Appendix for Maximum Weighted Likelihood via Rival Penalized EM for Density Mixture Clustering with Automatic Model Selection

Yiu-ming Cheung, *Member*, *IEEE*

1 Experimental Simulation

1.1 Experiment 1

To demonstrate the performance of the RPEM, we generated 1,000 synthetic data points from a mixture of three bivariate Gaussian densities:

$$\begin{aligned} p(\mathbf{x}|\mathbf{\Theta}^*) = {} & 0.3G\left[\mathbf{x}|\begin{pmatrix}1\\1\end{pmatrix},\begin{pmatrix}0.10, & 0.05\\ 0.05, & 0.20\end{pmatrix}\right] \\ & + 0.4G\left[\mathbf{x}|\begin{pmatrix}1.0\\5.0\end{pmatrix},\begin{pmatrix}0.10, & 0.0\\ 0.0, & 0.10\end{pmatrix}\right] \\ & + 0.3G\left[\mathbf{x}|\begin{pmatrix}5.0\\5.0\end{pmatrix},\begin{pmatrix}0.1, & -0.05\\ -0.05, & 0.1\end{pmatrix}\right]. \end{aligned} \tag{1}$$

Supposing k is equal to the true mixture number $k^* = 3$, we randomly located three seed points $\mathbf{m}_1$, $\mathbf{m}_2$, and $\mathbf{m}_3$ in the input space as shown in Fig. 2a, where the data constitute three well-separated clusters. Moreover, we initialized each of the $\mathbf{\Sigma}_j$s to be an identity matrix, and all β_js to be zero, i.e., we initialized $\alpha_1 = \alpha_2 = \alpha_3 = \frac{1}{3}$. Also, we set the learning rates $\eta = 0.001$ and $\eta_\beta = 0.0001$.

We performed the learning of RPEM and showed the Q value of (26) over the epochs in Fig. 2b. It can be seen that the Q value has converged after 40 epochs. Fig. 2a shows the positions of three converged seed points, which are all stably located at the corresponding cluster centers. A snapshot of the converged parameter values is:

$$\begin{aligned} &\alpha_1 = 0.3147, \quad \mathbf{m}_1 = \begin{pmatrix}1.0089\\0.9739\end{pmatrix}, \quad \mathbf{\Sigma}_1 = \begin{pmatrix}0.0986 & 0.0468\\ 0.0468 & 0.2001\end{pmatrix}, \\ &\alpha_2 = 0.3178, \quad \mathbf{m}_2 = \begin{pmatrix}5.0159\\5.0060\end{pmatrix}, \quad \mathbf{\Sigma}_2 = \begin{pmatrix}0.1127 & -0.0581\\ -0.0581 & 0.1128\end{pmatrix}, \\ &\alpha_3 = 0.3675, \quad \mathbf{m}_3 = \begin{pmatrix}0.9759\\4.9761\end{pmatrix}, \quad \mathbf{\Sigma}_3 = \begin{pmatrix}0.0938 & 0.0019\\ 0.0019 & 0.0928\end{pmatrix}. \end{aligned} \tag{2}$$

It can be seen that the RPEM has given out the well estimate of the true parameters with a permutation of subscript indices between 2 and 3. For comparison, we also performed the EM algorithm under the same experimental environment. We found the EM also worked well in this case with the similar convergent rate as the RPEM. Fig. 3b shows that the EM has successfully located the three seed points in the corresponding clusters.

In the above experiment, we have assumed that the number k of seed points is equal to the true number of input densities. In the following, we further investigated the performance robustness of RPEM when such an assumption is violated. With the same experimental data set, we randomly assigned seven seed points rather than three ones in the input space as shown in Fig. 4a and ran the RPEM. After 200 epochs, Fig. 4b shows the positions of seven seed points, among which the three ones

$$\mathbf{m}_1 = \begin{pmatrix}1.0089\\0.9739\end{pmatrix}, \quad \mathbf{m}_3 = \begin{pmatrix}0.9787\\4.9784\end{pmatrix}, \quad \mathbf{m}_4 = \begin{pmatrix}5.0171\\5.0065\end{pmatrix}, \tag{3}$$

have successfully stabilized at the corresponding cluster centers; meanwhile, the extra four seed points have been gradually pushed far away from the input data region and finally stayed at the outside. We further investigated the corresponding values of α_js. As shown in Fig. 5a, all of those corresponding to the extra densities have been approached to zero. According to the mixture model of (18), we know that the effects of a density component, say the jth one, in the model is determined by the value of α_j and the Mahalanobis distance between an input $\mathbf{x}_t$ and the density mean $\mathbf{m}_j$. The RPEM learning has led these two values of an extra density to zero. In other words, the effects of those extra densities have been fade out in the mixture model through the learning. Hence, the RPEM can automatically make the model selection. To further demonstrate this property, Fig. 6b shows the distribution of the three principal Gaussian density components learned via RPEM, i.e., the three density components whose corresponding α_js are the first three largest ones. Compared to the true input distribution in Fig. 6a, it can be seen that these three principal density components have well-estimated the true one. In contrast, under the same experimental setting, the EM let all seed points stay at some places biased from the cluster centers as shown in Fig. 4c. That is, EM cannot approach the Mahalanobis distance of an extra density to zero. Furthermore, Fig. 5b shows the learning curve of α_js. A snapshot of seven α_js' values is:

$$\begin{aligned} &\alpha_1 = 0.3121, \quad \alpha_2 = 0.1281, \quad \alpha_3 = 0.1362, \quad \alpha_4 = 0.1139, \\ &\alpha_5 = 0.1021, \quad \alpha_6 = 0.1036, \quad \alpha_7 = 0.1040. \end{aligned} \tag{4}$$

It can be seen that none of α_js tends to zero. Hence, EM is unable to select a model automatically. Fig. 6c shows the distribution of the three principal Gaussian density components learned via the EM, in which one Gaussian density is disappeared because the EM has made two principal density components mix together to approximate one true Gaussian density. Evidently, the EM cannot work at all in this case.

1041-4347/05/$20.00 © 2005 IEEE Published by the IEEE Computer Society

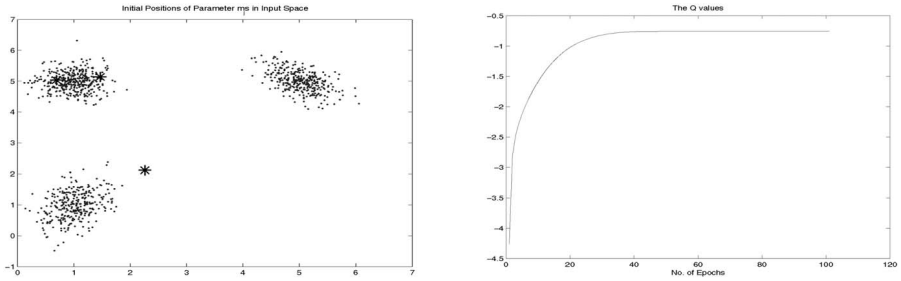


Fig. 2. In this figure, (a) shows the distribution of the inputs in Experiment 1, in which three seed points marked by "$*$" are randomly located in the input space and (b) gives out the Q value of (26) over the epochs when the model parameters are learned via the RPEM.

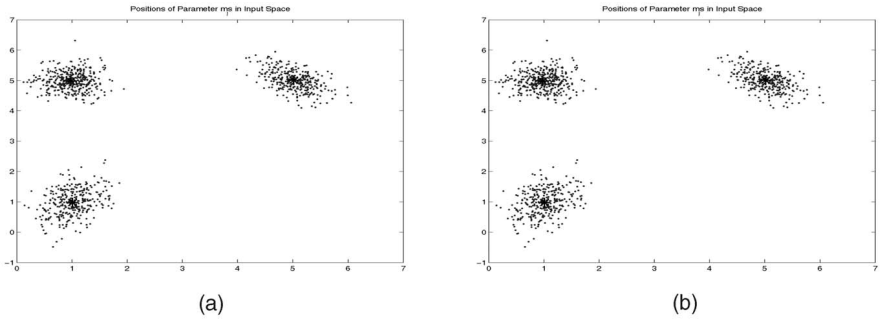


Fig. 3. The positions of three converged seed points learned by: (a) the RPEM and (b) the EM, respectively.

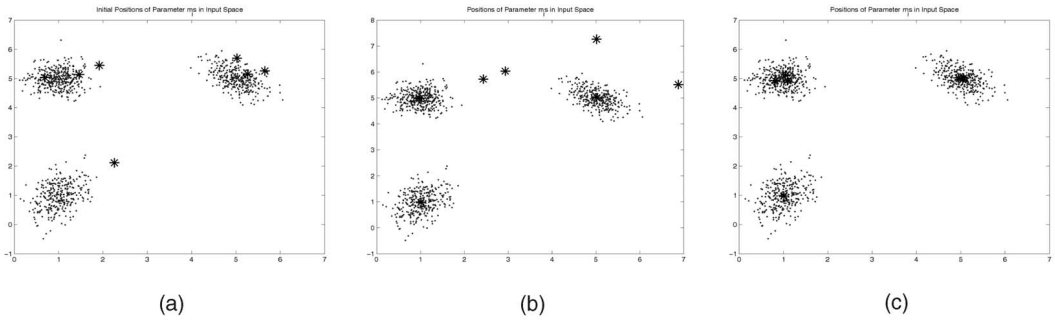


Fig. 4. The positions of three seed points marked by "$*$" in the input space: (a) the initial random positions, (b) the converged positions obtained via the RPEM, and (c) the converged positions obtained via the EM.

1.2 Experiment 2

Upon the data clusters well-separated in Experiment 1, we further investigated the performance of RPEM on the data clusters that were considerably overlapped. Similar to Experiment 1, we generated 1,000 synthetic data points from a mixture of three bivariate Gaussian densities:

$$\begin{aligned} p(\mathbf{x}|\mathbf{\Theta}^*) = {} & 0.3G\left[\mathbf{x}|\begin{pmatrix}1\\1\end{pmatrix},\begin{pmatrix}0.20, & 0.05\\0.05, & 0.30\end{pmatrix}\right] \\ & + 0.4G\left[\mathbf{x}|\begin{pmatrix}1.0\\2.5\end{pmatrix},\begin{pmatrix}0.20, & 0.00\\0.00, & 0.20\end{pmatrix}\right] \\ & + 0.3G\left[\mathbf{x}|\begin{pmatrix}2.5\\2.5\end{pmatrix},\begin{pmatrix}0.20, & -0.10\\-0.10, & 0.20\end{pmatrix}\right]. \end{aligned} \quad (5)$$

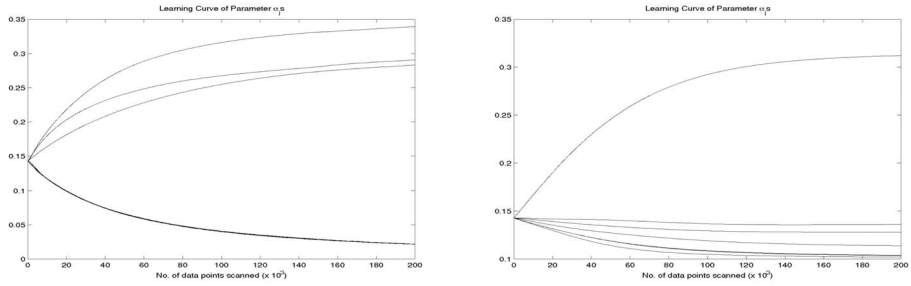
We set $k = 3$, and randomly assigned three seed points in the input space, as shown in Fig. 7a. Under the same experimental environment setting as Experiment 1, we performed the RPEM and EM. Figs. 7b and 7c show the stable positions of seed points learned by the RPEM and EM, respectively. A snapshot of α_js learned by them is:

$$\text{RPEM} : \alpha_1 = 0.3195, \quad \alpha_2 = 0.3626, \quad \alpha_3 = 0.3179, \quad (6)$$

$$\text{EM} : \alpha_1 = 0.3199, \quad \alpha_2 = 0.3315, \quad \alpha_3 = 0.3486. \quad (7)$$

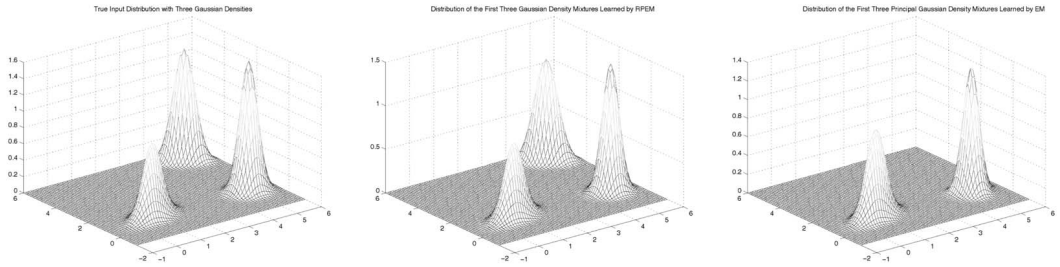
It can be seen that the α_js' estimate of RPEM is slightly better than the EM, although both of them work in this trial. Moreover, Fig. 8 shows the learning curve of seed points, in which we found that the RPEM learning is much faster than the EM. This scenario is consistent with the qualitative analysis in [25]. That is, the rival penalization mechanism can speed up the convergence of the seed points. We are going to theoretically analyze the convergence property of RPEM elsewhere because of the space limitation in this paper.

Furthermore, we investigated the RPEM performance when the number k of seed points was much larger than the true one. We arbitrarily set $k = 25$. As shown in Fig. 9a, we randomly located the 25 seed points in the input space and then learned about them as well as the other parameters by the RPEM. After 500 epochs, Fig. 9b shows the stable positions of 25 seed points, where three out of 25 seed points are located at the corresponding cluster centers,



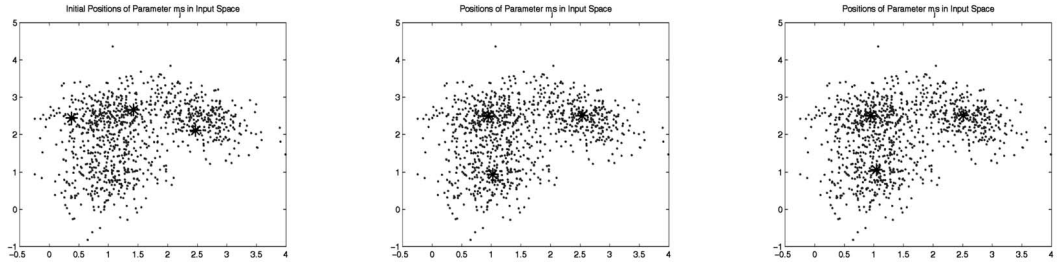
(a) (b)

Fig. 5. The learning curves of α_js obtained via: (a) the RPEM and (b) the EM, respectively.



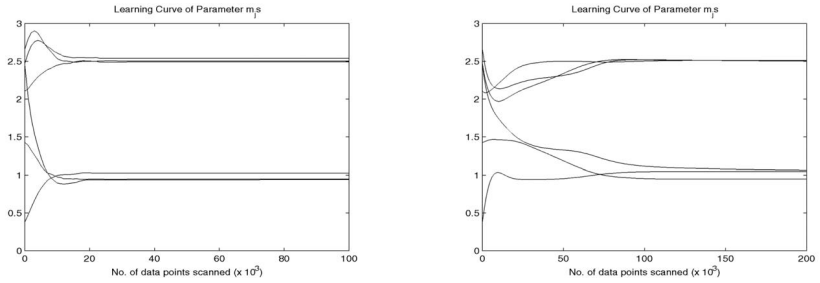
(a) (b) (c)

Fig. 6. In this figure, (a) shows the true input distribution, whereas (b) and (c) show the distribution of the first three principal Gaussian density



(a) (b) (c)

Fig. 7. The positions of three seed points marked by "$*$" in the input space in Experiment 2: (a) the initial random positions, (b) the converged positions obtained via the RPEM, and (c) the converged positions obtained via the EM.



(a) (b)

Fig. 8. In this figure, (a) shows the learning curves of three seed points learned by RPEM in Experiment 2, whereas (b) shows the curves learned by EM.

while the others stay at the boundaries or the outside of the clusters. A snapshot of converged α_js is:

$$\alpha_2 = 0.3203, \quad \alpha_4 = 0.2993, \quad \alpha_{23} = 0.3012, \tag{8}$$

while the others tend to zero, as shown in Fig. 10a. In other words, the input data set has been successfully recognized from the mixture of the three densities: 2, 4, and 23.

For comparison, we also showed the EM performance under the same experimental environment. Fig. 9c depicts the final positions of 25 seed points in the input space, where they are all biased from the cluster centers. Furthermore, Fig. 10b illustrates the learning curves of α_js, in which no one is approached to zero. Instead, the EM led 25 densities to compete each other without making extra

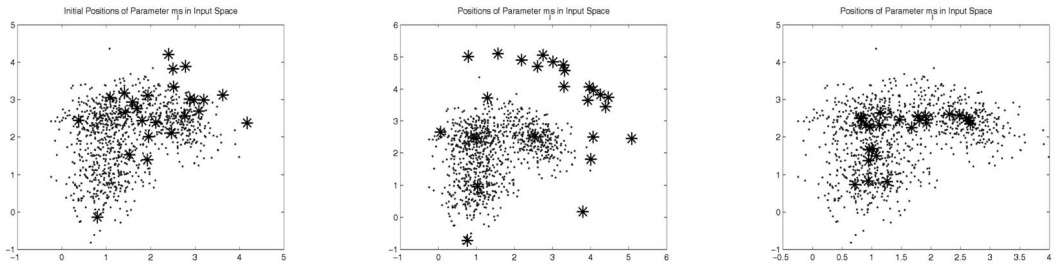


Fig. 9. The positions of 25 seed points marked by "$*$" in the input data space in Experiment 2: (a) the initial random positions, (b) the stable positions obtained via the RPEM, and (c) the stable positions obtained via the EM.

densities die. It turns out that the EM cannot work at all in this case. That is, similar to Experiment 1, this experiment has shown that the RPEM outperforms the EM upon the robust performance in terms of the mixture number k again.

1.3 Experiment 3

The previous experiment showed the performance of RPEM under the three clusters. In this experiment, we will investigate its performance when the true number of clusters is large. For the sake of visibility, we generated the data points from a mixture of 10 bivariate Gaussian density distributions with the proportions being:

$$\begin{array}{llll} \alpha_1^* = 0.10, & \alpha_2^* = 0.10, & \alpha_3^* = 0.15, & \alpha_4^* = 0.05, \\ \alpha_5^* = 0.10 & \alpha_6^* = 0.15, & \alpha_7^* = 0.05, & \alpha_8^* = 0.10, \\ \alpha_9^* = 0.10, & \alpha_{10}^* = 0.10. & & \end{array} \quad (9)$$

Also, we set k at 30. The other experimental setting was the same as Experiments 1 and 2. Fig. 11a shows the initial positions of 30 seed points in the input space. After 300 epochs, Fig. 11b shows the stable positions of those seed points. It can be seen that 10 out of 30 seed points have been successfully converged to the corresponding cluster centers; meanwhile, the other extra 20 seed points have been driven away from the input set and stayed at the boundary or the outside of the clusters. Actually, these corresponding extra densities have been faded out from the mixture. Fig. 11c shows the learning curves of α_js, in which 20 curves have converged toward zero and the other 10 curves converged to the correct values. That is, the RPEM has successfully identified that the data points are from the mixture of 10 Gaussian densities. A snapshot of the 10 largest convergent α_js' values is:

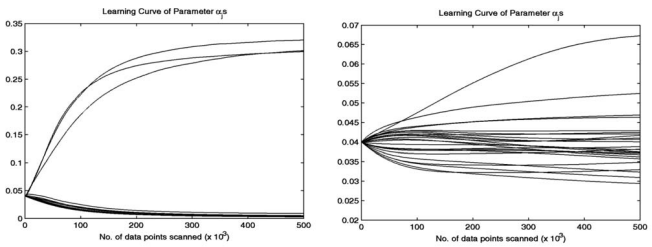


Fig. 10. In this figure, (a) and (b) show the learning curves of α_js via RPEM and EM, respectively.

$$\begin{array}{llll} \alpha_6 = 0.09, & \alpha_{10} = 0.10, & \alpha_{12} = 0.04, & \alpha_{13} = 0.10, \\ \alpha_{21} = 0.09 & \alpha_{23} = 0.10, & \alpha_{25} = 0.04, & \alpha_{27} = 0.14, \\ \alpha_{29} = 0.15, & \alpha_{30} = 0.09, & & \end{array} \quad (10)$$

whose values are very close to the true ones in (9). It can be seen that the RPEM has the robust performance even if both of k^* and k become large.

1.4 Experiment 4

In this experiment, we further investigated the robustness of RPEM in 10 clusters that were seriously overlapped. Fig. 12a shows the input distribution in the input space, where we randomly allocated 15 seed points. After 100 epochs, we found that seven seed points had stabilized at the cluster centers or the middle of two clusters as shown in Fig. 12b, while the other seed points had been driven far from the input sets. That is, the RPEM has led the model parameters into a local maximum solution and identified seven clusters only, but not the true 10 ones. Nevertheless, it can be seen from Fig. 12b that some of clusters have been seriously overlapped, which may be more reasonable to regard as a single cluster, rather than count on an individual basis. In this viewpoint, the results given by the RPEM are acceptable and correct even if the clusters are seriously overlapped.

1.5 Experiment 5

In the previous experiments, we consider the bivariate data points only for easy visual demonstration. This experiment will show the RPEM performance on high-dimensional data. We generated 3,000 data points from a mixture of four 30-dimension Gaussians with the coefficients:

$$\alpha_1^* = 0.2, \quad \alpha_2^* = 0.3, \quad \alpha_3^* = 0.2, \quad \alpha_4^* = 0.3. \quad (11)$$

The projection map of the inputs on two dimensions is shown in Fig. 13a. We randomly assigned seven seed points in the input space and learned them by RPEM. After 300 epochs, a snapshot of α_js' values is:

$$\begin{array}{llll} \alpha_1 = 0.2029, & \alpha_2 = 0.0067, & \alpha_3 = 0.2918, & \alpha_4 = 0.0065, \\ \alpha_5 = 0.1942, & \alpha_6 = 0.2907, & \alpha_7 = 0.0071, & \end{array} \quad (12)$$

in which α_1, α_3, α_5, and α_6 are very close to the true ones, meanwhile α_2, α_4 and α_7 tend to zero as shown in Fig. 13c. Fig. 13b shows the two-dimension projection of the converged seed points in the input space. We found that $\mathbf{m}_1$, $\mathbf{m}_3$, and $\mathbf{m}_5$ had successfully stabilized at the corresponding cluster centers, while $\mathbf{m}_2$ and $\mathbf{m}_4$ had been

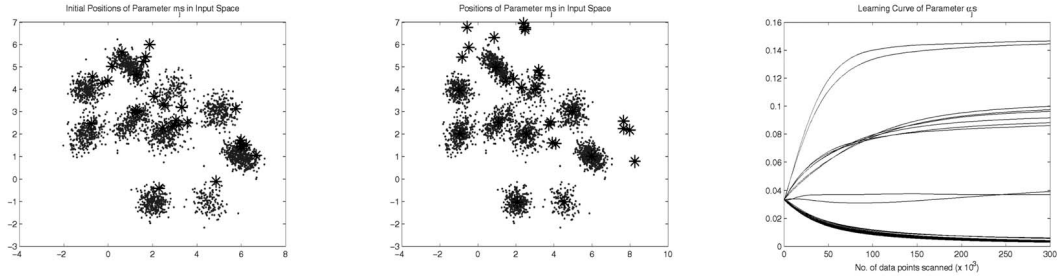


Fig. 11. The results obtained via the RPEM in Experiment 3: (a) the initial positions of 30 seed points marked by "$*$" in the input data space, (b) the stable positions of the seed points learned by the RPEM, and (c) the learning curves of α_js.

pushed away from the inputs and died. In Fig. 13b, it seems that the positions of two seed points, $\mathbf{m}_6$ and $\mathbf{m}_7$, are very close each other in the projection map. We further calculated their Euclidean distance. The value is 0.9654, which is over six times of the variance. That is, $\mathbf{m}_7$ is actually far from $\mathbf{m}_6$ in the original 30-dimension space. Hence, the RPEM has successfully identified the true data distribution in this trial.

1.6 Experiment 6

This experiment demonstrated the performance of RPEM in color image segmentation in comparison with the common *k*-means algorithm. We used the benchmark *Beach* image with 64×64 pixels as shown in Fig. 14a, in which the sky is neighbored with a small hillside and sea is connected with the sand beach. We performed the image segmentation in HSV color space. Before doing that, we applied Gaussian filter to smooth the image. We initially assigned 10 seed points as shown in Fig. 14b and learned about them by the RPEM and *k*-means algorithms, respectively. Fig. 15b shows the converged positions of these 10 seed points learned about them by the RPEM in HSV color space. It can be seen that the RPEM makes the four seed points remained and puts all other seed points far way from the data set. As a result, the image is segmented as shown in Fig. 15a, in which the sky is well-separated with the hillside, and so is it between the sea and the sand beach. In this trial, we noticed that the sky color was close to the sea color. This implies that the region of sky seriously overlaps the region of sea in HSV color space. Subsequently, it leads the RPEM to be trapped into a local optimal solution similar to the case in Experiment 4. Nevertheless, the results given by the RPEM in this experiment are still acceptable. In contrast, Figs. 16a and 16b show the results from *k*-means algorithm, in which we found that the *k*-means could not make a correct image segmentation at all.

In the previous experiments, we have numerically demonstrated the performance of RPEM in a variety of experimental environment by using both of synthetic and real-life data. It can be seen that the RPEM has a robust performance in all cases we have tried so far. Nevertheless, it should be noted that the RPEM requests the number k of seed points to be equal to or greater than the true k^*. Otherwise, the RPEM may lead some seed points to stable at the center of two or more clusters. To circumvent this limitation, we can develop another algorithm from the MWL framework by introducing a mechanism to increase or decrease the number of seed points dynamically without such a limitation. Since its discussion has been beyond the scope of this paper, we prefer to leave its details elsewhere.

2 Experimental Demonstrations for S-RPCL

To save space, we conducted two experiments to compare the S-RPCL and the RPCL. In each experiment, we used six seed points, whose initial positions were randomly assigned in the input space. Moreover, we randomly set the learning rate $\eta = 0.001$, while letting the delearning rate $\eta_r = 0.0001$ by default when using the RPCL.

2.1 Experiment 1

We used the 1,000 data points from a mixture of three Gaussian distributions:

$$\begin{aligned} p(\mathbf{x}|\mathbf{\Theta}^*) = {} & 0.3G\left[\mathbf{x}|\begin{pmatrix}1\\1\end{pmatrix},\begin{pmatrix}0.1, & 0\\ 0, & 0.1\end{pmatrix}\right] \\ & + 0.4G\left[\mathbf{x}|\begin{pmatrix}1\\5\end{pmatrix},\begin{pmatrix}0.1, & 0\\ 0, & 0.1\end{pmatrix}\right] \\ & + 0.3G\left[\mathbf{x}|\begin{pmatrix}5\\5\end{pmatrix},\begin{pmatrix}0.1, & 0\\ 0, & 0.1\end{pmatrix}\right], \end{aligned} \tag{13}$$

which forms three well-separated clusters with the six seed points $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_6$ randomly located at:

$$\begin{aligned} \mathbf{m}_1 &= \begin{pmatrix}2.2580\\1.9849\end{pmatrix}, \quad \mathbf{m}_2 = \begin{pmatrix}1.4659\\5.1359\end{pmatrix}, \quad \mathbf{m}_3 = \begin{pmatrix}0.6893\\5.0331\end{pmatrix} \\ \mathbf{m}_4 &= \begin{pmatrix}5.2045\\5.1298\end{pmatrix}, \quad \mathbf{m}_5 = \begin{pmatrix}1.9193\\5.4489\end{pmatrix}, \quad \mathbf{m}_6 = \begin{pmatrix}5.5869\\5.1937\end{pmatrix}. \end{aligned} \tag{14}$$

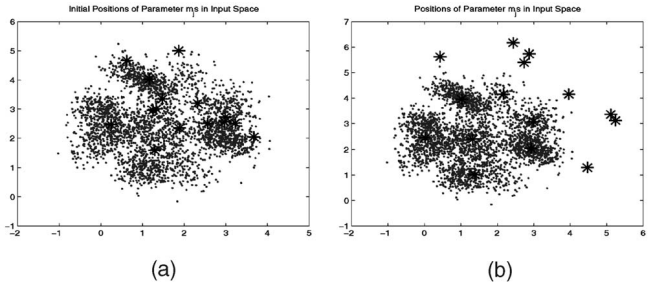


Fig. 12. The positions of 15 seed points marked by "$*$" in the input data space in Experiment 4: (a) the initial random positions and (b) the stable positions obtained via the RPEM.

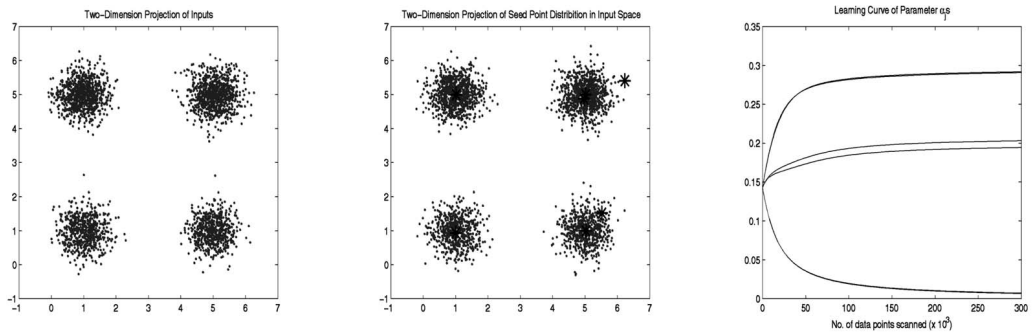


Fig. 13. The results obtained via the RPEM in Experiment 5: (a) the projection of 30-dimension data points on the plane, (b) the final positions of 7 seed points learned via the RPEM, and (c) the learning curves of α_js.

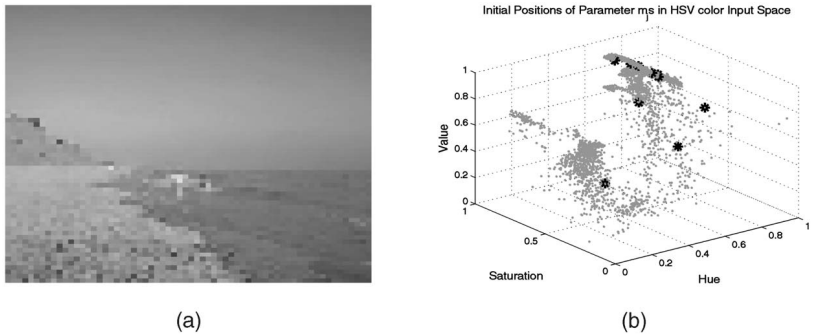


Fig. 14. (a) The benchmark "Beach" image and (b) the initial positions of 10 seed points in HSV color space.

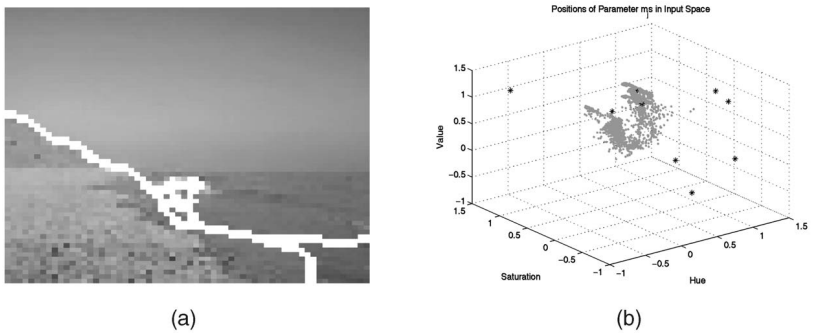


Fig. 15. In this figure, (b) shows the converged positions of seed points learned by the RPEM algorithm, while (a) shows the segmented image accordingly.

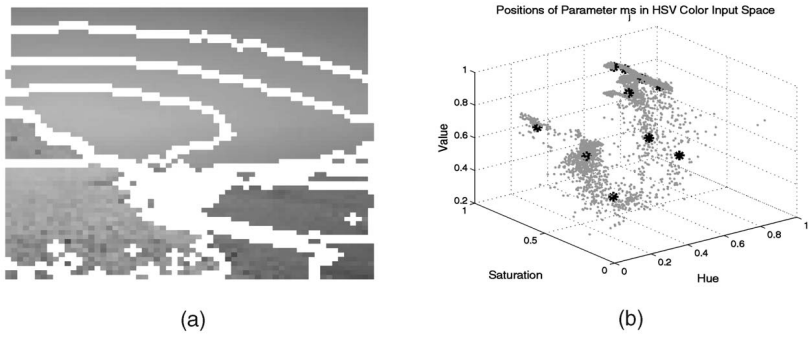


Fig. 16. In this figure, (b) shows the converged positions of seed points learned by the *k*-means algorithm, while (a) shows the segmented image accordingly.

Fig. 17a shows the positions of all seed points in the input space after 800 epochs, and Fig. 17b shows their learning trajectory. It can be seen that the S-RPCL has put three seed points, $\mathbf{m}_1$, $\mathbf{m}_2$, and $\mathbf{m}_4$, into the three cluster centers, meanwhile driving the other three extra seed points, $\mathbf{m}_3$, $\mathbf{m}_5$, and $\mathbf{m}_6$, far away from the input data set.

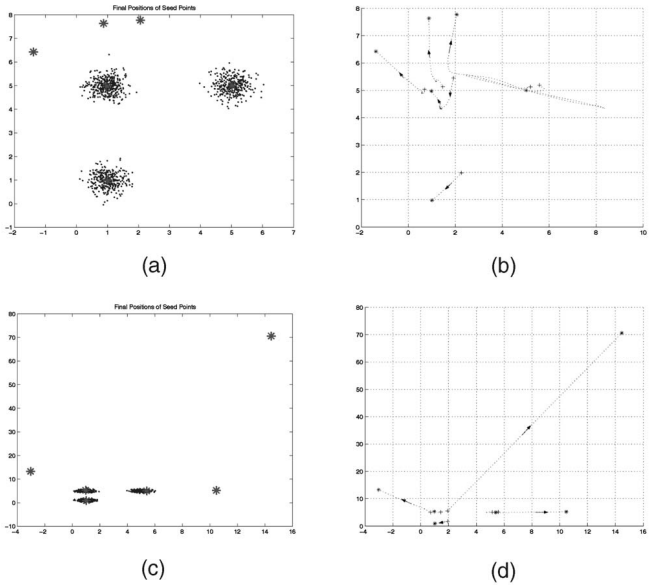


Fig. 17. In this figure, (a) shows the final positions of six seed points (marked by "$*$") obtained via the S-RPCL in Experiment 1 of Section 2.1 and (b) shows the learning trajectory of six seed points, in which "+" marks the initial positions of seed points, and "*" marks the final positions. It can be seen that the extra seed points have been gradually driving far away from the regions of the input data set. (c) shows a snapshot of the seed points learned by the RPCL in the input space and (d) is the learning trajectory of six seed points.

Based on the rival penalization equation in **Step 2** of Table 3, we know that the rival penalization strength will nonlinearly decrease as an extra seed point leaves the input data set, and they will finally become stable outside the input data set. For comparison, we also implemented the RPCL under the same experimental environment. Fig. 17c shows that the RPCL has successfully driven three extra points, $\mathbf{m}_3$, $\mathbf{m}_5$, and $\mathbf{m}_6$, to

$$\mathbf{m}_3 = \begin{pmatrix} -3.0326 \\ 13.2891 \end{pmatrix} \quad \mathbf{m}_5 = \begin{pmatrix} 14.4600 \\ 70.6014 \end{pmatrix}, \quad \mathbf{m}_6 = \begin{pmatrix} 10.4714 \\ 5.2240 \end{pmatrix}, \tag{15}$$

which are far away from the input data set, while the other three seed points:

$$\mathbf{m}_1 = \begin{pmatrix} 1.0167 \\ 0.9321 \end{pmatrix} \quad \mathbf{m}_2 = \begin{pmatrix} 0.9752 \\ 5.3068 \end{pmatrix}, \quad \mathbf{m}_4 = \begin{pmatrix} 5.4022 \\ 5.0054 \end{pmatrix}, \tag{16}$$

locate at the correct positions. Hence, the RPCL can work as well in this case. However, we have also noticed that, as shown in Fig. 17d, the RPCL always penalizes the extra seed points even if they are much farther away from the input data set. Consequently, the seed points as a whole will not tend to convergence, but those learned by the S-RPCL will.

2.2 Experiment 2

We further investigated the performance of S-RPCL by generating 1,000 data points from a mixture of three Gaussian distributions:

$$\begin{aligned} p(\mathbf{x}|\mathbf{\Theta}^*) = {} & 0.3G\left[\mathbf{x}|\begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 0.15, & 0 \\ 0, & 0.15 \end{pmatrix}\right] \\ & + 0.4G\left[\mathbf{x}|\begin{pmatrix} 1 \\ 2.5 \end{pmatrix}, \begin{pmatrix} 0.15, & 0 \\ 0, & 0.15 \end{pmatrix}\right] \\ & + 0.3G\left[\mathbf{x}|\begin{pmatrix} 2.5 \\ 2.5 \end{pmatrix}, \begin{pmatrix} 0.15, & 0 \\ 0, & 0.15 \end{pmatrix}\right], \end{aligned} \tag{17}$$

which forms three moderate overlapping clusters as shown in Fig. 18a. After 800 epochs, we found that the S-RPCL had given out the correct results as shown in Fig. 18b, but the RPCL could not work as shown in Fig. 18c, even if we increased the epoch number up to 1,000. Also, we further investigated the performance of RPCL by adjusting the delearning rate η_r along two directions: from 0.0001 to 0.00001 and from 0.0001 to 0.0009, respectively, with a constant step: 0.00001. Unfortunately, we could not find out an appropriate η_r in all cases we had tried so far to make RPCL successfully work.

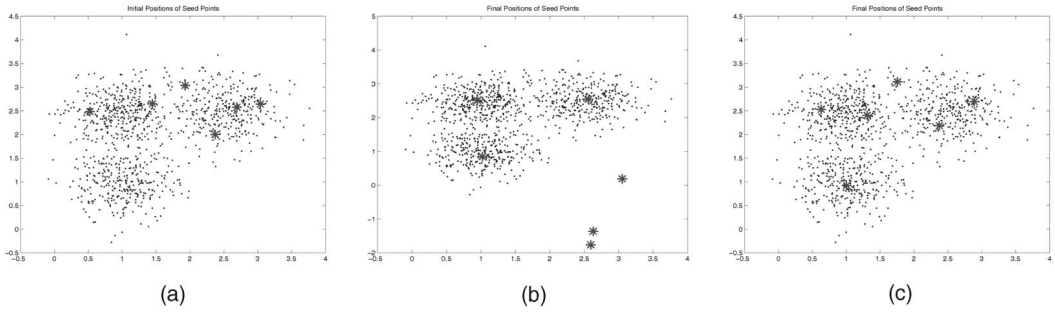


Fig. 18. In this figure, (a) shows the initial positions of six seed points marked by "$*$" in Experiment 2 of Section 2.2. (b) and (c) show the final positions of the converged seed points learned by the S-RPCL and RPCL, respectively.