use the notation fl(leg, date, time), the components can be retrieved by the functions $leg(fl)$, $d(fl)$ and $t(fl)$. The origin and destination of a flight fl are described by its leg, i.e. $O(fl) = O(leg(fl))$ and $D(fl) = D(leg(fl))$. The routing of a flight $rou(fl) \in ROU$, $fl \in FL$ describes the routing of the leg of the flight, i.e. rou(fl)=rou(leg(fl)).

In the airline world, flights have lots of other characteristics, such as aircraft type and different kinds of states.

Definition A.12 (Segment): A segment $seg \in SEG$ is an ordered set of flights $seg = (fl_i) \subset FL$, $i = (1..n) \in N$, a passenger can book under a special ID, the flight number.

The origin and destination of a segment are defined by the origin of the first flight and the destination of the last flight, i.e. $O(seg) = O(fl_1)$ and $D(seg) = D(fl_n)$. Segments are built by the airlines with the following restrictions:

- Every flight of the current flight schedule builds a segment.
- The set of the ordered legs of the flights of a segment is a routing, i.e. it is $rou(seg) = (leg_i) = (leg(fl_i)) \in ROU$.
- The departures of the flights fl_i differ not more than 24 hours, i.e.

$$d(fl_i) = d(fl_{i-1}) \tag{A.1}$$

and

$$time(fl_i) > time(fl_{i-1}) \tag{A.2}$$

or

$$d(fl_i) = d(fl_{i-1}) + 1 \tag{A.3}$$

and

$$time(fl_i) < time(fl_{i-1}) \tag{A.4}$$

holds for every $i = (2..n) \in N$

Lots of calculations and reports in the airline industry are based on the segment level. In general, segments consist of only one flight. In very few cases there exist segments containing more than one flight (so called multi leg segments). Segments are constructed to simplify the booking process in the airline industry. Multi leg segments are also constructed to follow the philosophy/policy of "one face to the customer". Passengers can book segments under one flight number, they buy one product, even if they have to change the plane during their trip.

If we want to have a look at bookings concerning network effects, we have to take into account that many passengers want to book more than one segment. People living near small airports often fly to a bigger airport first (inbound flight) before they take for instance a transatlantic flight (main segment). Often there is also a flight bringing them finally to the airport of destination (outbound flight). Other passengers are using hubs like Frankfurt to connect two longer flights, because the distance is too long to do it in one segment. A typical example are people travelling from India via Frankfurt to New York. People who want to fly more than one segment have to be accepted on all segments or none. As there is a price difference between a booking for two or more segments and the sum of prices which local passengers would have to pay for each segment, it is important for the optimisation process to know the network flows. That is why we define ODIs, which are representing sets of segments.

Definition A.13 (ODI): An $odi \in ODI$ is an ordered set of segments $odi = (seg_i) \in SEG$ (with $seg_i = (fl_{ij}) \in FL, j = 1..n(i)$) with the following properties:

- The set of the ordered legs of the segments is a routing, i.e. it is

$$rou(odi) = (leg_{ij}) = (leg(fl_{ij})) \in ROU \tag{A.5}$$

with leg_{ij} ordered by i,j.

- The dates of the flights fl_{ij} differ no more than 24 hours, i.e. it is

$$d(fl_{i,1}) = d(fl_{i-1,n(j-1)}) \tag{A.6}$$

and

$$time(fl_{i,1}) > time(fl_{i-1,n(j-1)}) \tag{A.7}$$

or

$$d(fl_{i,1}) = d(fl_{i-1,n(j-1)}) + 1 \tag{A.8}$$

and

$$time(fl_{i,1}) < time(fl_{i-1,n(j-1)}) \tag{A.9}$$

for every $i = 2..n \in \mathcal{N}$

- ODIs are built on the maximal segments, i.e. for any segment $seg \in SEG$, $seg \notin odi$ with $rou(seg) \subset rou(odi)$, there exists a segment $\overline{seg} \in ODI$ with $fl \in \overline{seg} \, \forall fl \in seg$.

An odi can be interpreted as the complete description of the locations and time of a simple trip (without return). So a passenger can, e.g., fly from

- Delhi on 10 of October 2001 at 23:10 to Frankfurt/ Main and from
- Frankfurt/ Main on 11 October 2001 at 8:40 to Berlin Tegel

ODI is the abbreviation of Origin Destination Itinerary.

The definition for a trip is nearly the same as for an odi, the only difference is that the segments must not build routings, i.e. airports can be repeated more than once. Trips can be used to model complex trips, e.g with return.

Definition A.14 (TRIP): A $trip \in TRIP$ is an ordered set of segments, it is $trip = (seg_i) \in SEG$ (with $seg_i = (fl_{ij}) \in FL$, $j = 1..n(i)$) with the following behaviour:

- $O(seg_i) = D(seg_i - 1), i = (2 \dots n) \in \mathcal{N}$
- The dates of the flights fl_{ij} differ no more than 24 hours (see (A.6) to (A.9)).
- Trips are built on the maximum segments.

To be able to learn information related to odis, we have to cluster similar odis over different departure dates.

Definition A.15 (ODO): An $odo \in ODO$ is a cluster of odis containing all the same routing and comparable departure times. Each odo contains maximal one odi at any given departure date $d \in D$. Each odi belongs to one and only one odo.

Odos (abbreviation for origin destination opportunity) represent stable history pools for odis. Note that they are not effected by flight number changes and small departure time changes.

A.4 Booking Conditions

Most of the (traditional) airlines offer different comfort levels, which are described with the term of compartments.

Definition A.16 (compartment): A compartment is an element of the countable set $COMP = (1..n) \in \mathcal{N}$ describing the comfort level during a flight.

The size of the set of compartments COMP is defined per airline. Compartments describe the comfort on board, such as the quality and distance of the seats, the food, the quality of entertainment, number of stewardesses, etc. Compartments are in general described by a name and a unique one letter ID, for instance, name(0)= FIRST, ID(0)=F, name(1)= BUSINESS, ID(1)=C and name(2)=ECONOMY, ID(2)=M.

However, not only the level of comfort determines the price. Bookings can also be made on different conditions concerning

- the possibilities of free cancellation and booking changes,
- the platform, place and time of the booking (e.g. bookings on internet),
- special booking conditions (e.g. for members of several companies),
- several passenger states like the senator state or "miles and more" passengers

and lots more.

The price of a booking depends on all of the conditions on which the booking has been made, i.e. on a plane a passenger can sit directly next to another passenger with exactly the same comfort but having paid only half or less of the price. The different conditions and prices are clustered in fareclasses.

Definition A.17 (fareclass): A fareclass $f \in F = (1..n) \in \mathcal{N}$ is a cluster of sets of booking conditions.

If passengers make a booking in a booking class, it does not mean, that they have to pay the same price. In some cases, the clustering into booking classes is made regarding the price, but that is not a general rule. Fareclasses can be seen as a description of the "value" of a passenger for an airline, which can be determined by the price the passenger is paying, but can also be an attention to customers who fly a lot with that airline or who belong to several companies. Fareclasses are defined per airline and the clustering rules can be very different. The only rule for the

clustering is that all passengers booking in a fareclass book the same compartment, i.e. a function $comp(f) \in COMP$ can be defined for every $f \in F$. Fareclasses are described by a one letter ID.

It is also relevant where a booking is made. That is why we define point of sales.

Definition A.18 (pos): A $pos \in POS = (Country\ of\ Origin,\ Country\ of\ Desinaton,\ Other)$ is an indicator in which country a booking on a given od has been made.

B. DESCRIPTION OF EXPERIMENTS AND THE APPENDED SOFTWARE

B.1 Introduction to the "Avanti" Software

The software *Avanti* has been developed in order to carry out experiments related to this thesis. It has been implemented in Visual C++ and uses MFC for the graphical user interface. It enables the reproduction of the presented experimental results as well as modifications concerning for instance parameter values. It also contains a data visualisation component and with that the opportunity to visualise and analyse all intermediate results on any requested level of detail.

Avanti is strongly based on the *forecasting kernel* developed by Lufthansa Systems Berlin. It uses forecasting methods implemented in the forecasting kernel as well as objects for data representation and manipulation. The forecasting kernel as well as the *Avanti* software are implemented as application independent tools. This means that they offer functionality in a manner that the methods could also be used for a completely different application.

In order to use the functionality, application specific information has to be provided. This information contains

- information about the input space (in terms of dimensions/levels) (see B.3.3)
- a description of the existing data (see B.3.4)
- a specification of the methods to be applied (see B.3.5) including
  - the order of processing
  - parameter settings

- a specification on which data calculation should be carried out.

Within *Avanti*, any calculation on data is carried out by software components which we will call *calculation components*. Each calculation component provides a certain functionality, such as loading data from files or application of a certain method of forecasting or history building. *Calculation components* can be parameterised and need a specification of data on which they should be carried out. Details will be provided in Subsection B.3.5.

Data is stored within *Avanti* in *multi dimensional data cubes*. Data cubes need a specification concerning their names and their dimensionality. This will be explained in detail in Subsection B.3.4.

B.2 How to Install Avanti

Start setup.exe in order to carry out the installation of *Avanti*.

The installation provides the executables *avanti.exe* for calculation with a graphical user interface and *avantiBatch.bat* for batch processing of different data directories (see B.3.11). A parameter file is provided as well.

Additionally, two subdirectories are created during the installation within the main directory into which *Avanti* is installed. The first one is the data directory containing all used experimental data. The data directory contains different subdirectories representing the different ODs and ODOs. The same files can be found in each subdirectory. The second directory is the directory containing the data group and dimension descriptions as well as the experimental descriptions (component lists and diversification lists) per experiment.

B.3 How to Run Experiments

B.3.1 Overview of the Graphical User Interface

The Graphical User Interface of *Avanti* is composed of four different views as shown in Figure 63.

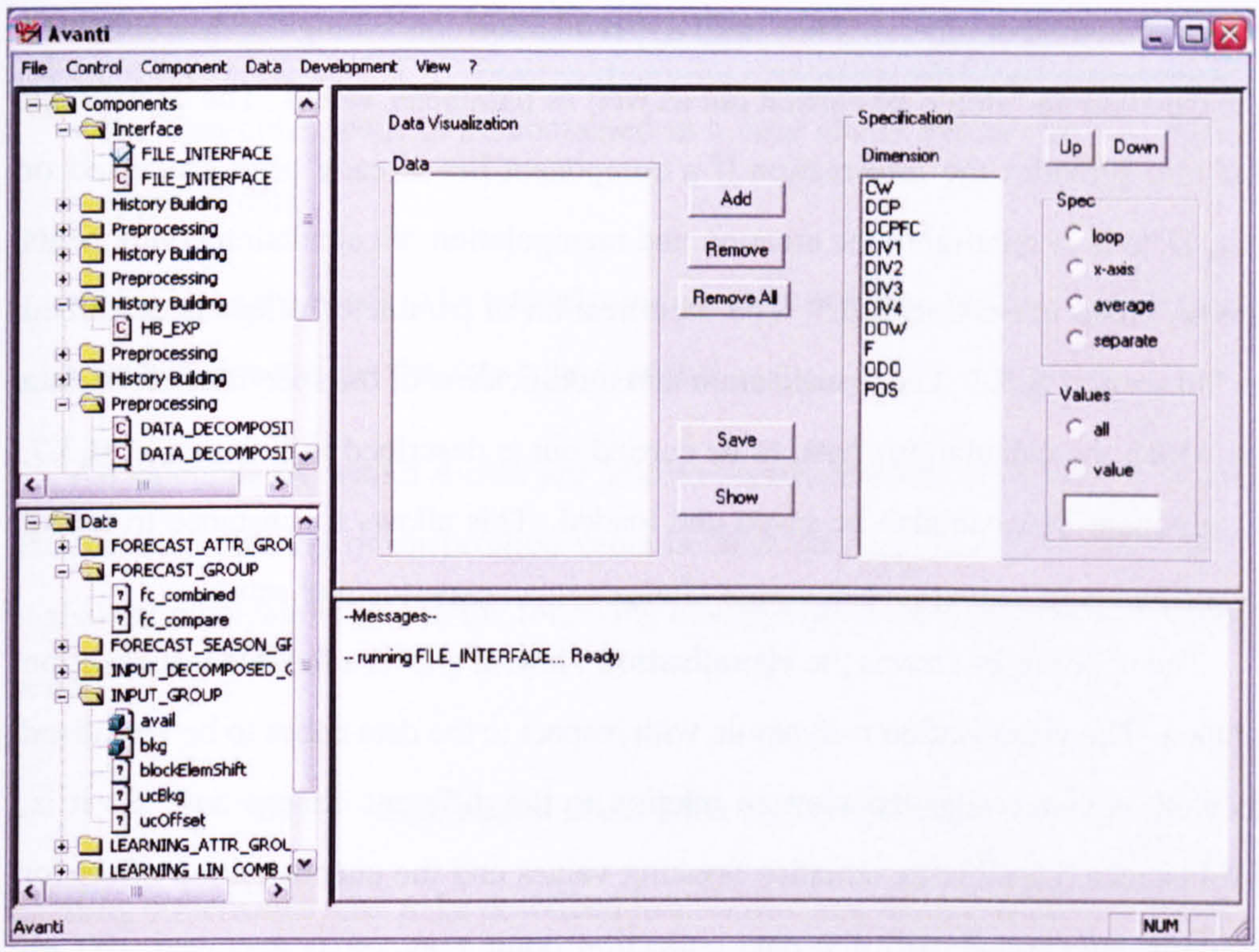

Fig. 63: Overview of the *Avanti* Graphical User Interface

The lower left view is the **data view**. This view contains information about the used data. The data is provided in a list grouped by *data groups*. A data group contains similar types of data. So we have, e.g., defined a data group *learning_attr_group* containing all data related to information about the attractiveness learned by different models. The data view also visualises the state of a data cube (like *created*, *loaded* or *updated*). Details about the data view are provided in Subsection B.3.2. The data view allows the selection of data cubes for visualisation or in order to request additional information concerning the dimensions of this data cube (see Subsection B.3.3).

The upper left view is the **component view**. The component view shows the specification of calculations to be carried out in terms of a list of calculation components which represent the setup of an experiment. Components represent a certain type of calculation to be carried out (like loading data, application of a certain

forecast method or others). They need a specification concerning the data on which the calculations should be carried out as well as parameter values. The component list also provides the information if a component has already been processed or not. Details in relation to the creation and manipulation of calculation components are provided in Section B.3.5. The modification of parameter values is described in Subsection B.3.6. The visualisation and modification of the specification of data on which the calculations have to be carried out is described in Subsection B.3.7. Component lists can also be saved and loaded. This allows for instance to repeat experiments or to incorporate minor changes in an experimental setup.

The upper right view is the **visualisation view.** It allows a flexible visualisation of data. The visualisation is dynamic with respect to the data cubes to be visualised as well as concerning the view in relation to the different dimensions. So it is, for instance, possible to visualise booking values and the unconstrained offset for a single fareclass F, with the data collection point τ as x-axis averaged over all departure weeks and each day of week represented as a separate line in one figure or in a separate figure. Details about how to visualise data together with examples are given in Subsection B.3.8.

The lower right view is the **message view.** It provides information about the current state of a calculation. The message view is described in Subsection B.3.10.

B.3.2 Handling and Visualisation of Data

Data cubes are characterised by dimensions in relation to the data that exists in a given context. So we have for instance booking data given for different fare-classes (F), point of sales (POS), departure weeks (DW), day of weeks (DOW) and data collection points (DCP). A data cube can only be created by specifying these dimensions.

The following information is needed in order to create a data cube:

- each dimension needs to be specified concerning the *extent of the dimension* indicating how many elements the dimension contains (for instance:

the number of existing days of week is 7)

- the data cube needs to be contained in a *data group* which clusters similar types of data

- the data cube needs to be related to a *data cube extent specification* indicating which dimensions the data cube contains

All calculations within *Avanti* are able to handle missing data values. In the data cubes a missing or unspecified value is characterisied as *-1000*. This value is also called *default value* in the following descriptions of the calculation components.

B.3.3 Information about Data Cube Dimensions

Existing dimensions have to be described in a file *dimensions.dat* which is expected in the component list directory specified by the global parameter *pComponentList-Directory* (for details about the meaning of this parameter and how to modify the parameter value see B.3.6).

Each line in the dimensions file represents a pair of a name of a dimension and its extent. An example is shown in Figure 64. The dimensions used for our application will be described in Section B.5.1.

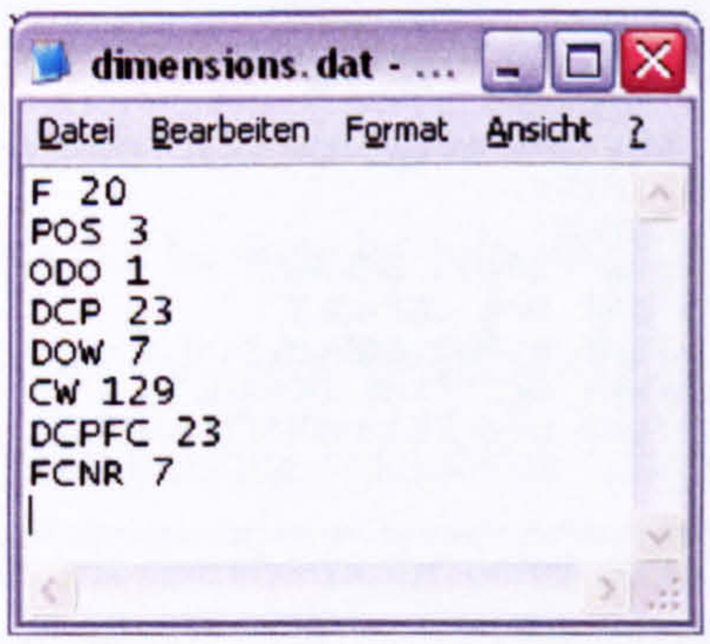

Fig. 64: Example for a specification of existing data dimensions in file *dimensions.dat*.

B.3.4 How to Specify Data Groups

Data groups contain lists of data cubes as well as information about their dimensions. They are specified in files called

<data_group_name>.avdg expected in the component list directory specified by the global parameter *pComponentListDirectory* (see B.3.6). Figure 65 shows an example.

The first line of a data group description file always contains the keyword *DATAGROUPNAME* and then the name of the specified data group. Then follow descriptions of one or more data cube extent specifications. Each extent specification is described in two lines. The first line contains the keyword *EXTENT* followed by the name of the extent specification. The second line contains a description of the used dimensions. It starts with the number of dimensions and then indicates the names of the dimensions. It is followed by a description of the data cubes. Each data cube is described by its name and its extent. The lines contain first the keyword *CUBEEXTENT*, then the name of the data cube and then the name of the extent specification, describing the structure of the data cube. We will describe the data groups defined in our experiments in Section B.5.2. A detailed description of the used data cubes will be provided per experiment in Section B.6.

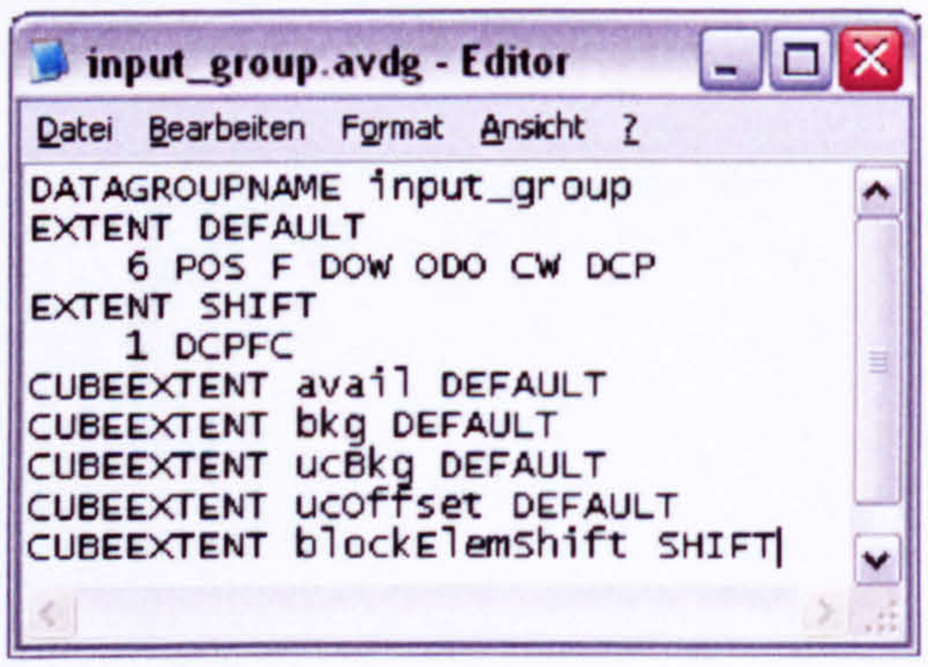

Fig. 65: Example for a data group *input_group.avdg*. It contains two cube extent specifications called *DEFAULT* and *SHIFT*. Then four data cubes called *bkg*, *avail*, *ucBkg* and *blockElemShift* are specified .

After having been loaded (see B.3.10), each existing data group is represented

as a directory in the *data view* (see Figure 63). The elements in the directories represent the data cubes. The symbol next to the name of the data cube shows if a data cube has already been loaded or updated. The symbol containing a question mark indicates a data cube which has not yet been used.

The dimensions of a data cube can be visualised in *Avanti* as well. *After having selected a data cube in the data view* select *Data/ ShowCubeDimensions* in the menu. A dialogue appears that shows all dimensions of the data cube together with their extents as shown in Figure 66.

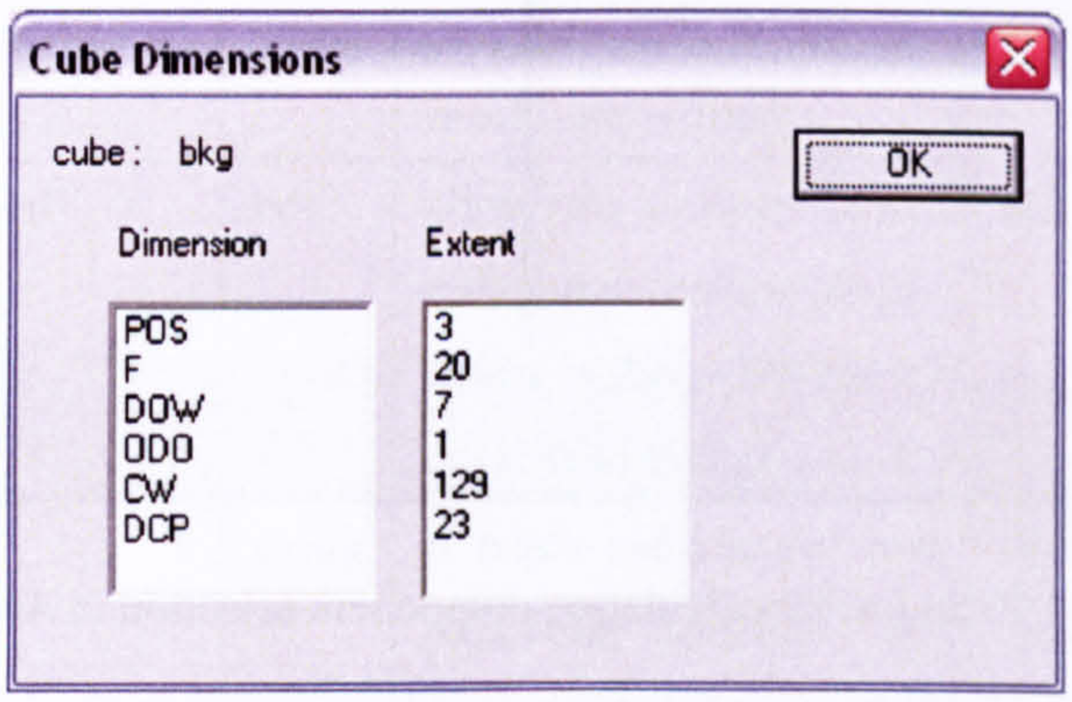

Fig. 66: Example for dimensions of a data cube in *Avanti*.

B.3.5 Calculation Components

Calculation components represent application independent calculation units. In order to carry out an experiment, it has to be defined which calculation components to use in which order. This can be done by selecting *Component/Add Component* in the menu. The dialogue for calculation component selection is shown in Figure 67. At the right hand side all existing calculation components are listed grouped by component types. The left list shows the selected components. The dialogue allows the selection of new components as well as changes concerning the order of calculation.

Component lists can also be loaded and saved over the menu (use *File/ Load-Component List*). This allows the re-use of experimental descriptions. At the mo-

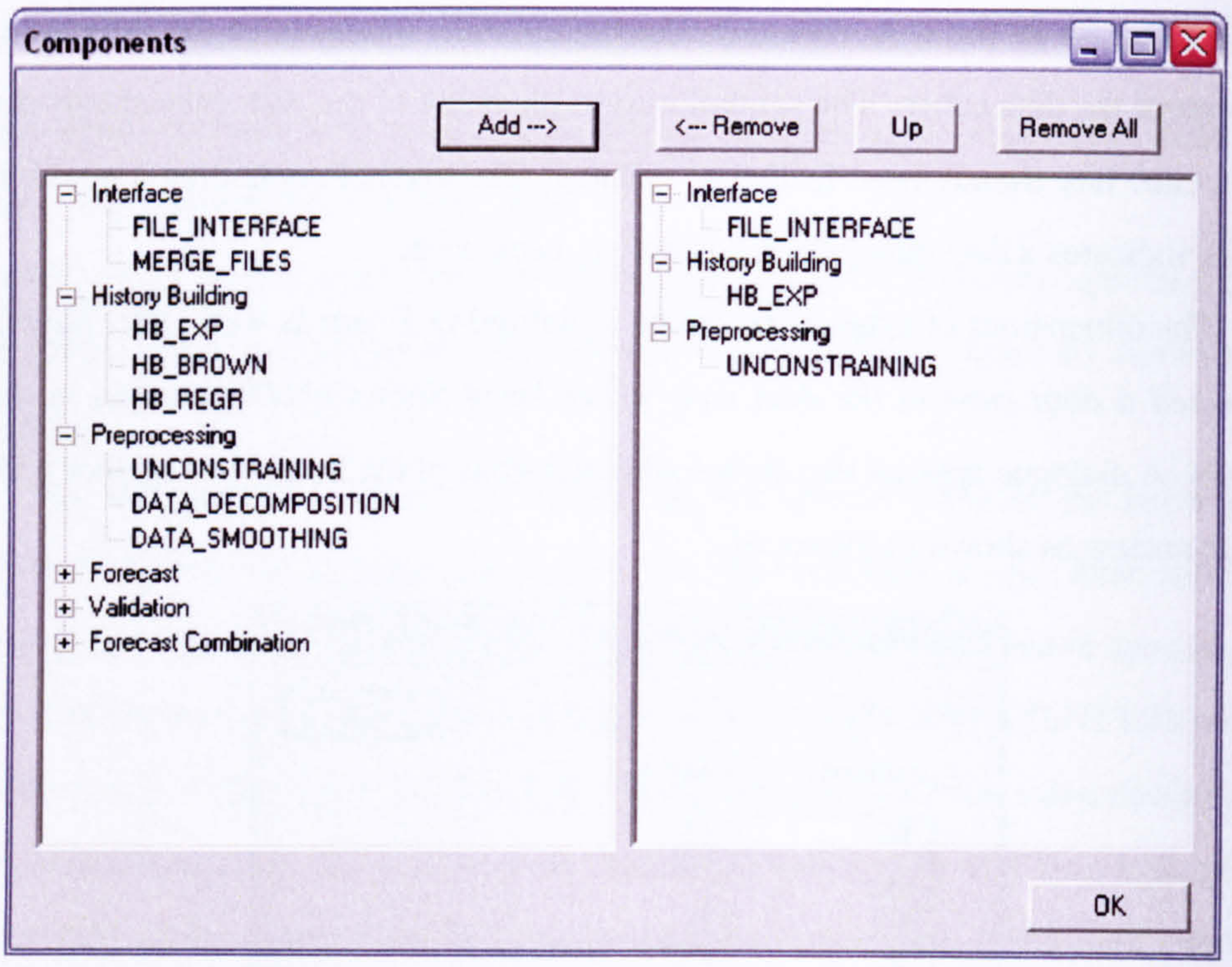

Fig. 67: Example for calculation component selection in *Avanti*.

ment when a component list is created or loaded, *Avanti* also automatically loads the data group specifications and shows the data visualisation view.

A special type of component list selection is carried out in case of global parameter setting *pAutomaticCalculation="on"*. In this case, the loading of the component list file *component_list.avcl* is carried out automatically when *Avanti* is started.

An overview of used calculation components will be given in Section B.4. The component lists applied for the different experiments are presented in detail in Section B.6.

B.3.6 Handling and Visualisation of Calculation Parameters

Two kinds of parameters are defined within *Avanti*: global parameters and parameters relating to special calculation components. Three types of parameters *flag (bool), value(int or float)*and *string* are supported for both kinds.

Global parameters are fixed for all calculations. They can be visualised and modified using the menu by selecting *Data/ SetGloabalParameter.* The following table shows the global parameters that are currently used.

parameter	type	description	example
pComponentListDirectory	string	directory of the component list	C:/Avanti/Experiments/
pDataDirectory	string	directory from which the data is loaded	C:/Avanti/Data/
pResultDirectory	string	directory into which the results are written	C:/Avanti/Results/
pBatchCalculation	bool	allows the successive calculation in relation to different subdirectories of the data directory	off
pBatch	string	if batch calculation, this parameter specifies the current batch element to be processed. If "batchlog", the current element is loaded from the batchlog.dat file.	"OD1-ODO1" or "batchlog"
pExperiment	string	a subdirectory specifying an experiment is used for loading of the component list and saving of the results	Experiment5
pAutomaticMode	bool	a component list as well as the data groups are loaded and the calculation is started automatically if the software runs in this mode	on

Component parameters are specified in relation to a specific calculation component. They can be visualised and modified *after having selected a component in the component view* via the menu with *Component/ SetComponentParameter*. Then a component-specific parameter dialogue appears which allows parameter values to be shown and overwritten. An example is shown in Figure 68. The meaning of the different parameters of the components applied in our experiments will be described in Section B.4.

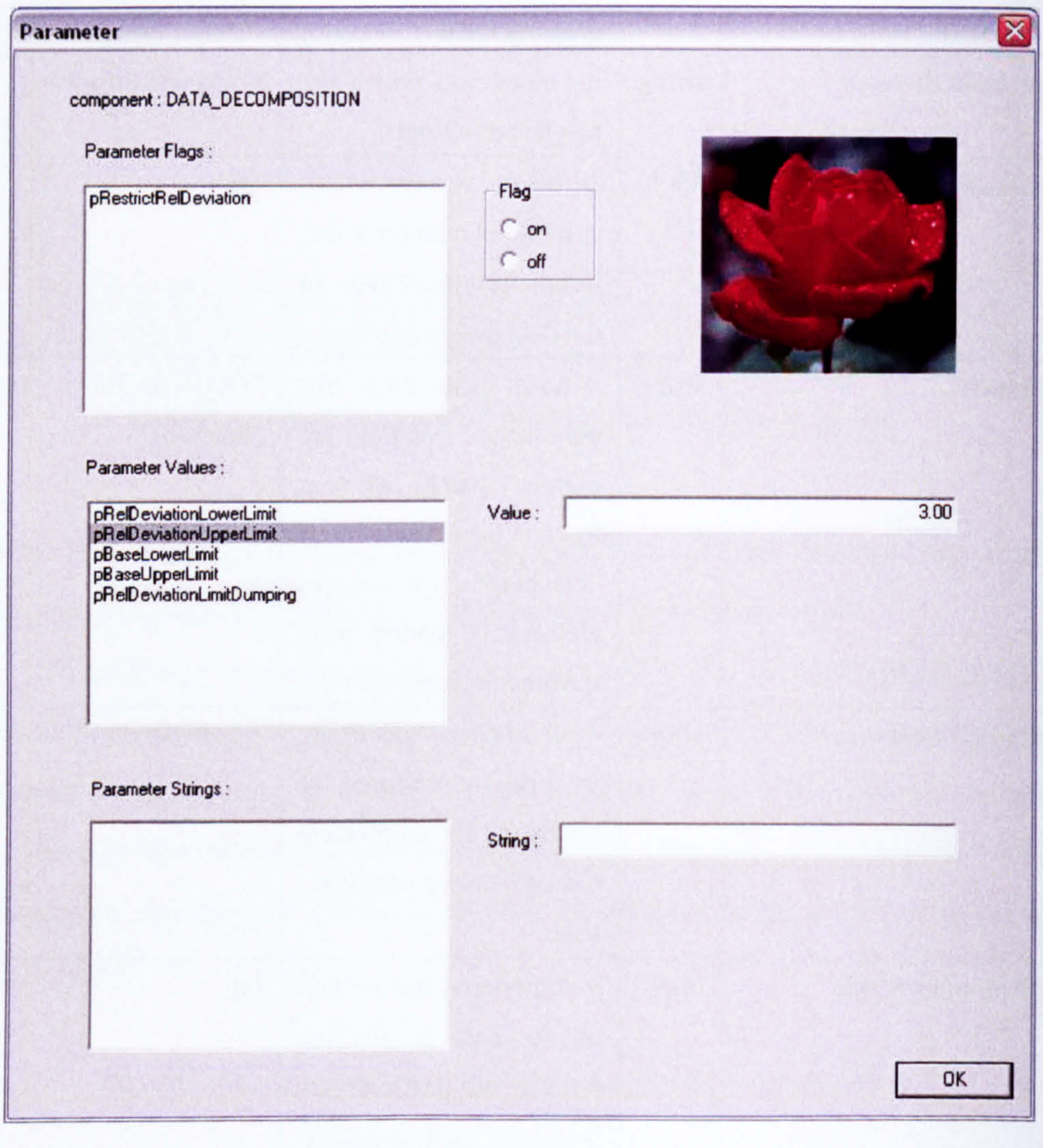

Fig. 68: Modification of parameter values in *Avanti*.

B.3.7 Specification of Data to be Used for a Calculation

Figure 69 shows an example for the dialogue that enables the specification and modification of data cubes to be used for a calculation. The dialogue can be reached *after having selected a component in the component view* via the menu with *Component/ Set Input/Result Cubes.*

In this dialogue the list on the left hand side contains the application independent internal names of the data corresponding the its general role for the calculation. The dialogue also provides the information if this data is input data, result data or both. The list at the right hand side contains the currently selected data cubes. An error will occur during the calculation if an external name remains *"UNDEFINED"* or if the indicated external name does not correspond to the name of a valid data cube. The external name can be modified by writing the new data cube name in the edit element at the bottom of the dialogue. Select the internal name with which this data cube should be used and press the button *Change Extern Name.*

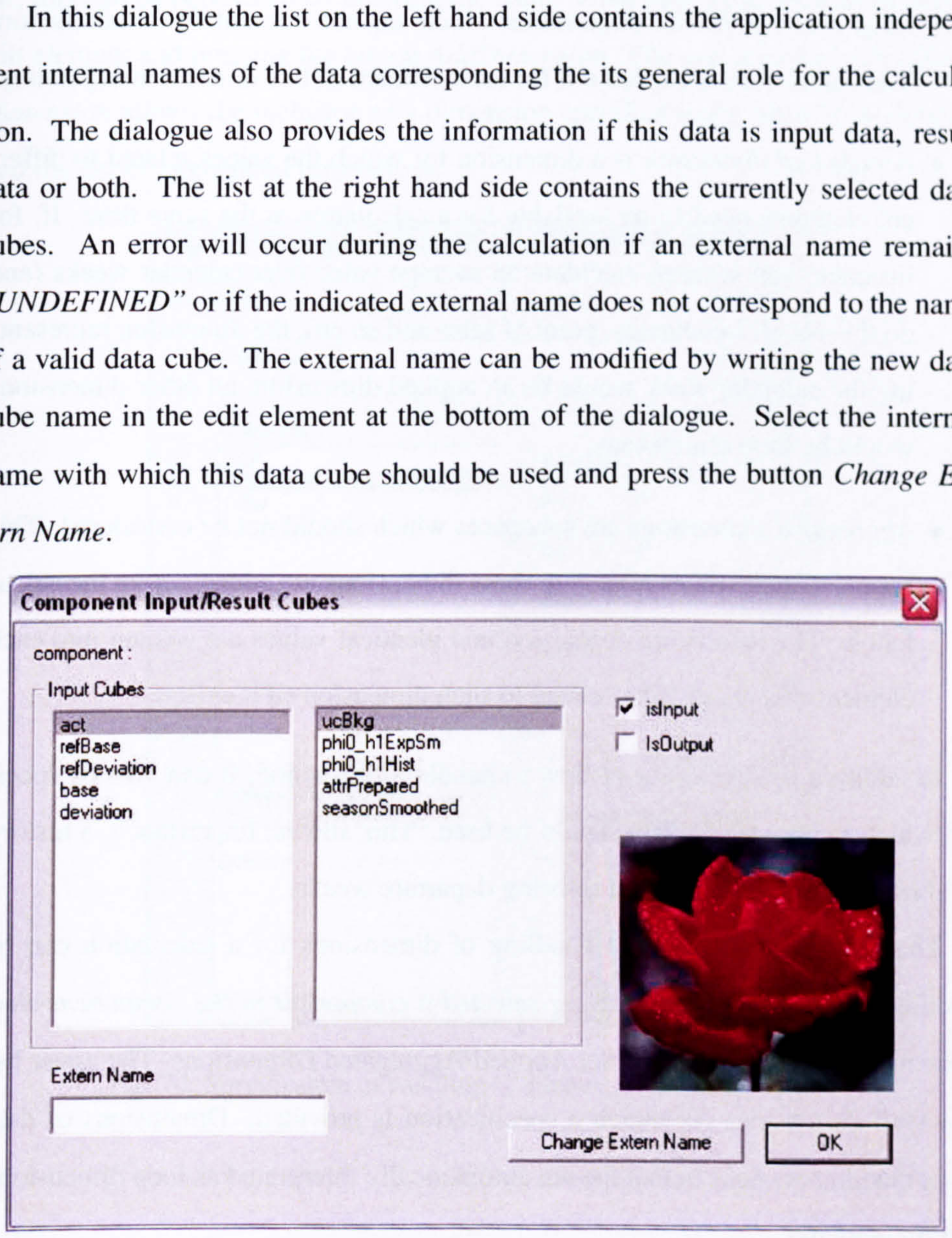

Fig. 69: Specification of data cubes to be used for a calculation.

A way of handling different data dimensions has to be defined as well. Often calculations need to be carried out for each element of a dimension in a separate manner, in other cases data corresponding to different elements of a dimension need to be available for a calculation.

Three types of interpretation of dimensions are possible within *Avanti*:

- *Loop dimensions* are dimensions which are not relevant for a calculation. The calculation is carried out for each element of this dimension separately.

- An *applied dimension* is a dimension for which the values related to different elements need to be available for a calculation at the same time. If, for instance, you want to calculate an average value over calendar weeks (and do this for all fareclasses, point of sales and so on), the dimension representing the calendar week would be an applied dimension, all other dimensions would be loop dimensions.

- *Aggregated dimensions* are subspaces which should not be considered. This means that all values related to these dimensions are added before the calculation. The results are duplicated and identical values are written into each element of a result cube related to such dimension (if it exists).

In addition to specifying of how to handle a dimension, it can also be specified which range of elements should be used. This allows, for instance, a history building only on a subset of the existing departure weeks.

The dialogue in which the handling of dimensions for a calculation can be specified can be called *after having selected a component in the component view* in the menu with *Component/ Set Applied/Aggregated Dimensions.* The upper list shows all dimensions for which a specification is provided. Dimensions of data cubes that do not occur in that list are automatically interpreted as loop dimensions over all elements.

The specified range of elements of a dimension to be used can be modified by first selecting this dimension in the list. Then it is possible to modify the first ele-

ment and the last element which should be considered for the selected dimension. The fact whether the selected dimension should be a loop dimension or not can also be modified. Dimensions which should be aggregated occur in the list at the bottom.

If you want to add a specification for a dimension that does not occur yet in the list, this can be performed by entering the name of the new dimension in the upper edit element and pressing the button *AddDimension*. The use of button *Aggregate-Dimension* allows the inclusion of a dimension specified in the upper edit element into the list of aggregated dimensions.

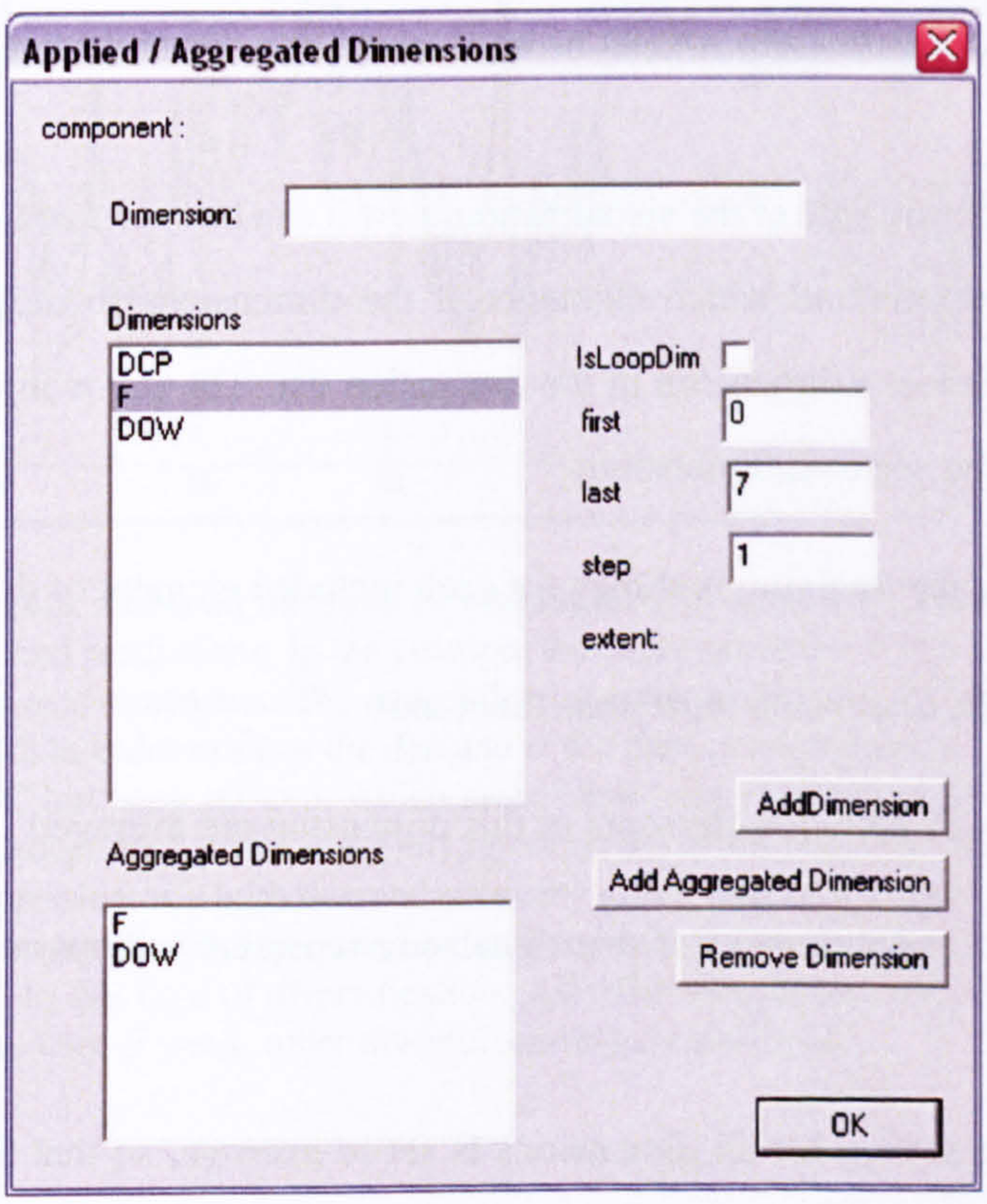

Fig. 70: Specification of handling of dimensions for a calculation.

B.3.8 Data Visualisation

The data visualisation view can be used in order to generate dynamic 2D-figures of any data included in a data cube in *Avanti*. In order to visualise data, the following has to be specified: (a) which data cubes to visualise, (b) which dimensions to use as x- axis and (c) to indicate restrictions to specific elements in relation to the existing dimensions.

The list box at the left hand side of the visualisation view contains a list of data cubes that should be visualised. *Select a data cube in the data view* and press the button *Add* in the visualisation view in order to add a data cube to this list. The buttons *Remove* and *Remove All* can be used in order to delete data cubes from the list.

At the right hand side of the visualisation view it can be specified how to handle different dimensions and which elements of the dimensions to include into the visualisation. Select a dimension in the dimension list. The following options can be chosen for the selected dimension:

- *loop:* a separate figure is shown for each included element of this dimension
- *x-axis:* the dimension represents the x-axis
- *average:* all included elements of this dimension are averaged
- *separate:* each element of this dimension represents a separate line in the figure

The default setting for all dimensions is set to *average*, so that the user only needs to indicate dimensions which should be handled differently.

For each dimension it is possible to decide whether to include all elements or a single element. The choice and the selection of the element to be visualised can be made in the value section of the data visualisation view.

After having included all data cubes that should be visualised and having specified the handling of the dimensions, the button *Show* can be pressed in order to

generate the visualisation. Figure 71 shows an example of the generated display. Pressing the button *Save* allows the saving of the visualised data values into an Excel compatible file.

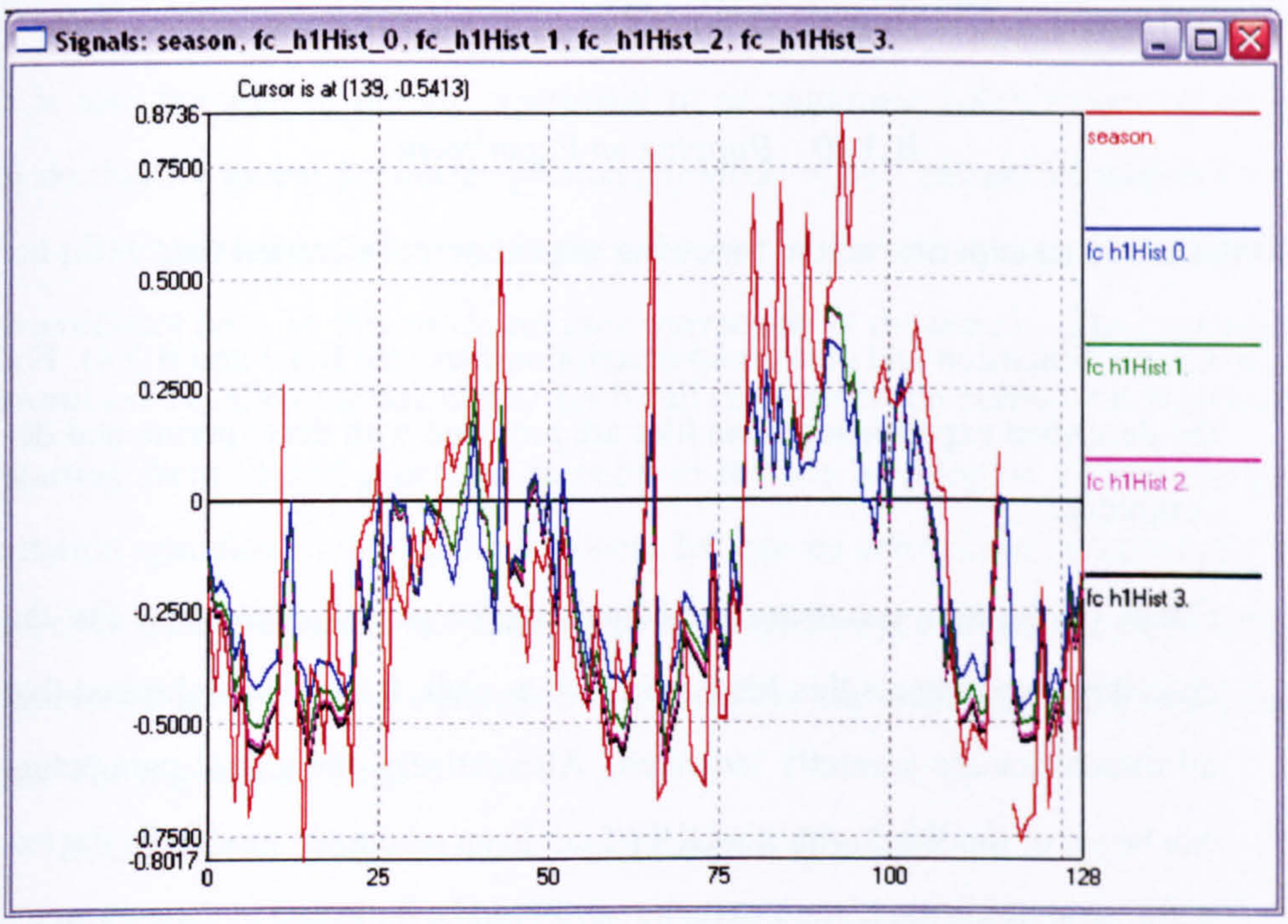

Fig. 71: The figure shows an example of visualisation of seasonal factors together with diversified predictions. In the example the departure week (DW) has been chosen as an x-axis dimension. The data collection point (dimension DCP) has been set to value 22 in order to show the demand at the time of the departure. The dimension DCPFC has been set to 5, which means that only predictions generated 70 days prior to departure are shown. Only fareclass (F) 16 has been selected in order to get an impression of a high demand economy class. Diversification dimension DIV1 has been handled as *separate*, so that a separate line is drawn for each prediction related to this type of diversification. All other dimensions (like Fareclass, Point of Sale, Day of week, other diversifications) are averaged.

B.3.9 Specification of a Diversification

A diversification of a parameter value or a level of data aggregation can be specified in the menu with *Control/Diversify Calculation*. A dialogue appears which enables the selection of the concerned components and setting the range of the diversified parameter value or choosing the diversified level.

Diversification specifications can also be loaded from files. Choose *File/Open Diversification List* in order to load a predefined diversification. For the experiments described in this thesis all the required diversification files are provided with the experiments.

B.3.10 Running an Experiment

In order to run an experiment, the following steps have to be carried out:

- Create dimension and data group description files (see B.3.3 and B.3.4). For the described experiments these files are provided with the experimental descriptions.

- Create or choose a parameter file containing the global parameters. For the described experiments this file is provided as well. It should be checked that all directories are correctly indicated. Alternatively, the global parameters can be set or modified with the GUI (if not in an automatic mode). In the experimental mode the correct experiment should be indicated in the parameter file before starting *Avanti*.

- Create or load a component list (B.3.5). If all global parameters are set correctly, the correct file will be suggerred automatically for the experiments and the correct data group descriptions will be loaded.

- Diversify the calculation, if necessary (B.3.9). For the repetition of a described experiment, load the diversification description, the correct diversification file is suggerred automatically.

- Run the calculation. You can run the whole experiment by selecting *Component/RunAll* in the menu. It is also possible to run only parts of the calculation. Select a component in the component view and choose *Component/RunSelected* (only calculates this single component) or *Component/RunToSelected* (calculates all to the selected component).

- It is then possible to visualise the results using the visualisation view or to save a selected data cubes calling *file/ SaveDataCube* in the menu.

B.3.11 Processing Different Data Directories in an Automatic Mode

It is also possible to run an experiment in an automatic batch mode. In order to do that set global parameter pAutomaticMode ="on", pBachCalculation="on" and pBatchElement="batchlog" in the parameter.txt file and start the dos batch file *avantiBatch.bat*. In this mode no user interaction is requested. The executable *avanti.exe* is called various times. Each call determines the current batch element (starting from 0) and generates an entry in the file batchlog.txt so that the calculation specified in the used component list can be carried out successively for different data directories. For such a type of calculation it is requested that the data directories related to the different batch elements are indicated in the parameter file. The requested format is

$BATCH < number >= " < directoryname > "$

(for instance BATCH1="OD1-ODO0"). For the experiments related to this thesis the correct batch element specifications are provided with the parameter.txt file. Note that *avantiBatch.bat* represents a very simple add-on to *Avanti* considering only the needs for the experiments related to this thesis. If, for instance, the number of batch element changes, *avantiBatch.bat* has to be adapted to the new number of batch elements.

B.4 Description of Applied Calculation Components

B.4.1 Component FILE_INTERFACE

Brief

name of the component	FILE_INTERFACE
type of the component	Interface
short description	loading and saving of data cubes

parameter	type	description
pCubesToLoad	string	comma separated names of the data cubes to be loaded, example: "bkg,avail"
pCubesToSave	string	comma separated names of the data cubes to be saved
pApplDimInFile	string	name of a dimension for which the elements should be represented in separate columns, can also be "UNDEFINED"

Detailed Description

The component loads and saves all data cubes indicated by the parameters *pCubesToLoad* and *pCubesToSave*. The structure of the data files should be as described in Table 2. In this example dimension "DCP" is the dimension indicated by parameter *pApplDimInFile*. Concrete examples for input data files can be found in the experimental data provided with the thesis.

B.4.2 Component UNCONSTRAINING

Brief

name of the component	UNCONSTRAINING
type of the component	Preprocessing
short description	calculates unconstrained data based on given constrained data, a historical estimation (reference) of the unconstrained behaviour and constraining information

Data Signals and Parameters

signal	input/ output	description
act	isInput	constrained data
ref	isInput	historical estimation of unconstrained behaviour
avail	isInput	constraining information, 0 = "open", 1 = "closed"
ucAct	isResult	unconstrained data
ucOffset	isResult	offset generated by the unconstraining

All data signals are expected to represent the situation at different data collection points τ.

Detailed Description

The general problem and idea of unconstraining has been described in Section 2.2.2. The objective of this calculation component is to approximate the demand lost in case of closed fareclasses. The lost parts are approximated based on an historical approximation of unconstrained behaviour.

Availability information is interpreted as punctual measurements. This means that if in a data collection point τ_1 an "closed" indicator has been measured and in the following data collection point τ_2 an "open" indicator has been measured, it is not clear at which moment the fareclass has been closed. It has therefore to be expected that also parts of the data in τ_2 have been lost. The unconstraining procedure handles this effects in interpreting all data collection points as containing incomplete data for which the availability of the current or the previous data collection point is "closed". Another effect of punctually measured availability information is that even for data collection points indicated as "closed" the fareclass could have been open for a certain period of time. It is then possible that nonzero

data is measured even in a "closed" fareclass. The unconstraining procedure has therefore to consider the data measured during closed periods.

The unconstraining algorithm consists of the following steps:

1. Determine an incomplete flag signal $isInc$:

   $\tau = 0$: $isInc_\tau = avail_\tau$

   $\tau > 0$: $isInc_\tau = max(avail_\tau, avail_{\tau-1})$

2. Approximate the lost data parts for all τ:

   $isInc_\tau = 0$: $ucOffset_\tau = 0$

   $isInc_\tau = 1$: $ucOffset_\tau = max(ref_\tau - act_\tau, 0)$

3. Calculate $ucAct_\tau = act_\tau + ucOffset_\tau$.

B.4.3 Component DATA_DECOMPOSITION

Brief

name of the component	DATA_DECOMPOSITION
type of the component	Preprocessing
short description	splits data into an absolute base component and a deviation component based on historical estimations for these components

Data Signals and Parameters

signal	input/ output	description
act	isInput	total data to be decomposed
refBase	isInput	estimation of the base component based on historical data
refDeviation	isInput	estimation of the deviation component based on historical data
base	isOutput	splitted data base component
deviation	isOutput	splitted data deviation component (factor, no deviation represented as 0)
parameter	**type**	**description**
pRestrictRelDeviation	bool	indicates is the resulting deviation should be restricted
pRelDeviation-LowerLimit	float	lower limit for resulting deviation
pRelDeviation-UpperLimit	float	upper limit for resulting deviation
pBaseLowerLimit	float	lower limit for resulting base value
pBaseUpperLimit	float	upper limit for resulting base value
pRelDeviation-LimitDumping	float	symmetric dumping of limits

Detailed Description

First the data decomposition generates an estimation for the base component based on two types of estimates: (a) the current data (act) with impact of expected deviations (refDeviation) eliminated; and (b) the expected behaviour learned from the history (refBase). As the base component is generally more stable than the devia-

tion component, the two estimates are combined with 30% impact given to (a) and 70% given to (b). In a second step an estimation for the deviation component is generated based on the data and the estimation for the base component.

The algorithm works as follows:

1. generate estimate ${}^1base = refBase$

2. generate estimate ${}^2base$:

   $act = UNDEFINED$: ${}^2base = UNDEFINED$

   $refDeviation = UNDEFINED$: ${}^2base = UNDEFINED$

   $refDeviation <= -0.95$: ${}^2base = UNDEFINED$

   $refDeviation > -0.95$: ${}^2base = \frac{act}{refDeviation+1}$

3. generate estimate *base*:

   $refDeviation = UNDEFINED$: $base = {}^1 base$

   ${}^2base = UNDEFINED$: $base = UNDEFINED$

   else $base = 0.7{}^1base + 0.3{}^2base$

4. assert range[pBaseLowerLimit,pBaseUpperLimit] of base, if value outside range, set to limit

5. generate estimate *deviation*:

   $act = UNDEFINED$: $deviation = UNDEFINED$

   $base = UNDEFINED$: $deviation = 0$

   $base < 0.05$: $deviation = 0$

   else $deviation = \frac{act}{base} - 1$

6. restrict deviation to range

   [pRelDeviationLowerLimit*pRelDeviationLimitDumping,

   pRelDeviationUpperLimit*pRelDeviationLimitDumping]

B.4.4 Component DATA_SMOOTHING

Brief

name of the component	DATA_SMOOTHING
type of the component	Preprocessing
short description	smoothing of data via weighted moving average

Data Signals and Parameters

signal	input/ output	description
act	isBoth	data used for smoothing
parameter	**type**	**description**
pSizeNeighbourhood	float	number of neighboured values to be used for smoothing
pOwnImpact	float	impact (weight) of the value to be smmothed

Detailed Description

The calculation component realises a smoothing of data via weighted moving average. The result is calculated by

$$^{result}act_i = \sum_{j=-pSizeNeighbourhood}^{pSizeNeighbourhood} w_j * act_{i+j} \tag{B.1}$$

with $w_0 = pOwnImpact$ and $w_j = \frac{1-pOwnImpact}{2*pSizeNeighbourhood}\ \forall j \neq 0$.

B.4.5 Component HB_EXP

Brief

name of the component	HB_EXP
type of the component	History Building
short description	realises history building via simple exponential smoothing

Data Signals and Parameters

signal	input/ output	description
act	isInput	data used for smoothing
ref	isBoth	smoothed result

The data is expected to contain values in relation to one dimension of equidistant time intervals (for our application the departure week).

parameter	type	description
pSmoothingFactor	float	smoothing factor indicating the impact of new data
pCycleSize	float	size of a cycle
pHistCycles	float	number of cycles used for initialisation
pRefMin	float	lower limit for the smoothed value
pRefMax	float	upper limit for the smoothed value

Detailed Description

Calculation component HB_EXP realises a simple exponential smoothing [Brown 63] of data given in input signal act. The result signal ref contains the smoothed result of the data learned until the moment when the data occurs. The smoothing factor

is provided by parameter pSmoothingFactor. It is possible to indicate cycles. This enables the learning of periodic behaviour and can be used, for instance, to learn separate smoothed values per calendar week (pCycleSize=53, smoothing values $0, 53, 106, \ldots$, values $1, 54, 107, \ldots$ and so on). It is also possible to indicate an initialisation period. It is assumed that for this period all data is known from the beginning. The learned values for this period contain the information of the whole initialisation period. The number of cycles to be used as the initialisation period is indicated by parameter pHistCycles. A lower and an upper limit for the resulting smoothed values are given by parameters pRefMin and pRefMax. If the learned values are outside of the range indicated by these two parameters, the values are set to the indicated limit.

B.4.6 Component HB_BROWN

Brief

name of the component	HB_BROWN
type of the component	History Building
short description	realises history building via brown model

Data Signals and Parameters

signal	input/ output	description
act	isInput	data used for smoothing
ref	isBoth	smoothed result
trend	isBoth	smoothed trend

The data is expected to contain values in relation to one dimension of equidistant time intervals (for our application the departure week).

parameter	type	description
pSmoothingFactor	float	smoothing factor indicating the impact of new data
pCycleSize	float	size of a cycle
pHistCycles	float	number of cycles used for initialisation
pRefMin	float	lower limit for the smoothed base value
pRefMax	float	upper limit for the smoothed base value
pTrendMin	float	lower limit for the smoothed trend value
pTrendMax	float	upper limit for the smoothed value

Detailed Description

The component carries out a history building using the Brown method [Brown 63] without any seasonal components. The interpretation of data and cycles is similar to the one described for component HB_EXP. Additionally to component HB_EXP, smoothed trend values are learned. Limits for these values can be defined by parameters pTrendMin and pTrendMax.

B.4.7 Component HB_REGR

Brief

name of the component	HB_REGR
type of the component	History Building
short description	realises history building via linear regression

Data Signals and Parameters

signal	input/ output	description
act	isInput	data used for smoothing
ref	isBoth	smoothed result
trend	isBoth	smoothed trend

The data is expected to contain values in relation to one dimension of equidistant time intervals (for our application the departure week).

parameter	type	description
pCycleSize	float	size of a cycle
pHistCycles	float	number of cycles used for initialisation
pRefMin	float	lower limit for the smoothed base value
pRefMax	float	upper limit for the smoothed base value
pTrendMin	float	lower limit for the smoothed trend value
pTrendMax	float	upper limit for the smoothed value

Detailed Description

Component HB_REGR carries out history building via linear regression. The interpretation of data and cycles is similar to the one described for component HB_BROWN.

B.4.8 Component FC_ATTR

Brief

name of the component	FC_ATTR
type of the component	Forecast
short description	forecast of the attractiveness

Data Signals and Parameters

signal	input/ output	description
act	isInput	current data y
ref	isInput	learned attractiveness base value ϕ_0^{attr}
trend	isInput	learned attractiveness trend value ϕ_1^{attr}
fc	isOutput	generated forecast
blockElemShift	isInput	number of time series intervals between a block element and the last block element

parameter	type	description
pUseTrend	bool	indicates if trend information is available
pDumpingTrend	bool	indicates if a dumping of the trend should by carried out
pAdaptation	bool	indicates if an adaptation to act should be carried out
pNbrBlockElems	float	number of block elements

Detailed Description

The component realises predictions for the attractiveness as described in (2.11) and (2.13). The base value ϕ_0^{attr} is expected to be contained in signal ref, the trend ϕ_1^{attr} is expected in signal trend. The basic prediction is provided by an estimation learned from historical data. This prediction can be represented as a single (constant) learned value comparable to model (2.11) (pUseTrend=false) or as a linear relationship comparable to model (2.13) (pUseTrend=true). A dumping of the trend is also possible. If parameter pDumpingTrend is true, instead of using

$$h_2^{attr}(x^{attr}, \phi^{attr}) = \phi_0^{attr} + \phi_1^{attr} * (t_d - t_p), \tag{B.2}$$

corresponding to (2.13), version

$$h_2^{attr}(x^{attr}, \phi^{attr}) = \phi_0^{attr} + \phi_1^{attr} * (log(t_d - t_p) + 1) \qquad \text{(B.3)}$$

reduces the effect of the trend for large values $t_d - t_p$.

The component generates predictions for different time series values to be predicted (in our case different departure weeks) going out from different process dates (dcps). The learned values (signals ref and trend) are expected in a manner that the values given at a departure week represent the value that has been learned *until that departure week.* The prediction considers only values which have been learned until the time of forecast generation. In order to be able to do this, the information has to be provided which data is known at a dcp or, in other words, how many time series intervals correspond to the distance between the dcp and the departure. This information is provided in signal blockElemShift. For instance, if the value of blockElemShift of dcp 1 is 26, this means that between dcp 1 and dcp 23 there is a time difference of 26 weeks. Therefore, if the prediction is generated at dcp 1, the values learned for the attractiveness at the final dcp are not allowed to use information of the previous 26 departure weeks.

After calculation of the basic prediction, an adaptation based on current bookings (act) is possible. If parameter pAdaptation is true, a simple additive adaptation is carried out. The forecast is corrected in a manner that for the historical dcps the part of the predicted attractiveness, which is expected to be already existing at this dcp, is replaced by the real values.

B.4.9 Component FC_LSB

Brief

name of the component	FC_LSB
type of the component	Forecast
short description	realises forecasts of the total demand corresponding to the method of the current system *Profit-Line.Yield/O&D*

Data Signals and Parameters

signal	input/ output	description
act	isInput	current data y
ref	isInput	learned attractiveness base value ϕ_0^{attr}
trend	isInput	learned attractiveness trend value ϕ_1^{attr}
relDeviation	isInput	learned seasonal factor ϕ_{cw}^{season}
avail	isInput	availability information
fc	isOutput	generated forecast
blockElemShift	isInput	number of time series intervals between a block element and the last block element

parameter	type	description
pAdaptToSeason	bool	indicates if seasonal information should be used
pUseTrend	bool	indicates if trend information is available
pDumpingTrend	bool	indicates if a dumping of the trend should by carried out
pAdaptationKind	float	type of adaptation to current data, 0: additive adaptation, 1: linear adaptation, 2: no adaptation
pNbrBlockElems	float	number of block elements

Detailed Description

The component calculates forecasts of the total demand corresponding to the method of the current system *ProfitLine.Yield/O&D*. The method is described in Section 2.2. The structure of the input and result signals corresponds to the one described for component FC_ATTR.

First, the component realises an adaptation to the expected seasonal behaviour (see Section 2.2.6). Then an adaptation to the current booking values is carried out similar to component FC_ATTR.

B.4.10 Component FC_SEASON

Brief

name of the component	FC_SEASON
type of the component	Forecast
short description	forecasting of the seasonal component

Data Signals and Parameters

signal	input/ output	description
realRelDeviation	isInput	current seasonal deviation (low level) $y^{season}_{t_{d,\tau}}$
smoothedRelDeviation	isInput	smoothed seasonal deviation (potentially higher level) $y^{season}_{t_{d,\tau}}$
histRelDeviation	isInput	learned seasonal factors $\widehat{y}_{cw}$
ref	isInput	current estimation of the attractiveness $\widehat{y}^{attr}_{t_{d,\tau}}$
fc	isOutput	generated forecast $h^{season}(x, \phi)$

All data is expected to be given for all data collection points.

parameter	type	description
pRestrictRelDeviation	bool	indicates if the used seasonal factors should be restricted
pMethod	float	method 1 to 3 (2.15)(2.17)(2.18)
pRelDeviation-LowerLimit	float	parameter ϕ_{low}
pRelDeviation-UpperLimit	float	parameter ϕ_{high}
pRelDevLimit-Dumping	float	symmetric dumping of parameters ϕ_{low} and ϕ_{high}

Detailed Description

The component calculates predictions corresponding to $h^{season}_1(x, \phi)$ (2.15), $h^{season}_2(x, \phi)$ (2.17) and $h^{season}_3(x, \phi)$ (2.18). The unconstrained demand information $y^{unc}_{t_{d,\tau}}$ used in equations (2.15),(2.17) and (2.18) is calculated based on the given

estimation of the attractiveness $\widehat{y}^{attr}_{t_{d,\tau}}$ and the current seasonal deviation $y^{season}_{t_{d,\tau}}$ by

$$y^{unc}_{t_{d,\tau}} = \widehat{y}^{attr}_{t_{d,\tau}} * (1 + y^{season}_{t_{d,\tau}}). \tag{B.4}$$

Parameters pRelDeviationLowerLimit and pRelDeviationUpperLimit are symmetrically dumped by parameter pRelDeviationLimitDumping. This means that it is

$$\phi_{low} = pRelDeviationLowerLimit * pRelDeviationLimitDumping \tag{B.5}$$

and

$$\phi_{high} = pRelDeviationUpperLimit * pRelDeviationLimitDumping. \tag{B.6}$$

This application of such a dumping parameter enables a diversification of both limits by diversification of a single parameter.

B.4.11 Component COMBINING_ADD_PARTS

Brief

name of the component	COMBINING_ADD_PARTS
type of the component	Forecast Combination
short description	fusion of decomposed forecasts

Data Signals and Parameters

signal	input/ output	description
fcBasis	isInput	forecast for the basis component
fcRelDev	isInput	forecast for the deviation component
fcResult	isOutput	combined result

Detailed Description

The component realises a fusion of forecasts related to two types of components : (a) a basis component represented in absolute values; and (b) a deviation component represented as factors (with 0 meaning no deviation).

The component calculates

$$fcResult = fcBasis * (1 + fcRelDev) \tag{B.7}$$

for each element of the input signals.

B.4.12 Component HB_LINEAR_COMBINATION

Brief

name of the component	HB_LINEAR_COMBINATION
type of the component	Forecast Combination
short description	learning of linear combination weights

Data Signals and Parameters

signal	input/ output	description
err	isInput	forecast errors e
act	isInput	target values y
weight	isBoth	learned combination weights
offset	isBoth	learned offset (only method F^{ols})

parameter	type	description
pMethod	float	linear combination model
pTrimmingMaxNbrFc	float	trimming: maximal number of input forecasts
pTrimmingMax- VarRatio	float	trimming: maximal error variance ratio

Detailed Description

This component offers the functionality to learn linear combination weights with different linear combination models. The model to be used can be specified with parameter pMethod. Accepted Values for parameter pMethod:

- 0:F^{ac} (see Section 3.2.3),
- 1:F^{outp} (see Section 3.2.4),
- 2:F^{var} (see Section 3.2.5),
- 3:F^{opt} (see Section 3.2.5) and
- 4:F^{ols} (see Section 3.2.6).

The input signal err contains the prediction errors ${}^{m}e_t$. It is expected that this signal contains first the errors related to method m_0 for all predictions generated for different time indices t, then related to method m_1 and so on. The number of methods M is derived from the size of the signal weights. Signal act contains the values y_t.

The calculation results are returned in a filled weight signal. If all forecasts related to a method m are default, the resulting weights are default as well. Signal offset is filled with nonzero values only in case of pMethod=F^{ols}. In this case, it contains the absolute offset learned by this model.

The input forecasts are trimmed depending on parameters pTrimmingMaxNbrFc and pTrimmingMaxVarRatio. They are ordered corresponding to their error

variance. If parameter pTrimmingMaxNbrFc is set to a value greater than zero, only the best pTrimmingMaxNbrFc forecasts are included into the combination procedure (the resulting weights of the other forecasts are default. If parameter pTrimmingMaxVarRatio is larger than 1, all forecasts with an error variance larger than pTrimmingMaxVarRatio times the error variance of the best forecast are excluded from the combination as well.

A special fallback solution has been implemented for model F^{ols}. In case of an insufficient number of input forecasts per forecast method m (number of valid rows $\geq$ 2* M) the model automatically switches to model F^{av}. One reason for insufficient numbers of input forecasts can be an incomplete flight schedule (cancelled flights).

B.4.13 Component HB_LINEAR_COMBINATION_STRUCTURE

Brief

name of the component	HB_LINEAR_COMBINATION_STRUCTURE
type of the component	Forecast Combination
short description	determination of linear combination weights using predefined or evolved multi step combination structures

Data Signals and Parameters

signal	input/ output	description
err	isInput	forecast errors e
act	isInput	target values y
weight	isBoth	learned combination weights
offset	isBoth	learned offset (only method F^{ols})

parameter	type	description
pCrossover	float	type of crossover
pMutation	float	type of crossover
pFitness	float	fitness measure
pInitMode	float	initialisation mode
pMaxStep	float	maximal number of combination steps
pMethod	float	linear combination model
pTrimmingMaxNbrFc	float	trimming: maximal number of input forecasts
pTrimmingMax-VarRatio	float	trimming: maximal error variance ratio
pDimensions	string	dimensions representing the forecast generation space
pOrder	string	order of the dimensions used for pooling

Detailed Description

This component realises the generation and evolution of different types of combination structures in order to determine linear combination weights. The structures which can be generated or evolved correspond to those described in Chapters 6 and 7.

The structures differ concerning structure, initialisation, crossover and mutation as well as concerning the applied combination functions and trimming restric-

tions.

If parameters pDimensions and pOrder are both given, one single combination structure is generated and applied. In the other cases, a population of structures is generated and an evolution is carried out. The resulting weights correspond to those generated by the best performing structure.

Details in relation to the evolution

The evolution is carried out on a population of eight chromosomes. The only exception is related to the crossover described in Section 7.3.2 which operates only on a population with a single chromosome. The number of crossovers is restricted to hundred, the evolution is also stopped if the fitness did not improve (with tolerance 10^{-6}) over more than fifty generations. In each step of the evolution the parents for crossover are selected first. The element with the worst fitness dies. Then the crossover is carried out, and mutation follows.

Crossover

Three types of crossover are supported. The type to use is indicated by parameter pCrossover. Type 0 (pCrossover=0) and 1 carry out the two types of child generation as described in Section 7.3.2. Type 2 represents the dimension independent crossover described in Section 7.2.3.

Mutation

Two different types of mutation are used. The first type of mutation corresponds to the one described in Section 7.3.2. It is used in order to calibrate the trimming percentage. If parameter pMutation=0 is set the adaptation is carried out per combination procedure. If parameter pMutation=1 the parameter is mutated in a global manner (the same value for all combination procedures). The second type of mutation (pMutation=2) is used in order to manipulate the input forecasts and the combination model (if not predefined in parameter pMethod) as described in Section 7.2.3.

Fitness Calculation

Depending on parameter pFitness, the fitness is calculated corresponding to

equations (7.2),(7.3) or (7.4). It is calculated based on out of sample predictions. The first half of the elements are used in order to determine the linear combination weights, the remaining elements are used for fitness evaluation.

Generation of an Initial Population

Three types of structure initialisation are used. Parameter pInitMode specifies which type to use.

Type 0 uses the information about the extents of the dimensions of the forecast generation space (pDimensions). The generated structures correspond to the ones described in Section 7.3.1. The order of the dimensions is determined randomly. If the maximum number of steps of the structures is restricted by parameter pMaxStep, different dimensions are clustered in order to fulfil that restriction. If there are, for instance, 4 dimensions given and pMaxStep is two, dimension 1 and 2 and dimension 3 and 4 are clustered.

Type 1 carries out a random initialisation as described in Section 7.2.2. The generated structures contain up to pMaxStep steps each containing between two and five combination procedures. The input forecasts for the combination procedures at step 0 are selected randomly as well.

Type 2 corresponds to structures generated by the pooling approach of Aiolfi and Timmermann. The algorithm is based on k-means clustering, it is described in Section 6.2.1. In the implementation in this Thesis the number of determined clusters is predefined to 4 clusters.

For all types of structure initialisation the parameters for the generated combination procedures are provided by parameters pMethod, pTrimmingMaxNbrFc and pTrimmingMaxVarRatio. If parameter pMethod of this component is set to -1, this indicates that the method to be used is not restricted and should be evolved. In this case, the initial setting in the combination procedures is 2 (corresponding to F^{var}), the parameter can then be modified between 0 (F^{av}), 1 (F^{outp}), 2 (F^{var}) and 3 (F^{opt}) by mutation.

Results of the component

The component calculates the linear combination weights and returns the weight to be used per input prediction. This means that in order to carry out the combination on out of sample data in order to generate predictions, it is not necessary any more to know the learned combination structure.

In case of an evolution, the component also generates two files (written into the result directory). The first file is called *element.dat*. It contains the elements of the best performing evolved combination structure. The second file *performance_graph.dat* contains the fitness of the evolved elements and shows the development of the fitness. It first contains the (ordered) fitness of the initial population. Then in each crossover step the performance of the generated child is added. As a last element the fitness of the structure which is considered as the best performing one at the end of the evolution is added.

B.4.14 Component LINEAR_COMBINATION

Brief

name of the component	LINEAR_COMBINATION
type of the component	Forecast Combination
short description	carries out a linear combination of forecasts

Data Signals and Parameters

signal	input/ output	description
fcInput	isInput	input forecasts ${}^{m}\widehat{y}$
fcCombined	isOutput	combined forecast ${}^{comb}\widehat{y}$
weight	isInput	combination weights w_m
offset	isInput	combination offset w_{M+1}

Detailed Description

The component carries out a linear combination of forecasts corresponding to equation 3.1. Input signal fcInput is expected to contain the different input forecasts ${}^m\widehat{y}$, the weights w_m are expected to be contained in input signal weight. If input signal offset does not contain the "UNDEFINED" indicator (represented as float value -1000) the extended version of linear combination including an offset (see 3.20) is carried out.

B.4.15 Component VALID_FC_REF

Brief

name of the component	VALID_FC_REF
type of the component	Validation
short description	calculates the (absolute) forecast error

Data Signals and Parameters

signal	input/ output	description
act	isInput	predicted target y
fc	isInput	forecast $\widehat{y}$
err	isOutput	forecast error e
parameter	**type**	**description**
pAbsError	bool	indicates if the error should be represented as an absolute error

Detailed Description

If parameter pAbsErrorThe = false the component calculates

$$err = fc - act. \tag{B.8}$$

In case of parameter pAbsErrorThe = true it calculates

$$err = |fc - act|. \tag{B.9}$$

B.4.16 Component ERROR_COVAR

Brief

name of the component	ERROR_COVAR
type of the component	Validation
short description	calculates error (co)variances

Data Signals and Parameters

signal	input/ output	description
err	isInput	forecast error
covar	isBoth	(co)variance of the forecast error

parameter	type	description
pCalcMad	bool	the mean absolute deviation should be calculated
pNbrElemsToAggr	float	number of values to be added before calculating the absolute error value

Detailed Description

The component calculates the mean absolute deviation as well as error covariances of input forecasts. The input signal err contains the error e related to each single predicted element. The result is the error covariance matrix (pCalcMad =false) or a mean absolute deviation vector (pCalcMad = true).

The number of input forecasts M is indicated by the size of signal covar. If parameter pCalcMad is true, M corresponds to the size of signal covar. If parameter

pCalcMad is false, M corresponds to the square root of the size of signal covar. Signal err is expected to contain first all the errors ${}^{1}e$ generated by forecast ${}^{1}\widehat{y}$, then the errors ${}^{2}e$ generated by forecast ${}^{2}\widehat{y}$ and so on. The number of predicted elements is determined by the size of signal err using the information about the number of input forecasts M. Before carrying out the error calculation, the values of each block of pNbrElemsToAggr elements are added. Then the number of input forecasts is determined in dividing the total number of elements by the determined number of input forecasts. Values larger than 1 for parameter pNbrElemsToAggr can be used if the data is available at a finer level than the error has to be calculated. If there are for instance forecast values concerning each point of sale available, but the error should be calculated in relation to the total demand of all point of sales, the input signal err can contain data containing the errors of each point of sale separately and parameter pNbrElemsToAggr contains the number of point of sales. In this case the data is first aggregated to the total demand level, then the error values are calculated.

Before determining the error covariances an outlier detection is carried out in order to remove extreme errors. All errors which differ from the average value by more than 1.5 times the standard deviation are set to the corresponding range limits. Additionally, all errors greater than 10 or smaller than -10 are set to the limits.

If parameter pCalcMad is true, the result signal contains the absolute error of each (aggregated) forecast ${}^{m}\delta_{e}^{2}$. If parameter pCalcMad is false, the result signal contains error covariances ${}^{1,1}\rho_{e}, {}^{1,2}\rho_{e}, \ldots, {}^{1,M}\rho_{e}, {}^{2,1}\rho_{e}, {}^{2,2}\rho_{e}, \ldots, {}^{2,M}\rho_{e}, \ldots, {}^{M,1}\rho_{e}, {}^{M,2}\rho_{e}, \ldots, {}^{M,M}\rho_{e}$.

B.5 Description of Dimensions and Data Cubes

B.5.1 Dimensions

The following dimensions have been defined (for details see Appendix A):

dimension	extent	description
F	20	fareclass
POS	3	point of sale
ODO	1	odo (extent 1 as calculation is carried out per ODO)
DCP	23	data collection point
DOW	7	day of week
DW	129	departure week
DCPFC	23	dcp of forecast generation
FCNR	7	number generated forecast (experiment 3)
FCNR2	7	number generated forecast (experiment 3)
COMB	5	linear combination model
DIV1	4	diversification of parameter pRelDeviation-LimitDumping
DIV2	2	diversification of model for seasonal prediction (0=$h_1^{season}(x,\phi)$, 1=$h_3^{season}(x,\phi)$)
DIV3	2	diversification of learning level Fareclass versus Compartment
DIV4	2	diversification of learning level per DOW or over all DOW
STR	8	multi level combination structure (experiment 6)

B.5.2 Used Data Groups

The following table summarises the defined data groups.

data group	description
input_group	given input data like bookings and availability information
input_decomposed_group	decomposed input data
learning_attr_group	learned parameters for the attractiveness component
learning_season_group	learned parameters for the seasonal component
learning_lin_comb_group	learned linear combination weights
forecast_attr_group	forecasts of the attractiveness component
forecast_season_group	forecasts of the seasonal component
forecast_group	total demand forecasts
validation_group	validation related data cubes

B.5.3 Used Data Cubes

In the following the most relevant data cubes are described per data group. The indicated dimensions represent only examples, the specification can vary between different experiments.

input_group	dimensions	description
bkg	POS F DOW ODO CW DCP	booking values
avail	POS F DOW ODO CW DCP	availability information
blockElemShift	DCPFC	number of weeks contained in a data collection point τ
ucBkg	POS F DOW ODO CW DCP	unconstrained booking value
ucOffset	POS F DOW ODO CW DCP	unconstraining offset

input_decomposed_group	dimensions	description
season	POS F DOW ODO CW DCP	seasonal factors at the low level
seasonPrepared	POS F DOW ODO CW DCP	restricted seasonal factors used for predictions
seasonSmoothed	POS F DOW ODO CW DCP	restricted and smoothed seasonal factors used for learning
attr	POS F DOW ODO CW DCP	attractiveness

learning_attr_group	dimensions	description
phi0_h1ExpSm	POS F DOW ODO CW DCP	learned attractiveness via simple exponential smoothing, parameter $\widehat{\phi_0}$ of $h_1^{attr}(x, \phi)$ (2.11)
phi0_h2Brown	POS F DOW ODO CW DCP	learned attractiveness via brown model, parameter $\widehat{\phi_0}$ of $h_2^{attr}(x, \phi)$ (2.13)
phi1_h2Brown	POS F DOW ODO CW DCP	learned attractiveness via brown model, parameter $\widehat{\phi_1}$ of $h_2^{attr}(x, \phi)$ (2.13)
phi0_h2Regr	POS F DOW ODO CW DCP	learned attractiveness via linear regression, parameter $\widehat{\phi_0}$ of $h_2^{attr}(x, \phi)$ (2.13)
phi1_h2Regr	POS F DOW ODO CW DCP	learned attractiveness via linear regression, parameter $\widehat{\phi_1}$ of $h_2^{attr}(x, \phi)$ (2.13)

learning_season_group	dimensions	description
phi0_h1Hist	POS F DOW ODO CW DCP	learned seasonal factors, parameters $\widehat{\phi}$ of $h_1^{season}(x, \phi)$ (2.15)

forecast_attr_group	dimensions	description
fc_h1ExpSm	POS F DOW CW ODO DCPFC	forecast $h_1^{attr}(x, \widehat{\phi})$ (exponential smoothing model 2.11)
fc_h2Brown	POS F DOW CW ODO DCPFC	forecast $h_2^{attr}(x, \widehat{\phi})$ (Brown model 2.13)
fc_h2Regr	POS F DOW CW ODO DCPFC	forecast $h_2^{attr}(x, \widehat{\phi})$ (linear regr. 2.13)

forecast_season_group	dimensions	description
fc_h1Hist	POS F DOW CW ODO DCPFC	forecast $h_1^{season}(x, \widehat{\phi})$ (hist. model 2.15)
fc_h2Add	POS F DOW CW ODO DCPFC	forecast $h_2^{season}(x, \widehat{\phi})$ (add. model 2.17)
fc_h3Mult	POS F DOW CW ODO DCPFC	forecast $h_3^{season}(x, \widehat{\phi})$ (mult. model 2.18)
fc_comb	POS F DOW CW ODO DCPFC	combined seasonal forecast

forecast_group	dimensions	description
fc_input	POS F DOW CW ODO DCPFC FCNR	individual forecasts ${}^{m}\widehat{y}$ used as inputs for the combination
fc_compare	POS F DOW CW ODO DCPFC	best individual forecasts ${}^{0}\widehat{y}$
fc_combined	POS F DOW CW ODO DCPFC	combined forecast ${}^{comb}\widehat{y}$

learning_lin_comb_group	dimensions	description
lin_comb_weight	POS F DOW ODO DCPFC FCNR COMB	linear combination weights w_m
lin_comb_offset	POS F DOW ODO DCPFC COMB	offset linear combination (differs from zero only in case of combination model F^{ols})

validation_group	dimensions	description
err_h1Hist	POS F DOW CW ODO DCPFC FCNR	deviation ${}^{m}\widehat{y} - y$
err_combined_bias	POS F DOW CW ODO DCPFC COMB	deviation ${}^{comb}\widehat{y} - y$
var_low	POS F DOW ODO DCPFC FCNR	error variance of the input forecasts at the low level
var_high	DOW ODO DCPFC FCNR	error variance of the input forecasts at the high level
var_combined_low	POS F DOW ODO DCPFC COMB	error variance of the combined forecast at the low level
var_combined_high	DOW ODO DCPFC COMB	error variance of the combined forecast at the high level

B.6 Experiments

B.6.1 Experiment1 : Determination of Basic Statistical Properties of the Data

Brief

name of the experiment	experiment1
short description	visualisation and statistics of input data

The objective of this experiment is the visualisation of input data. All available input data is loaded. It contains bookings (data cube bkg) and availability information (data cube avail). The input cube blockElemShift provides the information of how many calendar weeks correspond to a dcp. This information is used, for instance, in component FC_ATTR (see B.4.8) in order to avoid information being used for prediction which is not yet known at a given point of time.

Inputs and Results

input	description
bkg	booking values
avail	availability information
blockElemShift	number of weeks contained in a data collection point τ

Summary of the Calculation

The calculation contains only data loading functionality. The data can then be visualised in the visualisation view (see B.3.8). It is also possible to write basic statistical properties of the data into a file. In order to do this, select a data cube and then chose *File/Save Data Statistics* in the menu.

Figure 72 shows an example for a resulting statistics file. It contains basic statistical properties like average value or the number of default values in relation to each value of each dimension of the data cube (like for each fareclass, each point of sale, each dcp and so on).

Detailed Description of Applied Components

FILE_INTERFACE	
load booking and availability information	
cubes	
parameter	string: bkg,avail, UNDEFINED, DCP
applied dimensions	

Microsoft Excel - statistics_bkg.dat

Datei Bearbeiten Ansicht Einfügen Format Extras Daten Fenster ?

A2 = POS

	A	B	C	D	E	F	G	H
1	SUM	MEAN	STD	MIN	MAX	NBR_NONZERO	NBR_ZERO	NBR_DFT
2	POS							
3	201532	0.725833	3.22013	0	77	40233	237423	137724
4	157991	0.569017	2.02749	0	76	51492	226164	137724
5	157677	0.567886	2.4468	0	142	48307	229349	137724
6								
7								
8	F							
9	0	0	0	0	0	0	0	62307
10	0	0	0	0	0	0	0	62307
11	0	0	0	0	0	0	0	62307
12	27011	0.53995	1.71488	0	44	11280	38745	12282
13	7346	0.146847	0.597634	0	10	4366	45659	12282
14	2592	0.0518141	0.37879	0	10	1479	48546	12282
15	2013	0.0402399	0.282418	0	5	1268	48757	12282
16	656	0.0131134	0.229263	0	10	343	49682	12282
17	21004	0.41987	1.26031	0	25	9797	40228	12282
18	25004	0.49983	1.61243	0	41	9876	40149	12282
19	26267	0.525077	2.23702	0	56	8819	41206	12282
20	45811	0.915762	2.39236	0	51	14223	35802	12282
21	10284	0.205577	0.656614	0	8	6359	43666	12282
22	46105	0.921639	2.76413	0	76	14450	35575	12282
23	2551	0.0509945	0.539252	0	21	1185	48840	12282
24	42210	0.843778	2.59995	0	49	13044	36981	12282

statistics_bkg

Bereit NF

Fig. 72: Example for a data statistics file generated by *Avanti*.

FILE_INTERFACE	
load block element shift	
cubes	
parameter	string: blockElemShift, UNDEFINED, UNDEFINED
applied dimensions	

B.6.2 Experiment2 : Individual Forecast Calculation and Error Evaluation

Brief

name of the experiment	experiment2
short description	generation of 7 forecasts (see table 3) differing concerning the prediction of the attractiveness as well as concerning the season

The objective of the experiment is to experimentally compare the performance of 7 individual forecasts. The input forecasts differ concerning the attractiveness component (Section 2.2.5) as well as concerning the prediction of the seasonal behaviour (Section 2.2.6). For prediction of the attractiveness 3 methods are applied: a) a simple exponential smoothing approach $h_1^{attr}(x, \phi)$ (2.11), b) the brown model $h_2^{attr}(x, \phi)$ (see 2.13) and c) a linear regression model $h_2^{attr}(x, \phi)$ as described in (2.13). For prediction of the season 4 methods are applied: a) model $h_1^{season}(x, \phi)$ based on historically learned seasonal behaviour (2.15), b) and additive adaptation to the current behaviour $h_2^{season}(x, \phi)$ (2.17), c) a multiplicative adaptation to the current behaviour $h_3^{season}(x, \phi)$ (2.18) and finally d) a combined approach $h^{season}(x, \phi)$ in which we have already carried out a linear combination of a), b) and c) as described in (2.19). The experiment provides the individual forecasts as well as error variance and covariance information at the low level of forecasting and aggregated over Fareclasses and Point of Sales.

Inputs and Results

input	description
bkg	booking values
avail	availability information
blockElemShift	number of weeks contained in a data collection point τ

result	description
fc_Input	individual forecast used as input for the combination in later experiments
mad_low	error variance of the input forecasts at the low level
mad_high	error variance of the input forecasts at the high level

Summary of the Calculation

The calculation can be summarised in the following steps:

1. load the data
2. carry out unconstraining
3. decompose the input data
4. learn the attrativness
5. learn the historical seasonal behaviour over history weeks 0 to 52
6. learn the historical attractiveness over history weeks 0 to 52
7. generate the predictions for the attractiveness (all weeks)
8. generate the predictions for the seasonal behaviour (all weeks)
9. calculate the total demand forecasts (all weeks)
10. determine the individual forecast performance (all weeks)
11. save the results

Detailed Description of Applied Components

FILE_INTERFACE	
load booking and availability information	
cubes	
parameter	string: bkg,avail, UNDEFINED, DCP
applied dimensions	

FILE_INTERFACE	
load block element shift	
cubes	
parameter	string: blockElemShift, UNDEFINED, UNDEFINED
applied dimensions	

HB_EXP	
calculate first estimate for the attractiveness without consideration of unconstraining and seasonal effects	
cubes	bkg, phi0_h1ExpSm
parameter	float: 0.05,23,53,0,1000
applied dimensions	CW[appl,0,128],DCP[appl,0,22]

UNCONSTRAINING	
unconstrain the booking data	
cubes	bkg, phi0_h1ExpSm, avail, ucBkg, ucOffset
parameter	
applied dimensions	DCP[appl,0,22]

HB_EXP	
calculate second estimate for the attractiveness without consideration of seasonal effects	
cubes	ucBkg, phi0_h1ExpSm
parameter	float: 0.05,23,53,0,1000
applied dimensions	CW[appl,0,128],DCP[appl,0,22]

DATA_DECOMPOSITION	
calculate seasonal factors used for forecasting First/Business compartment	
cubes	ucBkg, phi0_h1ExpSm, phi0_h1Hist, attrPrepared, seasonPrepared
parameter	bool: 1, float: -0.5,6,0,1000,1
applied dimensions	DCP[appl,0,22],F[aggr,0,7],DOW[aggr,0,6]

DATA_DECOMPOSITION	
calculate seasonal factors used for forecasting Economy compartment	
cubes	ucBkg, phi0_h1ExpSm, phi0_h1Hist, attrPrepared, seasonPrepared
parameter	bool: 1, float: -0.5,6,0,1000,1
applied dimensions	DCP[appl,0,22],F[aggr,8,19],DOW[aggr,0,6]

DATA_DECOMPOSITION	
calculate first estimate seasonal factors First/Business compartment	
cubes	ucBkg, phi0_h1ExpSm, phi0_h1Hist, attrPrepared, seasonSmoothed
parameter	bool: 1, float: -0.5,3,0,1000,1
applied dimensions	DCP[appl,0,22],F[aggr,0,7],DOW[aggr,0,6]

DATA_DECOMPOSITION	
calculate first estimate seasonal factors Economy compartment	
cubes	ucBkg, phi0_h1ExpSm, phi0_h1Hist, attrPrepared, seasonSmoothed
parameter	bool: 1, float: -0.5,3,0,1000,1
applied dimensions	DCP[appl,0,22],F[aggr,8,19],DOW[aggr,0,6]

DATA_SMOOTHING	
smooth the determined seasonal factors	
cubes	seasonSmoothed
parameter	float: 5,0.1
applied dimensions	CW[appl,0,128]

HB_EXP	
learn seasonal behaviour	
cubes	seasonSmoothed, phi0_h1Hist
parameter	float: 0.6,53,1,-1,1000
applied dimensions	CW[appl,0,128]

DATA_DECOMPOSITION	
data decomposition under consideration of historical behaviour attractiveness and season	
cubes	ucBkg, phi0_h1ExpSm, phi0_h1Hist, attr, season
parameter	bool: 0, float: -0.5,3,0,1000,1
applied dimensions	DCP[appl,0,22]

HB_EXP	
learn attractiveness simple exponential smoothing	
cubes	attr, phi0_h1ExpSm
parameter	float: 0.1,23,53,0,1000
applied dimensions	CW[appl,0,128],DCP[appl,0,22]

HB_BROWN	
learn attractiveness brown model	
cubes	attr, phi0_h2Brown, phi1_h2Brown
parameter	float: 0.04,23,53,0,1000,-0.05,0.05
applied dimensions	CW[appl,0,128],DCP[appl,0,22]

HB_REGR	
learn attractiveness linear regression model	
cubes	attr, phi0_h2Regr, phi1_h2Regr
parameter	float: 23,53,0,1000,-0.1,0.1
applied dimensions	CW[appl,0,128],DCP[appl,0,22]

FC_ATTR	
predict attractiveness simple exponential smoothing model	
cubes	attr, phi0_h1ExpSm, phi1_h2Brown, phi0_h1Hist, avail, fc_h1ExpSm, blockElemShift
parameter	bool: 0, 0, 0, float: 0,23
applied dimensions	CW[appl,0,128],DCP[appl,0,22],DCPFC[appl,0,22]

FC_ATTR	
predict attractiveness brown model	
cubes	attr, phi0_h2Brown, phi1_h2Brown, phi0_h1Hist, avail, fc_h2Brown, blockElemShift
parameter	bool: 0, 1, 1, float: 0,23
applied dimensions	CW[appl,0,128],DCP[appl,0,22],DCPFC[appl,0,22]

FC_ATTR	
predict attractiveness linear regression model	
cubes	attr, phi0_h2Regr, phi1_h2Regr, phi0_h1Hist, avail, fc_h2Regr, blockElemShift
parameter	bool: 0, 1, 0, float: 0,23
applied dimensions	CW[appl,0,128],DCP[appl,0,22],DCPFC[appl,0,22]

FC_SEASON	
predict seasonal factors historical model	
cubes	season, seasonPrepared, phi0_h1Hist, attr, fc_h1ExpSm, fc_h1Hist
parameter	bool: 1, float: 1,-1,3,1
applied dimensions	DCP[appl,0,22],DCPFC[appl,0,22]

FC_SEASON	
predict seasonal factors additive adaptation	
cubes	season, seasonPrepared, phi0_h1Hist, attr, fc_h1ExpSm, fc_h2Add
parameter	bool: 1, float: 2,-0.5,2,1
applied dimensions	DCP[appl,0,22],DCPFC[appl,0,22]

FC_SEASON	
predict seasonal factors multiplicative adaptation	
cubes	season, seasonPrepared, phi0_h1Hist, attr, phi0_h1ExpSm, fc_h3Mult
parameter	bool: 1, float: 3,-0.5,2,1
applied dimensions	DCP[appl,0,22],DCPFC[appl,0,22]

FC_LSB	
predict total demand with the model used in the current system	
cubes	ucBkg, phi0_h1ExpSm, phi1_h2Brown, phi0_h1Hist, avail, fc_input, blockElemShift
parameter	bool: 1, 0, 0, float: 0,23
applied dimensions	CW[appl,0,128],DCP[appl,0,22],DCPFC[appl,0,22], FCNR[appl,0,0]

COMBINING_ADD_PARTS	
combine components of result forecast 1	
cubes	fc_h1ExpSm, fc_h3Mult, fc_input
parameter	
applied dimensions	FCNR[appl,1,1]

COMBINING_ADD_PARTS	
combine components of result forecast 2	
cubes	fc_h2Brown, fc_h3Mult, fc_input
parameter	
applied dimensions	FCNR[appl,2,2]

COMBINING_ADD_PARTS	
combine components of result forecast 3	
cubes	fc_h1ExpSm, fc_h2Add, fc_input
parameter	
applied dimensions	FCNR[appl,3,3]

COMBINING_ADD_PARTS	
combine components of result forecast 4	
cubes	fc_h2Regr, fc_h3Mult, fc_input
parameter	
applied dimensions	FCNR[appl,4,4]

COMBINING_ADD_PARTS	
combine components of result forecast 5	
cubes	fc_h2Regr, fc_h1Hist, fc_input
parameter	
applied dimensions	FCNR[appl,5,5]

COMBINING_ADD_PARTS	
combine components of result forecast 6	
cubes	fc_h2Regr, fc_h2Add, fc_input
parameter	
applied dimensions	FCNR[appl,6,6]

VALID_FC_REF	
calculate toal forecast errors	
cubes	ucBkg, fc_input, err_input_bias
parameter	bool: 0
applied dimensions	DCP[appl,22,22],DCPFC[appl,0,22]

ERROR_COVAR	
calculate mean absolute deviation low level	
cubes	err_input_bias, mad_low
parameter	bool: 1, float: 1
applied dimensions	CW[appl,93,128]

ERROR_COVAR	
calculate mean absolute deviation high level	
cubes	err_input_bias, mad_high
parameter	bool: 1, float: 60
applied dimensions	CW[appl,93,128],F[appl,0,19],POS[appl,0,2]

FILE_INTERFACE	
write results	
cubes	
parameter	string: UNDEFINED, mad_low, mad_high, DCPFC
applied dimensions	

B.6.3 Experiment3 : Combination of Forecasts calculated by Experiment 2

Brief

name of the experiment	experiment3
short description	combination of 7 forecasts (see table 3) differing concerning the prediction of the attractiveness as well as concerning the season by combination models F^{av}, F^{outp}, F^{var}, F^{opt} and F^{ols}

The objective of the experiment is to experimentally compare the performance of the 6 individual forecasts already described in the previous experiment with different combined versions. Five combination methods have been used for com-

bination of the six forecasts [1] F^{av}, F^{outp}, F^{var}, F^{opt},F^{ols} (see Section 3.2). The experiment provides the individual and combined forecasts as well as error variance and covariance information at the low level of forecasting and aggregated over Fareclasses and Point of Sales.

Inputs and Results

input	description
bkg	booking values
avail	availability information
blockElemShift	number of weeks contained in a data collection point τ

result	description
fc_input	individual forecast used as input for the combination
fc_combined	combined forecast
lin_comb_weight	linear combination weight
lin_comb_offset	offset linear combination
mad_low	error variance of the input forecasts at the low level
mad_high	error variance of the input forecasts at the high level
mad_combined_low	error variance of the combined forecast at the low level
mad_combined_high	error variance of the combined forecast at the high level

Summary of the Calculation

The calculation can be summarised in the following steps:

1. load the data

[1] Experiments of nonlinear methods F^{dyn} and F^{appr} (3.3) have been carried out as well but are not described in this experimental setup.

2. carry out unconstraining

3. decompose the input data

4. learn the attractiveness

5. learn the historical seasonal behaviour over history weeks 0 to 52

6. learn the historical attractiveness over history weeks 0 to 52

7. generate the predictions for the attractiveness (all weeks)

8. generate the predictions for the seasonal behaviour (all weeks)

9. calculate the total demand forecasts (all weeks)

10. determine the individual forecast performance (all weeks)

11. learn the combination weights based on weeks 53 to 92

12. combine the individual forecasts

13. determine the combined forecast performance for weeks 93 to 128

14. save the results

Detailed Description of Applied Components

FILE_INTERFACE	
load booking and availability information	
cubes	
parameter	string: bkg,avail, UNDEFINED, DCP
applied dimensions	

FILE_INTERFACE	
load block element shift	
cubes	
parameter	string: blockElemShift, UNDEFINED, UNDEFINED
applied dimensions	

HB_EXP	
calculate first estimate for the attractiveness without consideration of unconstraining and seasonal effects	
cubes	bkg, phi0_h1ExpSm
parameter	float: 0.05,23,53,0,1000
applied dimensions	CW[appl,0,128],DCP[appl,0,22]

UNCONSTRAINING	
unconstrain the booking data	
cubes	bkg, phi0_h1ExpSm, avail, ucBkg, ucOffset
parameter	
applied dimensions	DCP[appl,0,22]

HB_EXP	
calculate second estimate for the attractiveness without consideration of seasonal effects	
cubes	ucBkg, phi0_h1ExpSm
parameter	float: 0.05,23,53,0,1000
applied dimensions	CW[appl,0,128],DCP[appl,0,22]

DATA_DECOMPOSITION	
calculate seasonal factors used for forecasting First/Business compartment	
cubes	ucBkg, phi0_h1ExpSm, phi0_h1Hist, attrPrepared, seasonPrepared
parameter	bool: 1, float: -0.5,3,0,1000,1
applied dimensions	DCP[appl,0,22],F[aggr,0,7],DOW[aggr,0,6]

DATA_DECOMPOSITION	
calculate seasonal factors used for forecasting Economy compartment	
cubes	ucBkg, phi0_h1ExpSm, phi0_h1Hist, attrPrepared, seasonPrepared
parameter	bool: 1, float: -0.5,3,0,1000,1
applied dimensions	DCP[appl,0,22],F[aggr,8,19],DOW[aggr,0,6]

DATA_DECOMPOSITION	
calculate first estimate seasonal factors First/Business compartment	
cubes	ucBkg, phi0_h1ExpSm, phi0_h1Hist, attrPrepared, seasonSmoothed
parameter	bool: 1, float: -0.5,3,0,1000,1
applied dimensions	DCP[appl,0,22],F[aggr,0,7],DOW[aggr,0,6]

DATA_DECOMPOSITION	
calculate first estimate seasonal factors Economy compartment	
cubes	ucBkg, phi0_h1ExpSm, phi0_h1Hist, attrPrepared, seasonSmoothed
parameter	bool: 1, float: -0.5,3,0,1000,1
applied dimensions	DCP[appl,0,22],F[aggr,8,19],DOW[aggr,0,6]

DATA_SMOOTHING	
smooth the determined seasonal factors	
cubes	seasonSmoothed
parameter	float: 5,0.1
applied dimensions	CW[appl,0,128]

HB_EXP	
learn seasonal behaviour	
cubes	seasonSmoothed, phi0_h1Hist
parameter	float: 0.6,53,1,-1,1000
applied dimensions	CW[appl,0,128]

DATA_DECOMPOSITION	
data decomposition under consideration of historical behaviour attractiveness and season	
cubes	ucBkg, phi0_h1ExpSm, phi0_h1Hist, attr, season
parameter	bool: 0, float: -0.5,3,0,1000,1
applied dimensions	DCP[appl,0,22]

HB_EXP	
learn attractiveness simple exponential smoothing	
cubes	attr, phi0_h1ExpSm
parameter	float: 0.1,23,53,0,1000
applied dimensions	CW[appl,0,128],DCP[appl,0,22]

HB_BROWN	
learn attractiveness brown model	
cubes	attr, phi0_h2Brown, phi1_h2Brown
parameter	float: 0.04,23,53,0,1000,-0.05,0.05
applied dimensions	CW[appl,0,128],DCP[appl,0,22]

HB_REGR	
learn attractiveness linear regression model	
cubes	attr, phi0_h2Regr, phi1_h2Regr
parameter	float: 23,53,0,1000,-0.1,0.1
applied dimensions	CW[appl,0,128],DCP[appl,0,22]

FC_ATTR	
predict attractiveness simple exponential smoothing model	
cubes	attr, phi0_h1ExpSm, phi1_h2Brown, phi0_h1Hist, avail, fc_h1ExpSm, blockElemShift
parameter	bool: 0, 0, 0, float: 0,23
applied dimensions	CW[appl,0,128],DCP[appl,0,22],DCPFC[appl,0,22]

FC_ATTR	
predict attractiveness brown model	
cubes	attr, phi0_h2Brown, phi1_h2Brown, phi0_h1Hist, avail, fc_h2Brown, blockElemShift
parameter	bool: 0, 1, 1, float: 0,23
applied dimensions	CW[appl,0,128],DCP[appl,0,22],DCPFC[appl,0,22]

FC_ATTR	
predict attractiveness linear regression model	
cubes	attr, phi0_h2Regr, phi1_h2Regr, phi0_h1Hist, avail, fc_h2Regr, blockElemShift
parameter	bool: 0, 1, 0, float: 0,23
applied dimensions	CW[appl,0,128],DCP[appl,0,22],DCPFC[appl,0,22]

FC_SEASON	
predict seasonal factors historical model	
cubes	season, seasonPrepared, phi0_h1Hist, attr, fc_h1ExpSm, fc_h1Hist
parameter	bool: 1, float: 1,-1,3,1
applied dimensions	DCP[appl,0,22],DCPFC[appl,0,22]

FC_SEASON	
predict seasonal factors additive adaptation	
cubes	season, seasonPrepared, phi0_h1Hist, attr, fc_h1ExpSm, fc_h2Add
parameter	bool: 1, float: 2,-0.5,2,1
applied dimensions	DCP[appl,0,22],DCPFC[appl,0,22]

FC_SEASON	
predict seasonal factors multiplicative adaptation	
cubes	season, seasonPrepared, phi0_h1Hist, attr, fc_h1ExpSm, fc_h3Mult
parameter	bool: 1, float: 3,-0.5,2,1
applied dimensions	DCP[appl,0,22],DCPFC[appl,0,22]

FC_LSB	
predict total demand with the model used in the current system	
cubes	ucBkg, phi0_h1ExpSm, phi1_h2Brown, phi0_h1Hist, avail, fc_input, blockElemShift
parameter	bool: 1, 0, 0, float: 0,23
applied dimensions	CW[appl,0,128],DCP[appl,0,22],DCPFC[appl,0,22], FCNR[appl,0,0]

COMBINING_ADD_PARTS	
combine components of result forecast 1	
cubes	fc_h1ExpSm, fc_h3Mult, fc_input
parameter	
applied dimensions	FCNR[appl,1,1]

COMBINING_ADD_PARTS	
combine components of result forecast 2	
cubes	fc_h2Brown, fc_h3Mult, fc_input
parameter	
applied dimensions	FCNR[appl,2,2]

COMBINING_ADD_PARTS	
combine components of result forecast 3	
cubes	fc_h1ExpSm, fc_h2Add, fc_input
parameter	
applied dimensions	FCNR[appl,3,3]

COMBINING_ADD_PARTS	
combine components of result forecast 4	
cubes	fc_h2Regr, fc_h3Mult, fc_input
parameter	
applied dimensions	FCNR[appl,4,4]

COMBINING_ADD_PARTS	
combine components of result forecast 5	
cubes	fc_h2Regr, fc_h1Hist, fc_input
parameter	
applied dimensions	FCNR[appl,5,5]

COMBINING_ADD_PARTS	
combine components of result forecast 6	
cubes	fc_h2Regr, fc_h2Add, fc_input
parameter	
applied dimensions	FCNR[appl,6,6]

VALID_FC_REF	
calculate toal forecast errors	
cubes	ucBkg, fc_input, err_input_bias
parameter	bool: 0
applied dimensions	DCP[appl,22,22],DCPFC[appl,0,22]

ERROR_COVAR	
calculate mean absolute deviation low level	
cubes	err_input_bias, mad_low
parameter	bool: 1, float: 1
applied dimensions	CW[appl,93,128]

ERROR_COVAR	
calculate mean absolute deviation high level	
cubes	err_input_bias, mad_high
parameter	bool: 1, float: 60
applied dimensions	CW[appl,93,128],F[appl,0,19],POS[appl,0,2]

HB_LINEAR_COMBINATION	
determine linear combination weights model F^{av}	
cubes	err_input_bias, ucBkg, lin_comb_weight, lin_comb_offset
parameter	float: 0,-1,-1
applied dimensions	FCNR[appl,0,6],CW[appl,53,92],COMB[appl,0,0], DCP[appl,22,22]

HB_LINEAR_COMBINATION	
determine linear combination weights model F^{outp}	
cubes	err_input_bias, ucBkg, lin_comb_weight, lin_comb_offset
parameter	float: 1,-1,-1
applied dimensions	FCNR[appl,0,6],CW[appl,53,92],COMB[appl,1,1], DCP[appl,22,22]

HB_LINEAR_COMBINATION	
determine linear combination weights model F^{var}	
cubes	err_input_bias, ucBkg, lin_comb_weight, lin_comb_offset
parameter	float: 2,-1,-1
applied dimensions	FCNR[appl,0,6],CW[appl,53,92],COMB[appl,2,2], DCP[appl,22,22]

HB_LINEAR_COMBINATION	
determine linear combination weights model F^{opt}	
cubes	err_input_bias, ucBkg, lin_comb_weight, lin_comb_offset
parameter	float: 3,-1,-1
applied dimensions	FCNR[appl,0,6],CW[appl,53,92],COMB[appl,3,3], DCP[appl,22,22]

HB_LINEAR_COMBINATION	
determine linear combination weights model F^{ols}	
cubes	err_input_bias, ucBkg, lin_comb_weight, lin_comb_offset
parameter	float: 4,-1,-1
applied dimensions	FCNR[appl,0,6],CW[appl,53,92],COMB[appl,4,4], DCP[appl,22,22]

LINEAR_COMBINATION	
combine forecasts	
cubes	fc_input, fc_combined, lin_comb_weight, lin_comb_offset
parameter	
applied dimensions	FCNR[appl,0,6],CW[appl,0,128]

VALID_FC_REF	
calculate forecast errors	
cubes	ucBkg, fc_combined, err_combined_bias
parameter	bool: 0
applied dimensions	DCP[appl,22,22],DCPFC[appl,0,22]

ERROR_COVAR	
calculate mean absolute deviation low level	
cubes	err_combined_bias, mad_combined_low
parameter	bool: 1, float: 1
applied dimensions	CW[appl,93,128]

ERROR_COVAR	
calculate mean absolute deviation high level	
cubes	err_combined_bias, mad_combined_high
parameter	bool: 1, float: 60
applied dimensions	CW[appl,93,128],F[appl,0,19],POS[appl,0,2]

FILE_INTERFACE	
save results combination weights	
cubes	
parameter	string: UNDEFINED, lin_comb_weight, lin_comb_offset, DCPFC
applied dimensions	

FILE_INTERFACE	
save results combined forecasts	
cubes	
parameter	string: UNDEFINED, mad_combined_low,mad_combined_high,mad_low, mad_high,DCPFC
applied dimensions	

B.6.4 Experiment4 : Combination of Predictions for the Seasonal Demand Component

Brief

name of the experiment	experiment4
short description	combination of diversified seasonal forecasts by combination models F^{av}, F^{outp}, F^{var}, F^{opt} and F^{ols}

In this experiment the predictions of the seasonal component are diversified, the attractiveness component is predicted with a simple exponential smoothing model with additive adaptation to the current booking values. The function space has been diversified with the models $h_1^{season}(x, \phi)$ and $h_3^{season}(x, \phi)$. Diversified parameters applied for the calculation of seasonal factors: ϕ_{low} and ϕ_{high} (lower and upper limit of expected seasonal behaviour). In order to generate sets of range limits which are not completely unbalanced the initial parameters chosen for $\phi_{low} = -0.5$, and $\phi_{high} = 3$ have been dumped with different factors between 0 and 1. The generated predictions for the seasonal factors are combined by different linear combination models F^{av}, F^{outp}, F^{var}, F^{opt} and F^{ols} (see Section 3.2). The experiment provides the error variance information at the low level of forecasting and aggregated over Fareclasses and Point of Sales.

Inputs and Results

input	description
bkg	booking values
avail	availability information
blockElemShift	number of weeks contained in a data collection point τ

result	description
mad_combined_low	error variance of the combined forecast at the low level
mad_combined_high	error variance of the combined forecast at the high level
mad_compare_low	error variance of the compare forecast at the low level
mad_compare_high	error variance of the compare forecast at the high level
lin_comb_weight	linear combination weights
lin_comb_offset	offset linear combination

Summary of the Calculation

The calculation can be summarised in the following steps:

1. load the data
2. carry out unconstraining
3. decompose the input data
4. learn the attractiveness
5. learn the historical seasonal behaviour over history weeks 0 to 52
6. learn the historical attractiveness over history weeks 0 to 52

7. generate the prediction for the attractiveness (all weeks)

8. generate the diversified predictions for the seasonal behaviour (2 types of diversification, all weeks)

9. determine the seasonal forecast performance (all weeks)

10. learn the combination weights based on weeks 53 to 92

11. combine the seasonal forecasts

12. calculate the total demand forecasts (all weeks)

13. determine the combined forecast performance for weeks 93 to 128

14. save the results of the combined forecasts

15. calculate the forecast of the current system (compare forecasts)

16. determine the compare forecast performance for weeks 93 to 128

17. save the results of the compare forecasts

Detailed Description of Applied Components

FILE_INTERFACE	
load booking and availability information	
cubes	
parameter	string: bkg,avail, UNDEFINED, DCP
applied dimensions	

FILE_INTERFACE	
load block element shift	
cubes	
parameter	string: blockElemShift, UNDEFINED, UNDEFINED
applied dimensions	

HB_EXP	
calculate first estimate for the attractiveness without consideration of unconstraining and seasonal effects	
cubes	bkg, phi0_h1ExpSm
parameter	float: 0.05,23,53,0,1000
applied dimensions	CW[appl,0,128],DCP[appl,0,22]

UNCONSTRAINING	
unconstrain the booking data	
cubes	bkg, phi0_h1ExpSm, avail, ucBkg, ucOffset
parameter	
applied dimensions	DCP[appl,0,22]

HB_EXP	
calculate second estimate for the attractiveness without consideration of seasonal effects	
cubes	ucBkg, phi0_h1ExpSm
parameter	float: 0.05,23,53,0,1000
applied dimensions	CW[appl,0,128],DCP[appl,0,22]

DATA_DECOMPOSITION	
calculate first estimate of seasonal factors First/Business compartment	
cubes	ucBkg, phi0_h1ExpSm, phi0_h1Hist, attrPrepared, seasonSmoothed
parameter	bool: 1, float: -0.5,3,0,1000,1
applied dimensions	DCP[appl,0,22],F[aggr,0,7],DOW[aggr,0,6]

DATA_DECOMPOSITION	
calculate first estimate of seasonal factors Economy compartment	
cubes	ucBkg, phi0_h1ExpSm, phi0_h1Hist, attrPrepared, seasonSmoothed
parameter	bool: 1, float: -0.5,3,0,1000,1
applied dimensions	DCP[appl,0,22],F[aggr,8,19],DOW[aggr,0,6]

DATA_SMOOTHING	
smooth the determined seasonal factors	
cubes	seasonSmoothed
parameter	float: 5,0.1
applied dimensions	CW[appl,0,128]

HB_EXP	
learn seasonal behaviour (first estimate)	
cubes	seasonSmoothed, phi0_h1Hist
parameter	float: 0.6,53,1,-1,1000
applied dimensions	CW[appl,0,128]

DATA_DECOMPOSITION	
data decomposition under consideration of historical behaviour of attractiveness and season	
cubes	ucBkg, phi0_h1ExpSm, phi0_h1Hist, attr, season
parameter	bool: 0, float: -1,3,0,1000,1
applied dimensions	DCP[appl,0,22]

HB_EXP	
learn seasonal behaviour (improved estimate)	
cubes	attr, phi0_h1ExpSm
parameter	float: 0.1,23,53,0,1000
applied dimensions	CW[appl,0,128],DCP[appl,0,22]

DATA_DECOMPOSITION	
final data decomposition real data (low level)	
cubes	ucBkg, phi0_h1ExpSm, phi0_h1Hist, attr, season
parameter	bool: 1, float: -1,3,0,1000,1
applied dimensions	DCP[appl,0,22]

DATA_DECOMPOSITION	
data decomposition used for forecasting (diversified level First/Business)	
cubes	ucBkg, phi0_h1ExpSm, phi0_h1Hist, attrPrepared, seasonPrepared
parameter	bool: 1, float: -1,3,0,1000,1
applied dimensions	DCP[appl,0,22]

FC_LSB	
calculation of the total compare forecast	
cubes	ucBkg, phi0_h1ExpSm, phi1_h1ExpSm, phi0_h1Hist, avail, fc_compare, blockElemShift
parameter	bool: 1, 0, 0, float: 0,23
applied dimensions	CW[appl,0,128],DCP[appl,0,22],DCPFC[appl,0,22], FCNR[appl,0,0]

VALID_FC_REF	
calculation of compare forecast error	
cubes	ucBkg, fc_compare, err_compare_bias
parameter	bool: 0
applied dimensions	DCP[appl,22,22],DCPFC[appl,0,22]

ERROR_COVAR	
calculation mean absolute deviation low level compare forecast	
cubes	err_compare_bias, mad_compare_low
parameter	bool: 1, float: 1
applied dimensions	CW[appl,93,128]

ERROR_COVAR	
calculation mean absolute deviation high level compare forecast	
cubes	err_compare_bias, mad_compare_high
parameter	bool: 1, float: 60
applied dimensions	CW[appl,93,128],F[appl,0,19],POS[appl,0,2]

FC_ATTR	
forecast of the attractiveness component	
cubes	attr, phi0_h1ExpSm, phi1_h1ExpSm, phi0_h1Hist, avail, fc_h1ExpSm, blockElemShift
parameter	bool: 0, 0, 0, float: 0,23
applied dimensions	CW[appl,0,128],DCP[appl,0,22],DCPFC[appl,0,22]

FC_SEASON	
forecast diversified seasonal factors	
cubes	season, seasonPrepared, phi0_h1Hist, attr, fc_h1ExpSm, fc_h1Hist
parameter	bool: 1, float: 1,-1,3,1
applied dimensions	DCP[appl,0,22],DCPFC[appl,0,22]

VALID_FC_REF	
calculate forecast errors diversified seasonal factors	
cubes	season, fc_h1Hist, err_input_bias
parameter	bool: 0
applied dimensions	DCP[appl,22,22],DCPFC[appl,0,22]

HB_LINEAR_COMBINATION	
learn linear combination weights F^{av}	
cubes	err_input_bias, season, lin_comb_weight, lin_comb_offset
parameter	float: 0,-1,-1
applied dimensions	DIV1[appl,0,3],DIV2[appl,0,1], CW[appl,53,92],COMB[appl,0,0],DCP[appl,22,22]

HB_LINEAR_COMBINATION	
learn linear combination weights F^{outp}	
cubes	err_input_bias, season, lin_comb_weight, lin_comb_offset
parameter	float: 1,-1,-1
applied dimensions	DIV1[appl,0,3],DIV2[appl,0,1], CW[appl,53,92],COMB[appl,1,1],DCP[appl,22,22]

HB_LINEAR_COMBINATION	
learn linear combination weights F^{var}	
cubes	err_input_bias, season, lin_comb_weight, lin_comb_offset
parameter	float: 2,-1,-1
applied dimensions	DIV1[appl,0,3],DIV2[appl,0,1], CW[appl,53,92],COMB[appl,2,2],DCP[appl,22,22]

HB_LINEAR_COMBINATION	
learn linear combination weights F^{opt}	
cubes	err_input_bias, season, lin_comb_weight, lin_comb_offset
parameter	float: 3,-1,-1
applied dimensions	DIV1[appl,0,3],DIV2[appl,0,1], CW[appl,53,92],COMB[appl,3,3],DCP[appl,22,22]

HB_LINEAR_COMBINATION	
learn linear combination weights F^{ols}	
cubes	err_input_bias, season, lin_comb_weight, lin_comb_offset
parameter	float: 4,-1,-1
applied dimensions	DIV1[appl,0,3],DIV2[appl,0,1], CW[appl,53,92],COMB[appl,4,4],DCP[appl,22,22]

LINEAR_COMBINATION	
combine forecasts	
cubes	fc_h1Hist, fc_comb, lin_comb_weight, lin_comb_offset
parameter	
applied dimensions	DIV1[appl,0,3],DIV2[appl,0,1],CW[appl,0,128]

COMBINING_ADD_PARTS	
generate total forecast	
cubes	fc_h1ExpSm, fc_comb, fc_combined
parameter	
applied dimensions	

VALID_FC_REF	
calculate error combined total forecast	
cubes	ucBkg, fc_combined, err_combined_bias
parameter	bool: 0
applied dimensions	DCP[appl,22,22],DCPFC[appl,0,22]

ERROR_COVAR	
calculate mean absolute deviation low level	
cubes	err_combined_bias, mad_combined_low
parameter	bool: 1, float: 1
applied dimensions	CW[appl,93,128]

ERROR_COVAR	
calculate mean absolute deviation high level	
cubes	err_combined_bias, mad_combined_high
parameter	bool: 1, float: 60
applied dimensions	CW[appl,93,128],F[appl,0,19],POS[appl,0,2]

FILE_INTERFACE	
save errors	
cubes	
parameter	string: UNDEFINED, mad_combined_low,mad_combined_high,mad_compare_low, mad_compare_high, DCPFC
applied dimensions	

FILE_INTERFACE	
save learned combination weights	
cubes	
parameter	string: UNDEFINED, lin_comb_weight,lin_comb_offset, DCPFC
applied dimensions	

B.6.5 Experiment5 : Multi Level Combination of Predictions for the Seasonal Demand Component

Brief

name of the experiment	experiment5
short description	combination of multi level seasonal forecasts by combination models F^{av}, F^{outp}, F^{var}, F^{opt} and F^{ols}

In this experiment, in addition to the previous experiment, the level of calculation of seasonal factors is diversified in history building as well as in forecasting.

The set of input forecasts can be seen in Table 13.

Inputs and Results

input	description
bkg	booking values
avail	availability information
blockElemShift	number of weeks contained in a data collection point τ

result	description
mad_combined_low	error variance of the combined forecast at the low level
mad_combined_high	error variance of the combined forecast at the high level

Summary of the Calculation

The calculation can be summarised in the following steps:

1. load the data
2. carry out unconstraining
3. decompose the input data
4. learn the attractiveness
5. learn the historical seasonal behaviour over history weeks 0 to 52
6. learn the historical attractiveness over history weeks 0 to 52
7. generate the prediction for the attractiveness (all weeks)
8. generate the diversified predictions for the seasonal behaviour (multi level diversification, all weeks)
9. determine the seasonal forecast performance (all weeks)

10. learn the combination weights based on weeks 53 to 92

11. combine the seasonal forecasts

12. calculate the total demand forecasts (all weeks)

13. determine the combined forecast performance for weeks 93 to 128

14. save the results of the combined forecasts

15. calculate the forecast of the current system (compare forecasts)

16. determine the compare forecast performance for weeks 93 to 128

17. save the results of the compare forecasts

Detailed Description of Applied Components

FILE_INTERFACE	
load booking and availability information	
cubes	
parameter	string: bkg,avail, UNDEFINED, DCP
applied dimensions	

FILE_INTERFACE	
load block element shift	
cubes	
parameter	string: blockElemShift, UNDEFINED, UNDEFINED
applied dimensions	

HB_EXP	
calculate first estimate for the attractiveness without consideration of unconstraining and seasonal effects	
cubes	bkg, phi0_h1ExpSm
parameter	float: 0.05,23,53,0,1000
applied dimensions	CW[appl,0,128],DCP[appl,0,22]

UNCONSTRAINING	
unconstrain the booking data	
cubes	bkg, phi0_h1ExpSm, avail, ucBkg, ucOffset
parameter	
applied dimensions	DCP[appl,0,22]

HB_EXP	
calculate second estimate for the attractiveness without consideration of seasonal effects	
cubes	ucBkg, phi0_h1ExpSm
parameter	float: 0.05,23,53,0,1000
applied dimensions	CW[appl,0,128],DCP[appl,0,22]

DATA_DECOMPOSITION	
calculate first estimate of seasonal factors First/Business compartment	
cubes	ucBkg, phi0_h1ExpSm, phi0_h1Hist, attrPrepared, seasonSmoothed
parameter	bool: 1, float: -0.5,3,0,1000,1
applied dimensions	DCP[appl,0,22],F[aggr,0,7],DOW[aggr,0,6]

DATA_DECOMPOSITION	
calculate first estimate of seasonal factors Economy compartment	
cubes	ucBkg, phi0_h1ExpSm, phi0_h1Hist, attrPrepared, seasonSmoothed
parameter	bool: 1, float: -0.5,3,0,1000,1
applied dimensions	DCP[appl,0,22],F[aggr,8,19],DOW[aggr,0,6]

DATA_SMOOTHING	
smooth the determined seasonal factors	
cubes	seasonSmoothed
parameter	float: 5,0.1
applied dimensions	CW[appl,0,128]

HB_EXP	
learn seasonal behaviour (first estimate)	
cubes	seasonSmoothed, phi0_h1Hist
parameter	float: 0.6,53,1,-1,1000
applied dimensions	CW[appl,0,128]

DATA_DECOMPOSITION	
data decomposition under consideration of historical behaviour of attractiveness and season	
cubes	ucBkg, phi0_h1ExpSm, phi0_h1Hist, attr, season
parameter	bool: 1, float: -1,3,0,1000,1
applied dimensions	DCP[appl,0,22]

HB_EXP	
learn seasonal behaviour (improved estimate)	
cubes	attr, phi0_h1ExpSm
parameter	float: 0.1,23,53,0,1000
applied dimensions	CW[appl,0,128],DCP[appl,0,22]

DATA_DECOMPOSITION	
final data decomposition real data (low level)	
cubes	ucBkg, phi0_h1ExpSm, phi0_h1Hist, attr, season
parameter	bool: 1, float: -1,3,0,1000,1
applied dimensions	DCP[appl,0,22]

DATA_DECOMPOSITION	
data decomposition used for learning (diversified level First/Business)	
cubes	ucBkg, phi0_h1ExpSm, phi0_h1Hist, attrPrepared, seasonSmoothed2
parameter	bool: 1, float: -1,3,0,1000,1
applied dimensions	DCP[appl,0,22], F[diversified,0,7], DOW[diversified,0,6]

DATA_DECOMPOSITION	
data decomposition used for learning (diversified level Economy)	
cubes	ucBkg, phi0_h1ExpSm, phi0_h1Hist, attrPrepared, seasonSmoothed2
parameter	bool: 1, float: -1,3,0,1000,1
applied dimensions	DCP[appl,0,22], F[diversified,0,7], DOW[diversified,0,6]

DATA_SMOOTHING	
smoothing of the diversified decomposed data	
cubes	seasonSmoothed2
parameter	float: 2,0.2
applied dimensions	CW[appl,0,128]

HB_EXP	
learning of the diversified history	
cubes	seasonSmoothed2, phi0_h2Hist
parameter	float: 0.6,53,1,-1,1000
applied dimensions	CW[appl,0,128]

DATA_DECOMPOSITION	
data decomposition used for forecasting (diversified level First/Business)	
cubes	ucBkg, phi0_h1ExpSm, phi0_h1Hist, attrPrepared, seasonPrepared
parameter	bool: 1, float: -1,3,0,1000,1
applied dimensions	DCP[appl,0,22],F[diversified,0,7], DOW[diversified,0,6]

DATA_DECOMPOSITION	
data decomposition used for forecasting (diversified level Economy)	
cubes	ucBkg, phi0_h1ExpSm, phi0_h1Hist, attrPrepared, seasonPrepared
parameter	bool: 1, float: -1,3,0,1000,1
applied dimensions	DCP[appl,0,22],F[diversified,8,19], DOW[diversified,0,6]

FC_ATTR	
forecast of the attractiveness component	
cubes	attr, phi0_h1ExpSm, phi1_h1ExpSm, fc_h1ExpSm, blockElemShift
parameter	bool: 0, 0, 1, float: 23
applied dimensions	CW[appl,0,128],DCP[appl,0,22],DCPFC[appl,0,22], DIV1[appl,0,0], DIV2[appl,0,0], DIV3[appl,0,0], DIV4[appl,0,0]

FC_SEASON	
forecast diversified seasonal factors	
cubes	season, seasonPrepared, phi0_h2Hist, phi0_h1ExpSm, fc_h1Hist
parameter	bool: 1, float: 1,-1,3,1
applied dimensions	DCP[appl,0,22],DCPFC[appl,0,22]

VALID_FC_REF	
calculate forecast errors diversified seasonal factors	
cubes	season, fc_h1Hist, err_h1Hist
parameter	bool: 0
applied dimensions	DCP[appl,22,22],DCPFC[appl,0,22]

HB_LINEAR_COMBINATION	
learn linear combination weights	
cubes	err_h1Hist, season, lin_comb_weight, lin_comb_offset
parameter	float: 2,5,-1
applied dimensions	DIV1[appl,0,3],DIV2[appl,0,1],DIV3[appl,0,1], DIV4[appl,0,1],CW[appl,53,92],DOW[appl,0,6], DCP[appl,22,22]

LINEAR_COMBINATION	
combine forecasts	
cubes	fc_h1Hist, fc_comb, lin_comb_weight, lin_comb_offset
parameter	
applied dimensions	DIV1[appl,0,3],DIV2[appl,0,1],DIV3[appl,0,1], DIV4[appl,0,1],CW[appl,0,128]

COMBINING_ADD_PARTS	
generate total forecast	
cubes	fc_h1ExpSm, fc_comb, fc_combined
parameter	
applied dimensions	

VALID_FC_REF	
calculate error combined total forecast	
cubes	ucBkg, fc_combined, err_combined_bias
parameter	bool: 0
applied dimensions	DCP[appl,22,22],DCPFC[appl,0,22]

ERROR_COVAR	
calculate mean absolute deviation low level	
cubes	err_combined_bias, mad_combined_low
parameter	bool: 1, float: 1
applied dimensions	CW[appl,93,128]

ERROR_COVAR	
calculate mean absolute deviation high level	
cubes	err_combined_bias, mad_combined_high
parameter	bool: 1, float: 60
applied dimensions	CW[appl,93,128],F[appl,0,19],POS[appl,0,2]

FILE_INTERFACE	
save errors combined forecast	
cubes	
parameter	string: UNDEFINED, mad_combined_low,mad_combined_high, DCPFC
applied dimensions	

Variations of the Experiment

The experiment can be varied by using different trimming strategies. This can be reached by modification of the trimming parameters of component HB_LIN_COMBINATION.

B.6.6 Experiment6 : Comparison of Different Pooling Approaches

Brief

name of the experiment	experiment6
short description	combination of multi level seasonal forecasts by different predefined linear combination structures

In this experiment 4 diversifications are used:

- diversification of the function space ($h_1^{season}(x,\phi)$ and $h_3^{season}(x,\phi)$)
- diversification of parameters ϕ_{low}, and ϕ_{high}
- diversification of the level Fareclass aggregated to Compartment

- diversification of the level Day of Week (calculation per day of week or over all day of weeks)

The six combination structures MLP1 to MLP6 are described in Table 18. They are all based on the dimensions of the forecast generation space. The only difference between the structures is in the order of dimensions used in order to determine the pools of the next combination step.

Inputs and Results

input	description
bkg	booking values
avail	availability information
blockElemShift	number of weeks contained in a data collection point τ
result	**description**
varCombinedLow	error variance of the combined forecast at the low level
varCombinedHigh	error variance of the combined forecast at the high level

Summary of the Calculation

The calculation can be summarised in the following steps:

1. load the data
2. carry out unconstraining
3. decompose the input data
4. learn the attractiveness
5. learn the historical seasonal behaviour over history weeks 0 to 52

6. learn the historical attractiveness over history weeks 0 to 52

7. generate the prediction for the attractiveness (all weeks)

8. generate the diversified predictions for the seasonal behaviour (multi level diversification, all weeks)

9. determine the seasonal forecast performance (all weeks)

10. for six different predefined combination structures

    - determine combination weights based on weeks 53 to 92 with

    - combine the seasonal forecasts

    - calculate the total demand forecasts (all weeks)

    - determine the combined forecast performance for weeks 93 to 128

11. save the results of the combined forecasts

12. calculate the forecast of the current system (compare forecasts)

13. determine the compare forecast performance for weeks 93 to 128

14. save the results of the compare forecasts

Detailed Description of Applied Components

FILE_INTERFACE	
load booking and availability information	
cubes	
parameter	string: bkg,avail, UNDEFINED, DCP
applied dimensions	

FILE_INTERFACE	
load block element shift	
cubes	
parameter	string: blockElemShift, UNDEFINED, UNDEFINED
applied dimensions	

HB_EXP	
calculate first estimate for the attractiveness without consideration of unconstraining and seasonal effects	
cubes	bkg, phi0_h1ExpSm
parameter	float: 0.05,23,53,0,1000
applied dimensions	CW[appl,0,128],DCP[appl,0,22]

UNCONSTRAINING	
unconstrain the booking data	
cubes	bkg, phi0_h1ExpSm, avail, ucBkg, ucOffset
parameter	
applied dimensions	DCP[appl,0,22]

HB_EXP	
calculate second estimate for the attractiveness without consideration of seasonal effects	
cubes	ucBkg, phi0_h1ExpSm
parameter	float: 0.05,23,53,0,1000
applied dimensions	CW[appl,0,128],DCP[appl,0,22]

DATA_DECOMPOSITION	
calculate first estimate of seasonal factors First/Business compartment	
cubes	ucBkg, phi0_h1ExpSm, phi0_h1Hist, attrPrepared, seasonSmoothed
parameter	bool: 1, float: -0.5,3,0,1000,1
applied dimensions	DCP[appl,0,22],F[aggr,0,7],DOW[aggr,0,6]

DATA_DECOMPOSITION	
calculate first estimate of seasonal factors Economy compartment	
cubes	ucBkg, phi0_h1ExpSm, phi0_h1Hist, attrPrepared, seasonSmoothed
parameter	bool: 1, float: -0.5,3,0,1000,1
applied dimensions	DCP[appl,0,22],F[aggr,8,19],DOW[aggr,0,6]

DATA_SMOOTHING	
smooth the determined seasonal factors	
cubes	seasonSmoothed
parameter	float: 5,0.1
applied dimensions	CW[appl,0,128]

HB_EXP	
learn seasonal behaviour (first estimate)	
cubes	seasonSmoothed, phi0_h1Hist
parameter	float: 0.6,53,1,-1,1000
applied dimensions	CW[appl,0,128]

DATA_DECOMPOSITION	
data decomposition under consideration of historical behaviour of attractiveness and season	
cubes	ucBkg, phi0_h1ExpSm, phi0_h1Hist, attr, season
parameter	bool: 0, float: -1,3,0,1000,1
applied dimensions	DCP[appl,0,22]

HB_EXP	
learn seasonal behaviour (improved estimate)	
cubes	attr, phi0_h1ExpSm
parameter	float: 0.1,23,53,0,1000
applied dimensions	CW[appl,0,128],DCP[appl,0,22]

DATA_DECOMPOSITION	
final data decomposition real data (low level)	
cubes	ucBkg, phi0_h1ExpSm, phi0_h1Hist, attr, season
parameter	bool: 1, float: -1,3,0,1000,1
applied dimensions	DCP[appl,0,22]

DATA_DECOMPOSITION	
data decomposition used for learning (diversified level First/Business)	
cubes	ucBkg, phi0_h1ExpSm, phi0_h1Hist, attrPrepared, seasonSmoothed2
parameter	bool: 1, float: -1,3,0,1000,1
applied dimensions	DCP[appl,0,22], F[diversified,8,19], DOW[diversified,0,6]

DATA_DECOMPOSITION	
data decomposition used for learning (diversified level Economy)	
cubes	ucBkg, phi0_h1ExpSm, phi0_h1Hist, attrPrepared, seasonSmoothed2
parameter	bool: 1, float: -1,3,0,1000,1
applied dimensions	DCP[appl,0,22], F[diversified,0,7], DOW[diversified,0,6]

DATA_SMOOTHING	
smoothing of the diversified decomposed data	
cubes	seasonSmoothed2
parameter	float: 2,0.2
applied dimensions	CW[appl,0,128]

HB_EXP	
learning of the diversified history	
cubes	seasonSmoothed2, phi0_h2Hist
parameter	float: 0.6,53,1,-1,1000
applied dimensions	CW[appl,0,128]

DATA_DECOMPOSITION	
data decomposition used for forecasting (diversified level First/Business)	
cubes	ucBkg, phi0_h1ExpSm, phi0_h1Hist, attrPrepared, seasonPrepared
parameter	bool: 1, float: -1,3,0,1000,1
applied dimensions	DCP[appl,0,22],F[diversified,0,7], DOW[diversified,0,6]

DATA_DECOMPOSITION	
data decomposition used for forecasting (diversified level Economy)	
cubes	ucBkg, phi0_h1ExpSm, phi0_h1Hist, attrPrepared, seasonPrepared
parameter	bool: 1, float: -1,3,0,1000,1
applied dimensions	DCP[appl,0,22],F[diversified,8,19], DOW[diversified,0,6]

FC_ATTR	
forecast of the attractiveness component	
cubes	attr, phi0_h1ExpSm, phi1_h1ExpSm, fc_h1ExpSm, blockElemShift
parameter	bool: 0, 0, 1, float: 23
applied dimensions	CW[appl,0,128],DCP[appl,0,22],DCPFC[appl,0,22], DIV1[appl,0,0],DIV2[appl,0,0],DIV3[appl,0,0], DIV4[appl,0,0]

FC_SEASON	
forecast diversified seasonal factors	
cubes	season, seasonPrepared, phi0_h2Hist, attr, fc_h1ExpSm, fc_h1Hist
parameter	bool: 1, float: 1,-1,3,1
applied dimensions	DCP[appl,0,22],DCPFC[appl,0,22]

VALID_FC_REF	
calculate forecast errors diversified seasonal factors	
cubes	season, fc_h1Hist, err_h1Hist
parameter	bool: 0
applied dimensions	DCP[appl,22,22],DCPFC[appl,0,22]

HB_LINEAR_COMBINATION_STRUCTURE	
learn combination weights structure MLP1	
cubes	err_h1Hist, season, lin_comb_weight, lin_comb_offset
parameter	float: 0,0,0,0,4,2,10,3, string: DIV1,DIV2,DIV3,DIV4, 0123
applied dimensions	DIV1[appl,0,3],DIV2[appl,0,1],DIV3[appl,0,1], DIV4[appl,0,1],CW[appl,53,92],DOW[appl,0,6], DCP[appl,22,22]

HB_LINEAR_COMBINATION_STRUCTURE	
learn combination weights structure MLP2	
cubes	err_h1Hist, season, lin_comb_weight, lin_comb_offset
parameter	float: 0,0,0,0,4,2,10,3, string: DIV1,DIV2,DIV3,DIV4, 1023
applied dimensions	DIV1[appl,0,3],DIV2[appl,0,1],DIV3[appl,0,1], DIV4[appl,0,1],CW[appl,53,92],DOW[appl,0,6]

HB_LINEAR_COMBINATION_STRUCTURE	
learn combination weights structure MLP3	
cubes	err_h1Hist, season, lin_comb_weight, lin_comb_offset
parameter	float: 0,0,0,0,4,2,10,3, string: DIV1,DIV2,DIV3,DIV4, 2301
applied dimensions	DIV1[appl,0,3],DIV2[appl,0,1],DIV3[appl,0,1], DIV4[appl,0,1],CW[appl,53,92],DOW[appl,0,6]

HB_LINEAR_COMBINATION_STRUCTURE	
learn combination weights structure MLP4	
cubes	err_h1Hist, season, lin_comb_weight, lin_comb_offset
parameter	float: 0,0,0,0,4,2,10,3, string: DIV1,DIV2,DIV3,DIV4, 2310
applied dimensions	DIV1[appl,0,3],DIV2[appl,0,1],DIV3[appl,0,1], DIV4[appl,0,1],CW[appl,53,92],DOW[appl,0,6]

HB_LINEAR_COMBINATION_STRUCTURE	
learn combination weights structure MLP5	
cubes	err_h1Hist, season, lin_comb_weight, lin_comb_offset
parameter	float: 0,0,0,0,4,2,10,3, string: DIV1,DIV2,DIV3,DIV4, 0231
applied dimensions	DIV1[appl,0,3],DIV2[appl,0,1],DIV3[appl,0,1], DIV4[appl,0,1],CW[appl,53,92],DOW[appl,0,6]

HB_LINEAR_COMBINATION_STRUCTURE	
learn combination weights structure MLP6	
cubes	err_h1Hist, season, lin_comb_weight, lin_comb_offset
parameter	float: 0,0,0,0,4,2,10,3, string: DIV1,DIV2,DIV3,DIV4, 1230
applied dimensions	DIV1[appl,0,3],DIV2[appl,0,1],DIV3[appl,0,1], DIV4[appl,0,1],CW[appl,53,92],DOW[appl,0,6]

LINEAR_COMBINATION	
combine forecasts	
cubes	fc_h1Hist, fc_comb, lin_comb_weight, lin_comb_offset
parameter	
applied dimensions	DIV1[appl,0,3],DIV2[appl,0,1],DIV3[appl,0,1], DIV4[appl,0,1],CW[appl,0,128]

COMBINING_ADD_PARTS	
calculate total forecast	
cubes	fc_h1ExpSm, fc_comb, fc_combined
parameter	
applied dimensions	

VALID_FC_REF	
calculate combined forecast error	
cubes	ucBkg, fc_combined, err_combined_bias
parameter	bool: 0
applied dimensions	DCP[appl,22,22],DCPFC[appl,0,22]

ERROR_COVAR	
calculate mean absolute deviation low level	
cubes	err_combined_bias, mad_combined_low
parameter	bool: 1, float: 1
applied dimensions	CW[appl,93,128]

ERROR_COVAR	
calculate mean absolute deviation high level	
cubes	err_combined_bias, mad_combined_high
parameter	bool: 1, float: 60
applied dimensions	CW[appl,93,128],F[appl,0,19],POS[appl,0,2]

FILE_INTERFACE	
save results combined forecast errors	
cubes	
parameter	string: UNDEFINED, mad_combined_low,mad_combined_high, DCPFC
applied dimensions	

Variations of the Experiment

Alternative structures can be generated by modification of the last parameter of component

HB_LIN_COMBINATION_STRUCTURE.

B.6.7 Experiment7 : Comparison of Different Pooling Approaches

Brief

name of the experiment	experiment7
short description	generation and evolution of linear combination structures

This experiment uses the same diversified input forecasts for the seasonal component as in the previous experiment. Only one combination is carried out. The used combination structure is generated by component

HB_LIN_COMBINATION_STRUCTURE. This component enables the generation of dynamic combination structures, for instance using the approach of Aiolfi and Timmermann (see 6.2.1) as well as different evolutionary approaches as described in Chapter 7.

Inputs and Results

input	description
bkg	booking values
avail	availability information
blockElemShift	number of weeks contained in a data collection point τ

result	description
mad_combined_low	error variance of the combined forecast at the low level
mad_combined_high	error variance of the combined forecast at the high level
elements	file containing elements of the resulting combination structures
performance_graph	file containing fitness information

Summary of the Calculation

The calculation can be summarised in the following steps:

1. load the data
2. carry out unconstraining
3. decompose the input data
4. learn the attractiveness
5. learn the historical seasonal behaviour over history weeks 0 to 52
6. learn the historical attractiveness over history weeks 0 to 52
7. generate the prediction for the attractiveness (all weeks)
8. generate the diversified predictions for the seasonal behaviour (multi level diversification, all weeks)
9. determine the seasonal forecast performance (all weeks)
10. generate/evolve combination structures (weeks 53 to 92)
11. combine the seasonal forecasts
12. calculate the total demand forecasts (all weeks)

13. determine the combined forecast performance for weeks 93 to 128

14. save the results of the combined forecasts

15. calculate the forecast of the current system (compare forecasts)

16. determine the compare forecast performance for weeks 93 to 128

17. save the results of the compare forecasts

Detailed Description of Applied Components

FILE_INTERFACE	
load booking and availability information	
cubes	
parameter	string: bkg,avail, UNDEFINED, DCP
applied dimensions	

FILE_INTERFACE	
load block element shift	
cubes	
parameter	string: blockElemShift, UNDEFINED, UNDEFINED
applied dimensions	

HB_EXP	
calculate first estimate for the attractiveness without consideration of unconstraining and seasonal effects	
cubes	bkg, phi0_h1ExpSm
parameter	float: 0.05,23,53,0,1000
applied dimensions	CW[appl,0,128],DCP[appl,0,22]

UNCONSTRAINING	
unconstrain the booking data	
cubes	bkg, phi0_h1ExpSm, avail, ucBkg, ucOffset
parameter	
applied dimensions	DCP[appl,0,22]

HB_EXP	
calculate second estimate for the attractiveness without consideration of seasonal effects	
cubes	ucBkg, phi0_h1ExpSm
parameter	float: 0.05,23,53,0,1000
applied dimensions	CW[appl,0,128],DCP[appl,0,22]

DATA_DECOMPOSITION	
calculate first estimate of seasonal factors First/Business compartment	
cubes	ucBkg, phi0_h1ExpSm, phi0_h1Hist, attrPrepared, seasonSmoothed
parameter	bool: 1, float: -0.5,3,0,1000,1
applied dimensions	DCP[appl,0,22],F[aggr,0,7],DOW[aggr,0,6]

DATA_DECOMPOSITION	
calculate first estimate of seasonal factors Economy compartment	
cubes	ucBkg, phi0_h1ExpSm, phi0_h1Hist, attrPrepared, seasonSmoothed
parameter	bool: 1, float: -0.5,3,0,1000,1
applied dimensions	DCP[appl,0,22],F[aggr,8,19],DOW[aggr,0,6]

DATA_SMOOTHING	
smooth the determined seasonal factors	
cubes	seasonSmoothed
parameter	float: 5,0.1
applied dimensions	CW[appl,0,128]

HB_EXP	
learn seasonal behaviour (first estimate)	
cubes	seasonSmoothed, phi0_h1Hist
parameter	float: 0.6,53,1,-1,1000
applied dimensions	CW[appl,0,128]

DATA_DECOMPOSITION	
data decomposition under consideration of historical behaviour of attractiveness and season	
cubes	ucBkg, phi0_h1ExpSm, phi0_h1Hist, attr, season
parameter	bool: 0, float: -1,3,0,1000,1
applied dimensions	DCP[appl,0,22]

HB_EXP	
learn seasonal behaviour (improved estimate)	
cubes	attr, phi0_h1ExpSm
parameter	float: 0.1,23,53,0,1000
applied dimensions	CW[appl,0,128],DCP[appl,0,22]

DATA_DECOMPOSITION	
final data decomposition real data (low level)	
cubes	ucBkg, phi0_h1ExpSm, phi0_h1Hist, attr, season
parameter	bool: 1, float: -1,3,0,1000,1
applied dimensions	DCP[appl,0,22]

DATA_DECOMPOSITION	
data decomposition used for learning (diversified level First/Business)	
cubes	ucBkg, phi0_h1ExpSm, phi0_h1Hist, attrPrepared, seasonSmoothed2
parameter	bool: 1, float: -1,3,0,1000,1
applied dimensions	DCP[appl,0,22], F[diversified,0,7], DOW[diversified,0,6]

DATA_DECOMPOSITION	
data decomposition used for learning (diversified level Economy)	
cubes	ucBkg, phi0_h1ExpSm, phi0_h1Hist, attrPrepared, seasonSmoothed2
parameter	bool: 1, float: -1,3,0,1000,1
applied dimensions	DCP[appl,0,22], F[diversified,0,7], DOW[diversified,0,6]

DATA_SMOOTHING	
smoothing of the diversified decomposed data	
cubes	seasonSmoothed2
parameter	float: 2,0.2
applied dimensions	CW[appl,0,128]

HB_EXP	
learning of the diversified history	
cubes	seasonSmoothed2, phi0_h2Hist
parameter	float: 0.6,53,1,-1,1000
applied dimensions	CW[appl,0,128]

DATA_DECOMPOSITION	
data decomposition used for forecasting (diversified level First/Business)	
cubes	ucBkg, phi0_h1ExpSm, phi0_h1Hist, attrPrepared, seasonPrepared
parameter	bool: 1, float: -1,3,0,1000,1
applied dimensions	DCP[appl,0,22],F[diversified,0,7], DOW[diversified,0,6]

DATA_DECOMPOSITION	
data decomposition used for forecasting (diversified level Economy)	
cubes	ucBkg, phi0_h1ExpSm, phi0_h1Hist, attrPrepared, seasonPrepared
parameter	bool: 1, float: -1,3,0,1000,1
applied dimensions	DCP[appl,0,22],F[diversified,8,19], DOW[diversified,0,6]

FC_ATTR	
forecast of the attractiveness component	
cubes	attr, phi0_h1ExpSm, phi1_h1ExpSm, fc_h1ExpSm, blockElemShift
parameter	bool: 0, 0, 1, float: 23
applied dimensions	CW[appl,0,128],DCP[appl,0,22],DCPFC[appl,0,22], DIV1[appl,0,0],DIV2[appl,0,0],DIV3[appl,0,0], DIV4[appl,0,0]

FC_SEASON	
forecast diversified seasonal factors	
cubes	season, seasonPrepared, phi0_h2Hist, attr, fc_h1ExpSm, fc_h1Hist
parameter	bool: 1, float: 1,-1,3,1
applied dimensions	DCP[appl,0,22],DCPFC[appl,0,22]

VALID_FC_REF	
calculate forecast errors diversified seasonal factors	
cubes	season, fc_h1Hist, err_h1Hist
parameter	bool: 0
applied dimensions	DCP[appl,22,22],DCPFC[appl,0,22]

HB_LINEAR_COMBINATION_STRUCTURE	
generation/evolution of a combination structure and calculation of linear combination weights	
cubes	err_h1Hist, season, lin_comb_weight, lin_comb_offset
parameter	float: 1,1,0,0,4,2,10,-1, string: DIV1,DIV2,DIV3,DIV4, UNDEFINED
applied dimensions	DIV1[appl,0,3],DIV2[appl,0,1],DIV3[appl,0,1], DIV4[appl,0,1],CW[appl,53,92],DOW[appl,0,6], DCP[appl,22,22]

LINEAR_COMBINATION	
combination of the diversified seasonal predictions	
cubes	fc_h1Hist, fc_comb, lin_comb_weight, lin_comb_offset
parameter	
applied dimensions	DIV1[appl,0,3],DIV2[appl,0,1],DIV3[appl,0,1], DIV4[appl,0,1],CW[appl,0,128]

COMBINING_ADD_PARTS	
calculation of the total forecast	
cubes	fc_h1ExpSm, fc_comb, fc_combined
parameter	
applied dimensions	

VALID_FC_REF	
calculation of combined total forecast errors	
cubes	ucBkg, fc_combined, err_combined_bias
parameter	bool: 0
applied dimensions	DCP[appl,22,22],DCPFC[appl,0,22]

ERROR_COVAR	
calculate mean absolute deviation low level	
cubes	err_combined_bias, mad_combined_low
parameter	bool: 1, float: 1
applied dimensions	CW[appl,93,128]

ERROR_COVAR	
calculate mean absolute deviation high level	
cubes	err_combined_bias, mad_combined_high
parameter	bool: 1, float: 60
applied dimensions	CW[appl,93,128],F[appl,0,19],POS[appl,0,2]

FILE_INTERFACE	
write result cubes combined forecast error variance low and high level	
cubes	
parameter	string: UNDEFINED, mad_combined_low,mad_combined_high, lin_comb_weight,lin_comb_offset, DCPFC
applied dimensions	

Variations of the Experiment

The shown results are generated by variation of parameters of component HB_LINEAR_COMBINATION_STRUCTURE.

The following parameter settings have been used in order to represent the different structures mentioned in Chapters 6 and 7:

structure	parameter
CEW	float: 0,0,0,2,2,0,5,1.4, string: DIV1,DIV2,DIV3,DIV4, UNDEFINED
EV1	float: 2,2,0,1,4,-1,10,1.4, string: DIV1,DIV2,DIV3,DIV4, UNDEFINED
EV2	float: 2,2,0,0,4,2,10,1.4, string: DIV1,DIV2,DIV3,DIV4, UNDEFINED
EV3	float: 2,2,2,0,4,2,10,1.4, string: DIV1,DIV2,DIV3,DIV4, UNDEFINED
EV4	float: 2,2,0,0,2,2,10,1.4, string: DIV1,DIV2,DIV3,DIV4, UNDEFINED
EV5	float: 2,2,0,1,2,-1,10,1.4, string: DIV1,DIV2,DIV3,DIV4, UNDEFINED
EV6	float: 0,0,0,0,4,2,10,-1, string: DIV1,DIV2,DIV3,DIV4, UNDEFINED
EV7	float: 0,1,0,0,4,2,10,-1, string: DIV1,DIV2,DIV3,DIV4, UNDEFINED
EV8	float: 1,0,0,0,4,2,10,-1, string: DIV1,DIV2,DIV3,DIV4, UNDEFINED
EV9	float: 1,1,0,0,4,2,10,-1, string: DIV1,DIV2,DIV3,DIV4, UNDEFINED

BIBLIOGRAPHY

[Aiolfi 04] M. Aiolfi & A.G. Timmermann. *Persistence of forecasting performance and conditional combination strategies.* In Press, Corrected Proof, available online at http://econ.ucsd.edu/ atimmerm/aiolfi_timmermann.pdf, 2004.

[Aiolfi 05] M. Aiolfi & C.A. Favero. *Model Uncertainty, Thick Modelling and the Predictability of Stock Returns.* Journal of Forecasting, vol. 24, pages 233–254, 2005.

[Aksu 92] C. Aksu & I.Gunter. *An empirical analysis of the accuracy of SA, OLS, ERLS and NRLS combinations of forecasts.* International Journal of Forecasting, vol. 8, pages 27–43, 1992.

[Armstrong 89] J. S. Armstrong. *Combining Forecasts: The End of the Beginning or the Beginning of the End?* International Journal of Forecasting, vol. 5, pages 585–588, 1989.

[Armstrong 01] J.S. Armstrong, editeur. Principles of forecasting: A handbook for researchers and practitioners. Kluwer Academic Publishers, 2001.

[Bates 69] J. Bates & C. Granger. *The Combination of Forecasts.* Operations Research Quarterly, vol. 20, pages 451–468, 1969.

[Batista 04] G.E.A.P.A. Batista, P.C. Prati & M.C. Monard. *A study of the behavior of several methods for balancing machine learning training data.* SIGKDD Explorations, pages 20 – 29, 2004.

[Bezdek 99] J.C. Bezdek, J. Keller & R. Krishnapuram. *Fuzzy models and algorithms for pattern recognition and image processing.* Kluwer Academic Publishers, Boston, MA, 1999.

[Breimann 96] L. Breimann. *Bagging predictors.* Machine Learning, vol. 24, pages 123–140, 1996.

[Brockwell 87] P.J. Brockwell & R.A. Davis. Time series theory and methods. *Springer, 1987.*

[Brown 63] R.G. Brown. Smoothing, forecasting and prediction. Prentice-Hall, 1963.

[Brown 05a] G. Brown, J.L. Wyatt, R. Harris & X. Xao. *Diveristy Creation Methods: A survey and categorisation.* Journal of Information Fusion, pages 5–20, 2005.

[Brown 05b] G. Brown, J.L. Wyatt & P. Tino. *Mangaging Diversity in Regression Ensembles.* Journal of Machine Learning Research, pages 1621–1650, 2005.

[Bunn 75] D.W. Bunn. *A Bayesian approach to the linear combination of forecasts.* Operations Research Quartlery, vol. 26, pages 325–329, 1975.

[Bunn 85] E.W. Bunn. *Statistical efficiency on the linear combination of forecasts.* International Journal of Foreasting, vol. 1, pages 151–163, 1985.

[Bunn 89] D.W. Bunn. *Forecasting with more than one model.* Journal of Forecasting, vol. 8, pages 161–166, 1989.

[Chawla 04] N.V. Chawla, N.Japkowicz & A. Kolcz. *Editorial: special issue on learning from imbalanced data sets.* SIGKDD Explorations, pages 1 – 6, 2004.

[Clemen 89] R.T. Clemen. *Combining Forecasts: A review and annotated bibliography.* International Journal of Forecasting, vol. 5, pages 559–583, 1989.

[Cross 97] R.G. Cross. Revenue management. Broadway Books, 1997.

[de Menezes 00] L.M. de Menezes, D. W. Bunn & J.W. Taylor. *Review of guidelines for the use of combined forecasts.* European Journal of Operations Research, vol. 120, pages 190–204, 2000.

[Dempster 77] A. Dempster, N. Laird & D.Rubin. *Maximum likelihood from incomplete data via the EM algorithm.* J. Royal Statistical Society Series B, pages 1– 38, 1977.

[Deutsch 94] M. Deutsch. *The combination of forecasts using changing weights.* International Journal of Forecasting, vol. 10, pages 47–57, 1994.

[Druckner 94] H. Druckner. *Boosting and other ensemble methods.* Neural Computation, vol. 6, pages 1289–1301, 1994.

[Dunis 01] C. Dunis, A.G. Timmermann & J. Moody, editeurs. Developments in forecast combination and portfolio choice. John Wiley & Sons, 2001.

[Elliott 05] G. Elliott & A. Timmermann. *Optimal forecast combination under regime switching.* International Economic Review, 2005.

[Elliott 07] G. Elliott & A. Timmermann. *Economic Forecasting.* CEPR Discussion Paper no. 6158. London, Centre for Economic Policy Research. http://www.cepr.org/pubs/dps/DP6158.asp., 2007.

[Fiordaliso 98] A. Fiordaliso. *A nonlinear forecasts combination method based*

on Takagi- Sugeno fuzzy systems. IInternational Journal of Forecasting, vol. 14, pages 367–379, 1998.

[Fischer 99] I. Fischer & N. Harvey. *Combining forecasts: What information do judges need to outperform the simple average.* International Journal of Forecasting, vol. 15, pages 227–246, 1999.

[Fliedner 01] G. Fliedner. *Hierarchical forecasting: issues and use guidelines.* Industrial Management & Data Systems, vol. 101, pages 5 – 12, 2001.

[Flores 89] B.E. Flores & E.M. White. *Subjective versus objective combining forecasts: An experiment.* Journal of Forecasting, vol. 8, pages 331–341, 1989.

[Franses 63] P.H. Franses. Time series models for business and economic forecasting. Cambridge University Press, 1963.

[Freund 96] Y. Freund & R. Shapire. *Experiments with a new boosting algorithm.* Machine Learning: Proceedings of the Thirteenth International Conference, 1996.

[Gabrys 02] B. Gabrys. *Combining Neuro- Fuzzy Classifiers for Improved Generalisation and Reliability.* Proceedings IJCNN 2002 a part of WCCI 2002 Congr., Honolulu, USA, pages 2410 – 2415, 2002.

[Gabrys 03] B. Gabrys. *Learning hybrid neuro- fuzzy classifier models from data: to combine or not to combine?* Fuzzy Sets and Systems, vol. 147, pages 39 – 46, 2003.

[Geman 92] S. Geman, E. Bienenstock & R. Doursat. *Neural Networks and the Bias- Variance Dilemma.* Neural Computation, vol. 4, 1(1992), pages 1–58, 1992.

[Genest 86] C. Genest & J.V. Zideck. *Combining probability distributions: a critique and annotated bibliography.* Statistical Science, vol. 1, pages 114–148, 1986.

[Ghahramani 94] Z. Ghahramani & M.I. Jordan. *Supervised learning from incomplete data via an EM approach.* Cowan, J., Tesauro, G., Alspector, J., Advances in Neural Information Processing Systems 6. Morgan Kaufmann, 1994.

[Granger 84] C.W.J. Granger & R. Ramanathan. *Improved methods of forecasting.* Journal of Forecasting, vol. 3, pages 197–204, 1984.

[Granger 86] C.W.J. Granger & P. Newboldt. Forecasting economic time series. Academic Press, Inc., 1986.

[Granger 98] C.W.J. Granger. *Invited review combining forecasts- Twenty years later.* Journal of Forecasting, vol. 8, pages 167–173, 1998.

[Granger 04] C.W.J. Granger & Y. Jeon. *Thick modeling.* Econometric Modeling, vol. 21, pages 323–343, 2004.

[Guidolin 07] M. Guidolin & A. Timmermann. *Forecasts of US Short-term Interest Rates:A Flexible Forecast Combination Approach.* Working Paper 2005-059C, available online at http://research.stlouisfed.org/wp/2005/2005-059.pdf, 2007.

[Gunter 92] S.I. Gunter. *Nonnegativity restricted least squares combinations.* International Journal of Forecasting, vol. 8, pages 45–59, 1992.

[Hall 92] D. Hall. Mathematical techniques in multisensor data fusion. Artech House, Norwood, 1992.

[Hansen 90] L.K. Hansen & P. Salomon. *Neural network ensembles.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, pages 993–1000, 1990.

[Hansen 00] J.V. Hansen. Combining predictors. meta machine learning methods and bias/variance and ambiguity decompositions. PhD Dissertation, 2000.

[Hashem 96] S. Hashem. *Effects of Collinearity on Combining Neural Networks.* Connection Science, vol. 8, pages 315–336, 1996.

[Hathaway 96] R.J. Hathaway, J.C. Bezdek & W.P. Pedrycz. *A parametric model for fusing heterogeneous fuzzy data.* IEEE Trasactions on Fuzzy Systems, pages 270–281, 1996.

[He 05] C. He & X. Xu. *Combination of Forecasts Using Self- organising Algorithms.* Journal of Forecasting, pages 269–278, 2005.

[Holden 90] K. Holden & D.A. Peel. *Unbiasedness, Efficiency and the combination of economic forecasts.* Journal of Forecasting, vol. 8, pages 175–188, 1990.

[Jacobs 95] R.A. Jacobs. *Methods for combining experts probability assessments.* Neural Computation, vol. 7, pages 867–888, 1995.

[James 96] G. James & T. Hastie. *Generalisations of the bias/variance decomposition for prediction error.* Tech. rep., http://www.stat.stanford.edu/ gareth/ftp/papers/bv.ps, 1996.

[Jang 93] J.S.R. Jang. *ANFIS: Adaptive Network- based Fuzzy Inference Systems.* IEEE Transactions on Systems, Man and Cybernetics, vol. 23, pages 665–685, 1993.

[Jordan 95] M.I. Jordan & R.A. Jacobs. *Modular and Hierarchical Learning Systems.* In M.A. ARbib (Ed.): The Handbook of Brain

Theory and Neural Networks. Bradford Books/ MIT Press, pages 579 – 581, 1995.

[Keller 97] J.M. Keller & P. Gader. *Fuzzy logic and sensor fusion.* oral presentation at the 1st US Army Multidisciplinary U. Research Initiative on Demining, 1997.

[Kennedy 92] P. Kennedy. A guide to econometrics. The MIT Press, 1992.

[Klapper 98a] M. Klapper. *The influence of the variance- covariance structure on the performance of forecast combining techniques.* Technical report 39, sfb 475, University of Dortmund, 1998.

[Klapper 98b] M. Klapper. *Multivariate rank- based forecast combining techniques.* Technical report, sfb 475, University of Dortmund, 1998.

[Kotsiantis 03] S. Kotsiantis & P. Pintelas. *Mixture of expert agents for handling imbalanced data sets.* Annals of Mathematics, Computing and TeleInformatics, pages 46 – 55, 2003.

[Kotsiantis 06] S. Kotsiantis & D. Kanellopoulos. *Association Rules Mining: a recent overview.* GESTS international transactions on Computer Science and Engineering, pages 71 –82, 2006.

[Koza 92] J.R. Koza. Genetic programming. on the programming of computers by means of natural selection. MIT Press, 1992.

[Krogh 95] A. Krogh & J. Vedesby. *Neural network ensembles, cross validation and active learning.* G. Tesauso et al. (ed.), Advances in Neural Information Processing Systems, vol. 7, pages 231–238, 1995.

[Kuncheva 01] L.I. Kuncheva & C.J. Whitaker. *Ten measures of diversity in classifier ensembles: limits for two classifiers.* Proceedings

of the IEE Workshop on Intelligent Sensor Processing, vol. 1, pages 10/1–10/6, 2001.

[Kuncheva 03] L.I. Kuncheva. *That Elusive Diversity in Classifier Ensembles.* In First Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA), 2003.

[Littlestone 92] N. Littlestone. *The Weighted Mojority Algorithm.* Technical Report UCSC CRL-91-28, 1992.

[Liu 98] Y. Liu. Negative correlation learning and evolutionary neural netowrk ensembles. PhD thesis, University College, The University of New South Wales, Australian Defense Force Academy, Camberra, Australia, 1998.

[MacDonald 94] R. MacDonald & I.W. Marsh. *Combining exchange rate forecasts: What is the optimal consensus measure?* Journal of Forecasting, vol. 13, pages 313–333, 1994.

[Maclin 95] R. Maclin & J.W. Shavlik. *Combining the predictions of multiple classifiers: Using competitive learning to initialize neural networks.* Proceedings of the 14th International Conference on Artificial Intelligence, IJCAI-95, 1995.

[Makridakis 82] S. Makridakis. *The accuracy of extrapolation (time series) methods: results of a forecast competition.* Journal of Forecasting, vol. 1, pages 111–153, 1982.

[Makridakis 93] S. Makridakis. *The M2 Competition: A real-time judgementally based forecasting study.* International Journal of Forecasting, vol. 9, pages 5–22, 1993.

[Masters 95] T. Masters. Neural, novel & hybrid algorithms for time series prediction. John Wiley & Sons, 1995.

[McGill 99] McGill & van Ryzin. *Revenue Management: Research Overview and Prospects.* Transportation Science, vol. 33, 4, 1999.

[Merz 97] C.J. Merz & M.J. Pazzani. *A Principal Components Approach to Combining Regression Estimates.* Machine Learning, vol. 0, pages 1–25, 1997.

[Negnevitsky 05] M. Negnevitsky. Artificial intelligence: A guide to intelligent systems. Addison Wesley, 2 edition, 2005.

[Neuling 04] R. Neuling, S. Riedel & K.U. Kalka. *New approaches to origin and destination and no-show forecasting: Excavating the passenger name records treasure.* Journal of Revenue and Pricing Management, vol. 3, 1(2004), pages 62–72, 2004.

[Newboldt 74] P. Newboldt & C.W.J. Granger. *Experience with forecasting univariate time series and the combination of forecasts (with discussion).* Journal of the Royal Statistical Society, vol. Series A 137, pages 131–149, 1974.

[Nilsson 96] N.J. Nilsson. *Learning Machines: Foundations of trainable pattern- classifying systems.* NY: McGraw Hill, 1996.

[Nowlan 91] S. Nowlan. *Soft competitive adaptation: Neural network learning algorithms based on fitting statistical mixtures.* Tech report CS-91-126, Carnegie Mellon University., 1991.

[Opitz 99a] D. Opitz & R. Maclin. *Popular ensemble methods: An empirical study.* Journal of Artificial Intelligence Research, vol. 11, pages 169–198, 1999.

[Opitz 99b] D.W. Opitz & J.W. Shavlik. *A Genetic Algorithm Approach*

for Creating Neural-Network Ensembles. Combining Artificial Neural Nets, pages 79–97, 1999.

[Ozun 07] A. Ozun & A. Cifter. *Nonlinear Combination of Financial Forecast with Genetic Algorithm.* MPRA paper, available online at http://mpra.ub.uni-muenchen.de/2488/, 2007.

[Pak 02] K. Pak & N. Piersma. *Overview of OR techniques for airline revenue management.* Statistica Neerlandica, vol. 56, pages 479–495, 2002.

[Pedrycz 98] W. Pedrycz, J.C. Bezdek & R.J. Hathaway. *Two nonparametrical models for fusing heterogeneous fuzzy data.* IEEE Transactions on Fuzzy Systems, pages 270–281, 1998.

[Perrone 93] M. Perrone & L.N.Cooper. *When networks disaggree: Ensemble methods for hybrid neural networks.* R.I. Mammone (ed.), Neural Networks for Speech and Image Processing, London, Chapman and Hall, 1993.

[Provost 00] F. Provost. *Learnig with imbalanced data sets.* Proceedings of the AAAI'2000 Workshop of Imbalanced Data Sets., 2000.

[Rao 82] C.R. Rao. *Diversity: Its measurement, decomposition, apportionment and analysis.* Sankya: The Indian Journal of Statistics, vol. 44, pages 1–22, 1982.

[Raviv 96] Y. Raviv & N. Intrator. *Bootstrapping with Noise: An Effective Regularization Technique.* Connection Science, vol. 8, pages 355–372, 1996.

[Riedel 03] S. Riedel & B. Gabrys. *Adaptive Mechanisms in an Airline Ticket Demand Forecasting System.* Proceedings of the EUNITE'2003 conference, Oulu, Finland, 2003.

[Riedel 04] S. Riedel & B. Gabrys. *Hierarchical Multilevel Approaches of Forecast Combination.* Proceedings of the OR'2004 conference, Netherlands, 2004.

[Riedel 05a] S. Riedel & B. Gabrys. *Combination of Multi Level Forecasts.* accepted to Special Issue related to Data Fusion, Journal of VLSI Signal Processing Systems, 2005.

[Riedel 05b] S. Riedel & B. Gabrys. *Evolving Multilevel Forecast Combination Models - An Experimental Study.* Proceedings of the NISIS conference 2005, Albufeira, Portugal, 2005.

[Riedel 07a] S. Riedel & B. Gabrys. *Dynamic Pooling for the Combination of Forecasts generated using Multi Level Learning.* accepted for Effective Ensemble Methods -Special Session at IJCNN'2007, 2007.

[Riedel 07b] S. Riedel & B. Gabrys. *Pooling for Combination of Multi Level Forecasts.* Submitted for consideration to the IEEE Transactions on Knowledge and Data Engineering, 2007.

[Rogova 94] G. Rogova. *Combining the results of several neural network classifiers.* Neural Networks, vol. 7, pages 777–781, 1994.

[Rosen 96] B.E. Rosen. *Ensemble learning using decorrelated neural networks.* Connection Science, Special Issue on Combining Artificial Neural Networks: Ensemble Approaches, vol. 8, pages 373–384, 1996.

[Russell 87] T.D. Russell & E.E. Adam. *An empirical evaluation of alternative forecast combinations.* Management Science, vol. 33, pages 1267–1276, 1987.

[Ruta 00] D. Ruta & B. Gabrys. *An Overview of Classifier Fusion Methods*. in: Prof. M. Crowe (Ed.), Computing and Information Systems, University of Paisley, vol. 7, pages 1 – 10, 2000.

[Ruta 02] D. Ruta & B. Gabrys. *Set Analysis of Coincident Errors and Its Applications for Combining Classifiers*. Pattern Recognition and String Matching, Eds. D.Chen and X.Cheng, ISBN 1-4020-0953-4, Kluwer Academic Publishers,, 2002.

[Ruta 05] D. Ruta & B. Gabrys. *Classifier Selection for Majority Voting*. Special issue of the journal of information fusion on Diversity in Multiple Classifier Systems, vol. 6, pages 63–81, 2005.

[Ruta 07] D. Ruta & B. Gabrys. *Neural Network Ensembles for Time Series Prediction*. accepted for Effective Ensemble Methods -Special Session at IJCNN'2007, 2007.

[Sancetta 07] A. Sancetta. *Online Forecast Combination for Dependent Heterogeneous Data*. 2007.

[Schapire 90] R.E. Schapire. *The strength of weak learnability*. Machine Learning, vol. 5, pages 197–227, 1990.

[Schmittlein 90] D.C. Schmittlein, J.Kim & D.G. Morrison. *Combining forecasts: Optimal adjustments to theoretically optimal rules*. Management Science, vol. 36, pages 1044–1056, 1990.

[Sharkey 95] A.J.C. Sharkey & N.E. Sharkey. *How to improve the reliability of artificial neural networks*. Research Report CS-95-11, Department of Computer Schience, Univercity of Sheffield, 1995.

[Sharkey 96] A.J.C. Sharkey. *On Combining Artificial Neural Nets*. Connection Science, vol. 8, pages 299–313, 1996.

[Shi 99] S.M. Shi. *Improving the accuracy of nonlinear combined forecasting using neural networks.* Expert Systems wih Applications, vol. 16, pages 49–54, 1999.

[Talluri 04] K.T. Talluri & G.J. von Rysin. The theory and practice of revenue management. Springer, 2004.

[Theil 91] H. Theil. Applied ecoomic forecasting. Holland Publishing Company, Amsterdam, 1991.

[Timmermann 05] A.G. Timmermann. *Forecast Combinations.* Discussion Paper No. 5361, www.cepr.org/pubs/dps/DP5361.asp, 2005.

[Timmermann 06] A. Timmermann. Handbook of economic forecasting. Elsevier North Holland, 2006.

[Tumer 96] K. Tumer & J. Gosh. *Analysis of Decision Bounderies in linearly combined neural classifiers.* Pattern Recognition, vol. 26, pages 341–348, 1996.

[Waibel 89] A. Waibel, H. Sawai & K. Shikano. *Modularity and scaling in large phonemic neural networks.* IEEE Transactions on Accoustics, Speech and Signal Processing, vol. 37, pages 1888 – 1898, 1989.

[Weatherford 92] L.R. Weatherford & S.E. Bodily. *A Taxonomy and Research Overview of Perishable-Asset Revenue Management: Yield Management, Overbooking, and Pricing.* Operations Research, vol. 40, pages 831–844, 1992.

[Weiss 04] G. M. Weiss. *Mining with rarity: a unifying framework.* SIGKDD Explorations, pages 7 – 19, 2004.

[Winkler 83] R.L. Winkler & S. Makridakis. *The combination of forecasts.*

Journal of the Royal Statistical Society, vol. 146, pages 150–157, 1983.

[Witten 05] I. H. Witten & F.Eibe. Data mining: practical machine learning tools and techniques. Elsevier Inc., 2005.

[Xu 92] L. Xu. *Methods of combining multiple classifiers and their application to handwriting recognition.* IEEE Transactions on Systems, Man, Cybernetics, vol. 22, pages 418–435, 1992.

[Yang 04] Y. Yang. *Combining Forecasting Procedures: Some Theoretical Results.* Econometric Theory, pages 176–222, 2004.

[Zaki 00] H. Zaki. *Forecasting for airline revenue management.* J. Bus. Forecast. Methods Syst., vol. 19, pages 2–7, 2000.

[Zhou 01] Z.H. Zhou. *Combining regression estimators: GA-based selective neural network ensemble.* International Journal of Computational Intelligence and Applications, vol. 1, pages 341–356, 2001.