Refining Gaussian mixture model based on enhanced manifold learning

Jianfeng Shen a,b,c, Jiajun Bu d, Bin Ju b, Tao Jiang b, Hao Wu d, Lanjuan Li a,*

a State Key Laboratory for Diagnosis and Treatment of Infectious Diseases, The First Affiliated Hospital, College of Medicine, Zhejiang University, No. 79, Qingchun Road, Hangzhou, Zhejiang Province 310003, PR China
b Zhejiang Province Health Bureau, Center of Information, PR China
c Zhejiang Medical College, PR China
d College of Computer Science, Zhejiang University, PR China

A R T I C L E  I N F O
Article history:
Received 30 November 2010
Received in revised form 23 January 2012
Accepted 31 January 2012
Communicated by X. Gao
Available online 3 March 2012

Keywords:
Gaussian mixture model
Clustering
Manifold learning

A B S T R A C T
Gaussian mixture model (GMM) has been widely used for data analysis in various domains including text documents, face images and genes. GMM can be viewed as a simple linear superposition of Gaussian components, each of which represents a data cluster. Recent models, namely Laplacian regularized GMM (LapGMM) and locally consistent GMM (LCGMM) have been proposed to preserve the local manifold structure of the data for modeling Gaussian mixtures, and show superior performance than the original GMM. However, these two models ignore the global manifold structure without consideration of the widely separated points. In this paper, we introduce refined Gaussian mixture model (RGMM), which explicitly places separated points far apart from each other as well as brings nearby points closer together according to the probability distributions of Gaussians, in the hope of fully discovering the discriminating power of manifold learning for estimating Gaussian mixtures. We use EM algorithm to optimize the maximum likelihood function of RGMM. Experimental results on three real-world data sets demonstrate the effectiveness of RGMM in data clustering.

© 2012 Elsevier B.V. All rights reserved.

1. Introduction

Gaussian mixture model (GMM [1]) is one of the most popular methods for data analysis [2–4]. Compared to the single Gaussian, GMM uses a mixture of Gaussians, which gives a better characterization of real data by taking a linear combination of Gaussian densities. Each Gaussian density is called a component of the mixture which has its own mean and covariance. GMM is also usually used to cluster data. We can introduce discrete latent variables, which define cluster assignments of data points to specific components of the mixture, into Gaussian distributions. Clustering of data can thus be achieved through estimating the parameters associated with the latent variable of the Gaussian mixture. To find maximum likelihood estimators in Gaussian mixtures with latent variables, a general method, namely expectation-maximization (EM) algorithm can be used. It has been revealed that there is a close similarity between k-means algorithm and the EM algorithm for Gaussian mixtures [1]. The k-means algorithm performs the clustering in a hard way that each data point is assigned into uniquely one cluster, whereas the EM algorithm makes a soft assignment based on the posterior probabilities.

A limitation of original GMM is that it only considers the cases where the data is drawn from Euclidean space. However, recent studies [5,6] have shown that naturally occurring data, such as texts and images, cannot possibly “fill up” the ambient Euclidean space, rather it probably concentrates around or closes to a lower dimensional submanifold whose structure plays a fundamental role in developing various kinds of algorithms including dimensionality reduction, clustering, supervised learning and semi-supervised learning algorithms [7–12].

By incorporating the informative manifold structure, Laplacian regularized GMM (LapGMM [13]) and locally consistent GMM (LCGMM [14]) have been proposed to improve the original Gaussian mixture model. Based on the intuition that the neighboring data points on the manifold are likely to be sampled by the same Gaussian component, the two models incorporate a regularized term using the graph Laplacian [15] into the maximum likelihood function of the original GMM. In this way, the probability distributions over Gaussian components of data can vary smoothly along the geodesics of data manifold, and therefore the two models have shown superior results for data clustering compared to original GMM.

However, these two models only enhance the similarity among the distributions over Gaussian components of data points within a neighborhood, whereas place no constraint on the points...
that are widely separated. In other words, the manifold learning of these two models only consider the locality information of data geometry, whereas the global structure is not well preserved.

In this paper, we present a novel Gaussian mixture model to focus on fully discovering the discriminating power of manifold learning, which is referred to as refined Gaussian mixture model (RGMM). In RGMM, we refine the manifold learning and clearly enhance the separability of the probability distributions associated with widely separated points in addition to increasing proximity of those of nearby points, in the hope of preserving the global manifold structure as well as maintaining local consistency. The similar idea of putting symmetric constraints on nearby and far apart points on the manifold has been successfully applied to dimension reduction [16,17] as well as other applications [18], and enhanced learning performance has been reported.

We formulate RGMM by incorporating a regularized term into the standard GMM and use expectation–maximization (EM) algorithm to solve the optimization problem. We provide experimental results on three real-world data including text corpora, face images and clinical data and demonstrate the effectiveness of RGMM.

The rest of this paper is organized as follows. We provide the background and overview the related work in Section 2. We then background and notations

Let us begin with reviewing the formulation of Gaussian mixture models, as well as providing a brief description of the related work.

2. Background and notations

2.1. Mixtures of Gaussians

Suppose we have a data set of observations \( X = \{x_1, \ldots, x_n\} \), and we wish to model this data using a mixture of Gaussians, where the mixture distribution can be expressed as a linear superposition of Gaussians [1]

\[
P(x_n | \theta) = \sum_{k=1}^{K} \pi_k N(x_n | \mu_k, \Sigma_k),
\]

where each Gaussian density \( N(x_n | \mu_k, \Sigma_k) \) is called a component of the mixture and has its own mean \( \mu_k \) and covariance \( \Sigma_k \). The parameters \( \pi_k \) serve as mixing coefficients, which can be viewed as the prior probabilities of picking each k-th component. Therefore \( \pi_k \) satisfy the requirements \( 0 \leq \pi_k \leq 1 \) and \( \sum_{k=1}^{K} \pi_k = 1 \) to be probabilities. The form of the Gaussian mixture distribution is governed by the set of parameters \( \theta = \{\pi_k, \mu_k, \Sigma_k\} \). Our goal is to maximize the log of the likelihood function with respect to the parameters

\[
\ln P(X | \theta) = \sum_{n=1}^{N} \ln \left( \sum_{k=1}^{K} \pi_k N(x_n | \mu_k, \Sigma_k) \right).
\]

We shall see the solution for this maximum likelihood problem is complicated, since the presence of the inside summation over \( k \) prevents the logarithm function from acting directly on the Gaussian. We cannot obtain a closed form solution by setting the derivatives of the log likelihood to be zero [1].

To solve the above problem, let us introduce a set of latent variables \( Z = \{z_1, \ldots, z_n\} \), with a corresponding binary random variable \( z_n \) associated with \( x_n \). The values of each element \( z_{nk} \) of \( Z_n \) satisfy \( z_{nk} \in \{0, 1\} \) and \( \sum_{k=1}^{K} z_{nk} = 1 \), and thus \( x_n \) having a 1-of-\( K \) representation to serve as the indicator variable of \( x_n \). The marginal distribution over each \( z_n \) is specified in terms of the mixing coefficients \( \pi_k \), such that \( P(z_{nk} = 1) = \pi_k \). Now instead of the marginal distribution \( P(x_n | \theta) \), we are able to work with the joint distribution

\[
P(x_n, z_n | \theta) = \prod_{k=1}^{K} \pi_k^{z_{nk}} N(x_n | \mu_k, \Sigma_k)^{z_{nk}}.
\]

The log likelihood function for the complete data set \( \{X, Z\} \) takes the form

\[
\ln P(X, Z | \theta) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} (\ln \pi_k + \ln N(x_n | \mu_k, \Sigma_k))
\]

and this will lead to significant simplifications for solving the maximum likelihood problem, most notably through the introduction of the expectation–maximization (EM) algorithm [1].

2.2. Previous work

Recent studies [5,6] have demonstrated that naturally occurring data, such as texts and images, are probably drawn from a probability distribution that has resided on or been close to lower dimensional submanifolds of the ambient space. To model this underlying manifold structure, recent work on GMM [13,14] proposed to incorporate the so-called Laplacian regularizer [8] in the maximum likelihood function of GMM.

The Laplacian regularizer is based on spectral graph theory [6,15] and emphasizes on preserving the locality of the intrinsic data geometry. To characterize the local geometric structure of the data space, we can define an edge weight matrix \( W \) of the corresponding nearest neighbor Graph in the form

\[
W_{mn} = \begin{cases} 
1 & \text{if } x_m \in N_p(x_n) \text{ or } x_n \in N_p(x_m), \\
0 & \text{otherwise.}
\end{cases}
\]

Here \( N_p(x_n) \) denotes the data sets of \( p \) nearest neighbors of \( x_n \), which is specified according to some distance metric such as Euclidean distance.

Intuitively, if two data points \( x_n \) and \( x_m \) are within a neighborhood in the original space, they should lie close to each other in the latent space or in terms of the resulting low-rank representations. This is usually referred to as the manifold assumption [6,8]. Based on this idea, LapGMM [13] and LCGMM [14] are proposed to incorporate the geometric knowledge of the conditional probability distributions for improving Gaussian mixture model. In particular, they both put the pairwise similarity constraint on the conditional probability distributions \( P(z_n | x_m) \), which we denote as \( \gamma_{nm} \) for simplicity. They define a regularization term to be minimized

\[
\sum_{m,n=1}^{N} w_{mn} D(\gamma_{nm} | \gamma_{nn}),
\]

where \( D(\cdot, \cdot) \) measures the similarity or proximity between two distributions according to some specific distance metric. LapGMM adopts the Euclidean distance for the similarity measure, while LCGMM uses symmetric Kullback–Leibler (KL) divergence. By requiring this regularization term to be minimized, the distributions \( P(z_n | x_m) \) will smooth along the geodesics in the intrinsic geometry of the data. The two models obtain the objective function by incorporating (6) into the likelihood of original GMM

\[
\mathcal{L}(\theta) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} (\ln \pi_k + \ln N(x_n | \mu_k, \Sigma_k)) - \frac{\lambda}{2} \sum_{m,n=1}^{N} w_{mn} D(\gamma_{mn} | \gamma_{nn}),
\]

where \( \lambda \) is the regularization parameter. By exploiting the local geometric structure of the data manifold, LapGMM and LCGMM can
provide more discriminating power than original GMM for text or image clustering.

However, we should see the regularization term (6) only discourages placing far apart the distributions of points over Gaussians that correspond to near neighbors, in order to preserve local information, but places no direct constraint on pairs associated with dissimilar data points which have relative small \(w_{mn}\). In this respect, the global manifold structure is not well preserved since widely separated data points can be “collapsed” as similar as distributions over Gaussian components. Consequently, the two models fail to fully discover the discriminative structure of the data manifold.

3. Symmetrically discriminative GMM

In this section, we go to full exposition of our proposed model, named refined Gaussian mixture model (RGMM). We then use the expectation–maximization (EM) algorithm to solve the regularized maximum likelihood problem of RGMM.

3.1. The objective function

The intuition is that we should emphasize on preserving the global manifold structure of data, beside the local consistency. We start by maximizing the separability of the probability distributions over Gaussians associated with dissimilar points, which is given by

\[
\sum_{m,n=1}^{N} (1-w_{mn}) D(\gamma_m, \gamma_n),
\]

where we use symmetric KL divergence \(D(\cdot, \cdot)\) to measure the distance between two conditional probability distributions, which takes the form

\[
D(\gamma_m, \gamma_n) = \sum_{k=1}^{K} \frac{\gamma_{mk} \ln \gamma_{mk}^{n} + \gamma_{nk} \ln \gamma_{nk}^{m}}{\gamma_{mk}},
\]

where \(\gamma_{nk}\) is the \(k\)th element of \(\gamma_n\), representing the conditional probability of \(x_n\) over \(k\)th Gaussian component, expressed by \(\gamma_{nk} = P(z_{nk} = 1 | x_n)\).

Simultaneously, we have to minimize the proximity of the probability distributions over Gaussians that correspond to similar pairs, expressed by (6). By using a linear combination of the two objectives (8) and (6), we obtain the regularization term to be maximized

\[
\sum_{m,n=1}^{N} (1-w_{mn}) D(\gamma_m, \gamma_n) - \alpha \sum_{m,n=1}^{N} w_{mn} D(\gamma_m, \gamma_n),
\]

where \(\alpha > 0\) is the balance parameter. The term (10) clearly encourages both keeping the distributions over Gaussians associated with points near a neighborhood relatively similar and keeping the ones associated with widely separated pairs relatively dissimilar. In this respect, the term (10) can be viewed a symmetric version of the regularization term (8) in LapGMM and LCGMM. In particular, LapGMM and LCGMM only impose penalties on producing widely different conditional probability distributions of Gaussian components that correspond to similar data points. The local information of the data structure is thus preserved. But producing similar conditional probability distributions that correspond to far apart data points are not penalized. This means some global information of the data structure is missing [19]. Consequently, non-similar points may collapse in small latent space regions [17,16]. These algorithms may also show significant boundary effects such as sensitive to outliers and fail to handle sparse data un-uniformly sampled. The regularization term (10) proposed to incorporate into GMM here has no such limitations. To preserve the global information of data structure, it imposes direct constraint on pairs associated with distant data points. By this means, the resulting conditional probability distributions \(P(z_n | x_n)\) of the GMM will faithfully recover the discriminative structure of data.

With some simple derivations, we can rewrite (10) as

\[
\sum_{m,n=1}^{N} (1-\tau w_{mn}) D(\gamma_m, \gamma_n),
\]

where \(\tau = 1 + \alpha > 1\). We then incorporate (11) into the likelihood of original GMM and obtain the maximum likelihood function

\[
L(\theta) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} [\ln \pi_k + \ln N(x_n | \mu_k, \Sigma_k)]
\]

\[
+ \lambda \sum_{m,n=1}^{N} (1-\tau w_{mn}) D(\gamma_m, \gamma_n),
\]

where \(\lambda > 0\) is the regularization parameter. We formally name this model as refined Gaussian mixture model (RGMM). It explores the full discriminating power by symmetrically learning the manifold structure of data for modeling Gaussian mixtures.

3.2. Model fitting with EM

A general and powerful method for finding maximum likelihood solutions for models with latent variables is the expectation–maximization (EM) algorithm [20]. In this case of GMM, the latent variables are \(z_{nk}\) that indicate which Gaussian component a data point belongs to. When using EM, we should first choose some initial values for the parameters (in Experiments, we use the results of k-means algorithm as initial values). Then we alternate between the following two updates that are the so-called the E-step and the M-step, until convergence is reached. In the E-step, we evaluate the posterior probabilities of the latent variables based on the current values of the parameters. We then use these probabilities in M-step to re-estimate the values by maximizing the expected complete data log likelihood.

In the following, we describe the two steps in our RGMM algorithm for the estimation of the parameters \(\theta = \{\pi_k, \mu_k, \Sigma_k\}\).

E-step: The E-step solution of RGMM is exactly the same as that of the original GMM [1]. We can evaluate the posterior probabilities of the latent variables directly using Bayes’ theorem

\[
\gamma_{nk} = \frac{\pi_k N(x_n | \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(x_n | \mu_j, \Sigma_j)}.
\]

M-step: In the M-step, we should maximize the expected value of the complete data log likelihood function (12), which is given by

\[
E[L(\theta)] = \mathcal{R}_1(\theta) + \mathcal{R}_2(\theta) = \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_{nk} [\ln \pi_k + \ln N(x_n | \mu_k, \Sigma_k)]
\]

\[
+ \lambda \sum_{m,n=1}^{N} (1-\tau w_{mn}) D(\gamma_m, \gamma_n).
\]

We shall see the parameters \(\pi_k\) is only involved in the first part \(\mathcal{R}_1(\theta)\), which is exactly the expected complete data log likelihood for the original GMM in (7). For the maximization with respect to \(\pi_k\), we have to introduce a Lagrange multiplier to enforce the normalization constraint \(\sum_k \pi_k = 1\). Following the same steps in [1], we obtain

\[
\pi_k = \frac{1}{N} \sum_{n=1}^{N} \gamma_{nk}.
\]
We now discuss about how to re-estimate \( \mu_k \) and \( \Sigma_k \). Using the posterior probabilities \( \gamma_{nk} \) in (13), we can rewrite (9) as
\[
D(\gamma_{nk}) = \frac{1}{2} \sum_{n=1}^{K} \left[ (\gamma_{nk} - \gamma_{nk})(\Lambda_{n,k} - \Lambda_{m,k}) \right],
\]
where \( \Lambda_{n,k} = (x_n - \mu_k)^T \Sigma_n^{-1} (x_n - \mu_k) \). With some simple derivations, the relevant part of \( E[\ell(\theta)] \) with respect to \( \mu_k \) and \( \Sigma_k \) is given by
\[
\tilde{E}[\ell(\theta)] = \tilde{E}_1(\theta) + \tilde{E}_2(\theta) - \frac{1}{2} \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_{nk} \left[ \frac{1}{\Lambda_{n,k}} - \frac{1}{\Lambda_{m,k}} \right],
\]
where \( \Lambda_{n,k} = (x_n - \mu_k)^T \Sigma_n^{-1} (x_n - \mu_k) \). Setting the derivative of \( \tilde{E}[\ell(\theta)] \) in (17) with respect to the means \( \mu_k \) to zero, we obtain
\[
0 = \sum_{n=1}^{N} \gamma_{nk} \Sigma_n^{-1} (x_n - \mu_k) + \lambda \sum_{m=1}^{N} (1 - \tau \mu_m) \left[ \frac{1}{\Lambda_{n,k}} - \frac{1}{\Lambda_{m,k}} \right] (x_n - \mu_k).
\]
Multiplying by \( \Sigma_k \) and rearranging, we obtain the re-estimation equation
\[
\mu_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk} x_n + \frac{\lambda}{N_k} \sum_{m=1}^{N} (1 - \tau \mu_m) \left[ \frac{1}{\Lambda_{n,k}} - \frac{1}{\Lambda_{m,k}} \right] (x_n - \mu_k),
\]
where we have defined the effective number of points \( N_k \) assigned to cluster \( k \) as
\[
N_k = \sum_{n=1}^{N} \gamma_{nk}.
\]
If we set the derivative of \( \tilde{E}[\ell(\theta)] \) in (17) with respect to the \( \Sigma_k \) to zero, we obtain
\[
0 = \frac{1}{2} \sum_{n=1}^{N} \gamma_{nk} \Sigma_n^{-1} - \Omega_{n,k} + \frac{\lambda}{N_k} \sum_{m=1}^{N} (1 - \tau \mu_m) \left[ \frac{1}{\Lambda_{n,k}} - \frac{1}{\Lambda_{m,k}} \right] \Sigma_n^{-1} - \Omega_{m,k},
\]
where \( \Omega_{n,k} = (x_n - \mu_k)^T \Sigma_n^{-1} (x_n - \mu_k) \). After solving this equation, the re-estimation equation for \( \Sigma_k \) is given by
\[
\Sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk} \Omega_{n,k} + \frac{\lambda}{N_k} \sum_{m=1}^{N} (1 - \tau \mu_m) \left[ \frac{1}{\Lambda_{n,k}} - \frac{1}{\Lambda_{m,k}} \right] \Omega_{m,k}.
\]

The E-step and the M-step are repeatedly alternated until convergence is reached. We summarized our model fitting algorithm in Algorithm 1.

**Algorithm 1.** EM for RGMM.

**Input:**
- the data set \( X = \{x_n\} \) for \( n = 1, \ldots, N \)
- the parameters \( K, p, \lambda, \tau \), and the convergence threshold \( \epsilon \)

**Output:**
- \( \theta = \{\pi_k, \mu_k, \Sigma_k\} \) for \( k = 1, \ldots, K \)

**Procedure:**
1. construct \( p \) nearest neighbor graph with the weight matrix \( W \) as in (5)
2. initialize \( \theta^0 = \{\pi_k^0, \mu_k^0, \Sigma_k^0\} \) for all \( k \)
3. \( t = 0 \)
4. repeat
5. \( t = t + 1 \)
6. **E-step:**
7. compute the posterior probabilities \( \gamma_{nk} \) for all \( n \) and \( k \) as in (13):
\[
\gamma_{nk} = \frac{p^{-1} \pi_k \phi_{nk}}{\sum_{m=1}^{K} p^{-1} \pi_m \phi_{nk}},
\]
8. **M-step:**
9. compute \( \theta' = \{\pi_k', \mu_k', \Sigma_k'\} \) for all \( k \) as in (15), (19) and (22)
10. until \( L(\theta') - L(\theta^{t-1}) \leq \epsilon \)
11. return \( \theta' \)

**4. Experiments**

In this section, we conduct extensive experiments to evaluate our proposed RGMM on synthetic data and three real-world data sets including texts, face images and clinical data. We compare RGMM performance against the following algorithms:
- Locally consistent Gaussian mixture model (LCGMM [14]).
- Laplacian regularized Gaussian mixture model (LapGMM [13]).
- The original Gaussian mixture model (GMM [1]).
- Principal component analysis (PCA).

**4.1. Toy examples**

Let us look at two toy examples to illustrate the efficiency of the proposed algorithm. Fig. 1 gives the learning results of 1000 random samples drawn from Swiss roll. To unfold the 3D shape into 2D and recover the data structure, all the above-mentioned Gaussian mixture models (include GMM, LapGMM, LCGMM and RGMM) are trained using all the data points. We then evaluate the posterior probability for each component in the mixture distribution from which the data was generated. In particular, we use PCA to produce a 2D embedding from all the resulting values of the responsibilities \( \gamma_{nk} \) associated with data points \( x_n \). The parameters in RGMM is empirically tuned as \( p = 5, \lambda = 100 \) and \( \tau = 0.1 \). As we can see, PCA only or GMM cannot recover the faithful structure of the Swiss roll since they use no manifold information. LapGMM and LCGMM consider the manifold structure, but cannot unfold the Swiss roll faithfully either. They somewhat distort the data by compressing the roll into a curve. RGMM performs the best by preserving the intrinsic manifold structure.

Another illustrative example is a set of 1000 3D points randomly drawn from six random centers of clusters, where clusters are connected by lines. We can see the resulting 2D representation of each algorithm in Fig. 2. PCA and GMM perform well in this simple data where no much manifold information. However, LapGMM and LCGMM have trouble with the sparse connected lines and distort their shape, though the clusters are compressed into lines. We may infer they are not capable to preserving the relative relations of distant points and nearby points. RGMM has no such limitation and preserve the structure of both clusters and lines.

**4.2. Real-world data sets**

We perform the task of data clustering in our experiments to evaluate our proposed RGMM algorithm. The statistics of the three real-world data sets we investigate are summarized in Table 1.
The first one is the Reuters-21578 text data set, provided by Reuters and corrected by Lewis. This collection is divided unevenly across 65 different categories which have varying sizes from 1 to 3713. We use the subset of the 10 largest categories for our experiments, which includes 7285 documents and 18,221 distinct words. To randomize the experiment, we select $K$ (ranging from 2 to 10) categories among the subset, and randomly sample 100 documents for each selected category. For the 10th category which only have 90 documents, we use all these documents if this category is sampled. Ten test runs are performed for each given number of $K$.

The second one is the UMist data set, which contains grayscale face images (views) of 20 different people; there are 575 images in total, each with 112 x 92 feature size, manually cropped by Daniel Graham at UMist. We cluster the images into 20 categories according to different people. We select $K$ (ranging from 2 to 20) categories and perform 20 test runs to record the average performance for each $K$.

The third one is the Dermatology data set from the UCI’s data set repository. The aim is to diagnose the type of eryhemato-squamous disease. This data set contains the clinical features of 366 patients, which were grouped into 6 different classes of eryhemato-squamous disease. The patients were evaluated clinically with 34 features in total. For each selected $K$ (ranging from 2 to 6) classes, we repeat 10 test runs of the clustering task to record the average performance.

### 4.3. Clustering evaluation

We evaluate the clustering performance of each algorithm. For PCA, we use the standard $k$-means algorithm for the resulting data to perform clustering. The standard $k$-means algorithm itself serves as a baseline. Evaluation is conducted by comparing the cluster label of each sample generated using clustering algorithms with its ground truth label provided by the data set. The standard clustering metric accuracy (AC) is used to measure the clustering performance [21].

Given a data point $x_i$, let $r_i$ and $s_i$ be the cluster label and the label provided by the data set, respectively. The AC is defined as follows:

$$AC = \frac{1}{n} \sum_{i=1}^{n} \delta(s_i, \text{map}(r_i)),$$

where $n$ is the total number of samples and $\delta(x,y)$ is the delta function that equals one if $x=y$ and equals zero otherwise, and $\text{map}(r_i)$ is the permutation mapping function that maps each cluster label $r_i$ to the equivalent label from the data set. The best mapping can be found by using the Kuhn–Munkres algorithm [22].
Table 2
Clustering accuracy (mean ± std-dev %) on Reuters.

<table>
<thead>
<tr>
<th>K</th>
<th>RGMM</th>
<th>LCGMM</th>
<th>LapGMM</th>
<th>GMM</th>
<th>PCA+k-means</th>
<th>k-means</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>98.2 ± 1.5</td>
<td>98.3 ± 1.4</td>
<td>98.4 ± 1.4</td>
<td>96.9 ± 1.6</td>
<td>96.3 ± 1.4</td>
<td>96.7 ± 2.8</td>
</tr>
<tr>
<td>3</td>
<td>92.7 ± 8.2</td>
<td>89.6 ± 8.4</td>
<td>87.8 ± 8.6</td>
<td>86.2 ± 8.3</td>
<td>86.3 ± 8.6</td>
<td>87.4 ± 8.2</td>
</tr>
<tr>
<td>4</td>
<td>90.4 ± 5.8</td>
<td>88.7 ± 6.9</td>
<td>87.4 ± 7.7</td>
<td>85.0 ± 6.6</td>
<td>85.4 ± 6.9</td>
<td>86.6 ± 5.7</td>
</tr>
<tr>
<td>5</td>
<td>89.7 ± 3.9</td>
<td>87.5 ± 3.8</td>
<td>87.9 ± 4.0</td>
<td>83.9 ± 4.0</td>
<td>84.9 ± 3.9</td>
<td>82.4 ± 4.4</td>
</tr>
<tr>
<td>6</td>
<td>88.4 ± 5.9</td>
<td>85.6 ± 6.6</td>
<td>86.4 ± 5.7</td>
<td>83.0 ± 6.7</td>
<td>82.8 ± 6.8</td>
<td>83.7 ± 6.9</td>
</tr>
<tr>
<td>7</td>
<td>87.3 ± 3.2</td>
<td>83.4 ± 5.2</td>
<td>84.4 ± 4.9</td>
<td>80.4 ± 4.9</td>
<td>81.1 ± 5.2</td>
<td>82.0 ± 5.3</td>
</tr>
<tr>
<td>8</td>
<td>86.8 ± 4.3</td>
<td>81.7 ± 5.4</td>
<td>83.9 ± 3.0</td>
<td>80.3 ± 4.9</td>
<td>81.3 ± 5.1</td>
<td>81.6 ± 5.3</td>
</tr>
<tr>
<td>9</td>
<td>85.5 ± 2.5</td>
<td>82.4 ± 2.2</td>
<td>81.9 ± 1.9</td>
<td>79.1 ± 2.0</td>
<td>79.4 ± 2.3</td>
<td>79.6 ± 3.1</td>
</tr>
<tr>
<td>10</td>
<td>84.7 ± 2.7</td>
<td>82.3 ± 2.8</td>
<td>81.2 ± 2.2</td>
<td>77.2 ± 1.9</td>
<td>78.4 ± 2.5</td>
<td>77.5 ± 2.7</td>
</tr>
<tr>
<td>Avg</td>
<td>89.3</td>
<td>86.8</td>
<td>86.5</td>
<td>83.5</td>
<td>83.9</td>
<td>84.1</td>
</tr>
</tbody>
</table>

Table 3
Clustering accuracy (mean ± std-dev %) on UMass.

<table>
<thead>
<tr>
<th>K</th>
<th>RGMM</th>
<th>LCGMM</th>
<th>LapGMM</th>
<th>GMM</th>
<th>PCA+k-means</th>
<th>k-means</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>89.3 ± 12.5</td>
<td>82.9 ± 12.8</td>
<td>78.2 ± 12.4</td>
<td>73.8 ± 13.6</td>
<td>72.1 ± 13.1</td>
<td>72.3 ± 12.3</td>
</tr>
<tr>
<td>4</td>
<td>71.2 ± 11.2</td>
<td>65.5 ± 9.8</td>
<td>66.4 ± 10.1</td>
<td>63.8 ± 11.9</td>
<td>62.5 ± 11.1</td>
<td>60.9 ± 12.9</td>
</tr>
<tr>
<td>6</td>
<td>66.1 ± 8.2</td>
<td>61.7 ± 6.5</td>
<td>62.2 ± 7.1</td>
<td>61.1 ± 6.4</td>
<td>57.8 ± 8.0</td>
<td>56.6 ± 8.3</td>
</tr>
<tr>
<td>8</td>
<td>61.6 ± 4.9</td>
<td>56.8 ± 5.7</td>
<td>55.8 ± 6.8</td>
<td>52.9 ± 6.0</td>
<td>52.5 ± 6.1</td>
<td>53.2 ± 7.7</td>
</tr>
<tr>
<td>10</td>
<td>54.9 ± 2.4</td>
<td>51.3 ± 2.9</td>
<td>50.3 ± 3.0</td>
<td>47.2 ± 2.3</td>
<td>46.8 ± 2.1</td>
<td>47.1 ± 2.8</td>
</tr>
<tr>
<td>12</td>
<td>53.7 ± 1.9</td>
<td>52.3 ± 3.4</td>
<td>51.4 ± 4.1</td>
<td>50.1 ± 3.1</td>
<td>47.5 ± 3.8</td>
<td>47.7 ± 3.0</td>
</tr>
<tr>
<td>14</td>
<td>52.2 ± 1.9</td>
<td>50.1 ± 3.9</td>
<td>48.2 ± 4.2</td>
<td>45.8 ± 3.3</td>
<td>45.3 ± 4.0</td>
<td>44.7 ± 3.4</td>
</tr>
<tr>
<td>16</td>
<td>50.4 ± 2.8</td>
<td>47.9 ± 2.5</td>
<td>47.1 ± 2.6</td>
<td>44.6 ± 1.6</td>
<td>44.8 ± 2.8</td>
<td>43.4 ± 2.1</td>
</tr>
<tr>
<td>18</td>
<td>49.9 ± 1.9</td>
<td>46.8 ± 1.3</td>
<td>46.1 ± 1.8</td>
<td>42.6 ± 2.2</td>
<td>41.7 ± 2.2</td>
<td>41.6 ± 2.1</td>
</tr>
<tr>
<td>20</td>
<td>48.7</td>
<td>46.4</td>
<td>45.2</td>
<td>40.3</td>
<td>39.6</td>
<td>40.4</td>
</tr>
<tr>
<td>Avg</td>
<td>59.8</td>
<td>56.1</td>
<td>55.0</td>
<td>52.2</td>
<td>51.0</td>
<td>50.7</td>
</tr>
</tbody>
</table>

Table 4
Clustering accuracy (mean ± std-dev %) on Dermatology.

<table>
<thead>
<tr>
<th>K</th>
<th>RGMM</th>
<th>LCGMM</th>
<th>LapGMM</th>
<th>GMM</th>
<th>PCA+k-means</th>
<th>k-means</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>78.8 ± 21.1</td>
<td>75.9 ± 23.1</td>
<td>73.7 ± 22.9</td>
<td>73.9 ± 22.0</td>
<td>74.7 ± 21.7</td>
<td>74.6 ± 21.4</td>
</tr>
<tr>
<td>3</td>
<td>84.5 ± 16.7</td>
<td>80.6 ± 15.8</td>
<td>79.9 ± 16.2</td>
<td>78.7 ± 19.1</td>
<td>76.4 ± 21.4</td>
<td>74.5 ± 22.1</td>
</tr>
<tr>
<td>4</td>
<td>81.2 ± 9.2</td>
<td>78.4 ± 8.0</td>
<td>78.2 ± 8.21</td>
<td>75.3 ± 13.0</td>
<td>66.5 ± 10.4</td>
<td>65.4 ± 10.8</td>
</tr>
<tr>
<td>5</td>
<td>78.3 ± 5.4</td>
<td>76.7 ± 3.9</td>
<td>72.2 ± 6.1</td>
<td>68.2 ± 8.6</td>
<td>64.8 ± 4.9</td>
<td>61.5 ± 7.8</td>
</tr>
<tr>
<td>6</td>
<td>72.8</td>
<td>71.2</td>
<td>63.6</td>
<td>58.7</td>
<td>60.8</td>
<td>56.6</td>
</tr>
<tr>
<td>Avg</td>
<td>79.1</td>
<td>76.5</td>
<td>73.5</td>
<td>70.9</td>
<td>68.6</td>
<td>66.5</td>
</tr>
</tbody>
</table>

4.4. Parameter settings

We now discuss about the parameter selection. For LCGMM, LapGMM and our RGMM, the number of nearest neighbors p for each point is ideally searched from the grid: (3, 5, 10, 15, 20, 30), and the regularization parameters λ are tuned, respectively, among: (10−2, 10−1, 1, 10, 10², 10³, 10⁴). The balance parameter α for our RGMM is searched in: 10−1, 1, 10, 10², 10³), and τ = 1 + α is fixed. Best performance for each algorithm is reported.

4.5. Clustering results

Tables 2–4 illustrate the experimental results of all the clustering algorithms on the three data sets, respectively. We can see that RGMM consistently outperform all the other algorithms including recently proposed LapGMM and LCGMM in almost all the cases, in terms of clustering accuracy. In particular, RGMM yields 2.5% 3.7% and 2.6% average performance gains over the second best algorithm on Reuters, UMass and Dermatology, respectively. These results demonstrate RGMM is capable in exploiting the manifold structure of the data and effective in discriminating different clusters of data points.

Both LapGMM and LCGMM are also manifold learning based models, and they show superior performance compared to original GMM, PCA+k-means and k-means, as expected. But they do not show as much discriminating power as RGMM since they ignore the global manifold and have the potential problem that widely separated data points can be “collapsed” as similar distributions over Gaussian components. Moreover, GMM does not perform well and provide limited discriminating power since it consider no geometric structure of the data which is informative.

5. Conclusions and future work

In this paper, we have motivated a novel Gaussian mixture model based on refined manifold learning. The proposed model clearly encourages both keeping the distributions over Gaussians associated with widely separated pairs relatively dissimilar in addition to keeping ones associated with points near a neighborhood relatively similar. In this respect, it is a refined manifold learning based GMM with local and global consistency. We name it as refined Gaussian mixture model (RGMM). We solve the maximum likelihood problem of RGMM using expectation–maximization (EM) algorithm. We present experiments on several real-world data sets and demonstrate that our approach outperform previous models such as LapGMM and LCGMM in the task of data clustering.
For future work, we plan to present the detailed theoretical connections between RGMM and related manifold learning methods including Laplacian eigenmaps [6], locality preserving projections [7], and the elastic embedding algorithm [17], extending to Kernel GMM is also another interesting direction.

Acknowledgment

This work is supported by the “11th Five-Year Plan” National Scientific and Technological Support Key Project—Regional Health Data Center and Resource Sharing System (No. 2008BAH27804).

References


Jianfeng Shen received the MD degree in Medicine from Zhejiang University, China, in 2003. He is working in Information Center of Health Bureau of Zhejiang Province. His research interests include neurosurgery and health information.

Bin Ju received the BS and PhD degrees in Computer Science from Zhejiang University, China, in 1995 and 2000, respectively. He is a professor in College of Computer Science, Zhejiang University. His research interests include machine learning, data mining, information retrieval, and mobile database.

Jiajun Bu received the BS and PhD degrees in Computer Science from Zhejiang University, China, in 2005. He is currently a candidate for a PhD degree in Computer Science at Zhejiang University. His research interests include machine learning, data mining, and health information.

Hao Wu received the BS and MS degrees in Computer Science from Zhejiang University, China, in 2011. His research interests include health information, java developing, and data mining.

Tao Jiang received the MD degree in Medicine from Zhejiang University, China, in 2005. His research interests include health information, data mining, and information retrieval.

Jianfeng Shen received the MD degree in Medicine from Zhejiang University, China, in 2003. He is working in Information Center of Health Bureau of Zhejiang Province. His research interests include neurosurgery and health information.

Bin Ju received the BS and PhD degrees in Computer Science from Zhejiang University, China, in 1995 and 2000, respectively. He is a professor in College of Computer Science, Zhejiang University. His research interests include machine learning, data mining, information retrieval, and mobile database.

Jiajun Bu received the BS and PhD degrees in Computer Science from Zhejiang University, China, in 2005. He is currently a candidate for a PhD degree in Computer Science at Zhejiang University. His research interests include machine learning, data mining, and health information.

Hao Wu received the BS and MS degrees in Computer Science from Zhejiang University, China, in 2011. His research interests include health information, java developing, and data mining.

Tao Jiang received the MD degree in Medicine from Zhejiang University, China, in 2005. His research interests include health information, data mining, and information retrieval.

Jianfeng Shen received the MD degree in Medicine from Zhejiang University, China, in 2003. He is working in Information Center of Health Bureau of Zhejiang Province. His research interests include neurosurgery and health information.

Bin Ju received the BS and PhD degrees in Computer Science from Zhejiang University, China, in 1995 and 2000, respectively. He is a professor in College of Computer Science, Zhejiang University. His research interests include machine learning, data mining, information retrieval, and mobile database.