A Unified Probabilistic Model for Global and Local Unsupervised Feature Selection

Yue Guan GUAN.Y@HUSKY.NEU.EDU
Jennifer G. Dy JDY@ECE.NEU.EDU
ECE Department, Northeastern University, Boston, MA 02115

Michael I. Jordan JORDAN@CS.BERKELEY.EDU
EECS and Statistics Departments, University of California, Berkeley, CA 94720

Abstract

Existing algorithms for joint clustering and feature selection can be categorized as either global or local approaches. *Global methods* select a single cluster-independent subset of features, whereas *local methods* select cluster-specific subsets of features. In this paper, we present a unified probabilistic model that can perform both global and local feature selection for clustering. Our approach is based on a hierarchical beta-Bernoulli prior combined with a Dirichlet process mixture model. We obtain global or local feature selection by adjusting the variance of the beta prior. We provide a variational inference algorithm for our model. In addition to simultaneously learning the clusters and features, this Bayesian formulation allows us to learn both the number of clusters and the number of features to retain. Experiments on synthetic and real data show that our unified model can find global and local features and cluster data as well as competing methods of each type.

1. Introduction

Clustering is the process of grouping objects together based on some notion of similarity. Similarity is typically defined by a metric or probabilistic model, which are a function of the features or attributes describing the data samples. In many cases, particularly in applications involving high dimensions, not all of the features are needed: some are irrelevant and some are redundant. Irrelevant features are noisy features that do not reveal cluster structures. Redundant features are features that do not add cluster information to an existing set of selected features. The goal of feature selection is to remove both irrelevant and redundant features in order to improve the performance, decrease the complexity and improve the interpretability of a learning algorithm.

One can reduce the dimensionality by either feature transformation, where the original features are transformed into a lower dimensional space, or by feature/variable selection, where one selects a subset of the original set of features. Feature selection is desired for applications where one wishes to know which of the original features are important. Feature selection also helps in avoiding having to collect or calculate features that are not needed in the future.

Unsupervised feature selection algorithms can be categorized as filter, wrapper, and embedded methods. Filter methods utilize some intrinsic properties of the data to decide which features should be kept, without running the learning (i.e., clustering) algorithm that will ultimately be applied. For example, the filter method in Talavera (1999) selects features based on feature dependence; Manoranjan et al. (2002) chooses features based on the entropy of distances between data points; and He et al. (2006) picks features based on the Laplacian score. Wrapper methods, on the other hand, wrap feature search around the learning algorithms that will ultimately be applied, and utilize the learned results to select the features. Examples of wrapper methods are: Devaney & Ram (1997) applied both sequential forward and backward selection to search over the features, hierarchically clustered the data using COBWEB, and evaluated these feature subsets using the category utility metric; Vaithyanathan & Dom (1999) proposed a probabilistic

Appearing in *Proceedings of the 28th International Conference on Machine Learning*, Bellevue, WA, USA, 2011.

objective function based on a mixture of multinomials, a Bayesian approach to estimate the parameters, and distributional clustering to search candidate feature subsets; Dy & Brodley (2004) studied the issues involved in developing wrapper methods and examined maximum likelihood and scatter separability criteria for selecting features, mixture of Gaussians for clustering, and sequential forward search; Kim et al. (2002) applied an evolutionary local selection algorithm to search over the features and numbers of clusters on two clustering algorithms: K-means and Gaussian mixture clustering (with diagonal covariances); and, C. Maugis & Martin-Magniette (2009) select relevant features using backward stepwise selection for Gaussian mixture models and an integrated likelihood criterion approximated by the Bayesian information criterion to guide the search for features and to determine the number of clusters.

Wrapper approaches often lead to better performance compared to a filter approach for a particular learning algorithm. However, wrapper methods are more computationally expensive since one needs to run the learning algorithm for every candidate feature subset. Embedded methods lie somewhere between filter and wrapper approaches. They incorporate feature selection and clustering in one objective function formulation or algorithm. The approach in Law et al. (2002) is an embedded method. They added feature saliency, a measure of feature relevance, as a missing variable to a probabilistic objective function. To add feature saliency, they assumed that the features are conditionally independent. They then derived an expectation-maximization (EM) algorithm (Dempster et al., 1977) to estimate the feature saliency for a mixture of Gaussians. They are able to find the features and clusters simultaneously through a single EM run. Constantinopoulos et al. (2006); Chang et al. (2005) adopt the feature saliency model of Law et al. (2002); however, they provide a Bayesian formulation. Constantinopoulos et al. (2006) uses variational inference and Chang et al. (2005) applies expectation propagation to learn the model. The approach we introduce in this paper is an embedded method.

Another way to group feature selection algorithms for unlabeled data is based on whether or not the method selects global or local features. Global methods select a single set of features, whereas local methods select subsets of features, one subset for each cluster (where features in different clusters can vary). All the approaches mentioned in the previous paragraph are global methods. Local methods include co-clustering or bi-clustering algorithms (Hartigan, 1972) and subspace clustering algorithms (Fu & Banerjee, 2009). These methods try to maximize the coherence exhibited by a subset of instances on a subset of features. In microarray analysis, one may want to find the genes that respond similarly to the environment conditions; in text clustering, one may wish to consider the co-occurence of words and documents. Typical approaches to co-clustering alternate clustering the rows and the columns to find the co-clusters (Banerjee et al., 2004; Cho et al., 2004; Yang et al., 2002). Recently, Shafiei & Milios (2006) and Sohn & Xing (2009) introduce Dirichlet process mixtures for co-clustering and Fu & Banerjee (2009) provides a Bayesian formulation for discovering subspace clusters that allow overlap.

In this paper, we provide a unified probabilistic model that can be set to perform global or local feature selection by adjusting the variance of a beta variate in a beta-Bernoulli hierarchical prior on the features. We use this beta-Bernoulli prior in the context of a Dirichlet process mixture for clustering. We provide a variational inference method for our probabilistic Bayesian formulation. Such a model allows us to simultaneously learn the clusters and important features. Additional benefits of our model are: (1) as a Bayesian formulation, we can automatically learn the number of clusters and the number of features to keep; (2) our model is not limited by the conditional feature independence assumption (we allow dependencies between features), contrary to existing global embedded feature selection probabilistic formulations (Law et al., 2002; Constantinopoulos et al., 2006; Chang et al., 2005); and (3) the features in each cluster in our local feature selection model need not be disjoint, which might be desirable in some applications. Experiments on synthetic and real data show that our unified model can find global and local features and cluster data as well as competing methods of each type.

2. Review of Dirichlet Process Mixtures

Let $\mathbf{x}_n \in \mathbb{R}^D$ represent a data point. Consider a collection of N $\mathbf{x}_n$ data points, $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]^T$, where $(\cdot)^T$ is the transpose of a matrix. Clustering can be modeled by a finite mixture model (McLachlan & Basford, 1988). This model assumes that data are generated from a mixture of component density functions, where each component density, $p(\mathbf{x}_n|\theta_k)$ represents a cluster

$$p(\mathbf{x}_n) = \sum_{k=1}^{K} \pi_k p(\mathbf{x}_n|\theta_k), \tag{1}$$

where $p(\mathbf{x}_n|\theta_k)$ denotes a density function for cluster k with parameters θ_k, K is the number of clusters, and π_k is the mixing proportion or prior probability of component k, subject to the constraints: $\pi_k \geq 0$ and

$\sum_{k=1}^{K} \pi_k = 1$.

Instead of fixing K, we use a Dirichlet process mixture (DPM) framework to infer the number of clusters. The DPM can be obtained from a stick-breaking construction as follows (Sethuraman, 1994). First draw an independent collection of beta random variables, $v_k \sim \text{Beta}(1, \alpha)$, for $k = 1, 2, \ldots, \infty$, and form new variables $\pi_k = v_k \prod_{j=1}^{k-1}(1 - v_j)$. Define a cluster indicator variable, z_n, which is drawn from a discrete distribution parameterized by π, $z_n = \text{Discrete}(\pi)$:

$$p(z_n) = \prod_{k=1}^{\infty} \pi_k^{\{z_n = k\}} \tag{2}$$

If the cluster indicator z_n for the data point $\mathbf{x}_n$ is equal to k, then the data point is modeled as generated by a distribution parameterized by a parameter θ_k. In our experiments, we use a Gaussian cluster component model for real-valued features and a multinomial component model for text data. Finally, the parameters for each cluster component are drawn from a conjugate prior distribution with parameters η. Figure 1 provides a graphical model representation of the DPM.

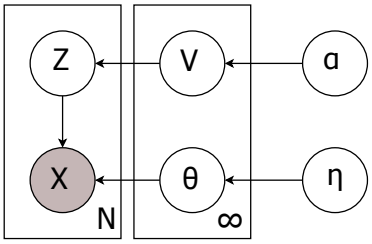

Figure 1. Graphical model of the stick-breaking construction for a Dirichlet process mixture.

3. Unified Global and Local Unsupervised Feature Selection

In the global case, our goal is to select the relevant features which reveals interesting cluster structure. We use a Dirichlet process mixture to learn the cluster structure as described in Section 2. We enable feature selection by adding a latent vector variable $\mathbf{y} \in \mathbb{R}^D$ whose elements, y_d, are either one or zero, indicating whether feature d is selected or not. The selected features are then the features utilized to form the mixture model. The unselected or noisy features are assumed to be generated from a single cluster distribution. We assume an isotropic Gaussian noise model for the Gaussian component case and a flat multinomial noise model for the multinomial case, similar to that in Law et al. (2002) and Vaithyanathan & Dom (1999).

In the local case, our goal is to select the relevant features describing each cluster component. We now use a latent feature indicator matrix $Y \in \mathbb{R}^{K \times D}$ whose elements, y_{kd}, are either one or zero indicating whether feature d is selected to model cluster component k or not. Features that are not selected in any cluster component are assumed to come from a Gaussian or multinomial noise distribution accordingly.

In our unified global and local feature selection model, we utilize the latent feature indicator matrix $Y \in \mathbb{R}^{K \times D}$. Since y_{kd} is either one or zero, we assume that it is generated from a Bernoulli distribution with parameter λ_d capturing the probability of selecting feature d in cluster k. In turn, λ_d is generated from a beta prior with parameters m and ρ, where m is the mean of a beta distribution and ρ is the variance. m controls the average percentage of features that will be selected in each cluster. One can allow m_k to be different for each cluster k. Here, for the sake of simplicity, we use the same m for all clusters in our experiments. We further generate m from a beta distribution with hyperparameters γ and β. We enable global or local feature selection by adjusting the variance ρ parameter. For the global case, we set γ and β so that m will be distributed from a U-shaped beta distribution as shown in Figure 2a to push m to either zero or one. When ρ is low, the beta distribution will have a shape either as shown in Figure 2b (when $m \approx 1$) or in Figure 2c (when $m \approx 0$). Because the variance of λ_d is low, feature d will be either selected or not together in all clusters. For the local case, the shape of the beta distribution in m is not critical. The key thing is to set ρ high, leading to a beta distribution for λ_d to have a broad distribution as shown in Figure 2d, allowing λ_d to be somewhere between zero and one. This allows local feature selection, enabling z_{kd} to select features in each cluster.

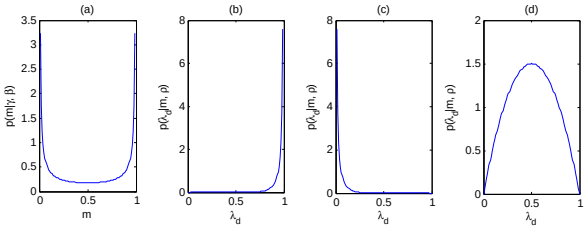

Figure 2. How to enable global or local feature selection through the parameters of the beta distribution. (a) Global case: U-shaped beta for m. (b-c) Global case: beta distribution for λ_d with the variance ρ set to a low value and $m \approx 1$ (b) and $m \approx 0$ (c). (d) Local case: example beta distribution for λ_d with variance ρ set to a high value.

The cluster component densities $p(\mathbf{x}_n|\theta_k)$ are then based only on the features whose $y_{kd} = 1$. We summarize the generative process as follows:

1. Generate m from a beta distribution, $m \sim \text{Beta}(\gamma, \beta)$.
2. λ_d is the probability of a feature to be selected. It is generated from a beta distribution with mean m and variance ρ. Setting ρ small leads to global feature selection and ρ large enables local feature selection.
3. y_{kd} is a feature indicator variable, identifying whether feature d is selected in cluster k. $y_{kd} \sim \text{Bernoulli}(\lambda_d)$.
4. Generate independent variables $\{v_k \sim \text{Beta}(1, \alpha)\}$, and form stick-breaking variables $\{\pi_k\}$ where $\pi_k = v_k \prod_{j=1}^{k-1}(1 - v_j)$.
5. Generate the cluster indicator variable, z_n, from a discrete distribution parameterized by π.
6. Generate θ_k, the parameters for cluster k, from the respective conjugate prior for our component density model (normal-inverse Wishart for the Gaussian and Dirichlet for the multinomial case), where η is the hyperparameter for this prior.
7. Let $\mathbf{x}_{n,k} = (x_{nd} : y_{kd} = 1)$ denote the nth data sample based only on the features that are selected for cluster k (i.e., $y_{kd} = 1$). Generate the samples, $\mathbf{x}_{n,k}$ from a component density (e.g., Gaussian or multinomial) with parameters θ_k associated for cluster k.
8. Generate θ_{noise}, the parameter for the noise distribution (isotropic Gaussian or flat multinomial) from its respective conjugate prior with parameters η_{noise}. Generate the unselected features from the noise distribution with parameter θ_{noise}.

The graphical model for our unified unsupervised global and local feature selection model is shown in Figure 3. The joint probability distribution is:

$$
\begin{aligned}
& p(X, \theta_k, \theta_{noise}, z_n, \pi, y_{kd}, \lambda_d, m) \\
&= p(X|\theta_k, \theta_{noise}, z_n, y_{kd}) p(\theta_k) p(\theta_{noise}) \\
&\quad p(z_n|\pi) p(\pi|\alpha) p(y_{kd}|\lambda_d) p(\lambda_d|m, \rho) p(m) \\
&= \prod_{n=1}^{N} p(\mathbf{x}_n|\theta_k, z_n, y_{kd} = 1) \prod_{k=1}^{K} p(\theta_k) p(\theta_{noise}) \\
&\quad \prod_{n=1}^{N} p(z_n|\pi) p(\pi) \prod_{k=1}^{K} \prod_{d=1}^{D} p(y_{kd}|\lambda_d) \\
&\quad \prod_{d=1}^{D} p(\lambda_d|m, \rho) p(m) \prod_{n=1}^{N} p(\mathbf{x}_n|\theta_{noise}, y_{kd} = 0)
\end{aligned}
$$

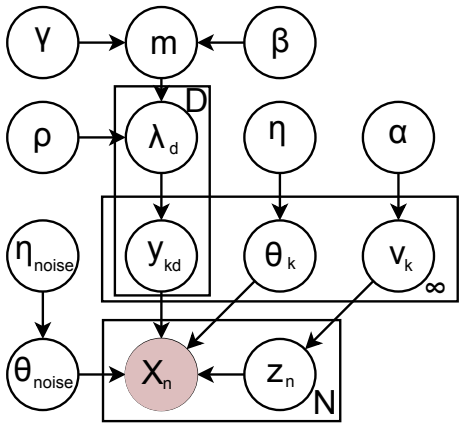

Figure 3. Graphical model for our unified global and local unsupervised feature selection.

4. Variational Inference

It is computationally intractable to evaluate the marginal likelihood, $p(D) = \int p(D, \phi) d\phi$, where $\phi = \{\phi_i\}$ represents the set of all parameters and latent variables. Variational methods allow us to approximate the marginal likelihood by maximizing a lower bound, $\mathcal{L}(Q)$, on the true log marginal likelihood (Wainwright & Jordan, 2006). $\ln p(D) = \ln \int p(D, \phi) d\phi = \ln \int Q(\phi) \frac{p(D,\phi)}{Q(\phi)} d\phi \geq \int Q(\phi) \ln \frac{p(D,\phi)}{Q(\phi)} d\phi = \mathcal{L}(Q(\phi))$, using Jensen's inequality. The difference between the log marginal $p(D)$ and the lower bound $\mathcal{L}(Q)$ is the Kullback-Leibler divergence between the approximating distribution $Q(\phi)$ and the true posterior $p(\phi|D)$. The idea is to choose a $Q(\phi)$ distribution that is simple enough that the lower bound can be tractably evaluated and flexible enough to get a tight bound. Here, we assume a distribution for $Q(\phi)$ that factorizes over all the parameters: $Q(\phi) = \prod_i Q_i(\phi_i)$. The $Q_i(\phi_i)$ that minimizes the KL divergence over all factorial distributions is

$$
Q_i(\phi_i) = \frac{\exp \langle \ln P(D, \phi) \rangle_{l \neq i}}{\int \exp \langle \ln P(D, \phi) \rangle_{l \neq i} d\phi_j} \tag{3}
$$

where $\langle \cdot \rangle_{l \neq i}$ is the expectation of ϕ_i with respect to other variables. We also apply the variational inference result for Dirichlet process mixture from Blei & Jordan (2006). The components of the approximate posterior $Q_i(\cdot)$ are given as follows:

$$
\begin{aligned}
Q(v_k) &\sim Beta(v_k|\gamma_{k,1}, \gamma_{k,2}) && (4) \\
Q(z_n) &\sim Discrete(z_n|\psi) && (5) \\
Q(y_{kd}) &\sim Bernoulli(p_{y_{kd}}) && (6) \\
Q(\lambda_d) &\sim Beta(\alpha_d, \beta_d) && (7) \\
Q(m) &\sim Beta(\alpha_m, \beta_m) && (8)
\end{aligned}
$$

where $\langle\cdot\rangle$ is an expectation operator with respect to the approximation $Q(\cdot)$.

We obtain the following update equations:

1. For v_k, we have the update equation from Blei & Jordan (2006):

$$\gamma_{k,1} = 1 + \sum_n \langle\psi_{n,k}\rangle \quad (9)$$

$$\gamma_{k,2} = \alpha + \sum_n \sum_{j=K+1}^{K} \langle\psi_{n,j}\rangle \quad (10)$$

2. The update equation for $\psi_{n,k}$ is the same as in the stick-breaking construction for DPM:

$$\psi_{n,k} \propto \langle\pi_k\rangle P(X_n|\langle\theta_k\rangle) \quad (11)$$

That is, every possible output is evaluated from Eq. (11) and the weight is normalized to sum to one.

3. For $p_{y_{kd}}$,

$$\begin{aligned} &L(p_{y_{kd}}) \\ &= p_{y_{kd}} \ln[P(X|\langle\theta_k\rangle, \langle z\rangle, y_{kd}=1)] \\ &+(1-p_{y_{kd}}) \ln[P(X|\langle\theta_{noise}\rangle, \langle z\rangle, y_{kd}=0)] \\ &+p_{y_{kd}} \ln\langle\lambda_d\rangle + (1-p_{y_{kd}}) \ln(1-\langle\lambda_d\rangle) \\ &-(p_{y_{kd}} \ln p_{y_{kd}} + (1-p_{y_{kd}}) \ln(1-p_{y_{kd}})) \\ &+\text{constant} \end{aligned}$$

The partial derivative of $L(p_{y_{kd}})$ with respect to $p_{y_{kd}}$ is:

$$\begin{aligned} \frac{\partial L(p_{y_{kd}})}{\partial p_{y_{kd}}} &= \ln[p(X|\langle\theta_k\rangle, \langle z\rangle, y_{kd}=1)] \\ &\quad - \ln[p(X|\langle\theta_{noise}\rangle, \langle z\rangle, y_{kd}=0)] \\ &\quad + \ln\lambda_d - \ln(1-\lambda_d) \\ &\quad - [\ln p_{y_{kd}} - \ln(1-p_{y_{kd}})] \end{aligned} \quad (12)$$

Let

$$\begin{aligned} P_e = \exp\{&\ln[p(X|\langle\theta_k\rangle, \langle z\rangle, y_{kd}=1)] \\ &- \ln[p(X|\langle\theta_{noise}\rangle, \langle z\rangle, y_{kd}=0)] \\ &+ \ln\lambda_d - \ln(1-\lambda_d)\} \end{aligned} \quad (13)$$

Then:

$$p_{y_{kd}} = \frac{P_e}{1+P_e} \quad (14)$$

is the update equation for $p_{y_{kd}}$.

4. For λ_d, since we are using a conjugate prior, we obtain a closed-form update:

$$\alpha_d = \sum_d \langle p_{y_{kd}}\rangle + \alpha_m \quad (15)$$

$$\beta_d = \sum_d (1 - \langle p_{y_{kd}}\rangle) + \beta_m \quad (16)$$

5. For m, we want to find the expectation of the following function based on $Q(m)$, $\ln p(D,\phi) = \ln\Gamma(\frac{m-m^2-\rho}{\rho}) - \ln\Gamma\frac{m^2-m^3-m\rho}{\rho} - \ln\Gamma\frac{m-2m^2+m^3-\rho+m\rho}{\rho} + \sum\frac{m^2-m^3-m\rho-\rho}{\rho}\ln\langle\lambda_d\rangle + \sum\frac{m-2m^2+m^3-2\rho+m\rho}{\rho}\ln(1-\langle\lambda_d\rangle)$ + constant. Since the optimal m that maximizes $\ln p(D,\phi)$ cannot be found in closed form, we obtain the solution by gradient ascent subject to the constraint $m \in (0.5-\sqrt{0.25-\rho}, 0.5+\sqrt{0.25-\rho})$.

We provide two common observation probability models for modeling cluster components in mixture models: the Gaussian component and the multinomial component model. The Gaussian model is widely used for real-valued data where samples are assumed to be variations of some prototype. A multinomial model is appropriate for discrete data, such as text.

Gaussian Component. Assuming $p(\mathbf{x}_{n,k}|\theta_k)$ comes from a Gaussian distribution, our parameter vector θ_k comprises the mean μ_k and covariance Σ_k of our Gaussian distribution in cluster k. We apply a normal-inverse Wishart distribution, the conjugate prior to a Gaussian distribution as our prior $p(\theta_k|\eta)$. The hyperparameter η is a vector composed of the mean $\mathbf{m}_0$, covariance S_0, inverse scale matrix Ψ_0, and parameter p_0. Applying a variational approximation, we have $Q(\theta_k)$ for each cluster k as:

$$\frac{\Psi_k^{d/2}}{(2\pi)^{d/2}|\Sigma_k|^{1/2}} \exp(-\frac{1}{2}(\mu_k-\mathbf{m}_k)^T\Psi_k\Sigma_k^{-1}(\mu_k-\mathbf{m}_k)) \frac{|S_k|^{p_k/2}|\Sigma_k|^{-(p_k+d+1)/2}\exp(-\frac{1}{2}\text{trace}(S_k\Sigma_k^{-1}))}{2^{p_kd/2}\pi^{d(d-1)/4}\prod_{j=1}^{d}\Gamma((p_k+1-j)/2)}$$

where $\mathbf{m}_k$, S_k, Ψ_k and p_k are the parameters of the posterior normal-inverse Wishart distribution for cluster k.

The update equations are:

$$\mathbf{m}_k = \frac{n_k\bar{\mathbf{x}}_k + \Psi_0\mathbf{m}_0}{n_k+\Psi_0} \quad (17)$$

$$\Psi_k = \Psi_0 + n_k \quad (18)$$

$$p_k = p_0 + n_k \quad (19)$$

$$S_k = S_0 + \sum_{\mathbf{x}_i\in k}(\mathbf{x}_i-\bar{\mathbf{x}}_k)(\mathbf{x}_i-\bar{\mathbf{x}}_k)^T + \frac{n_k\Psi_0}{n_k+\Psi_0}(\bar{\mathbf{x}}_k-\mathbf{m}_0)(\bar{\mathbf{x}}_k-\mathbf{m}_0)^T, \quad (20)$$

where $\bar{\mathbf{x}}_k$ is the sample mean of $\mathbf{x}$ in cluster k in the selected features and n_k are the number of samples in cluster k. The parameters μ_k and Σ_k are updated by their expected values under the variational distribution. The Gaussian noise parameters are updated similarly.

Multinomial Component. If $p(\mathbf{x}_{n,k}|\theta_k)$ comes from a multinomial distribution, our parameter θ_k is from a multinomial distribution with dimension q, where q is the number of selected features for that cluster. The prior η is the conjugate Dirichlet distribution with the same dimension and the Dirichlet distribution is set to uniform with parameter 1. Since $p(\mathbf{x}_{n,k}|\theta_k) = Mult(\mathbf{p}_k)$ and $p(\theta_k|\eta) = Dirichlet(\eta)$, the update equation is: $\mathbf{p}_k = \eta + \sum n_k$, where n_k is the number of samples in cluster k. The multinomial noise parameters are updated similarly.

5. Experiments

In this section, we investigate whether or not our proposed algorithm can simultaneously find reasonable clusters and features, both global and local. We test our algorithm on synthetic and real data.

In our experiments we set the hyperparameters as follows: γ and β for the beta distribution is set to 1, α for stick-breaking construction for the Dirichlet process mixture is set to 5. When we want to achieve local feature selection, we set $\rho = 0.2$ and we set $\rho = 0.01$ to achieve global feature selection. We normalize all our data such that each dimension is zero-centered and the variance is one.

5.1. Synthetic Data

We first test our algorithm on synthetic data. We generated two synthetic data sets: one to test local feature selection and the other for global feature selection. In both cases we generated data with 300 samples and 30 features, with each cluster having 100 samples. Each of the three clusters have low variance (here we set the variance for each feature to one). We assumed an isotropic Gaussian noise distribution for the noisy features and set the variance for each feature to some high value (in particular we set the variance of each feature to ten). For the local case (Synthetic Data 1), the first ten features are used to generate cluster one, the last fifteen features are used to generate the second cluster, and the third cluster is generated using features five to 25. Note that we intentionally allowed feature overlap. For the global case (Synthetic Data 2), the first fifteen features are used to generate the three clusters. Figure 4 displays the data matrix with the value for each feature in sample n coded in grayscale (lowest value in black and highest value in white).

Figure 4 shows tile plots, which are plots showing the learned partitioning of the features and clusters discovered by our algorithm. Tiles with the same color indicate which features and which samples belong to the same cluster. Black indicates unselected features. Notice from the figures on the left and right that we are able to learn the latent partitioning (tile) structures of Synthetic Data 1 and 2 correctly and perform local and global feature selection respectively.

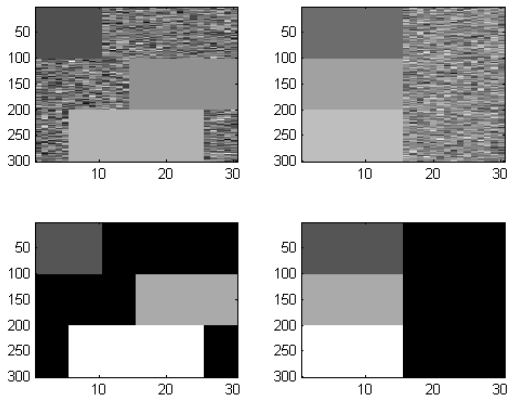

Figure 4. The top left figure is the data matrix for Synthetic Data 1 (local case) and the top right is Synthetic Data 2 (global case). The bottom left figure is the latent labeling obtained by our algorithm for local feature selection and the bottom right figure is the label obtained for global feature selection. We can see that all three clusters and the corresponding features are identified in both cases.

In general, local feature selection is more flexible and can learn global features if the data has that structure. In real applications, data often will not have exactly a global feature subset structure. But one may be interested in solving the global feature selection problem for reasons of interpretability or simplicity. A local model will not prefer a solution where all the clusters have the same q features over one with q features in each cluster where the features in different clusters may vary even when the model fit for the global one is only slightly worse. Our model allows us to impose this constraint to provide us with a global solution if that is what the user/application requires. Moreover, our model allows us to control the amount of sharing as well.

5.2. Real World Data

In this section, we test our algorithm on a set of real benchmark data. We use the chart, face, webKB, digits, mini-newsgroups, Reuters and glass data sets from the UCI Machine Learning Repository (Blake & Merz, 1998), a high-resolution computed tomography (HRCT) data (Dy & Brodley, 2004) and a yeast gene data (Cherry et al., 1997) in our experiments. Chart

Table 1. NMI results on real data.

	HFS(G)	Law	DPM	HFS(L)	Cc
Chart	**0.615** (8)	0.581 (4)	0.331 (7)	0.681 (7)	**0.712** (6)
Hrct	**0.482** (11)	0.363 (6)	0.207 (12)	**0.482** (11)	0.479 (8)
Faces	**0.627** (16)	0.454 (4)	0.481 (12)	**0.569** (12)	0.547 (4)
WebKB	**0.183** (7)	0.158 (1)	0.118 (9)	**0.258** (6)	0.214 (4)
Digits	**0.258** (13)	0.254 (6)	0.176 (5)	0.354 (11)	**0.364** (10)
miniNG	**0.361** (21)	0.159 (1)	0.129 (14)	**0.582** (24)	0.546 (20)
Yeast	**0.584** (6)	0.475 (2)	0.236 (3)	**0.650** (2)	0.576 (2)
Reuters	**0.276** (9)	0.153 (6)	0.042 (17)	0.389 (13)	**0.407** (5)
Glass	**0.730** (7)	0.572 (4)	0.413 (9)	0.731 (8)	**0.783** (6)

has 600 samples with 60 features and six classes. Hrct is a high-resolution computed tomography lung data with eight disease classes, 1545 instances and 183 features. This has a highly-skewed class distribution with the largest class comprising 604 samples and the smallest class containing only 30 samples. The Face data set consists of 640 face images from twenty people. Each person has 32 images with an image resolution of 32 by 30. WebKB is a webpage text data from four universities. We pre-processed the data by removing rare words, stop words, and retaining only the words with large variances. Digits has 11000 handwritten digit images, and each image is represented by its 16 by 16 gray level pixel intensities resulting in 256 dimensions. Each digit has 1100 instances. Newgroups is a text data from online news groups. The original data has 20,000 instances in 20 classes. After removing classes with small populations, we have 1300 documents with 800 words in 13 classes. We call this as our "miniNG" data. We applied stemming and removed low variance words. Yeast is a two-class yeast gene data set with 103 values of gene expression and 917 observations. Each observation is a binary indicator. Reuters is a collection of documents that appeared on Reuters newswire in 1987. The data after pre-processing has 8963 documents and 635 words in 7 classes. The class with the largest population has 2423 instances and the class with the lowest population has 338 instances. We applied stemming and removed low variance words. Glass data has nine attributes, six classes, and 214 samples.

We compare our approach to no feature selection, using a simple Dirichlet process mixture of Gaussians (DPM-GMM). Moreover, we compare our global hierarchical feature selection (HFS(G)) approach with the method in Law et al. (2002), and our local hierarchical feature selection (HFS(L)) to co-clustering (Cho et al., 2004). We compare with Law et al. (2002) because it is a global feature selection method that is also based on a mixture model with Gaussians as cluster components. However, unlike our model, they assume conditional independence between the features; we model the covariance between the features. Local feature selection is widely applied in co-clustering applications. There are several variants of co-clustering algorithms based on the distance metric used. Here, we compare with a co-clustering algorithm that is based on the squared Euclidean distance of every column or row vector to its column or row cluster mean vector. Both Euclidean distance and Gaussian probability models work on similar types of data.

We evaluate the performance of our clustering results based on the normalized mutual information (NMI) (Strehl & Ghosh, 2002) between our clustering results and the clusters based on the known "true" class labels. We normalize the mutual information, $MI(X,Y)$, to fall in the range $[0,1]$ by defining $NMI(X,Y) = MI(X,Y)/\sqrt{H(x)H(Y)}$, where $H(x)$ and $H(Y)$ denote entropy of X and Y. The higher the value of NMI, the better the consistency of the clustering result with respect to the labeled classes. Note that we did not utilize the labels in learning our clusters; we only use them for evaluation.

Table 1 presents the NMI results and the effective number of clusters in parenthesis, (k). The results show that our global hierarchical feature selection (HFS(G)) is better than Law et al. (2002) and without feature selection (DPM). Similarly, our local hierarchical feature selection approach (HFS(L)) is generally better than co-clustering (Cc). Cc is not easy to beat because it uses additional information (the number of clusters is assumed known and we use the number equal to the number of classes). Our HFS approach automatically learns this number of clusters from data.

6. Conclusions

In this paper, we have introduced a unified probabilistic formulation for both global and local unsupervised

feature selection. We achieve feature selection by utilizing a beta-Bernoulli hierarchical prior. We enable global feature selection by setting the variance low and local feature selection by setting the variance to a high value. We use this global/local feature selection prior in the context of a Dirichlet process mixture model to simultaneously learn both features and clusters. We developed a variational inference algorithm for approximate posterior inference under this formulation. Results on synthetic and real data show that our model can both discover clusters and features both globally and locally.

Acknowledgments

This work is partially supported by NSF IIS-0347532 and NSF IIS-0915910.

References

Banerjee, A., Dhillon, I. S., Ghosh, J., Merugu, S., and Modha, D. S. A generalized maximum entropy approach to Bregman co-clustering and matrix approximations. In *ACM SIGKDD Intn'l Conf. on Knowledge Discovery and Data Mining*, pp. 509–514, August 2004.

Blake, C.L. and Merz, C.J. UCI repository of machine learning databases. In *http://www.ics.uci.edu/∼mlearn/MLRepository.html*, 1998.

Blei, D. M. and Jordan, M. I. Variational inference for Dirichlet process mixtures. *Bayesian Analysis*, 1(1):121–144, 2006.

C. Maugis, G. Celeux and Martin-Magniette, M.-L. Variable selection for clustering with Gaussian mixture models. *Biometrics*, 65:701–709, 2009.

Chang, S., Dasgupta, N., and Carin, L. A Bayesian approach to unsupervised feature selection and density estimation using expectation propagation. In *IEEE Conf. on Computer Vision and Pattern Recognition*, volume 2, pp. 1043–1050, 2005.

Cherry, J. M., Ball, C., Weng, S., and Juvik, G. Genetic and physical maps of Saccharomyces cerevisiae. *Nature*, 387(6632):67–73, 1997.

Cho, H., Dhillon, I. S., Guan, Y., and Sra, S. Minimum sum-squared residue co-clustering of gene expression data. *SIAM Int'l Conf. on Data Mining*, pp. 114–125, 2004.

Constantinopoulos, C., Titsias, M. K., and Likas, A. Bayesian feature and model selection for Gaussian mixture models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28:1013–1018, 2006.

Dempster, A. P., Laird, N. M., and Rubin, D. B. Maximum likelihood from incomplete data via the EM algorithm. *Journal Royal Statistical Society, Series B*, 39(1):1–38, 1977.

Devaney, M. and Ram, A. Efficient feature selection in conceptual clustering. In *International Conference on Machine Learning*, pp. 92–97, 1997.

Dy, J. G. and Brodley, C. E. Feature selection for unsupervised learning. *Journal of Machine Learning Research*, 5:845–889, August 2004.

Fu, Q. and Banerjee, A. Bayesian overlapping subspace clustering. In *Ninth IEEE International Conference on Data Mining*, pp. 776–781, 2009.

Hartigan, J. A. Direct clustering of a data matrix. *Journal of the American Statistical Association*, 67(337):123–129, 1972.

He, X., Cai, D., and Niyogi, P. Laplacian score for feature selection. In *Advances in Neural Information Processing Systems 18*, pp. 507–514. MIT Press, 2006.

Kim, Y. S., Street, N., and Menczer, F. Evolutionary model selection in unsupervised learning. *Intelligent Data Analysis*, 6:531–556, 2002.

Law, M. H., Figueiredo, M., and Jain, A. K. Feature selection in mixture-based clustering. In *Advances in Neural Information Processing Systems 15*, 2002.

Manoranjan, D., Choi, K., Scheuermann, P., and Liu, H. Feature selection for clustering—a filter solution. In *IEEE Int'l Conf. on Data Mining*, pp. 115–122, 2002.

McLachlan, G. J. and Basford, K. E. *Mixture Models, Inference and Applications to Clustering*. Marcel Dekker, New York, 1988.

Sethuraman, J. A constructive definition of Dirichlet priors. *Statistica Sinica*, 4:639–650, 1994.

Shafiei, M. and Milios, E. Latent Dirichlet co-clustering. In *IEEE Int'l Conf. on Data Mining*, pp. 542–551, 2006.

Sohn, K. and Xing, E. A hierarchical Dirichlet process mixture model for haplotype reconstruction from multipopulation data. *Annals of Applied Statistics*, 2009.

Strehl, A. and Ghosh, J. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal on Machine Learning Research*, 3:583–617, 2002.

Talavera, L. Feature selection as a preprocessing step for hierarchical clustering. In *Int'l Conf. on Machine Learning*, pp. 389–397, 1999.

Vaithyanathan, S. and Dom, B. Model selection in unsupervised learning with applications to document clustering. In *Int'l Conf. on Machine Learning*, pp. 433–443, 1999.

Wainwright, M. J. and Jordan, M. I. Graphical models, exponential families and variational inference. *Foundations and Trends in Machine Learning*, 1:1–305, 2006.

Yang, J., Wang, W., Wang, H., and Yu, P. δ-clusters: Capturing subspace correlation in a large data set. In *International Conference on Data Engineering*, pp. 517–528, 2002.