Two-way Gaussian Mixture Models for High Dimensional Classification

Mu Qiao* Jia Li†

Abstract

Mixture discriminant analysis (MDA) has gained applications in a wide range of engineering and scientific fields. In this paper, under the paradigm of MDA, we propose a two-way Gaussian mixture model for classifying high dimensional data. This model regularizes the mixture component means by dividing variables into groups and then constraining the parameters for the variables in the same group to be identical. The grouping of the variables is not pre-determined, but optimized as part of model estimation. A dimension reduction property for a two-way mixture of distributions from a general exponential family is proved. Estimation methods for the two-way Gaussian mixture with or without missing data are derived. Experiments on several real data sets show that the parsimonious two-way mixture often outperforms a mixture model without variable grouping; and as a byproduct, significant dimension reduction is achieved.

Keywords: Two-way mixture model; Mixture of Gaussian distributions; High dimensional classification; Variable grouping; Dimension reduction

1 Introduction

Mixture discriminant analysis (MDA) developed by Hastie and Tibshirani [1] has enjoyed wide spread applications in engineering, for instance, to verify speakers [2], to classify types of limb motion [3], to predict topics of news articles [4], and to tag online text documents [5]. The prominence in broad applications held by mixture models speaks for their appeals, which come from several intrinsic strengths of the generative modeling approach to classification as well as the power of mixture modeling as a density estimation method for multivariate data [6].

Although discriminative approaches to classification, e.g., support vector machine [7], are often argued to be more favorable because they optimize the classification boundary directly, generative modeling methods hold multiple practical advantages including the ease of handling a large number of classes, the convenience of incorporating domain expertise, and the minimal effort required to treat new classes in an incremental learning environment. The mixture model, in particular, is inherently related to clustering or quantization if each mixture component is associated with one cluster [8, 9, 10]. This insight was exploited by Li and Wang [11] to construct a mixture-type density for sets of weighted and unordered vectors that form a metric but not vector space, another evidence for the great flexibility of mixture modeling.

As with other approaches to classification, many research efforts on MDA revolve around the issue of high dimensionality. For the Gaussian mixture, the issue boils down to the robust estimation of the component-wise covariance matrix and mean vector. Earlier work focused more on the covariance because the maximum likelihood estimation often yields singular or nearly singular matrices when the dimension

*Mu Qiao is a Ph.D student in Computer Science and Engineering and an MS student in Statistics at Penn State University. Email: muq103@cse.psu.edu

†Jia Li is an Associate Professor of Statistics and (by courtesy) Computer Science and Engineering at Penn State University. Email: jiali@stat.psu.edu

is high, causing numerical breakdown of MDA. The same issue arises for linear or quadratic discriminant analysis (LDA, QDA), less seriously than for MDA though. An easy way to tackle this problem is to use diagonal covariance matrices. Friedman [12] developed a regularized discriminant analysis in which the component-wise covariance matrix is shrunk towards a diagonal or a common covariance matrix across components. Banfield and Raftery [9] decomposed the covariance matrix into parts corresponding to the volume, orientation, and shape of each component. Parsimonious mixture models were then proposed by assuming shared properties in those regards for the covariances in different components.

Recently, research efforts have been devoted to constraining the mean vectors as well. It is found that when the dimension is extremely high, for instance, larger than the sample size, regularizing the mean vector results in better classification even when the covariance structure is maintained highly parsimoniously or when covariance is not part of the estimation. For instance, Guo et al. [13] extended the centroid shrinkage idea of Tibshirani et al. [14] and proposed to regularize the class means under the LDA model. Some dimensions of the mean vectors are shrunk to common values so that they become irrelevant to class labels, achieving variable selection. Pan and Shen [15] employed the L_1 norm penalty to shrink mixture component means towards the global mean so that some variables in the mean vectors are identical across components, again resulting in variable selection. Along this line of research, Wang and Zhu [16] proposed the L_∞ norm penalty to regularize the means and select variables.

In this paper, we investigate another approach to regularizing the mixture component means. Specifically, we divide the variables into groups and assume identical values for the means of variables in the same group under one component. This idea was first explored by Li and Zha [4] for a mixture of Poisson distributions (more accurately, a product of independent Poisson distributions for multivariate data). They called such a model a two-way mixture, reflecting the observation that the mixture components induce a partition of the sample points, each usually corresponding to a row in a data matrix, while the variable groups form a partition of the columns in the matrix. Another related line of research is the simultaneous clustering or biclustering approach [17, 18], where sample points and their variables are simultaneously clustered to improve the clustering effectiveness and cluster interpretability. Lazzeroni and Owen [17] introduced the notion of plaid model which leads to simultaneous clustering with overlapping. Unlike two-way mixture, the simultaneous clustering approach focuses on a set of data samples and does not provide a generative model for an arbitrary sample point, in a strict sense. Here, we study the two-way mixture of Gaussians for continuous data and derive its estimation method. The issue of missing data that tends to arise when the dimension is extremely high is addressed. Experiments are conducted on several real data sets with moderate to very high dimensions. A dimension reduction property of the two-way mixture of distributions from any exponential family is proved.

Our motivation for exploring the two-way mixture is multifold. First, in engineering applications, very differently from science where we seek a simple explanation, black box classifiers are well accepted. In scientific studies, the features (aka variables) often have natural meanings, for instance, each feature corresponds to a gene; and the purpose is to reveal the relationship between the features and some other phenomenon. Variable selection is desired because it identifies features relevant to the phenomenon. In engineering systems, the features are often defined and supplied artificially; and the purpose is to achieve good prediction performance with as much information as possible. Therefore selecting features may not be a concern, but how to combine their forces is critical. We thus focus on a parsimonious mixture model that can be more robustly estimated, but not implying the discard of any features. Moreover, estimating the two-way mixture model is computationally less intensive than selecting variables using L_1 or L_∞ norm penalty.

Second, from model estimation perspective, assuming identical means for variables in the same group is essentially to quantize the unconstrained means of the variables and replace those means by a smaller number of quantized values. Consider the following hypothetical setup. Suppose the means of k variables X_1, ..., X_k are independently sampled from a normal distribution $\mathcal{N}(0, s^2)$. Denote the means by μ_1, ..., μ_k. Suppose X_j, $j = 1$, ..., k, are independently sampled n times from $\mathcal{N}(\mu_j, \sigma^2)$, the samples denoted

by $x_j^{(i)}$, $i = 1, ..., n$, $j = 1, ..., k$. Without regularization, the maximum likelihood estimation for μ_j is $\hat{\mu}_j = \sum_{i=1}^{n} x_j^{(i)}/n$. The total expected squared error is $E\left[\sum_{j=1}^{k}(\mu_j - \hat{\mu}_j)^2\right] = k\sigma^2/n$. On the other hand, if the constrained estimator $\hat{\mu} = \sum_{j=1}^{k}\sum_{i=1}^{n} x_j^{(i)}/nk$ is used for all the μ_j's, the total expected squared error is $E\left[\sum_{j=1}^{k}(\mu_j - \hat{\mu})^2\right] = (k-1)s^2 + \sigma^2/n$. We see that if $s^2 < \sigma^2/n$, the constrained estimator $\hat{\mu}$ yields lower total expected squared error than $\hat{\mu}_j$, $j = 1, ..., k$. The rationale for the quantization strategy is that if we substitute $\hat{\mu}_j$'s as the true μ_j's and divide them into groups of similar values, the μ_j's in the same group are considered to be sampled from a distribution with a small s^2, and hence have a good chance of satisfying the inequality above.

The rest of the paper is organized as follows. The two-way Gaussian mixture model is formulated in Section 2. We consider two cases for the component covariance matrices: diagonal for very high dimensions and unconstrained for moderately high dimensions. In Section 3, for the two-way mixture of distributions from any exponential family, a dimension reduction property is presented, with proof in the Appendix. The estimation algorithm and the method to treat missing data are described in Section 4. Experimental results with comparisons are provided in Section 5. Finally, we conclude and discuss future work in Section 6.

2 Two-way Gaussian Mixture Model

Let $\mathbf{X} = (X_1, X_2, ..., X_p)^t$, where p is the dimension of the data, and the class label of $\mathbf{X}$ be $Y \in \mathscr{K} = \{1, 2, ..., K\}$. A sample of $\mathbf{X}$ is denoted by $\mathbf{x} = (x_1, x_2, ..., x_p)^t$. We present the notation for a general Gaussian mixture model assumed for each class before introducing the two-way model. The joint distribution of $\mathbf{X}$ and Y under a Gaussian mixture is $f(\mathbf{X} = \mathbf{x}, Y = k) = a_k f_k(\mathbf{x}) = a_k \sum_{r=1}^{R_k} \pi_{kr}\phi(\mathbf{x}|\boldsymbol{\mu}_{kr}, \boldsymbol{\Sigma}_{kr})$, where a_k is the prior probability of class k, satisfying $0 \le a_k \le 1$ and $\sum_{k=1}^{K} a_k = 1$, and $f_k(\mathbf{x})$ is the within-class density for $\mathbf{X}$. R_k is the number of mixture components used to model class k, and the total number of mixture components for all the classes is $M = \sum_{k=1}^{K} R_k$. Let π_{kr} be the mixing proportions for the rth component in class k, $0 \le \pi_{kr} \le 1$, $\sum_{r=1}^{R_k} \pi_{kr} = 1$. $\phi(\cdot)$ denotes the pdf of a Gaussian distribution: $\boldsymbol{\mu}_{kr}$ is the mean vector for component r of class k and $\boldsymbol{\Sigma}_{kr}$ is the corresponding covariance matrix. To avoid notational complexity, we write the above mixture model equivalently as follows

$$f(\mathbf{X} = \mathbf{x}, Y = k) = \sum_{m=1}^{M} \pi_m p_m(k)\phi(\mathbf{x}|\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m) \ , \tag{1}$$

where $1 \le m \le M$ is the new component label assigned in a stacked manner to all the components in all the classes. The prior probability for the mth component $\pi_m = a_k\pi_{kr}$ if m is the new label for the rth component in the kth class. Specifically, let $\overline{R}_k = \sum_{k'=1}^{k} R_{k'}$ and $\overline{R}_0 = 0$. Then $M = \overline{R}_K$. Let the set $\mathscr{R}_k = \{\overline{R}_{k-1}+1, \overline{R}_{k-1}+2, ..., \overline{R}_k\}$ be the set of new labels assigned to the R_k mixture components of class k. The quantity $p_m(k) = 1$ if component m "belongs to" class k and 0 otherwise. That is, $p_m(k) = 1$ only for $m \in \mathscr{R}_k$, which ensures that the density of $\mathbf{X}$ within class k is a weighted sum over only the components inside class k. Moreover, denote the associated class of component m by $b(m)$. If $p_m(k) = 1$, $b(m) = k$. Then we have $a_k = \sum_{m\in\mathscr{R}_k} \pi_m$ and $\pi_{kr} = \pi_{\overline{R}_{k-1+r}}/a_k$.

Two-way Mixture with Diagonal Covariance Matrices: If the data dimension is very high, we adopt diagonal covariance matrix $\boldsymbol{\Sigma}_m = diag(\sigma_{m,1}^2, ..., \sigma_{m,p}^2)$, i.e., the variables are independent within each mixture component. Model (1) becomes

$$f(\mathbf{X} = \mathbf{x}, Y = k) = \sum_{m=1}^{M} \pi_m p_m(k) \prod_{j=1}^{p} \phi(x_j|\mu_{m,j}, \sigma_{m,j}^2) \ . \tag{2}$$

In Model (2), the variables are in general not independent within each class as one class may contain multiple mixture components. To approximate the class conditional density, the restriction of diagonal

covariance matrix on each component can be compensated by having more additive components. With diagonal covariance matrices, it is convenient to treat missing values, a particularly useful trait for applications highly prone to missing values, for instance, microarray gene expression data where more than 90% of the genes miss some measurements [19]. We will show that the two-way Gaussian mixture model with diagonal covariance matrices can handle missing data effectively. On the other hand, for moderately high dimensional data, we will propose shortly a two-way mixture with full covariance matrices.

For Model (2), we need to estimate parameters $\mu_{m,j}$ and $\sigma^2_{m,j}$ for each dimension j in each mixture component m. When the dimension p is very high, sometimes $p \gg n$, we may need a more parsimonious model. We now introduce the two-way mixture model with a grouping structure imposed on the variables. In order not to confuse with the clustering structure of samples implied by the mixture components, we follow the naming convention used by Li and Zha [4]: "cluster" refers to a variable cluster and "component" means a component in the mixture distribution. For each class k, suppose the variables are grouped into L clusters. The cluster identity of variable j in class k is denoted by $c(k,j) \in \{1,2,...,L\}$, $k = 1$, ..., K, $j = 1$, ..., p, referred to as the *cluster assignment function*. The two-way Gaussian mixture is formulated as follows:

$$f(\mathbf{X} = \mathbf{x}, Y = k) = \sum_{m=1}^{M} \pi_m p_m(k) \prod_{j=1}^{p} \phi(x_j | \mu_{m,c(b(m),j)}, \sigma^2_{m,c(b(m),j)}) \ . \tag{3}$$

Within each mixture component, variables belonging to the same cluster have identical parameters since the second subscripts for μ and σ^2 are given by the variable cluster assignment function. Thus, for a fixed mixture component m, only L, rather than p, μ's and σ^2's need to be estimated. Also note that $c(k,j)$ is not pre-specified, but optimized as part of model estimation. In our current study, the cluster assignment function $c(k,j)$ depends on class label k, but extension to a component specific assignment is straightforward.

Two-way Mixture with Full Covariance Matrices: When the data dimension is moderately high, one may suspect that diagonal covariance matrices adopted in Model (2) are not efficient for modeling the data and full covariance matrices can fit the data better with a substantially fewer number of components in the mixture. In order to exploit a two-way mixture as entailed in (3), we propose to first model the within-class density by a Gaussian mixture $f(\mathbf{X} = \mathbf{x}, Y = k) = \sum_{m=1}^{M} \pi_m p_m(k) \phi(\mathbf{x} | \boldsymbol{\mu}_m, \widetilde{\boldsymbol{\Sigma}}_k)$, where $\widetilde{\boldsymbol{\Sigma}}_k$ is an unconstrained common covariance matrix across all the components in class k. Once $\widetilde{\boldsymbol{\Sigma}}_k$ is identified, a linear transform (a "whitening" operation) can be applied to $\mathbf{X}$ so that the transformed data follow a mixture with component-wise diagonal covariance matrix, more specifically, the identity matrix $\mathbf{I}$. Assume $\widetilde{\boldsymbol{\Sigma}}_k$ is non-singular and hence positive definite, we can write $\widetilde{\boldsymbol{\Sigma}}_k = (\widetilde{\boldsymbol{\Sigma}}_k^{\frac{1}{2}})^t (\widetilde{\boldsymbol{\Sigma}}_k^{\frac{1}{2}})$, where $\widetilde{\boldsymbol{\Sigma}}_k^{\frac{1}{2}}$ is full ranked. Let $W_k = ((\widetilde{\boldsymbol{\Sigma}}_k^{\frac{1}{2}})^t)^{-1}$ and $\mathbf{Z} = W_k \mathbf{X}$. The distribution of $\mathbf{Z}$ and Y is

$$g(\mathbf{Z} = \mathbf{z}, Y = k) = \sum_{m=1}^{M} \pi_m p_m(k) \phi(\mathbf{z} | W_k \boldsymbol{\mu}_m, \mathbf{I}) \ . \tag{4}$$

In the light of the above model for $\mathbf{Z}$, (3) is a plausible parsimonious model to impose on $\mathbf{Z}$ by the idea of forming variable clusters. In fact, the covariance matrix $\mathbf{I}$ in (4) is not as general as the diagonal covariance matrix assumed in Model (3). In our study, we adopt Model (3) directly for $\mathbf{Z}$ instead of fixing the covariance matrix to $\mathbf{I}$, allowing more flexibility in modeling. In initialization, however, it is reasonable to set the mean of $\mathbf{Z}$ in component m as $\boldsymbol{\nu}_m = W_k \boldsymbol{\mu}_m$ and the covariance matrix $\boldsymbol{\Sigma}_m = \mathbf{I}$.

In summary, let the two-way Gaussian mixture for $\mathbf{Z}$ be

$$g(\mathbf{Z} = \mathbf{z}, Y = k) = \sum_{m=1}^{M} \pi_m p_m(k) \prod_{j=1}^{p} \phi(z_j | \nu_{m,c(b(m),j)}, \sigma^2_{m,c(b(m),j)}) \ .$$

Since $\mathbf{X} = W_k^{-1}\mathbf{Z}$, we can transform $\mathbf{Z}$ back to $\mathbf{X}$ and obtain the distribution for the original data:

$$f(\mathbf{X} = \mathbf{x}, Y = k) = \sum_{m=1}^{M} \pi_m p_m(k) \phi(\mathbf{x}|W_k^{-1}\boldsymbol{\nu}_m, (W_k^{-1})\boldsymbol{\Sigma}_m(W_k^{-1})^t), \tag{5}$$

where $\boldsymbol{\nu}_m = (\nu_{m,c(b(m),1)}, ..., \nu_{m,c(b(m),p)})^t$, and $\boldsymbol{\Sigma}_m = diag\left(\sigma^2_{m,c(b(m),1)}, ..., \sigma^2_{m,c(b(m),p)}\right)$.

We thus have two options when employing the two-way Gaussian mixture: (a) if the data dimension is too high for using a full covariance matrix, we assume diagonal covariance matrix as in Model (3); (b) if a full covariance matrix is desired, we suggest Model (5) which involves essentially whitening all the mixture components and then assuming Model (3) for the transformed data.

As a final note, to classify a sample $\mathbf{X} = \mathbf{x}$, the Bayes classification rule is used: $\widehat{y} = \text{argmax}_k f(Y = k|\mathbf{X} = \mathbf{x}) = \text{argmax}_k f(\mathbf{X} = \mathbf{x}, Y = k)$.

3 Dimension Reduction

In this section, we present a dimension reduction property for the two-way mixture of distributions from a general exponential family. Consider a univariate distribution from an exponential family assumed for the jth variable in $\mathbf{X}$: $\phi(x_j|\boldsymbol{\theta}) = \exp\left(\sum_{s=1}^{S} \eta_s(\boldsymbol{\theta})T_s(x_j) - \mathbf{B}(\boldsymbol{\theta})\right) h(x_j)$. The parameter vector $\boldsymbol{\theta}$ is re-parameterized as the *canonical parameter vector* $\boldsymbol{\eta}(\boldsymbol{\theta})$ and the *cumulant generating function* $\mathbf{B}(\boldsymbol{\theta})$. $\mathbf{T}(x_j) = (T_1(x_j), ..., T_S(x_j))^t$ is the sufficient statistic vector of x_j with size S. For a two-way mixture model, variables in the same cluster within any class share parameters. We thus have the following model:

$$f(\mathbf{X} = \mathbf{x}, Y = k) \quad = \quad \sum_{m=1}^{M} \pi_m p_m(k) \prod_{j=1}^{p} \phi\left(x_j|\boldsymbol{\theta}_{m,c(b(m),j)}\right) \ . \tag{6}$$

Recall that $b(m)$ is the class which component m belongs to and $c(b(m), j)$ is the cluster index the jth variable belongs to. Model (6) implies a dimension reduction property for the classification purpose, formally stated below.

Theorem 3.1 *For x_j's in the lth variable cluster of class k, l = 1,..., L, k = 1, ..., K, define $\overline{\mathbf{T}}_{l,k}(\mathbf{x}) = \sum_{j:c(k,j)=l} \mathbf{T}(x_j)$, where $\mathbf{T}(x_j)$ is the sufficient statistic vector for x_j under the distribution from the exponential family. Given $\overline{\mathbf{T}}_{l,k}(\mathbf{x})$, l = 1,...,L, k = 1, ..., K, the class label Y is conditionally independent of $\mathbf{x} = (x_1, x_2, ..., x_p)^t$.*

This theorem results from the intrinsic fact about the exponential family: the size of the sufficient statistic is fixed when the sample size increases. Here, to be distinguished from the number of data points, the sample size refers to the number of variables in one cluster because within a single data point, these variables can be viewed as i.i.d. samples. Detailed proof for the theorem is provided in Appendix A.

In the above statement of the theorem, for notation simplicity, we assume the number of variable clusters under each class is always L. It is trivial to extend to the case where different classes may have different numbers of variable clusters. Since the size of the sufficient statistic $\mathbf{T}(x_j)$ is S, the total number of statistics needed to optimally predict class label Y is SKL. In the special case of Gaussian distribution, the size of $\mathbf{T}(x_j)$ is $S = 2$, where $T_1(x_j) = x_j$ and $T_2(x_j) = x_j^2$. In the experiment section, we will show that similar or considerably better classification performance can be achieved with $SKL \ll p$. If the way the variables are clustered is identical across different classes, i.e., $c(k, j)$ is invariant with k, the dimension sufficient for predicting class label Y is SL since $\overline{\mathbf{T}}_{l,k}$'s are identical for different k's.

4 Model Estimation

To estimate Model (3), the EM algorithms with or without missing data are derived.

Estimation without Missing Data: The parameters to be estimated include prior probabilities of the mixture components π_m, the Gaussian parameters $\mu_{m,l}$, $\sigma^2_{m,l}$, $m = 1, ..., M$, $l = 1, ..., L$, and the cluster assignment function $c(k, j) \in \{1, 2, ..., L\}$, $k = 1, ..., K$, $j = 1, ..., p$. Denote the collection of all the parameters and the cluster assignment function $c(k, j)$ at iteration t by $\psi_t : \psi_t = \{\pi_m^{(t)}, \mu_{m,l}^{(t)}, \sigma^{2\,(t)}_{m,l}, c^{(t)}(k, j) : m = 1, ..., M, l = 1, ..., L, k = 1, ..., K, j = 1, ..., p\}$. Let the training data be $\{(\mathbf{x}^{(i)}, y^{(i)}) : i = 1, ..., n\}$. The EM algorithm comprises the following two steps:

1. *E-step*: Compute the posterior probability, $q_{i,m}$ of each sample i belonging to component m.

$$q_{i,m} \propto \pi_m^{(t)} p_m(y^{(i)}) \prod_{j=1}^{p} \phi\left(x_j^{(i)} | \mu^{(t)}_{m,c^{(t)}(b(m),j)}, \sigma^{2\,(t)}_{m,c^{(t)}(b(m),j)}\right), \text{ subject to } \sum_{m=1}^{M} q_{i,m} = 1 .$$

2. *M-step*: Update ψ_{t+1} by $\psi_{t+1} = \underset{\psi'}{\text{argmax}} Q(\psi'|\psi_t)$, where $Q(\psi'|\psi_t)$ is given below. Specifically, the updated parameters are given by Eqs.(8) $\sim$ (11) to be derived shortly.

$$Q(\psi'|\psi_t) = \sum_{i=1}^{n} \sum_{m=1}^{M} q_{i,m} \log\left(\pi'_m p_m(y^{(i)}) \prod_{j=1}^{p} \phi\left(x_j^{(i)} | \mu'_{m,c'(b(m),j)}, \sigma^{2\prime}_{m,c'(b(m),j)}\right)\right) . \tag{7}$$

Based on (7), it is easy to see that the optimal $\pi_m^{(t+1)}$, subject to $\sum_{m=1}^{M} \pi_m^{(t+1)} = 1$, are given by

$$\pi_m^{(t+1)} \propto \sum_{i=1}^{n} q_{i,m} , \quad m = 1, ..., M . \tag{8}$$

The optimization of $\mu_{m,l}^{(t+1)}$, $\sigma^{2\,(t+1)}_{m,l}$, $m = 1, ..., M$, $l = 1, ..., L$, and $c^{(t+1)}(k, j)$, $k = 1, ..., K$, $j = 1, ..., p$, requires a numerical procedure. Our approach is to optimize the Gaussian parameters and the cluster assignment function alternately, fixing one in each turn. Let $\eta_{k,l}$ be the number of j's such that $c(k, j) = l$. In one round, $\mu_{m,l}^{(t+1)}$, $\sigma^{2\,(t+1)}_{m,l}$, and $c^{(t+1)}(k, j)$ are updated by the following equations. Each maximizes $Q(\psi_{t+1}|\psi_t)$ when the others are fixed.

$$\mu_{m,l}^{(t+1)} = \frac{\sum_{i=1}^{n} q_{i,m} \sum_{j:c^{(t)}(b(m),j)=l} x_j^{(i)}}{\eta_{b(m),l} \sum_{i=1}^{n} q_{i,m}} \tag{9}$$

$$\sigma^{2\,(t+1)}_{m,l} = \frac{\sum_{i=1}^{n} q_{i,m} \sum_{j:c^{(t)}(b(m),j)=l} (x_j^{(i)} - \mu_{m,l}^{(t+1)})^2}{\eta_{b(m),l} \sum_{i=1}^{n} q_{i,m}} \tag{10}$$

$$c^{(t+1)}(k, j) = \underset{l \in \{1,...,L\}}{\text{argmax}} \sum_{i=1}^{n} \sum_{m \in \mathscr{R}_k} q_{i,m} \left[-\frac{(x_j^{(i)} - \mu_{m,l}^{(t+1)})^2}{2\sigma^{2\,(t+1)}_{m,l}} - \log|\sigma_{m,l}^{(t+1)}| \right] . \tag{11}$$

The optimality of Eq.(9) and Eq.(10) can be shown easily as in the derivation of the EM algorithm for a usual mixture model. Given fixed Gaussian parameters $\mu_{m,l}^{(t+1)}$ and $\sigma^{2\,(t+1)}_{m,l}$, $Q(\psi_{t+1}|\psi_t)$ can be maximized by optimizing the cluster assignment function $c^{(t+1)}(k, j)$ separately for each class k and each variable j.

See [4] for the argument that applies here likewise. The optimality of $c^{(t+1)}(k,j)$ is then obvious because of the exhaustive search through all the possible values.

Eqs.(9)$\sim$ (11) can be iterated multiple times. However, considering the computational cost of embedding this iterative procedure in the M-step, we adopt the generalized EM (GEM) algorithm [20], which ensures that $Q(\psi_{t+1}|\psi_t) \geq Q(\psi_t|\psi_t)$ rather than solving $\max_{\psi_{t+1}} Q(\psi_{t+1}|\psi_t)$. Thus, Eqs.(9)$\sim$ (11) are applied only once. To see that $Q(\psi_{t+1}|\psi_t) \geq Q(\psi_t|\psi_t)$, let $\widetilde{\psi} = \{\pi_m^{(t+1)}, \mu_{m,l}^{(t+1)}, {\sigma^2}_{m,l}^{(t+1)}, c^{(t)}(k,j) : m = 1,...,M, l = 1,...,L, k = 1,...,K, j = 1,...,p\}$. It is straightforward to show that $Q(\psi_{t+1}|\psi_t) \geq Q(\widetilde{\psi}|\psi_t) \geq Q(\psi_t|\psi_t)$ based on the optimality of Eqs.(8)$\sim$ (10) conditioned on other parameters held fixed. The computational cost for each iteration of GEM is linear in $npML$.

To initialize the estimation algorithm, we first choose R_k, the number of mixture components for each class k. If the training sample size of each class is roughly equal, we assign the same number of components to each class for simplicity. Otherwise, the number of components in a class is determined by its corresponding proportion in the whole training data set. Then we randomly assign each sample to a mixture component m in the given class of that sample. The posterior probability $q_{i,m}$ is set to 1 if sample i is assigned to component m and 0 otherwise. Also, each variable is randomly assigned to a variable cluster l in that class. With the initial posterior probabilities and the cluster assignment function given, an M-step is applied to obtain the initial parameters. If any mixture component or variable cluster happens to be empty according to the random assignment, we initialize $\mu_{m,l}$ and ${\sigma^2}_{m,l}$ by the global mean and variance. During the estimation, we bound the variances ${\sigma^2}_{m,l}$ away from zero using a small fraction of the global variance in order to avoid the singularity of the covariance matrix .

Estimation with Missing Data: When missing data exist, due to the diagonal covariance matrices assumed in Model (3), the EM algorithm requires little extra computation. The formulas for updating the parameters in the M-step bear much similarity to Eqs. (8) $\sim$ (11). The key for deriving the EM algorithm when missing data exist is to compute $Q(\psi_{t+1}|\psi_t) = E[\log f(\mathbf{v}|\psi_{t+1}) \mid \mathbf{w}, \psi_t]$, where $\mathbf{v}$ is the complete data, $\mathbf{w}$ the incomplete, and $f(\cdot)$ the density function.

When there is no real missing data, EM takes the latent component identities of the sample points as the "conceptual" missing data. When some variables actually lack measurements, the missing data as viewed by EM contain not only the conceptually missing component identities but also the physically missing values of the variables. The derivation of $Q(\psi_{t+1}|\psi_t)$ when real missing data exist is provided in Appendix B. We present the EM algorithm below. Introduce $\Lambda(\cdot)$ as the *missing indicator function*, that is, $\Lambda(x_j^{(i)}) = 1$ if the value of $x_j^{(i)}$ is not missing and 0 otherwise.

1. *E-step*: Compute the posterior probability, $q_{i,m}$, $i = 1, ..., n$, $m = 1, ..., M$. Subject to $\sum_{m=1}^{M} q_{i,m} = 1$,

$$q_{i,m} \propto \pi_m^{(t)} p_m(y^{(i)}) \prod_{j=1}^{p} \left[\Lambda(x_j^{(i)}) \phi\left(x_j^{(i)} | \mu^{(t)}_{m,c^{(t)}(b(m),j)}, {\sigma^2}^{(t)}_{m,c^{(t)}(b(m),j)}\right) + \left(1 - \Lambda(x_j^{(i)})\right)\right] . \quad (12)$$

2. *M-step*: Update the parameters in ψ_{t+1} by the following equations.

$$\pi_m^{(t+1)} \propto \sum_{i=1}^{n} q_{i,m}, \text{ subject to } \sum_{m=1}^{M} \pi_m^{(t+1)} = 1 , \quad m = 1, ..., M . \quad (13)$$

For each $m = 1, ..., M$, $l = 1, ..., L$, let $\tilde{x}_{j,m,l}^{(i)} = \Lambda(x_j^{(i)}) x_j^{(i)} + (1 - \Lambda(x_j^{(i)}))\mu_{m,l}^{(t)}$.

$$\mu_{m,l}^{(t+1)} = \frac{\sum_{i=1}^{n} q_{i,m} \sum_{j:c^{(t)}(b(m),j)=l} \tilde{x}_{j,m,l}^{(i)}}{\eta_{b(m),l} \sum_{i=1}^{n} q_{i,m}} \quad (14)$$

$${\sigma^2}_{m,l}^{(t+1)} = \frac{\sum_{i=1}^{n} q_{i,m} \sum_{j:c^{(t)}(b(m),j)=l} (\tilde{x}_{j,m,l}^{(i)} - \mu_{m,l}^{(t+1)})^2 + (1 - \Lambda(x_j^{(i)})){\sigma^2}_{m,l}^{(t)}}{\eta_{b(m),l} \sum_{i=1}^{n} q_{i,m}} . \quad (15)$$

Let $\Omega_1 = -\frac{(x_j^{(i)} - \mu_{m,l}^{(t+1)})^2}{2\sigma_{m,l}^{2\,(t+1)}} - \log|\sigma_{m,l}^{(t+1)}|$, $\Omega_2 = -\frac{\left(\mu_{m,c^{(t)}(k,j)}^{(t)} - \mu_{m,l}^{(t+1)}\right)^2 + \sigma^{2\,(t)}_{m,c^{(t)}(k,j)}}{2\sigma_{m,l}^{2\,(t+1)}} - \log|\sigma_{m,l}^{(t+1)}|$.

$$c^{(t+1)}(k,j) = \underset{l\in\{1,\ldots,L\}}{\operatorname{argmax}} \sum_{i=1}^{n} \sum_{m\in\mathscr{R}_k} q_{i,m}[\Lambda(x_j^{(i)})\Omega_1 + (1-\Lambda(x_j^{(i)}))\Omega_2] \ . \tag{16}$$

5 Experiments

In this section, we present experimental results based on three data sets with moderate to very high dimensions: (1) Microarray gene expression data; (2) Text document data; (3) Imagery data. The two-way Gaussian mixture model (two-way GMM), MDA without variable clustering (MDA-n.v.c.) and Support Vector Machine (SVM) are compared for all the three data sets. Unless otherwise noted, the covariance matrices in the mixture models are diagonal because most of the data sets are of very high dimensions, e.g., $p \gg n$. To make our presentation concise, we also recall that the total number of mixture components for all the classes is always denoted by M, and the number of variable clusters in each class is denoted by L.

Microarray Gene Expression Data: We apply the two-way Gaussian mixture model to the microarray data used by Alizadeh et al. [21]. Every sample in this data set contains the expression levels of 4026 genes. There are 96 samples divided into 9 classes. Four classes of 78 samples in total are chosen for our experiment, in particular, 42 diffuse large B-cell lymphoma (DLBCL), 16 activated blood B (ABB), 9 follicular lymphoma (FL), and 11 chronic lymphocytic leukemia (CLL). The other classes are excluded because they contain too few points. Because the sample sizes of these 4 classes are quite different, the number of mixture components used in each class is chosen according to its proportion in the training data set. We experiment with a range of values for the total number of components M. The percentage of missing values in this data set is around 5.16%. The estimation method in the case of missing data is used. We use five-fold cross validation to compute the classification accuracy.

Fig.1 shows the classification error rates obtained by MDA-n.v.c.. The minimum error rate 10.90% is achieved when $M = 6$. Due to the small sample size, the classification accuracy of MDA degrades rapidly when M increases. For comparison, Fig.1 also shows the classification error rates obtained by the two-way GMM with $L = 20$. As we can see, the two-way GMM always yields a smaller error rate than MDA-n.v.c. at any M. With $L = 20$, the two-way GMM achieves the minimum error rate 7.26% when $M = 12$. In Fig.1, when $M = 4$, i.e., one Gaussian component is used to model each class, MDA is essentially QDA and the two-way GMM is essentially QDA with variable clustering. The error rate achieved by QDA without variable clustering is 13.26%, while that by QDA with variable clustering is a smaller value of 9.48%.

Table 1: The classification error rates in percent achieved by the two-way GMM for the microarray data

Error rate (%)	$L=5$	$L=10$	$L=30$	$L=50$	$L=70$	$L=90$	$L=110$	n.v.c.
$M=4$	8.69	**7.12**	9.48	10.82	10.82	11.93	11.93	13.26
$M=18$	**7.26**	10.02	8.60	10.82	8.46	9.48	8.46	35.30
$M=36$	7.35	**5.83**	7.17	6.15	7.34	7.48	6.23	44.65

Table 1 provides the classification accuracy of two-way GMM with different values of M and L. The minimum error rate in each row is in bold font. As Table 1 shows, for each row, when the number of mixture components is fixed, the lowest error rate is always achieved by the two-way GMM. According to Theorem 3.1, this data set can be classified with accuracy 7.26% by the two-way GMM at $M = 18$ and $L = 5$ using only 40 $(2KL = 40)$ dimensions, significantly smaller than the original dimension of 4026. If homogeneous variable clustering is enforced across different classes, that is, the cluster assignment

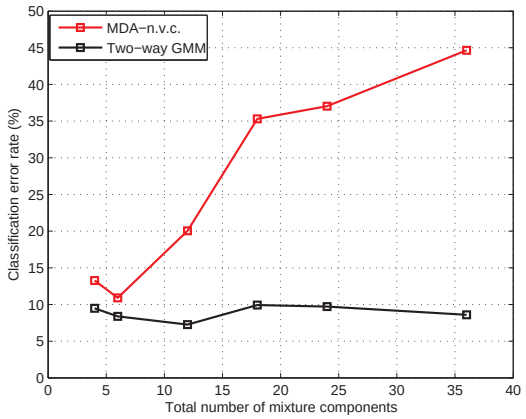

Figure 1: The classification error rates obtained for the microarray data using both MDA-n.v.c. and two-way GMM with $L = 20$ variable clusters. The total number of components M ranges from 4 to 36.

function $c(k, j)$ is invariant with class k, the classification accuracy is usually worse than the inhomogeneous clustering. Due to space limitations, we will not show the numerical results. All the results given in this section are based on inhomogeneous variable clustering.

We may use a data driven method, such as grid search and cross validation, to find the pair of M and L that gives the smallest error rate. Under some situations, the physical nature of the data may dominate the choices for M and L. For many other problems, the density of the data may be well approximated by mixture models with different values of M and L. For the purpose of classification, the mixture structure underlying the density function has no effect. It is known that discovering the true number of components assuming the distribution is precisely a mixture of Gaussian is a difficult problem and is out of the scope of this paper. Effort in this direction has been made by Tibshirani and Walther [22].

For comparison, we also apply SVM to this data set and obtain its classification accuracy with five-fold cross validation. We use the LIBSVM package [23] and the linear kernel with the default selection of the penalty parameter C. Missing values in the microarray data are replaced by the corresponding value from the nearest-neighbor sample in Euclidean distance. If the corresponding value from the nearest-neighbor sample is also missing, the next nearest sample is used. The classification error rate obtained by SVM is 0.00%. Although the minimum error rate of two-way GMM listed in Table 1, i.e., 5.83% at $M = 36$ and $L = 10$, is larger than that of SVM, it uses only 80 $(2KL = 80)$ dimensions comparing with the original dimension of 4026 used by SVM. Additionally, our focus here is not to compete with SVM, but to show that the parsimonious two-way mixture can outperform a mixture model without variable grouping.

Text Document Data: We perform experiments on the newsgroup data [24]. In this data set, there are twenty topics, each containing about 1000 documents (email messages). We use the bow toolkit to process this data set. Specifically, the UseNet headers are stripped and stemming is applied [25]. A document $\mathbf{x}^{(i)}$ is represented by a word count vector $(x_1^{(i)}, x_2^{(i)}, ..., x_p^{(i)})$, where p is the vocabulary size. The number of words occurred in the whole newsgroup data is about $78,000$. In our experiment, to classify a set of topics, we pre-select words to include in the word count vectors since many words are only related to certain topics and are barely useful for the topics chosen in the data set. We use the feature selection approach described in [4] to select the words that are of high potential for distinguishing the classes based on the variances of word counts over different classes. The feature selection in the preprocessing step is not aggressive because we still retain thousands of words. After selecting the words, we convert the word count vectors to word relative frequency vectors by normalization. Roughly half of the documents in each topic are randomly selected as training samples and the rest test samples.

We apply the two-way GMM to three different data sets, all with more than two classes. Five topics

Table 2: The classification error rates in percent achieved by the two-way GMM for the three text document data sets

(a) Data Set 1 with five classes and dimension $= 1000$

Error rate (%)	$L = 10$	$L = 30$	$L = 50$	$L = 70$	$L = 90$	$L = 110$	n.v.c.	*diff*
$M = 5$	9.19	**8.95**	9.07	9.27	9.15	9.15	**8.95**	0.00
$M = 20$	12.79	9.72	9.80	**8.58**	9.15	9.39	8.99	-0.41
$M = 60$	12.06	10.04	9.27	9.80	9.39	9.27	**8.54**	0.73

(b) Data Set 2 with five classes and dimension $= 3455$

Error rate (%)	$L = 10$	$L = 30$	$L = 50$	$L = 70$	$L = 90$	$L = 110$	n.v.c.	*diff*
$M = 5$	7.19	**6.91**	7.07	7.07	7.11	7.15	7.15	-0.24
$M = 20$	7.88	6.99	6.79	7.88	7.84	7.11	**6.06**	0.73
$M = 60$	10.91	7.03	7.43	7.72	7.43	7.35	**6.42**	0.61

(c) Data Set 3 with eight classes and dimension $= 5000$

Error rate (%)	$L = 5$	$L = 10$	$L = 20$	$L = 30$	$L = 40$	$L = 50$	n.v.c.	*diff*
$M = 8$	11.41	11.06	10.96	**10.79**	10.86	10.86	**10.79**	0.00
$M = 32$	15.58	11.71	11.11	11.66	11.24	11.91	**10.24**	0.87
$M = 96$	12.79	14.26	18.23	12.29	11.79	11.09	**11.01**	0.08

from the newsgroup data, referred to in short as, *comp.graphics, rec.sport.baseball, sci.med, sci.space, talk.politics.guns*, are used to form our first data set. Each document is represented by a vector containing the frequencies of 1000 words obtained by the feature selection approach aforementioned. In the second data set, we use the same topics as in the first one but increase the dimension of the word frequency vector to 3455. Our third data set is of dimension 5000 and contains eight topics: *comp.os.ms-windows.misc, comp.windows.x, alt.atheism, soc.religion.christian, sci.med, sci.space, sci.space, talk.politics.mideast*. In all the three data sets, the sample size of each topic in the training data set is around 500, roughly equal to that of the test data set. We assign the same number of mixture components to each class for simplicity. Only the total number of components M is specified in the discussion.

Table 2 provides the classification error rates of the two-way GMM on the three data sets with different values of M and L. When M is fixed in each row, the difference between the lowest error rate achieved by the two-way GMM and the error rate of MDA-n.v.c. is also calculated. These differences are under "*diff*" in the last column of each subtable. In Table 2(a), when $M = 5$ and 20, the lowest error rates obtained by the two-way GMM are equal to or smaller than the error rates of MDA-n.v.c.. When $M = 60$, MDA-n.v.c. gives the overall lowest error rate 8.54%, while the lowest error rate obtained by the two-way GMM is 9.27% at $L = 50$ or 110. When we increase the dimension of the word frequency vector and the number of topics to be classified, as in the second and third data sets, Table 2(b) and Table 2(c) show that the lowest error rate in each row is most of the time achieved by MDA-n.v.c.. However, the differences shown under the column of "*diff*" are always less than 1%. The performance of the two-way GMM is thus comparable to that of MDA-n.v.c., but is achieved at significantly lower dimensions. For instance, in Table 2(c), when $M = 32$, the value under "*diff*" is 0.87% and the lowest error rate of the two-way GMM is obtained at $L = 20$. According to Theorem 3.1, at $L = 20$, this data set is classified using 320 $(2KL = 320)$ dimensions versus the original dimension of 5000. Of particular interest is when $M = 5$ for the first and second data sets and $M = 8$ for the third data set. In those cases, a single component is assigned to each class, and hence MDA and the two-way GMM are essentially QDA with or without mean regularization. We find that for QDA, variable clustering results in lower error rates for the second data set and equal error rates for the other two.

Let us examine the two-way mixture models obtained for the two classes, *comp.os.ms-windows.misc* and

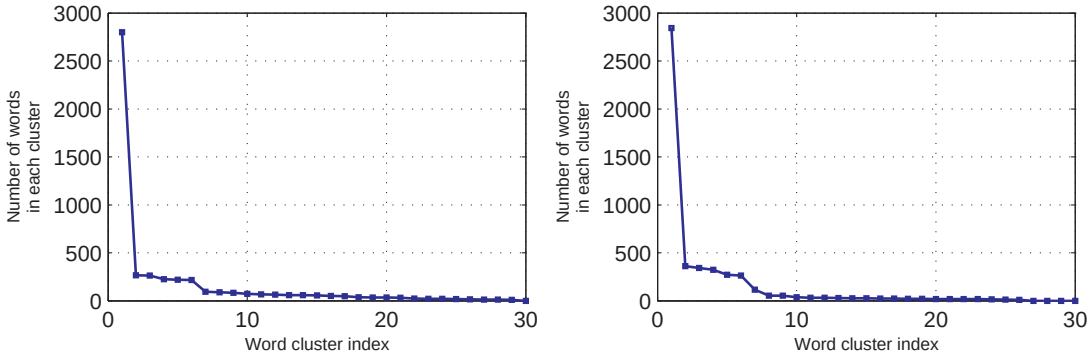

Figure 2: The sizes of the word clusters for *comp.os.ms-windows.misc* (left) and *comp.windows.x* (right).

comp.windows.x, in the third data set. Consider for example the models with $M = 32$ and $L = 30$. Fig.2 shows the number of words in each of the 30 word (aka variable) clusters for the two classes. These word clusters are indexed in an order of descending sizes. The sizes of these word clusters are highly uneven. In each case, the largest cluster accounts for more than half of the words. Moreover, the largest cluster contains words with nearly zero frequencies, which is consistent with the fact that for any particular topic class, a majority of the words almost never occur. They are thus treated indifferently by the model.

Classification error rates obtained by SVM for these three data sets are also reported. We use the linear kernel with different values of the penalty parameter C to do the classification. The value of C with the minimum cross validation error rate on the training data set is then selected and used for the final classification on the test data set. The SVM classification error rates on these three data sets are 7.98% (Data Set 1), 5.98% (Data Set 2) and 9.67% (Data Set 3), respectively. Comparing with the results listed in Table 2, SVM is only slightly better than MDA-n.v.c. and two-way GMM. However, two-way GMM achieves these error rates with a significantly smaller number of dimensions. Also, SVM is computationally more expensive and not scalable when the number of classes is large. Unlike two-way GMM, SVM does not provide a model for each class, which in some applications may be needed for descriptive purpose.

Imagery Data: The data set we used contains 1400 images each represented by a 64 dimensional feature vector. The original images contain 256×384 or 384×256 pixels. The feature vectors are generated as follows. We first divide each image into 16 even sized blocks (4×4 division). For each of the 16 blocks, the average L, U, V color components are computed. We also add the percentage of edge points in each block as a feature. The edges are detected by thresholding the intensity gradient at every pixel. In summary, every block has 4 features, 64 features in total for the entire image. These 1400 images come from 5 classes of different semantics: *mountain scenery* (300), *women* (300), *flower* (300), *city scene* (300), and *beach scene* (200), where the numbers in the parenthesis indicate the sample size of each class. Five-fold cross-validation is used to compute the classification accuracy. We use the same number of mixture components to model each class.

Table 3 lists the classification error rates obtained by the two-way GMM with a range of values for M and L. When M is fixed, as L increases, the error rates of the two-way GMM tend to decrease. In Table 3, the lowest error rate in each row is achieved by the two-way GMM. For this data set, because $2KL > 64$, dimension reduction is not obtained according to Theorem 3.1. However, the total number of parameters in the model is much reduced due to variable clustering, especially when M is large.

Since the dimension of the imagery data is moderate, at least comparing with the previous two data collections, we also experiment with the two-way GMM with full covariance matrices, that is, Model (5) in Section 2. Table 4 provides the classification error rates obtained by this model. When M is fixed, the lowest error rates are achieved by two-way GMM except at $M = 10$ and $M = 20$. Comparing Table 3 with Table 4, the performance of the two-way GMM with full covariance matrices is slightly worse than the

two-way GMM with diagonal covariance matrices. In other applications, it has also been noted that using diagonal covariance matrices often is not inferior to full covariance matrices even at moderate dimensions. One reason is that the restriction on covariance can be compensated by having more components. It is thus difficult to observe obvious improvement by relaxing the covariance.

We apply SVM with a *radial basis function* (RBF) kernel to the imagery classification problem. The penalty parameter C and the kernel parameter γ are identified by a grid search using cross validation. The final SVM error rate with five-fold cross validation is 31.00%. In Table 3, the minimum error rate of two-way GMM is 32.43% at $M = 40$ and $L = 36$. Similar to the previous examples, the classification accuracies of SVM and two-way GMM for the imagery data are very close. We also apply a variable selection based SVM to this classification problem since the dimension of the imagery data is moderately high. The wrapper subset evaluation method [26] and forward best-first search in WEKA [27] are employed to select the optimal subset of variables. In the wrapper subset evaluation method, the classification accuracy of SVM is used to measure the goodness of a particular variable subset. The final classification is obtained by applying SVM to the data with selected variables. For the SVMs involved in the variable selection scheme, the kernel function and the parameters are the same as those for the SVM without variable selection. The best subset of variables is of size 21, yielding a five-fold cross validation error rate of 34.93%. Comparing with the minimum error rates listed in Table 3 and Table 4, i.e., 32.43% ($M = 40$ and $L = 36$) and 33.21% ($M = 40$ and $L = 56$), the performance of SVM with variable selection is slightly worse than that of two-way GMM.

Table 3: The classification error rates in percent achieved by the two-way GMM for the imagery data

Error rate (%)	$L = 8$	$L = 12$	$L = 16$	$L = 24$	$L = 36$	$L = 48$	$L = 52$	$L = 56$	n.v.c.
$M = 5$	45.50	44.00	44.57	44.21	44.64	**43.50**	43.93	43.64	43.79
$M = 10$	40.29	37.86	36.93	35.57	35.43	35.07	35.50	**35.00**	35.57
$M = 20$	35.21	36.29	35.64	34.93	34.79	35.07	36.00	**33.93**	37.36
$M = 30$	35.43	36.07	34.86	34.64	**33.00**	34.57	34.36	34.93	34.64
$M = 40$	38.79	37.07	36.36	34.79	**32.43**	35.64	36.00	35.36	36.21
$M = 50$	37.50	35.50	33.93	34.07	**33.21**	34.14	34.93	34.93	36.50

Table 4: The classification error rates in percent achieved by the two-way GMM with full covariance matrices for the imagery data

Error rate (%)	$L = 8$	$L = 12$	$L = 16$	$L = 24$	$L = 36$	$L = 48$	$L = 52$	$L = 56$	n.v.c.
$M = 5$	46.57	45.86	44.21	44.57	44.57	43.93	43.93	**43.57**	43.79
$M = 10$	42.86	42.71	41.86	40.43	40.00	37.79	37.14	37.50	**35.71**
$M = 20$	42.21	43.14	39.50	38.21	36.86	37.07	36.07	35.57	**34.14**
$M = 30$	43.43	42.14	41.21	39.00	38.50	36.29	**35.79**	36.79	35.86
$M = 40$	43.64	42.00	41.50	38.43	36.29	36.07	33.64	**33.21**	33.29
$M = 50$	42.79	41.07	39.07	37.71	35.93	**33.86**	34.14	35.29	34.57

Computational Efficiency: We hereby report the running time of two-way GMM on a laptop with 2.66 GHz Intel CPU and 4.00 GB RAM. For the microarray data, when $M = 18$ and $L = 70$, it takes about 30 minutes to train the classifier on four fifths of the data and test the classifier on one fifth of the data (that is, to finish computation for one fold in a five-fold cross validation setup). For the text document data (2514 training samples, 2475 test samples, 5 classes, 3455 dimensions), when $M = 20$ and $L = 50$, it takes about

40 minutes to train and test the classifier. For the imagery data, at $M = 30$ and $L = 24$, two-way GMM with diagonal covariance matrices takes only 14 seconds to finish computation for one fold of the five-fold cross validation. The EM algorithm converges fast and the computational cost for each iteration is linear in $npML$. The longer running time required by the microarray as well as the text document data is because of the high dimensions and coding in Matlab. We expect much shorter running time if the experiments are conducted using C/C++. Although the grid search of M and L further increases the computation time, the search can be readily parallelized in a cluster computing environment.

6 Conclusions and Future Work

In this paper, we proposed the two-way Gaussian mixture model for classifying high dimensional data. A dimension reduction property is proved for a two-way mixture of distributions from any exponential family. Experiments conducted on multiple real data sets show that the two-way mixture model often outperforms the mixture without variable clustering. Comparing with SVM with and without variable selection, two-way mixture model achieves close or better performance. Given the importance of QDA as a fundamental classification method, we also investigated QDA with mean regularization by variable grouping, and found that the regularization results in better classification for all the data sets we experimented with.

For data sets arising out of engineering systems, variables, or features, often form natural groups according to their physical meanings. Such prior knowledge may be exploited in the future when we create variable groups in the two-way mixture. Another issue that can be explored is the component-wise whitening strategy we proposed for moderately high dimensional data when diagonal covariance matrices are considered too restrictive. In the current experiments, we did not observe gain from this strategy. It is worthy to study whether the approach can be improved by more robust estimation of covariance and whether new applications may benefit from the approach.

Acknowledgment

The authors would like to thank the reviewers for their valuable comments and suggestions. The work is supported by NSF under the grant CCF 0936948.

Appendix A

We now prove Theorem 3.1. Denote the number of variables in the lth cluster in class k by $\eta_{k,l}$, $\sum_{l=1}^{L} \eta_{k,l} = p$ for all k. Suppose variables in cluster l under class k are $\{j_1^{(k,l)}, j_2^{(k,l)}, ..., j_{\eta_{k,l}}^{(k,l)}\}$. The general two-way mixture model in (6) can also be written as

$$\begin{aligned} f(\mathbf{X} = \mathbf{x}, Y = k) &= \sum_{m=1}^{M} \pi_m p_m(k) \prod_{j=1}^{p} \phi\left(x_j | \boldsymbol{\theta}_{m,c(b(m),j)}\right) \\ &= \sum_{m \in \mathscr{R}_k} \pi_m \prod_{l=1}^{L} \prod_{i=1}^{\eta_{k,l}} \phi\left(x_{j_i^{(k,l)}} | \boldsymbol{\theta}_{m,l}\right) . \end{aligned} \tag{17}$$

Since the distribution of $x_{j_i^{(k,l)}}$ is from the exponential family, we have

$$\begin{aligned}\prod_{i=1}^{\eta_{k,l}} \phi\left(x_{j_i^{(k,l)}}|\boldsymbol{\theta}_{m,l}\right) &= \prod_{i=1}^{\eta_{k,l}} \exp\left(\sum_{s=1}^{S} \eta_s(\boldsymbol{\theta}_{m,l}) T_s(x_{j_i^{(k,l)}}) - \mathbf{B}(\boldsymbol{\theta}_{m,l})\right) h\left(x_{j_i^{(k,l)}}\right) \\ &= \exp\left(\sum_{s=1}^{S} \eta_s(\boldsymbol{\theta}_{m,l}) \sum_{i=1}^{\eta_{k,l}} T_s(x_{j_i^{(k,l)}}) - \eta_{k,l}\mathbf{B}(\boldsymbol{\theta}_{m,l})\right) \prod_{i=1}^{\eta_{k,l}} h\left(x_{j_i^{(k,l)}}\right) \qquad (18)\end{aligned}$$

We have defined $\overline{\mathbf{T}}_{l,k}(\mathbf{x}) = \sum_{i=1}^{\eta_{k,l}} \mathbf{T}(x_{j_i^{(k,l)}})$. More specifically, $\overline{\mathbf{T}}_{l,k}(\mathbf{x}) = (\overline{T}_{l,k,1}(\mathbf{x}), ..., \overline{T}_{l,k,S}(\mathbf{x}))^t$, where $\overline{T}_{l,k,s}(\mathbf{x}) = \sum_{i=1}^{\eta_{k,l}} T_s(x_{j_i^{(k,l)}})$, $s = 1, ..., S$. Substitute (18) into (17),

$$\begin{aligned}& f(\mathbf{X} = \mathbf{x}, Y = k) \\ = & \sum_{m \in \mathscr{R}_k} \pi_m \left[\prod_{l=1}^{L} \exp\left(\sum_{s=1}^{S} \eta_s(\boldsymbol{\theta}_{m,l}) \sum_{i=1}^{\eta_{k,l}} T_s(x_{j_i^{(k,l)}}) - \eta_{k,l}\mathbf{B}(\boldsymbol{\theta}_{m,l})\right)\right] \left[\prod_{l=1}^{L} \prod_{i=1}^{\eta_{k,l}} h(x_{j_i^{(k,l)}})\right] \\ = & \left[\sum_{m \in \mathscr{R}_k} \pi_m \prod_{l=1}^{L} \exp\left(\sum_{s=1}^{S} \eta_s(\boldsymbol{\theta}_{m,l}) \overline{T}_{l,k,s}(\mathbf{x}) - \eta_{k,l}\mathbf{B}(\boldsymbol{\theta}_{m,l})\right)\right] \left[\prod_{j=1}^{p} h(x_j)\right] .\end{aligned}$$

Because $f(Y = k|\mathbf{X} = \mathbf{x}) \propto f(\mathbf{X} = \mathbf{x}, Y = k)$,

$$f(Y = k|\mathbf{X} = \mathbf{x}) \propto \sum_{m \in \mathscr{R}_k} \pi_m \prod_{l=1}^{L} \exp\left(\sum_{s=1}^{S} \eta_s(\boldsymbol{\theta}_{m,l}) \overline{T}_{l,k,s}(\mathbf{x}) - \eta_{k,l} B(\boldsymbol{\theta}_{m,l})\right)$$

subject to $\sum_{k=1}^{K} f(Y = k \mid \mathbf{X} = \mathbf{x}) = 1$. As the posterior probability of Y given $\mathbf{X} = \mathbf{x}$ only depends on $\overline{T}_{l,k,s}(\mathbf{x})$, $\mathbf{X}$ and Y are conditionally independent given $\overline{T}_{l,k,s}(\mathbf{x})$, $l = 1, ..., L$, $k = 1, ..., K$, $s = 1, ..., S$, or equivalently, $\overline{\mathbf{T}}_{l,k}(\mathbf{x})$, $l = 1, ..., L$, $k = 1, ..., K$.

Appendix B

The E-step of EM computes $Q(\psi_{t+1}|\psi_t)$ and the M-step maximizes it. $Q(\psi_{t+1}|\psi_t) = E[\log f(\mathbf{v}|\psi_{t+1}) \mid \mathbf{w}, \psi_t]$, where $\mathbf{v}$ is the complete data, $\mathbf{w}$ the incomplete, and $f(\cdot)$ the density function. Let $\tau^{(i)}$ be the latent component identity of $\mathbf{x}^{(i)}$. We abuse the notation $\Lambda(\mathbf{x}^{(i)})$ slightly to mean the non-missing variables in $\mathbf{x}^{(i)}$. Here $\mathbf{v} = \{\mathbf{x}^{(i)}, y^{(i)}, \tau^{(i)} : i = 1, ..., n\}$, and $\mathbf{w} = \{\Lambda(\mathbf{x}^{(i)}), y^{(i)} : i = 1, ..., n\}$. $Q(\psi_{t+1}|\psi_t) = \sum_{i=1}^{n} E\left[\log f(\mathbf{x}^{(i)}, \tau^{(i)}, y^{(i)}|\psi_{t+1}) \mid \Lambda(\mathbf{x}^{(i)}), y^{(i)}, \psi_t\right]$, where

$$\begin{aligned}& E\left[\log f(\mathbf{x}^{(i)}, \tau^{(i)}, y^{(i)}|\psi_{t+1}) \mid \Lambda(\mathbf{x}^{(i)}), y^{(i)}, \psi_t\right] \\ = & E\left[\log \pi_{\tau^{(i)}}^{(t+1)} \mid \Lambda(\mathbf{x}^{(i)}), y^{(i)}, \psi_t\right] + E\left[\log p_{\tau^{(i)}}(y^{(i)}) \mid \Lambda(\mathbf{x}^{(i)}), y^{(i)}, \psi_t\right] + \\ & \sum_{j=1}^{p} E\left[\log \phi(x_j^{(i)} \mid \mu^{(t+1)}_{\tau^{(i)}, c^{(t+1)}(b(\tau^{(i)}), j)}, \sigma^{2\,(t+1)}_{\tau^{(i)}, c^{(t+1)}(b(\tau^{(i)}), j)}) \mid \Lambda(\mathbf{x}^{(i)}), y^{(i)}, \psi_t\right] \qquad (19)\end{aligned}$$

Let $q_{i,m}$ be the posterior probability for $\Lambda(\mathbf{x}^{(i)})$ being in component m under ψ_t, as given in Eq.(12).

The first term in (19), $E[\log \pi_{\tau^{(i)}}^{(t+1)} \mid \Lambda(\mathbf{x}^{(i)}), y^{(i)}, \psi_t] = \sum_{m=1}^{M} q_{i,m} \log \pi_m^{(t+1)}$. The second term in (19) is zero. For the third term, consider each j separately. If $x_j^{(i)}$ is not missing, that is, $\Lambda(x_j^{(i)}) = 1$,

the distribution of the complete data $\{x_j^{(i)}, \tau^{(i)}, y^{(i)}\}$ conditioned on the incomplete data is random only in terms of $\tau^{(i)} \in \{1, ..., M\}$, which is the pmf given by the posterior probabilities $q_{i,m}$. Thus,

$$\begin{aligned} & E\left[\log \phi(x_j^{(i)} \mid \mu^{(t+1)}_{\tau^{(i)},c^{(t+1)}(b(\tau^{(i)}),j)}, {\sigma^2}^{(t+1)}_{\tau^{(i)},c^{(t+1)}(b(\tau^{(i)}),j)}) \mid \Lambda(\mathbf{x}^{(i)}), y^{(i)}, \psi_t\right] \\ = & \sum_{m=1}^{M} q_{i,m} \cdot \left[\log \frac{1}{\sqrt{2\pi {\sigma^2}^{(t+1)}_{m,c^{(t+1)}(b(m),j)}}} - \frac{\left(x_j^{(i)} - \mu^{(t+1)}_{m,c^{(t+1)}(b(m),j)}\right)^2}{2{\sigma^2}^{(t+1)}_{m,c^{(t+1)}(b(m),j)}}\right] . \end{aligned}$$

If $x_j^{(i)}$ is missing, that is, $\Lambda(x_j^{(i)}) = 0$, the distribution of the complete data $\{x_j^{(i)}, \tau^{(i)}, y^{(i)}\}$ conditioned on the incomplete data is random in terms of both $\tau^{(i)} \in \{1, ..., M\}$ and the variable $x_j^{(i)}$. The conditional distribution of $\tau^{(i)}$ is still given by the posterior probabilities $q_{i,m}$, $m = 1$, ..., M. The conditional distribution of $x_j^{(i)}$ given $\{\Lambda(\mathbf{x}^{(i)}), y^{(i)}, \tau^{(i)} = m\}$ under ψ_t is $\mathcal{N}(\mu^{(t)}_{m,c^{(t)}(b(m),j)}, {\sigma^2}^{(t)}_{m,c^{(t)}(b(m),j)})$. Thus

$$\begin{aligned} & E[\log \phi(x_j^{(i)} \mid \mu^{(t+1)}_{\tau^{(i)},c^{(t+1)}(b(\tau^{(i)}),j)}, {\sigma^2}^{(t+1)}_{\tau^{(i)},c^{(t+1)}(b(\tau^{(i)}),j)} \mid \Lambda(\mathbf{x}^{(i)}), y^{(i)}, \psi_t] \\ = & \sum_{m=1}^{M} q_{i,m} \cdot \left[\log \frac{1}{\sqrt{2\pi {\sigma^2}^{(t+1)}_{m,c^{(t+1)}(b(m),j)}}} - \frac{\left(\mu^{(t)}_{m,c^{(t)}(b(m),j)} - \mu^{(t+1)}_{m,c^{(t+1)}(b(m),j)}\right)^2 + {\sigma^2}^{(t)}_{m,c^{(t)}(b(m),j)}}{2{\sigma^2}^{(t+1)}_{m,c^{(t+1)}(b(m),j)}}\right] . \end{aligned}$$

In summary, $Q(\psi_{t+1} \mid \psi_t)$ is given by the formula below. Let

$$\Delta_1 = \frac{(x_j^{(i)} - \mu^{(t+1)}_{m,c^{(t+1)}(b(m),j)})^2}{2{\sigma^2}^{(t+1)}_{m,c^{(t+1)}(b(m),j)}}, \quad \Delta_2 = \frac{\left(\mu^{(t)}_{m,c^{(t)}(b(m),j)} - \mu^{(t+1)}_{m,c^{(t+1)}(b(m),j)}\right)^2 + {\sigma^2}^{(t)}_{m,c^{(t)}(b(m),j)}}{2{\sigma^2}^{(t+1)}_{m,c^{(t+1)}(b(m),j)}}.$$

Then

$$\begin{aligned} & Q(\psi_{t+1} \mid \psi_t) = \sum_{i=1}^{n}\sum_{m=1}^{M} q_{i,m} \log \pi_m^{(t+1)} + \\ & \sum_{i=1}^{n}\sum_{m=1}^{M}\sum_{j=1}^{p} q_{i,m} \cdot \left[\log \frac{1}{\sqrt{2\pi {\sigma^2}^{(t+1)}_{m,c^{(t+1)}(b(m),j)}}} - \left(\Lambda(x_j^{(i)})\Delta_1 + (1 - \Lambda(x_j^{(i)}))\Delta_2\right)\right] \end{aligned}$$

Based on the obtained $Q(\psi_{t+1} \mid \psi_t)$, the formulas for updating the parameters in Eqs.(13) $\sim$ (16) can be easily derived.

References

[1] T. Hastie and R. Tibshirani, "Discriminant Analysis by Gaussian Mixtures," *Journal of the Royal Statistical Society Series B*, vol. 58, pp. 155-176, 1996.

[2] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Processing*, vol. 10, pp. 19-41, 2000.

[3] Y. Huang, K. B. Englehart, B. Hudgins, and A. D. C. Chan, "A Gaussian mixture model based classification scheme for myoelectric control of powered upper limb prostheses," *IEEE Trans. Biomedical Eng.*, vol. 52, pp. 1801-1811, 2005.

[4] J. Li and H. Zha, "Two-way Poisson mixture models for simultaneous document classification and word clustering," *Computational Statistics and Data Analysis*, vol. 50, pp. 163-180, 2006.

[5] Y. Song, Z. Zhuang, H. Li, Q. Zhao, J. Li, W.-C. Lee, and C. L. Giles, "Real-time automatic tag recommendation," in *Proc. ACM SIGIR*, Singapore, 2008, pp. 515-522.

[6] C. Fraley and A. E. Raftery, "Model-based clustering, discriminant analysis, and density estimation," *Journal of the American Statistical Association*, vol. 97, pp. 611-631, 2002.

[7] B. Schölkopf, C. J. C. Burges, and A. J. Smola, *Advances in kernel methods: support vector learning*, MIT Press, 1999.

[8] G. Celeux and G. Govaert, "A classification EM algorithm for clustering and two stochastic versions," *Computational Statistics and Data Analysis*, vol. 14, pp. 315-332, 1992.

[9] J. D. Banfield and A. E. Raftery, "Model-based Gaussian and non-Gaussian clustering," *Biometrics*, vol. 49, pp. 803-821, 1993.

[10] G. J. McLachlan and D. Peel, *Finite Mixture Models*, New York : Wiley, 2000.

[11] J. Li and J. Z. Wang, "Real-time computerized annotation of pictures," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 30(6), pp. 985-1002, 2008.

[12] J. H. Friedman, "Regularized discriminant analysis," *Journal of the American Statistical Association*, vol. 84(405), pp. 165-175, 1989.

[13] Y. Guo, T. Hastie, and R. Tibshirani, "Regularized linear discriminant analysis and its application in microarrays," *Biostatistics*, vol. 8(1), pp. 86-100, 2006.

[14] R. Tibshirani, T. Hastie, B. Narasimhan, and G. Chu, "Class prediction by nearest shrunken centroids, with applications to DNA microarrays," *Statistical Science*, vol. 18(1), pp. 104-117, 2003.

[15] W. Pan and X. Shen, "Penalized Model-Based Clustering with Application to Variable Selection," *Journal of Machine Learning Research*, vol. 8, pp. 1145-1164, 2007.

[16] S. Wang and J. Zhu, "Variable Selection for Model-Based High-Dimensional Clustering and Its Application to Microarray Data," *Biometrics*, vol. 64, pp. 440-448, 2008.

[17] L. Lazzeroni and A. Owen, "Plaid Models for Gene Expression Data," *Statistica Sinica*, vol. 12(1), pp. 61-86, 2002.

[18] H. Zha, X. He., C. Ding, M. Gu, and H. Simon, "Bipartite graph partitioning and data clustering," in *Proc. ACM CIKM*, 2001, pp. 25-32.

[19] M. Ouyang et al., "Gaussian mixture clustering and imputation of microarray data," *Bioinformatics*, vol. 20, pp. 917-923, 2004.

[20] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society Series B*, vol. 39(1), pp. 1-38, 1977.

[21] A. A. Alizadeh et al., "Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling," *Nature*, vol. 403, pp. 503-511, 2000.

[22] R. Tibshirani and G. Walther, "Cluster validation by prediction strength," *Journal of Computational and Graphical Statistics*, vol. 14(3), pp. 511-528, 2005.

[23] C.C. Chang and C.J. Lin. (2001). LIBSVM : a library for support vector machines. [Online]. Available: http://www.csie.ntu.edu.tw/cjlin/libsvm

[24] K. Lang, "NewsWeeder: Learning to Filter Netnews," in *Proc. 12th Int. Machine Learning Conf.*, 1995, pp. 331-339.

[25] A. McCallum. (1996). Bow: A Toolkit for Statistical Language Modeling, Text Retrieval, Classification and Clustering. [Online]. Available: http://www.cs.cmu.edu/mccallum/bow

[26] R. Kohavi and G. H. John, “Wrappers for feature subset selection,” *Artificial Intelligence*, vol. 97(1-2), pp. 273-324, 1997.

[27] M. Hall et al., “The WEKA Data Mining Software: An Update,” *SIGKDD Explorations*, vol. 11(1), pp. 10-18, 2009.