Information Preserving Multi-Objective Feature Selection for Unsupervised Learning

Ingo Mierswa
Artificial Intelligence Unit
Department of Computer Science
University of Dortmund
ingo.mierswa@uni-dortmund.de

Michael Wurst
Artificial Intelligence Unit
Department of Computer Science
University of Dortmund
wurst@ls8.cs.uni-dortmund.de

ABSTRACT

In this work we propose a novel, sound framework for evolutionary feature selection in unsupervised machine learning problems. We show that unsupervised feature selection is inherently multi-objective and behaves differently from supervised feature selection in that the number of features must be maximized instead of being minimized. Although this might sound surprising from a supervised learning point of view, we exemplify this relationship on the problem of data clustering and show that existing approaches do not pose the optimization problem in an appropriate way. Another important consequence of this paradigm change is a method which segments the Pareto sets produced by our approach. Inspecting only prototypical points from these segments drastically reduces the amount of work for selecting a final solution. We compare our methods against existing approaches on eight data sets.

Track: Learning Classifier Systems and other Genetics-Based Machine Learning

Categories and Subject Descriptors: I.5.2 [Computing Methodologies]: Pattern Recognition

General Terms: Algorithms, Experimentation

Keywords: Multi-objective feature selection, unsupervised learning, Pareto front segmentation

1. INTRODUCTION

Feature selection for unsupervised learning is a challenging new application in the field of genetics-based machine learning. In this work we show how the rigorous definition of competing criteria leads to a novel and sound theoretical framework based on multi-objective optimization. Our approach creates Pareto sets which share an interesting property: depending on the number of inherent patterns each front shows several kinks. These kinks allow an interpretable segmentation from which the user can select few prototypes which drastically reduces the effort of selecting a final solution. This turns unsupervised feature selection into a case study for automatic segmentation of highly complex Pareto sets.

Today, machine learning consists of two paradigms: supervised and unsupervised learning. For supervised learning a set of labeled data points must be given. The learning method should merely find a function which *predicts* the label for unseen data points. Supervised learning methods cannot be applied if no information is known beforehand. Unsupervised machine learning should rather *describe* the data set. Hence, the task is to automatically find the inherent, natural patterns of the data. Such natural patterns can express useful information for a decision maker. Typical application areas include customer segmentation, information retrieval, and image analysis [13, 17].

The search for a proper supervised prediction function can usually be formulated as an optimization problem where the number of wrong predictions for the known data points should be minimized. A similar criterion for validity does not exist in the unsupervised setting. The optimization function of an unsupervised algorithm cannot rely on given patterns in order to decide if a found pattern is "correct" or "wrong". The validity of discovered patterns also depends on the background knowledge and intention of the user. It is therefore often desirable that unsupervised learning methods present more than one solution to the user.

One of the main problems for both supervised and unsupervised learning algorithms is to decide which dimensions of the data space should be taken into account. The dimensions of the data space are called *features*, the corresponding selection problem is called *feature selection*. The prediction accuracy of a learned decision function can be dramatically increased if redundant or noisy features are omitted during learning. Although the problem is the same for both learning paradigms, the consequences might be rather different. The supervised feature selection problem can be solved by minimizing the number of used features while prediction accuracy is preserved. The search for the best feature subset out of all 2^M possible subsets usually requires heuristics for larger dimensions M. Genetic algorithms have demonstrated their ability to solve this problem several times before [18, 22]. In addition, minimizing the number of features and maximizing prediction accuracy is a multi-objective optimization problem since removing necessary features from the data set will decrease accuracy [7].

The same problem exists for unsupervised learning. The existence of noisy or redundant features can cover inherent data clusters and omitting those features might reveal the actual natural patterns. There are several approaches

GECCO'06, July 8–12, 2006, Seattle, Washington, USA.

which try to directly identify promising feature subsets for clustering [19]. However, these approaches do not reflect the multi-objective character of the problem setting. Therefore, state of the art feature selection approaches for unsupervised learning use multi-objective optimization. They transfer the idea of minimizing the number of features while the clustering optimization criterion should be preserved [14, 15, 16].

Defining the optimization problem in this way is not appropriate. Under very weak assumptions we will show that the Pareto set will collapse into one singular point. Even if the population does not collapse it tends to cover only small fraction of the solution space.

Nevertheless, there *is* a trade-off between the number of features and cluster validity. However, the number of features must be *maximized* instead of minimized in order to achieve this competition. Although this might sound surprising at first, the change of optimization direction has a natural origin in the aim of unsupervised learning. In order to describe the data set at hand the amount of information which could be derived from the used feature set should be preserved during the feature selection process.

We will solve the corresponding multi-objective optimization problem with NSGA-II. The resulting Pareto sets are more beneficial for users since they provide a larger coverage of possible candidate solutions. The Pareto fronts produced by our selection approach can also be segmented into meaningful regions. This eases the selection of a final solution from the set of Pareto optimal points. We enable feature selection for density based clustering schemes as well. In contrast to combinatorial clustering algorithms like k-means these clustering algorithms are able to find non-Gaussian clusters like rings or spirals. Hence, we propose an improved feature selection approach which is applicable to a wide variety of clustering algorithms and provides interpretable segmentations of the complex Pareto set.

1.1 Outline

In section 2 we will introduce the problem of data clustering and corresponding optimization criteria for unsupervised learning. In section 3 we will discuss existing approaches for unsupervised feature selection. Although the transfer from supervised learning is an appealing idea we will show that these approaches will not lead to complete Pareto fronts for this type of problem. In section 4 we will discuss how simply changing the optimization direction for one of the criteria leads to a natural multi-objective optimization problem which will be solved with NSGA-II. Furthermore, we will introduce a segmentation procedure for the resulting Pareto fronts. Section 5 presents results on several artificial and real-world data sets and compares the discussed approaches. Section 6 concludes this paper.

2. DATA CLUSTERING

One of the most important approaches to unsupervised learning is data clustering. The aim of cluster analysis is to group data points into sets of similar data points. Given a data set X, an individual data point is denoted as $x_i \in X$. A cluster is a subset of data points $C_k \subseteq X$. In principle, clusters may overlap. However, most clustering algorithms are designed to produce partitions of data points, i. e. a set of clusters $C_1 \dots C_K$ such that $C_k \cap C_l \neq \emptyset \Rightarrow C_k = C_l$ (clusters do not overlap) and $\bigcup_{k=1}^{K} C_k = X$ (each data point is covered by a cluster).

2.1 Combinatorial clustering algorithms

Cluster analysis aims at assigning items to clusters where elements within a cluster are more similar to each other than to the elements of other clusters. This notion can be expressed as optimization problem using a distance measure $d(x_i, x_j)$ on the set of data points:

$$W_d = \sum_{k=1}^{K} \sum_{x_i \in C_k} \sum_{x_j \in C_k} d(x_i, x_j). \tag{1}$$

It can be shown that by minimizing W_d the mean pairwise difference between pairs of points in different clusters is maximized [12]. A very efficient approach optimizing this function is k-means clustering. It is based on the squared Euclidean distance. In the following we assume that the data points are represented by a set of M real valued features, i. e. $x_i \in \mathbb{R}^M$, and that x_{im} is the value of the m-th feature for data point x_i. The Euclidean distance of two points x_i and x_j is

$$d(x_i, x_j) = \sqrt{\sum_{m=1}^{M} (x_{im} - x_{jm})^2}. \tag{2}$$

It can be shown that optimizing W_d with respect to the squared Euclidean distance is equivalent to optimizing the function

$$W = \sum_{k=1}^{K} \sum_{x_i \in C_k} \sum_{m=1}^{M} (x_{im} - c_{km})^2 \tag{3}$$

where c_{km} is the m-th value of the centroid of cluster C_k. The centroid is the point with the smallest distance to all points in C_k. It can be calculated as

$$c_{km} = \frac{\sum_{x_i \in C_k} x_{im}}{|C_k|}. \tag{4}$$

The k-means algorithm uses this relationship by applying an alternating optimization procedure [11]. In each step every data point is assigned to the cluster with the nearest centroid. Then the centroids are recalculated for each cluster and data points are newly assigned to clusters based on the new centroids. This alternation stops if there is no further change in cluster assignment or after a maximal number of steps. The centroids are initialized with random data points drawn from X. k-means does not guarantee to find an optimal solution and is sensitive to the choice of initial points. Therefore, a common strategy is to start k-means several times with different random initializations. For its simplicity and efficiency, k-means is one of the most popular clustering algorithms.

A natural choice for evaluating a set of clusters produced by k-means is W, i. e. the function it optimizes. This measure has several drawbacks. Most important, it is not normalized with respect to the feature values and to the number of clusters. With an increasing number of clusters, W decreases monotonically. W decreases as well for a decreasing number of features. For these reasons, it is not well suited as criterion for unsupervised feature selection problems.

Several other evaluation measures for k-means were proposed. Probably the most important is the Davies-Bouldin

(DB) index [4]. It is calculated as

$$DB = \frac{1}{K}\sum_{k=1}^{K} \max_{k,l \neq k} \left\{ \frac{S_k + S_l}{d(c_k, c_l)} \right\} \quad (5)$$

where S_k and S_l is the average within cluster distances for cluster C_k and C_l respectively which is defined as

$$S_k = \frac{1}{|C_k|} \sum_{x_i \in C_k} d(x_i, c_k). \quad (6)$$

The DB Index takes into account only the relative separation of the two clusters which are worst separated. This value is normalized as it is divided by the distance between the corresponding centroids. Therefore, it is less sensitive to the number of clusters and the number of features. Focusing on the clusters that are worst separated allows for a very fine grained optimization, as the parts of the clustering that are well separated do not overshadow these important parts. Please note that DB is only used in order to evaluate a cluster structure. There is no efficient algorithm which optimizes DB directly.

2.2 Gaussian mixtures

Gaussian mixture clustering assumes that the underlying data generating process consists of a mixture of different overlapping Gaussian distributions and that each distribution represents an individual cluster. Clustering is achieved by estimating the parameters of the underlying distributions and assigning each data point to the distribution that most likely produced it. This notion can be formalized by

$$L = \log \sum_{x_i \in X} \sum_{k=1}^{K} p_k g_k(x_i | \mu_k, \Sigma_k) \quad (7)$$

where g_k is a multivariate Gaussian distribution representing cluster C_k with mixture parameters μ_k and Σ_k. The probability p_k is the apriori probability of a data point belonging to cluster C_k.

The expectation maximization (EM) approach employs an alternating optimization procedure similar to the one applied in k-means clustering [6]. First each data point is assigned to the distribution that most likely generated it, then the parameters of all distributions (and thus clusters) are recalculated based on the data points in each cluster. This procedure is repeated until no further change occurs or a maximum number of steps is reached.

2.3 Graph-based clustering

Instead of assigning data points to clusters directly, we can first impose a graph (X, E) on these points. Two data points x_i and x_j are connected if they are sufficiently similar to each other. For a maximum distance threshold d_{max}, x_i and x_j are connected if $d(x_i, x_j) \leq d_{max}$. The by far most common clustering criterion is to regard clusters as connected components in the distance graph. This approach is denoted as *single link clustering*. Finding such connected components in a graph is usually achieved by a graph search [3].

The value d_{max} determines the coherence of the resulting clusters. If a fixed number of clusters K is given, we can find a maximal value for d_{max} such that the graph contains at most K connected components and thus clusters. In this case, the resulting d_{max} can be used as evaluation measure as it denotes the strength of the weakest link in the cluster structure. The weakest link is the point at which a given cluster would split up into two components if d_{max} would be decreased.

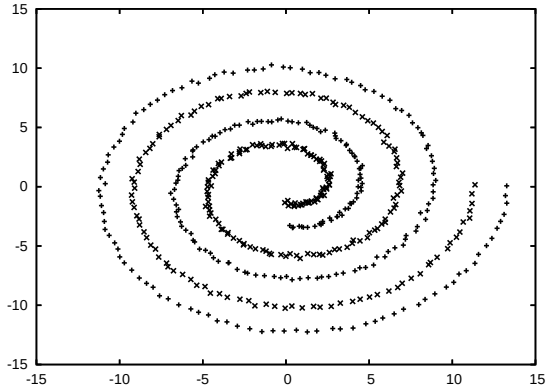


Figure 1: The SPIRAL data set created for single link clustering.

Graph based clustering is very popular and can be combined with many advanced connectivity criteria as in *density based clustering* [8] or *support vector clustering* [1]. In contrast to k-means and Gaussian mixtures these algorithms can detect clusters of any shape while k-means is limited to find spherical clusters boundaries. Figure 1 shows an example of a structure that cannot be clustered with neither EM nor k-means. Structures like these play an important role in applications as image recognition or astronomical data analysis [20].

3. MULTI-OBJECTIVE FEATURE SELECTION FOR CLUSTERING

Multi-objective optimization is a natural choice for selecting appropriate feature subsets for clustering problems. The current state of the art is represented by the work described in [14, 15] and [16]. In the following we will describe both approaches and show that they are both limited in several ways. These limitations are a result of the way the multi-objective optimization problem is posed.

Kim, Street, and Menczer introduce four performance criteria for k-means clustering[1][14, 15]. The first one is a variant of W that is normalized by the number of features

$$W_{norm} = \frac{1}{M} W. \quad (8)$$

A variant of between cluster distance is used as a second measure. However, this measure behaves essentially in the same way as normalized W (minimizing within cluster distance is equivalent to maximizing between cluster distance). The third measure represents the number of clusters K which should be minimized. The last measure captures the number of features nf that should be minimized as well.

In the following theorem we show that for a given number of clusters K minimizing W_{norm} and the number of features leads to exactly one Pareto optimal point. This optimal point always selects one single feature from the dataset, in

[1]In the original work all criteria are normalized by a constant. This, however, has no influence on Pareto optimality.

particular the one that leads to a minimal loss with respect to the used clustering performance criterion.

THEOREM 1. *Minimizing W_{norm} and the number of features nf leads to one single Pareto optimal point.*

Proof: For W_{norm} we can denote the loss of an individual feature m as

$$a_m = \sum_{k=1}^{K} \sum_{x_i \in C_k} (x_{im} - c_{km})^2 \tag{9}$$

In order to minimize the number of features selecting only one feature is optimal. We show that always

$$W_{norm} \geq \min_{1 \leq m \leq M} \{a_m\}. \tag{10}$$

That means that the performance can only decrease by adding any feature but the one that optimizes a_m. It can easily be seen that

$$W_{norm} = \frac{1}{M} \sum_{m=1}^{M} a_m \tag{11}$$

$$\geq \frac{1}{M} \sum_{m=1}^{M} \min_{1 \leq m \leq M} \{a_m\} \tag{12}$$

$$= \min_{1 \leq m \leq M} \{a_m\} \tag{13}$$

Using W_{norm} for optimization is not a well suited approach for feature selection in clustering problems as it leads to trivial solutions. A similar proof can be given for normalized between cluster distance.

In [16] a normalized variant of DB is proposed as alternative performance criterion to W_{norm}:

$$DB_{norm} = \frac{1}{M} DB. \tag{14}$$

This approach is better suited, as DB is normalized with respect to the feature space. However, this criterion is very sensitive. If the feature set contains for example a real valued feature that takes discrete values only, then choosing this one feature is again Pareto optimal. However, this one feature does almost certainly not represent the complete dataset in the descriptive sense mentioned in the introduction. In section 5 we will see several examples for which the Pareto set collapses into a single trivial solution.

Furthermore, using normalized DB does lead to a "build-in" competition between the number of features M and the cluster quality measure $\frac{1}{M} DB$. Therefore, even clustering random data produces a Pareto front that exhibits a $1/x$ relationship (see figure 2 (c)).

The major problem of both approaches is that they do not pose the problem correctly from the point of view of multi-objective optimization. In the next section we give an alternative problem formulation that solves the described difficulties.

4. INFORMATION PRESERVING FEATURE SELECTION

In the last sections we discussed several quality measurements for different clustering schemes. In the following we assume that all criteria should be maximized during feature selection. Criteria which should be minimized in the original problem definition are therefore multiplied by -1. In contrast to the existing approaches discussed in section 3 we do not minimize the number of features but maximize nf. This prevents the algorithm from selecting trivial solutions and leads to more complete Pareto sets of diversive natural clusterings. The fitness evaluation is done by performing a clustering scheme on the reduced feature sets. Depending on the used scheme we use DB (equation 5) for k-means clustering and d_{max} for single link clustering. Since there is a natural competition between maximizing the number of features nf and the selected cluster criterion we do not need to apply an artificial normalization factor.

The feature selection problem is inherently multi-objective and cannot be solved with single-objective evolutionary algorithms. In the clustering setting the user has no idea of criteria weights and, furthermore, there exist no simple decision about correct or wrong clusterings. Such a decision would totally depend on the amount of information the user can obtain from different clusterings. Therefore, we try to maintain as much information as possible and aim at finding all solutions which are optimal for arbitrary criteria weight vectors. These solutions are called *Pareto-optimal*.

Evolutionary algorithms can optimize more than one target function by introducing special selection operators [23]. Due to the population based approach of evolutionary algorithms a broad selection of Pareto-optimal solutions can be found during one run. The user can select one of these solutions after optimization. Additionally, multi-objective evolutionary algorithms do not strongly depend on form and continuity of the Pareto-optimal set [2]. We will see that for clustering with non-normalized optimization criteria the Pareto front is neither nicely shaped nor continuous.

We use NSGA-II as a multi-objective feature selection wrapper [5]. NSGA-II employs a selection technique which first sorts all individuals into levels of non-domination. Individuals from the first levels are added to the next generation until the desired population size is reached. Before adding individuals from the last possible level this level is sorted with respect to the crowding distance in order to preserve diversity in the population.

Individuals are bit vectors of length M indicating if a feature should be selected or not. The population size is set to $2M$, the maximal number of generations is 1000. A bit flip mutation is performed with probability $1/M$ and uniform crossover with probability 0.9.

4.1 Finding Interesting Points in the Pareto front

The Pareto plots derived by the multi-objective optimization procedure described in the last section show a clear structure. This structure is not accidental but reflects structure in the underlying data. We can exploit this Pareto plot structure in order to discover patterns in the underlying data set. Moreover, exploiting this structure drastically reduces the effort of selecting a final solution from the Pareto set.

The basic idea is to find points at which the trade-off between the number of features and the cluster quality significantly changes. As this trade-off is represented by the slope of the Pareto plot at a given point, we want to find points where the change in slope is maximal. In the following, this notion is formalized. We assume that all Pareto optimal points are sorted and that the p-th point is denoted by the pair (DB_p, nf_p). The value α_p then represents the

slope at point p by

$$\alpha_p = \arctan\left(\frac{DB_{p+1} - DB_p}{nf_{p+1} - nf_p}\right). \quad (15)$$

As we are interested in points at which the slope significantly changes we further calculate

$$\Delta\alpha_p = \frac{\alpha_p}{\alpha_{p-1}} \quad (16)$$

with $\alpha_0 = \pi/2$.

A value of $\Delta\alpha_p$ greater than 1 indicates that adding an additional feature has a significant smaller negative influence on the cluster coherence that for the proceeding features. A value smaller than 1 indicates that an additional feature has a stronger negative influence. The points between a strong increase and a strong decrease in slope often represent redundant features or very coherent sets of features (vertical parts). Adding an individual feature does not change the performance much. The features between a decrease and an increase represent areas with many noisy and incoherent features (horizontal parts). Adding such features has a direct negative influence on the cluster quality for each of the features that is added. For an example please refer to section 5.3 and figure 4.

5. EVALUATION

In this section we will discuss the results of both the approach proposed in this paper and the normalized minimization approach discussed in section 3. We applied both algorithms to several synthetic and real world data sets.

5.1 The data sets

In order to measure the effect of the artificial normalization factor necessary for feature set minimization we applied the algorithms on a grid data set (GRID) and a random data set (RANDOM) containing only white noise. Another artificial data set (GM) consisting of 32 Gaussian clusters with random standard deviations between 0.0 and 1.0 in five dimensions was created. This data set was enriched with ten additional single Gaussian noise features with standard deviation 0.5.

We also applied both algorithms on two clustering benchmark datasets, namely the IRIS data set [10] and the WPBC (Wisconsin Prognostic Breast Cancer) data set [21]. The latter is especially interesting because of many redundant features. This allows us to check how well both approaches are able to cope with redundancy.

In order to evaluate feature selection for non-standard cluster boundaries we also generated an artificial data set consisting of two spirals and apply single link clustering to it. Such clusterings cannot be found be combinatorial clustering algorithms as k-means. Figure 1 shows the two non-noise dimensions of the created data set.

Table 1 summarizes the properties of all data sets. All experiments were performed with the freely available machine learning environment YALE [9]. The complete data sets, programs, and experiment configurations are available on our web-site[2].

[2]http://www-ai.cs.uni-dortmund.de

5.2 Results

Figure 2 shows all Pareto sets for the simultaneous optimization of the used cluster criterion and the feature set size. It should be noted that in most cases the population converges to the final front after less than 20 generations. Moreover, NSGA-II selection was again able to sustain the found solution until the end of optimization.

It can clearly be seen that in all cases the Pareto sets provided by our information preserving approach contain more points than the results of the normalized minimization approach. If there is only one feature with a relative small standard deviation, the Pareto set of the minimization approach will still collapse (GRID, IRIS-NN, and SPIRAL). Moreover, the normalization factor $1/x$ introduces a convex front although there is nothing to optimize at all. This effect can be seen for the random data set, where the minimization approach finds a convex Pareto front while the front provided by our approach is still linear. For both the normal IRIS data set and the IRIS-GN data set enriched with noise features the proposed approach finds the complete Pareto set while the minimization approach was only able to find a small number of feature subsets. Since it is not clear which clustering is "correct" beforehand the user should be able to select from the complete Pareto front. The same applies for the other real-world data set WPBC (figure 2 shows the results for $K = 2$).

We did not focus on the simultaneous optimization of the number of clusters although this is of course possible. Since our approach does not differ from existing approaches in this respect we concentrated on the usability of information preservation. However, for the real world data set WPBC we simultaneously optimize K in the range of $[2, 10]$ which leads to a three dimensional Pareto set (figure 3). The Pareto plot shows the influence of the number of clusters on the kinks in the Pareto fronts. There are two regions with distinct deviations from the convex hull of the Pareto set. One small kink for $K = 8$ in the rear region and one bigger kink for $k = 2, 3, 4$ in the front region. These kinks were also totally covered by the $1/x$ structure of the normalization approach and it would not be possible to detect a structure at all. Furthermore, redundant features can easily be determined in the almost vertical parts of the front.

5.3 Pareto front segmentation

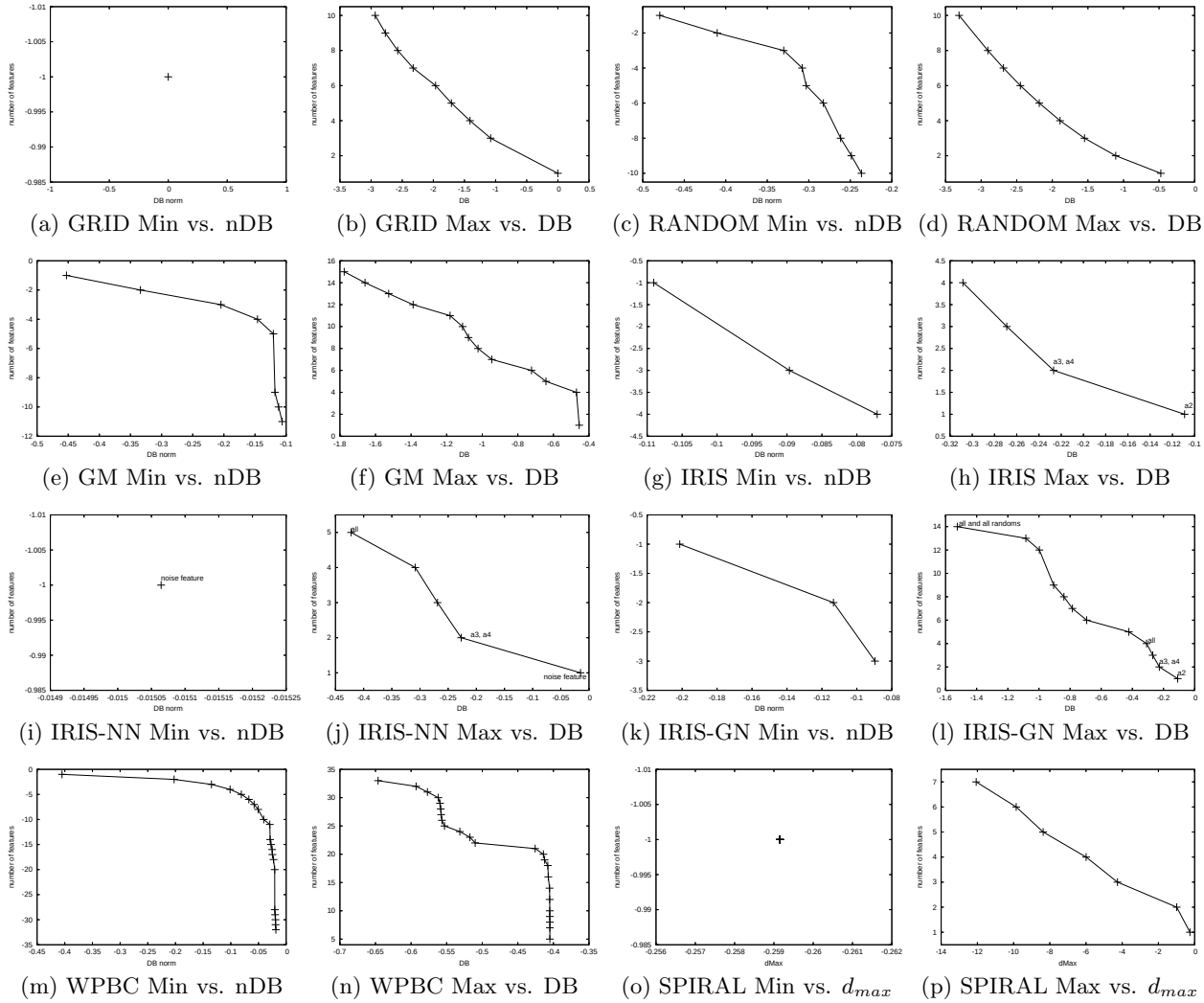
Beside the fact that the Pareto sets are more complete, there is another advantage of the non-normalized maximization: kinks are not covered by the $1/x$ structure. In contrast, these kinks can easily be discovered in the result of our approach. We applied the segmentation algorithm described in section 4.1 on the Pareto set delivered for WPBC and $k = 2$. The kinks between neighbors can clearly be seen if the number of features is maximized (figure 2 (n)) instead of minimized (figure 2 (m)) and the clustering criterion was not normalized. Selecting the five points with highest absolute deviances of $\Delta\alpha_p$ to 1 leads to an interpretable segmentation of the result. Figure 4 shows the selected points and the segments.

6. CONCLUSION

We presented a novel multi-objective evolutionary framework for feature selection in unsupervised machine learning settings. We exemplified this framework on the task of data clustering which plays an important role in a wide variety of

abbr.	properties	N	M	noise	σ_o	σ_n	K	Results
GRID	equidistant values in all dimensions	3125	5	0	n.a.	n.a.	0	(a) and (b)
RANDOM	uniformly distributed values	500	10	10	n.a	∞	0	(c) and (d)
GM	Gaussian mixture with 32 clusters	1000	15	10	0.5	0.5	32	(e) and (f)
IRIS	Iris data set without noise features	150	4	0	0.8	n.a.	3	(g) and (h)
IRIS-NN	Iris data set with nominal noise	150	5	1	0.8	0.01	3	(i) and (j)
IRIS-GN	Iris data set with Gaussian noise	150	14	10	0.8	0.8	3	(k) and (l)
WPBC	WPBC data set without noise features	198	34	0	33.2	n.a.	?	(m) and (n)
SPIRAL	Two spirals around the origin	500	7	5	5.5	5.5	2	(o) and (p)

Table 1: The used data sets. The first column summarizes the abbreviations used in the text, the second summarizes some properties of the data set. N is the total number of examples, M the total number of features. The column *noise* defines how many features of M where explicitly added noise features. The next columns define the mean standard deviation of the original features (σ_o) and the noise features (σ_n). The column K indicates the number of clusters if it is known. The last column indicates which Pareto sets were found for the data set with both approaches.



(a) GRID Min vs. nDB (b) GRID Max vs. DB (c) RANDOM Min vs. nDB (d) RANDOM Max vs. DB

(e) GM Min vs. nDB (f) GM Max vs. DB (g) IRIS Min vs. nDB (h) IRIS Max vs. DB

(i) IRIS-NN Min vs. nDB (j) IRIS-NN Max vs. DB (k) IRIS-GN Min vs. nDB (l) IRIS-GN Max vs. DB

(m) WPBC Min vs. nDB (n) WPBC Max vs. DB (o) SPIRAL Min vs. d_{max} (p) SPIRAL Max vs. d_{max}

Figure 2: The Pareto fronts for all data sets. The left result for each dataset is achieved by the approach discussed in section 3 for a normalized value nDB (DB_{norm}). It can clearly be seen that these results are not as complete and that kinks are covered by the artificial $1/x$ structure. The results on the right are achieved by our information preserving maximization approach.

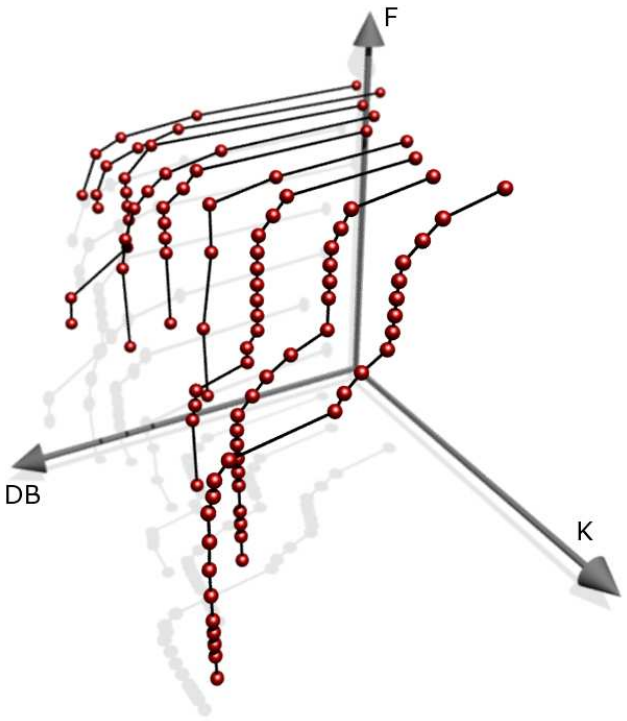


Figure 3: We applied information preserving feature selection on the real-world data set WPBC. The number of features (F), the Davies Bouldin clustering criterion (DB), and the number of clusters (K) should be simultaneously optimized. The result is a three dimensional Pareto set containing all necessary information allowing a decision about the best clustering. The kinks could be used to segment the Pareto set and ease the analysis of the front.

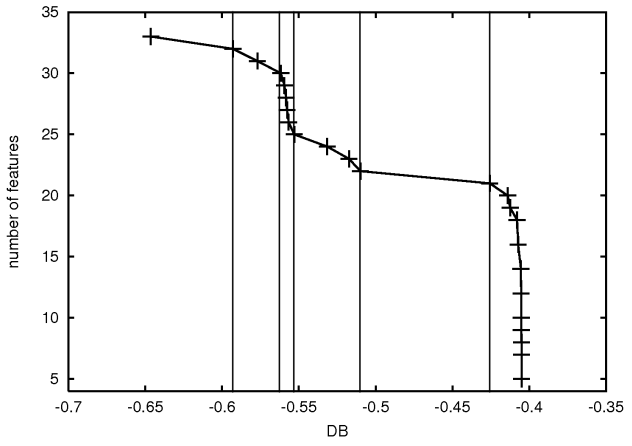


Figure 4: The five points with the highest absolute deviation of $\Delta\alpha_p$ to 1 are marked with perpendicular lines. This leads to an interpretable segmentation of the Pareto front which eases the process of selecting a final solution from the Pareto set.

applications ranging from pattern recognition to customer relationship management and web search. Clustering is an ideal test case for evolutionary computation methods for several reasons. First, it is an inherently multi-objective problem. There is usually not one correct result as for supervised learning. Users rather explore the space of results interactively to gain insight into the natural patterns within the data set. Second, the approach proposed in this work yields Pareto sets that show significant inner structure. This structure is not accidental but reflects patterns in the underlying data. We presented a generic method for an automatic Pareto set segmentation and showed that the discovered segments can be interpreted with respect to unsupervised feature selection. This turns clustering into a reference application of automatic Pareto set analysis. We argued that these benefits can only be achieved if the optimization problem has been posed in a sound way. Although maximizing the number of features during feature selection might sound surprising at first, this paradigm change can be motivated by the aim of unsupervised learning: the search for descriptive, natural patterns. In particular, we have shown that existing approaches to multi-objective unsupervised feature selection based on minimization are not appropriate as they produce trivial or incomplete solution sets. Another contribution of our approach is its applicability to other clustering algorithms than EM or k-means. This is essential in applications in which clusters of any shape must be found, e. g. by means of density based clustering algorithms.

In our future work we plan to incorporate additional state of the art clustering algorithms, as well as other unsupervised learning techniques as association rule learning. We also plan to explore possibilities to use evolutionary multi-objective optimization for the problem of unsupervised feature *construction* as well. Although our approach has the same runtime as existing approaches, large-scale unsupervised feature selection might not be feasible. The impact of sampling should be analyzed in order to reduce the total runtime without loosing information about the data set. In our opinion, evolutionary computation is a very promising solution to overcome essential limitations of current unsupervised machine learning approaches.

7. ACKNOWLEDGMENTS

This work was supported by the *Deutsche Forschungsgemeinschaft (DFG)* within the *Collaborative Research Center "Reduction of Complexity for Multivariate Data Structures"*.

8. REFERENCES

[1] A. Ben-Hur, D. Horn, H. T. Siegelmann, and V. Vapnik. Support vector clustering. *Journal of Machine Learning Research*, 2001.

[2] C. A. Coello Coello. A comprehensive survey of evolutionary-based multiobjective optimization techniques. *Knowledge and Information Systems*, 1(3):129–156, 1999.

[3] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press, 2001.

[4] D. Davies and D. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(2):224–227, 1979.

[5] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multi-objective genetic algorithm: Nsga-ii. Technical report, Kanpur Genetic Algorithms Laboratory, Indian Institute of Technology, 2002.

[6] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

[7] C. Emmanouilidis, A. Hunter, and J. MacIntyre. A multiobjective evolutionary setting for feature selection and a commonality-based crossover operator. In *Proc. of the Congress on Evolutionary Computation (CEC)*, pages 309–316, 2000.

[8] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. of the International Conference on Knowledge Discorvery in Databases (KDD)*, pages 226–231, 1996.

[9] S. Fischer, R. Klinkenberg, I. Mierswa, and O. Ritthoff. Yale: Yet Another Learning Environment – Tutorial. Technical Report CI-136/02, Collaborative Research Center 531, University of Dortmund, Dortmund, Germany, 2002.

[10] R. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936.

[11] J. A. Hartigan and M. A. Wong. A k-means clustering algorithm. *Applied Statistics*, 28(1):100–108, 1979.

[12] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer series in statistics. Springer, 2001.

[13] A. Jain, M. Murty, and P. Flynn. Data clustering: a review. *ACM Computing Surveys*, 3(31):264–323, 1999.

[14] Y. Kim, W. Street, and F. Menczer. Feature selection in unsupervised learning via evolutionary search. In *Proc. of the 6th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 365–369, New York, NY, USA, 2000. ACM Press.

[15] Y. Kim, W. N. Street, and F. Menczer. Evolutionary model selection in unsupervised learning. *Intelligent Data Analysis*, 6:531–556, 2002.

[16] M. Morita, R. Sabourin, F. Bortolozzi, and C. Suen. Unsupervised feature selection using multi-objective genetic algorithms for handwritten word recognition. In *Proc. of the 7th International Conference on Document Analysis and Recognition (ICDAR)*, 2003.

[17] F. Murtagh. *Clustering in massive data sets*, pages 501–543. Kluwer Academic Publishers, 2002.

[18] M. Raymer, W. Punch, E. Goodman, L. Kuhn, and A. Jain. Dimensionality reduction using genetic algorithms. *IEEE Transactions on Evolutionary Computation*, 4, 2000.

[19] V. Roth and T. Lange. Feature selection in clustering problems. In *Proc. of Neural Information Processing Systems (NIPS)*, 2003.

[20] J. Sander, M. Ester, H.-P. Kriegel, and X. Xu. Density-based clustering in spatial databases: The algorithm gdbscan and its applications. *Data Minining and Knowledge Discovery*, 2(2):169–194, 1998.

[21] W. Wolberg, W. Street, D. Heisey, and O. Mangasarian. Computer-derived nuclear "grade" and breast cancer prognosis. *Analytical and Quantitative Cytology and Histology*, 17:257–264, 1995.

[22] J. Yang and V. Honavar. Feature subset selection using a genetic algorithm. *IEEE Intelligent Systems*, 13:44–49, 1998.

[23] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.