Cooperative and Penalized Competitive Learning for Clustering Analysis

Hong Jia
Department of Computer Science, Hong Kong Baptist University
hjia@comp.hkbu.edu.hk

Abstract

Competitive learning approaches with penalization or cooperation mechanism have been applied to unsupervised data clustering due to their attractive ability of automatic cluster number selection. In this paper, we further investigate the properties of different competitive strategies and propose a novel learning algorithm called Cooperative and Penalized Competitive Learning (CPCL), which implements the cooperation and penalization mechanisms simultaneously in a single competitive learning process. The integration of these two different kinds of competition mechanisms enables the CPCL to have good convergence speed, precision and robustness. Experiments on synthetic and real data sets are performed to investigate the proposed algorithm. The promising results demonstrate its superiority.

1 Introduction

As a very efficient approach to clustering analysis, competitive learning has been widely applied to a variety of research areas such as data mining [1], image progress [2], Bioinformatics [3] and so forth. In the literature, a typical competitive learning algorithm is k-means [4] that learns k pre-assigned seed points (also called units or centroids interchangeably) on the basis of minimizing the mean-square-error (MSE) function. Although k-means has a variety of applications in different areas, it suffers from a selection problem of cluster number as pointed out in [5, 9]. That is, k-means needs to pre-assign the number of clusters exactly; otherwise, it will almost always give out an incorrect clustering result.

To solve this selection problem, there have been two main kinds of techniques in the literature. The first one is to utilize a criterion, such as the Akaike's information criterion (AIC) [6, 19], the minimum description length (MDL) [21], the Bayesian inference criterion (BIC) [7] and the minimum message length (MML) [8, 20], to select an appropriate number of clusters by optimizing a nonlinear function over all candidates of cluster number. However, due to the repeating process for different values of cluster number k, these methods incur a large computational cost [22]. The other kind of technique is to introduce some competitive learning mechanisms into an algorithm so that it can perform automatic cluster number selection during the learning process. For example, the Rival Penalized Competitive Learning (RPCL) [9] can automatically select the cluster number by gradually driving extra seed points far away from the input data set. In this learning approach, for each input, not only the winner is updated to adapt to the input, but also the rival nearest to the winner (i.e., the second winner) is penalized by a much smaller fixed rate (also called delearning rate hereinafter). Nevertheless, the empirical studies have also found that the RPCL may completely break down without an appropriate delearning rate. Under the circumstances, paper [10] has proposed an improved version, namely Rival Penalization Controlled Competitive Learning (RPCCL), which determines the rival-penalized strength through an adaptive way based on the distance between the winner and the rival relative to the current input. Subsequently, the delearning rate in this algorithm can be fixed at the same value as the learning rate. However, both of RPCL and RPCCL always penalized the extra seed points even if they are much far away from the input data set. Consequently, the seed points as a whole will not tend to convergence. By contrast, another variant of RPCL called Stochastic RPCL (S-RPCL) [11], developed from the Rival Penalized Expectation-Maximization (RPEM) algorithm, can lead to a convergent learning process by penalizing the nearest rival stochastically based on its posterior probability. Nevertheless, when the data clusters are overlapped, the convergence speed of S-RPCL, as well as the RPCL, may become slow and the final locations of seed points may have a bias from the cluster centers.

Alternatively, Competitive and Cooperative Learning (CCL) [12] implements a cooperative learning process, in which the winner will dynamically select several nearest competitors to form a cooperative team to adapt to the input together. The CCL can make all the seed points converge to the corresponding cluster centers and the number of those seed points stayed at different positions is exactly the cluster number. Nevertheless, further experiments indicate that the performance of CCL is somewhat sensitive to the initial positions of seed points. To overcome this difficulty, Li et al. [13] have proposed an improved variant; namely Cooperation Controlled Competitive Learning (CCCL) method, in which the learning rate of each seed point within the same cooperative team is adjusted adaptively based on the distance between the cooperator and the current input. The CCCL has inherited the merits of CCL and is insensitive to the initialization of the seed points. Nevertheless, the CCCL may still not work well if the initial seed points are all gathered in one cluster.

In this paper, we will present a new competitive learning algorithm, namely Cooperative and Penalized Competitive Learning (CPCL), which performs the two different kinds of learning mechanisms simultaneously: cooperation and penalization, during the single competitive learning process. That is, given an input, the winner generated from the competition of all seed points will not only dynamically select several nearest competitors to form a cooperative team to adapt to the input together, but also penalize some other seed points which compete intensively with it. The cooperation mechanism here enables the closest seed points to update together and gradually converge to the corresponding cluster centers while the penalization mechanism supplies the other seed points with the opportunity to wander in the clustering space and search for more appropriate cluster centers. Consequently, this algorithm features the fast convergence speed and the robust performance against the initialization of the seed points. The experiments have demonstrated the outstanding performance of the CPCL on both synthetic and real data. Furthermore, it is also found that the CPCL is robust against the overlap of the data cluster to a certain level, and gives a quite good estimate of the cluster centers.

The rest of this paper is organized as follows. Section II describes the proposed Cooperative and Penalized Competitive Learning approach and gives out the corresponding algorithm. Then, Section III shows the experimental results on some synthetic and real data sets. Finally, we draw a conclusion in Section IV.

2 Cooperative and Penalized Competitive Learning (CPCL) Approach

2.1 Cooperation and Penalization Mechanisms in CPCL

Here we will first describe the cooperation and penalization mechanisms of CPCL learning approach. Suppose N inputs, $X_1, X_2, ..., X_N$, come from k^* unknown clusters, and k ($k \geq k^*$) seed points $m_1, m_2, ..., m_k$ are randomly initialized in the input space. Subsequently, given an input

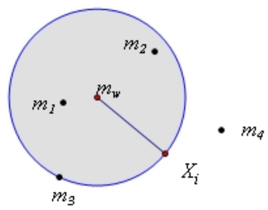


Figure 1. The territory of the winner m_w, indicated by a shadow circle, in the competition with the other seed points as given an input X_i.

X_i each time, as described in [14], the winner among k seed points is determined by

$$I(j|X_i) = \begin{cases} 1 \ \ if \ \ j = \arg\min_{1 \leq r \leq k} \gamma_r ||X_i - m_r||^2, \\ 0 \ \ otherwise; \end{cases} \tag{1}$$

with the relative winning frequency γ_r of m_r defined as

$$\gamma_r = \frac{n_r}{\sum_{j=1}^{k} n_j}, \tag{2}$$

where n_j indicates the winning times of m_j in the past. That means the winning chances of frequent winning seed points are gradually reduced by an implicit penalty. After selecting out the winner m_w, as shown in Fig. 1, the circle centered at m_w with the radius $||m_w - X_i||$ is regarded as the territory of m_w. Any other seed points which have intruded into this territory will be dominated by m_w. That is, any other seed points fallen into or on the circle, as m_1, m_2 and m_3 in Fig. 1, will either cooperate with the winner or be penalized by it.

The winner m_w in this learning approach always chooses the seed points nearest to it as its cooperators and the number of cooperators needed by the winner is gradually increased as the learning process repeats. For example, if current status is the first epoch of the whole learning algorithm, the winner will not choose any cooperators but penalize all the seed points which have intruded into its territory. Then, in the second learning epoch, the winner will select one seed point which is nearest to it in its territory to form a cooperating team and penalize the other intruders. Consequently, for the *t*-th learning epoch, the number of cooperators chosen by the winner can be denoted as C_t, where $C_t = \min\{t-1, k-1\}$. This kind of cooperating scheme ensures that the seed points have enough opportunities to drift in the whole input space and converge smoothly.

After choosing cooperators, each member in the cooperating team, denoted as m_o, will be updated by

$$m_o^{new} = m_o + \eta \frac{\|m_w - X_i\|}{\max(\|m_w - X_i\|, \|m_o - X_i\|)}(X_i - m_o), \quad (3)$$

where η is a specified positive learning rate. It means that all the cooperative units tend to move toward the point X_i and the learning strength of different seed points is adjusted adaptively based on the distance between the cooperator and the current input. Since we have a factor γ involving in (1) when selecting the winner seed, the nearest seed point to X_i is not always the winner. Therefore, the "max" function in (3) is necessary. We can find that a cooperator will have a full learning rate η as $\|m_o - X_i\| \leq \|m_w - X_i\|$. Otherwise, the learning strength is gradually attenuated when the distance between the cooperator and the current input increases.

The other non-cooperating seed points in the winner's territory, denoted as m_p, will be penalized with a dynamical penalizing rate:

$$m_p^{new} = m_p - \eta \frac{\|m_w - X_i\|}{\|m_p - X_i\|}(X_i - m_p). \quad (4)$$

That is, all the penalized seed points will be moved away from the X_i and the closer the seed point is to the input, the more penalization it will suffer from.

As a whole, at the earlier stage of CPCL learning approach, the penalization mechanism plays a leading role, which leads the initial seed points to drift in the input space to find a more appropriate cluster center. But during the next period, the cooperation is strengthened while the penalization is weakened, this makes all the seed points converge to the corresponding cluster centers gradually.

2.2 The CPCL Algorithm

Based on the previous description, the CPCL algorithm can be given as follows:

Step1: Pre-specify the number k of clusters ($k \geq k^*$), and initialize the k seed points $\{m_1, m_2, ..., m_k\}$. Set $t = 1$, $i = 1$ and $n_j = 1$ with $j = 1, 2, ..., k$, where t and i are used to record the number of epochs and input data, respectively.

Step2: Given an input X_i, calculate $I(j|X_i)$ by (1).

Step3: Determine the winner unit m_w. Let S_w be the set of seed points fallen into the territory of m_w. That is, let $S_w = \emptyset$, and then we span S_w by

$$S_w = S_w \cup \{m_j | \, \|m_w - m_j\| \leq \|m_w - X_i\|, j \neq w\}. \quad (5)$$

Step4: Sort the units in S_w based on the distance between each unit to the winner m_w. We denote these units as: $m'_1, m'_2, ..., m'_s$, with

$$\|m'_1 - m_w\| \leq \|m'_2 - m_w\| \leq \cdots \leq \|m'_s - m_w\|, \quad (6)$$

where $s = |S_w|$.

Step5: Select a subset S_c of S_w to form a cooperating team of m_w, where $S_c = \{m'_1, m'_2, ..., m'_k\}$ with $c = |S_c| = min\{s, t-1\}$. Then update all members in S_c by (3).

Step6: Let $S_p = S_w - S_c$, then, penalize all seed points in S_p by (4).

Step7: Update the winner m_w by

$$m_w^{new} = m_w + \eta \cdot (X_i - m_w). \quad (7)$$

Step8: Update n_w by $n_w^{new} = n_w^{old} + 1$. Let $i = i + 1$, $t = 1 + \lfloor i/N \rfloor$.

The above step 2 to step 8 are iterated for each input until all the seed points converge.

3 Experimental Results

3.1 Experiment 1

To demonstrate the performance of the CPCL algorithm in comparison with the CCCL and S-RPCL, we generated 1,000 data points from a mixture of three 2-dimension Gaussian densities:

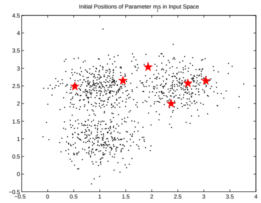
$$\begin{aligned} p(X|\Theta) = 0.3G\left[X|\begin{pmatrix} 1.0 \\ 1.0 \end{pmatrix}, \begin{pmatrix} 0.15, 0.0 \\ 0.0, 0.15 \end{pmatrix}\right] \\ + 0.4G\left[X|\begin{pmatrix} 1.0 \\ 2.5 \end{pmatrix}, \begin{pmatrix} 0.15, 0.0 \\ 0.0, 0.15 \end{pmatrix}\right] \\ + 0.3G\left[X|\begin{pmatrix} 2.5 \\ 2.5 \end{pmatrix}, \begin{pmatrix} 0.15, 0.0 \\ 0.0, 0.15 \end{pmatrix}\right] \end{aligned} . \quad (8)$$

It can be seen from Fig. 2(a) that the data points generated from this mixture densities are overlapped moderately and each cluster is ball-shaped. For each algorithm, the learning rate η was set at 0.001 and the parameter φ in CCCL algorithm was set to 0.5 according to [13]. Six seed points were randomly initialized in the input space and Fig. 2(a) has given out their positions.

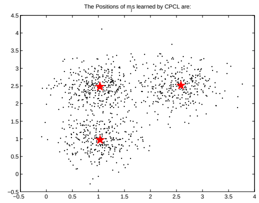
After 200 learning epochs, the positions of seed points obtained by the three algorithms are shown in Fig. 2(b) to Fig. 2(d), respectively. It can be seen that all the three algorithms have identified the true number of clusters successfully. However, the s-RPCL had not located the cluster centers accurately yet. A snapshot of the class centers obtained by S-RPCL is:

$$m_1 = \begin{pmatrix} 1.0180 \\ 0.9303 \end{pmatrix}, m_2 = \begin{pmatrix} 0.9264 \\ 2.5452 \end{pmatrix}, m_3 = \begin{pmatrix} 2.3989 \\ 2.4025 \end{pmatrix}. \quad (9)$$

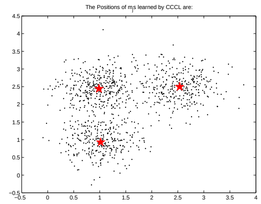
Moreover, Fig. 3 shows the learning curves of m_js via each method and the convergence time of each algorithm is given out in Table 1.It can be seen that the CPCL converges faster than the other two algorithms. This scenario shows the good performance of CPCL in terms of convergence rate and precision.



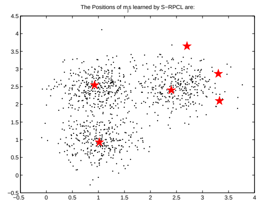
(a)



(b)

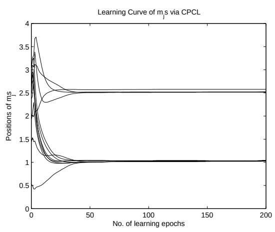


(c)

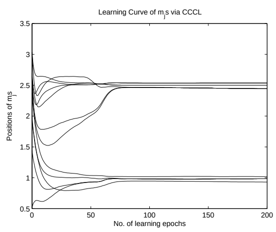


(d)

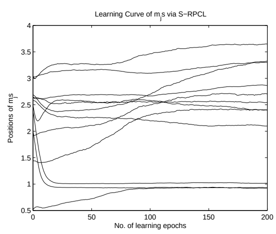
Figure 2. The positions of six seed points marked by "★" in the input space in Experiment 1: (a) the initial random positions, (b) the positions learned via the CPCL, (c) the positions learned via the CCCL, and (d) the positions obtained by the S-RPCL.



(a)



(b)



(c)

Figure 3. The learning curves of six seed points in Experiment 1: (a) the learning curves obtained by CPCL, (b) the learning curves obtained by CCCL, and (c) the learning curves obtained via S-RPCL.

3.2 Experiment 2

In this experiment, we further investigated the performance of CPCL on the mixture clusters that were seriously

Table 1. Convergence Time of Each Algorithm in Experiment 1

Methods	CPCL	CCCL	S-RPCL
Convergence time	4.3679s	6.4311s	9.9803s

overlapped and some cluster was in elliptical shape. Similar to Experiment 1, 1,000 data points were generated from a mixture of three 2-dimension Gaussian densities:

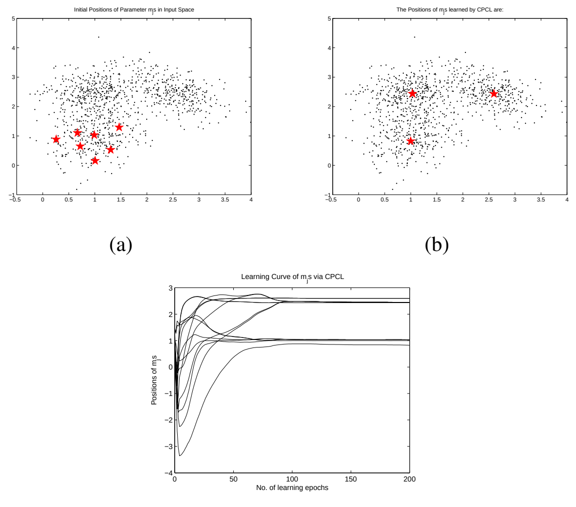
$$\begin{aligned} p(X|\Theta) = 0.3G&\left[X|\begin{pmatrix}1.0\\1.0\end{pmatrix},\begin{pmatrix}0.2,0.05\\0.05,0.3\end{pmatrix}\right]\\ +0.4G&\left[X|\begin{pmatrix}1.0\\2.5\end{pmatrix},\begin{pmatrix}0.2,0.0\\0.0,0.2\end{pmatrix}\right]\\ +0.3G&\left[X|\begin{pmatrix}2.5\\2.5\end{pmatrix},\begin{pmatrix}0.2,-0.1\\-0.1,0.2\end{pmatrix}\right] \end{aligned}. \quad (10)$$

Furthermore, in order to verify that the CPCL algorithm is insensitive to the initial positions of seed points, we forcibly generated seven highly centralized seed points, which were located in one cluster region. Fig. 4(a) has given out the positions of input data and initial seed points. Similarly, the number of learning epochs was set to 200. As shown in Fig. 4(b), the seed points learned by CPCL had been converged accurately to the corresponding cluster centers. To further demonstrate the efficiency of CPCL, Fig. 4(c) shows the learning curves of m_js, which converged during the first 100 epochs. These attractive results indicate that the penalization mechanism in CPCL can supplies the initial seed points with sufficient opportunity to wander in the clustering space even though they are centralized seriously. Moreover, it also can be seen from this experiment that, although the concept of winner's territory in CPCL is based on Euclidean distance only, this new algorithm can work well on not only ball-shaped data clusters but also elliptical ones.

3.3 Experiment 3

The CPCL algorithm has shown its good performance under three clusters during the previous experiments. In this experiment, we will further investigate its learning capability when the true number of clusters is much larger. 1,000 data points were generated from a mixture of 10 2-dimension Gaussian density distributions which were moderately overlapped. And the proportions of the mixture components are heterogeneous.

We randomly initialized 20 seed points in the input space as shown in Fig. 5(a). After 400 learning epochs, the stable positions of seed points learned by CPCL are shown in Fig. 5(b). It is obvious that the true number of clusters has been identified and all the seed points have been converged

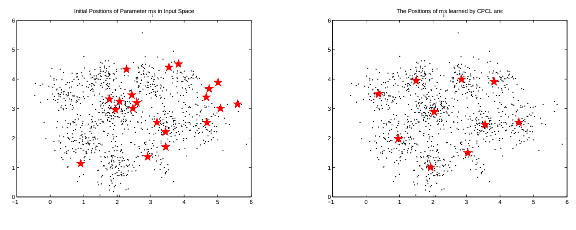


(a) (b)

(c)

Figure 4. The results of Experiment 2: (a) the initial positions of seven seed points marked by "★" in the input space, (b) the positions of seed points learned by CPCL, and (c)the learning curves of seven seed points.

to the exact 10 cluster centers. So, we can see that CPCL algorithm has the robust performance even if the values of k^* and k both become large.



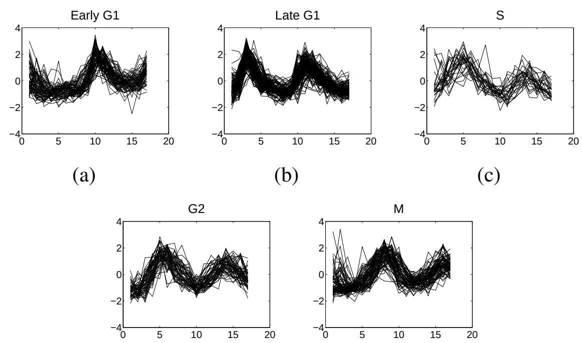
(a) (b)

Figure 5. In this figure, (a) shows the initial positions of 20 seed points marked by "★" in the input space, and (b) shows the positions of seed points learned by CPCL.

3.4 Experiment 4

The previous experiments have showed the performance of CPCL under the synthetic data clustering. In this experiment we will investigate its efficiency on the real-world microarray data–the yeast cell cycle data published by Cho et al.[15]. The original data contained the expression profiles of 6220 genes over 17 time points taken at 10 minute intervals, covering nearly two cell cycles. The data set we used was comprised of 384 genes which had expression levels peaking at different time points corresponding to the five phases of the cell cycle. Hence, it is expected that each of the 384 genes can be assigned to one of the five clusters [15]. This subset of data is available at *http://www.cs.washington.edu/homes/kayee/model* and the standardized data was adopted in this experiment.

We assumed that the true number of clusters was unknown and arbitrarily initialized 10 seed points in the running of CPCL algorithm. For further analysis, we compared the proposed approach with other three methods: the EM algorithm based on BIC (called *Method I* hereinafter) [17], the supervised clustering method (Method II) [18] and the support vector machines algorithm (Method III) [16]. After the implement of clustering, each gene had four possible outcomes: false positive (FP), false negative (FN), true positive (TP) and true negative (TN). And the total error rate can be defined as FP+FN [18]. The clustering results of the four methods are given out in Table 2, where the results of method I-III are obtained from [18]. Furthermore, the total error rates of different algorithms are summarized in Table 3. We can see that, in terms of the small error rates, our method has a good performance in this experiment. In addition, the five groups of genes whose expression level peak at different phases of the cell cycle formed by CPCL algorithm is shown in Fig. 6. It can be observed that the genes with similar attributes have been clustered together, which indicated that the proposed method is indeed effective for the cluster analysis of gene expression data.



(a) (b) (c)

(d) (e)

Figure 6. The five groups of genes classified by the CPCL algorithm .

4 Concluding Remarks

In this paper, we have presented a novel competitive learning algorithm named Cooperative and Penalized Competitive Learning (CPCL), which performs the competition

Table 2. Comparison of the Clustering Results of the Four Methods on the Microarray Data

Cell division phase	Methods	FP	FN	TP	TN
Early G1 (67 genes)	CPCL	24	18	49	293
	Method I	50	12	55	267
	Method II	21	21	46	296
	Method III	38	10	57	279
Late G1 (135 genes)	CPCL	38	22	113	211
	Method I	28	40	95	221
	Method II	24	35	100	225
	Method III	43	10	125	206
S (75 genes)	CPCL	11	58	17	298
	Method I	33	49	26	276
	Method II	37	36	39	272
	Method III	72	18	57	237
G2 (52 genes)	CPCL	24	22	30	308
	Method I	28	41	11	304
	Method II	18	29	23	314
	Method III	46	5	47	286
M (55 genes)	CPCL	26	3	52	303
	Method I	38	42	13	291
	Method II	19	8	47	310
	Method III	47	2	53	283

Table 3. The Total Error Rates of the Four Methods on the Microarray Data

Methods	FP	FN	FP+FN
CPCL	123	123	246
Method I	177	184	361
Method II	119	129	248
Method III	246	45	291

with the two different kinds of mechanisms simultaneously: cooperation and penalization. On the one hand, the cooperation mechanism enables the closest seed points to update together and gradually converge to the corresponding cluster centers, which gives the algorithm good convergence speed and high precision. On the other hand, the penalization mechanism provides the other seed points with the opportunity to wander in the clustering space, which enables it to perform the clustering problem with the robustness against the initialization of the seed points and the overlap of the data clusters. Experimental results on both synthetic data and the yeast cell cycle microarray data have shown the outstanding performance of the proposed approach.

References

[1] U. Fayyad, G. Piatetsky-Shpiro, P. Smyth, and R. Uthurusamy. *Advances in Knowledge Discovery and Data Mining*. MIT Press, 1996.

[2] T. Uchiyama and M.A. Arib. Color Image Segmentation Using Competitive Learning. *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, no. 12, pp. 1197-1206, Dec. 1994.

[3] Y. Lu, S.Y. Lu, F. Fotouhi, Y.P. Deng, and S. Brown. Incremental Genetic K-means Algorithm and Its Application in Gene Expression Data Analysis. *BMC Bioinformatics*, vol. 5, no. 172, Oct. 2004.

[4] J.B. MacQueen. Some Methods for Classification and Analysis of Multivariate Observations. *Proc. 5-th Berkeley Symp. Mathematical Statistics and Probability*, vol. 1, pp. 281-297, Berkeley, Calif., USA, 1967.

[5] Y.M. Cheung. K*-means: A New Generalized K-Means Clustering Algorithm. *Pattern Recognition Letters*, vol. 24, no. 15, pp. 2883-2893, 2003.

[6] H. Akaike. Information Theory and an Extension of the Maximum Likelihood Principle. *Proc. Second International Symp. Information Theory*, pp. 267-281, 1973.

[7] G. Schwarz. Estimating the Dimension of a Model. *The Annals of Statistics*, vol. 6, no. 2, pp. 461-464, 1978.

[8] C.S. Wallace and D.L. Dowe. Minimum Message Length and Kolmogorov Complexity. *The Computer Journal*, vol. 42, no. 4, pp. 270-283, 1999.

[9] L. Xu, A. Krzyzak, and E. Oja. Rival Penalized Competitive Learning for Clustering Analysis, RBF Net, and Curve Detection. *IEEE Trans. Neural Networks*, vol. 4, no. 4, pp. 636-648, Jul. 1993.

[10] Y.M. Cheung. Rival Penalization Controlled Competitive Learning for Data Clustering with Unknown Cluster Number. *Proc. 9th Int'l Conf. Neural Information Processing*, pp. 18-22, Singapore, Nov. 2002.

[11] Y.M. Cheung. Maximum Weighted Likelihood via Rival Penalized EM for Density Mixture Clustering with Automatic Model Selection. *IEEE Trans. Knowledge and Data Engineering*, vol. 17, no. 6, pp. 750-761, Jun. 2005.

[12] Y.M. Cheung. A Competitive and Cooperative Learning Approach to Robust Data Clustering. *Proc. IASTED Int'l Conf. Neural Networks and Computational Intelligence*, pp. 131-136, Grindelwald, Switzerland, 2004.

[13] T. Li, W.J. Pei, S.P. Wang, and Y.M. Cheung. Cooperation Controlled Competitive Learning Approach for Data Clustering. *Proc. Int'l Conf. Computational Intelligence and Security*, pp. 24-29, 2008.

[14] S.C. Ahalt, A.K. Krishnamurty, P. Chen, and D.E. Melton. Competitive Learning Algorithms for Vector Quantization. *Neural Networks*, vol. 3, no. 3, pp. 277-291, 1990.

[15] R.J. Cho, M.J. Campbell, E.A. Winzeler, L. Steinmetz, A. Conway, L. Wodicka, T.G. Wolfsberg, A.E. Gabrielian, D. Landsman, D.J. Lockhart, and R.W. Davis. A Genome-Wide Transcriptional Analysis of the Mitotic Cell Cycle. *Molecular Cell*, vol. 2, pp. 65-73, 1998.

[16] M.P.S. Brown, W.N. Grundy, D. Lin, N. Cristianini, C.W. Sugnet, T.S. Furey, M. Ares, and D. Haussler. Knowledge-Based Analysis of Micro- Array Gene Expression Data by Using Support Vector Machines. *Proc. Natl Academy of Sciences of the USA*, vol. 97, pp. 262-267, 2000.

[17] K.Y. Yeung, C. Fraley, A. Murua, A.E. Raftery, and W.L. Ruzzo. Model- Based Clustering and Data Transformations for Gene Expression Data. *Bioinformatics*, vol. 17, pp. 977-987, 2001.

[18] Y. Qu and S. Xu. Supervised Cluster Analysis for Microarray Data Based on Multivariate Gaussian Mixture. *Bioinformatics*, vol. 20, pp. 1905-1913, 2004.

[19] T. Bengtsson and J.E. Cavanaugh. An Improved Akaike Information Criterion for State-space Model Selection. *Computational Statistics & Data Analysis*, vol. 50, no. 10, pp. 2635-2654, Jun. 2006.

[20] N. Bouguila and D. Ziou. Unsupervised Selection of a Finite Dirichlet Mixture Model: An MML-based Approach. *IEEE Trans. Knowledge and Data Engineering*, vol. 18, no. 8, pp. 993-1009, Aug. 2006.

[21] S. Marsland, C.J. Twining, and C.J. Taylor. A Minimum Description Length Objective Function for Groupwise Non-rigid Image Registration. *Image and Vision Computing*, vol. 26, no. 3, pp. 333-346, Mar. 2008.

[22] Z. Lu and H.H.S. Ip. Generalized Competitive Learning of Gaussian Mixture Models. *IEEE Trans. Systems, Man, and Cybernetics*, vol. 39, no. 4, pp. 901-909, Aug. 2009.