

The Mahalanobis Distance Based Rival Penalized Competitive Learning Algorithm

Jinwen Ma and Bin Cao

Department of Information Science,
School of Mathematical Sciences and LMAM,
Peking University, Beijing, 100871, China
jwma@math.pku.edu.cn

Abstract. The rival penalized competitive learning (RPCL) algorithm has been developed to make the clustering analysis on a set of sample data in which the number of clusters is unknown, and recent theoretical analysis shows that it can be constructed by minimizing a special kind of cost function on the sample data. In this paper, we use the Mahalanobis distance instead of the Euclidean distance in the cost function computation and propose the Mahalanobis distance based rival penalized competitive learning (MDRPCL) algorithm. It is demonstrated by the experiments that the MDRPCL algorithm can be successful to determine the number of elliptical clusters in a data set and lead to a good classification result.

1 Introduction

Clustering analysis is important not only in statistics but also in many aspects of practical applications. Analysis of clusters on a set of (unlabeled) sample data by means of mixture distribution, called mixture-model cluster analysis, is one of the most difficult problems in statistics, and its main task is to select a proper number of clusters in a sample data set. In the terminology of artificial neural networks, it is the problem of unsupervised classification on a sample data set where the number of clusters is unknown. Its importance and difficulty has been noted by many researchers (e.g., [1]-[2]).

As an adaptive version of the classical k -means clustering, competitive learning (CL) has been developed from the field of neural networks and provided us a promising tool for unsupervised classification [3]. In the basic form of competitive learning, as each input sample is presented, the winning unit or neuron of the output layer of a CL network is activated and its corresponding weight vector is modified. While the rest of the units in the output layer are not activated and their corresponding weight vectors are not modified. This form of competitive learning is referred to as the simple or classical competitive learning (CCL) or winner-take-all (WTA) learning [4]. In the late of the 1980's, it was found that this simple learning mechanism has the dead unit (or under-utilized) problem. In order to solve it, the Frequency Sensitive Competitive Learning (FSCL) algorithm was developed under the light of conscience mechanism [5].

However, these conventional competitive learning algorithms cannot be applied to solving the unsupervised classification problem on a set of sample data where the number of clusters is unknown. Xu, Krzyzak & Oja discussed this problem and showed that it is equivalent to the selection of an appropriate number of the units [6]. To tackle this problem, the Rival Penalized Competitive Learning (RPCL) algorithm was further proposed by adding a new mechanism into FSCL [6]. The basic idea is that for each input, not only the weight vector of the winner unit is modified (rewarded) to adapt to the input, but also the weight vector of its rival (the 2nd winner) is de-learned (penalized) by a smaller learning rate. When the learning and de-learning rates are appropriately selected, RPCL has the ability of automatically allocates an appropriate number of weight vectors for a sample data set, with the weight vectors of the extra units being pushed far away from the sample data. Actually, the RPCL algorithm is demonstrated well and has been applied to many fields such as clustering, vector quantization, and the training of RBF neural networks. Moreover, it has also been generalized to several versions for different types of sample data [7].

Via theoretical analysis, it has been recently proved that the RPCL algorithm can be considered as a special case of the distance sensitive RPCL (or DSRPCL) algorithms constructed by minimizing a cost function [8]. However, the distance between a sample point and a weight vector in the cost function is mainly analyzed and used in the form of the Euclidean distance, which actually limits the clusters to the spherical forms.

In this paper, we use the Mahalanobis distance instead of the Euclidean distance in the cost function and propose the Mahalanobis distance based rival penalized competitive learning (MDRPCL) algorithm that can be successful to determine the number of elliptical clusters in a data set and lead to a good classification result.

2 The Cost Function and the MDRPCL Algorithm

Given a set of sample data $\mathcal{S} = \{X^\mu\}_{\mu=1}^N$ with $X^\mu = [x_1^\mu, x_2^\mu, \dots, x_d^\mu]^T \in \mathbb{R}^d$, the simple competitive learning or FSCL algorithm can be regarded as the adaptive versions of the well known k -mean algorithm, which essentially minimize the Mean Square Error (MSE) cost function as follows:

$$E_{MSE}(\mathbf{W}) = \frac{1}{2} \sum_{ij\mu} M_i^\mu (x_j^\mu - w_{ij})^2 = \frac{1}{2} \sum_{\mu} \| X^\mu - W_{c(\mu)} \|^2 \tag{1}$$

where $\mathbf{W} = [W_1, W_2, \dots, W_k]$, and each $W_i = [w_{i1}, w_{i2}, \dots, w_{id}]^T \in \mathbb{R}^d$ is just the i -th weight vector in consideration. Moreover, we have

$$M_i^\mu = \begin{cases} 1 & \text{if } i = c(\mu) \text{ such that } \| X^\mu - W_{c(\mu)} \| = \min_j \| X^\mu - W_j \|, \\ 0 & \text{otherwise,} \end{cases}$$

with $c(\mu)$ denoting the index of the winner unit (or its weight vector) for the μ^{th} sample point.

However, $E_{MSE}(\mathbf{W})$ cannot be used for detecting a correct number k because it decreases monotonically as the number of the units increases. Thus, it does not apply to the RPCL algorithm. Recently, Ma and Wang [8] constructed the following cost function to describe the mechanism of the RPCL algorithm:

$$E(\mathbf{W}) = E_1(\mathbf{W}) + E_2(\mathbf{W}) \tag{2}$$

$$E_1(\mathbf{W}) = \frac{1}{2} \sum_{\mu} \|X^{\mu} - W_{c(\mu)}\|^2, \quad E_2(\mathbf{W}) = \frac{1}{P} \sum_{\mu, i \neq c(\mu)} \|X^{\mu} - W_i\|^{-P},$$

where $\mathbf{W} = \text{vec}[W_1, W_2, \dots, W_n]$ and P is a positive number.

This new cost function is composed of the square mean error E_1 and the model selection term E_2 . Actually, the minimization of E_2 can allocate the proper number of weight vectors to the sample data, while the minimization of E_1 makes the classification of the sample data possible. It is shown by theoretical analysis and experiment in [8] that the minimization of this cost function can lead to a correct number of weight vectors located around the centers of the clusters in the sample data, respectively, with the other weight vectors being driven away far from the sample data. However, the discussions in [8] are mainly based on the Euclidean distance for the cost function, which essentially assumes that the clusters in the sample data are spherical. To relax this constraint, we can use the Mahalanobis distance instead of the Euclidean one in the cost function. That is, the distance $\|W_i - X^{\mu}\|$ is given by

$$\|X^{\mu} - W_i\|^2 = (X^{\mu} - W_i)^T \Sigma_i^{-1} (X^{\mu} - W_i), \tag{3}$$

where Σ_i is the covariance matrix of the cluster i assumed positive definite. Thus, the cost function given in Eq.(2) becomes a function of the parameters $(W_1, \Sigma_1), \dots, (W_n, \Sigma_n)$.

With the above preparations, we now construct the Mahalanobis distance based rival penalized competitive learning algorithm as follows. First we get the derivatives of E with respect to W_j :

$$\begin{aligned} \frac{\partial E}{\partial W_j} = & - \sum_{\mu} \delta_{c(\mu)}^j \Sigma_j^{-1} (X^{\mu} - W_j) \\ & + \sum_{\mu, j} (1 - \delta_{c(\mu)}^j) \|X^{\mu} - W_j\|^{-P-2} \Sigma_j^{-1} (X^{\mu} - W_j) \end{aligned}$$

where δ_j^i is the Kronecker number.

Then, the MDRPCL algorithm for the update of the weight vectors can be constructed as a gradient descent algorithm of the cost function as follows.

$$W_j^{(t)} = W_j^{(t-1)} + \Delta W_j, \tag{4}$$

where $\Delta W_j = -\eta \frac{\partial E}{\partial W_j}$ and $\eta > 0$ is the learning rate which is generally selected as a small positive constant number.

Instead of the batch mode, we can get the following adaptive algorithm for the update of the weight vectors: at current step t , we random choose a sample X^μ , and have

$$\frac{\partial E_1}{\partial W_j} = \begin{cases} -\Sigma_j^{-1}(X^\mu - W_j), & \text{if } j = c(\mu); \\ 0, & \text{otherwise,} \end{cases} \quad (5)$$

$$\frac{\partial E_2}{\partial W_j} = \begin{cases} 0, & \text{if } j = c(\mu); \\ \left\| X^\mu - W_j \right\|^{-P-2} \Sigma_j^{-1}(X^\mu - W_j), & \text{otherwise,} \end{cases} \quad (6)$$

Meanwhile, we use the following update rule to modify the covariance matrix Σ_j at each step in the batch mode:

$$\Sigma_j^{(t)} = \begin{cases} (1 - \eta') \Sigma_j^{(t-1)} + \eta' \sum_{\mu} (X^\mu - W_{c(\mu)})(X^\mu - W_{c(\mu)})^T, & \text{if } j = c(\mu); \\ \Sigma_j^{(t-1)}, & \text{otherwise,} \end{cases} \quad (7)$$

and in the adaptive mode:

$$\Sigma_j^{(t)} = \begin{cases} (1 - \eta') \Sigma_j^{(t-1)} + \eta' (X^\mu - W_{c(\mu)})(X^\mu - W_{c(\mu)})^T, & \text{if } j = c(\mu); \\ \Sigma_j^{(t-1)}, & \text{otherwise.} \end{cases} \quad (8)$$

where $\eta' > 0$ is a small positive constant number. This update rule was given by Xu [7] for the extension of the RPCL algorithm to the data sets of elliptical clusters. Although it does not follow the gradient learning rule of the cost function exactly, it can maintain a good stability on the convergence of the covariance matrices. Actually, the exact gradient learning rule often leads to a degenerate covariance matrix such that the algorithm cannot be convergent.

Since the MDRPCL algorithm is a type of gradient descent algorithm, it may be possible to trap in a local minimum so that it may lead to a wrong clustering result. In order to overcome this problem, we can apply an simulated annealing mechanism to the MDRPCL algorithm in the same way as the ASRPCL algorithm given in [8]. We will use this simulated annealing method to make the classification of the wine data in the next section.

3 Experimental Results

In this section, some simulated experiments are carried out to demonstrate the performance of the MDRPCL. Moreover, the MDRPCL algorithm is applied to the classification of the wine data.

3.1 On Simulated Data Sets

We conducted two experiments on the set of samples drawn from a mixture of four bivariate Gaussians densities (i.e., $d = 2$), which can observed from the backgrounds of Fig.1(a)&(b), respectively. Clearly, all the four Gaussian distributions in each case were selected to be elliptical, with a certain degree

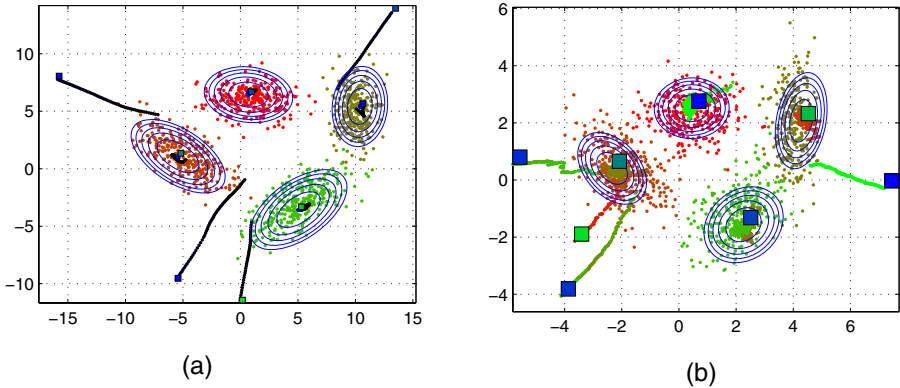


Fig. 1. Trajectories of weight vectors (solid blocks) during the learning process of the MDRPCL algorithm. (a). The batch MDRPCL algorithm in which the learning rate $\eta = 0.001$, $\eta' = 0.0001$, $P = 0.02$, and the number of iterations was 100. (b). The adaptive MDRPCL algorithm in which the parameters were selected as above, but the number of iterations was 1000. Here, the contour lines of each Gaussian based on a pair of final estimated mean vector and covariance matrix obtained from the MDRPCL algorithm are retained unless its density is less than e^{-3} (peak).

of overlap. We implemented the batch MDRPCL algorithm on the data set given in Fig. 1(a) and the adaptive MDRPCL algorithm with the simulated annealing mechanism on the data set given in Fig. 1(b), which has a higher degree of overlap. In the both experiments, we let the parameters W_j being selected randomly and independently from the samples and took $p = 0.02$. The experimental results with $n = 8$ in the two cases are also given in Fig. 1(a)&(b), respectively. We can observe that four Gaussians based on the estimated weight vectors W_j and the covariance matrix Σ_j are finally located accurately at the actual Gaussians, respectively, while the other weight vectors are driven far away from the sample data. That is, the correct number of the elliptical clusters have been detected on each of these data sets. Moreover, the classification accuracies on the two data sets of Fig. 1(a)&(b) can reach at 98% and 93%, respectively.

3.2 On the Wine Data Set

Moreover, we applied the adaptive MDRPCL algorithm with the simulated annealing mechanism to the classification (or recognition) of the wine data set which is a typical real data set for testing a classification or clustering algorithm and can be downloaded from the web: <http://www.ics.uci.edu/~mlern/MLRepository.html>. Actually, it consists of 178 samples of three types of wine. Each datum is 13-dimensional and consists of chemical analysis of a sample from certain type of wine. We first regularized these wine data as into a interval of $[0,3]$ and then apply the simulated annealing MDRPCL algorithm to solving the unsupervised classification problem of the wine data by setting $k = 6$ with the parameters w_{ij} being selected randomly from the interval $[0,3]$ and let $P = 0.2$.

It is shown by the experiments that the simulated annealing MDRPCL algorithm can detect the three classes in the wine data set with a classification accuracy of 100% or 99.64%(there is only one error) which is a rather good result for the unsupervised learning methods.

4 Conclusions

In this paper, we have investigated the function of the Mahalanobis distance in the special cost function associated with the RPCL algorithm and proposed a Mahalanobis distance based RPCL algorithm. By relaxing the Euclidean distance into the Mahalanobis one, the proposed RPCL algorithm is able to be applied to a sample data set of elliptical clusters. It is demonstrated by the experiments that the MDRPCL algorithm can be successful to determine the number of elliptical clusters in a simulated or real data set and lead to a good classification result.

Acknowledgement

This work was supported by the Natural Science Foundation of China for Project 60471054.

References

1. Galinski, R.B., Harabasz, J.A.: A Dendrite Method for Cluster Analysis. *Communications in Statistics* 3(1974) 1-27
2. Hartigan, J.A.: Distribution Problems in Clustering. In J. Van Ryzin (ed.): *Classification and Clustering*. Academic Press, New York (1977) 45-72
3. Nasrabadi, N., King, R.A.: Image Coding Using Vector Quantization: A Review. *IEEE Trans. Commun.* 36 (1988) 957-971
4. Hecht-Nielsen, R.: *Neurocomputing*. Addison-Wesley, Reading MA (1990)
5. Ahalt, S.C., Krishnamurty, A.K., Chen, P., Melton, D.E.: Competitive Learning Algorithms for Vector Quantization. *Neural Networks* 3 (1990) 277-291
6. Xu, L., Krzyzak, A., Oja, E.: Rival Penalized Competitive Learning for Clustering Analysis, RBF Net, and Curve Detection. *IEEE Trans. Neural Networks* 4 (1993) 636-649
7. Xu, L.: Rival Penalized Competitive Learning, Finite Mixture, and Multisets Clustering. In: *Proc. of International Joint Conference on Neural Networks (IJCNN'98)*. vol.3. IEEE Press, Alaska (1998) 2525-2530
8. Ma, J., Wang, T.: A Cost Function Approach to Rival penalized competitive Learning (RPCL). *IEEE Trans. Systems, Man, and Cybernetics B* in Press.