Comparing the Effectiveness of Support Vector Machines with Fuzzy, Neuro-fuzzy and Genetic Programming Approaches in Result Prediction of Football Games

A. Tsakonas[1], G. Dounias[1], S. Shtovba[2] and V. Vivdyuk[2]

[1]University of the Aegean (Greece),
[2]Vinnitsa State Technical University (Ukraine)

Abstract
A soft computing method for result prediction of football games based on machine learning techniques such as support vector machines is proposed in the article. The model is taking into account the following features of football teams: difference of infirmity factors; difference of dynamics profile; difference of ranks; host factor; personal score of the teams. Testing shows that the proposed model achieves a satisfactory estimation of the actual game outcomes. The current work concludes with the recommendation of support vector machines technique as a powerful approach, for the creation of result prediction models of diverse sport championships.

1. Introduction

The prediction of sport game results consists an important task in bookmaking business. Besides, this task can perform as a good benchmark problem for testing diverse techniques of extrapolation and prediction under difficult conditions of limited available statistics and uncertainties of influence factors. Soft computing [1] is meant as a large variety of new powerful techniques for intelligent data analysis, which provide a suitable way for handling complexity, uncertainty and fuzziness of real-world problems. The aim of the present paper is to demonstrate an example of how to predict football game winners by applying soft-computing techniques, such as support vector machines. Data representing the Ukrainian football championship during 10 last years are used for the creation and testing of the intelligent prognostic models applied within this paper.

2. The problem statement

From the cybernetic point of view, the task of creating football winner prediction models is reduced to that of finding out functional mapping of the form:

$$X = \{x_1, x_2, \ldots, x_n\} \to D \in \{d_1, d_2, d_3\}, \qquad (1)$$

where, X: denotes a vector of features (i.e. influence factors), such as team level, climate conditions, playing place, results of past games etc.;

D: denotes the football game result for assessment of one of the terms:

d_1 : «host team's win», d_2 : «draw» and d_3 : «guest team's win».

3. Feature selection

From the authors' point of view, the features carrying the major influence on the game results are:

x_1 : difference of infirmity factors (as number of traumatized and disqualified players of host team, minus the same players of guest team);

x_2 : difference of dynamics profile (as score of host team for five last games minus score of guest team for the five last games);

x_3 : difference of ranks (host team's rank, minus guest team's rank, in the current championship);

x_4 : host factor (as *HP/HG - GP/GG*, where *HP* denotes the total home points of the host team in the current championship; *HG* is the number of played home games by the host team; *GP* is the total guest points of the guest team in the current championship; *HG* is the number of played guest games by the guest team);

x_5 : personal score (as goal difference for all the games of the teams involved, within 10 years).

Note, that the above features do not consist confidential information, but it is easy for the decision maker to know the feature values before the game.

4. Support Vector Machines model

Support Vector Machines (SVM) [2] are a relatively new computational intelligence technique, related to the machine learning concept. SVMs are used in pattern recognition as well as in regression estimation and linear operator inversion. SVMs have interesting attributes, different than other CI techniques, such as neural networks, as SVMs find always a global minimum and they have a simple geometric interpretation. SVMs are also capable of handling large number of data or attributes and their learning is comparable in terms of speed with that of neural networks. The main characteristic of a SVM is its kernel, and the selection of the best kernel for a given problem is still a research issue. More specifically, in order to estimate a classification function $f: x \rightarrow \{\pm 1\}$, the most important is to select an estimate f from a well restricted so-called *capacity* of the learning machine. Small capacities may not be sufficient to approximate complex functions, while large capacities may fail to generalize, which is the effect of overfitting.

In contrast to the neural networks' approach, where early stopping is used to avoid overfitting, in SVMs, is limited according to the statistical theory of learning from small samples [3]. The simpler decision functions are the linear functions. In the case of SVM, the implementation of linear functions corresponds to finding a large margin separating between two classes. This margin is the minimum distance of the training data points to the separation surface. The procedure to find the maximum margin separation is a convex quadratic problem (QP) [4]. An additional parameter enables the SVM to misclassify some outlying training data in order to get larger margin between the rest training data, without however affecting the optimization be the QP. If we transform the input data into a feature space F using a map $\Phi : x \rightarrow$ F, a linear learning machine is extended to a non-linear one.

In SVMs the latter procedure is applied implicitly. What we have to supply is a dot product of pairs of data points $\Phi(X)$, $\Phi(y) \in$ F in feature space. Thus, to compute these dot products, we supply the so-called *kernel* functions which define the feature space via: $k(x,y)=(\Phi(x)*\Phi(y))$. We don't need to know Φ, because the mapping is performed implicitly. SVMs can also learn which of the features implied by the kernel are distinctive for the two classes. The selection of the appropriate kernel function may boost the learning process. Although our problem is actually a multi-class classification (predict the winner with three possible outcomes: home, host, draw) little research or none has been done in the one-step multi-class [5]. Thus we solve this classification problem as a common regression problem, where the SVM algorithm has to minimize the mean square error. Then, in order to get the predicted outcome, the following rules are applied to the de-normalized forecasted values:

- If *forecasted_value*>=0 consider positive or zero score result (host team will not win)
- If *forecasted_value*<0 consider negative or zero score result (home team will not win)

While SVM classification must be applied between two classes, we select to ignore the draw case as a special case (a no winner case) keeping the sign of the output indicating the predicted class. The algorithm was fed with 105 training data records and the SVM was tested on 70 test data records. All data were normalized in [-1,1] range. We selected as kernel function the dot function (simple multiplication) as we had no evidence for the appropriateness of other, more complex functions. We also set C=1000 and *epsilon*=0.01. The reader can find detailed analysis on the factors presented below in [Burges 1998]. The following results were obtained after 1377 iterations:

Train Set Mean square error: 0.052297589

Test Set Mean square error: 0.053676842

Table 1 - Support Vector Machine attributes

Support Vectors	97
Bounded SVs	90
Minimum and maximum value of the alphas	min SV: -9.7087379 max SV: 9.7087379
2-norm of the hyperplane vector,	|w| =0.12128035
Estimation (by the two last examples) of VCdim	VCdim <= 1.3774434
Hyperplane vectors for the attributes	
w[0] = 0.2527201	
w[1] = -0.010411425	
w[2] = 0.28175218	
w[3] = 0.18387293	
w[4] = 0.099184523	
b	0.06384628

By applying the classification rules described in the previous paragraph we received the following results:

Correct Prediction on Test Set: 43 out of 70 examples (accuracy 61.4%). Table 1 presents the model attributes.
In order to compare our model with other approaches, we considered results obtained by other computational intelligent approaches, in previous work [6]. Those results were obtained for a prediction including the draw result of the matches, thus their quotation is here indicatory. Also, results for the fuzzy model and the neural network include the classification score on an 175-element set (training and testing sets). These results can help however to draw general conclusions on the effectiveness of the method in this data set.

Table 2 – Comparison of the model with other approaches

Model	Correct classification
Fuzzy model	64 % (both sets)
Neural network	64 % (both sets)
Genetic programming model	64.28 % (test set)
Support Vector Machines	61.4% (test set)

5. Conclusions - Further Research

This paper demonstrated the application of statistical or entropy-based approaches, such as support vector machines (SVM). The latter, relatively new computational intelligence approache, was implemented in a common for SVM "±1" outcome basis, with positive values corresponding to a host-team-will-not-win outcome and negative values to a home-team-will-not-win outcome. Results denote the competitiveness of this approach. Further research in this domain, may involve hybrid computational intelligent schemes while those systems have been proved in many cases capable of capturing nearly stochastic or chaotic processes offering a high classification and prediction rate.

References

1. Zadeh L. Applied Soft Computing –Foreword // *Applied Soft Computing*, vol.1 (2000).
2. Vapnik V.N, Statistical learning theory, Wiley (1998).
3. Vapnik V.N., The nature of statistical learning theory, Springer-Verlag (1995).
4. Boser B., Guyon I. and Vapnik V.P., A training algorithm for optimal margin classifiers // *COLT*, pp 144-52 (1992).
5. Burges C., A Tutorial on Support Vector Machines for Pattern Recognition, Kluwer
6. Tsakonas A.,Dounias G., Shtovba S., Vivdyuk V., Soft Computing-Based Result Prediction of Football Games // *Proc. ICIM-2002*, Lviv, pp 15-21 (2002)