A Competitive and Cooperative Learning Approach to Robust Data Clustering*

Yiu-ming Cheung
Department or Computer Science
Hong Kong Baptist University, Hong Kong
E-mail: ymc@comp.hkbu.edu.hk

ABSTRACT
This paper presents a new semi-competitive learning paradigm named *Competitive and Cooperative Learning* (CCL), in which seed points not only compete each other for updating to adapt to an input each time, but also dynamically cooperate to achieve the learning task. This competitive and cooperative mechanism can automatically merge those extra seed points, meanwhile making the seed points gradually converge to the corresponding cluster centers. Consequently, CCL can perform a robust clustering analysis without prior knowing the exact cluster number so long as the number of seed points is not less than the true one. The experiments have successfully shown its outstanding performance on data clustering.

KEY WORDS
Cooperative and Competitive Learning, Semi-Competitive Learning, Rival Penalization Controlled Competitive Learning, Clustering Analysis, Cluster Number.

1 Introduction

As an efficient tool of clustering analysis, competitive learning has been applied to a wide variety of research problems, such as data compression [1], signal processing [2], neural networks [3, 4], and so forth. In the literature, k-means [5] is a typical competitive-learning based clustering algorithm, in which k pre-assigned seed points (also called *units* interchangeably) compete each other, and only the winner is updated to adapt to an input each time. Apart from a number of successful examples, some experiments have also found that the k-means has two major drawbacks as pointed out in [6]:

1. There is the *dead-unit* problem. That is, if some units are initialized far away from the input data set in comparison with the other units, they then immediately become the dead unit without any winning chance in the forthcoming competitive learning process;

2. It needs to pre-determine the cluster number. When k equals to the true cluster number k^*, the k-means algorithm can correctly find out the cluster centers. Otherwise, some of seed points will not locate at the centers of the corresponding clusters. Instead, they are either at some boundary points among different clusters or at points biased from some cluster centers. Consequently, it will lead to an incorrect clustering result.

To circumvent the first problem, an extension of k-means named Frequency Sensitive Competitive Learning (FSCL) algorithm [7] was proposed, in which the winning chance of a seed point is penalized along with the increase of past winning frequency, and vice versa. Although FSCL can almost always successfully assign one or more seed points to each cluster without dead-unit problem, it meets the same second problem as the k-means.

In the past decade, some competitive learning algorithms have been proposed to perform clustering without knowing the exact cluster number. For example, one variant of k-means named *incremental clustering* is to gradually increase the number k of clusters under the control of a threshold value, which is unfortunately hard to be decided as well. Another example is Probabilistic Validation (PV) approach [8] that performs clustering analysis by projecting the high-dimension inputs into one dimension via maximizing the projection indices. It has been shown that the PV can find out the correct number of clusters with a high probability. However, this algorithm is essentially applicable to a small number of clusters only, and requests the clusters to be well-separated with the overlaps ignorable. Otherwise, its two-level clustering validation procedure will be quite tedious, and the probability of finding the correct number of clusters decreases. In the literature, another algorithm developed from FSCL is Rival Penalized Competitive Learning (RPCL) [9] that for each input, not only the winner of the seed points is updated to adapt to the input, but also its rival is de-learned by a smaller constant learning rate (also called *de-learning rate* hereafter). Empirical studies have shown that RPCL can indeed select the correct cluster number automatically by driving extra seed points far away from the input data set. However, some experiments have also found that its performance is sensitive to the selection of the de-learning rate. If the rate is selected too small, there is no enough penalizing forces to drive extra seed points away from the input data set. Conversely, if the rate is too large, the desired seed points will be forced

*The work described in this paper was supported by Faculty Research Grant of Hong Kong Baptist University with Project Number: FRG/03-04/I-12.

to drift far away from the input set as well as the extra ones. To circumvent this difficulty, we have therefore proposed Rival Penalization Controlled Competitive Learning (RPCCL) algorithm [10] that dynamically adjusts the rival-penalized strength based on the distance between the winner and the rival relative to the current input. Compared to the RPCL, this algorithm always fixes the de-learning rate at the same value as the learning rate without requesting further selection. However, as well as RPCL, RPCCL often drives those extra seed points far away from the input data set without convergence, although the desired seed points will stably locate at the corresponding cluster centers.

In this paper, we will present a new semi-competitive learning paradigm named *Competitive and Cooperative Learning* (CCL), in which seed points not only compete each other for updating to adapt to an input each time, but also the winner will dynamically select several nearest competitors to form a cooperative team to adapt to the input together. As a whole, the seed points locating in the same cluster will have more opportunity to cooperate each other than competition to achieve the learning task, and vice versa. Subsequently, it can lead to those seed points automatically merge and gradually converge to the corresponding cluster centers with uniformly sharing the same winning probability. That is, CCL can perform a robust clustering analysis without prior knowing the exact cluster number so long as the number of seed points is not less than the true one. The simulation results have demonstrated the outstanding performance of CCL on data clustering.

2 Overview of k-means and FSCL Algorithms

2.1 k-means Algorithm

Suppose there are k seed points in the input space, denoted as $\mathbf{w}_1$, $\mathbf{w}_2$, ..., $\mathbf{w}_k$, respectively. The k-means [5] aims to partition N inputs: $\mathbf{x}_1$, $\mathbf{x}_2$, ..., $\mathbf{x}_N$, into k^* true clusters by repeatedly adjusting those $\mathbf{w}_j$s such that the following distortion error:

$$E(\mathbf{X}, \mathbf{W}) = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{k} \|\mathbf{x}_i - \mathbf{w}_j\|^2 \qquad (1)$$

is minimized, where $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$, and $\mathbf{W} = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_k\}$. An adaptive version of k-means learns k seed points by the following steps:

Step 1. Pre-specify the number k of clusters, and initialize the seed points $\{\mathbf{w}_j\}_{j=1}^k$.

Step 2. Given an input $\mathbf{x}_i$, calculate the indicator function $I(j|\mathbf{x}_i)$ by

$$I(j|\mathbf{x}_t) = \begin{cases} 1, & \text{if } j = \arg\min_{1 \le r \le k} \|\mathbf{x}_i - \mathbf{w}_r\|^2 \\ 0, & \text{otherwise.} \end{cases} \qquad (2)$$

Step 3. Update the winning seed point $\mathbf{w}_c$, i.e., $I(c|\mathbf{x}_i) = 1$, by

$$\mathbf{w}_c^{\text{new}} = \mathbf{w}_c^{\text{old}} + \eta(\mathbf{x}_i - \mathbf{w}_c^{\text{old}}), \qquad (3)$$

meanwhile other seed points are unchanged, where η is a small positive learning rate.

The above **Step 2** and **Step 3** are iterated for each input until all seed points converge. As shown in Eq.(2), k-means determines the winner among k seed points exclusively based on the Euclidean distance between the current input and each seed point. Under the circumstances, if a seed point is initialized far away a little from the input date set compared to the other seed points, this point will then be isolated, and become a so-called *dead* unit because it has no winning chance in the forthcoming competitive learning process.

2.2 Frequency Sensitive Competitive Learning (FSCL)

To deal with the *dead unit* problem in the k-means, Ahalt et al. [7] proposed a frequency sensitive competitive learning approach, in which, apart from considering the distance of $\mathbf{w}_i$s to the input, an implicit penalty is also given to those seed points that have high relative winning frequency in the past competitions. Subsequently, given an input $\mathbf{x}_i$ each time, instead of Eq.(2), FSCL determines the winner among k seed points by

$$I(j|\mathbf{x}_t) = \begin{cases} 1, & \text{if } j = \arg\min_{1 \le r \le k} \gamma_r \|\mathbf{x}_i - \mathbf{w}_r\|^2 \\ 0, & \text{otherwise} \end{cases} \qquad (4)$$

with the relative winning frequency γ_r of $\mathbf{w}_r$ defined as

$$\gamma_r = \frac{n_r}{\sum_{j=1}^k n_j}, \qquad (5)$$

where n_r is the winning times of $\mathbf{w}_r$ in the past. After selecting out the winner, FSCL then updates the winner only by Eq.(3) in the same way as k-means, and meanwhile adjusting the corresponding n_c with

$$n_c^{\text{new}} = n_c^{\text{old}} + 1. \qquad (6)$$

The FSCL algorithm can be summarized as follows:

Step 1. Pre-specify the number k of clusters, initialize the seed points $\{\mathbf{w}_j\}_{j=1}^k$, and set $n_j = 1$ with $j = 1, 2, \ldots, k$.

Step 2. Given an input $\mathbf{x}_i$, calculate $I(j|\mathbf{x}_i)$ by Eq.(4);

Step 3. Update the winning seed point $\mathbf{w}_c$ and its n_c only by Eq.(3) and Eq.(6), respectively.

FSCL can almost always successfully distribute k seed points into the input data set without dead-unit problem. However, it needs to pre-assign the number k of clusters. If k is not equal to the true k^*, FSCL will lead to an incorrect clustering result similar to the k-means.

3 Competitive and Cooperative Learning (CCL) Approach

To circumvent the difficulty of deciding the cluster number in advance, we propose the CCL approach, which need not decide the exact cluster number k^* before clustering. Instead, as long as the assigned number k of seed points is not less than k^*, CCL enables all seed points to gradually converge into the corresponding cluster centers with some seed points staying at the same cluster centers without repelling each other.

The basic idea of the CCL is that the k seed points not only compete each other for updating to adapt to an input each time, but also the winner will dynamically select several nearest competitors to form a cooperative team to adapt to the input together. That is, the seed points locating in the same cluster will have more opportunity to cooperate each other than competition to achieve the learning task, and vice versa. Subsequently, on an average the learning of these seed points are *independently* conducted towards the corresponding cluster center with uniformly sharing the data points in the cluster, i.e., a seed point regards its nearest competitors as the transparent ones without any competition. As a result, CCL enables to perform clustering successfully as $k \geq k^*$.

Actually, such a competitive and cooperative learning mechanism is also more consistent with the real social scenario, in which it is generally believed that a competitive scheme with some cooperations in a team group can not only motivate the aggressiveness of each member, but also lead to a more harmonic working environment in the group. Consequently, it is easier to achieve the designated goal in comparison with the pure competitive or penalized competitive learning schemes.

Given an input $\mathbf{x}_i$, the CCL procedure consists of three separate steps. Firstly, it determines a leader which is definitely a winner among k seed points. Secondly, this leader will form a cooperating team, in which all seed points are the winners. In this paper, such a cooperating team is determined on the basis of the distance between the winner leader and the other seed points relative to the distance between the leader and the current input. That is, as shown in Figure 1, the leader $\mathbf{w}_c$ regards those seed points fallen into the circle centered at $\mathbf{w}_c$ with the radius $\|\mathbf{w}_c - \mathbf{x}\|$ as the cooperating members, like $\mathbf{w}_{c1}$ and $\mathbf{w}_{c2}$ in Figure 1. Thirdly, all the winners in the team are updated to adapt to the input. In the CCL, the extra award for the winner leader is that CCL updates the winning times n_c of the leader only. Subsequently, the CCL algorithm can be given as follows:

Step 1. Pre-specify the number k of clusters with $k \geq k^*$, initialize the seed points $\{\mathbf{w}_j\}_{j=1}^k$, and set $n_j = 1$ with $j = 1, 2, \ldots, k$.

Step 2. Given an input $\mathbf{x}_i$, calculate $I(j|\mathbf{x}_i)$ by Eq.(4).

Step 3. Let the cooperating set $C = \{\mathbf{w}_c\}$. We then span C by

$$C = C \cup \{\mathbf{w}_j \mid \|\mathbf{w}_c - \mathbf{w}_j\| \leq \|\mathbf{w}_c - \mathbf{x}_i\|\}. \quad (7)$$

That is, all of those seed points fallen into the circle centered at $\mathbf{w}_c$ with the radius $\|\mathbf{w}_c - \mathbf{x}_i\|$ are the winners as well as $\mathbf{w}_c$, but the others outside the circle are not.

Step 4. Update all members in C by

$$\mathbf{w}_u^{\text{new}} = \mathbf{w}_u^{\text{old}} + \eta(\mathbf{x}_i - \mathbf{w}_u^{\text{old}}), \quad (8)$$

where $\mathbf{w}_u \in C$. Furthermore, we here only update n_c by Eq.(6) without uniformly distributing the contribution of this winning to all other n_js. The benefit is that we can finally use γ_js to estimate the proportion that the data from each cluster.

The above **Step 2** and **Step 4** are repeatedly iterated for each input until all seed points converge.

Before closing this section, please note that CCL enables each extra seed point to finally locate at one of cluster centers. Hence, we can determine the exact cluster number by counting the number of those seed points stayed at different positions.

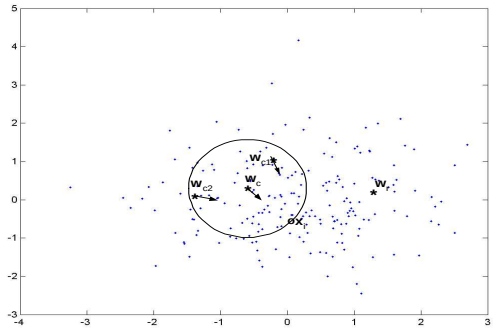


Figure 1. The positions of four seed points marked by '*' in a cluster, in which $\mathbf{w}_c$ is the winner leader, which chooses $\mathbf{w}_{c1}$ and $\mathbf{w}_{c2}$ fallen into the circle centered at $\mathbf{w}_c$ with the radius $\|\mathbf{w}_c - \mathbf{x}_i\|$ as the cooperating members, and together to adapt to the input $\mathbf{x}_i$ marked by 'o'. In contrast, $\mathbf{w}_r$ is outside the circle. Hence, $\mathbf{w}_r$ is a loser in the competition.

4 Experimental Simulations

4.1 Experiment 1

To demonstrate the performance of CCL, we randomly generated $2,000$ data points from a mixture of three 2-dimension Gaussians:

$$\begin{aligned} p(\mathbf{x}; \boldsymbol{\Theta}) &= 0.3G(\mathbf{x}|\boldsymbol{\mu}_1, 0.1\mathbf{I}) + 0.4G(\mathbf{x}|\boldsymbol{\mu}_2, 0.1\mathbf{I}) \\ &\quad + 0.3G(\mathbf{x}|\boldsymbol{\mu}_3, 0.1\mathbf{I}), \end{aligned} \quad (9)$$

with

$$\boldsymbol{\mu}_1 = [1,1]^T, \quad \boldsymbol{\mu}_2 = [1,5]^T, \quad \text{and} \quad \boldsymbol{\mu}_3 = [5,5]^T, \tag{10}$$

where $\mathbf{I}$ is a 2×2 identity matrix, T is a transpose operation of a matrix, and $G(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the Gaussian probability density function of $\mathbf{x}$ with the mean $\boldsymbol{\mu}$ and co-variance $\boldsymbol{\Sigma}$. Furthermore, we set the learning rate $\eta = 0.001$, and randomly initialized the positions of five seed points in the input space, as shown in Figure 2(a). Figure 2(b) shows their learning curve. It can be seen that all seed points have been converged after 15 learning epochs. A snapshot value of convergent seed points are:

$$\begin{aligned} \mathbf{w}_1 &= [1.0191, 0.9907]^T, \quad \mathbf{w}_2 = [0.9808, 4.9944]^T \\ \mathbf{w}_3 &= [0.9808, 4.9944]^T, \quad \mathbf{w}_4 = [5.0310, 4.9913]^T \\ \mathbf{w}_5 &= [0.9808, 4.9944]^T, \end{aligned} \tag{11}$$

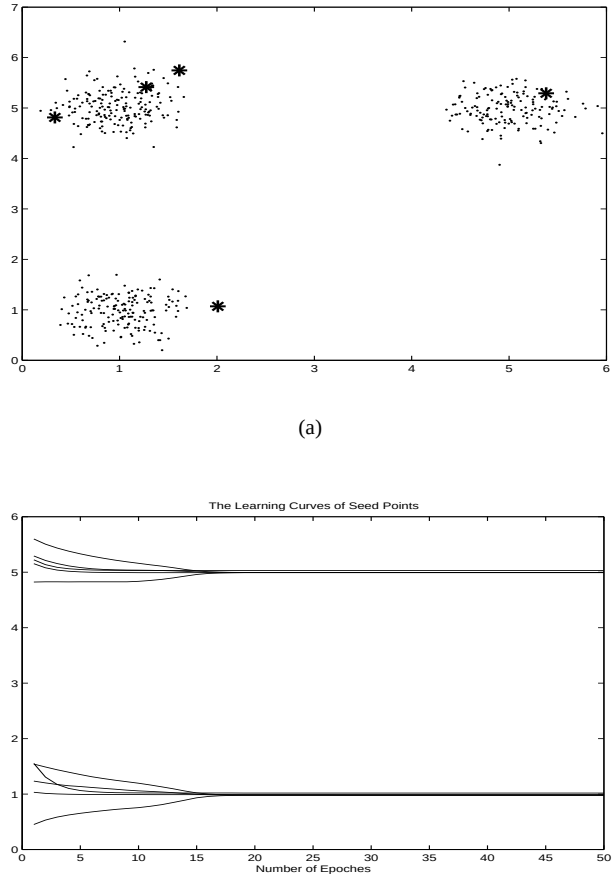
with their relative winning frequency γ_js being:

$$\begin{aligned} \gamma_1 &= 0.3090, \quad \gamma_2 = 0.1282, \quad \gamma_3 = 0.1282 \\ \gamma_4 &= 0.3064, \quad \gamma_5 = 0.1282, \end{aligned} \tag{12}$$

in which we found that $\mathbf{w}_2$, $\mathbf{w}_3$ and $\mathbf{w}_5$ converged to the same cluster center with uniformly sharing the same winning chance, meanwhile $\mathbf{w}_1$ and $\mathbf{w}_4$ converged to the other two cluster centers, respectively. That is, the exact three data clusters have been automatically detected. Furthermore, the summation of γ_2, γ_3 and γ_5 is 0.3846, which actually is an estimate of the prior probability that data from $G(\mathbf{x}|\boldsymbol{\mu}_2, 0.1\mathbf{I})$, whereas γ_1 and γ_4 are the estimate of the prior probabilities of $G(\mathbf{x}|\boldsymbol{\mu}_1, 0.1\mathbf{I})$ and $G(\mathbf{x}|\boldsymbol{\mu}_3, 0.1\mathbf{I})$, respectively. Figure 3(a) shows the trajectory of these seed points' learning, and Figure 3(b) shows their final positions. It can be seen that CCL has successfully performed clustering under the environment that there are two extra seed points. Furthermore, we also investigated the CCL performance with six and seven seed points, respectively. As shown in Figure 4, the experimental result is the same as the previous case.

4.2 Experiment 2

In this experiment, we further investigated the CCL performance on those data with the moderate overlap, as shown in Figure 5(a). Under the same experimental environment as Experiment 1, we performed the CCL with the five, six, and seven seed points, respectively. Figure 5(b) shows the final distribution of five seed points in the input data set. Once again, it can be seen that all seed points have been converged to the corresponding cluster centers with some locating at the same position. The results from the other two cases are also the same. That is, CCL has performed a correct clustering with automatically detecting the exact cluster number $k^* = 3$.



(a)

(b)

Figure 2. (a) The initial positions of five seed points marked by '*' in the input space, and (b) the learning curve of five seed points.

4.3 Experiment 3

The CCL performs clustering without considering the covariance information of each cluster. In this experiment, we further investigated the robustness of its performance on elliptical data clusters, rather than the ball-shape ones in Experiment 1 and 2. Again, we investigated the three cases: $k = 5, 6$, and 7, respectively, whose results were all the same. Figure 6 shows the results of the five seed points. It can be seen that CCL has successfully performed the clustering in the same way as the previous experiments.

5 Conclusion

This paper has presented a *Competitive and Cooperative Learning* (CCL) algorithm, which provides a new way for data clustering without prior knowing exact cluster number. CCL is a semi-competitive learning, in which seed points

not only compete each other for updating to adapt to an input each time, but also some neighbor seed points dynamically cooperate to adapt to the input together. This cooperative and competitive scheme can lead to those seed points gradually merging and converging into the cluster center with sharing the same winning probability. Consequently, the CCL can perform a robust clustering analysis with automatically determining the correct cluster number as long as $k \geq k^*$. The experiments have successfully shown its outstanding performance on data clustering.

References

[1] W.P. Li and Y.P. Zhang, Vector-based signal processing and quantization for image and video compression, *Proceedings of the IEEE*, 83(2), 1995, 317–335.

[2] A. Gersho and R.M. Gray, Vector quantization and signal processing, Kluwer Academic Publisher, Boston, 1992.

[3] T. Kohonen, Self-organized formation of topologically correct feature maps, *Biological Cybernetics*, 43, 1982, 59–69.

[4] J.E. Moody and C. Darken, Fast learning in networks of locally-tuned processing units, *Neural Computation*, 1, 1989, 281–294.

[5] J.B. MacQueen, Some methods for classification and analysis of multivariate observations, *Proceedings of* 5^{nd} *Berkeley Symposium on Mathematical Statistics and Probability*, 1, Berkeley, Calif.: University of California Press, 1967, 281–297.

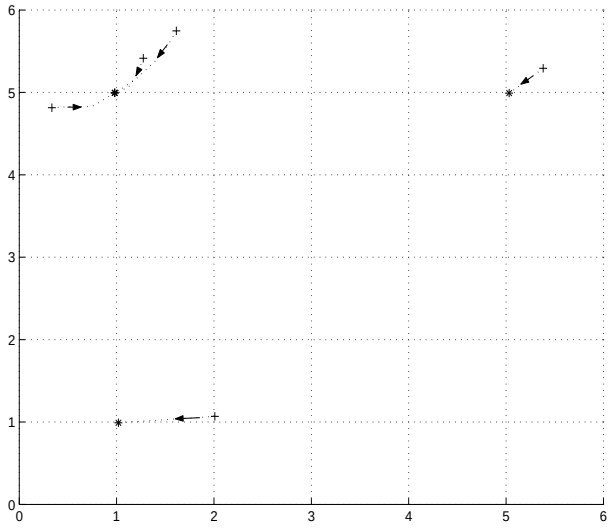
[6] Y.M. Cheung, k*-means: a new generalized k-means clustering algorithm, *Pattern Recognition Letters*, 24, 2003, 2883–2893.

[7] S.C. Ahalt, A.K. Krishnamurty, P. Chen and D.E. Melton, Competitive learning algorithms for vector quantization, *Neural Networks*, 3, 1990, 277–291.

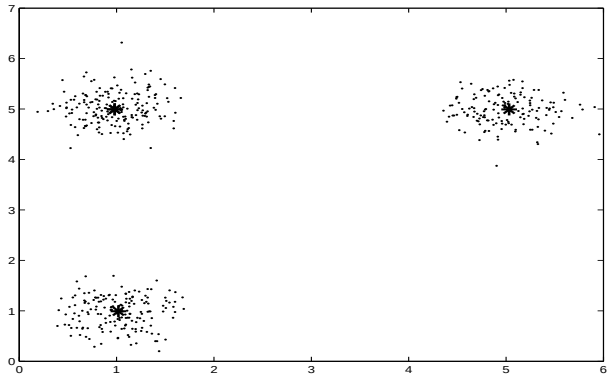
[8] M. Har-even and V.L. Brailovsky, Probabilistic validation approach for clustering, *Pattern Recognition Letters*, 16, 1995, 1189–1196.

[9] L. Xu, A. Krzyżak and E. Oja, Rival penalized competitive learning for clustering analysis, RBF net, and curve detection, *IEEE Transaction on Neural Networks*, Vol. 4, pp. 636–648, 1993.

[10] Y.M. Cheung, Rival penalization controlled competitive learning for data clustering with unknown cluster number, *Proceedings of 9th International Conference on Neural Information Processing* (Paper ID: 1983 in CD-ROM Proceeding) , Singapore, November 18-22, 2002.

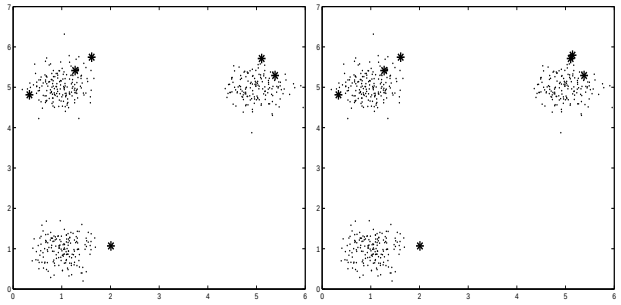


(a)

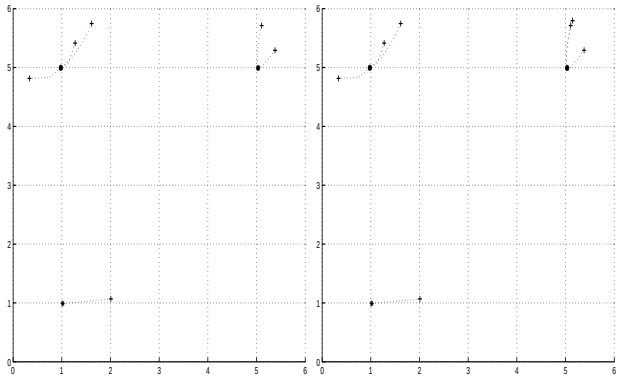


(b)

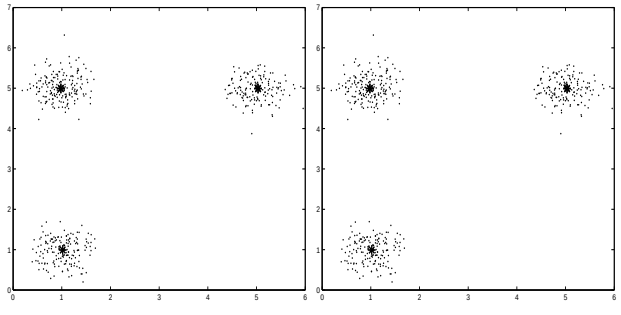
Figure 3. (a) The learning trajectory of five seed points, in which and hereafter figures '+' marks the initial positions of seed points, and '*' marks the final positions; (b) The final distribution of five seed points.



(a) (b)

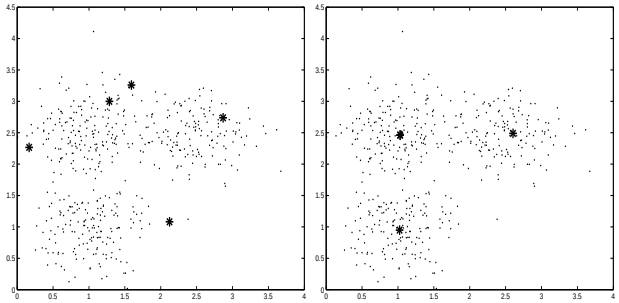


(c) (d)



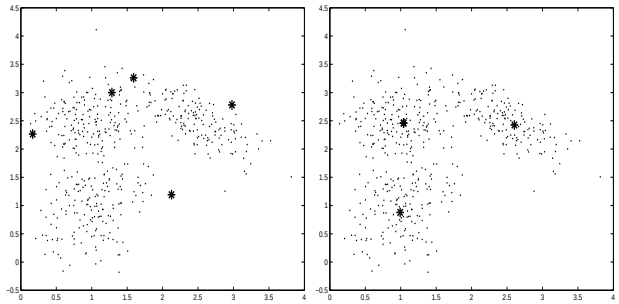
(e) (f)

Figure 4. Sub-figure (a)(c)(e) show the initial position, learning trajectory, and final positions of six seed points respectively, whereas Sub-figure (b)(d)(f) show the situation of seven seed points.



(a) (b)

Figure 5. (a) The initial positions of five seed points marked by '*' in the input space, and (b) the final positions of five seed points.



(a) (b)

Figure 6. (a) The initial positions of five seed points marked by '*' in the input space, and (b) the final positions of five seed points.