Simultaneous Localized Feature Selection and Model Detection for Gaussian Mixtures

Yuanhong Li, Ming Dong*, *Member, IEEE* and Jing Hua, *Member, IEEE*
Department of Computer Science
Wayne State University, Detroit, MI 48202

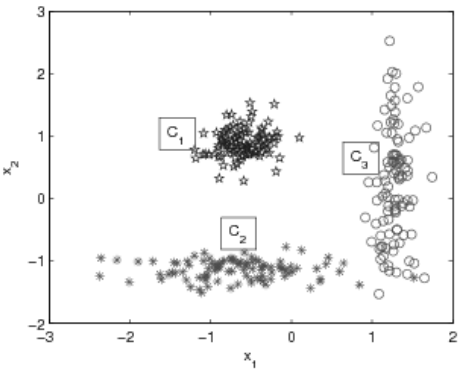
***Abstract*—In this paper, we propose a novel approach of simultaneous localized feature selection and model detection for unsupervised learning. In our approach, local feature saliency, together with other parameters of Gaussian mixtures, are estimated by Bayesian variational learning. Experiments performed on both synthetic and real-world datasets demonstrate that our approach is superior over both global feature selection and subspace clustering methods.**

***Index Terms*—Unsupervised, localized, feature selection, Bayesian.**

I. Introduction

Clustering is the unsupervised classification of data objects into different groups (clusters) such that objects in one group are similar together and dissimilar from another group. Data clustering found its applications in many fields, such as information discovering, text mining, web analysis, image grouping, and bioinformatics. Typically, a clustering algorithm considers all the available features to "learn" from data. In practice, however, some features can be irrelevant and therefore hinder the clustering performance, especially in high-dimensional datasets. A viable solution is *feature selection*, a technique that chooses the "best" feature subset for clustering.

Feature selection has been extensively studied in supervised learning scenarios [1], [3]–[7]. In unsupervised learning, feature selection becomes a more complex problem due to the unavailability of class labels [8]–[10]. The objective of feature selection is threefold: improving the performance of clustering, facilitating a fast and cost-efficient solution, and providing a better understanding of the underlying process that generated the data. In general, unsupervised feature selection algorithms conduct feature selection in a *global* sense by producing a common feature subset for all the clusters. However, this can be invalid in the clustering practice, where the local intrinsic property of data plays a more important role [2]. In the illustrative example shown in Figure 1 (a), the relevant feature subset for cluster C_1 is $\{x_1, x_2\}$, while clusters C_2 and C_3 can be grouped using $\{x_2\}$ and $\{x_1\}$, respectively. A common feature subset, i.e., $\{x_1, x_2\}$, is unable to reflect the inherent structural property of the three clusters. Apparently, clustering with *local* features is highly desired. From a probability perspective, the local saliency of the l-th feature with respect to the j-th cluster indicates that the sample distribution of the j-th cluster has a strong peak on the l-th feature. On the other hand, a non-salient feature does not have such a natural cluster



(a) Three clusters in 2-D view.

(b) Distribution of individual clusters on each feature.

Fig. 1: A three-cluster dataset with cluster C_1 embedded in feature set $\{x_1, x_2\}$, cluster C_2 embedded in feature subset $\{x_2\}$, and cluster C_3 embedded in feature subset $\{x_1\}$.

structure (see Figure 1 (b) for an example). Moreover, in real-world problems, the number of clusters is usually unknown, and needs to be detected in the clustering process. Note that different feature subsets may lead to different numbers of clusters. Feature selection and model detection are strongly dependent [11]. This suggests that these two objectives must be pursued simultaneously.

In this paper, we address the problem of simultaneous localized feature selection and model detection for unsupervised learning. We propose a novel localized Bayesian inference approach of Gaussian mixtures, which computes the local feature saliency, the number of clusters, and other parameters of a mixture model through variational learning. The rest of this paper is organized as follows: Section II briefly reviews related work in the literature. The Gaussian mixture model with local feature saliency is presented in Section III. Variational learning is used in Section IV to identify the

*:Corresponding author, email: mdong@cs.wayne.edu

mixture model and perform localized feature selection. The experimental results are presented in Section V, and Section VI concludes the paper.

II. Related Work

Most global approaches for unsupervised feature selection can be categorized as filters, wrappers or hybrids [12]. Filter approaches pre-select features and then provide the selected features as an input to a clustering algorithm. Wrapper approaches, on the other hand, incorporate the clustering algorithm in feature searching and selection. A hybrid method tries to incorporate the advantages of both filters and wrappers. It uses criteria, typically independent to mining algorithms, to find a feature subset. A mining algorithm is then employed to decide the final feature subset. Mitra et al. [10] propose a filter approach that accomplishes feature selection in two steps. In the first step, the features are partitioned into a number of subsets by k-Nearest-Neighbor (KNN) rule based on the feature similarity calculated from an information compression index. Then, the feature that has the most compact subset is selected, and its k neighboring features are discarded. Dash et al. [13] introduce an entropy measure which is low if the data has distinct clusters, and high otherwise. The feature importance is evaluated based on this entropy measure, and a relevant feature subset is chosen accordingly. The wrapper method presented in [9] evaluates the cluster quality over different feature subsets by normalizing cluster separability (for k-means clustering) or likelihood (for Expectation Maximization (EM) clustering) using the cross-projection method. The candidate feature subsets are generated by a sequential forward search. The number of clusters is estimated by merging clusters one at a time based on the Bayesian Information Criterion (BIC). Law et al. [11] assume that features are independent given a mixture component, and follow a common distribution up to a probability. The complement of this probability is defined as feature saliency and estimated by the Maximum Likelihood (ML) or Maximum A Priori (MAP) with the EM algorithm using Gaussian mixture models. Minimal Message Length (MML) is used to estimate the number of components. [14] and [8] employ the same Gaussian mixture model as in [11] to describe the feature relevance, but integrate model detection and feature selection under the Bayesian framework. Dash and Liu propose a hybrid approach in [15], which uses entropy measure to rank the importance of features, and then use k-means to decide the final feature subset. More recently, Raftery and Dean [16] recast the problem of comparing two nested subset of variables as a model comparison problem, and solve it using approximate Bayes factors. A greedy add-and-remove algorithm is used to find the local optimum in the model space.

All the aforementioned algorithms select features globally, which may be unsuitable for some clustering problems as described in Section I. In these cases, feature subsets associated to individual clusters are more useful and can provide us a better understanding of the data. In bipartite graph partitioning [17], [18], features are grouped together with patterns in each cluster. However, features are divided exclusively, which prevents the possibility of a feature being relevant to more than one cluster. Other approaches along this direction, usually referred as *subspace clustering* [19]–[21], aim to seek high density areas embedded in a high dimensional feature space. CLIQUE [22] combines density analysis and grid-based clustering to find low dimensional clusters embedded in the high dimensional space. However, it requires the grid size and the density threshold as input parameters. PROCLUS [23] samples dataset, and then selects k clusters and repeatedly improves the clustering. In PROCLUS, the average dimensionality of subspaces is required as an input, which is difficult to be determined *a priori*. SURFING [21] tries to rank feature subsets based on a density-like measure. Clustering is performed subsequently on the top-ranked feature subsets. Greedy forward search is adopted to navigate the possible subspaces. More recently, COSA [24] assigns weights to each dimension for each instance based on the dispersion of its k nearest neighbors (knn), yielding a distance matrix from the weighted inverse exponential distance. The distance matrix is then processed by distance-based clustering methods, i.e., hierarchical clustering. After clustering, the overall importance value for each dimension of each cluster is calculated. Again, COSA requires a parameter, λ, which controls the strength of incentive for clustering on more dimensions, together with the number of clusters. In [25], the Parsimonious Model with Gaussian Mixtures (PMGM) is proposed, attempting to find component-specific feature space via shared distributions. PMGM encodes hard saliency in the EM-based optimization, and provides model selection using BIC.

In this paper, we propose a localized feature saliency measure and integrate it into the Bayesian inference framework. Our approach can discover clusters embedded in local feature subspaces. Note that, the goal of our method fundamentally differs from that of traditional subspace clustering. In subspace clustering, subspaces which contain high density areas are usually detected first, and then the embedded clusters are discovered. Our approach, on the other hand, focuses on detecting clusters and their individual feature subsets simultaneously through variational Bayesian learning. No prior knowledge is required in the proposed method. The soft saliency encoded in our model can be used for both feature selection and feature evaluation.

III. Mixture Model and Localized Feature Saliency

A. Mixture Model

From a *model-based* perspective, each cluster can be mathematically represented by a parametric distribution. One of the most widely used distributions is the Gaussian (Normal) distribution. In statistical pattern recognition, the sampling distribution of the sample mean is approximately normal, even if the distribution of the population, from which the sample is taken, is not normal. In addition, the normal distribution maximizes information entropy among all distributions with known mean and variance, which makes it natural choice of the underlying distribution for data summarized in terms of sample mean and variance. If we model the entire dataset by a mixture of Gaussians, the clustering problem is reduced to a problem of estimating the parameters of the Gaussian mixture.

A finite mixture of densities with K components is represented by $p(y) = \sum_{j=1}^{K} \pi_j p(y|\theta_j)$, where π_j are called mixing coefficients, and θ_j are the parameters corresponding to component j. We use π to denote the set $\{\pi_j\}_{j=1,\ldots,K}$, and similarly for $\theta \equiv \{\theta_j\}_{j=1,\ldots,K}$. Considering an observed dataset $\mathcal{Y} \equiv \{y_i|i = 1,\ldots,N\}$, where data points y_i are drawn from the mixture distribution independently, we introduce a set of latent variables $z_{ij} \in \{0, 1\}$ such that $z_{ij} = 1$ indicates that a given data point y_i is drawn from component j, and $z_{ij} = 0$ otherwise. Conditional on $Z = \{z_{ij}\}$ and θ, the likelihood is given by,

$$P(\mathcal{Y}|\theta, Z) = \prod_{i=1}^{N}\prod_{j=1}^{K} p(y_i|\theta_j)^{z_{ij}}. \tag{1}$$

B. Localized Feature Saliency

We assume that features are conditionally independent, and the importance of a feature is different for different clusters. The feature relevance is represented by a matrix $S = \{s_{jl}\}_{K\times D}$, where $s_{jl} = 1$ indicates that feature l is associated with component j, and $s_{jl} = 0$ otherwise. Let $\rho_{jl} = \Pr(s_{jl} = 1)$ be the probability that feature l is relevant to component j. Motivated by [11], the likelihood for Gaussian mixtures with localized feature saliency can be obtained through the following proposition.

Proposition 1: Let $p(\cdot|\theta_{jl})$ represent the distribution of a salient feature l for the component j, and $q(\cdot|\lambda_{jl})$ be the distribution if feature l is non-salient to j. Assuming that the features are conditionally independent, the likelihood function can be written as,

$$p(\mathcal{Y}|\theta) = \prod_{i=1}^{N}\sum_{j=1}^{K}\pi_j \prod_{l=1}^{D}\big(\rho_{jl}p(y_{il}|\theta_{jl}) + (1-\rho_{jl})q(y_{il}|\lambda_{jl})\big), \tag{2}$$

where $\theta = \{\{\pi_j\},\{\theta_{jl}\},\{\rho_{jl}\},\{\lambda_{jl}\}\}$ is the set of all the parameters. The proof is provided in the Appendix.

The mixture components with localized feature saliency can be interpreted as follows. Assume that samples $\mathcal{Y}_j$ are clustered to component j with a feature association indicator vector $(s_{jl})|_{l=1,\ldots,D}$. The distribution of $\mathcal{Y}_j$ has significant cluster structure (strong peak) on feature subset $\mathcal{F}_+$ in which $s_{jl} = 1$, while the distribution on feature subset $\mathcal{F}_-$, in which $s_{jl} = 0$, lacks such a cluster structure. The probability ρ_{jl} indicates the weighting of the l-th feature on the j-th cluster. By maximizing the overall likelihood, the model produces clusters embedded in different feature subsets. Moreover, our formulation can be further explained under the framework of generative models. We first create the component pool. Given component j, each pattern is generated in a feature-by-feature manner. Specifically, the value of the l-th feature is drawn from the distribution $p(\cdot|\theta_{jl})$ (salient features) or from the distribution $q(\cdot|\lambda_{jl})$ (non-salient features), based on the result of tossing a coin with the bias of ρ_{jl} on the head. The dataset is thereby created by sampling N patterns independently from the component pool with a priori π_j.

IV. Localized Feature Selection with Variational Learning

The parameters of the above mixture model can be estimated by Maximal Likelihood (ML) with EM, or by Variational Learning of Bayesian approximation (VB). ML method treats the parameters as unknown but fixed, while VB places a prior probability on the parameters. These two algorithms usually produce identical results in many cases [26]. However, in order to integrate cluster number estimation, ML method usually requires other criteria, such as Entropy measure or Minimal Message Length (MML). For VB approach, this process can be implemented through a proper choice of prior probability over mixing coefficients. Another problem encountered in ML is that singular components lead to infinite likelihood, which does not happen in VB. Here, we present the Variational Bayesian approach to approximate the parameters in the model presented in Section III.

A. Variational Approximation

In general, to evaluate the likelihood of mixtures, conditioned on the mixing coefficients, we must marginalize the parameters as follows,

$$P(\mathcal{Y}|\theta) = \int P(\mathcal{Y}, \Theta|\theta)\, d\Theta, \tag{3}$$

where $\Theta \equiv \{\theta, z, Z, S\}$ denotes all the parameters and latent variables. The integral sign represents the joint integral over θ and the summation over z and S. This integral is analytically intractable. Therefore, we use variational methods to find a lower bound for $P(\mathcal{Y}|\pi)$.

Consider the following transformation applied to the log marginal likelihood,

$$\ln P(\mathcal{Y}|\theta) \geq \int Q(\Theta)\ln\frac{P(\mathcal{Y},\Theta|\theta)}{Q(\Theta)}\, d\Theta = \mathcal{L}(Q). \tag{4}$$

The function $\mathcal{L}(Q)$ forms a rigorous lower bound on the true log marginal likelihood. Through a suitable choice of the Q distribution, the quantity $\mathcal{L}(Q)$ may be tractable to compute. From Equation (4), the difference between the true log likelihood $\ln P(\mathcal{Y}|\pi)$ and the bound $\mathcal{L}(Q)$ is given by Kullback-Leibler divergence $\mathrm{KL}(Q||P)$. $Q(\Theta)$ is chosen from some family of distributions such that the lower bound $\mathcal{L}(Q)$ is sufficiently simplified for evaluation. Since the true log likelihood is independent of Q, we approximate $P(\Theta)$ with $Q(\Theta)$ by minimizing the KL divergence. Assuming that $Q(\Theta)$ factorizes over subsets $\{\Theta_i\}$ of the variables in Θ, $Q(\Theta) = \prod_i Q_i(\Theta_i)$, the KL divergence can then be minimized over all possible factorial distributions by performing free-form minimization over Q_i,

$$Q_i(\Theta_i) = \frac{\exp\langle\ln P(\mathcal{Y},\Theta)\rangle_{k\neq i}}{\int \exp\langle\ln P(\mathcal{Y},\Theta)\rangle_{k\neq i}\, d\Theta_i}, \tag{5}$$

where $\langle\cdot\rangle_{k\neq i}$ denotes an expectation with respect to the distributions $Q_k(\Theta_k)$ for all $k \neq i$. Equation (5) shows that the sufficient statistics of each distribution Q_i depends on the moments of other distributions $Q_{k\neq i}$, which implies an iterative solution for the estimation of the variational variables. In other

words, with a sufficient parameter initialization, the statistics can be updated by taking each factor in turn and replacing its sufficient statistics with the revised estimates. In each iteration of the re-estimation process, the KL divergence is reduced, while both the lower bound, $\mathcal{L}(Q)$, and the likelihood are increased. Hence, the convergence is guaranteed.

B. Local feature saliency with variational learning

We now apply Bayesian variational approach to the mixture of Gaussians with localized feature saliency. Given the sets of hidden variables $Z = \{z_j^{(i)}\}$ and $S = \{s_{jl}^{(i)}\}$, the distribution of the Gaussian mixture is

$$p(\mathcal{Y}|Z,S,\theta) = \prod_{i=1}^{N}\prod_{j=1}^{K}\Big[\prod_{l=1}^{D}\big(p(y_{il}|\theta_{jl})\big)^{s_{jl}^{(i)}}\big(q(y_{il}|\lambda_{jl})\big)^{1-s_{jl}^{(i)}}\Big]^{z_j^{(i)}}, \tag{6}$$

where $\mu = \{\mu_{jl}\}$ and $T = \{\tau_{jl}\}$ denote the means and inverse variances of the "useful" subcomponents, while $\epsilon = \{\epsilon_{jl}\}$ and $\gamma = \{\gamma_{jl}\}$ are the sets of parameters for the "noisy" subcomponents. The distribution of the hidden variable Z (given the mixing probabilities $\pi = \{\pi_j\}$) and the distribution of the hidden variable S (given the mixing probabilities $\rho = \{\rho_{jl}\}$) are governed as,

$$P(Z|\pi) = \prod_{i=1}^{N}\prod_{j=1}^{K}\pi_j^{z_{ij}}, \tag{7}$$

$$P(S|\rho) = \prod_{i=1}^{N}\prod_{j=1}^{K}\prod_{l=1}^{D}\rho_{jl}^{s_{jl}^{(i)}}(1-\rho_{jl})^{1-s_{jl}^{(i)}}. \tag{8}$$

In order to accomplish model selection, the above Bayesian model is augmented with conjugate priors over the means and inverse covariances,

$$P(\mu) = \prod_{j=1}^{K}\prod_{l=1}^{D}\mathcal{N}(\mu_{jl}|m_l,c), \tag{9}$$

$$P(T) = \prod_{j=1}^{K}\prod_{l=1}^{D}\Gamma(\tau_{jl}|\alpha,\beta), \tag{10}$$

where $\Gamma(\cdot)$ is the gamma distribution, m_l, c, α, and β are hyperparameters that control the prior distributions. The hyperparameters are chosen such that the prior distribution is broad enough to cover the whole dataset. Since the actual model parameters are represented by the means of the corresponding distributions, they are not sensitive to these hyperparameters. Particularly, we set m to the mean of the dataset, $c = \alpha = \beta = 10^{-7}$.

For conjugate hierarchical models, the expressions on the right side of Equation (5) will have the same functional forms as in the priors . Applying Equation (5) to the above Bayesian model and taking an iterative optimization, we can obtain (more details regarding the derivation can be found in the Appendix),

$$Q_Z(Z) = \prod_{i=1}^{N}\prod_{j=1}^{K} r_{ij}^{z_{ij}}, \tag{11}$$

$$Q_\mu(\mu) = \prod_{j=1}^{K}\prod_{l=1}^{D}\mathcal{N}(\mu_{jl}|m_{jl}^v, c_{jl}^v), \tag{12}$$

$$Q_T(T) = \prod_{j=1}^{K}\prod_{l=1}^{D}\Gamma(\tau_{jl}|\alpha_{jl}^v, \beta_{jl}^v), \tag{13}$$

$$Q_S(S) = \prod_{i=1}^{N}\prod_{j=1}^{K}\prod_{l=1}^{D}\omega_{ijl}^{s_{jl}^{(i)}}(1-\omega_{ijl})^{1-s_{jl}^{(i)}}, \tag{14}$$

where $r_{ij}, m_{jl}^v, c_{jl}^v, \alpha_{jl}^v, \beta_{jl}^v$, and ω_{ijl} are variational parameters for maximization and determining the density involved in Q, defined by

$$r_{ij} = \frac{\pi_j \tilde{r}_{ij}}{\sum_{j=1}^{K}\pi_j\tilde{r}_{ij}}, \tag{15}$$

$$\tilde{r}_{ij} = \exp\{\frac{1}{2}\sum_{l=1}^{d}\omega_{ijl}[\psi(\alpha_{jl}^v) - \log\beta_{jl}^v - \frac{\alpha_{jl}^v}{\beta_{jl}^v}((y_l^i - m_{jl}^v)^2 + \frac{1}{c_{jl}^v})]\}, \tag{16}$$

$$m_{jl}^v = \frac{cm_i + (\alpha_{jl}^v/\beta_{jl}^v)\sum_{i=1}^{n} r_{ij}\omega_{ijl}y_l^i}{c + (\alpha_{jl}^v/\beta_{jl}^v)\sum_{i=1}^{n} r_{ij}\omega_{ijl}}, \tag{17}$$

$$c_{jl}^v = c + \frac{\alpha_{jl}^v}{\beta_{jl}^v}\sum_{i=1}^{N} r_{ij}\omega_{ijl}, \tag{18}$$

$$\alpha_{jl}^v = \alpha + \frac{1}{2}\sum_{i=1}^{N} r_{ij}\omega_{ijl}, \tag{19}$$

$$\beta_{jl}^v = \beta + \frac{1}{2}\sum_{i=1}^{N} r_{ij}\omega_{ijl}[(y_l^i - m_{jl}^v)^2 + \frac{1}{c_{jl}^v}], \tag{20}$$

$$\omega_{ijl} = \frac{\rho_{jl}\tilde{\omega}_{ijl}}{\rho_{jl}\tilde{\omega}_{ijl}} + (1-\rho_{jl}), \tag{21}$$

$$\tilde{\omega}_{ijl} = \exp\{\frac{1}{2}r_{ij}[\psi(\alpha_{jl}^v) - \log\beta_{jl}^v - \frac{\alpha_{jl}^v}{\beta_{jl}^v}((y_l^i - m_{jl}^v)^2 + \frac{1}{c_{jl}^v})]\}, \tag{22}$$

$$\xi_{ijl} = \exp\{-\frac{1}{2}\gamma_{jl}(y_l^i - \epsilon_{jl})^2 + \frac{1}{2}\log\gamma_{jl}\}, \tag{23}$$

where $\psi(x)$ is the *digamma* function $\psi(x) = d\log\Gamma(x)/dx$.

The model parameters $\pi_j, \rho_{jl}, \epsilon_{jl}$, and γ_{jl} are given by the mean values of corresponding variational factors:

$$\pi_j = \frac{1}{N}\sum_{i=1}^{N} r_{ij}, \tag{24}$$

$$\rho_{jl} = \frac{1}{N}\sum_{i=1}^{N}\omega_{ijl}, \tag{25}$$

$$\epsilon_{jl} = \frac{\sum_{i=1}^{N}\omega_{ijl}y_l^i}{\sum_{i=1}^{N}\omega_{ijl}}, \tag{26}$$

$$\frac{1}{\gamma_{jl}} = \frac{\sum_{i=1}^{N}\omega_{ijl}(y_l^i - \epsilon_{jl})^2}{\sum_{i=1}^{N}\omega_{ijl}}. \tag{27}$$

The above steps iterate alternatively until convergence. This model has a property that the components with similar parameters fitting the same Gaussian will compete with each other, yielding a dominant cluster. Thus, we can initialize the model with a large number of clusters, and eliminate the trivial clusters during iteration. Finally, the algorithm will produce a model with localized feature saliency represented by ρ_{jl} and identify the number of clusters simultaneously.

One should notice that seeking the feature saliency for individual clusters introduces more parameters than global feature selection approaches. Consequently, this increases the potential risks posed by local extrema. To this end, variational learning is a better choice for the optimization than EM. Unlike EM, which assumes an unknown but fixed value for a parameter, variational learning formulates the model parameters as distributions. The variational parameters are initialized based on broad distributions. In addition, the estimated model parameters are represented by the means of the corresponding approximation functions. Therefore, variational learning can provide robust and stable optimization results, and can also alleviate the overfitting problem, often suffered by EM.

C. Computational Complexity

The computational complexity of the proposed algorithm is $\mathcal{O}(NDK)$ in each iteration. The total computational time depends on the number of iterations required for converging. Specifically, in each iteration, we have to compute measures in Equations (15)-(27). Computing ξ_{ijl} is $\mathcal{O}(1)$. There are (NDK) ξs so that it requires $\mathcal{O}(NDK)$. Similarly, ω and $\tilde{\omega}$ require $\mathcal{O}(NDK)$. Computing α_{jl}^v requires to navigate through all the samples, resulting in the complexity $\mathcal{O}(NDK)$. Similar results can be obtained for β^v, c^v, m^v, and $\tilde{r}$. For r, the complexity is $\mathcal{O}(NK)$, since the summation of Equation (15) can be re-used. The complexity for ρ, ϵ, and γ is $\mathcal{O}(NDK)$. For p, it is $\mathcal{O}(NK)$. In summary, the overall computational complexity for one iteration is $\mathcal{O}(NDK)$.

D. Advantages of the proposed approach

The proposed method integrates localized feature selection, model detection and clustering into a unified framework. Its major advantages are summarized as follows,

1) Compared with global methods, our method can reveal cluster-wise feature relevance, hence providing users more accurate information about the underlying model which generates the data.
2) Compared with subspace clustering methods, our method does not require users to provide values of the parameters that are critical but almost impossible to be set in advance, for example, the number of clusters, the density threshold, or the desired dimensionality.
3) Our method avoids heuristical navigation through the large pool of possible feature subsets. The computational cost for each iteration of the proposed algorithm is $\mathcal{O}(NDK)$. It does not grow exponentially with D or N. Therefore, our method is scalable to large datasets.

V. Experimental Results

In general, the performance of an unsupervised feature selection algorithm is difficult to be evaluated. Localized feature selection makes it even more difficult as we have an additional layer of complexity brought by the association of clusters with different feature subsets. To thoroughly evaluate the proposed Localized Feature Selection with Variational Bayesian (LFSVB) algorithm, we have compared it with the leading unsupervised feature selection methods on both synthetic and real-world datasets. Specifically, in the comparison, we choose a global method proposed in [14], which is also based on the Bayesian framework with variational learning (GFSVB). In addition, we have selected a recently published and well-accepted subspace clustering method, viz., COSA [24]. Unlike other subspace clustering approaches that usually yield only hard-decisions (either accept or reject a feature), COSA can produce soft feature saliency (feature importance), similar to our approach, and thus make the comparison more meaningful. Note that COSA software is publicly available [1]. Finally, we also compare our approach with the parsimonious model with Gaussian mixtures (PMGM) [25].

A. Synthetic Data

1) Synthetic datasets with hard feature saliency: First, we have applied the four algorithms (LFSVB, GFSVB, COSA and PMGM) to 100 synthetic datasets with 0-1 (hard) feature saliency (a feature is either relevant or irrelevant). As we know the underlying model from which the patterns were sampled, the performance of an algorithm is assessed through whether the algorithm can find the given model. The synthetic datasets are created by a data generator. It first generates c Gaussian components $\mathcal{N}(\mu_{\mathbf{j}}, \mathbf{\Sigma_j})$, $j = 1, \cdots, c$, separately, where $\mathbf{\Sigma_j}$ is restricted to a diagonal matrix. The values of $\mu_{\mathbf{j}}$ are chosen randomly from -4 to 4 and from 0.1 to 0.3 for Σ_j. Components can have different numbers of features D_j, and different numbers of patterns N_j. Those Gaussians are then embedded into subsets of a D-dimensional background with Gaussian noise $\mathcal{N}(\mathbf{0}, \mathbf{I})$. Specifically, we randomly select D_1 features from the background data, and replace the first N_1 positions with component 1. This embeds the first component into the background. Similarly, we can embed all the rest clusters into the background. Finally, a D-dimensional dataset consisting of c Gaussian mixtures, with each component corresponding to an individual relevant feature subset, is generated. The total number of patterns is $N = \sum_{j=1}^{c} N_j$. In our experiment, one hundred datasets are generated with dimensionality (D) varying from 10 to 200, the number of salient features (D_j) from 1 to 8, the cluster size (N_j) from 100 to 500, and the number of clusters from 3 to 7.

We initialize LFSVB with $k = 20$. The global feature selection approach is initialized in the same manner. COSA is initialized with default settings. COSA-distance matrix is computed, and then processed by hierarchal clustering. Clusters are manually selected based on the visual inspection of the dendrogram. Feature importance is normalized so that the value of the top-rank-feature is 1.

[1] http://www-stat.stanford.edu/~jhf/COSA.html

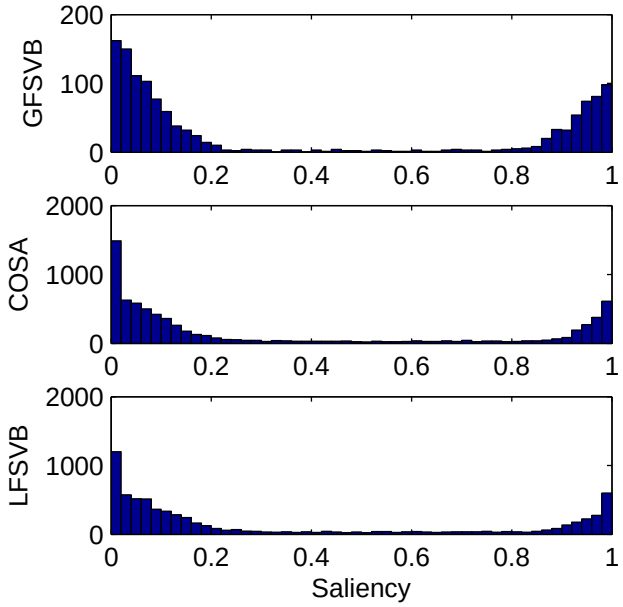


Fig. 2: Histograms of feature saliency on 100 synthetic datasets for GFSVB (uper panel), COSA (middle panel), and LFSVB (lower panel), respectively.

Note that, PMGM produces binary feature weights (either 0 or 1), while the other three algorithms yield feature weight in the range of [0,1]. To evaluate the performance of the algorithms for feature selection, we need to find a cut-off threshold of feature saliency for LFSVB, GFSVB, and COSA. Figure 2 shows the histograms of the feature saliency obtained by GFSVB, COSA, and LFSVB, respectively, for all the clusters in the 100 datasets. Clearly, the saliency values are mainly distributed in the range of [0,0.2] and [0.8,1]. In the following experiments, we simply choose 0.5 as the cut-off threshold for the three algorithms.

We compute four quantities to evaluate the performance of the algorithms: (1) accuracy of cluster number $ACN = \frac{|\hat{c}-c|}{c}$, where $\hat{c}$ is the estimated number of clusters and c is the true value; (2) clustering accuracy $CA = 1 - \frac{\tilde{N}}{N}$, where $\tilde{N}$ is the number of mis-clustered samples; (3) feature precision $FP_j = \frac{|\hat{D}_j \bigcap D_j|}{|\hat{D}_j \bigcup D_j|}$, where $\hat{D}_j$ and D_j are the estimated and true feature subset for cluster j, respectively, and $|\cdot|$ represents the set length; and (4) feature recall $FR_j = \frac{|\hat{D}_j \bigcap D_j|}{|D_j|}$. The statistical summary over the 100 synthetic datasets are reported in Table I, while an example is provided in Table II, showing the results for the synthetic dataset (syn_0) with 30 features and 3 clusters.

Compare to global feature selection. From the example in Table II, we can see clearly that the proposed algorithm correctly detects the underlying clusters as well as the feature subsets corresponding to each cluster. On the other hand, GFSVB yields a feature subset which is close to the union of feature subsets identified by LFSVB, except that feature 19 is missing. Table I shows that, over the 100 synthetic datasets, LFSVB yields higher accuracy than GFSVB on cluster number estimation. The cluster accuracy of GFSVB is slightly higher than that of LFSVB. However, both the algorithms can discover the clusters very well. The feature recall measure of GFSVB is high, but the feature precision measure of GFSVB is low, while both values of LFSVB are high. This indicates that the global feature selection algorithm can detect if a feature is relevant to the dataset, however, it can not determine if a salient feature really plays a critical role on a particular cluster. On the other hand, the proposed model not only provides information on whether a feature is relevant or not, but also shows which cluster the feature is relevant or irrelevant to.

Compare to subspace clustering. As an example, Table II shows that localized feature subsets for $C1$ and $C2$ are correctly identified by COSA. It misses a salient feature (feature 26) for cluster 3, while LFSVB can recognize all three clusters with the corresponding feature subsets. The overall cluster accuracy of COSA is slightly better than that of LFSVB, while LFSVB outperforms COSA on feature precision and feature recall, as shown in Table I. Moreover, COSA only produces a COSA-distance matrix and requires other clustering algorithms for subsequent processing. The number of clusters is also required as an input. On the other hand, our method provides a fully-automated solution by integrating localized feature selection, model detection, and clustering into a unified framework.

Compare to parsimonious model with Gaussian mixture. The example results shown in Table II and the statistical measures shown in Table I indicate that the proposed algorithm performs equivalently to PMGM. Notice that PMGM yields binary feature weight (either 0 or 1), while our algorithm produces feature saliency as a probability measure in the range of [0,1]. Subsequently, the proposed method can be applied for both feature selection and feature evaluation.

2) Synthetic dataset with soft feature saliency: The feature saliency in real-world datasets is usually soft, which means that it can be any value between 0 or 1. To approximate this situation, we generate a 20-dimensional dataset where the feature saliency is distributed in the range of $[0, 1]$. This dataset contains 2 Gaussian components $\mathcal{N}(\mu_{\mathbf{1}}, \mathbf{\Sigma_1})$ and $\mathcal{N}(\mu_{\mathbf{2}}, \mathbf{\Sigma_2})$, where $\mu_{\mathbf{1}} = (0.5, \ldots, 0.5)$, $\mu_{\mathbf{2}} = (-0.5, \ldots, -0.5)$, $\mathbf{\Sigma_1}$ and $\mathbf{\Sigma_2}$ are both diagonal, having $(0.2, \ldots, 0.2)$ on the diagonal terms. The feature saliency of clusters 1 and 2 are $(0.05, 0.10, \ldots, 1)$ and $(1, 0.95, \ldots, 0.05)$, respectively. Each component contains 500 points. We generate the data based on the procedure described in Section III with a common distribution of $\mathcal{N}(\mathbf{0}, \mathbf{I})$.

We run the four algorithms on this dataset 10 times. Both LFSVB and PMGM detect two clusters successfully, while GFSVB yields 3 clusters. For COSA, we manually select the clusters. Table III shows the feature saliency obtained by LFSVB, COSA, PMGM, and GFSVB, respectively. We can see that GFSVB determines that all feature saliency is greater than 0.5. PMGM can discover that the two clusters have different relevant feature subsets. However, it does not obtain the true feature saliency due to its binary coding scheme. On the other hand, LFSVB and COSA correctly discover that feature relevance associated to cluster 1 is different from that of cluster 2. Specifically, the relevance of features increases

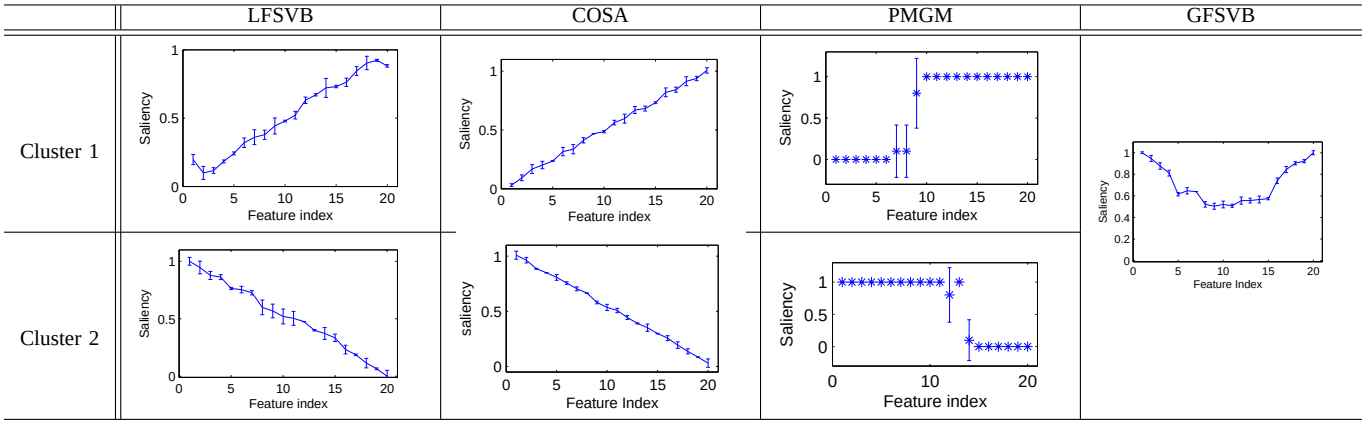
TABLE I: Statistical summary on 100 synthetic datasets, where $\overline{ACN}$ is the average accuracy of cluster number, $\overline{CA}$ is the average clustering accuracy, $\overline{FP_j}$ is the average feature precision, and $\overline{FR_j}$ is the average feature recall. For COSA, the number of clusters $(\hat{c})$ is set manually with visual inspection of the dendrogram (denoted by *).

algorithm	$\overline{ACN}$	$\overline{CA}$	$\overline{FP_j}$	$\overline{FR_j}$
GFSVB	0.952 (0.015)	0.922 (0.020)	0.384 (0.086)	0.941 (0.022)
COSA	0.992* (0.03)	0.933 (0.011)	0.892 (0.015)	0.897 (0.021)
LFSVB	0.980 (0.017)	0.910 (0.023)	0.925 (0.017)	0.950 (0.025)
PMGM	0.983 (0.017)	0.914 (0.026)	0.920 (0.022)	0.945 (0.018)

TABLE II: Experimental results on synthetic dataset (syn_0) with hard feature saliency. For COSA, the number of clusters $(\hat{c})$ is set manually with visual inspection of the dendrogram (denoted by *). *Truth* indicates the actual model which generates the dataset. C1,C2, and C3 represent different clusters.

Data	Algo.	$\hat{c}$	accuracy	Feature subset
syn_0 $D = 30$	Truth	3	-	C1:[8, 19, 30], C2:[5, 23, 24], C3:[7, 16, 26]
	LFSVB	3	99.2%	C1:[8, 19, 30], C2:[5, 23, 24], C3:[7, 16, 26]
	COSA	3*	98.5%	C1:[8, 19, 30], C2:[5, 23, 24], C3:[7, 16]
	GFSVB	3	98.3%	[5, 8, 16, 23, 24, 26, 30]
	PMGM	3	99.0%	C1:[8, 19, 30], C2:[5, 23, 24], C3:[7, 16, 26]

TABLE III: Average feature saliency on the synthetic dataset with soft feature saliency. The feature saliency is in a decreasing order for cluster 1, and in a increasing order for cluster 2.



with feature index for cluster 1, but decreases for cluster 2. This provides additional and more accurate information than GFSVB and PMGM.

B. Real-world datasets

For the evaluation on real-world datasets, we utilized six datasets: *Heart*, *Ion*, *Vehicle*, *Wine*, *WDBC*, and *Yeast*, from the UCI machine learning repository [27], with varying number of features, patterns, and categories, as summarized in Table IV. Class labels are provided in the datasets for supervised learning, which are excluded during the clustering process. A confusion matrix is computed according to the true class labels and the cluster index. Based on confusion matrix, mutual information is calculated as

$$I(\mathcal{X};\mathcal{Y}) = -\sum_{\mathcal{X}}\sum_{\mathcal{Y}} p(x,y)\log\frac{p(x,y)}{p(x)p(y)}, \qquad (28)$$

where x and y are true labels and cluster index respectively, $p(x,y)$ is the joint probability, and $p(x)$ and $p(y)$ are the marginal probability distribution of $\mathcal{X}$ and $\mathcal{Y}$, respectively. The mutual information measures the dependence between $\mathcal{X}$ and $\mathcal{Y}$. Thus, a higher value of I indicates that the clustering results are closer to the true class group.

TABLE IV: Summary of the UCI datasets, where N is the number of samples, D the number of attributes, and c the number of classes.

data	Description	N	D	c
Heart	Heart Disease of Statlog	270	13	2
Ion	Ionosphere Database	351	34	2
Vehicle	vehicle classification	846	18	4
Wine	wine recognition	178	13	3
WDBC	Diagnostic breast cancer	569	30	2
Yeast	Protein Localization Sites	1484	8	10

Table V shows the mean and standard deviation of the cluster numbers and mutual information over 10 runs of the four algorithms. Again, cluster numbers for COSA are set manually based on the dendrogram. On the average mutual information, LFSVB outperforms GFSVB on five (out of six) datasets (Heart, Ion, Vehicle, Wine, and Yeast). On WDBC, it is as good as GFSVB. LFSVB also outperforms COSA on five (out of six) datasets (Ion, Vehicle, Wine, WDBC and Yeast). The proposed algorithm outperforms PMGM on two datasets

TABLE V: Mutual information I and the estimated cluster number $\hat{c}$, represented by mean and standard deviation over 10 different runs, on UCI datasets. For COSA, the number of clusters is determined manually (denoted by *).

Data	Algo	$\hat{c}$(std)	I(std)
Heart	LFSVB	2.8(0.8)	0.15(0.07)
	COSA	2*	**0.21**(0.01)
	GFSVB	3.0(0.7)	0.09(0.06)
	PMGM	3.1(0.6)	0.11 (0.05)
Ion	LFSVB	3.8(1.1)	**0.33**(0.1)
	COSA	4*	0.30(0.01)
	GFSVB	3.4(0.9)	0.21(0.05)
	PMGM	3.3(0.8)	0.31 (0.05)
Vehicle	LFSVB	9.9(1.7)	**0.63**(0.05)
	COSA	9*	0.48(0.01)
	GFSVB	10.5(1.5)	0.58(0.09)
	PMGM	9.5(1.6)	0.60 (0.04)
Wine	LFSVB	3.1(0.3)	**1.44**(0.07)
	COSA	3*	1.26(0.01)
	GFSVB	3.4(0.7)	1.42(0.06)
	PMGM	3.2(0.6)	1.42 (0.07)
WDBC	LFSVB	6.3(0.8)	**0.68**(0.02)
	COSA	10*	0.59(0.01)
	GFSVB	7.6 (0.9)	0.67(0.02)
	PMGM	8.1(0.6)	**0.68** (0.03)
Yeast	LFSVB	11.4(2.1)	**0.39**(0.06)
	COSA	13*	0.15(0.02)
	GFSVB	6.8(0.8)	0.36(0.01)
	PMGM	8.2(1.5)	0.38(0.05)

(Heart, Vehicle). On the other datasets, those two algorithms have similar performance.

LFSVB shows that different relevant feature subsets are associated with different clusters, whose sizes are usually smaller than the global relevant feature subset. PMGM also selects a feature subset for each component. The difference between LFSVB and PMGM is that LFSVB evaluates the relevance of a feature with a saliency value in the range of [0, 1] while PMGM uses a binary one. In addition, model detection is fully integrated in LFSVB through variational learning. A separate measure such as BIC is not required.

VI. Conclusion

In this paper, we have proposed a novel approach of simultaneous localized feature selection and model detection for unsupervised learning. Our approach provides a fully-automated solution to identify useful patterns embedded in feature subspaces by integrating local feature selection, model detection, and clustering into a unified Bayesian framework. We have demonstrated the advantages of our algorithm over global feature selection and subspace clustering methods on both synthetic and real-world datasets.

Appendix

(1) Likelihood for Gaussian mixtures with localized feature saliency

Proof: Let $S = \{s_{jl}^{(i)}\}$, and $Z = \{z_j^{(i)}\}$ represent the hidden variables. The dataset is assumed to be drawn from the Gaussian distribution

$$p(\mathcal{Y}|Z,S,\theta) = \prod_{i=1}^{N}\prod_{j=1}^{K}\Big[\prod_{l=1}^{D}\big(p(y_{il}|\theta_{jl})\big)^{s_{jl}^{(i)}}\big(q(y_{il}|\lambda_{jl})\big)^{1-s_{jl}^{(i)}}\Big]^{z_j^{(i)}}. \tag{29}$$

(30)

Given mixing probabilities $\pi = \{\pi_j\}$ and $\rho = \{\rho_{jl}\}$, the distributions of hidden variable Z and S are

$$P(Z|\pi) = \prod_{i=1}^{N}\prod_{j=1}^{K}\pi_j^{z_j^{(i)}}, \tag{31}$$

$$P(S|\rho) = \prod_{i=1}^{N}\prod_{j=1}^{K}\prod_{l=1}^{D}\rho_{jl}^{s_{jl}^{(i)}}(1-\rho_{jl})^{1-s_{jl}^{(i)}}. \tag{32}$$

The joint distribution $p(\mathcal{Y}, Z, S|\theta)$ is

$$\begin{aligned} p(\mathcal{Y},Z,S|\theta) &= p(\mathcal{Y}|\theta)P(Z|\pi)P(S|\rho) \\ &= \prod_{i=1}^{N}\prod_{j=1}^{K}\Big[\prod_{l=1}^{D}\pi_j\big(\rho_{jl}p(y_{il}|\theta_{jl})\big)^{s_{jl}^{(i)}}\big((1-\rho_{jl})q(y_{il}|\lambda_{jl})\big)^{1-s_{jl}}\Big]^{z_j^{(i)}}. \end{aligned} \tag{33}$$

(34)

The likelihood is obtained by margining hidden variables Z and S from the above equation. Notice that $z_j^{(i)} \in \{0,1\}$ and $s_{jl}^{(i)} \in \{0,1\}$, we have

$$p(\mathcal{Y}|\theta) = \prod_{i=1}^{N}\sum_{j=1}^{K}\pi_j\prod_{l=1}^{D}\big(\rho_{jl}p(y_{il}|\theta_{jl}) + (1-\rho_{jl})q(y_{il}|\lambda_{jl})\big). \tag{35}$$

■

(2) Derivation of variational parameter update

We factorize $Q(\Theta)$ as

$$Q(\Theta) = Q_Z(Z)Q_S(S)Q_\mu(\mu)Q_T(T). \tag{36}$$

Let us consider the derivation of the update equation for the factor $Q(Z)$ by applying Equation (5) and taking logarithm on both sides,

$$\ln Q_z(Z) = \langle \ln p(\mathcal{Y},\mathcal{S},\theta)\rangle + const., \tag{37}$$

where $\langle\cdot\rangle$ represents the expectation of S and θ. Absorbing any terms that do not depend on Z into the additional normalization constant, we have

$$\ln Q_z(Z) = \langle \ln p(Z|\pi)\rangle_\pi + \langle \ln p(\mathcal{Y}|Z,S,\theta)\rangle_\theta + const. \tag{38}$$

Substituting the two terms on the right side, and absorbing any term that are independent of Z, we get

$$\ln Q_Z(Z) = \sum_{i=1}^{N}\sum_{j=1}^{K} z_{ij}\ln\tilde{r}_{ij} + const., \tag{39}$$

where $\tilde{r}_{ij}$ has the form of Equation (16). Note that for each value of i, the quantities $\langle z_{ij}\rangle$ are binary and sum to 1. $Q_Z(Z)$ can be normalized to

$$Q_Z(Z) = \prod_{i=1}^{N}\prod_{j=1}^{K} r_{ij}^{z_{ij}}, \tag{40}$$

which has the form of Equation (15).

Similarly, applying Equation (14) to Equation (5), we have

$$\begin{aligned}\ln Q_S(S) =& \langle \ln p(S|\rho)\rangle_\rho + \langle \ln p((Y|Z,S,\theta)\rangle_\theta + cons \\ =& \sum_{i=1}^{N}\sum_{j=1}^{K}\sum_{l=1}^{D}\{s_{jl}^{(i)} \ln \tilde{\omega}_{ijl} + \\ & (1-s_{jl}^{(i)}) \ln(1-\tilde{\omega}_{ijl})\} + const., \end{aligned} \tag{41}$$

where $\tilde{\omega}_{ijl}$ has the form of Equation (22). Thus, $Q_S(S)$ has the form of

$$Q_S(S) \propto \prod_{i=1}^{N}\prod_{j=1}^{K}\prod_{l=1}^{D} \tilde{\omega}_{ijl}^{s_{jl}^{(i)}} (1-\omega_{ijl})^{1-s_{jl}^{(i)}}. \tag{42}$$

Normalization $Q_S(S)$ yields to Equation (21).

Applying Equation (5) to $Q_\mu(\mu)$,

$$\begin{aligned}\ln Q_\mu(\mu) =& \sum_{j=1}^{K}\sum_{l=1}^{D} \ln p(\mu_{jl}) + \langle \ln p(Z|\pi)\rangle_Z + \\ & \sum_{i=1}^{N}\sum_{j=1}^{K}\sum_{l=1}^{D} \langle z_j^{(i)}\rangle [s_{jl}^{(i)} \ln p(y_{ijl}|\mu_{jl},\theta_{jl}) + \\ & (1-s_{jl}^{(i)}) \ln p(y_{ijl}|\lambda_{jl})] + const. \end{aligned} \tag{43}$$

This leads to a Gaussian distribution

$$Q_\mu(\mu) = \prod_{j=1}^{K}\prod_{l=1}^{D} \mathcal{N}(\mu|m_{jl}^v, c_{jl}^v), \tag{44}$$

where m_{jl}^v and c_{jl}^v have the form of Equations (17) and (18), respectively.

For further details on the derivation of variational learning, readers may refer to [28].

ACKNOWLEDGMENT

This research was partially funded by National Science Foundation under grants: IIS-0713315 and CNS-0751045, and by the 21st Century Jobs Fund Award, State of Michigan, under grant: 06-1-P1-0193.

REFERENCES

[1] A. Jain and D. Zongker, "Feature selection: Evaluation, application, and small sample performance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 2, pp. 153–158, 1997.

[2] Y. Li, M. Dong and J. Hua, "Localized feature selection for clustering," *Pattern Recognition Letters*, vol. 29, pp. 10–18, 2008.

[3] S. Avidan, "Joint feature-basis subset selection," in *IEEE Transaction on Computer Vision and Pattern Recognition (CVPR'04)*, June 2004.

[4] Y. Wu and A. Zhang, "Feature selection for classifying high-dimensional numerical data," in *IEEE Transaction on Computer Vision and Pattern Recognition (CVPR'04)*, June 2004.

[5] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.

[6] S. K. Singhi and H. Liu, "Feature subset selection bias for classification learning," in *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburgh, Pennsylvania, 2006, pp. 849–856.

[7] M. Dong and R. Kothari, "Feature subset selection using a new definition of classifiability," *Pattern Recognition Letters*, vol. 23, pp. 1215–1225, 2003.

[8] S. Chang, N. Dasgupta, and L. Carin, "A bayesian approach to unsupervised feature selection and density estimation using expectation propagation," in *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, vol. 2, June 2005, pp. 1043–1050.

[9] J. G. Dy and C. E. Brodley, "Feature selection for unsupervised learning." *Journal of Machine Learning Research*, vol. 5, pp. 845–889, 2004.

[10] P. Mitra, C. Murthy, and S. Pal, "Unsupervised feature selection using feature similarity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 301–312, 2002.

[11] M. H. Law, M. A. T. Figueiredo, and A. K. Jain, "Simultaneous feature selection and clustering using mixture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1154–1166, 2004.

[12] H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 4, pp. 491–502, 2005.

[13] M. Dash, K. Choi, P. Scheuermann, and H. Liu, "Feature selection for clustering - a filter solution," in *IEEE International Conference on Data Mining*, 2002, pp. 115–122.

[14] C. Constantinopoulos, M. K. Titsias, and A. Likas, "Bayesian feature and model selection for gaussian mixture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 6, pp. 1013–1018, 2006.

[15] M. Dash and H. Liu, "Feature selection for clustering," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2000, pp. 110–121.

[16] A. E. Raftery and N. Dean, "Variable selection for model-based clustering," *Journal of the American Statistical Association*, vol. 101, no. 473, pp. 168–178, 2006.

[17] H. Zha, X. He, C. Ding, M. Gu, and H. Simon, "Bipartite graph partitioning and data clustering," in *Proceedings of ACM CIKM 2001*, 11 2001, pp. 25–32.

[18] M. Rege, M. Dong, and F. Fotouhi, "Co-clustering documents and words using bipartite isoperimetric graph partitioning," in *IEEE International Conference on Data Mining (ICDM)*, Hong Kong, 2006.

[19] L. Parsons, E. Haque, and H. Liu, "Subspace clustering for high dimensional data: a review," *SIGKDD Explor. Newsl.*, vol. 6, no. 1, pp. 90–105, 2004.

[20] Q. Ke and T. Kanade, "Robust subspace clustering by combined use of knnd metric and svd algorithm," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2004)*. IEEE, June 2004, pp. 592–599.

[21] C. Baumgartner, C. Plant, K. Kailing, H.-P. Kriegel, and P. Kröger, "Subspace selection for clustering high-dimensional data," in *Proc. 4th IEEE int. Conf. on Data Ming (ICDM 04)*, 2004, pp. 11–18.

[22] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, "Automatic subspace clustering of high dimensional data for data mining applications," in *1998 ACM-SIGMOD Int. Conf. Management of Data*, Seattle, Washington, 1998, pp. 94–105.

[23] C. C. Aggarwal, C. M. Procopiuc, J. L. Wolf, P. S. Yu, and J. S. Park, "Fast algorithms for projected clustering," in *Proc. ACM-SIGMOD Intl. Conf. Management of Data*, 1999, pp. 61–72.

[24] J. H. Friedman and J. J. Meulman, "Clustering objects on subsets of attributes," *Journal of the Royal Statistical Society: Series B*, vol. 66, no. 4, pp. 815–849, 2004.

[25] M. W. Graham and D. J. Miller, "Unsupervised learning of parsimonious mixtures on large spaces with integrated feature and component selection," *IEEE Transactions on Signal Processing*, vol. 54, no. 4, pp. 1289–1303, 2006.

[26] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. John Willey & Sons, Inc, 2000.

[27] A. Asuncion and D. Newman, "UCI machine learning repository," 2007. [Online]. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html

[28] C. M. Bishop, *Pattern Recognition and Machine Learning*, 1st ed., ser. Information science and statistics. Springer, 2006, ch. 10.