Finite Mixture Models with Negative Components [⋆]

Baibo Zhang and Changshui Zhang

State Key Laboratory of Intelligent Technology and Systems
Department of Automation, Tsinghua University, Beijing 100084, P. R. China

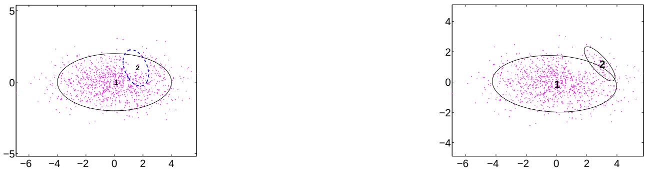
Abstract. Mixture models, especially mixtures of Gaussian, have been widely used due to their great flexibility and power. Non-Gaussian clusters can be approximated by several Gaussian components, however, it can not always acquire appropriate results. By cancelling the nonnegative constraint to mixture coefficients and introducing a new concept of "negative components", we extend the traditional mixture models and enhance their performance without increasing the complexity obviously. Moreover, we propose a parameter estimation algorithm based on an iteration mechanism, which can effectively discover patterns of "negative components". Experiments on some synthetic data testified the reasonableness of the proposed novel model and the effectiveness of the parameter estimation algorithm.

1 Introduction

In the field of statistical learning, finite mixture models (FMM) have been widely used and have continued to receive increasing attention over years due to their great flexibility and power [1]. The capability of representing arbitrary complex probability density functions (pdf's) enables them to have many applications not only in unsupervised learning fields [2], but also in (Bayesian) supervised learning scenarios and in parameter estimation of class-conditional pdf's [3]. Especially, Gaussian Mixture Models (GMM) have been widely employed in various applications[1,2,3].

GMM can accommodate data of varied structure, since one non-Gaussian component can usually be approximated by several Gaussian ones [4,5]. However, this approximation can not always acquire appropriate results. To form an intuitive image of this fact, a sample set is generated by a Gaussian model and partly "absorbed" by another one, i.e. there is a "hole" in the data cloud as Fig.1a shows. Fitting this sample set by GMM yields a solution shown in Fig.1b. This solution is achieved by the Competitive Expectation Maximization algorithm (CEM) [6], and the component number is auto-selected by a criterion similar to *Minimum Message Length* (MML) [7]. Although this solution is not bad, it is obvious that in the "hole" area , densities are estimated higher than they should be.

[⋆] This work is supported by the project (60475001) of the National Natural Science Foundation of China.



(a) TRUE Model (b) GMM Estimation

Fig. 1. Samples generated by a component and partly absorbed by another one (average log likelihood in (a) and (b) is 0.353 and 0.351, respectively).

In the definition of traditional mixture models, the coefficients of mixture components are nonnegative. In fact, to satisfy the constraint of pdf, it only requires to meet the following two conditions: the sum of the mixture coefficients equals 1, and the probability density at any point is nonnegative. The mixture coefficients are not necessary to be nonnegative.

In this paper, we endeavor to extend mixture models by cancelling the nonnegative constraint to mixture coefficients. We introduce a new concept of "Negative Component", i.e. a component with a negative mixture coefficient.

The rest of this paper is organized as follows. We will describe this proposed model in Sect.2. A parameter estimation algorithm based on an iteration mechanism is given in Sect.3. Experiments are presented in Sect.4, followed by a short discussion and conclusion in Sect.5.

2 Finite Mixture Models with Negative Components

It is said a d-dimensional random variable $x = [x_1, x_2, \cdots, x_d]^T$ follows a k-component finite mixture distribution, if its pdf can be written as

$$p(x|\theta) = \sum\nolimits_{m=1}^{k} \alpha_m p(x|\theta_m), \tag{1}$$

where α_m is the prior probability of the mth component and satisfies

$$\alpha_m \geq 0, \text{ and } \sum\nolimits_{m=1}^{k} \alpha_m = 1. \tag{2}$$

Different descriptions of $p(x|\theta_m)$ can be assigned to different kinds of mixture models. We focus on FMM and demonstrate algorithms by means of GMM.

If the nonnegative constraint of mixture coefficients is cancelled, mixture models will be more powerful to fit data clouds. Equation (2) is modified to

$$\sum\nolimits_{m=1}^{k} \alpha_m = 1. \tag{3}$$

To ensure $p(x)$ satisfies the constraint of pdf, we should add a new constraint:

$$p(x) \geq 0, \forall x. \tag{4}$$

For convenience, we call finite mixture models with negative components NegFMM, and call the corresponding GMM version NegGMM.

2.1 An Interpretation to NegFMM

In NegFMM, a component with a positive coefficient is called a "Positive Component" and the negative one a "Negative Component". Let k^+ and k^- denote the number of positive components and negative ones, respectively. $k = k^+ + k^-$ is the total component number. For convenience, positive components and negative ones are separated as follows

$$p(x|\theta) = \sum\nolimits_{m=1}^{k^+} \alpha_m p(x|\theta_m) + \sum\nolimits_{m=k^+ +1}^{k} \alpha_m p(x|\theta_m) \tag{5}$$

Defining $a = -\sum_{m=k^+ +1}^{k} \alpha_m$, $\sum_{m=1}^{k^+} \alpha_m = 1 + a$. Let $\beta_m^+ = \alpha_m/(1+a)$, $\beta_m^- = -\alpha_{k^+ + m}/a$. Obviously, β_m^+, $\beta_m^- \geq 0$, and $\sum_{m=1}^{k^+} \beta_m^+ = 1$, $\sum_{m=1}^{k^-} \beta_m^- = 1$.

Defining $p^+ = \sum_{m=1}^{k^+} \beta_m^+ p(x|\theta_m)$, $p^- = \sum_{m=1}^{k^-} \beta_m^- p(x|\theta_{k^+ + m})$, p^+ and p^- are traditional mixture models. So NegFMM can be expressed as

$$p_M = (1+a)p^+ - ap^-. \tag{6}$$

p^+ is called "Positive Pattern" and p^- is called "Negative Pattern". When $a = 0$, NegFMM will degrade to FMM. In this paper, we only focus on the case of $a > 0$.

Moving p^- to the left side, (6) can be rewritten as

$$p^+ = \tfrac{1}{1+a} p_M + \tfrac{a}{1+a} p^-. \tag{7}$$

Then the positive pattern p^+ is expressed as a mixture of the model p_M and the negative pattern p^-. This expression clearly shows that the negative pattern can not exist independently and it is only a part of the positive pattern.

2.2 The Nonnegative Density Constraint for NegGMM

NegFMM introduces the nonnegative density constraint (4). In this section, we will further analyze this constraint in the case of NegGMM.

This constraint is to ensure $(1+a)p^+ - ap^- = p^+ + a(p^+ - p^-) \geq 0$. When $p^+ - p^- \geq 0$, $p_M \geq 0$ is met. When $p^+ - p^- < 0$, it means

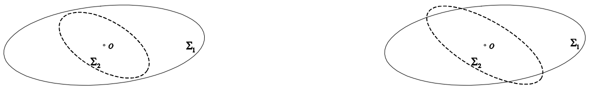
$$a \leq p^+/(p^- - p^+). \tag{8}$$

We will show that this constraint can be decomposed to two parts, i.e. the constraint to covariance matrices and the constraint to a, corresponding to the nonnegative condition for infinite x and finite x, respectively.

The Covariance Constraint. For Gaussian distribution, the covariance matrix describes the density decaying rate in any direction. For a direction r ($\|r\| = 1$), the variance satisfies $\sigma_r^2 = r^T \Sigma r$, because

$$\sigma_r^2 = \int [r^T(x-\mu)]^2 p(x)dx = r^T [\int (x-\mu)(x-\mu)^T p(x)dx] r = r^T \Sigma r.$$

For the case of $k^+ = k^- = 1$, if there is a direction r where the variance of p^- is larger than p^+, the right side of (8) will approach zero when x goes to infinite along the direction r. This will lead to $a = 0$. So the covariance constraint is $\sigma_{1r}^2 \geq \sigma_{2r}^2, \quad \forall r, \quad \|r\| = 1$. Fig.2 illustrate this case: the model in Fig.2a satisfies the covariance constraint while the model in Fig.2b does not.



(a) (b)

Fig. 2. Illustration of the covariance constraint.

In the general case of NegGMM with arbitrary k^+ and k^-, the constraint will be similar. In any direction, variances of all negative components must be not larger than the maximum variances of all positive components,

$$\max_{1 \leq m \leq k^+} \{\sigma_{mr}^2) \geq \max_{k^+ + 1 \leq m \leq k} \{\sigma_{mr}^2), \quad \forall r, \quad \|r\| = 1. \tag{9}$$

The Constraint to *a*. If NegGMM satisfies the covariance constraint, there exists a threshold $a_T > 0$. If $a = a_T$, $\min\limits_{x} p(x) = 0$. So the constraint to a is

$$a \leq a_T, \tag{10}$$

where $a_T = \min\limits_{x \in \{x | p^- - p^+ > 0\}} \{p^+ / (p^- - p^+)\}$.

3 A Framework of Parameter Estimation

Assuming that samples in the set $X = \{x_1, x_2, \cdots, x_n\}$ are independently drawn from the NegGMM model, how to estimate parameters of the model from X is a difficult problem, since no samples originate from the negative pattern.

To estimate an appropriate number of components, many deterministic criteria are proposed [1]. In this paper, we do not consider the problem of choosing the numbers of components. We take the Maximum Likelihood (ML) estimation as our object function,

$$J = \frac{1}{n} \sum\nolimits_{i=1}^{n} \log(p(x_i | \theta)). \tag{11}$$

The EM algorithm is a widely used class of iterative algorithms for Maximum Likelihood or Maximum A Posteriori (MAP) estimation in problems with incomplete data, e.g. parameter estimation to mixture models [8,9]. However, the EM framework is difficult to be directly used in NegGMM, because of the existence of negative coefficients terms.

3.1 Basic Ideas

Parameters of negative pattern p^- can not be directly estimated from the sample set X. According to (7), p^- can be viewed as the result of subtracting p_M from p^+, where p^+ can be estimated, but p_M is unknown. Intuitively, p_M can be approximated by the sample density function p_s which can be estimated by the Parzen window based methods. Then (7) can be approximated as

$$p^+ = \tfrac{1}{1+a} p_s + \tfrac{a}{1+a} p^- . \tag{12}$$

At first p^+ is estimated according to X, then p^- is estimated according to (12). After that a is estimated under the nonnegative density constraint. Then p^+ is reestimated using the information of p^- and a, and so on.

p^+, p^- and a are optimized separately, i.e. when one part is being optimized, the other parts are fixed. This is similar to the idea of Gibbs Sampling [10].

In order to estimate p^-, we first sampling p^+ to get a sample set. Then, based on (12), we use a modified EM algorithm to estimate p^- with a fixed mixture component p_s.

In order to estimate p^+, we sampling p^- and weight the sample set according to a. The union of the weighted sample set and X can be viewed as a sample set generated by p^+. Then p^+ can be estimated by EM.

In order to estimate a, we first estimate the threshold a_T. Then, under the constraint of $a \leq a_T$, we search for the most appropriate a which leads to the highest likelihood value.

The Manifold Parzen Window Algorithm. To estimate p^-, the sample density function p_s needs to be estimated to approximate p_M. To ensure a satisfying result, this estimation should be as accurate as possible. Usually, the sample distribution is inhomogeneous, so the traditional Parzen window method can not promise to obtain a good estimation due to a uniform isotropic covariance.

In this paper, we use the manifold Parzen window algorithm proposed by Vincent [11]. The main idea is that the covariance matrix of sample x_i is calculated by its neighbor points

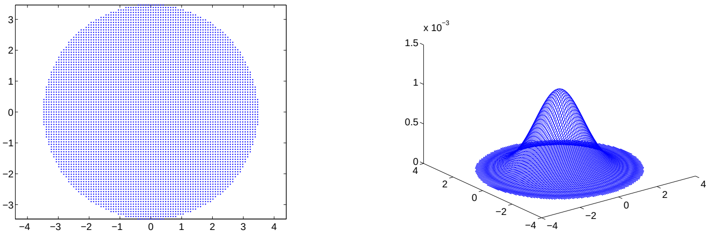
$$\Sigma_{K_i} = \frac{\sum_{j \neq i} \mathrm{K}(x_j; x_i)(x_j - x_i)(x_j - x_i)^T}{\sum_{j \neq i} \mathrm{K}(x_j; x_i)}, \tag{13}$$

where the neighbor constraint $\mathrm{K}(x; x_i)$ could be soft, e.g. a spherical Gaussian centered on x_i, or hard, e.g. only assigning 1 to the nearest k neighbors and 0 to others. Vincent used the latter in his experiments. Considering the data sparsity in high-dimension space, Vincent added two parameters to enhance the algorithm, i.e. the manifold dimension d and the noise variance σ^2. The first d eigenvectors with large eigenvalues to Σ_{K_i} are kept, zeroing the other eigenvalues and then adding σ^2 to all eigenvalues. Based on a criterion of average negative log likelihood, these three parameters are determined by cross validation.

In low-dimension space, Σ_{K_i} is supposed to be nonsingular. So only one parameter, i.e. the neighbor number k, needs to be predetermined. The computational cost for cross validation will be reduced greatly.

Grid Sampling. In order to estimate p^-, we can randomly sampling p^+. But the randomicity will lead to very unstable estimations of p^- because of small number of sampling. To solve this problem, we can increase the amount of sampling which will make the succeeding algorithm very slow, or change random sampling to grid sampling which is adopted in this paper.

For the standard Gaussian model $N(0, I)$, all grid vectors whose lengths are less than d_{scope} will be preserved, and the weight of a grid vector is in proportion to the model density at the point. In our experiments, d_{scope} is determined by experience. The grid space d_{space} can be determined according to precision requirement and computational cost. Let (S_g, W_g) denote the grid set, where S_g are grid vectors and W_g the corresponding grid point weights. Fig.3 shows the Grid sampling for the standard 2D Gaussian model.



(a) Grid Samples

(b) Sample Weights

Fig. 3. Grid sampling for standard 2D Gaussian model, d_{scope}=3.5, d_{space}=0.08.

For a general Gaussian model $N(\mu, \Sigma)$, where $\Sigma = U \Lambda U^T$, the grid sampling set (S, W) is converted from the standard set $(S_g, W_g\ )$ by $W = W_g$ and $S = U\Lambda^{1/2} S_g + \mu \cdot [1, 1, \cdots, 1]_{1\times|S_g|}$.

For traditional Gaussian mixtures, the grid sampling set (S, W) is the union of grid sets of all components, weighting W by component priors once more.

Estimating $\boldsymbol{p^-}$ with one fixed component by EM. EM is widely used to estimate the parameters of mixture models [8,9]. Our goal is to estimate p^- based on (12). Now we have a sample set (S, W) originating from p^+ (by grid sampling), a component p_s with fixed parameters (estimated using the manifold Parzen window method) and fixed mixture coefficients (determined by a).

For maximum likelihood estimation, the object function is

$$\sum w_i \ln\left(\frac{a}{1+a} p^-(s_i) + \frac{1}{1+a} p_s(s_i)\right)$$

Similar to the EM algorithm for mixture models[8], the updating formulas can be deduced easily.

E-Step: The posterior to the lth component of p^- can be calculated as

$$p(l|s_i) = \frac{a\beta_l^- p_l^-(s_i)}{ap^-(s_i) + p_s(s_i)}, l = 1, 2, \cdots, k^-.$$

This formula is similar to the E-Step in the standard GMM-EM algorithm, except that the denominator contains an additional term $p_s(s_i)$.
M-Step: The updating formulas to the lth component are very similar to the M-Step in the standard GMM-EM algorithm,

$$\beta_l^- = \frac{\sum_i w_{li}}{\sum_{m=1}^{k^-} \sum_i w_{mi}}, \quad \mu_l^- = \frac{\sum_i w_{li} s_i}{\sum_i w_{li}}, \quad \Sigma_l^- = \frac{\sum_i w_{li}(s_i - \mu_l^-)(s_i - \mu_l^-)^T}{\sum_i w_{li}},$$

where w_{li} denotes the weight of s_i to the lth component, and $w_{li} = w_i p(l|s_i)$.

3.2 Scheme of the Parameter Estimation Algorithm

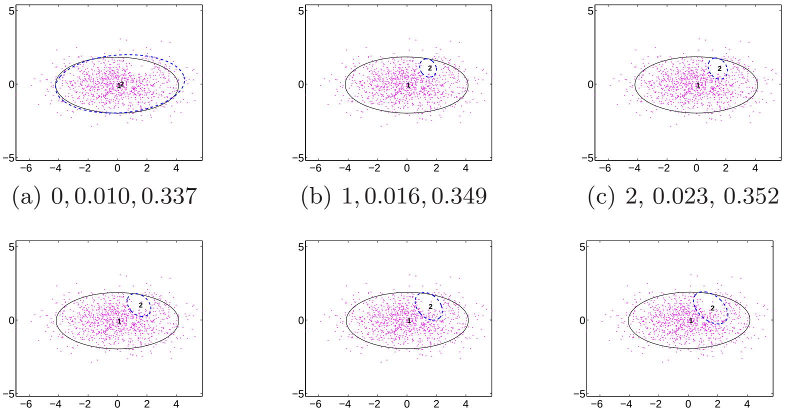
To sum up, the scheme is described as follows:

1. **Initialization:**
   Assign numbers of components k^+ and k^-;
   Estimate sample density function p_s by manifold Parzen window algorithm;
   On the sample set X, estimate p^+ using the standard EM algorithm;
   Initialize p^- randomly or by k-means based methods on the grid sampling set of p^+ (p_s is used in this step);
   Set a to be a small number, e.g. $a = 0.01$, and set iteration counter $t = 0$.
2. **One iteration:**
   Fixing p^+ and a, estimate p^- by the modified EM algorithm described above;
   Fixing p^- and a, estimate p^+ by EM, where the sample set is the union of X (weight is 1) and the grid sample set (S^-, W^-) of p^- (weight is a);
   Fixing p^+ and p^-, estimate a under the constraint (10), maximizing (11);
   The counter $t = t + 1$.
3. **End condition:**
   Repeat the iteration 2, until the object function does not change or arrives at the maximal steps. Ouput θ^* with the maximal J.

4 Experiments

Example 1. We use 1000 samples from a 2-component 2-dimension NegGMM shown in Fig.1a. The parameters are: $\alpha_1 = 1.05$, $\alpha_2 = -0.05$, $\mu_1 = [0,\ 0]^T$, $\mu_2 = [1.5,\ 1]^T$, $\mathbf{\Sigma}_1 = \begin{bmatrix} 4 & \\ & 1 \end{bmatrix}$, and $\mathbf{\Sigma}_2 = \begin{bmatrix} 0.2 & -0.1 \\ -0.1 & 0.4 \end{bmatrix}$.

Fig.4 shows the optimization procedure. In this paper, real line and dashed denote positive and negative components respectively, and p^- is initialized by k-means based methods. Fig.4a shows one initial state. Fig.4b~f show some intermediate states of the searching procedure. The best estimation is given in Fig.4f (9^{th} iteration).



(a) $0, 0.010, 0.337$ (b) $1, 0.016, 0.349$ (c) 2, 0.023, 0.352

(d) $3, 0.032, 0.355$ (e) $5, 0.047, 0.358$ (f) $9, 0.074, 0.359$

Fig. 4. Example 1: (a) Initialization (b-e) $1^{st} \sim 5^{th}$ iterations (f) the best estimation (values of t, a, J are given below each graph).

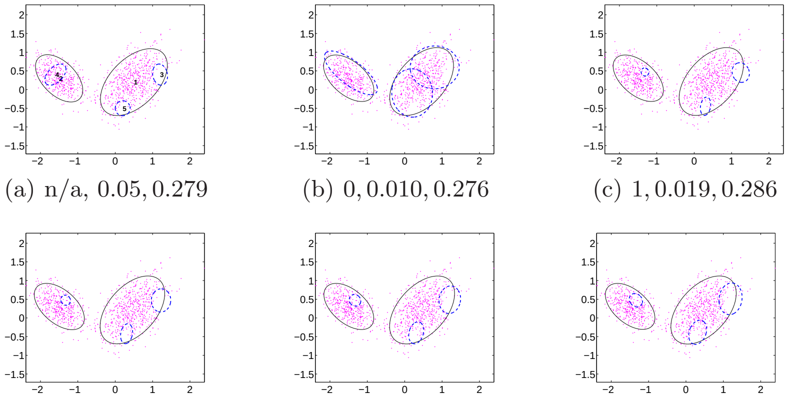
Example 2. We use 1000 samples from a 5-component 2-dimension NegGMM shown in Fig.5a, where $k^+ = 2$, $k^- = 3$ and $a = 0.05$. The parameters are:

$$\alpha_1 = 0.63,\ \alpha_2 = 0.42,\ \alpha_3 = -0.01,\ \alpha_4 = -0.03,\ \alpha_5 = -0.01$$
$$\mu_1 = [1.5,\ 0.2]^T,\ \mu_2 = [-1.5,\ 0.3]^T,$$
$$\mu_3 = [1.2,\ 0.4]^T,\ \mu_4 = [-1.6,\ 0.4]^T,\ \mu_5 = [0.2,\ -0.5]^T$$
$$\mathbf{\Sigma}_1 = \begin{bmatrix} 0.2 & 0.1 \\ 0.1 & 0.2 \end{bmatrix},\ \mathbf{\Sigma}_2 = \begin{bmatrix} 0.1 & -0.05 \\ -0.05 & 0.1 \end{bmatrix}$$
$$\mathbf{\Sigma}_3 = \begin{bmatrix} 0.01 & -0.002 \\ -0.02 & 0.02 \end{bmatrix},\ \mathbf{\Sigma}_4 = \begin{bmatrix} 0.02 & 0.01 \\ 0.01 & 0.02 \end{bmatrix},\ \mathbf{\Sigma}_5 = \begin{bmatrix} 0.01 & \\ & 0.01 \end{bmatrix}$$

Fig.5b$\sim$f plot some intermediate states of the searching procedure. The final parameter estimation is given in Fig.5f (4^{th} iteration) where $J = 0.293$. If use the traditional GMM, the best estimation given by CEM is the same as p^+ in the initial state (plotted by real lines in Fig.5b) and the corresponding J equals 0.275.

In our experiments, we do not check the covariance constraint (9). Because the sample set to estimate p^- originates from p^+, and p^+ contains a fixed component p_s, the estimation of p^- will satisfy the covariance constraint in general. This is also testified by experiments.

For p^- and a, there is observable difference between estimations and true values. It is mainly due to two reasons. The first is the large sampling error between X and the true model (this is also supported by comparing likelihood between the estimation and the true model). The second is that the samples from p^- can not be observed and the estimation algorithm may bring bias.



(a) n/a, $0.05, 0.279$
(b) $0, 0.010, 0.276$
(c) $1, 0.019, 0.286$
(d) $2, 0.032, 0.291$
(e) $3, 0.047, 0.292$
(f) $4, 0.059, 0.293$

Fig. 5. Example 2: (a) TRUE model (b) Initialization (c-e) 1^{st} $\sim 3^{rd}$ iterations (f) the best estimation (values of t, a, J are given below each graph).

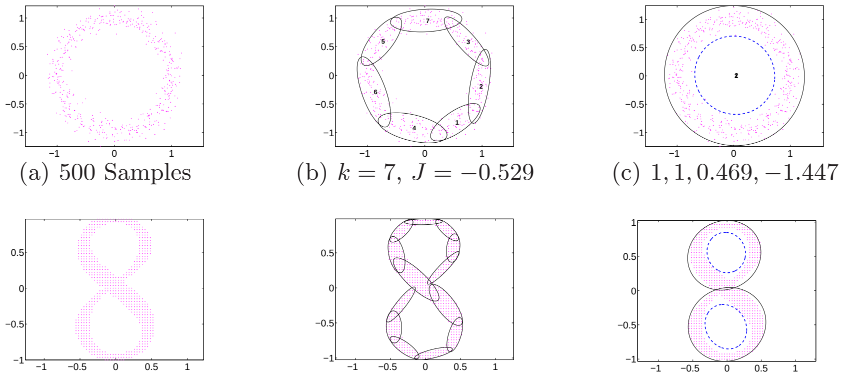
Some Interesting Examples. Fig.6 illustrates some interesting results. The first column contains sample sets, the second column contains estimations by GMM-CEM where component numbers are auto selected, and the last column contains estimations by NegGMM where component numbers are assigned by us. The first row is a synthetic ring with 500 samples (Fig.6a), the second row is 654 samples drawing from an image of digital "8" (Fig.6d). To traditional GMM, estimations of 7 and 13 components are given respectively (Fig.6b and Fig.6e). These solutions are very good. For NegGMM, estimation results are very interesting (Fig.6c and Fig.6f), though likelihood is lower.

5 Discussion and Conclusion

In this paper, we extend the traditional mixture models by cancelling the nonnegative constraint to mixture coefficients and introduce the concept of "negative pattern". The power and flexibility of mixture models are enhanced without increasing the complexity obviously.

The proposed parameter estimation framework can effectively discover patterns of lower density relative to positive pattern p^+ due to three tricks. The manifold Parzen window algorithm proposed by Vincent gives a very good estimation of sample density function p_s. The grid sampling helps to gain a very stable estimation of the nonnegative pattern. And the modified EM algorithm gives a final estimation effectively.

Due to the data sparsity, mixture models are difficult to be directly employed in high-dimension space. For the high-dimension case, there are two classes of processing methods. The first is to reduce the data dimension by linear or nonlinear methods, and the second is to constraint the model by priors or hypotheses.



(a) 500 Samples (b) $k = 7$, $J = -0.529$ (c) $1, 1, 0.469, -1.447$

(d) 654 Samples (e) $k = 13$, $J = -0.003$ (f) $2, 2, 0.448, -0.853$

Fig. 6. Some interesting results: (a,d) Sample Sets; (b,e) GMM Estimation by CEM; (c,f) NegGMM Estimation(k^+, k^-, a, J).

In complex situations, it is very difficult to find an acceptable solution for mixture models by standard EM because of its greed nature. In the future, it is necessary to do more research on split, merge and annihilation mechanism of NegFMM as our previous work[6].

References

1. McLachlan, G., Peel, D.: Finite Mixture Models. New York: John Wiley & Sons (2000)
2. Jain, A.K., Dubes, R.: Algorithm for Clustering Data. Englewood Cliffs. N.J.: Prentice Hall (1988)
3. Hinton, G., Dayan, P., Revow, M.: Modeling the manifolds of images of handwritten digits. IEEE Trans. on Neural Networks **8** (1997) 65–74
4. Dasgupta, A., Raftery, A.E.: Detecting features in spatial point processes with clutter via model-based clustering. Journal of the American Statistical Association **93** (1998) 294–302
5. Fraley, C., Raftery, A.E.: How many clusters? which clustering method? – answers via model-based cluster analysis. The Computer Journal **41** (1998) 578–588
6. Zhang, B., Zhang, C., Yi, X.: Competitive EM algorithm for finite mixture models. Pattern Recognition **37** (2004) 131–144
7. Figueiredo, M.A., Jain, A.K.: Unsupervised learning of finite mixture models. IEEE Trans. on PAMI **24** (2002) 381–396
8. Bilmes, J.A.: A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden markov models. Technical Report ICSI TR-97-021, UC Berkeley (1997)
9. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum-likelihood from incomplete data via the EM algorithm. Journal of Royal Statistic, Society B **39** (1977) 1–38
10. Geman, S., Geman, D.: Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images. IEEE Transactions on PAMI **6** (1984) 721–741
11. Vincent, P., Bengio, Y.: Manifold Parzen windows. In: NIPS. (2002)