Available online at www.sciencedirect.com
SCIENCE DIRECT®
Pattern Recognition 39 (2006) 313–315
PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/patcog

Rapid and brief communication

Eliminating redundancy and irrelevance using a new MLP-based feature selection method

E. Gasca[a], J.S. Sánchez[b,*], R. Alonso[a]

[a] *Lab. Reconocimiento de Patrones, Instituto Tecnológico de Toluca, Av. Tecnológico s/n, 52140 Metepec, Edomex, México*
[b] *Dept. Llenguatges i Sistemes Informàtics, Universitat Jaume I, Av. Sos Baynat s/n, 12071 Castelló de la Plana, Spain*

Received 14 June 2005

Abstract

This paper presents a novel feature selection method based on the use of a multilayer perceptron (MLP). The algorithm identifies a subset of relevant, non-redundant attributes for supervised pattern classification by estimating the relative contribution of the input units (those representing the attributes) to the output neurons (those corresponding to the problem classes). The experimental results suggest that the proposed method works well on a variety of real-world domains.

Keywords: Feature selection; Multilayer perceptron; Relative contribution

1. Introduction

Supervised learning algorithms employ a collection of instances (or training set) to estimate the class label of new input samples. The instances are generally represented by a number of attributes (or features) and a class value. However, not all of the attributes result equally important for a specific task. In fact, some features can be redundant or irrelevant. By removing such redundant and irrelevant attributes, a classifier with higher predictive accuracy can often be obtained. On the other hand, as the number of features grows, the number of training instances needed grows exponentially and therefore, in many practical situations it is necessary to reduce the dimensionality of the data.

In general, techniques for dimensionality reduction focus either on selecting a proper subset from the original set of I attributes, or on mapping the initial I-dimensional data onto the K-dimensional space, where $K < I$. While the former corresponds to *feature selection*, the latter represents *feature extraction*. Many feature selection and extraction techniques can be found in the literature.

Neural networks have been widely applied to a huge variety of supervised pattern classification problems. During the last decade, neural networks have also been successfully employed for feature selection and extraction [1,2,4,5]. The present paper proposes a new feature selection strategy based on using neural networks. More specifically, a three-layer multilayer perceptron (MLP) trained by the backpropagation algorithm is here used as the tool to determine which attributes are to be removed from the original set.

2. A new MLP-based feature selection algorithm

The problem of feature selection involves finding an "appropriate" set of attributes under a certain objective function. The search strategy employed in our algorithm consists of computing the *relative contribution* of each attribute to the output units. To this end, we measure the contribution of input element i to output element k, taking into account the contribution of the hidden layer elements ($j = 1, \ldots, J$), as follows [3]:

$$C_{ik} = \frac{\sum_{j=1}^{J} \omega_{jk}\beta_j\omega_{ji}}{\sum_{i=1}^{I}\left|\sum_{j=1}^{J} \omega_{jk}\beta_j\omega_{ji}\right|}, \quad (1)$$

* Corresponding author. Tel.: +34 964 728350; fax: +34 964 728435.
E-mail address: sanchez@uji.es (J.S. Sánchez).

0031-3203/$30.00
doi:10.1016/j.patcog.2005.09.002

where I denotes the number of input units. The ω_{ji} and ω_{jk} are the connection strengths between the input (i) and hidden (j) layers, and between the hidden and output (k) layers, respectively. The approximation to the contribution of hidden layers β_j can be estimated in terms of the value of the jth hidden layer output for the tth training instance, say O_{jt}. To obtain β_j, we will average $O_{jt}(1-O_{jt})$ across the T training patterns: $\beta_j = (1/T)\sum_{t=1}^{T} O_{jt}(1-O_{jt})$.

As objective function for the feature selection algorithm, we use a concept called *prominence*, which indicates the real relevance of an attribute for a given classification task. More specifically, we determine the order of feature relevance by averaging the frequency of the contributions. The algorithm can be written as follows:

1. Estimate the contribution of each attribute i to each output unit k.
2. For each output unit, sort the I contributions in descending order.
3. For each output node, compute how many times (N_{il}) the attribute i is in the lth position.
4. Compute the prominence P_i of the attribute i as $P_i = \sum_{l=1}^{I} \frac{1}{2^l} N_{il}$.
5. Sort the attributes according to their values of prominence in descending order.

Once the attributes have been sorted according to their relevance, we are to identify attributes with similar information, that is, we search for redundant features. To this end, we compute the correlation index between pairs of attributes (r_{ab}): two attributes are said to be correlated if $|r_{ab}| > 0.707$.

$$r_{ab} = \frac{c_{ab}}{\sqrt{c_{aa}c_{bb}}}, \tag{2}$$

where $c_{ab} = (1/T)\sum_{t=1}^{T} \left(x_a^t - \mu_a\right)\left(x_b^t - \mu_b\right)$ is the correlation between attributes a and b. x_a^t and x_b^t are the values of the ath and bth attributes for the tth instance, and μ_a and μ_b denote their expected values.

If two attributes are correlated ($|r_{ab}| > 0.707$), we propose two alternatives for deciding which feature (a or b) has to be removed:

Opt-I: Discard the attribute with the lowest prominence value.

Opt-II: Discard the attribute with the lowest prominence value, provided that it is among the $I/2$ top ranked features, that is, there are at least $I/2$ features occupying a better (lower) position within the relevance list generated in Step 5 of the above algorithm.

3. Experimental results and discussion

In our experiments, we have included a total of nine benchmark databases taken from the UCI Machine Learning Database Repository (http://www.ics.uci.edu/~mlearn). The main characteristics of these data sets have been summarized in Table 1. The 10-fold cross-validation error estimate method has been employed for each database.

The experiments have been carried out with a three-layer MLP. Only one hidden layer has been used and the number of hidden units is set to $(I + 1)$, where I is the number of features. The network has been trained by using the backpropagation algorithm with a sigmoidal activation function for both the hidden and output layers. The backpropagation algorithm is simple and easy to compute. It often converges rapidly to a local minimum, but it may not find a global minimum and in some cases, it may not converge at all. To overcome this problem, a momentum term can be added to the minimization function, and a variable learning rate can be applied. In our experiments, the learning rate and momentum are 0.9 and 0.7, respectively, and the number of training iterations is 30,000.

The contribution of the attributes on the output neurons has been measured with 60% of the available patterns. Afterwards, the feature selection methods proposed in this paper have been applied to each of the 10 partitions. The results here included correspond to the average of those achieved in the 10 repetitions.

From the results reported in Table 2, some preliminary comments can be drawn. In general, the second approach clearly obtains lower error rates than the first algorithm: it is better in 7 out of 9 databases (both approaches tie in two problems). When comparing the error rates given by Opt-II with those of the original set (i.e., without performing feature selection), one can observe that the results are quite similar in all cases, but obviously the feature selection algorithm still obtains a significant reduction in the dimensionality of the data. This may become especially important for processing speed efficiency in applications where large training sets and original feature space dimensionalities are involved.

Results corresponding to Opt-I indicate that some features eliminated, although linearly correlated, could be still relevant for the particular classification task. These

Table 1
Some characteristics of the data sets used in the experiments

	Cancer	Glass	Image	Iono	Iris	Liver	Sonar	Vehicle	Wine
Features	9	9	19	34	4	6	60	18	13
Classes	2	6	7	2	3	2	2	4	3
Patterns	683	214	2310	352	150	345	208	848	178

Table 2
Average error rates

	Cancer	Glass	Image	Iono	Iris	Liver	Sonar	Vehicle	Wine
Original	11.3	**54.9**	7.5	11.1	**3.3**	**34.8**	41.9	**23.9**	**3.9**
Opt-I	11.3	55.9	6.1	13.4	7.3	41.9	40.9	48.8	5.0
	(7)	(8)	(11)	(31)	(2)	(5)	(29)	(6)	(11)
Opt-II	**10.6**	55.9	**4.7**	**9.4**	**3.3**	**34.8**	**33.0**	25.4	5.0
	(8)	(8)	(14)	(32)	(4)	(6)	(43)	(13)	(11)

Values in brackets indicate the number of relevant attributes picked by each selection strategy. Bold type represents the best alternative for each database.

removals produce a considerable degradation in classifier performance. In fact, the error rates corresponding to Opt-I are higher than those of Opt-II and the original set of features, clearly due to the elimination of some relevant features. For instance, when applied to Vehicle database, Opt-I discards 12 out of 18 attributes but the error rate is 24.9% and 23.4% higher than those achieved by using the whole set of features and by applying Opt-II, respectively.

4. Conclusions

In the present paper, a new feature selection algorithm based on neural networks has been introduced. In particular, we propose two strategies to identify and remove irrelevant, redundant attributes from the original set of features by using the relative contribution of the input neurons to the output units in a three-layer MLP.

From the experiments carried out, it seems evident that in practice Opt-II is significantly better than Opt-I in terms of error rate, due to the fact that Opt-I wrongly eliminates some relevant attributes (those that are linearly correlated to other features).

Comparing the error rates resulting from Opt-II with those produced with the original set of features, it has to be remarked that in general, differences are not statistically significant. However, the employment of some feature selection strategy should still result especially useful in terms of data dimensionality reduction, which obviously derives in lower computational loads (both in storage requirements and computing times).

Acknowledgements

This work has been supported in part by Grants SEP-2003-C02-44225 from the Mexican CONACYT and TIC2003-08496 from the Spanish CICYT.

References

[1] T. Cibas, F. Soulie, P. Gallinari, Variable selection with neural networks, Neurocomputing 12 (1996) 223–248.
[2] R. Lotlikar, R. Kothari, Bayes-optimality motivated linear and multilayered perceptron-based dimensionality reduction, IEEE Trans. Neural Networks 11 (2000) 452–463.
[3] B. Mak, R.W. Blanning, An empirical measure of element contribution in neural networks, IEEE Trans. Systems, Man, Cybernetics 28 (1998) 561–564.
[4] R. Setiono, H. Liu, Neural network feature selector, IEEE Trans. Neural Networks 8 (1997) 654–662.
[5] A. Verikas, M. Bacauskiene, Feature selection with neural networks, Pattern Recognition Lett. 23 (2002) 1323–1335.