International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 11 (2017) pp. 2745-2753
© Research India Publications. http://www.ripublication.com

A New Statistical competitive Data Clustering Approaches: SRPCL and SRPCCL

Eddarouich Souad[1]*, Zouad Rabab and Hammouch Ahmed *****

** Regional Educational Center CRMEF, Computer Science Department, Rabat, Morocco.*

*** ERIT, LRGE Laboratory, ENSET, Electrical Engineering Department, Mohammed V University, Rabat, Morocco.*

****Postgraduate School of Technical Education, Enset, B.P., 6207 Avenue des Forces Armées Royales, Rabat 10100, Morocco.*

[1]ORCID: 0000-0001-7416-069

Abstract

Finding useful patterns in datasets has attracted considerable interest recently, and one of the most widely studied techniques is data clustering. Clustering is the unsupervised classification of patterns (observations, data items, or feature vectors) into clusters (groups) and it is one of the most important research and application areas of neural networks. This paper summarizes some of the most important developments in competitive learning algorithms for data clustering and presents two new fast learning approaches based on the competitive concept. The first one used the rival competitive learning (RPCL) concepts and the second one investigates the rival penalized controlled competitive learning (RPCCL) concepts, to dynamically control the rival-penalizing forces.

Keyword: k-means competitive learning RPCL RPCCL; mode detection procedure.

INTRODUCTION

Cluster analysis is an important statistical methodology used in a wide variety of fields including machine learning, pattern recognition, image analysis, information retrieval, and bioinformatics …etc.

The objective of clustering is to partition M data points $x = \{x_1, x_2, ... x_M\}$ into K non-empty subsets such that alike data are grouped together and data in different subsets or clusters are not alike. The statistical approach in cluster analysis postulates that the input pattern are drawn from an underlying probability density function (pdf) which describes the distribution of the data points through the data space. Regions of high local density, which might correspond to significant classes in the population, can be found from the peaks or the modes of the density function estimated from the available patterns. Then, the key problem is to partition the data space with a multimodal pdf into subspaces over which the pdf is unimodal [1].

Up to now, numerous techniques have been proposed to deal with the clustering problem, but neural networks are considered as the most interesting alternative to the conventional methods [1], [2], [3]. The advantage of neural networks lies in their ability to adjust themselves to the data without any explicit specification of functional or distributional form for the underlying model. Second, they are nonlinear models, which make them flexible in modeling real world complex relationships. Finally, neural networks have shown their effectiveness in a variety task of real world: medical diagnosis [4], recognition of handwritten signatures, charactersand digits [5], [6], [7], weather forecasting [8], [9], [10], financial predictions [11], [12] and classification applications[13],[14],[15]. In the literature, k-means [16] is a popular competitive learning algorithm; it starts with K random centers, then picks up a vector randomly from the input data set, computes the distance with each center, chooses the minimum one and moves the center towards to the corresponding input vector by a ratio.

The k-means algorithm is easy to implement but it suffers from two major drawbacks: 1) the problem of dead units i.e. if the initial positions of some centers are inappropriately chosen (far away from the inputs comparing to the other centers), they may never be trained and, therefore, immediately become dead units. 2) We must specify the exact number of k, if this number is not equal to the true cluster number; the performance of k-means algorithm deteriorates rapidly. Eventually, some of the seed points are not located at the centers of the corresponding clusters. Instead, they are either at some boundary points between different clusters or at points biased from some cluster centers. After the K-means comes the standard competitive learning [17] to implement the K-means in neuronal structure that has the advantage of parallelism, robustness and fault tolerance. To circumvent the problems of dead units, the frequency sensitive competitive learning (FSCL) [18] has been proposed to perform a better clustering.

Other algorithms developed were Rival Penalized Competitive Learning (RPCL) [19] and its variants such as DPRCL [20] and RPCCL [21], which try to remove the problem of pre-selecting the number k.

We propose, in the present paper, a new approach for the detection of the mode without using neither differential operators nor any procedures of filtering. This approach is based on neural network [7,8], and more particularly on the rival competitive network. It begins by the estimation of the probability density function (pdf), followed by a training competitive neural network. This stage allows detecting the local maxima of the pdf, considered as marquers of the modes of this function and also as the prototypes of present clusters in the data set. In this context, two alternatives of training are proposed here. The first one used the RPCL concepts and the second one investigates the RPCCL concepts, to dynamically control the rival-penalizing forces.

The section II of this paper introduce briefly each competitive data clustering approach derived from the k-means algorithm. Section III is consecrated to the presentation of the New Statistical competitive Data Clustering Approaches: SRPCL and SRPCCL. Section IV illustrates by means of examples using data generated artificially, the advantage of the new approach, compares the performance of the two training procedures SRPCL and SRPCCL.

OVERVIEW OF THE APPROACHES

Frequency sensitive learning algorithm

The frequency sensitive competitive learning (FSCL) [18] introduced in 1990 is an extension of the k-means algorithm to overcome the problem of dead units. The basic idea is to introduce a relative winning frequency or conscience term [22] into the similarity measurement between an input and the seed points. The larger the winning frequency of a seed point, the more it is penalized. The relative winning frequency coefficient is defined as:

$$\rho_k = \frac{n_j}{\sum_{r=1}^{k} n_r} \tag{1}$$

The winner is selected according to this equation:

$$g = Arg \min_{0 \le k \le K} \left(\rho_k D\left[C_k, X\right]\right) \tag{2}$$

Where n_i is the cumulative number of the occurrences of the center C_k in the past and $D\left[C_k, X\right]$ is the distance between the center C_k and the winner.

Although the FSCL algorithm can almost perform a successfully clustering without the dead units problem, but it suffers from the same second problem as k-means: the need to specify the exact number of k clusters. This algorithm has been used extensively for vector quantization of speech waveforms [23] and for image compression [24].

Rival penalized competitive learning

RPCL is an improvement of the FSCL algorithm that automatically determines the number of classes in the presence of a data sample while keeping the characteristic of solving the problem of dead units. The basic idea of this learning rule is to converge the winner center toward to the input, and push the rival center i.e. the second winner away from the data set [19].

Therefore, the updating of the winner and the rival center is done according to these equations:

$$\begin{cases} C_k(t) = C_k(t-1) + \alpha_g(t)\left[X - C_k(t-1)\right] & if \quad k = g \\ C_k(t) = C_k(t-1) - \alpha_r(t)\left[X - C_k(t-1)\right] & if \quad k = r \\ C_k(t) = C_k(t-1) & if \quad k \neq g \quad et \quad k \neq r \end{cases} \tag{3}$$

Where: $0 \le \alpha_g(t) \le 1$ and $0 \le \alpha_r(t) \le 1$ are respectively, the learning coefficients for the winner center and its rival. And $\alpha_g(t)$ is much greater than $\alpha_r(t)$. In some literatures, $\alpha_g(t) = 0.001$ and $\alpha_r(t) = 0.0001$

However, the use of RPCL clustering offers an interesting alternative to the K-means and FSCL algorithms, but its convergence is very sensitive to the choice of parameters $\alpha_g(t)$ *and* $\alpha_r(t)$. Among the applications of RPCL, we mention color image segmentation [25], nonlinear channel equalization [26] and images features extraction [27].

Rival penalized controled competitive learning

The RPCCL algorithm [21] is similar to the RPCL because it has the same advantage of automatically determining the number of classes. Indeed, if the number of the classes fixed by the user, in the initialization phase, is higher than the real number of classes, extra neurons are driven away from all observations. In comparison with the RPCL algorithm, the RPCCL algorithm applies a new mechanism to dynamically control the penalization of those neurons (the rivals) by introducing a new term called the penalization strength [21]:

$$p(X, C_g(t), C_r(t)) = \frac{\min\left(D\left[X, C_g(t)\right], D\left[C_g(t), C_r(t)\right]\right)}{D\left[C_g(t), C_r(t)\right]} \tag{4}$$

Where $C_g(t)$ and $C_r(t)$ are respectively, the winner center and its rival. The updating of the weight vectors of the rival becomes:

$$C_r(t) = C_r(t-1) - \alpha_r(t) p(X, C_g(t), C_r(t))\left[X - C_r(t-1)\right] \tag{5}$$

From (5), it can be seen that the rival will be fully penalized with the rate $\alpha_r(t)$ if the distance between the winning center and its first rival is smaller than the distance between the winning center and the input. Otherwise, the rival will be penalized with the strength $\alpha_r(t)p(X, C_g(t), C_r(t))$.

Consequently, the RPCCL generally drives the undesired centers far away from the clusters much faster than the RPCL, because its de-learning rate is greater.

THE SRPCL AND SRPCCL PROCEDURES

The pdf's mode detection procedure is a neuronal probabilistic approach presented in [28]. This approach is carried out in three stages processing: the first one consists in estimating the underlying pdf using a non-parametric estimator. In the second stage, an artificial neuronal network with competitive training is used for extracting the local maxima of the pdf. Finally in the third stage, the pdf modes are detected using a probabilistic approach.

Based on that procedure and also on the the concepts of the RPCL and RPCCL approaches, we propose in the following section a two alternatives learning approaches named respectively statistical rival penalised competitive learning (SRPCL) and statistical rival penalised controled competitive learning (SRPCCL) to detect the local maxima of the pdf. Each step of the procedure will be explained below.

The estimation of underlying probability density function

Let $\Gamma = \{X_1, X_2, ..., X_Q\}$, be the set of Q N-dimensional observations of a random variable X with a probability density function $P(X)$.To estimate this underlying density function when what is available is only a set $X_q = \{x_{q,1}, x_{q,2}, ..., x_{q,n}, ..., x_{q,N}\}$, q=1,2,...,Q of Q observations, the analyst may use non-parametric techniques.

The Parzen window method [29] proves well adapted to the proposed procedure in this paper. However, this estimation procedure needs prohibitive calculus when the dimension of the space is very important. So, we have opted for the fast estimation algorithm which is proposed by Postaire and Vasseur [30].

First, the range of variation of each component of these observations, is normalised to the interval [0,R], where R is an integer such as $R \geq 2$, by means of the transformation defined as:

$$y_{n,q} = \frac{\left(x_{n,q} - \min_q x_{n,q}\right)}{\left(\max_q x_{n,q} - \min_q x_{n,q}\right)} * R \tag{6}$$

Each axis of the so normalized data space is then partitioned into R exclusive intervals of unit width. This discritisation defines a set of $\Re^N$ hypercubes of unit side length. Each hypercube noted $H(X)$, is a site defined by its N coordinates $x_1, x_2, ..., x_n, x_N$ which are the integer parts of the coordinates of its center X.

To be more specific, let $y_q = \lfloor y_{1,q}, y_{2,q}, ..., y_{n,q}, ..., y_{N,q} \rfloor$, q=1, 2, Q be the Q observations in the normalized space. Each observation Y_q is found inside a non-empty hypercube with the coordinates $x_n = \text{int}(y_{n,q})$, n=1,2,..,N, where $\text{int}(y_{n,q})$ designates the integer parts of $y_{n,q}$. If several observations fall in the same hypercube, this one appears many times on the list of non–empty hypercubes. Furthermore, the number of times the hypercube $H(X)$ appears in that list indicates the number of data points $q[H(X)]$which fall in this hypercube. Subsequently, the value of the local density estimated is:

$$p(X) = \frac{q[H(X)]}{Q} \tag{7}$$

Since the volume of $H(X)$ is equal to unity.

So, this fast procedure allows only the estimation of the underlying probability density function at the centers of the non-empty hypercubes whose number never exceeds the number Q of available observations. At the centers of the hypercube cells, which are not on that list, the density estimates are known to be null. At the end of this fast algorithm, all the available information for clustering is in the discrete set $\underline{X}$ of estimated values of the underlying probability density function $p(X)$.

The extraction of local maxima by neural network

In order to detect the local maxima of the pdf (the pdf's modes), a competitive learning algorithm with an arbitrary number of these local maxima has been used [28].

The architecture and the activation functions of the elaborated network for the extraction of these maxima will be the same as those of the neural networks with competitive training (RNAC) [28]. In the training algorithm, we work, and only, on the pdf by presenting sequentially to the network the centers of the non-empty hypercubes of the set $\underline{X}$ instead of the Q observations like in the RNAC.

This technique allows us to envisage a great reduction of convergence time of the elaborated network because the number of the non-empty hypercubes presented to the network is widely less than the observations presented to RNAC (see the experiments section)..

The neural network elaborated is composed of two layers: the input layer and the output layer ("Fig. 1"). The first one is made of N units I_n, n=1,2,...,N, such that unit I_n is solicited by the attribute X_n of the non-empty hypercube $H(X)$ when this one is presented to the network. However, each output neurone materialises an hypercube which represents the site of one local maximum of the pdf in the set X. The proposed method is unsupervised, the number of the output units, which is that of local maxima, is firstly initialised arbitrary.

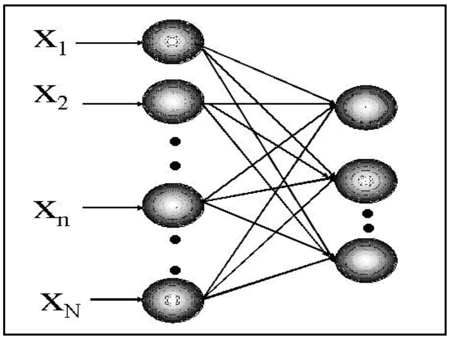


Figure 1. Competitive Neural Network

Let $H^t(X)$, be a non-empty hypercube where the coordinates $x_1, x_2, ..., x_n, x_N$ of its center are presented to the network at the iteration t. Each output neurone defines a non-empty hypercube from the subset X, and is materialised with the mean vector $\mu_k^t(X)$ of this neurone in which is associated the value of the pdf function $f\left[\mu_k^t(X)\right]$ estimated in this hypercube. We note by $D\left[\mu_k^t(X), H^t(X)\right]$ the distance which separates the mean vector $\mu_k^t(X)$ from the center of the presented hypercube $H^t(X)$ [28].

The training of the network has been done by the competitive concepts, so as to estimate its parameters namely: the local maxima of the pdf and their locations in the discrete space X [28].

So, on every iteration we present to the network the coordinates of the center of a non-empty hypercube. The K output neurone with respectively the mean vectors $\mu_k^t(X)$, k=1,2,...,K gets on competition by calculating for each one the distance $D\left[\mu_k^t(X), H^t(X)\right]$ and comparing the values of the pdf associated to $\mu_k^t(X)$ and to $H^t(X)$. As we will see in the following algorithm, the exposed information by the new hypercube $H^t(X)$ is used to update the mean vector and the value of the pdf of the winner and the rival neurones.

Training algorithm using SRPCL:

- *Initialisation phase:*
  - Initialise the mean vectors $\mu_k(X)$; k=1,2,...,K, of the K output neural, with an arbitrary choice of K non-empty hypercubes from the set $\underline{X}$.
  - Initialise the coefficients of the training function β_0 and τ.
  - Initialise the iteration parameter to zero.
  - Initialise the set Ω into an empty set: $\Omega = \phi$.
- *Processing phase:*

1. t = t+1;
2. Present to the network, with an arbitrary pulling, a non-empty hypercube $H^t(X) \in C_{\underline{X}}^{\Omega}$.
3. $\Omega = \Omega \cup H^t(X)$.
4. Search the output neurone which is its mean vector;

   $\mu_g^t(X)$ is the closest to $H^t(X)$. Inspired by the FSCL[18], it is proposed to multiply the distance between the mean vector of a neurone and the center of the hypercube which is presented to the entry of the network with a conscience term proper to each neurone. The winner neurone is defined by calculating the distance so as:

$$D\left[\mu_g^t(X), H^t(X)\right] = \frac{1}{\alpha_g^t} \min_{k=1}^{K} \alpha_k^t D\left[\mu_k^t(X), H^t(X)\right] \tag{8}$$

The second winner or the rival is defined by calculating the distance:

$$D\left[\mu_r^t(X), H^t(X)\right] = \min_{k=1}^{K} \alpha_k^t D\left[\mu_k^t(X), H^t(X)\right] \tag{9}$$

International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 11 (2017) pp. 2745-2753
© Research India Publications. http://www.ripublication.com

5. Compare the functions $P\left[\mu_g^t(X)\right]$ and $P\left[H^t(X)\right]$. If $P\left[\mu_g^t(X)\right] < P\left[H^t(X)\right]$ update the winner and the rival neurones as follow :

$$\begin{cases} \mu_g^t(X) = \mu_g^{t-1}(X) + \alpha_g(t)\left[H_i^t(X) - \mu_g^{t-1}(X)\right] \\ \mu_r^t(X) = \mu_r^{t-1}(X) - \alpha_r(t)\left[H_i^t(X) - \mu_r^{t-1}(X)\right] \\ p\left[\mu_g^t(X)\right] = p\left[H^t(X)\right] \end{cases} \tag{10}$$

Where $\alpha_g(t)$ and $\alpha_r(t)$ are determined by the analyst in the initialization phase.

Else go to step2

6. Stopping criteria: Compare $\mu_k^t(X)$ to $\mu_k^{t-1}(X)$ for k=1,2,.,K. If $(\mu_k^t(X) \neq \mu_k^{t-1}(X)) \forall k = 1,2,...K$, go to step 1.

Else, end of the processing.

Training algorithm using SRPCCL:

The SRPCCL training algorithm is close to the SRPCL algorithm, but utilizes a novel mechanism to control the rival penalization. The idea of this mechanism is that the rival should be fully penalized if the winner suffers from severe competition from the rival; otherwise, the penalization strength should be proportional to the degree of competition level. Thus the SRPCCL algorithm is exactly the same as RPCL, the only difference is that the rival neuron will be penalized in step 5 as follows:

$$\mu_r^t(X) = \mu_r^{t-1}(X) - P\left(H^t(X); \mu_g(X), \mu_r(X)\right)\alpha_r(t)\left[H_i^t(X) - \mu_r^{t-1}(X)\right] \tag{11}$$

Where:

$$P\left(H^t(X); \mu_g(X), \mu_r(X)\right) = \frac{\min\left[D\left[\mu_g(X), \mu_r(X)\right], D\left[\mu_g(X), H^t(X)\right]\right]}{D\left[\mu_r(X), \mu_g(X)\right]}$$

The goal of this learning stage is to detect the local maxima of the pdf. To adjust the number K of these hypercubes, which is the output neurons, we propose a new method that consists in placing all the rival neurons in a pointer during the learning phase and to use that pointer for eliminating all the extra local maxima in the final stage of the process.

EXPERIMENTS

To show the performance of the new clustring approaches, we conducted the following experiment using Gaussian and non-Gaussian classes generated artificially. The statistical parameters of Gaussians are:

Table 1. Statistical parameters of gaussian classes

Classes	Statistical Parameters		
	Means	***Variances***	***Number of points***
1	0.0914 0.0914	3.2064 2.7353	400
2	5.0488 9.0809	3.3884 3.9367	400

Where attributes describing the observations of two classes nonlinearly distributed in the sample of this example are defined by:

$$x_1 = A_1 \cos\theta + B_1$$
$$x_2 = A_2 \sin\theta + B_2$$

With A1 = A2 = 10. the statistical parameters of two non-Gaussian classes is shown in the table 2,

TABLE I. Statistical parameters of non-gaussian classes

Classes	Statistical Parameters		
	θ	***B1***	***B2***	***Number of points***
3	$m = 75$ $s = 45$	$\mu_1 = 0$ $\sigma_1 = 3$	$\mu_1 = -2$ $\sigma_1 = 3$	500
4	$m = 225$ $s = 30$	$\mu_1 = 4$ $\sigma_1 = 2$	$\mu_1 = -5$ $\sigma_1 = 2$	300

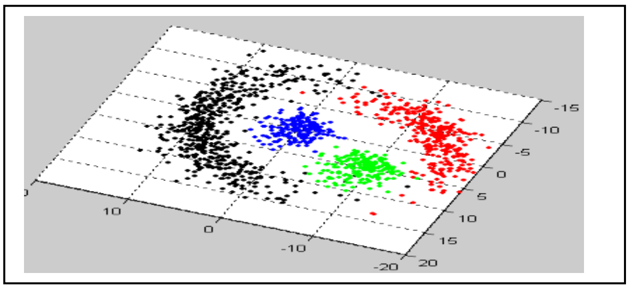


Figure 2. The generated data set

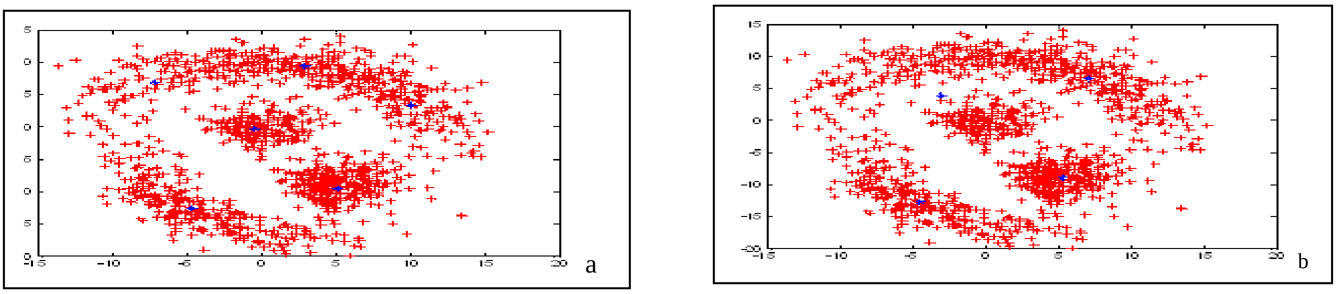


Figure 3(a) The final positions of center of classes obtained via FSCL algorithm with epoch = 100 and six neurons as final number of clusters. **(b)** The final positions after determining the exact number of classes.

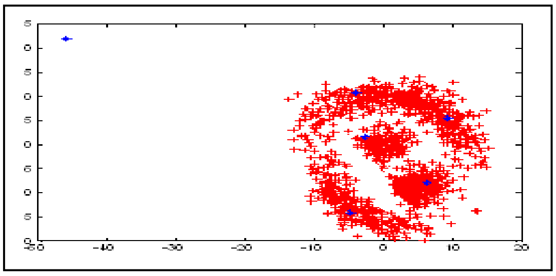


Figure 4. The final position of six center of classes obtained via RPCL with epoch = 100,, where only one neuron is far away from the input set.

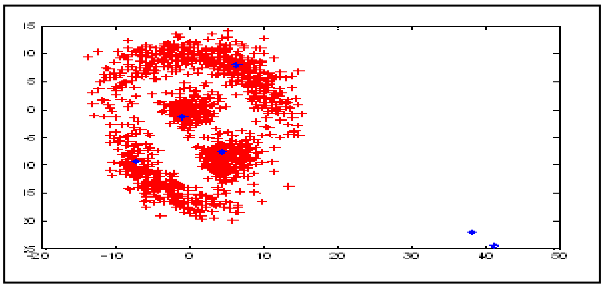
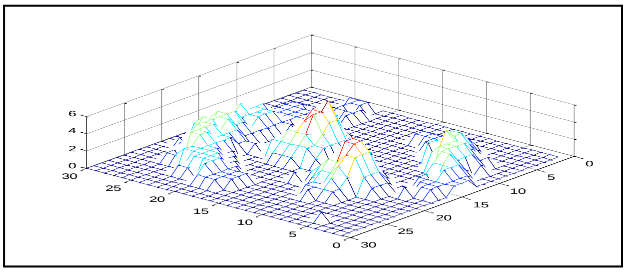
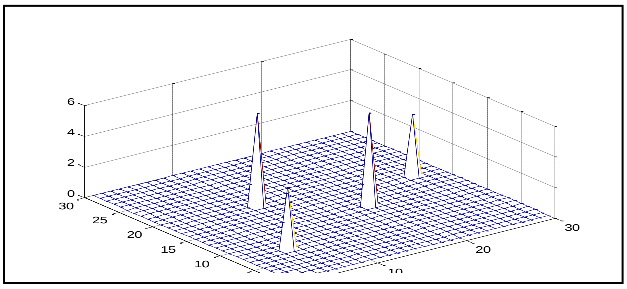


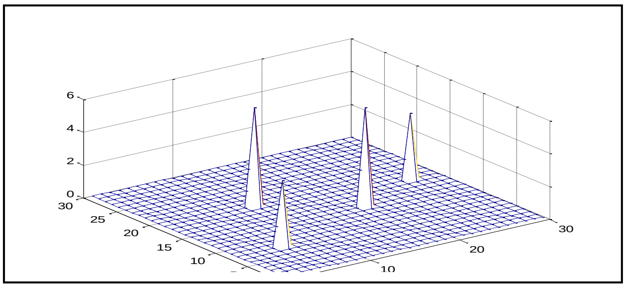
Figure 5. The final position obtained via RPCCL with epoch = 100, where all ahe extra neurons are far away from the input set.



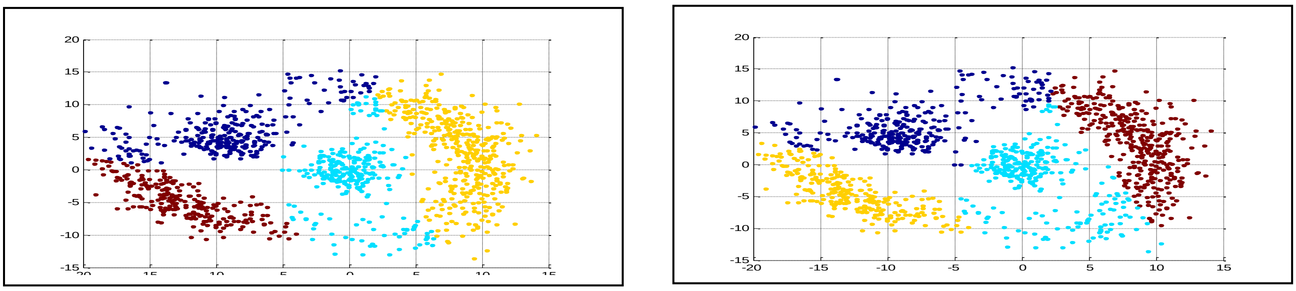
(a)



(b)



(c)



(d) (e)

Figure 6. The SRPCLand SRPCCL procedures. (a) Underlying pdf. (b) The modes of the pdf using SRPCL. (b) The modes of the pdf using SRPCCL (d) The classified data using SRPCL (error=0.4266) (e) The classified data using SRPCCL(error=0.3772).

The experiments have shown the outstanding performance and rapidity of the statistical training approaches in comparison with RPCL, RPCCL and FSCL. The so detected modes during the step of learning are used in the last step of this approach for the classification process ‘fig 6-d’ and ‘fig 6-e’. Compared to the K-means clustering or to the clustering approaches based on the different competitive learning schemes, the proposed approach has proven, that does not pass by any thresholding and does not require any prior information on the number of classes nor on the structure of their distributions in the dataset.

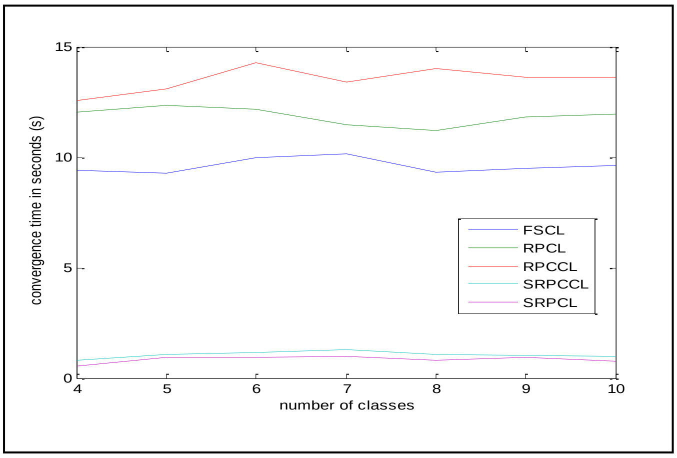


Figure 7. Comparison of the convergence time of different algorithms

To better achieve the reliability of the mode detection procedure, we compared its performance with those of RPCL, FSCL and RPCCL. “Fig. 7” shows the convergence time of the four procedures, applied on the above data set, depending on the number of the output neurons of the network.

According to the results in “Fig. 7”, we see that the proposed algorithm is more efficient than the FSCL, RPCL and RPCCL. These algorithms process observation by observation, while the proposed network handles the non-empty hypercubes. Given that the number of these hypercubes is significantly less than the total number of observations of the sample, the proposed algorithm converges faster.

CONCLUSION

We have further investigated some important algorithms in the competitive learning. An unsupervised clustering method based on mode detection of underlying pdf has been introduced; this approach has proven, additionally to its speed, the ability to automatically determine the number of classes without any human intervention.

REFERENCES (10 PT)

[1] E.W. Forgy, “Cluster Analysis of Multivariate Data: Efficiency Versus Interpretability of Classifications,” Proc. Biometric Soc. Meetings, 1965.

[2] H.Q. Wang and D.S. Huang, “A Novel Clustering Analysis Based on PCA and SOMs for Gene Expression Patterns,” Lecture Notes in Computer Science, vol. 3174, pp. 476-481, Springer-Verlag, 2004.

[3] L. Xu, A. Krzyz jak, and E. Oja, “Rival Penalized Competitive Learning for Clustering Analysis, RBF Net, and Curve Detection,” IEEE Trans. Neural Networks, vol. 4, pp. 636-648, 1993.

[4] Amato, Filippo, et al. "Artificial neural networks in medical diagnosis." Journal of applied biomedicine 11.2 (2013): 47-58.

[5] Miah, Mohammad Badrul Alam, et al.

"Handwritten Courtesy Amount and Signature Recognition on Bank Cheque using Neural Network." International Journal of Computer Applications 118.5 (2015).

[6] Yang, Jie, et al. "Robust Multi-Layer Hierarchical Model for Digit Character Recognition." JOURNAL OF ELECTRICAL ENGINEERING & TECHNOLOGY 10.2 (2015): 699-707.

[7] Sharma, Ankit, and Dipti R. Chaudhary. "Character recognition using neural network." International Journal of Engineering Trends and Technology (IJETT)-Volume4 (2013): 662-667.

[8] Rajkumar, R., A. James Albert, and D. Chandrakala. "Weather Forecasting using Fuzzy Neural Network (FNN) and Hierarchy Particle Swarm Optimization Algorithm (HPSO)." Indian Journal of Science and Technology8.12 (2015).

[9] Shrivastava, Gyanesh, et al. "Application of artificial neural networks in weather forecasting: a comprehensive literature review." International Journal of Computer Applications 51.18 (2012).

[10] Matouq, Mohammed, et al. "The climate change implication on Jordan: A case study using GIS and Artificial Neural Networks for weather forecasting."Journal of Taibah University for Science 7.2 (2013): 44-55.

[11] Aydin, Alev Dilek, and Seyma Caliskan Cavdar. "Prediction of Financial Crisis with Artificial Neural Network: An Empirical Analysis on Turkey." International Journal of Financial Research 6.4 (2015): 36.

[12] Geng, Ruibin, Indranil Bose, and Xi Chen. "Prediction of financial distress: An empirical study of listed Chinese companies using data mining." European Journal of Operational Research 241.1 (2015): 236-247.

[13] Harikumar, R. "Performance analysis of neural networks for classification of medical images with wavelets as a feature extractor." International Journal of Imaging Systems and Technology 25.1 (2015): 33-40.

[14] Levi, Gil, and Tal Hassner. "Age and gender classification using convolutional neural networks." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2015.

[15] Capizzi, Giacomo, et al. "Automatic classification of fruit defects based on co-occurrence matrix and neural networks." Computer Science and Information Systems (FedCSIS), 2015 Federated Conference on. IEEE, 2015.

[16] J.B. MacQueen, "Some Methods for Classification and Analysis of Multivariate Observations," Proc. Fifth Berkeley Symp. Math. Statistics and Probability, vol. 1, Berkeley, Calif.: Univ. of California Press, pp. 281-297,1967.

[17] Ruhmelhart, D.E. et Zipser. D. Feature discovery by competitive learning. cognitive science vol. 9, pp 75-112, 1985.

[18] S.C. Ahalt, A.K. Krishnamurthy, P. Chen, and D.E. Melton, Competitive Learning Algorithms for Vector Quantization, Neural Networks, vol. 3, pp. 277-291, 1990.

[19] L. Xu. A. Krryzak, and E. Oja, "Rival Penalized Competitive Leaming for Clustering Analysis, REF Net, and Curve Detection", IEEE Transactions on Neural Networks. Val. 4, No. 4, pp. 636449, 1993.

[20] Corina Botoca, Georgeta Budura, "Complex data clustering using a new competitive learning algorithm", ELEC. ENERG. vol. 19, no. 2, pp. 261-269, 2006.

[21] Yiu-ming Cheung: On Rival Penalization Controlled Competitive Learning for clustering with automatic Cluster Number selection [J], IEEE Trans. Knowledge and Data Engineering, VOL.17, NO.11, 1583- 1588, NOV.2005.

[22] S.C.AHA & A.K.KRISHNAMURTHY & CHEN. Prakoon Competitive Learning Algorithms for Vector quatization.. Neural Networks, vol-3, pp. 277-290, 1990.

[23] Ahalt, S. C., A. K. Krishnamurthy, P. Chen and D.E. Melton. "Competitive leaming algorithms for vector qunatization." Neural Networks, Vol. 3, pp 277-290, 1990.

[24] Bouzerdoum, A. and T. J. Alton. "Image compression using frequency sensitive competitive leaming." Proc. of IEEE Workshop on Visual Signal Processing & Communications, pp 89-92.21-22 September 1993, Melbourne, Australia.

[25] B. Fritzke, ―Growing Cell Structures—A Self-Organizing Network for Unsupervised and Supervised Learning, Neural Networks, vol. 7, no. 9, pp. 1441-1460, 1994.

[26] S.C. Ahalt, A.K. Krishnamurthy, P. Chen, and D.E. Melton, Competitive Learning Algorithms for Vector Quantization, Neural Networks, vol. 3, pp. 277-291, 1990.

[27] P. Guo, C.L. Philip Chen, and M.R. Lye, ―Cluster Number Selection for a Small Set of Samples Using the Bayesian Ying-Yang Model, IEEE Trans. Neural Networks, vol. 13, no. 3, pp. 757-763, 2002.

[28] Souad Eddarouich and Abderrahmane Sbihi, Neural Network for Modes Detection in Pattern Classification, ICTIS'07. Information and Communication Technologies International Symposium, Morocco, Fez, 3-5 April 2007,pp. 300-303.

[29] E. Parzen . "On Estimation of a Probability Density Function and Mode". Ann. Math. Stat., vol. 33, pp. 1065-1076. 1962.

[30] J.-G. Postaire et C. P. A. Vasseur "A fast Algorithm for non Parametric Probability Density Estimation". IEEE, Trans. on Pattern Anal. and Machine Intel. PAMI-4, n°6, pp. 663-666, 1982.