Gaussian Mixture Model with Local Consistency

Jialu Liu **Deng Cai*** **Xiaofei He**

State Key Lab of CAD&CG, College of Computer Science,
Zhejiang University, China
remenberl@gmail.com, {dengcai,xiaofeihe}@cad.zju.edu.cn
*Corresponding author

Abstract

Gaussian Mixture Model (GMM) is one of the most popular data clustering methods which can be viewed as a linear combination of different Gaussian components. In GMM, each cluster obeys Gaussian distribution and the task of clustering is to group observations into different components through estimating each cluster's own parameters. The Expectation-Maximization algorithm is always involved in such estimation problem. However, many previous studies have shown naturally occurring data may reside on or close to an underlying submanifold. In this paper, we consider the case where the probability distribution is supported on a submanifold of the ambient space. We take into account the smoothness of the conditional probability distribution along the geodesics of data manifold. That is, if two observations are "close" in intrinsic geometry, their distributions over different Gaussian components are similar. Simply speaking, we introduce a novel method based on manifold structure for data clustering, called *Locally Consistent Gaussian Mixture Model* (LCGMM). Specifically, we construct a nearest neighbor graph and adopt Kullback-Leibler Divergence as the "distance" measurement to regularize the objective function of GMM. Experiments on several data sets demonstrate the effectiveness of such regularization.

Introduction

Clustering is an unsupervised classification of patterns (observations, data items, or feature vectors) into groups(clusters) (Jain, Murty, and Flynn 1999). The goal of it is to organize objects into groups such that members within each group are similar in some way. Therefore, a cluster is a collection of objects which are "close" between them and are "dissimilar" to others belonging to different clusters. Data clustering is one of the common techniques in exploratory data analysis. It has been addressed in many contexts and has drawn enormous attention in many fields, including data mining, machine learning, pattern recognition and information retrieval.

The clustering algorithms can be roughly divided into two categories: similarity-based and model-based. Similarity-based clustering algorithms are designed on the basis of similarity function between data observations without any probability assumption. K-means (Duda, Hart, and Stork 2000) and spectral clustering (Ng, Jordan, and Weiss 2001; Shi and Malik 1997) are two representative examples. The former is designed to minimize the sum of distances between the assumed cluster centers and data samples, while the latter usually clusters the data points using the top eigenvectors of *graph Laplacian* (Chung 1997), which is defined on the affinity matrix of data points. From the graph partitioning perspective, spectral clustering tries to find the best cut of the graph, aiming at optimizing the predefined criterion function. Normalized cut (Shi and Malik 1997) is one of the most well applied criterion functions.

Unlike similarity-based methods, model-based clustering can generate soft partition which is sometimes more flexible. Model-based methods use mixture distributions to fit the data and the conditional probabilities are naturally used to assign probabilistic labels. One of the most widely used mixture models for clustering is Gaussian Mixture Model (Bishop 2006). Each Gaussian density is called a component of the mixture and has its own mean and covariance. In many applications, their parameters are determined by maximum likelihood, typically using the Expectation-Maximization algorithm (Dempster, Laird, and Rubin 1977).

GMM assumes that the probability distribution generating the data is supported on the Euclidean space. However, many previous studies (Tenenbaum, de Silva, and Langford 2000; Roweis and Saul 2000; Belkin and Niyogi 2001) have shown naturally occurring data may reside on or close to an underlying submanifold. It has also been shown that learning performance can be significantly enhanced if the manifold (geometrical) structure is exploited (Ng, Jordan, and Weiss 2001; Belkin, Niyogi, and Sindhwani 2006; Cai, Wang, and He 2009; Cai, He, and Han 2010).

In this paper, we propose a novel model-based algorithm for data clustering, called *Locally Consistent Gaussian Mixture Model* (LCGMM), which explicitly considers the manifold structure. Following the intuition that naturally occurring data may reside on or close to a *submanifold* of the ambient space, we incorporate a regularizer into the objective function of Gaussian Mixture Model after constructing a nearest neighbor graph and adopting Kullback-Leibler Divergence as the "distance" measurement. It is important to note that the work presented here is fundamentally based on

our previous work LapGMM (He et al. 2010). The major difference is that LapGMM constructs the regularizer using Euclidean distance and uses generalized EM to estimate the parameters. In this work, we use KL-Divergence to measure the "distance" of two probability distributions which is much more natural. Moreover, by using KL-Divergence, the new objective function can be solved more effectively by ordinary EM algorithm.

Background

Gaussian mixture model can be viewed as a linear superposition of different Gaussian components in which each is a basis function or a "hidden" unit, aiming at offering a comparatively richer model than the single Gaussian (Bishop 2006):

$$P(x|\Theta) = \sum_{k=1}^{K} \pi_k p_k(x|\theta_k)$$

where each component prior (π_k) can be viewed as positive weights in an output layer and satisfying $\sum_{k=1}^{K} \pi_k = 1$. And all parameters here are represented by Θ where $\Theta = (\pi_1, \ldots, \pi_K, \theta_1, \ldots, \theta_K)$. Note that each θ_k describes a Gaussian density function p_k, meaning that $p_k(x|\theta_k) \sim \mathcal{N}(x|\mu_k, \Sigma_k)$.

The optimal parameter Θ is determined by Maximum Likelihood (ML) principle. Given observations $\mathcal{X} = (x_1, x_2, \ldots, x_N)$, ML tries to find Θ such that $P(\mathcal{X}|\Theta)$ is a maximum. For the sake of efficient optimization, it is typical to introduce the log likelihood function defined as follows:

$$\begin{aligned}\mathcal{L}(\Theta) =& \log P(\mathcal{X}|\Theta) = \log \prod_{i=1}^{N} P(x_i|\Theta) \\ =& \sum_{i=1}^{N} \log \left(\sum_{k=1}^{K} \pi_k p_k(x_i|\theta_k) \right)\end{aligned}$$

Since the above log likelihood function contains the log of the sum, it is difficult to find the optimal solution. By introducing the latent variable $P(c|x)$ which represents the possibility of observation x belonging to the component c, the *complete* log likelihood function is (Bishop 2006):

$$\sum_{i=1}^{N} \sum_{k=1}^{K} P(c_k|x_i) \Big(\log \pi_k + \log \mathcal{N}(x_i|\mu_k, \Sigma_k) \Big) \quad (1)$$

With this complete log likelihood, we are able to obtain estimates for Θ under the assumption that $P(c|x)$ is fixed. This procedure is known as Expectation-Maximization algorithm (Dempster, Laird, and Rubin 1977) , which is a powerful method for finding maximum likelihood solutions for models with latent variables. It is a process of iteration which alternates between an expectation (E) step computing an expectation of the latent variable ($P(c|x)$ in the GMM case), and a maximization (M) step computing the parameters (Θ) which maximize the complete log likelihood. Parameters computed either in E or M step are alternatively fixed during the other step as known quantities.

In fact, there is a close similarity between K-means and EM algorithm for Gaussian mixtures (Bishop 2006). The K-means algorithm does the clustering in a *hard* way, in which each sample is associated directly with only one cluster, while the EM algorithm makes a comparatively *soft* assignment relied on the posterior probabilities. It is noticeable that we can derive the K-means algorithm as a non-probabilistic limit of EM for GMM. For more information, please see (Bishop 2006).

Gaussian Mixture Model with Local Consistency

Naturally occurring data may be generated with possibly much fewer degrees of freedom than what the ambient dimension would suggest (Tenenbaum, de Silva, and Langford 2000; Roweis and Saul 2000). Thus, the general GMM might not obtain an ideal result since it doesn't consider the case when the data is supported on a submanifold of the ambient space. In this section, we introduce a novel method to show how geometric knowledge of the probability distribution is incorporated into learning a Gaussian mixture model.

GMM with Locally Consistent Regularizer

Recall the standard framework of learning from examples. There is a probability distribution P on $X \times \mathbb{R}$ according to which examples are generated for function learning. Unlabeled examples are simply $x \in X$ drawn according to the marginal distribution P_X of P. Previous studies have shown that there may be connection between the marginal and conditional distributions (Belkin, Niyogi, and Sindhwani 2006). Therefore, we make a specific assumption about the connection between the distribution of observations P_X and the conditional distribution $P(c|x_i)$, where c represents the clusters. That is, within some neighboring samples, their $P(c|x_i)$ are "similar" to a certain degree. In another way, the conditional probability distribution $P(c|x)$ varies smoothly along the geodesics in the intrinsic geometry of P_X. This is usually referred to as *local consistency assumption* (Zhou et al. 2003; Cai, Wang, and He 2009), which plays an essential role in developing various kinds of algorithms including dimensionality reduction (Belkin and Niyogi 2001) and semi-supervised learning algorithms (Belkin, Niyogi, and Sindhwani 2006; Zhu and Lafferty 2005).

To measure the "similarity" (or "distance") between two distributions, it is common to use Kullback–Leibler Divergence (KL-Divergence). Given two distributions $P_i(c)$ and $P_j(c)$, the KL-Divergence between them is defined as below:

$$D\big(P_i(c) \| P_j(c)\big) = \sum_{c} P_i(c) \log \frac{P_i(c)}{P_j(c)} \quad (2)$$

The above equation is not symmetric, we can use

$$\mathcal{D}_{ij} = \frac{1}{2} \Big(D\big(P_i(c) \| P_j(c)\big) + D\big(P_j(c) \| P_i(c)\big) \Big) \quad (3)$$

to measure the distance between distributions $P_i(c)$ and $P_j(c)$.

Recent studies on spectral graph theory (Chung 1997) and manifold learning theory (Belkin and Niyogi 2001) have demonstrated that the local geometric structure can be effectively modeled through a nearest neighbor graph on a scatter

of data points. Consider a graph with N vertices where each vertex corresponds to a data point. Define the edge weight matrix W as follows:

$$W_{ij} = \begin{cases} 1 & \text{if } x_i \in N_p(x_j) \text{ or } x_j \in N_p(x_i). \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

where $N_p(x_i)$ denotes the data sets of p nearest neighbors of x_i.

Let $P_i(c) = P(c|x_i)$, with the weight matrix of the nearest neighbor graph in Eq. (4), the following term can be used to measure the smoothness of $P(c|x)$ on the graph:

$$\begin{aligned} \mathcal{R} &= \sum_{i,j=1}^{N} \mathcal{D}_{ij} W_{ij} \\ &= \frac{1}{2} \sum_{i,j=1}^{N} \Big(D\big(P_i(c) \| P_j(c)\big) + D\big(P_j(c) \| P_i(c)\big) \Big) W_{ij} \end{aligned} \quad (5)$$

The smaller of $\mathcal{R}$, the smoother of $P(c|x)$ over the graph (consequently along the geodesics in the intrinsic geometry of the data).

Incorporating the above smoothness term into the likelihood of original GMM, we have

$$\begin{aligned} \mathfrak{L} =& \mathcal{L} - \lambda \mathcal{R} \\ \propto & \sum_{i=1}^{N} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(x_i | \mu_k, \Sigma_k) \\ & - \frac{\lambda}{2} \sum_{i,j=1}^{N} \Big(D\big(P_i(c) \| P_j(c)\big) + D\big(P_j(c) \| P_i(c)\big) \Big) W_{ij} \end{aligned} \quad (6)$$

where $P_i(c)$ is the abbreviation of $P(c|x_i)$ and λ is the regularization parameter. Since this approach incorporates local consistency through a regularizer, we call it Locally Consistent Gaussian Mixture Model (LCGMM). The idea of incorporating locally consistent regularization in GMM model has also been studied in our previous work LapGMM (He et al. 2010). The major difference is that LapGMM constructs the regularizer using Euclidean distance. While in this work, we use the divergence measure which leads to a new objective function as well as a nice EM algorithm.

In the next subsection, we show how to apply EM algorithm to maximize this regularized log-likelihood function.

Model Fitting with EM

To find maximum likelihood estimation when there exist latent variables, we need to use the EM algorithm. In our case, the latent variables are the Gaussian components to which the data points belong. Firstly, we need to estimate values to perform the E-step, computing expectations for the latent variables. Then we use these variables to obtain the parameters which maximize the log likelihood (M-step). These two steps are repeated until a certain stopping criterion is reached.

The parameters of LCGMM is the same as that of GMM. For simplicity, we use Θ to denote all the parameters, $\Theta = (\pi_1, \ldots, \pi_K, (\mu_1, \Sigma_1), \ldots, (\mu_k, \Sigma_k))$.

E-step:

The E-step for LCGMM is exactly the same as that in original GMM. The posterior probabilities for the latent variables are $P(c_k|x_i)$, which can be computed by simply applying Bayes' formula (Bishop 2006):

$$P(c_k|x_i) = P(c_k = 1|x_i) = \frac{\pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_i|\mu_j, \Sigma_j)} \quad (7)$$

M-step:

With simple derivations (Bishop 2006), one can obtain the expected *complete* data log-likelihood for LCGMM:

$$\begin{aligned} & Q(\Theta) = Q_1(\Theta) - Q_2(\Theta) \\ =& \sum_{i=1}^{N} \sum_{k=1}^{K} P(c_k|x_i) \Big(\log \pi_k + \log \mathcal{N}(x_i|\mu_k, \Sigma_k) \Big) \\ & - \frac{\lambda}{2} \sum_{i,j=1}^{N} \Big(D\big(P_i(c) \| P_j(c)\big) + D\big(P_j(c) \| P_i(c)\big) \Big) W_{ij} \end{aligned} \quad (8)$$

Notice that $Q(\Theta)$ has two parts. The first part $Q_1(\Theta)$ is exactly the expected complete data log-likelihood for GMM in Eq. (1). The second part $Q_2(\Theta)$ is the locally consistent regularizer which only involves the parameters $\{\mu_k, \Sigma_k\}_{k=1}^K$. Thus, the M-step re-estimation equation for π_k will be exactly the same as that in GMM. It is (Bishop 2006):

$$\pi_k = \frac{\sum_{i=1}^{N} P(c_k|x_i)}{N} \quad (9)$$

Now let us derive the re-estimation equation for $\{\mu_k, \Sigma_k\}_{k=1}^K$.

With the posterior probabilities for the latent variables in Eq. (7) estimated in E-step, we have:

$$\begin{aligned} & D\big(P_i(c) \| P_j(c)\big) = \sum_{k=1}^{K} P_i(c_k) \log \frac{P_i(c_k)}{P_j(c_k)} \\ =& \sum_{k=1}^{K} P(c_k|x_i) \log \frac{\mathcal{N}(x_i|\mu_k, \Sigma_k)}{\mathcal{N}(x_j|\mu_k, \Sigma_k)} + O(x_i||x_j) \\ =& \sum_{k=1}^{K} P(c_k|x_i) \Big(\frac{1}{2} (x_j - \mu_k)^T \Sigma_k^{-1} (x_j - \mu_k) \\ & - \frac{1}{2} (x_i - \mu_k)^T \Sigma_k^{-1} (x_i - \mu_k) \Big) + O(x_i||x_j) \end{aligned} \quad (10)$$

where

$$O(x_i||x_j) = \log \frac{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_j|\mu_k, \Sigma_k)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k)}$$

Note that $O(x_i||x_j) + O(x_j||x_i) = 0$, which means only the former term of Eq. (10) will be involved in the following computation.

The relevant part (only relevant to $\{\mu_k, \Sigma_k\}_{k=1}^K$) of $Q(\Theta)$ is:

$$\tilde{Q}(\Theta) = \tilde{Q}_1(\Theta) - Q_2(\Theta) \quad (11)$$

where

$$\begin{aligned} \tilde{Q}_1(\Theta) =& \sum_{i=1}^{N} \sum_{k=1}^{K} P(c_k|x_i) \Big(\frac{1}{2} \log(|\Sigma_k^{-1}|) \\ & - \frac{1}{2} (x_i - \mu_k)^T \Sigma_k^{-1} (x_i - \mu_k) \Big) \end{aligned} \quad (12)$$

By taking the derivative of Eq. (11) with respect to μ_k and setting it to zero, we get:

$$\sum_{i=1}^{N} P(c_k|x_i)\Big(\Sigma_k^{-1}(x_i-\mu_k)\Big) - \frac{\lambda}{2}\sum_{i,j=1}^{N}\Big(\big(P(c_k|x_i)-P(c_k|x_j)\big)\Sigma_k^{-1}(x_i-x_j)\Big)W_{ij}=0$$

By solving the equation above, one obtains the M–step re-estimation equation for μ_k:

$$\mu_k = \frac{\sum_{i=1}^{N} x_i P(c_k|x_i)}{N_k} - \frac{\lambda\sum_{i,j=1}^{N}\Big(\big(P(c_k|x_i)-P(c_k|x_j)\big)(x_i-x_j)\Big)W_{ij}}{2N_k} \tag{13}$$

where

$$N_k = \sum_{i=1}^{N} P(c_k|x_i)$$

Let $S_{i,k}=(x_i-\mu_k)(x_i-\mu_k)^T$, we have:

$$(x_i-\mu_k)^T\Sigma_k^{-1}(x_i-\mu_k) = \mathrm{Tr}\left(S_{i,k}\Sigma_k^{-1}\right) = \mathrm{Tr}\left(\Sigma_k^{-1}S_{i,k}\right)$$

where $\mathrm{Tr}(\cdot)$ denotes the trace of a matrix. We can rewrite the Eq. (11) as:

$$\tilde{Q}_1(\Theta) = \frac{1}{2}\sum_{i=1}^{N}\sum_{k=1}^{K} P(c_k|x_i)\Big(\log(|\Sigma_k^{-1}|) - \mathrm{Tr}\left(\Sigma_k^{-1}S_{i,k}\right)\Big)$$

and

$$Q_2(\Theta) = \frac{\lambda}{4}\sum_{i,j=1}^{N}\sum_{k=1}^{K}\Bigg(\big(P(c_k|x_i)-P(c_k|x_j)\big)\Big(\mathrm{Tr}\left(\Sigma_k^{-1}S_{j,k}\right)-\mathrm{Tr}\left(\Sigma_k^{-1}S_{i,k}\right)\Big)\Bigg)W_{ij}$$

By taking the derivative of Eq. (11) with respect to Σ_k^{-1} and setting it to zero[1], we get:

$$\frac{1}{2}\sum_{i=1}^{N} P(c_k|x_i)\Big(\Sigma_k - S_{i,k}\Big) = \frac{\lambda}{4}\sum_{i,j=1}^{N}\big(P(c_k|x_i)-P(c_k|x_j)\big)\big(S_{j,k}-S_{i,k}\big)W_{ij}$$

Solving the above equation, we obtain the M-step re-estimation equation for Σ_k:

$$\Sigma_k = \frac{\sum_{i=1}^{N} P(c_k|x_i)S_{i,k}}{N_k} - \frac{\lambda\sum_{i,j=1}^{N}\big(P(c_k|x_i)-P(c_k|x_j)\big)\big(S_{i,k}-S_{j,k}\big)W_{ij}}{2N_k} \tag{14}$$

[1]Note that $\partial\log|M|/\partial M = (M^{-1})^T$, $\partial\mathrm{Tr}(MN)/\partial M = N^T$ and both Σ_k and $S_{i,k}$ are symmetric matrices.

When the regularization parameter $\lambda = 0$, we can easily see the above M-step re-estimation equations (Eq. 13 and 14) boil down to the M-step in original GMM. The E-step (Eq. 7) and M-step (Eq. 9, 13 and 14) are alternated until a termination condition is met.

Experiment

In this section, several experiments were conducted to demonstrate the effectiveness of our proposed approach. We begin with the description of the data sets used in our experiment.

Data Sets

Eight real world data sets are used in the experiment. Two of them are image data (face image and hand written digit image). Another two of them are from the "The Elements of Statistical Learning" web site[2] and the rest four are all from UC Irvine Machine Learning Repository[3]. The important statistics of these data sets are summarized below (see also Table 1):

- The **MNIST** database of handwritten digits from Yann LeCun's page[4]. Here we use the test set which contains 10000 examples.
- The **Yale** face image database[5]. It containing 165 gray-scale face images from 15 individuals. Each individual has 11 images. The images demonstrate variations in lighting condition and facial expression.
- The **Waveform** model which is described in (Breiman et al. 1984). The "The Elements of Statistical Learning" web site[6] provides an instance with 800 samples. Each sample has 21 features and there is 3 classes.
- The **Vowels** data set which has 990 samples of eleven steady state vowels of British English.
- The **Libras** movement data set. It contains 15 classes of 24 instances each, where each class refers to a hand movement type in LIBRAS.
- The **Control Charts** data set. It contains 600 examples of control charts and there are six different classes of control charts.
- The **Cloud** data set contains 2048 samples. Each sample has 10 features and there is 2 classes.
- The **Breast Cancer** Wisconsin data set. It is computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image. There are 569 instances and each is described by 30 features.

[2]http://www-stat.stanford.edu/ tibs/ElemStatLearn/
[3]http://archive.ics.uci.edu/ml/
[4]http://yann.lecun.com/exdb/mnist/
[5]http://cvc.yale.edu/projects/yalefaces/yalefaces.html
[6]http://www-stat.stanford.edu/ tibs/ElemStatLearn/

Table 1: Statistics of the eight data sets

data set	size	# of features	# of classes
MNIST	10000	784	10
Yale	165	4096	15
Waveform	800	21	3
Vowels	990	10	11
Libras	360	90	15
Control Chart	600	60	6
Cloud	2048	10	2
Breast Cancer	569	30	2

Table 2: Clustering accuracy on the eight data sets (%)

Data sets	LCGMM	GMM	k-means	Ncut
MNIST	**73.6**	66.6	53.1	68.8
Yale	54.3	29.1	51.5	**54.6**
Libras	**50.8**	35.8	44.1	48.6
Chart	**70.0**	56.8	61.5	58.8
Cloud	**100.0**	96.2	74.4	61.5
Breast	**95.5**	94.7	85.4	88.9
Vowel	**36.6**	31.9	29.0	29.1
Waveform	75.3	**76.3**	51.9	52.3

Evaluation Metric

We evaluate the clustering results by comparing the obtained labels using clustering algorithms with the provided ground truth. Specific speaking, the accuracy (AC) (Cai, He, and Han 2005) is adopted to measure the performance. Given a point $\mathbf{x}_i$, let r_i and s_i represent the obtained label and the label provided by the data set, respectively. The AC is defined as follows:

$$\text{AC} = \frac{\sum_{i=1}^{N} \delta(s_i, map(r_i))}{N} \tag{15}$$

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 map(r_i) is the permutation mapping function that maps the obtained label r_i to the equivalent label from the data set. The best mapping can be realized by adopting the Kuhn-Munkres algorithm (Lovasz and Plummer 1986).

Compared Algorithms

To demonstrate how the clustering performance can be improved by our method, we compared the following four clustering algorithms:

- Locally Consistent Gaussian Mixture Model (**LCGMM**), the method proposed in this paper. There are two parameters in LCGMM algorithm: the number of nearest neighbors p and the regularization parameter λ. In our experiments, we empirically set them to 20 and 0.1, respectively. The model selection will be discussed in the later section.
- The classical Gaussian Mixture Model (**GMM**) approach (Bishop 2006).
- The traditional k**-means** algorithm.
- Spectral clustering algorithm based on normalized cut criterion (**Ncut**)(Shi and Malik 1997).

Table 3: Clustering accuracy on MNIST (%)

K	LCGMM	GMM	k-means	Ncut
2	93.5±2.1	92.3±2.0	89.7±1.9	**93.8±2.0**
3	**90.6±1.4**	88.9±1.3	79.6±2.1	89.3±2.1
4	**83.5±1.6**	78.1±1.5	67.6±1.3	75.1±1.4
5	**80.9±1.0**	75.4±0.9	67.0±1.1	76.7±0.9
6	**85.3±1.1**	79.0±1.0	65.5±1.2	79.5±0.8
7	**82.5±1.3**	73.1±0.9	61.2±0.3	76.7±0.4
8	**81.9±0.8**	72.4±0.5	59.4±0.7	74.7±0.1
9	**75.1±0.5**	67.2±0.4	55.9±0.4	70.7±0.2
10	**73.6**	66.6	53.1	68.8
Avg	**83.0**	77.0	66.6	78.4

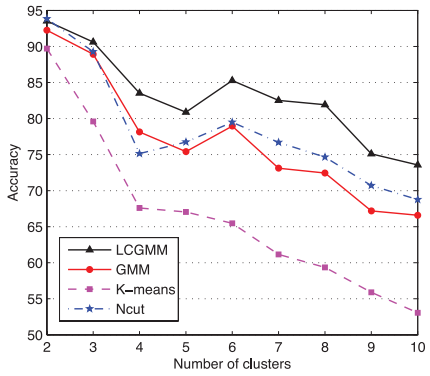

Figure 1: Clustering accuracy on MNIST data set

Among these four algorithms, LCGMM and GMM are model based approaches while k-means and Ncut are similarity based algorithms. k-means and GMM consider the Euclidean structure of the data while Ncut and LCGMM consider the intrinsic geometrical structure of the data.

Results

Table (2) shows the clustering accuracy of the four methods on all the eight data sets. As we can see, LCGMM outperforms all of its competitors on six data sets and ranks No.2 on the remaining two data sets. Specifically, LCGMM achieves 7% performance gain on MNIST over the second best method, 4.5% on Libras, 19% on Chart, 4% on Cloud, 0.8% on Breast and 14.7% on Vowel. On the remaining two data sets, LCGMM has 0.55% performance loss on Yale and 1.3% on Waveform comparing with the best method. Overall, LCGMM is the best one among all the four compared algorithms. k-means algorithm performs the worst and the performances of GMM and Ncut are comparable. Specifically, GMM is the best of the remaining three algorithms on four data sets (Cloud, Breast, Vowel and Waveform) and Ncut is the best of the three algorithms on three data sets (MNIST, Yale and Libras). Both LCGMM and GMM are model based approaches. It is interesting to note that GMM performs very poor on Yale and Libras while LCGMM performs reasonably well on these two data sets. Our reason is that traditional GMM fails to consider the local geometric structure of the data. By incorporating the locally consistent regularizer, LCGMM avoids this limitation.

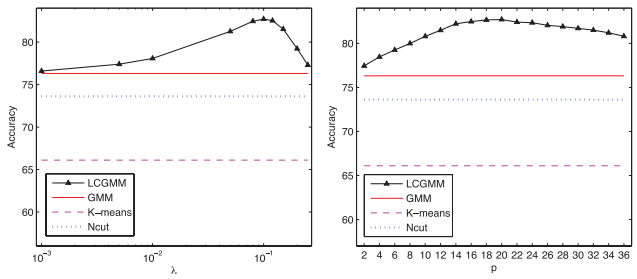

Figure 2: Performance of LCGMM vs. parameters λ and p. LCGMM performs relatively stable with respect to both parameters. It achieves the best performance when λ is around 0.1.

To further examine the behaviors of these four methods, we chose MNIST data set and conducted a through study. Table (3) and Figure (1) show the clustering accuracies of the four methods on MNIST. The evaluations were conducted with the cluster numbers ranging from two to ten. For each given cluster number K (except for 10), 20 test runs were conducted on different randomly chosen clusters and the average performance as well as the standard deviation are reported. We can clearly see that LCGMM is the best among all the four methods. It is interesting to note that the performance improvement of LCGMM over other methods increases as the number of clusters increases.

Model Selection

There are two essential parameters in our algorithm: the number of nearest neighbors p and the regularization parameter λ. The parameter p can somehow define the range of "locality" and the parameter λ decides the degree of smoothness of the model on this graph. We already know that LCGMM boils down to the original GMM when $\lambda = 0$.

Figure 2 shows how the average performance of LCGMM on MNIST varies with the parameters λ and p, respectively. As we can see, LCGMM performs relatively stable with respect to both parameters. As we have described, LCGMM uses a p-nearest neighbor graph to capture the local geometric structure of the data space. The success of LCGMM relies on how the assumption that a data point shares the same label with its p-nearest neighbor holds. It is expected that performance of LCGMM decreases as the p increases (after p is larger than 20).

Conclusions

We have introduced a novel algorithm, called Locally Consistent Gaussian Mixture Model, for clustering. It takes into consideration of the intrinsic geometry of the marginal distribution by incorporating a regularizer into the log-likelihood function, aiming at smoothing the conditional probability distribution along the geodesics of data manifold. Specifically, we construct a nearest neighbor graph to detect the underlying nonlinear manifold structure and use KL-Divergence to measure the distance between the posterior probabilities. Experimental results on eight real world data sets show the effectiveness of our method.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China under Grants 60905001, 60702072 and 90920303, National Key Basic Research Foundation of China under Grant 2009CB320801. Any opinions, findings, and conclusions expressed here are those of the authors and do not necessarily reflect the views of the funding agencies.

References

Belkin, M., and Niyogi, P. 2001. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems 14*.

Belkin, M.; Niyogi, P.; and Sindhwani, V. 2006. Manifold regularization: A geometric framework for learning from examples. *Journal of Machine Learning Research* 7:2399–2434.

Bishop, C. M. 2006. *Pattern Recognition and Machine Learning*.

Breiman, L.; Friedman, J.; Stone, C. J.; and Olshen, R. A. 1984. *Classification and Regression Trees*. Chapman & Hall/CRC.

Cai, D.; He, X.; and Han, J. 2005. Document clustering using locality preserving indexing. *IEEE Transactions on Knowledge and Data Engineering* 17(12):1624–1637.

Cai, D.; He, X.; and Han, J. 2010. Locally consistent concept factorization for document clustering. *IEEE Transactions on Knowledge and Data Engineering* To appear.

Cai, D.; Wang, X.; and He, X. 2009. Probabilistic dyadic data analysis with local and global consistency. In *ICML'09*.

Chung, F. R. K. 1997. *Spectral Graph Theory*, volume 92 of *Regional Conference Series in Mathematics*.

Dempster, A. P.; Laird, N. M.; and Rubin, D. B. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*.

Duda, R. O.; Hart, P. E.; and Stork, D. G. 2000. *Pattern Classification*. Wiley-Interscience Publication.

He, X.; Cai, D.; Shao, Y.; Bao, H.; and Han, J. 2010. Laplacian regularized gaussian mixture model for data clustering. *IEEE Transactions on Knowledge and Data Engineering* To appear.

Jain, A. K.; Murty, M. N.; and Flynn, P. J. 1999. Data clustering: a review. *ACM Comput. Surv.*

Lovasz, L., and Plummer, M. 1986. *Matching Theory*. North Holland, Budapest: Akadémiai Kiadó.

Ng, A. Y.; Jordan, M.; and Weiss, Y. 2001. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*.

Roweis, S., and Saul, L. 2000. Nonlinear dimensionality reduction by locally linear embedding. *Science* 290(5500):2323–2326.

Shi, J., and Malik, J. 1997. Normalized cuts and image segmentation. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*.

Tenenbaum, J.; de Silva, V.; and Langford, J. 2000. A global geometric framework for nonlinear dimensionality reduction. *Science* 290(5500):2319–2323.

Zhou, D.; Bousquet, O.; Lal, T.; Weston, J.; and Schölkopf, B. 2003. Learning with local and global consistency. In *NIPS 16*.

Zhu, X., and Lafferty, J. 2005. Harmonic mixtures: combining mixture models and graph-based methods for inductive and scalable semi-supervised learning. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, 1052–1059.