A Method to Improve Generalization of Neural Networks: Application to the Problem of Bankruptcy Prediction

Armando Vieira[1], João C. Neves [2], Bernardete Ribeiro[3]

[1]ISEP and Centro de Física Computacional, University of Coimbra, 3000 Coimbra, Portugal, asv@isep.ipp.pt
[2] ISEG, Rua Miguel Lupi 20, 1200 Lisboa, Portugal
[3]Department of Informatics Engineering, University of Coimbra, Portugal

Abstract

The Hidden Layer Learning Vector Quantization is used to correct the prediction of multilayer perceptrons in classification of high-dimensional data. Corrections are significant for problems with insufficient training data to constrain learning. Our method, HLVQ-C, allows the inclusion of a large number of attributes without compromising the generalization capabilities of the network. The method is applied to the problem of bankruptcy prediction with excellent results.

1 Introduction

Many actual machine learning problems, like image analysis, neurophysiology, remote detection and bioinformatics, involve classification and clustering of high-dimensional data. However, most classification methods have a poor performance when a large number of features are used. Since the search space increase exponentially with the number of features included, large training sets are usually required to obtain a reliable model. If the learning machine is an Artificial Neural Network, complex architectures may be needed with the risk of overfitting. Moreover, training is slower, due to the presence of many local minima, and often becomes an ill-conditioned problem [1].

Some classifiers, such as minimum distance, often perform relatively well on low dimensional data, but they show limited performance in high dimensional spaces. Dimensionality reduction through feature selection is a useful, and often necessary, preprocessing step. For difficult classification problems, with non-linear correlated inputs or samples containing many atypical examples, care must be taken during feature selection not to discharge information necessary for class separation.

In a previous work we showed that Hidden Layer Learning Vector Quantization (HLVQ) is an efficient method for classification [2]. HLVQ performs a non-linear feature selection that effectively reduce the dimensionality of the problem and improve class discrimination. In this work we apply HLVQ to correct the predictions of multi layer perceptron (MLP) in classification problems. The objective is to correct the outputs of the MLP, especially for points in poorly sampled regions in the training process. With this method we can handle more features without compromising the generalization capabilities of the MLP.

Next section presents the HLVQ algorithm and in section 3 it is used to implement the purposed correction algorithm. Section 4 presents brief description of Support Vector Machines (SVM), and section 5 describes the bankruptcy prediction problem. Section 6 presents the results of our method in comparison to the most conventional method of discrimination, the Multiple Discriminant Analysis (MDA) and more recent techniques such as Support Vector Machines.

2 The Hidden Layer Learning Vector Quantization (HLVQ)

The Hidden Layer Learning Vector Quantization (HLVQ) is an algorithm recently proposed for classification of high dimensional data [2, 3]. It is implemented in three steps. First, a multilayer perceptron is trained using back-propagation. Second, supervised Learning Vector Quantization is applied to the outputs of the last hidden layer to obtain the code-vectors $\vec{w}_{ci}$ corresponding to each class c_i in which data are to be classified. Each example, $\vec{x}_i$, is assigned to the class c_k having the smallest Euclidian distance to the respective code-vector:

$$k = \min_{j} \left\| \vec{w}_{c_j} - \vec{h}(\vec{x}) \right\| \qquad (1)$$

where $\vec{h}$ is a vector containing the outputs of the hidden layer and $\|\cdot\|$ denotes the usual Euclidian distance. In the third step the MLP is retrained but with two differences regarding conventional multilayer training. First the error correction is not applied to the output layer but directly to the last hidden layer. The output layer is therefore ignored from now on. The second difference is in the error correction backpropagated to each hidden node:

$$E_1 = \frac{1}{2} \sum_{i=1}^{N_h} \left(\vec{w}_{ck} - \vec{h}(\vec{x}_i) \right)^2 \qquad (2)$$

After retraining the MLP a new set of code-vectors,

$$\vec{w}_{c_i}^{new} = \vec{w}_{ci} + \Delta\vec{w}_{ci} \tag{3}$$

is obtained according to the following training scheme:

$$\begin{aligned} \Delta\vec{w}_{ci} &= \alpha(n)(\vec{x} - \vec{w}_{c_i}) \text{ if } \vec{x} \in \text{ class } c_i, \\ \Delta\vec{w}_{c_i} &= 0 \qquad\qquad \text{if } \vec{x} \notin \text{class } c_i \end{aligned} \tag{4}$$

The parameter α is the learning rate, which should decrease with iteration n to guarantee convergence. Steps two and three are repeated following an iterative process. The stopping criterium is met when a minimum classification error is found.

The distance of given example x to each prototype is:

$$d_i = \left\|\vec{h}(\vec{x}) - \vec{w}_{c_i}\right\| \tag{5}$$

which is a proximity measure to each class.

3 HLVQ-C

One drawback of multilayer perceptrons is their poor performance in sparse regions not covered by training data, common in high dimensional datasets. To alleviate this situation we propose the following method to correct the MLP output. The objective is to identify these regions and evaluate the consistency of the perceptron results against the HLVQ prediction. After training the MLP and HLVQ, the algorithm runs according to the following steps.

Each test example, $\vec{x}^i$, is included in the training set and the neural network retrained. Since the class membership of this example is unknown, we first assign it to class "0" and determine the corresponding output $y_0(\vec{x}^i) = y_0^i$ as well as the respective distances to each class prototype obtained with HLVQ,

$$\begin{aligned} &\vec{d}_0^i = (d_0^{c0}, d_0^{c1}) = \\ &\left(\left\|h_0(\vec{x}^i) - w_{c0}\right\|, \left\|h_0(\vec{x}^i) - w_{c1}\right\|\right) \end{aligned} \tag{6}$$

In a second step the network is retrained considering the example as class "1". The new output $y_1(\vec{x}^i) = y_1^i$ and the respective distances to the prototypes are obtained in a similar way, thus:

$$\begin{aligned} &\vec{d}_1^i = (d_1^{c0}, d_1^{c1}) = \\ &\left(\left\|h_1(\vec{x}^i) - w_{c0}\right\|, \left\|h_1(\vec{x}^i) - w_{c1}\right\|\right) \end{aligned} \tag{7}$$

From these outputs, y_0^i and y_1^i, the choice is made following the heuristic rule:

$$\begin{aligned} y^i &= y_0^i \text{ if } d_0^{c0} < d_0^{c1} \\ y^i &= y_1^i \text{ if } d_1^{c1} < d_1^{c0}. \end{aligned}$$

We call this method Hidden Layer Learning Vector Quantization Correction, HLVQ-C, and it corrects the class separation hyperplane.

Figure 1, show how the method works. Before correction, a new point "?", located outside the region covered by the training data, is classified as class 1 by the MLP, since it is above the separation hyperplane. However, the distance d_1 to class 0 code-vector (X) is smaller than the distance to the class 1 code-vector (Y) - d_2. HLVQ-C modifies the MLP prediction and this element is now correctly assigned to class 0.

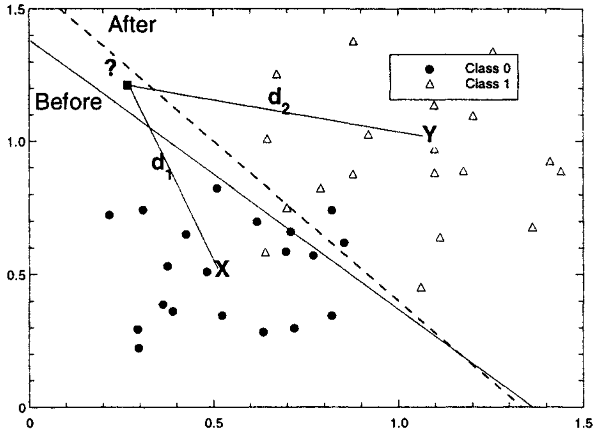

Figure 1: How HLVQ-C corrects the MLP predictions. Full line indicate the separation hyperplane before a new point "?" is included. Dashed line is the same but after retraining considering the new point as class 0.

4 Support Vector Machines (SVM)

Support vector machines (SVM) are a new learning-by-example paradigm spanning a broad range of classification, regression and density estimation problems [15]. Equipped with a sound mathematical background, SVM treat both the problem of complexit minimization while maximizing generalization. The advantage, regarding other approaches, is that SVM generalization error is not dependent on the dimensionality of the input data. The learning method uses input-output training examples from the data set

$$D = \{(\mathbf{x}_i, y_i) \in X \subseteq R^N \times Y \ : 1 \le i \le l\} \tag{8}$$

such that f classifies correctly test data (x,y) generated from the same underlying probability distribution $P(x,y)$. Using the loss function defined by :

$$\mathrm{V}(\mathrm{y}_i, \mathrm{f}(\mathbf{x}_i)) = |1 - \mathrm{y}_i \mathrm{f}(\mathbf{x}_i)| + \tag{9}$$

The learning problem can be formulated minimizing the function (10) using

$$\frac{1}{\mathrm{l}} \sum_{\mathrm{i}=1}^{\mathrm{l}} \mathrm{V}(\mathrm{y}_i, \mathrm{f}(\mathbf{x}_i)) + \|\mathrm{f}\|_{\mathrm{F}_k}^2 \tag{10}$$

The minimizer of (10) has the form:

$$f(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i k(\mathbf{x}, \mathbf{x}_i) + b \tag{11}$$

with $\alpha_i, b \in \boldsymbol{R}$. The equivalent quadratic programming problem originally proposed in [16] is:

$$\min_{f \in \mathbf{F}, \xi} \Phi(f, \xi) = \frac{C}{l} \sum_{i=1}^{l} \xi_i + \frac{1}{2} \|f\|_k^2 \tag{12}$$

subject to constraints:

$$\begin{aligned} y_i f(\mathbf{x}_i) &\geq 1 - \xi_i \quad i = 1, \cdots, l \\ \xi_i &\geq 0 \qquad\quad i = 1, \cdots, l \end{aligned} \tag{13}$$

where C is the penalty constant (regularization parameter) and ξ the slack variable.

Introducing Lagrange multipliers:

$$\max_{\boldsymbol{\alpha}} W(\boldsymbol{\alpha}) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \tag{14}$$

and solving with respect to α_i, under the constraints $0 \leq \alpha_i \leq \frac{C}{l}$, $i = 1 \ldots, l$ and $\sum_{i=1}^{l} \alpha_i y_i = 0$, the solution has again the form of Eq. (11). The empirical error measured by $\sum_{i=1}^{l} \xi_i$ is minimized while controlling the learning machine capacity.

5 Application to the Bankruptcy Problem

The bankruptcy prediction problem is to discriminate between healthy and distressed companies based on the record of several financial indicators from previous years [4]. Several modern approaches have been used to analyze this problem, ranging from Support Vector Machines, Genetic Algorithms and Neural Networks [4-10]. Although several authors have demonstrated that these methods are in general more accurate in signaling a distressed company, traditional Linear Discriminant Analysis (MDA) is still widely used. The success of MDA can be explained by its simplicity and the fact that, in most cases, the difference in performance are not substantial. Another reason to explain the analyst's resilience in using new approaches, concerns the quality of databases used to benchmark the predictions. The most common weakness are the small size of the database, limited historical records and the use of unbalanced samples, containing much more healthy companies than financially distressed ones.

We used a sample obtained from Diana, a database containing about 780,000 financial statements of French companies. The initial sample consisted of financial ratios on 2,800 non-financial French companies, for the years of 1998, 1999 and 2000, with at least 35 employees. From these companies, 311 were declared bankrupted in 2000 and 272 presented a restructuring plan ("Plan de redressement") to the court for creditors approval. We decided not to distinguish these two categories as both signal companies in financial distress.

The sample thus has 583 financial distressed firms, most of them of small and medium size, with a number of employees from 35 to 400. From the initial 30 financial ratios we select the 17 most relevant ratios excluding those with either a small average sensitivity or a small variance of the sensitivity. High correlated ratios were also excluded. All data were normalized to zero mean and unity variance.

Neural networks containing from 5 to 20 hidden nodes were tested. A hidden layer of 15 neurons, a learning rate of 0.1 and a momentum term of 0.25 were chosen.

The bankruptcy is a dynamic process. In order to capture the evolution and the trend of the financial position we also used three annual variances of the following ratios: debt ratio, value added per employees and margin before extra items and taxes, thus extending our previous study [3].

6. Results and conclusions

To test HLVQ-C, we used a balanced dataset containing 583 healthy and 583 distressed companies using ten-fold cross validation. We discriminate the accuracy of the classifiers for type I, type II error and the overall misclassification. Type I error is the percentage of undetected bankruptcies while type II error is the percentage of healthy companies predicted as bankrupt.

Table 1 summarizes the results with data from 1999, one-year prior to the announcement of bankruptcy. Our method surpasses all others, both in the overall accuracy, and, more important, on type I error. This term has a much higher cost for banks and insurance companies than type II error [4].

Support Vector Machines (SVM) reduced substantially error type I while keeping error type II at almost the same level as MLP. This may be due to the fact that the optimal compromise between minimization of the empirical risk and complexity has not been reached. However, the method could be improved using unlabeled data and the geometry of the separation hyperplane would allow less misclassifications of the type I.

Type I error is always higher than type II error since distressed companies has a more heterogeneous pattern and are therefore harder to classify. Finally, note that Multiple Discriminant Analysis (MDA) has a very small accuracy.

Table 1: Generalization errors (in percentage) for several machine learning algorithms.

Model	Error I	Error II	Total
MDA	26.4	21.0	23.7
SVM	17.6	12.2	14.8
MLP	25.7	13.1	19.4
HLVQ-C	11.1	10.6	10.8

We presented a technique for output correction of multilayer perceptrons, called HLVQ-C. With this technique a MLP can be trained with a large set of features without compromising the generalization. Corrections introduced by HLVQ-C can be substantial for high-dimensional data and small training datasets.

To our knowledge, results obtained with Hidden Layer Learning Vector Quantization Correction, represent the greatest improvement with respect to discriminant analysis. Support Vector Machines is also a very competitive approach.

References

[1] Bishop C. M. (1996) Neural Networks for Pattern Recognition, Oxford University Press, Oxford.

[2] Vieira A. and Barradas N. P. (2003) A training algorithm for classification of high dimensional data, Neurocomputing, 50C, 461-472.

[3] Vieira A. , Ribeiro B., Mukkamala S., Neves J. C. and Sung A. H. (2004) On the Performance of Learning Machines for Bankruptcy Detection, Second IEEE Int. Conf. on Computational Cybernetics, Vienna, Austria, August 30 – September 1, 223 – 227.

[4] Altman, E.I. (1989) Measuring Corporate Bond Mortality and Performance. Journal of Finance, **44**, 909-1022.

[5] G. Zhang, M. Y. Hu, B. E. Patuwo, D. C. Indro (1999) Artificial neural networks in bankruptcy prediction: General framework and cross-validation analysis, Europ. J. Op. Research. **116**, 16.

[6] Coats P.K., Fant L.F. (1993) Recognising Financial Distress Patterns Using a Neural Network Tool. Financial Management, 142-155.

[7] Jain, B., Nag B. (1997) Performance Evaluation of Neural Network Decision Models. Journal of Management Information Systems **14** (2), 201-216.

[8] Lee K.C., Han I., Known Y. (1996) Hybrid neural network models for bankruptcy predictions, Decision Support Systems 18, 63-72.

[9] Shah J.R., Murtaza, M. B. (2000) A Neural Network Based Clustering Procedure for Bankruptcy Prediction, American Business Review (June), 80-86.

[10] Varetto F. (1998) Genetic Algorithms Applications in the Analysis of Insolvency Risk, Journal of Banking and Finance, **22**, 1421-1439.

[11] Vieira A., P.A. Castillo and J.J. Merelo (2003) Comparison of HLVQ and GProp in the problem of bankruptcy prediction, IWANN03 - International Workshop on Artificial Neural Networks, LNCS 2687, Springer-Verlag, 655-662.

[12] J. S. Grice, M. T. Dugan (2001) The Limitations of Bankruptcy Prediction Models: Some Cautions for the Researcher Rev. of Quant. Finance and Account., **17** (2), 151.

[13] A. F. Atiya (2001) Bankruptcy Prediction for Credit Risk Using Neural Networks: A Survey and New Results, IEEE Trans. Neural. Net. 12 (4).

[15] C. Cortes and V. Vapnik, (1995) Support vector networks, Machine Learning, vol. 20, pp.273-297.

[16] V. Vapnik, (1995) The Nature of Statistical Learning Theory. New York, Springer.