Simultaneous Feature Selection and Clustering Using Mixture Models

Martin H.C. Law, *Student Member*, *IEEE*, Mário A.T. Figueiredo, *Senior Member*, *IEEE*, and Anil K. Jain, *Fellow*, *IEEE*

Abstract—Clustering is a common unsupervised learning technique used to discover group structure in a set of data. While there exist many algorithms for clustering, the important issue of feature selection, that is, what attributes of the data should be used by the clustering algorithms, is rarely touched upon. Feature selection for clustering is difficult because, unlike in supervised learning, there are no class labels for the data and, thus, no obvious criteria to guide the search. Another important problem in clustering is the determination of the number of clusters, which clearly impacts and is influenced by the feature selection issue. In this paper, we propose the concept of *feature saliency* and introduce an *expectation-maximization* (EM) algorithm to estimate it, in the context of mixture-based clustering. Due to the introduction of a *minimum message length* model selection criterion, the saliency of irrelevant features is driven toward zero, which corresponds to performing feature selection. The criterion and algorithm are then extended to simultaneously estimate the feature saliencies and the number of clusters.

Index Terms—Feature selection, clustering, unsupervised learning, mixture models, minimum message length, EM algorithm.

1 Introduction

The goal of clustering is to discover a "natural" grouping in a set of patterns, points, or objects, without knowledge of any class labels. Clustering, or cluster analysis, is prevalent in any discipline that involves analysis of multivariate data. It is, of course, impractical to exhaustively list the numerous uses of clustering techniques. Image segmentation, an important problem in computer vision, can be formulated as a clustering problem [21], [28], [55]. Documents can be clustered [23] to generate topical hierarchies for information access [53] or retrieval [5]. Clustering is also used to perform market segmentation [2], [11] as well as in biology, e.g., to study genome data [3].

Many clustering algorithms have been proposed in different application scenarios [25], [29]. They can be divided roughly into two categories: *hierarchical clustering*, which creates a "tree" with branches merging at different levels, and *partitional clustering*, which divides the data into different "flat" clusters. The input of clustering algorithms can either be a proximity matrix containing the similarities/dissimilarities between all pairs of points, or a pattern matrix, where each item is described by a vector of attributes, also called *features*. In this paper, we shall focus on partitional clustering with a pattern matrix as input.

In principle, the more information we have about each pattern, the better a clustering algorithm is expected to perform. This seems to suggest that we should use as many features as possible to represent the patterns. However, this is not the case in practice. Some features can be just "noise," thus not contributing to (or even degrading) the clustering process. The task of selecting the "best" feature subset is known as *feature selection*, sometimes as *variable selection* or *subset selection*.

Feature selection is important for several reasons, the fundamental one being arguably that noisy features can degrade the performance of most learning algorithms (see the example in Fig. 1). In supervised learning, it is known that feature selection can improve the performance of classifiers learned from limited amounts of data [49]; it leads to more economical (both in storage and computation) classifiers and, in many cases, it may lead to interpretable models. Feature selection is particularly important for data sets with large numbers of features, e.g., classification problems in molecular biology may involve thousands of features [3], [62], and a Web page can be represented by thousands of different key-terms [58]. Appearance-based image classification methods may use each pixel as a feature [6], thus easily involving thousands of features.

Feature selection has been widely studied in the context of supervised learning (see [7], [24], [33], [34] and references therein), where the ultimate goal is to select features that can achieve the highest accuracy on unseen data. Feature selection has received comparatively very little attention in unsupervised learning or clustering. One important reason is that it is not at all clear how to assess the relevance of a subset of features without resorting to class labels. The problem is made even more challenging when the number of clusters is unknown, since the optimal number of clusters and the optimal feature subset are interrelated, as illustrated in Fig. 2 (taken from [16]). Note that methods based on variance (such as *principal components analysis*) need not select good features for clustering, as features with large variance can be independent of the intrinsic grouping of the data (see example in Fig. 3).

Most feature selection algorithms (such as [9], [33], [47]) involve a combinatorial search through the space of all feature subsets. Usually, heuristic (nonexhaustive) methods have to be adopted, because the size of this space is

- *M.H.C. Law and A.K. Jain are with the Department of Computer Science and Engineering, Michigan State University, 3115 Engineering Building, East Lansing, Michigan 48824-1226. E-mail: {lawhiu, jain}@cse.msu.edu.*
- *M.A.T. Figueiredo is with the Instituto de Telecomunicações, Instituto Superior Técnico, Torre Norte, Piso 10, Av. Rovisco Pais, 1049-001 Lisboa, Portugal. E-mail: mtf@lx.it.pt.*

Manuscript received 15 May 2003; accepted 27 Feb. 2004.
Recommended for acceptance by B.J. Frey.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0077-0503.

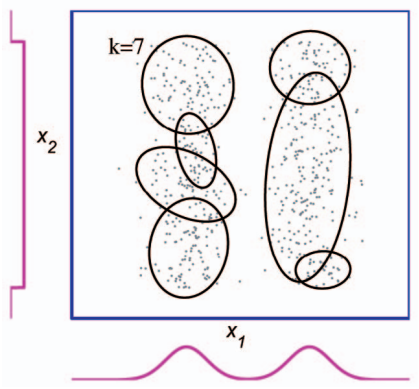


Fig. 1. A uniformly distributed irrelevant feature (x_2) makes it difficult for the Gaussian mixture learning algorithm in [18] to recover the two underlying clusters. If only feature x_1 is used, however, the two clusters are easily identified. The curves along the horizontal and vertical axes of the figure indicate the marginal distribution of x_1 and x_2, respectively.

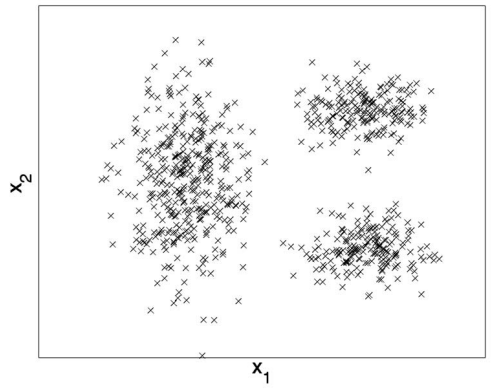


Fig. 2. Number of clusters is interrelated with feature subset used. The optimal feature subsets for identifying three, two, one clusters in this data set are $\{x_1, x_2\}$, $\{x_1\}$, and $\{x_2\}$, respectively. On the other hand, the optimal number of clusters for feature subsets $\{x_1, x_2\}$, $\{x_1\}$, and $\{x_2\}$ are also three, two, one, respectively.

exponential in the number of features. In this case, one generally loses any guarantee of optimality of the selected feature subset.

In this paper, we propose a solution to the feature selection problem in unsupervised learning by casting it as an estimation problem, thus avoiding any combinatorial search. Instead of selecting a subset of features, we estimate a set of real-valued (actually in $[0, 1]$) quantities (one for each feature) which we call the *feature saliencies*. This estimation is carried out by an EM algorithm derived for the task. Since we are in the presence of a model-selection-type problem, it is necessary to avoid the situation where all the saliencies take the maximum possible value. This is achieved by adopting a *minimum message length* (MML, [60], [61]) penalty, as was done in [18] to select the number of clusters. The MML criterion encourages the saliencies of the irrelevant features to go to zero, allowing us to prune the feature set. Finally, we integrate the process of feature saliency estimation into the algorithm proposed in [18], thus obtaining a method which is able to simultaneously perform feature selection and determine the number of clusters. Although the algorithm is presented with respect to Gaussian mixture-based clustering, one can extend it to other types of model-based clustering as well. The algorithm first appears in [38].

The remainder of this paper is organized as follows: In Section 2, we review approaches for feature selection and

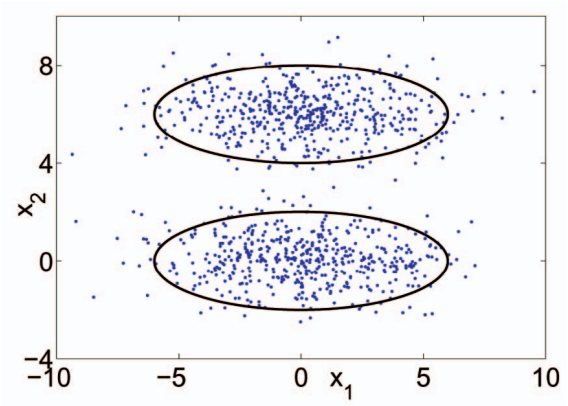


Fig. 3. Feature x_1, although explaining more data variance than feature x_2, is spurious for the identification of the two clusters in this data set.

previous attempts to solve the feature selection problem in unsupervised learning. The details of our approach are presented in Section 3. Experimental results are reported in Section 4, followed by comments on the proposed algorithm in Section 5. Finally, we conclude in Section 6 and outline some future work directions.

2 Related Work

Most of the literature on feature selection pertains to supervised learning (both classification [24] and regression [40]). Feature selection algorithms can be broadly divided into two categories [7], [33]: *filters* and *wrappers*. The filter approaches evaluate the relevance of each feature (subset) using the data set alone, regardless of the subsequent learning algorithm. RELIEF [32] and its enhancement [36] are representatives of this class, where the basic idea is to assign feature weights based on the consistency of the feature value in the k nearest neighbors of every data point. Information-theoretic methods are also used to evaluate features: the mutual information between a relevant feature and the class labels should be high [4]. Nonparametric methods can be used to compute mutual information involving continuous features [37]. A feature can be regarded as irrelevant if it is conditionally independent of the class labels given other features. The concept of Markov blanket is used to formalize this notion of irrelevancy in [34].

On the other hand, wrapper approaches [33] invoke the learning algorithm to evaluate the quality of each feature (subset). Specifically, a learning algorithm (e.g., a nearest neighbor classifier, a decision tree, a naive Bayes method) is run on a feature subset and the feature subset is assessed by some estimate of the classification accuracy. Wrappers are usually more computationally demanding, but they can be superior in accuracy when compared with filters, which ignore the properties of the learning task at hand [33].

Both approaches, filters and wrappers, usually involve combinatorial searches through the space of possible feature subsets; for this task, different types of heuristics, such as sequential forward or backward searches, floating search, beam search, bidirectional search, and genetic search have been suggested [9], [33], [47], [63]. It is also possible to construct a set of *weak* (in the boosting sense [20]) classifiers, with each one using only one feature, and then apply boosting, which effectively performs feature

selection [59]. It has also been proposed to approach feature selection using rough set theory [35].

All of the approaches mentioned above are concerned with feature selection in the presence of class labels. Comparatively, not much work has been done for feature selection in unsupervised learning. Of course, any method conceived for supervised learning that does not use the class labels could be used for unsupervised learning; it is the case for methods that measure feature similarity to detect redundant features, using, e.g., mutual information [53] or a maximum information compression index [42]. In [16], [17], the normalized log-likelihood and cluster separability are used to evaluate the quality of clusters obtained with different feature subsets. Different feature subsets and numbers of clusters, for multinomial model-based clustering, are evaluated using marginal likelihood and cross-validated likelihood in [58]. The algorithm described in [52] uses automatic relevance determination priors to select features when there are two clusters. In [13], the clustering tendency of each feature is assessed by an entropy index. A genetic algorithm is used in [31] for feature selection in k-means clustering. In [56], feature selection for symbolic data is addressed by assuming that irrelevant features are uncorrelated with the relevant features. Reference [14] describes the notion of "category utility" for feature selection in a conceptual clustering task. The CLIQUE algorithm [1] is popular in the data mining community and it finds hyperrectangular shaped clusters using a subset of attributes for a large database. The wrapper approach can also be adopted to select features for clustering; this has been explored in our earlier work [19], [38].

All the methods referred above perform "hard" feature selection (a feature is either selected or not). There are also algorithms that assign weights to different features to indicate their significance. In [43], weights are assigned to different groups of features for k-means clustering based on a score related to the Fisher discriminant. Feature weighting for k-means clustering is also considered in [41], but the goal there is to find the best description of the clusters after they are identified. The method described in [46] can be classified as learning feature weights for conditional Gaussian networks. An EM algorithm based on Bayesian shrinking is proposed in [22] for unsupervised learning.

3 EM Algorithm for Feature Saliency

In this section, we propose an EM algorithm for performing mixture-based (or model-based) clustering with feature selection. In mixture-based clustering, each data point is modeled as having been generated by one of a set of probabilistic models [25], [39]. Clustering is then done by learning the parameters of these models and the associated probabilities. Each pattern is assigned to the mixture component that most likely generated it. Although the derivations below refer to Gaussian mixtures, they can be generalized to other types of mixtures.

3.1 Mixture Densities

A finite mixture density with K components is defined by

$$p(\mathbf{y}) = \sum_{j=1}^{K} \alpha_j \, p(\mathbf{y}|\theta_j), \tag{1}$$

where $\forall_j, \alpha_j \geq 0; \sum_j \alpha_j = 1$; each θ_j is the set of parameters of the jth component (all components are assumed to have the same form, e.g., Gaussian); and $\theta \equiv \{\theta_1, \ldots, \theta_K, \alpha_1, \ldots, \alpha_K\}$ will denote the full parameter set. The goal of mixture estimation is to infer θ from a set of N data points $\mathcal{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$, assumed to be samples of a distribution with density given by (1). Each $\mathbf{y}_i$ is a D-dimensional feature vector $[y_{i1}, \ldots, y_{iD}]^T$. In the sequel, we will use the indices i, j, and l to run through data points (1 to N), mixture components (1 to K), and features (1 to D), respectively.

As is well-known, neither the *maximum likelihood* (ML) estimate,

$$\widehat{\theta}_{\mathrm{ML}} = \arg\max_{\theta}\{\log p(\mathcal{Y}|\theta)\},$$

nor the *maximum a posteriori* (MAP) estimate (given some prior $p(\theta)$)

$$\widehat{\theta}_{\mathrm{MAP}} = \arg\max_{\theta}\{\log p(\mathcal{Y}|\theta) + \log p(\theta)\},$$

can be found analytically. The usual choice is the EM algorithm, which finds local maxima of these criteria [39]. This algorithm is based on a set $\mathcal{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$ of N *missing* (latent) labels, where $\mathbf{z}_i = [z_{i1}, \ldots, z_{iK}]$, with $z_{ij} = 1$ and $z_{ip} = 0$, for $p \neq j$, meaning that $\mathbf{y}_i$ is a sample of $p(\cdot|\theta_j)$. For brevity of notation, sometimes we write $z_i = j$ for such $\mathbf{z}_i$. The complete data log-likelihood, i.e., the log-likelihood if $\mathcal{Z}$ was observed, is

$$\log p(\mathcal{Y}, \mathcal{Z}|\theta) = \sum_{i=1}^{N}\sum_{j=1}^{K} z_{ij} \log\left[\alpha_j p(\mathbf{y}_i|\theta_j)\right]. \tag{2}$$

The EM algorithm produces a sequence of estimates $\{\widehat{\theta}(t),\ t = 0, 1, 2, \ldots\}$ using two alternating steps:

- **E-step:** Compute $\mathcal{W} = E[\mathcal{Z}|\mathcal{Y}, \widehat{\theta}(t)]$, the expected value of the missing data given the current parameter estimate, and plug it into $\log\ p(\mathcal{Y}, \mathcal{Z}|\theta)$, yielding the so-called Q-function $Q(\theta, \widehat{\theta}(t)) = \log p(\mathcal{Y}, \mathcal{W}|\theta)$. Since the elements of $\mathcal{Z}$ are binary, we have

$$\begin{aligned} w_{i,j} &\equiv E\Big[z_{ij}|\,\mathcal{Y}, \widehat{\theta}(t)\Big] = \Pr\Big[z_{ij} = 1|\,\mathbf{y}_i, \widehat{\theta}(t)\Big] \\ &= \frac{\widehat{\alpha}_j(t)\ p(\mathbf{y}_i|\widehat{\theta}_j(t))}{\sum_{k=1}^{K} \widehat{\alpha}_k(t)\ p(\mathbf{y}_i|\widehat{\theta}_k(t))}. \end{aligned} \tag{3}$$

  Notice that α_j is the a priori probability that $z_{ij} = 1$ (i.e., that $\mathbf{y}_i$ belongs to cluster j), while w_{ij} is the corresponding a posteriori probability, after observing $\mathbf{y}_i$.
- **M-step:** Update the parameter estimates,

$$\widehat{\theta}(t+1) = \arg\max_{\theta}\ \{Q(\theta, \widehat{\theta}(t)) + \log p(\theta)\},$$

  in the case of MAP estimation, or without $\log p(\theta)$ in the ML case.

3.2 Feature Saliency

In this section, we define the concept of *feature saliency* and derive an EM algorithm to estimate its value. We assume that the features are conditionally independent given the (hidden) component label, that is,

Fig. 4. A graphical model for the probability model in (5) for the case of four features ($D = 4$) with different indicator variables. $\phi_l = 1$ corresponds to the existence of an arc from z to y_l, and $\phi_l = 0$ corresponds to its absence. (a) $\phi_1 = 1$, $\phi_2 = 1$, $\phi_3 = 0$, $\phi_4 = 1$. (b) $\phi_1 = 0$, $\phi_2 = 1$, $\phi_3 = 1$, $\phi_4 = 0$.

$$p(\mathbf{y}|\theta) = \sum_{j=1}^{K} \alpha_j \, p(\mathbf{y}|\theta_j) = \sum_{j=1}^{K} \alpha_j \prod_{l=1}^{D} p(y_l|\theta_{jl}), \tag{4}$$

where $p(\cdot|\theta_{jl})$ is the pdf of the lth feature in the jth component. This assumption enables us to utilize the power of the EM algorithm. In the particular case of Gaussian mixtures, the conditional independence assumption is equivalent to adopting diagonal covariance matrices, which is a common choice for high-dimensional data, such as in naïve Bayes classifiers, latent class models, as well as in the emission densities of continuous hidden Markov models.

Among different definitions of feature irrelevancy (proposed for supervised learning), we adopt the one suggested in [48], [58], which is suitable for unsupervised learning: the lth feature is irrelevant if its distribution is independent of the class labels, i.e., if it follows a common density, denoted by $q(y_l|\lambda_l)$. Let $\Phi = (\phi_1, \ldots, \phi_D)$ be a set of binary parameters, such that $\phi_l = 1$ if feature l is relevant and $\phi_l = 0$, otherwise. The mixture density in (4) can then be rewritten as

$$p(\mathbf{y}|\Phi, \{\alpha_j\}, \{\theta_{jl}\}, \{\lambda_l\}) = \sum_{j=1}^{K} \alpha_j \prod_{l=1}^{D} [p(y_l|\theta_{jl})]^{\phi_l} [q(y_l|\lambda_l)]^{1-\phi_l}. \tag{5}$$

A related model for feature selection in supervised learning has been considered in [44], [48]. Intuitively, Φ determines which edges exist between the hidden label z and the individual features y_l in the graphical model illustrated in Fig. 4, for the case $D = 4$.

Our notion of *feature saliency* is summarized in the following steps: 1) We treat the ϕ_ls as missing variables and 2) we define the *feature saliency* as $\rho_l = P(\phi_l = 1)$, the probability that the lth feature is relevant. This definition makes sense, as it is difficult to know for sure that a certain feature is irrelevant in unsupervised learning. The resulting model (likelihood function) is written as (see the proof in Appendix A)

$$p(\mathbf{y}|\theta) = \sum_{j=1}^{K} \alpha_j \prod_{l=1}^{D} (\rho_l p(y_l|\theta_{jl}) + (1 - \rho_l) q(y_l|\lambda_l)), \tag{6}$$

where $\theta = \{\{\alpha_j\}, \{\theta_{jl}\}, \{\lambda_l\}, \{\rho_l\}\}$ is the set of all the parameters of the model. An intuitive way to see how (6) is obtained is to notice that $[p(y_l|\theta_{jl})]^{\phi_l}[q(y_l|\lambda_l)]^{1-\phi_l}$ can be written as $\phi_l \, p(y_l|\theta_{jl}) + (1 - \phi_l) \, q(y_l|\lambda_l)$, because ϕ_l is binary. The form of $q(.|.)$ reflects our prior knowledge about the distribution of the nonsalient features. In principle, it can be any 1D distribution (e.g., a Gaussian, a student-t, or even a mixture). We shall limit $q(.|.)$ to be a Gaussian, since this leads to reasonable results in practice.

Equation (6) has a generative interpretation. As in a standard finite mixture, we first select the component label j by sampling from a multinomial distribution with parameters $(\alpha_1, \ldots, \alpha_K)$. Then, for each feature $l = 1, \ldots, D$, we flip a biased coin whose probability of getting a head is ρ_l; if we get a head, we use the mixture component $p(.|\theta_{jl})$ to generate the lth feature; otherwise, the common component $q(.|\lambda_l)$ is used. A graphical model representation of (6) is shown in Fig. 5 for the case $D = 4$.

3.2.1 EM Algorithm

By treating $\mathcal{Z}$ (the hidden class labels) and Φ as hidden variables, one can derive (see details in Appendix B) the following EM algorithm for parameter estimation:

- **E-step:** Compute the following quantities:

$$a_{ijl} = P(\phi_l = 1, y_{il}|z_i = j) = \rho_l \, p(y_{il}|\theta_{jl}), \tag{7}$$

$$b_{ijl} = P(\phi_l = 0, y_{il}|z_i = j) = (1 - \rho_l) \, q(y_{il}|\lambda_l), \tag{8}$$

$$c_{ijl} = P(y_{il}|z_i = j) = a_{ijl} + b_{ijl}, \tag{9}$$

$$w_{ij} = P(z_i = j|\mathbf{y}_i) = \frac{\alpha_j \prod_l c_{ijl}}{\sum_j \alpha_j \prod_l c_{ijl}}, \tag{10}$$

$$u_{ijl} = P(\phi_l = 1, z_i = j|\mathbf{y}_i) = \frac{a_{ijl}}{c_{ijl}} w_{ij}, \tag{11}$$

$$v_{ijl} = P(\phi_l = 0, z_i = j|\mathbf{y}_i) = w_{ij} - u_{ijl}. \tag{12}$$

- **M-step:** Reestimate the parameters according to following expressions:

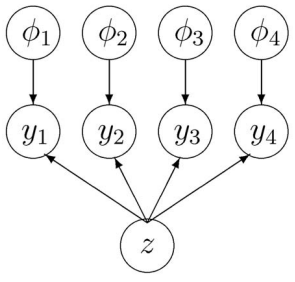


Fig. 5. A graphical model showing the mixture density in (6). The variables $z, \phi_1, \phi_2, \phi_3, \phi_4$ are "hidden" and only y_1, y_2, y_3, y_4 are observed.

Input: Training data $\mathcal{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$, minimum number of components K_{min}
Output: Number of components K, mixture parameters $\{\theta_{jl}\}$, $\{\alpha_j\}$, parameters of common distribution $\{\lambda_l\}$ and feature saliencies $\{\rho_l\}$
  Initialization: Set the parameters of a large number of mixture components randomly
    Set the common distribution to cover all data
    Set the feature saliency of all features to 0.5
  while $K > K_{min}$ **do**
    while not reach local minimum **do**
      Perform E-step according to Eqs. (7) to (12)
      Perform M-step according to Eqs. (14) to (17), (21) and (22)
      If α_j becomes zero, the j-th component is pruned.
      If ρ_l becomes 1, $q(y_l|\lambda_l)$ is pruned. If ρ_l becomes 0, $p(y_l|\theta_{jl})$ are pruned for all j
    end while
    Record the current model parameters and its message length
    Remove the component with the smallest weight
  end while
  Return the model parameters that yield the smallest message length

Fig. 6. The unsupervised feature saliency algorithm.

$$\widehat{\alpha_j} = \frac{\sum_i w_{ij}}{\sum_{ij} w_{ij}} = \frac{\sum_i w_{ij}}{n}, \tag{13}$$

$$\widehat{\text{Mean in}}\ \theta_{jl} = \frac{\sum_i u_{ijl}\, y_{il}}{\sum_i u_{ijl}}, \tag{14}$$

$$\widehat{\text{Var in}}\ \theta_{jl} = \frac{\sum_i u_{ijl}\, (y_{il} - (\widehat{\text{Mean in}}\ \theta_{jl}))^2}{\sum_i u_{ijl}}, \tag{15}$$

$$\widehat{\text{Mean in}}\ \lambda_l = \frac{\sum_i (\sum_j v_{ijl})\, y_{il}}{\sum_{ij} v_{ijl}}, \tag{16}$$

$$\widehat{\text{Var in}}\ \lambda_l = \frac{\sum_i (\sum_j v_{ijl})(y_{il} - (\widehat{\text{Mean in}}\ \lambda_l))^2}{\sum_{ij} v_{ijl}}, \tag{17}$$

$$\widehat{\rho_l} = \frac{\sum_{i,j} u_{ijl}}{\sum_{i,j} u_{ijl} + \sum_{i,j} v_{ijl}} = \frac{\sum_{i,j} u_{ijl}}{n}. \tag{18}$$

In these equations, the variable u_{ijl} measures how important the ith pattern is to the jth component, when the lth feature is used. It is thus natural that the estimates of the mean and the variance in θ_{jl} are weighted sums with weight u_{ijl}. Similar relationship exists between $\sum_j v_{ijl}$ and λ_l. The term $\sum_{ij} u_{ijl}$ can be interpreted as how likely it is that ϕ_l equals one, explaining why the estimate of ρ_l is proportional to $\sum_{ij} u_{ijl}$.

3.3 Model Selection

Standard EM for mixtures exhibits some weaknesses which also affect the EM algorithm introduced above: it requires knowledge of K, and a good initialization is essential for reaching a good local optimum. To overcome these difficulties, we adopt the approach in [18], which is based on the MML criterion [61], [60].

The MML criterion for our model (see details in Appendix C) consists of minimizing, with respect to θ, the following cost function (after discarding the order one term)

$$\begin{aligned} -\log p(\mathcal{Y}|\theta) + \frac{K+D}{2}\log N + \frac{R}{2}\sum_{l=1}^{D}\sum_{j=1}^{K}\log(N\alpha_j\rho_l) \\ + \frac{S}{2}\sum_{l=1}^{D}\log(N(1-\rho_l)), \end{aligned} \tag{19}$$

where R and S are the number of parameters in θ_{jl} and θ_l, respectively. If $p(.|.)$ and $q(.|.)$ are univariate Gaussians (arbitrary mean and variance), $R = S = 2$. From a parameter estimation viewpoint, (19) is equivalent to a *maximum a posteriori* (MAP) estimate,

$$\begin{aligned} \hat{\theta} = \arg\max_{\theta} \Bigg\{ \log p(\mathcal{Y}|\theta) - \frac{RD}{2}\sum_{l=1}^{K}\log\alpha_j - \frac{S}{2}\sum_{l=1}^{D}\log(1-\rho_l) \\ - \frac{RK}{2}\sum_{l=1}^{D}\log\rho_l \Bigg\}, \end{aligned} \tag{20}$$

with the following (Dirichlet-type, but improper) priors on the α_js and ρ_ls:

$$p(\alpha_1, \ldots, \alpha_K) \propto \prod_{j=1}^{K} \alpha_j^{-RD/2},$$

$$p(\rho_1, \ldots, \rho_D) \propto \prod_{l=1}^{D} \rho_l^{-RK/2}(1-\rho_l)^{-S/2}.$$

Since these priors are conjugate with respect to the complete data likelihood, the EM algorithm undergoes a minor modification: The M-step (13) and (18) are replaced by

$$\widehat{\alpha_j} = \frac{\max(\sum_i w_{ij} - \frac{RD}{2}, 0)}{\sum_j \max(\sum_i w_{ij} - \frac{RD}{2}, 0)} \tag{21}$$

$$\widehat{\rho_l} = \frac{\max(\sum_{i,j} u_{ijl} - \frac{KR}{2}, 0)}{\max(\sum_{i,j} u_{ijl} - \frac{KR}{2}, 0) + \max(\sum_{i,j} v_{ijl} - \frac{S}{2}, 0)}. \tag{22}$$

In addition to the log-likelihood, the other terms in (19) have simple interpretations. The term $\frac{K+D}{2}\log N$ is a standard MDL-type [50] parameter code-length corresponding to K α_j values and D ρ_l values. For the lth feature in the jth component, the "effective" number of data points for estimating θ_{jl} is $N\alpha_j\rho_l$. Since there are R parameters in each θ_{jl}, the corresponding code-length is $\frac{R}{2}\log(N\rho_l\alpha_j)$. Similarly, for the lth feature in the common component, the number of

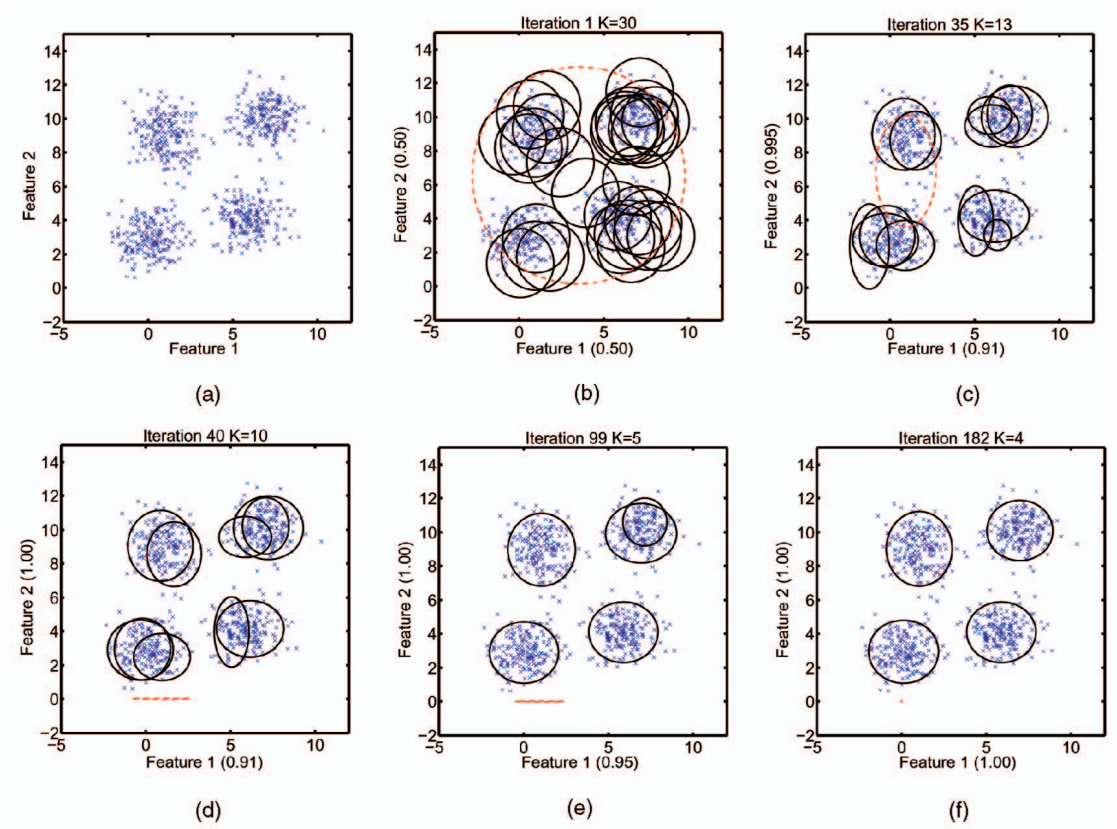


Fig. 7. The solid ellipses represent the Gaussian mixture components; the dotted ellipse represents the common density. The number in parenthesis along the axis label is the feature saliency; when it reaches 1, the common component is no longer applicable to that feature. Thus, in (d), the common component degenerates to a line; when the feature saliency for feature 1 also becomes 1, as in (f), the common density degenerates to a point at $(0, 0)$. (a) The data set, (b) initialization, (c) a snapshot, (d) ρ_2 is "pruned" to 1, (e) a local minimum ($K = 5$), and (f) the best local minimum.

effective data points for estimation is $N(1 - \rho_l)$. Thus, there is a term $\frac{S}{2}\log(N(1 - \rho_l))$ in (19) for each feature.

One key property of (21) and (22) is their pruning behavior, forcing some of the α_j to go to zero and some of the ρ_l to go to zero or one. This pruning behavior also has the indirect benefit of protecting us from almost singular covariance matrices: the weight of such component is usually very small, and it is likely to be pruned in the next few iterations. Concerns that the message length in (19) may become invalid at these boundary values can be circumvented by the arguments in [18]: When ρ_l goes to zero, the lth feature is no longer salient and ρ_l and $\theta_{1l}, \ldots, \theta_{Kl}$ are removed; when ρ_l goes to 1, θ_l and ρ_l are dropped.

Finally, since the model selection algorithm determines the number of components, it can be initialized with a large value of K, thus alleviating the need for a good initialization, as shown in [18]. Because of this, a component-wise version of EM can be adopted [10], [18]. The algorithm is summarized in Fig. 6.

3.4 Postprocessing of Feature Saliency

The feature saliencies generated by the algorithm in Fig. 6 attempt to find the best way to *model* the data, using different component densities. Alternatively, we can consider feature saliencies that best *discriminate* between different components. This can be more appropriate if the ultimate goal is to discover well-separated clusters. If the components are well-separated, each pattern is likely to be generated by one component only. Therefore, one quantitative measure of the separability of the clusters is

$$J = \sum_{i=1}^{N} \log P(z_i = t_i | \mathbf{y}_i), \tag{23}$$

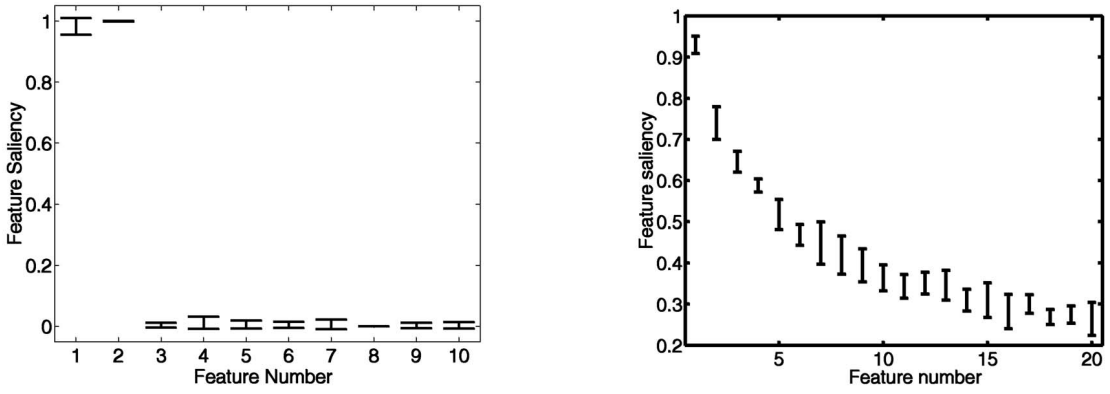
where $t_i = \arg\max_j P(z_i = j|\mathbf{y}_i)$. Intuitively, J is the sum of the logarithms of the posterior probabilities of the data, assuming that each data point was indeed generated by the component with maximum posterior probability (an implicit assumption in mixture-based clustering). J can then be maximized by varying ρ_l while keeping the other parameters fixed.

Unlike the MML criterion, J cannot be optimized by an EM algorithm. However, defining

$$h_{ilj} = \frac{p(y_{il}|\theta_{jl}) - q(y_l|\lambda_l)}{\rho_l p(y_{il}|\theta_{jl}) + (1 - \rho_l) q(y_{il}|\lambda_l)},$$
$$g_{il} = \sum_{j=1}^{K} w_{ij} h_{ilj},$$

it is easy to show that

$$\frac{\partial}{\partial \rho_l} \log w_{ij} = h_{ilj} - g_{il},$$
$$\frac{\partial^2}{\partial \rho_l \partial \rho_m} \log w_{ij} = \sum_{i=1}^{N} (g_{il} g_{im} - \sum_{j=1}^{K} w_{ij} h_{ilj} h_{imj}), \quad \text{for } l \neq m,$$
$$\frac{\partial^2}{\partial \rho_l^2} \log w_{ij} = \sum_{i=1}^{n} (g_{il}^2 - h_{ilj}^2).$$



(a) (b)

Fig. 8. Feature saliencies for (a) the 10-D 4 Gaussian data set used in Fig. 7a, and (b) the Trunk data set. The mean values plus and minus one standard deviation over ten runs are shown. Recall that features 3 to 10 for the 4 Gaussian data set are the noisy features. (a) Features saliencies: 4 Gaussian and (b) features saliencies: trunk.

TABLE 1
Real World Data Sets

Name	Full name	N	D	c	Normalized?
`wine`	wine recognition	178	13	3	yes
`wdbc`	Wisconsin diagnostic breast cancer	569	30	2	yes
`image`	image segmentation	2320	18	7	yes
`texture`	Texture data set	4000	19	4	no
`zernike`	Zernike moments of digit images	2000	47	10	yes

Each data set has N data points with D features from c classes. The feature with a constant value in `image` is discarded.

The gradient and Hessian of J can then be calculated accordingly, if we ignore the dependence of t_i on ρ_l. We can then use any constrained nonlinear optimization software to find the optimal values of ρ_l in $[0, 1]$. We have used the MATLAB optimization toolbox in our experiments. After obtaining the set of optimized ρ_l, we fix them and estimate the remaining parameters using the EM algorithm.

4 Experimental Results

4.1 Synthetic Data

The first synthetic data set consists of 800 data points from a mixture of four equiprobable Gaussians $\mathcal{N}(\mathbf{m}_i, \mathbf{I}), i = \{1, 2, 3, 4\}$, where $\mathbf{m}_1 = \binom{0}{3}$, $\mathbf{m}_2 = \binom{1}{9}$, $\mathbf{m}_3 = \binom{6}{4}$, $\mathbf{m}_4 = \binom{7}{10}$ (Fig. 7a). Eight "noisy" features (sampled from a $\mathcal{N}(0, 1)$ density) are then appended to this data, yielding a set of 800 10-dimensional patterns. We ran the proposed algorithm 10 times, each initialized with $K = 30$; the common component is initialized to cover the entire set of data, and the feature saliency values are initialized at 0.5. The stopping threshold is 10^{-7}. A typical run of the algorithm is shown in Fig. 7. In all the 10 runs with this mixture, the four components were always correctly identified. The saliencies of all the 10 features, together with their standard deviations (error bars), are shown in Fig. 8a. We can conclude that, in this case, the algorithm successfully locates the true clusters and correctly assigns the feature saliencies.

In the second experiment, we consider the Trunk data [24], [57]: two 20-dimensional Gaussians $\mathcal{N}(\mathbf{m}_1, \mathbf{I})$ and $\mathcal{N}(\mathbf{m}_2, \mathbf{I})$, where $\mathbf{m}_1 = (1, \frac{1}{\sqrt{2}}, \ldots, \frac{1}{\sqrt{20}})$, $\mathbf{m}_2 = -\mathbf{m}_1$. Data are obtained by sampling 5,000 points from each of these two Gaussians. Note that the features are arranged in descending order of relevance. As above, the stopping threshold is set to 10^{-7} and the initial values of K to 30. In all the 10 runs performed, the two components are always detected. The feature saliencies are shown in Fig. 8b. The lower the rank number, the more important is the feature. We can see the general trend that as the feature number increases, the saliency decreases, in accordance with the true characteristics of the data.

4.2 Real Data

We tested our algorithm on several data sets with different characteristics (Table 1). The wine recognition data set (`wine`) contains results of chemical analysis of wines grown in different cultivars. The goal is to predict the type of a wine based on its chemical composition; it has 178 data points, 13 features, and three classes. The Wisconsin diagnostic breast cancer data set (`wdbc`) was used to obtain a binary diagnosis (benign or malignant) based on 30 features extracted from cell nuclei presented in an image; it has 576 data points. The image segmentation data set (`image`) contains 2,320 data points with 19 features from seven classes; each pattern consists of features extracted from a 3×3 region taken from seven types of outdoor images: brickface, sky, foliage, cement, window, path, and grass. The texture data set (`texture`) consists of 4,000 19-dimensional Gabor filter features from a collage of four Brodatz textures [27]. A data set (`zer`) of 47 Zernike moments extracted from images of handwriting numerals (as in [26]) are also used; there are 200 images for each digit,

TABLE 2
The Result of the Proposed Algorithm over 20 Random Runs

	Algorithm in Fig. 6		After postprocessing	Using all the features	
	error (in %)	$\widehat{c}$	error (in %)	error (in %)	$\widehat{c}$
wine	6.61 (3.91)	3.1 (0.31)	6.61 (3.23)	8.06 (3.73)	3 (0)
wdbc	9.55 (1.99)	5.65 (0.75)	9.35 (2.07)	10.09 (2.00)	2.70 (0.57)
image	20.19 (1.54)	23.1 (1.74)	20.28 (1.60)	32.84 (5.1)	13.8 (1.94)
texture	4.04 (0.76)	36.17 (1.19)	4.02 (0.74)	4.85 (0.98)	31.42 (2.81)
zernike	52.09 (2.52)	11.3 (0.98)	51.99 (2.32)	56.42 (3.62)	10 (0)

"Error" corresponds to the mean of the error rates on the testing set when the clustering results are compared with the ground truth labels. $\widehat{c}$ denotes the number of Gaussian components estimated. Note that postprocessing does not change the number of Gaussian components. The numbers in parenthesis are the standard deviation of the corresponding quantities.

totaling 2,000 patterns. The data sets wine, wdbc, image, and zernike are from the UCI machine learning repository (http://www.ics.uci.edu/~mlearn/MLSummary.html). Normalization to zero mean and unit variance is performed for all but the texture data set, so as to make the contribution of different features roughly equal a priori. Since these data sets were collected for supervised classification, the class labels are not involved in our experiment, except for evaluation of the clustering results.

Each data set was first randomly divided into two halves: one for training, another for testing. The algorithm in Fig. 6 was run on the training set. The feature saliency values can be post-processed as described in Section 3.4. We evaluate the results by interpreting the components as clusters and compare them with the ground truth labels. Each data point in the test set is assigned to the component that most likely generated it, and the pattern is classified to the class represented by the component. We can then compute the error rates on the test data. For comparison, we also run the mixture of Gaussian algorithm in [18] using all the features, with the number of classes of the data set as a lower bound on the number of components. This gives us a fair ground for comparing Gaussian mixtures with and without feature saliency. In order to ensure that we have enough data with respect to the number of features for the algorithm in [18], the covariance matrices of the mixture components are restricted to be diagonal, but are different for different components. The entire procedure is repeated 20 times and the results are shown in Table 2. We also show the feature saliency values of different features in different runs as gray-level image maps in Fig. 9.

From Table 2, we can see that the proposed algorithm reduces the error rates when compared with using all the features for all five data sets. The improvement is more significant for the image data set, but this may be due to the increased number of components estimated. The high error rate for zernike is due to the fact that digit images are inherently more difficult to cluster: for example, "4" can be written in a manner very similar to "9" and it is difficult for any unsupervised learning algorithm to distinguish among them. The postprocessing can increase the "contrast" of the feature saliencies, as the image maps in Fig. 9 show, without deteriorating the accuracy. It is easier to perform "hard" feature selection using these postprocessed feature saliencies, if this is a must for the application.

5 Discussion

5.1 Complexity

The major computational load in the proposed algorithm is in the E-step and the M-step. Each E-step iteration computes $O(NDK)$ quantities. As each quantity can be computed in constant time, the time complexity for E-step is also $O(NDK)$. Similarly, the M-step takes $O(NDK)$ time. The total amount of computation depends on the number of iterations required for convergence.

At first sight, the amount of computation seems to be demanding. However, a close examination reveals that each iteration (E-step and M-step) of the standard EM algorithm also takes $O(NDK)$ time. The value of K in the standard EM, though, is usually smaller, because the proposed algorithm starts with a larger number of components. The number of iterations required for our algorithm is also, in general, larger because of the increase of the number of parameters. Therefore, it is true that the proposed algorithm takes more time than the standard EM algorithm *with one parameter setting*. However, the proposed algorithm can determine both the number of clusters and feature subsets. If we want to achieve the same goal with the standard EM algorithm using a wrapper approach, we need to rerun EM multiple times, with different number of components and different feature subsets. The computational demand is much heavier than the proposed algorithm, even with heuristic search to guide the selection of feature subsets. Another strength of the proposed algorithm is that by initialization with a large number of Gaussian components, the algorithm is less sensitive to the local minimum problem than the standard EM algorithm. We can further reduce the complexity by adopting optimization techniques applicable for standard EM for Gaussian mixture, such as sampling the data, compressing the data [8], or using efficient data structures [45], [54].

For the postprocessing step in Section 3.4, each computation of the quantity J and its gradient and Hessian takes $O(NDK)$ time. The number of iterations is difficult to predict, as it depends on the optimization routine. However, we can always put an upper bound on the number of iterations and trade speed for the optimality of the results.

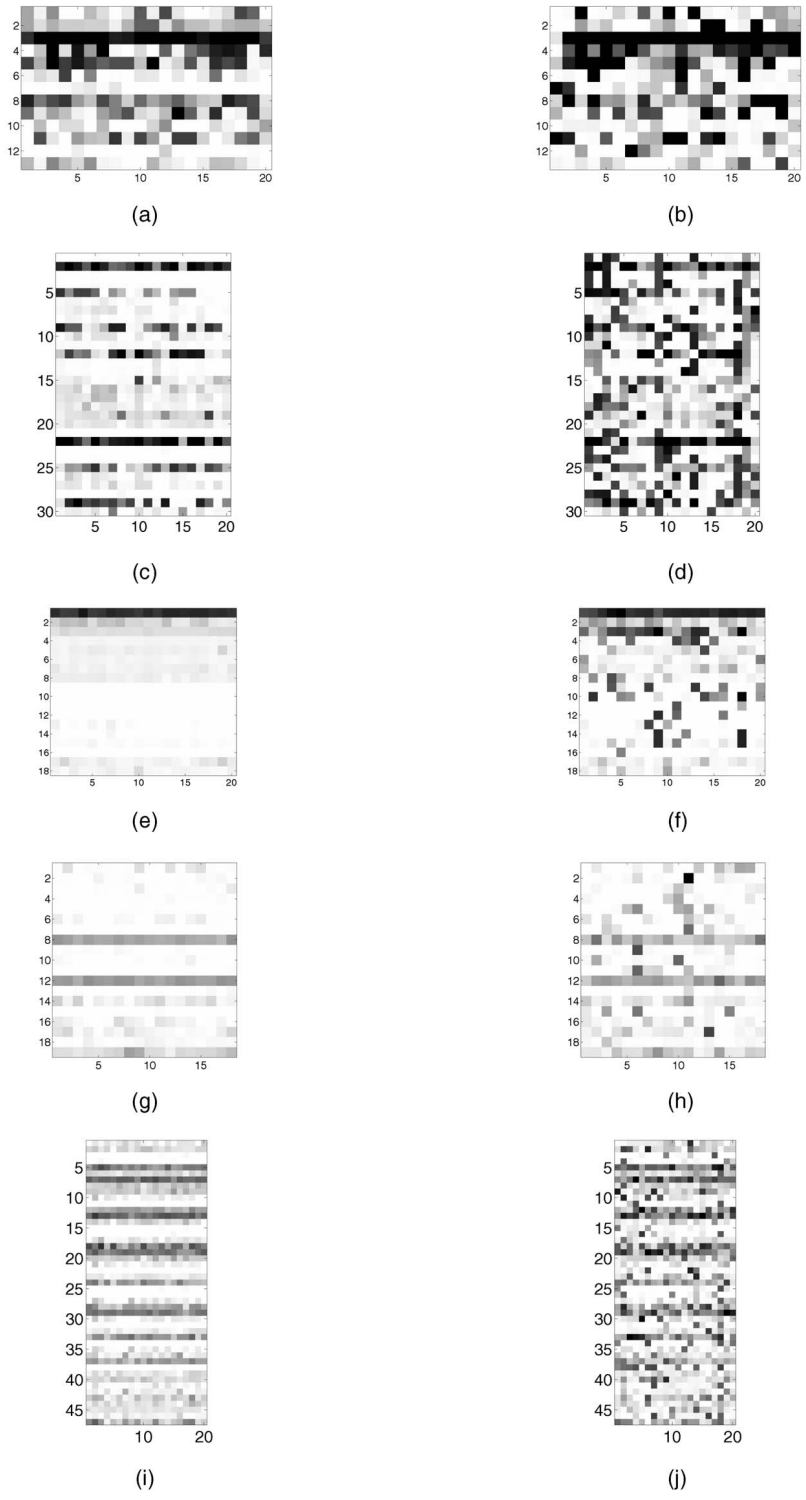


Fig. 9. Image maps of feature saliency for different data sets with and without the postprocessing procedure. Feature saliency of 1 (0) is shown as a pixel of gray level 255 (0). The vertical and horizontal axes correspond to the feature number and the trial number, respectively. (a) `wine`, proposed algorithm, (b) `wine`, after postprocessing, (c) `wdbc`, proposed algorithm, (d) `wdbc`, after postprocessing, (e) `image`, proposed algoriothm, (f) `image`, after postprocessing, (g) `texture`, proposed algorithm, (h) `texture`, after postprocessing, (i) `zernike`, proposed algorithm, and (j) `zernike`, after postprocessing.

5.2 Relation to Shrinkage Estimate

One interpretation of (6) is that we "regularize" the distribution of each feature in different components by the common distribution. This is analogous to the shrinkage estimator for covariance matrices of class-conditional densities [15], which is the weighted sum of an estimate of the class-specific covariance matrix, and the "global" covariance matrix estimate. In (6), the pdf of the lth feature is also a weighted sum of a component-specific pdf and a common density. An important difference here is that the weight ρ_l is estimated from the data, using the MML principle, instead of set heuristically, as is commonly done. As shrinkage estimators

have found empirical success to combat data scarcity, this "regularization" viewpoint is an alternative explanation for the usefulness of the proposed algorithm.

5.3 Limitation of the Proposed Algorithm

A limitation of the proposed algorithm is the feature independence assumption (conditioned on the component). While, empirically, the violation of the independence assumption usually does not affect the accuracy of a classifier (as in supervised learning) or the quality of clusters (as in unsupervised learning), this has some negative influence on the feature selection problem. Specifically, a feature that is redundant because its distribution is independent of the component label given another feature cannot be modeled under the feature independence assumption. As a result, both features are kept. This explains why, in general, the feature saliencies are somewhat high. The postprocessing in Section 3.4 can cope with this problem because it considers the posterior distribution and, therefore, can discard features that do not help in identifying the clusters directly.

5.4 Extension to Semisupervised Learning

Sometimes, we may have some knowledge of the class labels of different Gaussian components. This can happen when, say, we adopt a procedure to combine different Gaussian components to form a cluster (e.g., as in [51]), or in a semisupervised learning scenario, where we can use a small amount of labeled data to help us identify which Gaussian component belongs to which class. This additional information can suggest combination of several Gaussian components to form a single class/cluster, thereby allowing the identification of non-Gaussian clusters. The postprocessing step can take advantage of this information.

Suppose we know there are C classes and the posterior probability that pattern $\mathbf{y}_i$ belongs to the cth class, denoted r_{ic}, can be computed as $r_{ic} = \sum_{j=1}^{K} \beta_{cj} P(z_i = j|\mathbf{y}_i)$. For example, if we know that the components 4, 6, and 10 are from class 2, we can set $\beta_{2,4} = \beta_{2,6} = \beta_{2,10} = 1/3$ and the other $\beta_{2,j}$ to be zero. The postprocessing is modified accordingly: Redefine t_i in (23) to $t_i = \arg\max_c r_{ic}$, i.e., it becomes the class label for $\mathbf{y}_i$ in view of the extra information; replace $\log P(z_i = t_i|\mathbf{y}_i)$ in (23) by $\log r_{i,t_i}$. The gradient and Hessian can still be computed easily after noting that

$$\begin{aligned} \frac{\partial w_{ij}}{\partial \rho_l} &= w_{ij} \frac{\partial}{\partial \rho_l} \log w_{ij} = w_{ij}(h_{ilj} - g_{il}) \\ \frac{\partial}{\partial \rho_l} \log r_{ic} &= \frac{1}{r_{ic}} \sum_{j=1}^{K} \beta_{cj} \frac{\partial}{\partial \rho_l} w_{ij} = \sum_{j=1}^{K} \frac{\beta_{cj} w_{ij}}{r_{ic}} h_{ilj} - g_{il}. \end{aligned} \tag{24}$$

We can then optimize the modified J in (23) to carry out the postprocessing.

5.5 A Note on Maximizing the Posterior Probability

The sum of the logarithm of the maximum posterior probability considered in the postprocessing in Section 3.4 can be regarded as the sample estimate of an unorthodox type of entropy (see [30]) for the posterior distribution. It can be regarded as the limit of Renyi's entropy $R_\alpha(P)$ when α tends to infinity, where

$$R_\alpha(P) = \frac{1}{1-\alpha} \log \sum_{j=1}^{K} p_i^\alpha. \tag{25}$$

When this entropy is used for parameter estimation under the maximum entropy framework, the corresponding procedure is closely related to minimax inference. Other functions on the posterior probabilities can also be used, such as the Shannon entropy of the posterior distribution. Preliminary studies show that the use of different types of entropy does not affect the results significantly.

6 Conclusions

In this paper, we have presented an EM algorithm to estimate the importance of different features and the best number of components for Gaussian-mixture clustering. The proposed algorithm can avoid running EM many times with different numbers of components and different feature subsets, and can achieve better performance than using all the available features for clustering. The usefulness of the algorithm was demonstrated on both synthetic and benchmark real data sets.

There are several avenues for future work. The space complexity of the proposed algorithm is $O(NDK)$, which can slow down the algorithm significantly when the data set (which is of size $O(ND)$) is so large that the intermediate variables cannot be held in memory. How to extend the algorithm to cope with this is a challenging problem. We also may attempt to model the dependency between different features explicitly. Merging the proposed algorithm, which is basically a wrapper, with other filter techniques, can lead to a hybrid algorithm that is applicable for data sets with enormous numbers of features. We can replace the mixture of Gaussians by a mixture of multinomial distribution, thereby making the proposed algorithm also applicable to categorical data. One may also extend the current algorithm to handle different salient features for different components. Finally, principles other than MML, such as variational Bayes [12], can be adopted to perform model selection.

Appendix A
The Mixture Model

Recall (5), which is the conditional density of $\mathbf{y}$, given $\Phi = (\phi_1, \ldots, \phi_D)$,

$$p(\mathbf{y}|\Phi) = \sum_{j=1}^{K} \alpha_j \prod_{l=1}^{D} (p(y_l|\theta_{jl}))^{\phi_l} (q(y_l|\lambda_l))^{1-\phi_l}.$$

We treat Φ as a set of missing variables and define $\rho_l = P(\phi_l = 1)$, for $l = 1, \ldots, D$, as a set of parameters to be estimated (the feature saliencies). We assume the ϕ_ls are mutually independent and also independent of the hidden component label z for any pattern $\mathbf{y}$. Thus,

$$\begin{aligned} p(\mathbf{y}, \Phi) &= p(\mathbf{y}|\Phi) p(\Phi) \\ &= \left(\sum_{j=1}^{K} \alpha_j \prod_{l=1}^{D} (p(y_l|\theta_{jl}))^{\phi_l} (q(y_l|\lambda_l))^{1-\phi_l} \right) \prod_{l=1}^{D} \rho_l^{\phi_l} (1-\rho_l)^{1-\phi_l} \\ &= \sum_{j=1}^{K} \alpha_j \prod_{l=1}^{D} (\rho_l p(y_l|\theta_{jl}))^{\phi_l} ((1-\rho_l) q(y_l|\lambda_l))^{1-\phi_l}. \end{aligned} \tag{26}$$

The marginal density for $\mathbf{y}$ is

$$\begin{aligned} p(\mathbf{y}) &= \sum_{\Phi} p(\mathbf{y}, \Phi) \\ &= \sum_{j=1}^{K} \alpha_j \sum_{\Phi} \prod_{l=1}^{D} (\rho_l p(y_l|\theta_{jl}))^{\phi_l} ((1-\rho_l)\ q(y_l|\lambda_l))^{1-\phi_l} \\ &= \sum_{j=1}^{K} \alpha_j \prod_{l=1}^{D} \sum_{\phi_l=0}^{1} (\rho_l p(y_l|\theta_{jl}))^{\phi_l} ((1-\rho_l)\ q(y_l|\lambda_l))^{1-\phi_l} \\ &= \sum_{j=1}^{K} \alpha_j \prod_{l=1}^{D} \big(p(y_l|\theta_{jl})\rho_l + q(y_l|\lambda_l)(1-\rho_l)\big), \end{aligned} \tag{27}$$

which is (6). Note that the features are independent, given the component label z.

Appendix B

Deriving the EM Algorithm

The complete-data log-likelihood for the model in (6) is

$$P(\mathbf{y}_i, z_i = j, \Phi) = \alpha_j \prod_{l=1}^{D} (\rho_l p(y_{il}|\theta_{jl}))^{\phi_l} ((1-\rho_l) q(y_{il}|\lambda_l))^{1-\phi_l}. \tag{28}$$

Define the following quantities:

$$\begin{aligned} &w_{ij} = P(z_i = j|\mathbf{y}_i), \qquad u_{ijl} = P(z_i = j, \phi_l = 1|\mathbf{y}_i), \\ &v_{ijl} = P(z_i = j, \phi_l = 0|\mathbf{y}_i). \end{aligned}$$

They are calculated using the current parameter estimate θ^{now}. Note that $u_{ijl} + v_{ijl} = w_{ij}$ and $\sum_{i=1}^{N}\sum_{j=1}^{K} w_{ij} = n$. The expected complete data log-likelihood based on θ^{now} is

$$\begin{aligned} &E_{\theta^{\text{now}}}[\log P(\mathcal{Y}, \mathbf{z}, \Phi)] \\ &= \sum_{i,j,\Phi} P(z_i = j, \Phi|\mathbf{y}_i) \Bigg(\log \alpha_j + \sum_{l} \Big(\phi_l (\log p(y_{il}|\theta_{jl}) + \\ &\quad \log \rho_l) + (1-\phi_l)(\log q(y_{il}|\lambda_l) + \log(1-\rho_l)) \Big) \Bigg) \\ &= \sum_{i,j} P(z_i = j|\mathbf{y}_i) \log \alpha_j + \sum_{i,j} \sum_{l} \sum_{\phi_l=0}^{1} P(z_i = j, \phi_l|\mathbf{y}_i) \\ &\quad \Big(\phi_l (\log p(y_{il}|\theta_{jl}) + \log \rho_l) \\ &\quad + (1-\phi_l)(\log q(y_{il}|\lambda_l) + \log(1-\rho_l)) \Big) \\ &= \underbrace{\sum_{j} \Big(\sum_{i} w_{ij} \Big) \log \alpha_j}_{\text{part 1}} + \underbrace{\sum_{j,l} \sum_{i} u_{ijl} \log p(y_{il}|\theta_{jl})}_{\text{part 2}} \\ &\quad + \underbrace{\sum_{l} \sum_{i,j} v_{ijl} \log q(y_{il}|\lambda_l)}_{\text{part 3}} \\ &\quad + \sum_{l} \Bigg(\underbrace{\log \rho_l \sum_{i,j} u_{ijl} + \log(1-\rho_l) \sum_{i,j} v_{ijl}}_{\text{part 4}} \Bigg). \end{aligned}$$

The four parts in the equation above can be maximized separately. Recall that the densities $p(.)$ and $q(.)$ are univariate Gaussian and are characterized by their means and variances. As a result, maximizing the expected complete data log-likelihood leads to the M-step in (13)-(18). Finally, observe that

$$\begin{aligned} P(\phi_l = 1|z_i = j, \mathbf{y}_i) &= \frac{P(\phi_l = 1, \mathbf{y}_i|z_i = j)}{P(\mathbf{y}_i|z_i = j)} \\ &= \frac{\rho_l p(y_l|\theta_{jl}) \prod_{l' \neq l} \big(\rho_{l'} p(y_{l'}|\theta_{jl'}) + (1-\rho_{l'}) q(y_{l'}|\lambda_{l'})\big)}{\prod_{l'} \big(\rho_{l'} p(y_{l'}|\theta_{jl'}) + (1-\rho_{l'}) q(y_{l'}|\lambda_{l'})\big)} \\ &= \frac{\rho_l p(y_l|\theta_{jl})}{\rho_l p(y_l|\theta_{jl}) + (1-\rho_l) q(y_l|\lambda_l)} = \frac{a_{ijl}}{c_{ijl}}. \end{aligned}$$

Therefore, (11) follows because

$$u_{ijl} = P(\phi_l = 1|z_i = j, \mathbf{y}_i) P(z_i = j|\mathbf{y}_i) = \frac{a_{ijl}}{c_{ijl}} w_{ij}. \tag{29}$$

Appendix C

Applying Minimum Message Length

The *minimum message length* (MML) criterion is given by (see [18] for details and references)

$$\begin{aligned} \hat{\theta} = \arg\min_{\theta} \Big\{ &- \log p(\theta) - \log p(\mathcal{Y}|\theta) + \frac{1}{2} \log |\mathbf{I}(\theta)| + \\ &\frac{c}{2} \Big(1 + \log \frac{1}{12}\Big) \Big\}, \end{aligned} \tag{30}$$

where θ is the set of parameter of the model, c is the dimension of θ, $\mathbf{I}(\theta) = -E[D_\theta^2 \log p(\mathcal{Y}|\theta)]$ is the (expected) Fisher information matrix (the negative expected value of the Hessian of the log-likelihood), and $|\mathbf{I}(\theta)|$ is the determinant of $\mathbf{I}(\theta)$. The information matrix for the model (6) is very difficult to obtain analytically. Therefore, as in [18], we approximate it by the information matrix of the complete data log-likelihood, $\mathbf{I}_c(\theta)$. By differentiating the logarithm of (28), we can show that

$$\begin{aligned} \mathbf{I}_c(\theta) = \text{block} - \text{diag} \Big[&\mathcal{M}, \frac{1}{\rho_1(1-\rho_1)}, \ldots, \frac{1}{\rho_D(1-\rho_D)}, \\ &\alpha_1 \rho_1 \mathbf{I}(\theta_{11}), \ldots, \alpha_1 \rho_D \mathbf{I}(\theta_{1D}), \\ &\alpha_2 \rho_1 \mathbf{I}(\theta_{21}), \ldots, \alpha_2 \rho_D \mathbf{I}(\theta_{2D}), \ldots, \alpha_K \rho_1 \mathbf{I}(\theta_{K1}), \ldots, \\ &\alpha_K \rho_D \mathbf{I}(\theta_{KD}), (1-\rho_1)\mathbf{I}(\lambda_1), \ldots, (1-\rho_D)\mathbf{I}(\lambda_D) \Big], \end{aligned} \tag{31}$$

where $\mathcal{M}$ is the information matrix of the multinominal distribution with parameters $(\alpha_1, \ldots, \alpha_K)$. The size of $\mathbf{I}(\theta)$ is $(K + D + KDR + DS)$, where R and S are the number of parameters in θ_{jl} and λ_l, respectively. Note that $(\rho_l(1-\rho_l))^{-1}$ is the information of a Bernoulli distribution with parameter ρ_l. Thus, we can write

$$\begin{aligned} |\mathbf{I}_c(\theta)| = &\log \mathbf{I}(\{\alpha_j\}) + \sum_{l=1}^{D} \log I(\rho_l) + R \sum_{j=1}^{K} \sum_{l=1}^{D} \log(\alpha_j \rho_l) \\ &+ \sum_{j=1}^{K} \sum_{l=1}^{D} \log \mathbf{I}(\theta_{jl}) + S \sum_{l=1}^{D} \log(1-\rho_l) + \sum_{l=1}^{D} \mathbf{I}(\lambda_l). \end{aligned} \tag{32}$$

For the prior densities of the parameters, we assume that different groups of parameters are independent. Specifically, $\{\alpha_j\}$, ρ_l (for different values of l), θ_{jl} (for different values of j and l), and λ_l (for different values of l) are independent. Furthermore, since we have no knowledge about the parameters, we adopt noninformative Jeffrey's priors (see [18] for details and references) which are proportional to the square root of the determinant of the corresponding information matrices. When we substitute $p(\theta)$ and $|\mathbf{I}(\theta)|$ into (30), and drop the order-one term, we obtain our final criterion, which is (19):

$$\hat{\theta} = \arg\min_{\theta} \Big\{ -\log p(\mathcal{Y}|\theta) + \frac{1}{2}(K + D + KDR + DS) \log n + \frac{R}{2}\sum_{j=1}^{K}\sum_{l=1}^{D}\log(\alpha_j \rho_l) + \frac{S}{2}\sum_{l=1}^{D}\log(1-\rho_l) \Big\}.$$

ACKNOWLEDGMENTS

This research was supported by ONR grant number N000140410183. M.A.T. Figueiredo's research was supported by the (Portugese) Foundation for Science and Technology under project POSI/33143/SRI/2000.

REFERENCES

[1] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, "Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications," *Proc. 1998 ACM SIGMOD Int'l Conf. Management of Data,* pp. 94-105, 1998.

[2] P. Arabie and L. Hubert, "Cluster Analysis in Marketing Research," *Advanced Methods of Marketing Research,* R.P. Bagozzi, ed., pp. 160-189, 1994.

[3] P. Baldi and G.W. Hatfield, *DNA Microarrays and Gene Expression.* Cambridge Univ. Press, 2002.

[4] R. Battiti, "Using Mutual Information for Selecting Features in Supervised Neural Net Learning," *IEEE Trans. Neural Networks,* vol. 5, no. 4, pp. 537-550, 1994.

[5] S.K. Bhatia and J.S. Deogun, "Conceptual Clustering in Information Retrieval," *IEEE Trans. Systems, Man, and Cybernetics, Part B,* vol. 28, no. 3, pp. 427-436, 1998.

[6] J. Bins and B. Draper, "Feature Selection from Huge Feature Sets," *Proc. Eighth Int'l Conf. Computer Vision,* pp. 159-165, 2001.

[7] A. Blum and P. Langley, "Selection of Relevant Features and Examples in Machine Learning," *Artificial Intelligence,* vol. 97, nos. 1-2, pp. 245-271, 1997.

[8] P. Bradley, U. Fayyad, and C. Reina, "Clustering Very Large Database Using EM Mixture Models," *Proc. 15th Int'l Conf. Pattern Recognition (ICPR-2000),* pp. 76-80, 2000.

[9] R. Caruana and D. Freitag, "Greedy Attribute Selection," *Proc. 11th Int'l Conf. Machine Learning,* pp. 28-36, 1994.

[10] G. Celeux, S. Chrétien, F. Forbes, and A. Mkhadri, "A Component-Wise EM Algorithm for Mixtures," *J. Computational and Graphical Statistics,* vol. 10, pp. 699-712, 2001.

[11] A. Chaturvedi and J.D. Carroll, "A Feature-Based Approach to Market Segmentation via Overlapping k-Centroids Clustering," *J. Marketing Research,* vol. 34, no. 3, pp. 370-377, 1997.

[12] A. Corduneanu and C.M. Bishop, "Variational Bayesian Model Selection for Mixture Distributions," *Proc. Eighth Int'l Conf. Artificial Intelligence and Statistics,* pp. 27-34, 2001.

[13] M. Dash and H. Liu, "Feature Selection for Clustering," *Proc. Pacific-Asia Conf. Knowledge Discovery and Data Mining,* 2000.

[14] M. Devaney and A. Ram, "Efficient Feature Selection in Conceptual Clustering," *Proc. 14th Int'l Conf. Machine Learning,* pp. 92-97, 1997.

[15] R. Duda, P. Hart, and D. Stork, *Pattern Classification.* New York: John Wiley and Sons, 2001.

[16] J.G. Dy and C.E. Brodley, "Feature Subset Selection and Order Identification for Unsupervised Learning," *Proc. 17th Int'l Conf. Machine Learning,* pp. 247-254, 2000.

[17] J.G. Dy, C.E. Brodley, A. Kak, L.S. Broderick, and A.M. Aisen, "Unsupervised Feature Selection Applied to Content-Based Retrieval of Lung Images," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, no. 3, pp. 373-378, Mar. 2003.

[18] M.A.T. Figueiredo and A.K. Jain, "Unsupervised Learning of Finite Mixture Models," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 3, pp. 381-396, Mar. 2002.

[19] M.A.T. Figueiredo, A.K. Jain, and M.H. Law, "A Feature Selection Wrapper for Mixtures," *Proc. First Iberian Conf. Pattern Recognition and Image Analysis,* pp. 229-237, 2003.

[20] Y. Freund and R.E. Schapire, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," *J. Computational and Graphical Statistics,* vol. 55, no. 1, pp. 119-139, 1997.

[21] H. Frigui and R. Krishnapuram, "A Robust Competitive Clustering Algorithm with Applications in Computer Vision," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 21, no. 5, pp. 450-465, May 1999.

[22] P. Gustafson, P. Carbonetto, N. Thompson, and N. de Freitas, "Bayesian Feature Weighting for Unsupervised Learning, with Application to Object Recognition," *Proc. Ninth Int'l Workshop Artificial Intelligence and Statistics (AISTAT03),* 2003.

[23] M. Iwayama and T. Tokunaga, "Cluster-Based Text Categorization: A Comparison of Category Search Strategies," *Proc. 18th ACM Int'l Conf. Research and Development in Information Retrieval,* pp. 273-281, 1995.

[24] A. Jain and D. Zongker, "Feature Selection: Evaluation, Application, and Small Sample Performance," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 2, pp. 153-157, Feb. 1997.

[25] A.K. Jain and R.C. Dubes, *Algorithms for Clustering Data.* Prentice Hall, 1988.

[26] A.K. Jain, R. Duin, and J. Mao, "Statistical Pattern Recognition: A Review," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 1, pp. 4-38, Jan. 2000.

[27] A.K. Jain and F. Farrokhnia, "Unsupervised Texture Segmentation Using Gabor Filters," *Pattern Recognition,* vol. 24, pp. 1167-1186, 1991.

[28] A.K. Jain and P. Flynn, "Image Segmentation Using Clustering," *Advances in Image Understanding,* pp. 65-83, 1996.

[29] A.K. Jain, M.N. Murty, and P.J. Flynn, "Data Clustering: A Review," *ACM Computing Surveys,* vol. 31, no. 3, pp. 264-323, Sept. 1999.

[30] J.N. Kapur, *Measures of Information and Their Applications.* New Delhi, India: Wiley, 1994.

[31] Y. Kim, W. Street, and F. Menczer, "Feature Selection in Unsupervised Learning via Evolutionary Search," *Proc. Sixth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining,* pp. 365-369, 2000.

[32] K. Kira and L. Rendell, "The Feature Selection Problem: Traditional Methods and a New Algorithm," *Proc. 10th Nat'l Conf. Artificial Intelligence (AAAI-92),* pp. 129-134, 1992.

[33] R. Kohavi and G. John, "Wrappers for Feature Subset Selection," *Artificial Intelligence,* vol. 97, nos. 1-2, pp. 273-324, 1997.

[34] D. Koller and M. Sahami, "Toward Optimal Feature Selection," *Proc. 13th Int'l Conf. Machine Learning,* pp. 284-292, 1996.

[35] J. Komorowski, L. Polkowski, and A. Skowron, "Rough Sets: A Tutorial," *Rough-Fuzzy Hybridization: A New Method for Decision Making,* Singapore: Springer-Verlag, 1998.

[36] I. Kononenko, "Estimating Attributes: Analysis and Extensions of RELIEF," *Proc. Seventh European Conf. Machine Learning,* pp. 171-182, 1994.

[37] N. Kwak and C.-H. Choi, "Input Feature Selection by Mutual Information Based on Parzen Window," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 12, pp. 1667-1671, Dec. 2002.

[38] M.H. Law, A.K. Jain, and M.A.T. Figueiredo, "Feature Selection in Mixture-Based Clustering," *Advances in Neural Information Processing Systems 15,* pp. 625-632, Cambridge, Mass.: MIT Press, 2003.

[39] G. McLachlan and K. Basford, *Mixture Models: Inference and Application to Clustering.* New York: Marcel Dekker, 1988.

[40] A.J. Miller, *Subset Selection in Regression.* London: Chapman & Hall, 2002.

[41] B. Mirkin, "Concept Learning and Feature Selection Based on Square-Error Clustering," *Machine Learning,* vol. 35, pp. 25-39, 1999.

[42] P. Mitra and C.A. Murthy, "Unsupervised Feature Selection Using Feature Similarity," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 3, pp. 301-312, Mar. 2002.

[43] D. Modha and W. Scott-Spangler, "Feature Weighting in k-Means Clustering," *Machine Learning,* vol. 52, no. 3, pp. 217-237, 2003.

[44] J. Novovicová, P. Pudil, and J. Kittler, "Divergence Based Feature Selection for Multimodal Class Densities," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 18, no. 2, pp. 218-223, Feb. 1996.
[45] D. Pelleg and A.W. Moore, "X-Means: Extending k-Means with Efficient Estimation of the Number of Clusters," *Proc. 17th Int'l Conf. Machine Learning,* pp. 727-734, 2000.
[46] J. Pena, J. Lozano, P. Larranaga, and I. Inza, "Dimensionality Reduction in Unsupervised Learning of Conditional Gaussian Networks," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, no. 6, pp. 590-603, June 2001.
[47] P. Pudil, J. Novovicovà, and J. Kittler, "Floating Search Methods in Feature Selection," *Pattern Recognition Letters,* vol. 15, pp. 1119-1125, 1994.
[48] P. Pudil, J. Novovicová, and J. Kittler, "Feature Selection Based on the Approximation of Class Densities by Finite Mixtures of the Special Type," *Pattern Recognition,* vol. 28, no. 9, pp. 1389-1398, 1995.
[49] S.J. Raudys and A.K. Jain, "Small Sample Size Effects in Statistical Pattern Recognition: Recommendations for Practitioners," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 13, no. 3, pp. 252-264, Mar. 1991.
[50] J. Rissanen, *Stochastic Complexity in Stastistical Inquiry.* Singapore: World Scientific, 1989.
[51] S.J. Roberts, R.M. Everson, and I. Rezek, "Maximum Certainty Data Partitioning," *Pattern Recognition,* vol. 33, no. 5, pp. 833-839, 1999.
[52] V. Roth and T. Lange, "Feature Selection in Clustering Problems," *Advances in Neural Information Processing Systems 16,* Cambridge, Mass.: MIT Press, 2004.
[53] M. Sahami, "Using Machine Learning to Improve Information Access," PhD thesis, Computer Science Dept., Stanford Univ., 1998.
[54] P. Sand and A.W. Moore, "Repairing Faulty Mixture Models Using Density Estimation," *Proc. 18th Int'l Conf. Machine Learning,* pp. 457-464, 2001.
[55] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 8, pp. 888-905, Aug. 2000.
[56] L. Talavera, "Dependency-Based Feature Selection for Clustering Symbolic Data," *Intelligent Data Analysis,* vol. 4, pp. 19-28, 2000.
[57] G. Trunk, "A Problem of Dimensionality: A Simple Example," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 1, no. 3, pp. 306-307, 1979.
[58] S. Vaithyanathan and B. Dom, "Generalized Model Selection for Unsupervised Learning in High Dimensions," *Advances in Neural Information Processing Systems 12,* pp. 970-976, Cambridge, Mass.: MIT Press, 1999.
[59] P. Viola and M. Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 511-518, 2001.
[60] C.S. Wallace and D.L. Dowe, "MML Clustering of Multi-State, Poisson, von Mises Circular and Gaussian Distributions," *Statistics and Computing,* vol. 10, pp. 73-83, 2000.
[61] C.S. Wallace and P. Freeman, "Estimation and Inference via Compact Coding," *J. Royal Statistical Soc. (B),* vol. 49, no. 3, pp. 241-252, 1987.
[62] E. Xing, M. Jordan, and R. Karp, "Feature Selection for High-Dimensional Genomic Microarray Data," *Proc. 18th Int'l Conf. Machine Learning,* pp. 601-608, 2001.
[63] J. Yang and V. Honavar, "Feature Subset Selection Using a Genetic Algorithm," *IEEE Intelligent Systems,* vol. 13, pp. 44-49, 1998.

Martin H.C. Law received the BEng degree and MPhil degree in computer science from Hong Kong University of Science and Technology. After working in Hong Kong Baptist University and Hong Kong University of Science and Technology for more than two years, he moved to the US and is currently a PhD candidate in the Department of Computer Science and Engineering at Michigan State University. His research interests include data clustering, mixture models, manifold learning, dimensionality reduction, and kernel methods. He is a student member of the IEEE.

Mário A.T. Figueiredo (S'87-M'95-SM'2000) received the EE, MSc, and PhD degrees in electrical and computer engineering, all from the Instituto Superior Técnico (I.S.T.), the engineering school of the Technical University of Lisbon, Portugal, in 1985, 1990, and 1994, respectively. Since 1994, he has been with Department of Electrical and Computer Engineering, I.S.T. He is also a researcher and area coordinator at the Institute of Telecommunications, Lisbon. In 1998, he held a visiting position with the Department of Computer Science and Engineering at Michigan State University, East Lansing, Michigan. His scientific interests include image processing and analysis, computer vision, statistical pattern recognition, and statistical learning. He received the Portuguese IBM Scientific Prize in 1995 for work on unsupervised image restoration. He is an associate editor of the journals *Pattern Recognition Letters*, *IEEE Transactions on Image Processing*, and *IEEE Transactions on Mobile Computing*. He is also guest coeditor of special issues of the journals *IEEE Transactions on Pattern Analysis and Machine Intelligence* and *IEEE Transactions on Signal Processing*. He was a cochair of the 2001 and 2003 Workshops on Energy Minimization Methods in Computer Vision and Pattern Recognition, and local chair of the 2004 Joint IAPR International Workshops on Syntactical and Structural Pattern Recognition and Statistical Pattern Recognition. He has been a member of program committees of several international conferences, including CVPR, EECV, ICASSP, and ICPR. He is a senior member of the IEEE.

Anil K. Jain is a University Distinguished Professor in the Departments of Computer Science & Engineering and Electrical & Computer Engineering at Michigan State University. He was the department chair from 1995-1999. His research interests include statistical pattern recognition, exploratory pattern analysis, texture analysis, document image analysis, and biometric authentication. Several of his papers have been reprinted in edited volumes on image processing and pattern recognition. He received the best paper awards in 1987 and 1991, and received certificates for outstanding contributions in 1976, 1979, 1992, 1997, and 1998 from the Pattern Recognition Society. He also received the 1996 *IEEE Transactions on Neural Networks* Outstanding Paper Award. He is a fellow of the IEEE, ACM, and International Association of Pattern Recognition (IAPR). He has received a Fulbright Research Award, a Guggenheim fellowship, and the Alexander von Humboldt Research Award. He delivered the 2002 Pierre Devijver lecture sponsored by the IAPR. He holds six patents in the area of fingerprint matching. He is the author of the following books: *Handbook of Fingerprint Recognition* (Springer 2003) (received the 2003 PSP award from the Association of American Publishers), *BIOMETRICS: Personal Identification in Networked Society* (Kluwer 1999), *3D Object Recognition Systems* (Elsevier 1993), *Markov Random Fields: Theory and Applications* (Academic Press 1993), *Neural Networks and Statistical Pattern Recognition* (North-Holland 1991), *Analysis and Interpretation of Range Images* (Springer-Verlag 1990), *Algorithms For Clustering Data* (Prentice-Hall 1988), and *Real-Time Object Measurement and Classification* (Springer-Verlag 1988).