Gaussian mixture learning via robust competitive agglomeration

Zhiwu Lu a,b, Yuxin Peng a,*, Horace H.S. Ip b

a Institute of Computer Science and Technology, Peking University, Beijing 100871, China
b Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong

Abstract

When learning Gaussian mixtures from multivariate data, it is crucial to select the appropriate number of components and simultaneously avoid local optima. To resolve these problems, we follow the idea of competitive agglomeration which is originally used for fuzzy clustering and propose two robust algorithms for Gaussian mixture learning. Through some asymptotic analysis, we find that such robust competitive agglomeration can lead to automatic model selection on Gaussian mixtures and also make our algorithms less sensitive to initialization than the EM algorithm. Experiments demonstrate that our algorithms can achieve promising results just as our theoretic analysis.

1. Introduction

As a powerful statistical modeling tool for multivariate data, Gaussian mixture model has been widely used in many applications such as pattern recognition, computer vision, and image analysis (Hajji, 2005; Diplaros et al., 2007; Persson et al., 2008). The standard method used for fitting Gaussian mixtures to the data is the EM algorithm (Render and Walker, 1984; McLachlan and Krishnan, 1997), which converges to a maximum likelihood estimation of the mixture parameters. However, the number of components in the mixture is usually assumed to be fixed and must be provided first. In many instances, this key information is not available, and we have to select $k^*$ to best fit the data before or during parameter estimation.

The traditional method to solve this model selection problem is to choose the optimal number $k^*$ of components via some statistical criteria such as minimum description length (MDL) (Rissanen, 1978; Marsland et al., 2008), and minimum message length (MML) (Wallace and Dow, 1999; Bougula and Ziou, 2006). However, the entire parameter estimation for Gaussian mixtures has to be repeated at different values of $k$, and the process of evaluating these criteria incurs a large computational cost. Moreover, since we usually use the EM algorithm to make the estimation of mixture parameters at each given $k$, the final model selection may be misled by the local convergence problem of the EM algorithm, which will be shown in the later experiments. Other than this deterministic criterion-based method, there are some even more computationally demanding methods to solve the model selection problem. That is, we can take into account the stochastic simulation (Roberts et al., 2001; Kato, 2008) and resampling (McLachlan, 1987; McLachlan and Khan, 2004) methods to infer the optimal mixture model.

Actually, some clustering methods have been developed to solve the model selection problem. One influential work is clustering via competitive agglomeration (Frigui and Krishnapuram, 1997; Grira et al., 2005), which includes a regularization term defined by the memberships of samples in the cost function of fuzzy $c$-means (de Carvalho, 2007; Hung et al., 2008) to determine the number of clusters automatically. Following this idea of competitive agglomeration, we can regularize the cost function (i.e., likelihood) of Gaussian mixture learning by the full entropy of posterior probabilities (the analogy to memberships of samples). Here, the full entropy is used to control the model complexity of Gaussian mixtures, which is slightly different from the original competitive agglomeration (Frigui and Krishnapuram, 1997). Of course, we could consider other regularization terms such as the Dirichlet and entropic priors (Figueiredo and Jain, 2002; Brand, 1999). However, these priors are defined only by the mixing probabilities of Gaussian mixtures and then not able to result in a close-form solution. In this paper, similar to the EM algorithm, two robust close-form algorithms are developed to implement the proposed mechanism of competitive agglomeration on Gaussian mixtures.

Our main contribution is proposing two robust competitive agglomeration (RCA) algorithms for Gaussian mixture learning which are: (1) able to automatically select the number of components (i.e., automatic model selection), and (2) less sensitive to initialization than EM. We also give some asymptotic analysis of our
algorithms and show in theory that such robust competitive agglomeration leads to automatic model selection on Gaussian mixtures. Moreover, the results of handwritten digit recognition on the US Postal Service (USPS) handwritten digit database (Hull, 1994) and unsupervised image segmentation on the Berkeley segmentation database (Martin et al., 2001) demonstrate that our algorithms outperform other related works in the literature such as the MDL based EM (MDL-EM) algorithm (Carson et al., 2002) and the MML based EM (MML-EM) algorithm (Figueiredo and Jain, 2002) even in these challenging applications.

The remainder of this paper is organized as follows. In Sections 2 and 3, we apply competitive agglomeration to Gaussian mixture learning and then develop two robust algorithms. In Section 4, some asymptotic analysis is followed to recover the convergence properties of robust competitive agglomeration. Section 5 then presents the results of the simulation and real-life experiments by robust competitive agglomeration. Finally, Section 6 gives the conclusions drawn from our experimental results using the developed algorithms.

2. Competitive agglomeration for Gaussian mixture learning

We first give a simple introduction of Gaussian mixture model. A random variable $x \in \mathbb{R}^d$ is said to follow a $k$-component Gaussian mixture distribution, if its probability density function (pdf) can be written as:

$$p(x|\Theta_k) = \sum_{l=1}^k \pi_l p(x|\theta_l), \quad \sum_{l=1}^k \pi_l = 1, \quad \pi_l \geq 0,$$

$$p(x|\theta_l) = \frac{1}{(2\pi)^{d/2} |\Sigma_l|^{1/2}} e^{-\frac{1}{2} (x-\mu_l)^T \Sigma_l^{-1} (x-\mu_l)},$$

where $\{\pi_l\}_{l=1}^k$ is the set of mixing probabilities in the mixture, each $p(x|\theta_l)$ is a Gaussian pdf with the mean vector and covariance matrix $\theta_l = (\mu_l, \Sigma_l)$, and $\Theta_k = \{\pi_l, \theta_l\}_{l=1}^k$ is the complete set of mixture parameters.

To learning a $k$-component Gaussian mixture model $p(x|\Theta_k)$ from multivariate data, we can define the objective function as follows. Given a set of $N$ independent samples $S = \{x_l\}_{l=1}^N$, the negative log-likelihood is given by

$$L(\Theta_k) = -\frac{1}{N} \sum_{l=1}^N \ln \left( \sum_{l=1}^k \pi_l p(x_l|\theta_l) \right).$$

The well-known EM algorithm for Gaussian mixture learning can be regarded as a process of minimizing $L(\Theta_k)$. That is, $L(\Theta_k)$ is used as the objective function of the EM algorithm.

We now apply competitive agglomerate (Frigui and Krishnapuram, 1997; Girra et al., 2005) to Gaussian mixture learning. It should be noted that the original competitive agglomerate aim to determine the number of clusters automatically by including a regularization term defined by the memberships of samples in the cost function of the standard fuzzy $k$-means. We can follow this idea and regularize the cost function (i.e., likelihood) of Gaussian mixture learning by the full entropy of posterior probabilities (the analogy to memberships of samples). Here, the full entropy is used to control the model complexity of Gaussian mixtures. In the following, we will elaborate this extension of competitive agglomeration.

Since we can obtain the posterior probability that $x_l$ arises from the $l$th component in the Gaussian mixture model:

$$p(l|x_l) = \frac{p(x_l|\theta_l)\pi_l}{\sum_{j=1}^k p(x_l|\theta_j)\pi_j},$$

where $l = 1, \ldots, k$, the discrete Shannon entropy of these posterior probabilities for the sample $x_l$ is calculated as:

$$E(x_l) = -\frac{1}{N} \sum_{l=1}^k p(l|x_l) \ln p(l|x_l).$$

According to information theory, this entropy has a nice property: $E(x_l)$ is globally minimized at $p(l|x_l) = 1, p(l|x_l) = 0 (l \neq l_0)$, i.e., the sample $x_l$ is deterministically classified into the $l_0$th component in the mixture model.

We now consider the average Shannon entropy of the Gaussian mixture model $p(x|\Theta_k)$ over the sample set $S$:

$$E(\Theta_k) = -\frac{1}{N} \sum_{l=1}^N E(x_l) = -\frac{1}{N} \sum_{l=1}^N \sum_{l=1}^k p(l|x_l) \ln p(l|x_l),$$

and use it to regularize the log-likelihood function by $H(\Theta_k) = L(\Theta_k) + \gamma E(\Theta_k)$.

where $\gamma > 0$ is the regularization factor. That is, the entropy $E(\Theta_k)$ is a regularization term for competitive agglomeration on Gaussian mixtures to reduce the model complexity. Although the Dirichlet and entropic priors have also been used as regularization terms to penalize the likelihood $L(\Theta_k)$ in (Brand, 1999; Figueiredo and Jain, 2002), these priors are defined only by the mixing probabilities $\{\pi_l\}_{l=1}^k$ of the Gaussian mixture model and then not able to result in a close-form solution.

3. Two robust competitive agglomeration algorithms

In this section, we propose two robust competitive agglomeration algorithms for Gaussian mixture learning, which both take a form similar to the EM algorithm. These two algorithms differ only in the way to estimate the posterior probabilities $P(l|x_l) (l = 1, \ldots, k)$. On one hand, when the posterior probabilities are considered as functions of mixture parameters $\Theta_k$ according to Eq. (4), they can be estimated by $\min H(\Theta_k)$, i.e., first estimate $\Theta_k$ and then $P(l|x_l)$. This robust competitive agglomeration algorithm is denoted as RCA1 in the following. On the other hand, when the posterior probabilities are considered as part of mixture parameters (i.e., we need to learn both $\Theta_k = (P(l|x_l), l = 1, \ldots, N, \ l = 1, \ldots, k)$ and $\Theta_k$, they can be estimated by $\min H(\Theta_k|\Theta_k)$. This robust competitive agglomeration algorithm is denoted as RCA2 in the following. In the next section, we will show that the mechanism of competitive agglomeration can make the proposed two algorithms avoid some types of local optima, which is the reason why RCA is called. Moreover, the robust initialization of the proposed two algorithms is also provided in detail at the end of this section.

3.1. RCA1 algorithm

We first derive a robust competitive agglomeration algorithm (i.e., RCA1) when the posterior probabilities are considered as functions of mixture parameters $\Theta_k$. At each iteration with $\gamma$ fixed, since $\sum_{l=1}^k \pi_l = 1$, we can compose the following Lagrange function for the task of $\min H(\Theta_k)$:

$$Q(\Theta_k, \lambda) = H(\Theta_k) + \lambda \left( \sum_{l=1}^k \pi_l - 1 \right).$$

Using the general methods for matrix derivatives with respect to $\Theta_k$ and $\lambda$, we are led to the following series of equations:

$$\frac{\partial Q}{\partial \pi_l} = -\frac{1}{N} \sum_{l=1}^N \frac{U(l|x_l) + \lambda}{\pi_l},$$

$$\frac{\partial Q}{\partial \pi_l} = -\frac{1}{N} \sum_{l=1}^N \pi_l U(l|x_l) \Sigma_l^{-1} (x_l - m_l).$$
\[
\frac{\partial Q}{\partial \alpha_i} = \frac{1}{2N} \sum_{t=1}^{N} U(l|x_t) \Sigma_t^{-1} (M_t - \Sigma_t)^{-1},
\]
(11)

\[
\frac{\partial Q}{\partial \beta} = \frac{1}{k} \alpha_i - 1, \tag{12}
\]

where \(M_t = (x_t - m_i)(x_t - m_i)^T\) and

\[
U(l|x_t) = P(l|x_t)(1 + \ln P(l|x_t) + E(x_t|\Theta_k)).
\]
(13)

By setting these derivatives of \(Q(\Theta_k, \lambda)\) to be zeros, we then have:

\[
\hat{\alpha}_i = \sum_{l=1}^{N} U(l|x_t) / \sum_{l=1}^{N} \sum_{j=1}^{N} U(l|x_t).
\]
(14)

\[
\hat{\beta}_i = \sum_{l=1}^{N} U(l|x_t) / \sum_{l=1}^{N} U(l|x_t).
\]
(15)

\[
\hat{\Sigma}_i = \sum_{l=1}^{N} U(l|x_t) M_t / \sum_{l=1}^{N} U(l|x_t).
\]
(16)

These explicit expressions actually give us a robust competitive agglomeration algorithm: at each iteration, we first update \(P(l|x_t)\) according to Eq. (4), and then update \(\Theta_k\) according to Eqs. (14)-(16). Though similar to the standard EM algorithm, this RCA1 algorithm makes some important modification of the M-step. That is, during updating \(\Theta_k\), the mechanism of competitive agglomeration is enforced among all the components, which then leads to automatic model selection on Gaussian mixtures.

3.2. RCA2 algorithm

We further derive another robust competitive agglomeration algorithm (i.e., RCA2) when the posterior probabilities are considered as part of mixture parameters \(\Theta_k^p = (P(l|x_t), t = 1, \ldots, N, l = 1, \ldots, k)\). At each iteration with \(\gamma\) fixed, the task of \(\min H(\Theta_k^p, \Theta_k)\) can be implemented by a two-step minimization procedure similar to the EM algorithm: (1) E-step: fix \(\Theta_k\), \(\Theta_k^p = \arg \min H(\Theta_k^p, \Theta_k)\); and (2) M-step: fix \(\Theta_k^p\), \(\Theta_k = \arg \min H(\Theta_k^p, \Theta_k)\).

For the E-step of RCA with \(\Theta_k\) fixed, we can compose the following Lagrange function using N Lagrange multipliers \(\lambda = (\lambda_1, \ldots, \lambda_N)\):

\[
Q(\Theta_k^p, \lambda) = H(\Theta_k^p) + \frac{1}{N} \sum_{t=1}^{N} \lambda_t \left( \sum_{l=1}^{N} P(l|x_t) - 1 \right). \tag{17}
\]

By setting the derivatives of \(Q(\Theta_k^p, \lambda)\) with respect to \(P(l|x_t)\) and \(\lambda_t\) to be zeros, we have

\[
\ln P(x_t|\theta|\lambda) + (\gamma - 1)(1 + \ln P(l|x_t)) = \lambda_t,
\]
(18)

\[
\sum_{l=1}^{N} P(l|x_t) = 1. \tag{19}
\]

From the above equations, we can then obtain the following solution to estimate \(P(l|x_t)\):

\[
\hat{P}(l|x_t) = (p(x_t|\theta)|\lambda_t)^{\gamma - 1} \left( \sum_{l=1}^{N} p(x_t|\theta)|\lambda_t \right)^{-\gamma - 1}, \tag{20}
\]

which is the Gibbs distribution. If \(0 < \gamma < 1\), the Gibbs distribution is more peaked than the estimated posterior according to Eq. (4). That is, the belonging component of \(x_t\) is forced to become clearer, which exactly coincides with our idea to introduce competitive agglomeration.

For the M-step of RCA with \(\Theta_k^p\) fixed, \(\hat{\Theta}_k = \arg \min H(\Theta_k^p, \Theta_k)\) is equivalent to \(\hat{\Theta}_k = \arg \min L(\Theta_k)\). Since the EM algorithm is an implementation of minimizing \(L(\Theta_k)\), the update rules for \(\Theta_k\) can be set the same form as the M-step of the EM algorithm:

\[
\hat{\alpha}_i = \frac{1}{N} \sum_{l=1}^{N} \hat{P}(l|x_t), \tag{21}
\]

\[
\hat{\beta}_i = \sum_{l=1}^{N} \hat{P}(l|x_t) / \sum_{l=1}^{N} \hat{P}(l|x_t). \tag{22}
\]

\[
\hat{\Sigma}_i = \sum_{l=1}^{N} \hat{P}(l|x_t) M_t / \sum_{l=1}^{N} \hat{P}(l|x_t). \tag{23}
\]

Through the above two-step minimization procedure, we have actually presented another robust competitive agglomeration algorithm: at each iteration, we first update \(P(l|x_t)\) according to Eq. (20), and then update \(\Theta_k\) according to Eqs. (21)-(23). Though similar to the standard EM algorithm, this RCA algorithm makes some important modification of the E-step. That is, during updating \(\Theta_k\), the mechanism of competitive agglomeration is implemented on these posterior probabilities using the Gibbs distribution, which are more peaked than the estimated posterior according to Eq. (4). Hence, the belonging component of each sample \(x_t\) is forced to become clearer, which leads to automatic model selection on Gaussian mixtures.

3.3. Robust initialization

We always implement robust competitive agglomeration with \(k = k_{\text{max}}\) and \(\gamma = \gamma_{\text{max}}\). That is, the number of components \(k\) is initialized a large value \(k_{\text{max}}\) to make sure \(k > k\) (\(k\) is the true number of components) and then we annihilate those components with \(\chi_t\) reduced below a threshold \(T\) (e.g., \(T = 0.01\)) after certain iterations, while the regularization factor \(\gamma\) is gradually reduced from \(\gamma_{\text{max}}\) to zero by \(\gamma = \gamma_{\text{max}}(1 + ct)\) where \(t\) denotes the number of iterations and we set \(c = 0.1\). Moreover, the mean vectors and covariance matrices of the mixture components are initialized by some clustering methods (e.g., k-means). In the experiments, our two algorithms are stopped if \(||H(\hat{\Theta}_k) - H(\hat{\Theta}_{k-1})||/H(\hat{\Theta}_k)) < 10^{-4}\). Here, it should be noted that the above initialization method and convergence criterion are also used similarly by MDL-EM and MML-EM.

Since the strength of competitive agglomeration during iteration is controlled by \(\gamma = \gamma_{\text{max}}\), according to Eq. (7) in the Appendix it is important to select this parameter appropriately for identifying the correct number of components. When the mixture overlap is lower, we can select \(\gamma_{\text{max}}\) in a large range (e.g., [0.5, 0.8]) and the number of components can be correctly determined after only one run of robust competitive agglomeration. However, when the mixture overlap becomes high, the run of robust competitive agglomeration may drop some components if \(\gamma_{\text{max}}\) is large.

We can solve this problem by introducing the MDL criterion into the run of robust competitive agglomeration, which always starts at small \(\gamma_{\text{max}}\) (e.g., \(\gamma_{\text{max}} \in [0.05, 0.50]\)). If the selected component number \(k > k_{\text{max}}\) after convergence, we can evaluate this candidate model using MDL and then restart the run of robust competitive agglomeration with \(k = k - 1\) by annihilating the least probable component with smallest \(\chi_t\) (i.e., in a similar way as Figueiredo and Jain, 2002). We try to find the optimal model with smallest MDL via repeating the run of robust competitive agglomeration until \(k = k_{\text{max}}\). Such kind of robust competitive agglomeration is different from the traditional approach (e.g., MDL-EM in Carson et al., 2002) in that we only need make a few MDL evaluations since there may be some components annihilated during each run of robust competitive agglomeration.
4. Asymptotic analysis of robust competitive agglomeration

In this section, we try to give an asymptotic analysis of the above RCA, and then prove the promising property of automatic model selection on Gaussian mixtures when there is a certain degree of overlap in the mixture.

Due to the randomness in the sample set, we have to consider RCA asymptotically, i.e., we let $N \to \infty$. The object function $H(\hat{\theta}_k)$ of RCA estimated on the sample set in Eq. (7) is rewritten as $H_n(\hat{\theta}_k)$. Likewise, the estimated functions $L(\hat{\theta}_k)$ and $E(\hat{\theta}_k)$ in Eqs. (3) and (6) are also rewritten as $L_n(\hat{\theta}_k)$ and $E_n(\hat{\theta}_k)$, respectively. According to the probability theory, we have

$$ H(\hat{\theta}_k) = \lim_{n \to \infty} H_n(\hat{\theta}_k) = \lim_{n \to \infty} (L_n(\hat{\theta}_k) + \gamma E_n(\hat{\theta}_k)) = L(\hat{\theta}_k) + \gamma E(\hat{\theta}_k), $$

and now $L(\hat{\theta}_k)$ and $E(\hat{\theta}_k)$ are updated as

$$ L(\hat{\theta}_k) = \lim_{n \to \infty} L_n(\hat{\theta}_k) = -\int p(x|\hat{\theta}_k') \ln p(x|\hat{\theta}_k) dx, $$

$$ E(\hat{\theta}_k) = \lim_{n \to \infty} E_n(\hat{\theta}_k) = \int E(x|\hat{\theta}_k)p(x|\hat{\theta}_k') dx, $$

where $\hat{\theta}_k' = \{x_i', \theta_i'\}_{i=1}^{k'}$ denotes the set of the true parameters in Gaussian mixtures which the sample data come from. Specifically, $k'$ is the number of the actual components and $\{x_i, \theta_i\}$ is the set of true parameters of the $l$th component for the actual mixture pdf.

In the following, we give asymptotic analysis of RCA in the case that the Gaussian mixture model $p(x|\theta_k')$ has a certain degree of component overlap. According to information theory, $E(x|\hat{\theta}_k')$ is high when the belonging component of $x$ is obscure, i.e., the component overlap is large; otherwise, $E(x|\hat{\theta}_k')$ is low when the belonging component of $x$ is clear, i.e., the component overlap is small. Hence, the average entropy $E(x|\hat{\theta}_k')$ can be used to measure the overlap of the Gaussian mixture model. In this paper, we assume that the overlap of the true Gaussian mixture model $p(x|\theta_k')$ should not be too high, i.e., the average entropy $E(x|\hat{\theta}_k')$ should be constrained as $E(x|\hat{\theta}_k') < M \ll \ln k'$. Note that the true Gaussian mixture model $p(x|\theta_k')$ will tend to the maximum overlap $E(x|\hat{\theta}_k') = \ln k'$, when $P(lk) = 1/k', l = 1, \ldots, k'$ at each data $x$.

Moreover, the Gaussian mixture model we consider here is assumed to be identifiable. That is, in the cases that all the components in the Gaussian mixture are different, $p(x|\theta_k) = p(x|\theta_k')$ if and only if $\theta_k \supseteq \theta_k'$ with $k \geq k'$ and the mixing probabilities of the other $k' - k'$ extra components in $\theta_k$ being zeros. We now investigate the asymptotic convergence properties of RCA on the Gaussian mixture model $p(x|\theta_k)$ and have the following theorem (see the Appendix for the proof).

**Theorem 1.** Suppose that the Gaussian mixture model $p(x|\theta_k)$ is identifiable, and the overlap of the true Gaussian mixtures $p(x|\theta_k')$ is not too high, i.e., $|E(x|\hat{\theta}_k')| = 0$. If $\theta_k^0(\gamma) = \arg \min_{\theta_k} H(\hat{\theta}_k)$, we have $\theta_k^0(\gamma) \supseteq \theta_k'$ with $k' \geq k'$ and the mixing probabilities of the other $k' - k'$ components in $\theta_k^0(\gamma)$ being zeros, when the regularization factor $\gamma \to 0$.

According to Theorem 1, we can find that robust competitive agglomeration with $\gamma \to 0$ leads to automatic model selection on Gaussian mixtures if we let $k > k'$ and annihilate the components with negligible mixing probabilities. That is, if the model scale is actually defined by the number of positive mixing probabilities in a Gaussian mixture model, it will be equal to $k'$ via minimizing $H(\hat{\theta}_k)$. Thus, the true model scale can be correctly detected through the minimization of $H(\hat{\theta}_k)$.

Though competitive agglomeration is originally introduced into the maximum likelihood estimation to resolve the model selection problem, we can also find that minimum $H(\hat{\theta}_k)$ may escape some types of local minima and then avoid the initialization dependence. That is, when local minima of the negative likelihood $L(\theta_k)$ arise during minimizing $H(\hat{\theta}_k)$, the average entropy $E(\hat{\theta}_k)$ may still keep large and these local minima may then be avoided. For example, the EM algorithm may not escape one type of local minima when two or more components in the mixture have similar parameters (i.e., the overlap $E(\hat{\theta}_k)$ is high) and then share the same data. However, robust competitive agglomeration can promote the competition among these components by minimum $H(\hat{\theta}_k)$, and only one of them will survive with the other annihilated.

5. Experimental results

Although the two RCA algorithms can be used for other types of mixture models, we only consider Gaussian mixtures to present their performances in this section. To make comparison with other related works in the literature such as MDL-EM (Carson et al., 2002) and MML-EM (Figueiredo and Jain, 2002), we first make simulation experiments on three sample sets generated from Gaussian mixtures. Moreover, we apply the two RCA algorithms to handwritten digit recognition on the USPS handwritten digit database (Hull, 1994) and unsupervised image segmentation on the Berkeley segmentation database (Martin et al., 2001).

5.1. Simulation results

To present the performance of the RCA algorithms, we first carry out simulation experiments on the sample data set of $N = 1000$.
samples randomly generated from a bivariate 4-component Gaussian mixture as shown in Fig. 1a. The detailed parameters of this mixture can be found in (Figueiredo and Jain, 2002). Note that there are two components completely overlapped with the other one in this mixture and then they are difficult to identify.

In the second example, we use \( N = 1200 \) samples randomly generated from a bivariate 6-component Gaussian mixture given by Fig. 1b. The parameters of this mixture are:

\[
\begin{align*}
\alpha_1 & = \alpha_2 = 0.25, \quad \alpha_1 = \alpha_2 = 0.125, \\
\beta_1 & = [-4, 0]^T, \quad \beta_2 = [0, -6]^T, \quad \beta_3 = [0.8, 3.2]^T, \\
\beta_4 & = [3.2, 0.8]^T, \quad \beta_5 = [-1, -2]^T, \quad \beta_6 = [-3, -4]^T, \\
\Sigma_1 & = \Sigma_2 = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}, \quad \Sigma_3 = \Sigma_4 = \begin{bmatrix} 0.5 & -0.25 \\ -0.25 & 0.5 \end{bmatrix}, \\
\Sigma_5 = \Sigma_6 & = \text{diag}(0.25, 0.25).
\end{align*}
\]

The situation in this Gaussian mixture is also complicated, i.e., two smallest clusters \( l = 5, 6 \) may be considered one Gaussian component as compared with their neighbors \( l = 1, 2 \).

In the following experiments, we set \( k_{\text{max}} = 2 \) and \( k_{\text{max}} = 20 \) for all the four algorithms. As for the two RCA algorithms, we simply set \( \gamma_{\text{max}} = 0.2 \). Moreover, we run all the four algorithms (Matlab code) on a Pentium D 2.8 GHz computer with 1.0 GB memory. To make an objective evaluation, we further list the average estimated number of components, the failure rate of identifying the \( k \) true components, and the average running time over 250 independent trials for fitting the two mixtures given by Fig. 1 in Tables 1 and 2, respectively. Note that only the average estimated number of components is not enough for evaluating the four algorithms. For example, the values of this criterion are little different for fitting components is not enough for evaluating the four algorithms. For this reason, we use the average estimated number of components, the failure rate of identifying the \( k \) true components, and the average running time over 250 independent trials for fitting the two mixtures given by Fig. 1 in Tables 1 and 2, respectively. Note that only the average estimated number of components is not enough for evaluating the four algorithms. For example, the values of this criterion are little different for fitting the second mixture by the four algorithms. However, we actually find that the MDL-EM algorithm mostly converges to the local optima in the experiments. Hence, the failure rate of identifying the \( k \) true components is also used for the evaluation of the four algorithms.

In terms of the ability of finding the global optimal solution (i.e., identifying the true number of components and simultaneously avoiding local optima), we can find from Tables 1 and 2 that the two RCA algorithms perform significantly better than the MDL-EM and MML-EM algorithms. Here, the highest failure rate obtained by MDL-EM on the second data set is due to that the standard EM algorithm mostly converges to the local optima on this data set. More concretely, since the MDL criterion is computed based on the estimation of mixture parameters by the standard EM algorithm, the final model selection using MDL also suffers from the local convergence problem. On the contrary, our two RCA algorithms and MML-EM succeed in avoiding the local optima on this data set.

In order to further explain the mechanism of automatic detection of the component numbers, we present the evolution of RCA (only RCA1 is considered) during fitting the first mixture from one of the successful trials. As shown in Fig. 2, we can observe that when two or more components fall into the same data (at a high probability), RCA can promote the competition among them and then make only one of them survive with the other annihilated, which supports our theoretic analysis given in Section 4.

When the computational cost is concerned, we can conclude from Tables 1 and 2 that the MML-EM algorithm is the lowest, the two RCA algorithms are some more, and the MDL-EM algorithm is the highest. Though our RCA algorithms incur more computational cost than MML-EM, they perform much better for identifying the true number of components. Hence, we think our RCA algorithms are preferred if we make an overall comparison.

### 5.2. Handwritten digit recognition

The performance of our two RCA algorithms is also tested on real-life data. More specifically, we apply our two RCA algorithms to the USPS handwritten digit database (Hull, 1994). Some examples for the digits 0–4 are shown in Fig. 3. Each digit is an image of \( 16 \times 16 \) pixels, with 256 gray-scale intensities. We create a data set using 75% samples for each of the digits 0–4 (totally 4071 samples), and then cluster the data with a single mixture model. The dimension of the intensity feature vector is originally 256 and then reduced to 20 using principal component analysis. The data set is further normalized, so that the mean is zero and the variance unit in each dimension.

After fitting a single Gaussian mixture to the data with each algorithm, we compare them in terms of the “classification” error rate using previously unseen test samples (25% samples for each digit, i.e., totally 1356 samples). To compute this error, we assign each training sample to the component with maximum responsibility (i.e. the posterior probability given by Eq. (4)). After this hard clustering, we label each component with the class of the majority of its data. In order to classify a test sample, its responsibility with respect to each component is computed, and the class label of the component with maximum responsibility is assigned to it.

We select \( k_{\text{max}} = 5 \) and \( k_{\text{max}} = 30 \) for each algorithm, while the other parameters are set in the same way as the above simulation. The average number of components, error rate, and running time over 60 independent trials for handwritten digit recognition by the four algorithms are listed in Table 3. The 95% confidence interval is also provided for the error rate. It can be observed that our two RCA algorithms perform better than the MDL-EM and MML-EM algorithms since they lead to the smallest error rates. More importantly, considering the associated 95% confidence intervals, we can conclude that our two RCA algorithms (particularly RCA2) achieve obvious improvements. In addition, we can find that our two RCA algorithms incur less computational cost than MDL-EM.

### 5.3. Unsupervised image segmentation

Finally, we apply our two RCA algorithms to unsupervised image segmentation which aims to automatically determine the number of regions (objects) in an image during segmentation. Note that unsupervised image segmentation plays an important role in region-based image retrieval, since the image databases are often huge in this application and the prior setting of region number for each image is no longer feasible. In the following, we aim to resolve this model selection problem by the proposed two RCA algorithms.
The Berkeley segmentation database (Martin et al., 2001) is used to test our RCA algorithms in the application of unsupervised image segmentation. This benchmark has 300 images along with human (ground truth) segmentations by different individuals. The evaluation of our segmentation algorithms can be achieved by the adjusted Rand (AR) index (Hubert and Arabie, 1985) which takes values between $-1$ and $1$, and a higher AR score indicates that a higher percentage of pixel pairs in a test segmentation have...
Under this hypergeometric model, the AR index between a test segmentation $S_{\text{test}}$ of an image and a set of $K$ ground truth segmentations $S_k = (S_{k1}, S_{k2}, \ldots, S_{kK})$ can be given by:

$$
AR(S_{\text{test}}, S_k) = \frac{1}{K} \sum_{k=1}^{K} \frac{\sum_{i<j} n_i^{ij} - \sum_i n_i^j \sum_j n_j^i / \binom{N}{2}}{\frac{1}{2} \sum_i n_i^j + \sum_i n_i^* / \binom{N}{2} - \sum_i \sum_j n_i^j / \binom{N}{2}}.
$$

where $n_i$ is the number of pixels in region $R_i$ according to $S_{\text{test}}$, $n_i^j$ is the number of pixels in region $R_j$ according to $S_k$, $n_i^*$ is the number of pixels that are in region $R_i$ according to $S_{\text{test}}$ and also in region $R_j$ according to $S_k$, and $N$ is the total number of pixels in the image.

In the segmentation, we consider an 8-dimensional vector of color, texture, and position features for each pixel of an image, and these features are obtained just as Carson et al. (2002). The three color features are the coordinates in the $L'a'b'$ color space, and we smooth these features to avoid over-segmentation arising from local color variations due to texture. The three texture features are contrast, anisotropy, and polarity, which are extracted at an automatically selected scale. The position features are simply the $(x, y)$ position of the pixel. Once pixels in an image with these combined features are grouped into regions by our RCA algorithms, we further merge those regions smaller than 1 percent of the image with the adjacent regions if they are similar in the color/texture feature space.

In all the experiments, we simply set $\gamma_{\text{max}} = 0.3$ for our RCA algorithms. Moreover, we select the region number in $[2, 8]$ (i.e., $k_{\text{min}} = 2$ and $k_{\text{max}} = 8$) for the four algorithms. We then compare these algorithms on the whole database, and their average AR indices for all the 300 images are listed in Table 4. The 95% confidence interval is also provided for the AR index. Just as our theoretic

Table 3
The average number of components, error rate, and running time for handwritten digit recognition. The 95% confidence interval is also provided for the error rate.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th># Components</th>
<th>Error rate (%)</th>
<th>Running time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDL-EM</td>
<td>9.6</td>
<td>8.0 ± 0.3</td>
<td>341.6</td>
</tr>
<tr>
<td>MML-EM</td>
<td>9.3</td>
<td>6.8 ± 0.5</td>
<td>278.8</td>
</tr>
<tr>
<td>RCA1</td>
<td>13.3</td>
<td>5.7 ± 0.4</td>
<td>303.8</td>
</tr>
<tr>
<td>RCA2</td>
<td>14.2</td>
<td>5.0 ± 0.4</td>
<td>325.0</td>
</tr>
</tbody>
</table>

Table 4
The average AR index and running time for segmenting all the 300 images in the Berkeley segmentation database. The 95% confidence interval is also provided for the AR index.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>AR index</th>
<th>Running time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDL-EM</td>
<td>0.443 ± 0.021</td>
<td>270.0</td>
</tr>
<tr>
<td>MML-EM</td>
<td>0.413 ± 0.023</td>
<td>33.6</td>
</tr>
<tr>
<td>RCA1</td>
<td>0.474 ± 0.022</td>
<td>50.2</td>
</tr>
<tr>
<td>RCA2</td>
<td>0.493 ± 0.021</td>
<td>88.1</td>
</tr>
</tbody>
</table>

Fig. 4. Segmentation results for some sample images by the four segmentation algorithms. The first row is the original images. The second to fifth rows are the results obtained by MDL-EM, MML-EM, RCA1, and RCA2, respectively.
analysis in Section 4, our RCA algorithms outperform MDL-EM and MML-EM due to robust competitive agglomeration. More importantly, considering the associated 95% confidence intervals, we can conclude that our RCA algorithms (particularly RCA2) achieve obvious improvements.

Some segmentation samples are also shown in Fig. 4, and we can find that our RCA algorithms successfully detect the object of interest even from the confusing background. However, the MDL-EM algorithm may converge to local optima and the background may be split into two regions (see images #8023, #249061, and #271031), while the MML-EM algorithm can not control the strength of component annihilation and the object of interest may be merged with other regions (see images #94079, #249061, and #271031).

The average running time taken by the four algorithms on the Berkeley segmentation database is also listed in Table 4. As expected, our algorithms run much faster than MDL-EM which incurs many evaluations of the MDL criterion. As compared with MML-EM, our algorithms can be considered computationally comparable. If the segmentation results are also taken into account, we can conclude that our algorithms perform generally better.

6. Conclusion

We have investigated the model selection and parameter estimation for Gaussian mixtures through implementing robust competitive agglomeration. Our asymptotic analysis of RCA then shows that robust competitive agglomeration can lead to automatic model selection on Gaussian mixtures and also make our two RCA algorithms less sensitive to initialization than the standard EM algorithm. The simulation and real-life experiments then demonstrate that our two RCA algorithms can provide superior results just as our theoretic analysis. Moreover, we can also find that our two RCA algorithms generally achieve comparable results. Particularly, in the two real-life applications, RCA2 performs better than RCA1, but incurs more computational cost.

Although our RCA algorithms outperform MML-EM, they are less efficient in terms of the running time. For future work, we need to adapt other fast optimization techniques to implement our RCA. Since our RCA has only been used for Gaussian mixture learning in this paper, we will further apply it to other types of mixture models such as mixtures of experts for supervised learning and mixtures of factor analyzers (Utsugi and Kumagai, 2001) for dimensionality reduction. As for mixture model-based image segmentation, our algorithms can be extended to incorporate the spatial information into model estimation just as Diplaros et al. (2007). Moreover, since we can implement region-based image retrieval if the segmented regions are assigned with color/texture features, our algorithms can further be evaluated in this application.

Acknowledgements

The work described in this paper was fully supported by the National Natural Science Foundation of China under Grant Nos. 60873154 and 60503062, the Beijing Natural Science Foundation of China under Grant No. 4082015, the Program for New Century Excellent Talents in University under Grant No. NCET-06–0009, and the National Development and Reform Commission High-tech Program of China under Grant No. 2008-2441.

Appendix

This appendix presents the proof of Theorem 1 in Section 4. We begin with $\theta_i^*(\gamma) = \arg \min_{\theta_i} H(\theta_i)$ at each iteration with $\gamma$ fixed, and have $H(\theta_i^*(\gamma)) \leq H(\theta_i^*)$. It then follows from Eq. (24) that

$$L(\theta_i^*(\gamma)) - L(\theta_i^*) \leq \gamma(E(\theta_i^*) - E(\theta_i^*(\gamma))).$$

Under information theory, $E(\theta_i^*) \geq 0$ and $E(\theta_i^*(\gamma)) \geq 0$. Hence, with $E(\theta_i^*) < M$, it follows that $0 \leq E(\theta_i^*) < M$. According to Eq. (26), we have

$$L(\theta_i^*(\gamma)) - L(\theta_i^*) \leq \gamma E(\theta_i^*) < \gamma M.$$  

The difference between $\theta_i^*(\gamma)$ and $\theta_i^*$ can be measured by

$$D_{KL}(p(x|\theta_i^*(\gamma)), p(x|\theta_i^*) = \int p(x|\theta_i^*) \ln \frac{p(x|\theta_i^*)}{p(x|\theta_i^*(\gamma))} dx,$$

where $D_{KL}(\cdot, \cdot)$ is the Kullback-Leibler distance between two probability densities and it always keeps $D_{KL}(\cdot, \cdot) \geq 0$. Since $D_{KL}(p(x|\theta_i^*(\gamma)), p(x|\theta_i^*) = L(\theta_i^*(\gamma)) - L(\theta_i^*)$, according to Eq. (27), we have

$0 \leq D_{KL}(p(x|\theta_i^*(\gamma)), p(x|\theta_i^*)) < \gamma M$. 

When $\gamma \rightarrow 0$, $D_{KL}(p(x|\theta_i^*(\gamma)), p(x|\theta_i^*) = 0$, i.e., $p(x|\theta_i^*) = p(x|\theta_i^*(\gamma))$ under information theory. Based on the identifiability of the Gaussian mixture model, we then have $\theta_i^*(\gamma) \geq \theta_i^*$ with $\kappa^2 \geq \kappa^1$ and the mixing probabilities of the other $\kappa^2 - \kappa^1$ components in $\theta_i^*(\gamma)$ being zeros, when the regularization factor $\gamma \rightarrow 0$.

References


