Bayesian Supervised Dictionary learning

B. Babagholami-Mohamadabadi
CE Dept.
Sharif University
Tehran, Iran

A. Jourabloo
CE Dept.
Sharif University
Tehran, Iran

M. Zolfaghari
CE Dept.
Sharif University
Tehran, Iran

M.T. Manzuri-Shalmani
CE Dept.
Sharif University
Tehran, Iran

Abstract

This paper proposes a novel Bayesian method for the dictionary learning (DL) based classification using Beta-Bernoulli process. We utilize this non-parametric Bayesian technique to learn jointly the sparse codes, the dictionary, and the classifier together. Existing DL based classification approaches only offer point estimation of the dictionary, the sparse codes, and the classifier and can therefore be unreliable when the number of training examples is small. This paper presents a Bayesian framework for DL based classification that estimates a posterior distribution for the sparse codes, the dictionary, and the classifier from labeled training data. We also develop a Variational Bayes (VB) algorithm to compute the posterior distribution of the parameters which allows the proposed model to be applicable to large scale datasets. Experiments in classification demonstrate that the proposed framework achieves higher classification accuracy than state-of-the-art DL based classification algorithms.

1 Introduction

Sparse signal representation (Wright et al., 2010), has recently gained much interest in computer vision and pattern recognition. Sparse codes can efficiently represent signals using linear combination of basis elements which are called atoms. A collection of atoms is referred to as a dictionary. In sparse representation framework, dictionaries are usually learned from data rather than specified apriori (i.e wavelet).
It has been demonstrated that using learned dictionaries from data usually leads to more accurate representation and hence can improve performance of signal reconstruction and classification tasks (Wright et al., 2010). Several algorithms have been proposed for the task of dictionary learning (DL), among which the K-SVD algorithm (Aharon et al., 2006), and the Method of Optimal Directions (MOD) (Engan et al., 1999), are the most well-known algorithms. The goal of these methods is to find the dictionary $D = [\boldsymbol{d}_1, \boldsymbol{d}_2, ..., \boldsymbol{d}_K]$, and the matrix of the sparse codes $A = [a_1, a_2, ..., a_N]$, which minimize the following objective function

$$[\hat{A}, \hat{D}] = argmin_{A,D} \|X - DA\|_F^2, \quad s.t. \quad \|x_i\|_0 \leq T \ \ \forall i, \tag{1}$$

where $X = [x_1, x_2, ..., x_N]$ is the matrix of N input signals, K is the number of the dictionary atoms, $\|.\|_F$ denotes the Frobenius norm, and $\|x\|_0$ denotes the l_0 norm which counts the number of non-zero elements in the vector x.

2 Related Work

Classical DL methods try to find a dictionary, such that the reconstructed signals are fairly close to the original signals, therefore they do not work well for classification tasks. To overcome this problem, several methods have been proposed to learn a dictionary based on the label information of the input signals.
Wright (Wright et al., 2009), used training data as the atoms of the dictionary for Face Recognition (FR) tasks. This method determines the class of each query face image by evaluating which class leads to the minimal reconstruction error. Although the result of this method on face databases are promising, it is not appropriate for noisy training data. Being unable to utilize the discriminative information of the training data is another weakness of this method.
Yang (Yang et al., 2010), learned a dictionary for each class and obtained better FR results than Wright method. Yang (Yang et al., 2011), utilized Fisher Discriminant Analysis (FDA) to learn a sub-dictionary for each class in order to make the sparse coefficients more discriminative. Ramirez (Ramirez et al., 2010), added a structured incoherence penalty term to the

objective function of the class specific sub-dictionary learning problem to make the sub-dictionaries incoherent. Mairal (Mairal et al., 2009), introduced a supervised DL method by embedding a logistic loss function to learn a single dictionary and a classifier simultaneously. Given a limited number of labeled examples, most DL based classification methods suffer from the following problem: since these algorithms only provide point estimation of the dictionary, the sparse codes, and the classifier which could be sensitive to the choice of training examples, they tend to be unreliable when the number of training examples is small. In order to address the above problem, this paper presents a Bayesian framework for supervised dictionary learning, termed **Bayesian Supervised Dictionary Learning**, that targets tasks where the number of training examples is limited. Using the full Bayesian treatment, the proposed framework for dictionary learning is better suited to dealing with a small number of training examples than the non-Bayesian approach.

Dictionary learning based on the Bayesian non-parametric models was originally proposed by Zhou (Zhou et al., 2009), in which a prior distribution is put on the sparse codes (each sparse code is modeled as an element-wise multiplication of a binary vector and a weight vector) which satisfies the sparsity constraint. Although the results of this method can compete with the state of the art results in denoising, inpainting, and compressed sensing applications, it does not work well for classification tasks due to its incapability of utilizing the class information of the training data.

To address the above problem, we extend the Bayesian non-parametric models for classification tasks by learning the dictionary, the sparse codes, and the classifier simultaneously. The contributions of this paper are summarized as follows:

- The noise variance of the sparse codes (the sparsity level of the sparse codes) and the dictionary is learn based on the Beta-Bernoulli process (Paisley et al., 2009) which allows us to learn the number of the dictionary atoms as well as the dictionary elements.
- A logistic regression classifier (multinomial logistic regression (Bohning, 1992), classifier for multiclass classification) is incorporated into the probabilistic dictionary learning model and is learned jointly with the dictionary and the sparse codes which improves the discriminative power of the model.
- The posterior distributions of the dictionary, the sparse codes, and the classifier is efficiently computed via the VB algorithm which allows the proposed model to be applicable to large-scale datasets.
- The Bayesian prediction rule is used to classify a test instance and therefore the proposed model is less prone to overfitting, specially when the size of the training data is small. Precisely speaking, test instances are classified by weighted average of the parameters (the dictionary, the sparse codes of the test instances , and the classifier), weighted by the posterior probability of each parameter value given the training data.
- Using the Beta-Bernoulli process model, many components of the learned sparse codes are exactly zero, which is different from the widely used Laplace prior, in which many coefficients of the sparse codes are small but not exactly zero.

The remainder of this paper is organized as follows: Section 3 briefly reviews the Beta-Bernoulli process. The proposed method is introduced in Section 4. Experimental results are presented in Section 5. We conclude and discuss future work in Section 6.

3 Beta-Bernoulli Process

The Beta process $B \sim BP(c, B_0)$ is an example of a Lévy process which was originally proposed by Hjort for survival analysis (Hjort, 1990), and can be defined as a distribution on positive random measures over a measurable space $(\Omega, \mathcal{F})$.

B_0 is the base measure defined over Ω and $c(\omega)$ is a positive function over Ω which is assumed constant for simplicity. The Lévy measure of $B \sim BP(c, B_0)$ is defined as

$$\nu(d\pi, d\omega) = c\pi^{-1}(1-\pi)^{c-1} d\pi B_0(d\omega). \qquad (2)$$

In order to draw samples from $B \sim BP(c, B_0)$, Kingman (Kingman, 1993), proposed a procedure based on the Poisson process which goes as follows.

First, a non-homogeneous Poisson process is defined on $\Omega \times \mathcal{R}^+$ with intensity function ν. Then, $Poisson(\lambda)$ number of points $(\pi_k, \omega_k) \in [0,1] \times \Omega$ are drawn from the Poisson process ($\lambda = \int_{[0,1]} \int_{\Omega} \nu(d\omega, d\pi) = \infty$). Finally, a draw from $B \sim BP(c, B_0)$ is constructed as

$$B_\omega = \sum_{k=1}^{\infty} \pi_k \delta_{\omega_k}, \qquad (3)$$

where δ_{ω_k} is a unit point measure at ω_k (δ_{ω_k} equals one if $\omega = \omega_k$ and is zero otherwise) . It can be seen from equation 3, that B_ω is a discrete measure (with probability one), for which $B_\omega(A) = \sum_{k:\omega_k \in A} \pi_k$, for any set $A \subset \mathcal{F}$.

If we define $Z = Be(B)$ as a Bernoulli process with B_ω defined as 3, a sample from this process can be drawn as

$$Z = \sum_{k=1}^{\infty} z_k \delta_{\omega_k}, \tag{4}$$

where z_k is generated by

$$z_k \sim Bernoulli(\pi_k). \tag{5}$$

If we draw N samples $(Z_1, ..., Z_N)$ from the Bernoulli process $Be(B)$ and arrange them in matrix form, $\mathbf{Z} = [Z_1, ..., Z_N]$, then the combination of the Beta process and the Bernoulli process can be considered as a prior over infinite binary matrices, with each column Z_i in the matrix $\mathbf{Z}$ corresponding to a location, δ_{ω_i}. By marginalizing out the measure B, the joint probability distribution $P(Z_1, ..., Z_N)$ corresponds to the Indian Buffet Process (Thibaux et al., 2007).

4 Proposed Method

All previous classification based sparse coding and dictionary learning methods have three shortcomings. First, the noise variance or the sparsity level of the sparse codes must be specified apriori in order to define the stopping criteria for estimating the sparse codes. Second, the number of the dictionary atoms must be set in advance (or determined via the cross-validation technique). Third, a point estimate of the dictionary and the sparse codes are used to predict the class label of the test data points which can result in overfitting. To circumvent these shortcomings, we propose a Bayesian probabilistic model in terms of the Beta-Bernoulli process which can infer both the dictionary size and the noise variance of the sparse codes from data. Furthermore, our approach integrates a logistic regression classifier (multinomial logistic regression classifier for multiclass classification) into the proposed probabilistic model to learn the dictionary and the classifier simultaneously while most of the algorithms learn the dictionary and the classifier separately.

4.1 Problem Formulation

Consider we are given a training set of N labeled signals $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N] \in R^{M \times N}$, each of them may belong to any of the c different classes. We first consider the case of $c = 2$ classes and later discuss the multiclass extension. Each signal is associated with a label $(y_i \in \{-1, 1\}, i = 1, ..., N)$. We model each signal $\boldsymbol{x}_i$, as a sparse combination of atoms of a dictionary $D \in R^{M \times K}$, with an additive noise ϵ_i. The Matrix form of the model can be formulated as

$$X = DA + E, \tag{6}$$

where $X_{M \times N}$ is the set of the input signals, $A_{K \times N}$ is the set of the K dimensional sparse codes, and $E \sim \mathcal{N}(0, \gamma_x^{-1} I_M)$ is the zero-mean Gaussian noise with precision value γ_x (I_M is an $M \times M$ Identity matrix). Following (Zhou, 2009), we model the matrix of the sparse codes (A) as an element-wise multiplication of a binary matrix (Z) and a weight matrix (S). Hence, the model of equation 6 can be reformulated as

$$X = D(Z \odot S) + E, \tag{7}$$

where $\odot$ is the element-wise multiplication operator. We put a prior distribution on the binary matrix Z using the extension of the Beta-Bernoulli process which takes two scalar parameters a and b and was originally proposed by (Paisley, 2009). A sample from the extended Beta process $B \sim BP(a, b, B_0)$ with base measure B_0 may be represented as

$$B_\omega = \sum_{k=1}^{K} \pi_k \delta_{\omega_k}, \tag{8}$$

where,

$$\pi_k \sim Beta(a/K, b(K-1)/K), \quad \omega_k \sim B_0. \tag{9}$$

This sample will be a valid sample from the extended Beta process, if $K \to \infty$. B_ω can be considered as a vector of K probabilities that each probability π_k corresponds to the atom ω_k. In our framework, we consider each atom ω_k as the k-th atom of the dictionary (d_k) and we set the base measure B_0 to a multivariate zero-mean Gaussian distribution $\mathcal{N}(0, \gamma_d^{-1} I_K)$ (with precision value γ_d) for simplicity. So, by letting $K \to \infty$, the number of the dictionary atoms can be learned from the training data. To model the weights $(\boldsymbol{s}_i)_{i=1}^N$, we use a zero-mean Gaussian distribution with precision value γ_s.

In order to make the dictionary discriminative for the classification purpose, we incorporate a logistic regression classifier to our probabilistic model. More precisely, if $\boldsymbol{\alpha}_t = \boldsymbol{z}_t \odot \boldsymbol{s}_t$ be the sparse code of a test instance x_t, the probability of $y_t = +1$ can be computed using the logistic sigmoid acting on a linear function of $\boldsymbol{\alpha}_t$ so that

$$P(y_t = +1 \mid \boldsymbol{z}_t, \boldsymbol{s}_t, \boldsymbol{w}, w_0) = \sigma(\boldsymbol{w}^T(\boldsymbol{z}_t \odot \boldsymbol{s}_t) + w_0), \tag{10}$$

where $\sigma(x)$ is the logistic function which is defined as

$$\sigma(x) = \frac{1}{1 + e^{-x}}. \tag{11}$$

As the probability of the two classes must sum to 1, we have $P(y_t = -1 \mid \boldsymbol{z}_t, \boldsymbol{s}_t, \boldsymbol{w}, w_0) = 1 - P(y_t = +1 \mid \boldsymbol{z}_t, \boldsymbol{s}_t, \boldsymbol{w}, w_0)$. Since the logistic function has the property that $\sigma(-x) = 1 - \sigma(x)$, we can write the class conditional probability more concisely as

$$P(y_t \mid \boldsymbol{z}_t, \boldsymbol{s}_t, \boldsymbol{w}, w_0) = \sigma\big(y_t[\boldsymbol{w}^T(\boldsymbol{z}_t \odot \boldsymbol{s}_t) + w_0]\big), \tag{12}$$

where $\boldsymbol{w} \in R^K$ and $w_0 \in R$ are the parameters of the classifier which are drawn from $\mathcal{N}(0, \gamma_w^{-1} I_K)$ and $\mathcal{N}(0, \gamma_w^{-1})$ respectively. We typically place non-informative Gamma hyper-priors on $\gamma_x, \gamma_d, \gamma_s$ and γ_w. The proposed hierarchical probabilistic model for the binary classification given the training data $(X, Y) = (\boldsymbol{x}_i, y_i)_{i=1}^N$, can be expressed as

$$P(X \mid D, Z, S, \gamma_x) \sim \prod_{j=1}^{N} \mathcal{N}(\boldsymbol{x}_j; D(\boldsymbol{z}_j \odot \boldsymbol{s}_j), \gamma_x^{-1} I_M), \tag{13}$$

$$P(\gamma_x \mid a_x, b_x) \sim Gamma(\gamma_x; a_x, b_x), \tag{14}$$

$$P(Z \mid \Pi) \sim \prod_{i=1}^{N} \prod_{k=1}^{K} Bernoulli(z_{ki}; \pi_k), \tag{15}$$

$$P(\Pi \mid a_\pi, b_\pi, K) \sim \prod_{k=1}^{K} Beta(\pi_k; a_\pi/K, b_\pi(K-1)/K), \tag{16}$$

$$P(S \mid \gamma_s) \sim \prod_{i=1}^{N} \prod_{k=1}^{K} \mathcal{N}(s_{ki}; 0, \gamma_s^{-1}), \tag{17}$$

$$P(\gamma_s \mid a_s, b_s) \sim Gamma(\gamma_s; a_s, b_s), \tag{18}$$

$$P(D \mid \gamma_d) \sim \prod_{i=1}^{M} \prod_{k=1}^{K} \mathcal{N}(d_{ik}; 0, \gamma_d^{-1}), \tag{19}$$

$$P(\gamma_d \mid a_d, b_d) \sim Gamma(\gamma_d; a_d, b_d), \tag{20}$$

$$P(Y \mid Z, S, \boldsymbol{w}, w_0) \sim \prod_{j=1}^{N} \sigma(y_j[\boldsymbol{w}^T(\boldsymbol{z}_j \odot \boldsymbol{s}_j) + w_0]), \tag{21}$$

$$P(\boldsymbol{w} \mid \gamma_w) \sim \mathcal{N}(\boldsymbol{w}; 0, \gamma_w^{-1} I_K), \tag{22}$$

$$P(\gamma_w \mid a_w, b_w) \sim Gamma(\gamma_w; a_w, b_w), \tag{23}$$

$$P(w_0 \mid \gamma_{w_0}) \sim \mathcal{N}(w_0; 0, \gamma_w^{-1}), \tag{24}$$

where $\Pi = [\pi_1, \pi_2, ..., \pi_K]$.
$\Phi = \{a_\pi, b_\pi, a_x, b_x, a_d, b_d, a_s, b_s, a_w, b_w, K\}$ are the hyper-parameters of the proposed model. The graphical representation of the probabilistic proposed model is shown in Fig. 1. For multiclass extension, we use the multinomial logistic regression classifier which is a model of the form

$$P(y_t = c \mid \boldsymbol{z}_t, \boldsymbol{s}_t, \Xi) = \frac{exp(\boldsymbol{w}_c^T(\boldsymbol{z}_t \odot \boldsymbol{s}_t))}{\sum_{c'=1}^{C} exp(\boldsymbol{w}_{c'}^T(\boldsymbol{z}_t \odot \boldsymbol{s}_t))}, \tag{25}$$

where C is the number of classes and $\Xi = [\boldsymbol{w}_1, ..., \boldsymbol{w}_C]$ are the parameters of the classifier which are drawn from multivariate zero-mean Gaussian distribution with precision value γ_w ($\boldsymbol{w}_c \sim \mathcal{N}(0, \gamma_w^{-1} I_K)$). The

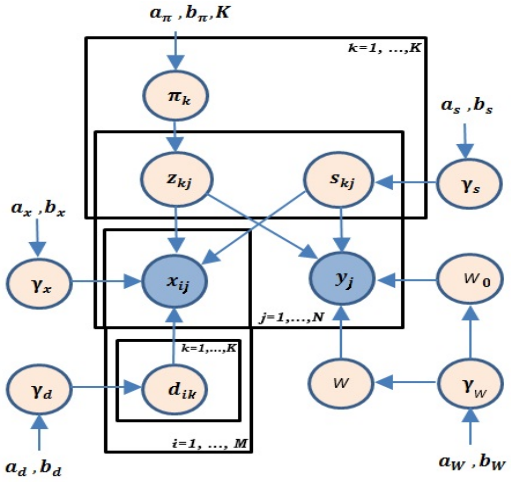

Figure 1: The graphical representation of the proposed binary classification model (blue shadings indicate observations).

hierarchical probabilistic model for the multiclass classification is the same as the model for the binary classification, except for the equations 21-24 which are replaced by

$$P(Y \mid Z, S, \Xi) \sim \prod_{j=1}^{N} \frac{exp\big(\boldsymbol{w}_{y_j}^T(\boldsymbol{z}_j \odot \boldsymbol{s}_j)\big)}{\sum_{c=1}^{C} exp(\boldsymbol{w}_c^T(\boldsymbol{z}_j \odot \boldsymbol{s}_j))}, \tag{26}$$

$$P(\Xi \mid \gamma_w) \sim \prod_{c=1}^{C} \mathcal{N}(\boldsymbol{w}_c; 0, \gamma_w^{-1} I_K), \tag{27}$$

$$P(\gamma_w \mid a_w, b_w) \sim Gamma(\gamma_w; a_w, b_w). \tag{28}$$

4.2 Posterior Inference

Due to the intractability of computing the exact posterior distribution of the hidden variables, in this section, we derive a Variational Bayesian algorithm (Beal, 2003), to approximate the posterior distribution over the hidden variables of the proposed probabilistic model given the training data.
The goal of the variational inference is to approximate the true posterior distribution over the hidden variables with a variational distribution which is closest in KL divergence to the true posterior distribution. A brief review of the VB algorithm for the exponential family distributions provided in the Supplementary Material[1] (see appendix A).
In our variational inference framework, we use the finite Beta-Bernoulli approximation, in which the number of the dictionary atoms (K) is truncated and set

[1]The supplementary Material can be downloaded from http://ce.sharif.edu/~jourabloo/papers/SM.pdf

to a finite but large number. If K is large enough, the analyzed data using this number of dictionary atoms, will reveal less than K components.
In the following two sections, we derive the variational update equations for the binary and the multiclass classification models.

4.2.1 Variational Inference For the Binary Classification

In the proposed binary classification model, the hidden variables are

$$W = \Big\{ \Pi = [\pi_1, \pi_2, ..., \pi_K], Z = [\boldsymbol{z}_1, \boldsymbol{z}_2, ..., \boldsymbol{z}_N], \\ S = [\boldsymbol{s}_1, \boldsymbol{s}_2, ..., \boldsymbol{s}_N], D = [\boldsymbol{d}_1, \boldsymbol{d}_2, ..., \boldsymbol{d}_K], \boldsymbol{w}, w_0, \\ \gamma_s, \gamma_w, \gamma_x, \gamma_d \Big\}.$$

We use a fully factorized variational distribution which is as follows

$$q(\Pi, Z, S, D, \boldsymbol{w}, w_0, \gamma_s, \gamma_w, \gamma_x, \gamma_d) = \\ \prod_{k=1}^{K} \prod_{i=1}^{M} q_{\pi_k}(\pi_k) q_{d_{ik}}(d_{ik}) \prod_{j=1}^{N} \prod_{k=1}^{K} q_{z_{kj}}(z_{kj}) q_{s_{kj}}(s_{kj}) \times \\ q_{\boldsymbol{w}}(\boldsymbol{w}) q_{w_0}(w_0) q_{\gamma_s}(\gamma_s) q_{\gamma_w}(\gamma_w) q_{\gamma_x}(\gamma_x) q_{\gamma_d}(\gamma_d).$$

It's worth noting that instead of using the parameterized variational distribution, we use the factorized form of the variational inference which is called Mean Field method (Beal, 2003). More precisely, we derive the form of the distribution $q(x)$ by optimizing the KL divergence over all possible distributions.
Based on the graphical model of Fig. 1, the joint probability distribution of the observations (training data) and the hidden variables can be expressed as

$$P(X, Y, W \mid \Phi) = \\ \prod_{j=1}^{N} \Big(P(x_j \mid \boldsymbol{z}_j, \boldsymbol{s}_j, D, \gamma_x) P(y_j \mid \boldsymbol{z}_j, \boldsymbol{s}_j, \boldsymbol{w}, w_0) \Big) \times \\ \prod_{k=1}^{K} \Big(P(\pi_k \mid a_\pi, b_\pi) \prod_{j=1}^{N} P(z_{kj} \mid \pi_k) P(s_{kj} \mid \gamma_s) \times \\ \prod_{i=1}^{M} P(d_{ik} \mid \gamma_d) \Big) P(\boldsymbol{w} \mid \gamma_w) P(w_0 \mid \gamma_w) P(\gamma_s \mid a_s, b_s) \times \\ P(\gamma_x \mid a_x, b_x) P(\gamma_w \mid a_w, b_w) P(\gamma_d \mid a_d, b_d). \quad (29)$$

In the binary classification model, all of the distributions are in the conjugate exponential family except for the logistic function. Due to the non-conjugacy between the logistic function and Guassian distribution, deriving the VB update equations in closed-form is intractable. To overcome this problem, we use the local lower bound to the sigmoid function proposed by (Jaakkola et al., 2000), which states that for any $x \in R$ and $\xi \in [0, +\infty]$

$$\frac{1}{1 + exp(-x)} \geq \sigma(\xi) exp\Big((x - \xi)/2 - \lambda(\xi)(x^2 - \xi^2) \Big), \quad (30)$$

where,

$$\lambda(\xi) = \frac{-1}{2\xi} \big(\frac{1}{1 + exp(-\xi)} - \frac{1}{2} \big). \quad (31)$$

ξ is the free variational parameter which is optimized to get the tightest possible bound. Hence, we replace each sigmoid factor in the joint probability distribution (equation 29) with the above lower bound (equation 30), then we use the EM algorithm to optimize the factorized variational distribution and the free parameters ($\boldsymbol{\xi} = \{\xi_1, \xi_2, ..., \xi_N\}$) which computes the variational posterior distribution in the E-step and maximizes the free parameters in the M-step. All update equations are available in the Supplementary Material (see appendix B).

4.2.2 Variational Inference For the Multiclass Classification

In the proposed multiclass classification model, because of non-conjugacy between the multinomial logistic regression function (equation 25) and the Gaussian distribution, deriving the VB update equations in closed-form is intractable. To tackle this non-conjugacy problem, we utilize the following simple inequality which was originally proposed by (Bouchard, 2007), which states that for every $\{\beta_c\}_{c=1}^{C} \in R$ and $\alpha \in R$,

$$log \big(\sum_{c=1}^{C} e^{\beta_c} \big) \leq \alpha + \sum_{c=1}^{C} log \big(1 + e^{\beta_c - \alpha} \big). \quad (32)$$

If we replace x with $\alpha - \beta_c$ in the equation 30, and take the logarithm of the both sides of that equation, we have

$$\log(1 + e^{\beta_c - \alpha}) \leq \lambda(\xi) \big((\beta_c - \alpha)^2 - \xi^2 \big) - \\ \log \sigma(\xi) + \big((\beta_c - \alpha) + \xi \big)/2. \quad (33)$$

Then, by replacing each term in the summation of the right hand side of the equation 32 with the upper bound of the equation 33, we have

$$\log \big(\sum_{c=1}^{C} e^{\beta_c} \big) \leq \alpha + \sum_{c=1}^{C} \log \big(1 + e^{\beta_c - \alpha} \big) \\ \leq \sum_{c=1}^{C} \Big(\lambda(\xi_c) \big((\beta_c - \alpha)^2 - \xi_c^2 \big) - \log \sigma(\xi_c) \Big) + \\ \alpha + \frac{1}{2} \sum_{c=1}^{C} (\beta_c - \alpha + \xi_c). \quad (34)$$

We utilize the above inequality for approximating the denominator of the right hand side of the equation 26. So, for the proposed multiclass classification model, the free parameters are $\{\{\alpha_i\}_{i=1}^{N}, \{\xi_{ij}\}_{i=1,j=1}^{N,C}\}$.
We derive an EM algorithm that computes the variational posterior distribution in the E-step and maximizes the free parameters in the M-step. Details of the update equations are available in the Supplementary Material (see appendix C).

4.3 Class Label Prediction

After computing the posterior distribution, in order to determine the target class-label y_t of a given test instance x_t, we first compute the predictive distribution of the target class label given the test instance by integrating out the hidden variables ($\{D, \gamma_x, \boldsymbol{z}_t, \boldsymbol{s}_t, \boldsymbol{w}, w_0\}$ for binary classification model, and $\{D, \gamma_x, \boldsymbol{z}_t, \boldsymbol{s}_t, [\boldsymbol{w}_c]_{c=1}^{C}\}$ for multiclass classification model), then we pick the label with the maximum probability value. For binary classification, this procedure can be formulated as

$$\hat{y}_t = argmax_{y_t \in \{-1,1\}} P(y_t \mid x_t, T), \tag{35}$$

where $T = (\boldsymbol{x}_j, y_j)_{j=1}^{N}$ is the training data. $P(y_t \mid x_t, T)$ can be computed as

$$\begin{aligned}
&P(y_t \mid x_t, T) = \\
&\sum_{\boldsymbol{z}_t} \iiint P(y_t, \boldsymbol{s}_t, \boldsymbol{z}_t, \boldsymbol{w}, w_0 \mid x_t, T) d\boldsymbol{s}_t \, d\boldsymbol{w} \, dw_0 \\
&= \sum_{\boldsymbol{z}_t} \iiint P(y_t \mid \boldsymbol{s}_t, \boldsymbol{z}_t, \boldsymbol{w}, w_0, x_t, T) \times \\
&\qquad P(\boldsymbol{s}_t, \boldsymbol{z}_t, \boldsymbol{w}, w_0 \mid x_t, T) d\boldsymbol{s}_t \, d\boldsymbol{w} \, dw_0 \\
&= \sum_{\boldsymbol{z}_t} \iiint \sigma(y_t[\boldsymbol{w}^T(\boldsymbol{s}_t \odot \boldsymbol{z}_t) + w_0]) P(\boldsymbol{s}_t, \boldsymbol{z}_t \mid x_t, T) \times \\
&\qquad P(\boldsymbol{w} \mid T) P(w_0 \mid T) d\boldsymbol{s}_t \, d\boldsymbol{w} \, dw_0 \\
&\approx \sum_{\boldsymbol{z}_t} \iiint \sigma(y_t[\boldsymbol{w}^T(\boldsymbol{s}_t \odot \boldsymbol{z}_t) + w_0]) P(\boldsymbol{s}_t, \boldsymbol{z}_t \mid x_t, T) \times \\
&\qquad q^*(\boldsymbol{w}) q^*(w_0) d\boldsymbol{s}_t \, d\boldsymbol{w} \, dw_0,
\end{aligned} \tag{36}$$

where we replaced $P(\boldsymbol{w} \mid T)$ and $P(w_0 \mid T)$ with the approximate posterior distributions $q^*(\boldsymbol{w})$ and $q^*(w_0)$ respectively.
Since the above expression cannot be computed in closed form, we resort to Monte Carlo sampling to approximate that expression. In other words, we approximate the distribution $P(\boldsymbol{s}_t, \boldsymbol{z}_t \mid x_t, X, Y) q^*(\boldsymbol{w}) q^*(w_0)$ with l samples, then we compute $P(y_t \mid x_t, T)$ as

$$P(y_t \mid x_t, T) \approx \frac{1}{l} \sum_{l} \sigma(y_t[(\boldsymbol{w}^l)^T(\boldsymbol{s}_t^l \odot \boldsymbol{z}_t^l) + w_0^l]), \tag{37}$$

where r^l is the l-th sample of the hidden variable r.
Since the approximate posterior distributions $q^*(\boldsymbol{w})$ and $q^*(w_0)$ are Gaussian (see the appendix B of the Supplementary Material), sampling from these distributions is straightforward. $P(\boldsymbol{s}_t, \boldsymbol{z}_t \mid x_t, T)$ can be computed as

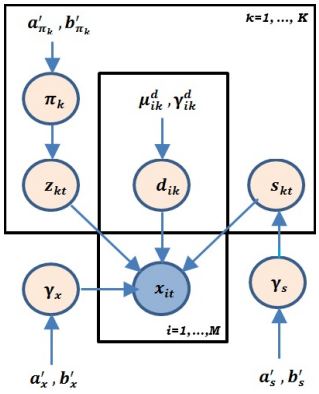

Figure 2: The graphical model of the Gibbs sampling method.

$$P(\boldsymbol{s}_t, \boldsymbol{z}_t \mid x_t, T) = \frac{P(x_t \mid \boldsymbol{s}_t, \boldsymbol{z}_t, T) P(\boldsymbol{s}_t, \boldsymbol{z}_t \mid T)}{P(x_t \mid T)}, \tag{38}$$

which cannot be directly sampled from. Therefore, to sample from $P(\boldsymbol{s}_t, \boldsymbol{z}_t \mid x_t, T)$, we sample from $P(\boldsymbol{s}_t, \boldsymbol{z}_t, D, \Pi, \gamma_s, \gamma_x \mid x_t, T)$ based on the Gibbs sampling method (Robert et al., 2004), then simply ignore the values for $D, \Pi, \gamma_s, \gamma_x$ in each sample. The graphical model in Fig. 2 shows all the relevant parameters and conditional dependence relationships, by which the Gibbs sampling equations are derived. The detailes of the Gibbs sampling equations are avalable in the Supplementary Material (see appendix D). It should be noted that the parameters of the variables in Fig. 2 are the updated parameters of the variational posterior distribution which were computed using VB algorithm (see appendix B of the Supplementary Material). For multiclass extension, the posterior distribution over the class label of a test instance x_t can be approximated as

$$P(y_t = c \mid x_t, T) \approx \frac{1}{l} \sum_{l} \frac{exp\big((\boldsymbol{w}_c^l)^T(\boldsymbol{z}_t^l \odot \boldsymbol{s}_t^l)\big)}{\sum_{c'=1}^{C} exp((\boldsymbol{w}_{c'}^l)^T(\boldsymbol{z}_t^l \odot \boldsymbol{s}_t^l))}, \tag{39}$$

where $\{\boldsymbol{w}_c^l\}_{c=1}^{C}$ are the l-th samples of the approximate posterior distributions $\{q^*(\boldsymbol{w}_c)\}_{c=1}^{C}$, and $(\boldsymbol{z}_t^l, \boldsymbol{s}_t^l)$ is the l-th sample of the posterior distribution $P(\boldsymbol{s}_t, \boldsymbol{z}_t \mid x_t, T)$.
Sampling from $\{q^*(\boldsymbol{w}_c)\}_{c=1}^{C}$ is straightforward. Sampling from the distribution $P(\boldsymbol{s}_t, \boldsymbol{z}_t \mid x_t, T)$ for multiclass classification model is the same as sampling from that distribution for the binary classification model (see appendix D).

5 Experimental Results

In this section, we verify the performance of the proposed method on various applications such as digit recognition, face recognition, and spoken letter recognition. For applications which include more than two classes, we use one versus all binary classification (one classifier for each class) based on the proposed binary classification model (PM_b) as well as the proposed multiclass classification model (PM_m).
All of the experimental results are averaged over several runs of randomly generated splits of the data. Moreover, in all experiments, all Gamma priors are set as Gamma $(10^{-6}, 10^{-6})$ to make the prior distributions uninformative. The parameters a_π, b_π of the Beta distribution are set with $a_\pi = K$ and $b_\pi = K/2$ (many other settings of a_π and b_π yield similar results). For the Gibbs sampling inference, we discard the initial 500 samples (burn-in period), and collect the next 1000 samples to present the posterior distribution over the sparse code of a test instance.

5.1 Digit Recognition

We apply the proposed method on two handwritten digit recognition datasets MNIST (LeCun et al., 1998), and USPS (Hull, 1994). The MNIST dataset consists of 70000 28×28 images, and the USPS dataset composes of 9298 16×16 images. We reduced the dimensionality of both datasets by retaining only the first 100 principal components to speed up training. Details of the experiments for the digit databases are summarized in Table 1.
We compare the proposed models (PM_b, PM_m) with state of the art methods such as the Sparse Representation for Signal Classification (denoted by SRSC) (Huang et al., 2006), the supervised DL method with generative training and discriminative training (denoted by SDL-G and SDL-D) (Mairal, 2009), and the Fisher Discriminant Dictionary learning (denoted by FDDL) (Yang et al., 2011). Furthermore, the results of two classical classification methods, K-nearest neighbor (K=3) and linear SVM are also reported. The average recognition accuracies (over 10 runs) together with the standard deviation is shown in Table 2, from which we can see that the proposed methods outperform the other methods approximately by 3.5%.
The improvement in performance compared to other methods is due to the fact that the number of the training data points are small. Precisely speaking, the methods SRSC, SDL-G, SDL-D and FDDL are optimization based learners (MAP learners from probabilistic point of view) which can overfit small-size training data. In contrast, the proposed method does weighted averaging over the dictionary, the sparse codes, and the classifier, weighted by their posterior distributions and hence is relatively immune to overfitting. From Table 2, We also observe that the one versus all binary classifier (PM_b) has slightly better performance than the multiclass classifier (PM_m), but has more computational complexity than the multiclass classifier. Moreover, because of small number of the training data, generative SDL (SDL-G) has better performance than discriminative SDL (SDL-D).
In order to demonstrate the ability of the proposed method to learn the number of the dictionary atoms as well as the dictionary elements, we plot the sorted values of $\langle \Pi \rangle$ For the MNIST dataset, inferred by the algorithm (Fig. 3). As can be seen, the algorithm inferred a sparse set of factors, fewer than the 250 initially provided.
To further analyze the performance of the proposed method on various number of training data points, we illustrate the change in the classification accuracy on the MNIST digit dataset over successive iterations, for which we add more labeled samples at each iteration. Fig. 5 plots the recognition rates of different methods versus different number of training data points, from which we can see that improvement in the accuracy of the optimization based methods (FDDL, SDL-G, SDL-D) is larger than the proposed multi class classification method. This is due to the fact that when the number of the training data grows, the likelihood of overfitting the training data is reduced.
We also plot the sorted values of $\langle \Pi \rangle$ For the MNIST dataset for 1000 training data points, inferred by the algorithm (Fig. 4). As can be seen, when the number of the training data points increases, we need more dictionary atoms to capture the complexity of the data points.

Table 1: Properties of the digit datasets and experimental parameters

	MNIST	USPS
examples (train)	250	250
examples (test)	1000	1000
classes	10	10
input dimensions	784	256
features after PCA	100	100
runs	10	10
K (number of dictionary atoms)	250	250

5.2 Face Recognition

We then perform the face recognition task on the widely used extended Yale B (Lee et al., 2005), and AR (Martinez et al., 1998), face databases. The extended Yale B database consists of $2,414$ frontal-face

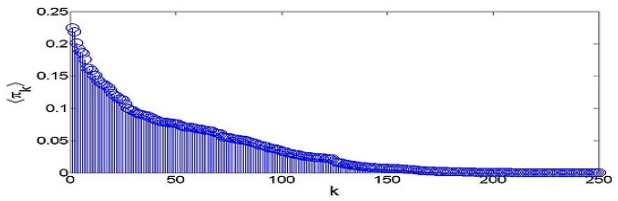

Figure 3: Inferred $\langle \Pi \rangle$ for the MNIST dataset (250 training samples).

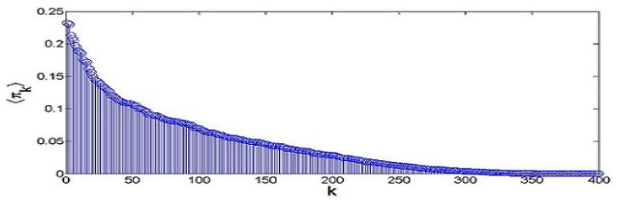

Figure 4: Inferred $\langle \Pi \rangle$ for the MNIST dataset (1000 training samples).

images from 38 individuals (about 64 images per individual), and the AR database consists of over 4,000 frontal images from 126 individuals which was generated in two sessions, each of them consists of 14 images per individual. The extended Yale B and AR images are normalized to 54×48 and 60×40 respectively. We use the Eigenface (Turk et al., 1991), with dimension 300 for both extended Yale B and AR datasets. For the extended Yale B database, each training set comprised of 20 images per individual, and the remaining images were used to test. For AR dataset, seven images from the first session are used for training, the remaining seven images from the second session are used for testing. Details of the experiments for the face databases are summarized in Table 3.

To illustrate the superiority of the proposed models, we compare our methods with the best result of discriminative KSVD (denoted DKSVD) (Zhang et al., 2010), dictionary learning with structure incoherence (denoted DLSI) (Ramirez et al., 2010), FDDL, K-NN, and SVM. The results of these experiments on the face databases are listed in Table 4. Again, due to the lack of enough number of the training data, our methods have better performance than the other methods.

5.3 Spoken Letter Recognition

Finally, we apply our method on the Isolet database (Blake et al., 1998), from UCI Machine Learning Repository which consists of 6238 examples and 26 classes corresponding to letters of the alphabet. We reduced the input dimensionality (originally at 617) by projecting the data onto its leading 100 principal components. We use 250 samples for training and 1000 samples for testing. The truncation level K for this experiment is set to 400. We also use only a subset of 10 classes of the Isolet dataset. The average recognition accuracies (over 10 runs) is shown in Table 5, from which we can see that the proposed methods outperform the other methods approximately by 3%.

Table 2: Classification accuracy of different methods on Digit datasets.

	MNIST	USPS
SVM	79.3± 2.0	80.7± 1.5
3-NN	80.4± 1.4	81.4± 2.1
SDL-D	80.2± 2.1	83.5± 1.9
SRSC	78.9± 1.2	80.2± 1.2
SDL-G	81.3± 1.4	84.0± 1.3
FDDL	81.1± 1.8	83.8± 1.7
PM_m	**84.9** ± 1.3	**86.6** ± 1.0
PM_b	**85.8** ± 1.1	**87.4** ± 0.9

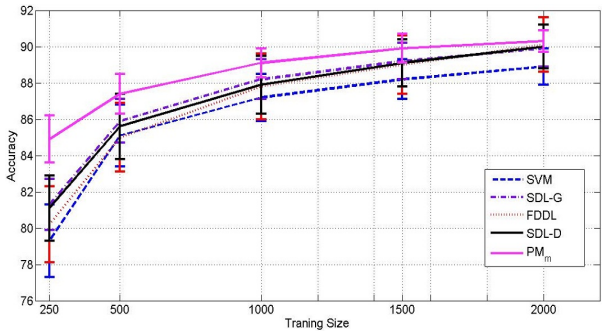

Figure 5: The recognition rate of different methods versus the number of training data for MNIST dataset.

6 Conclusion

We developed new models for the dictionary learning based pattern classification tasks based on the Beta-Bernoulli process, and a new algorithm based on the variational inference which allows our method scales to large data sets. We also used Bayesian prediction rule to determine the label of the unknown samples which

Table 3: Properties of data sets and experimental parameters.

	E-Yale B	AR
examples (train)	760	700
examples (test)	1654	700
classes	38	100
input dimensions	2592	2400
features after PCA	300	300
runs	10	10
K	500	600

Table 4: Classification accuracy of different methods on Face datasets.

	E-Yale B	AR
SVM	88.8± 1.2	87.1± 1.3
3-NN	65.9± 1.8	73.5± 2.1
DLSI	85.0± 1.6	73.7± 1.4
DKSVD	75.3± 1.4	85.4± 1.2
FDDL	91.9± 1.0	92.0± 1.3
PM_m	**94.7**± 1.3	**94.2**± 1.2
PM_b	**95.1**± 1.1	**94.9**± 1.0

Table 5: Classification accuracy of different methods on Isolet dataset.

Method	SVM	DLSI	FDDL	PM_b	PM_m
Accuracy	90.9	88.6	90.5	**93.3**	**92.9**

makes our method be suitable for small size training data. The experimental results on digit recognition, face recognition and spoken letter classification clearly demonstrated the superiority of the proposed model to many state-of-the-art dictionary learning based classification methods. For the future work, we will apply our method on the semi-supervised classification tasks.

References

M. Aharon, M. Elad, and A. Bruckstein (2006). k -svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 4311-4322.

M. Beal (2003). Variational algorithms for approximate bayesian inference. *Doctoral dissertation, University College London.*

C.L. Blake, and C.J. Merz (1998). Uci repository of machine learning databases. *University of California, Department of Information and Computer Science.*

D. Bohning (1992). Multinomial logistic regression algorithm. *Annals Inst. Stat. Math.*

G. Bouchard (2007). Efficient bounds for the softmax function.*In NIPS.*

N.L. Hjort (1990). Nonparametric bayes estimators based on beta processes in models for life history data. *Annals of Statistics*, 1259-1294.

K. Huang, and S. Aviyente (2006). Sparse representation for signal classification. *In NIPS.*

J. J. Hull (1994). A database for handwritten text recognition research. *IEEE Trans. Pattern Anal. Mach. Intell*, 550-554.

T. Jaakkola, and M. I. Jordan (2000). Bayesian parameter estimation via variational methods. *Statistics and Computing*, 25-37.

J. F. C. Kingman (1993). Poisson Processes. *Oxford University Press.*

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner (1998). Gradient-based learning applied to document recognition. *Proc. of the IEEE.*

K. Lee, J. Ho, and D. Kriegman (2005). Acquiring linear subspaces for face recognition under variable lighting. *In IEEE TPAMI*, 27(5): 684-698.

J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman (2009). Supervised dictionary learning. *In NIPS.*

A. Martinez, and R. benavente (1998). The AR face database. *CVC Tech. Report.*

J. Paisley, and L. Carin (2009). Nonparametric factor analysis with beta process priors. *In Proc. International Conference on Machine Learning.*

I. Ramirez, P. Sprechmann, and G. Sapiro (2010). Classification and clustering via dictionary learning with structured incoherence and shared features. *In CVPR.*

C.P. Robert, and G. Casella (2004). Monte carlo statistical methods. *Springer Verlag.*

R. Thibaux, and M. I. Jordan (2007). Hierarchical beta processes and the Indian buffet process. *In AISTATS.*

M. Turk, and A. Pentland (1991). Eigenfaces for recognition. *J. Cognitive Neuroscience*, 3(1):71-86.

J. Wright, Y. Ma, J. Mairal, G. Sapiro, T.S. Huang, and S. Yan (2010). Sparse representation for computer vision and pattern recognition, *Proceedings of the IEEE.*

J. Wright, A.Y. Yang, A. Ganesh, S.S. Sastry, and Y. Ma (2009). Robust Face Recognition via Sparse Representation. *IEEE TPAMI*, 210-227.

M. Yang, L. Zhang, X. Feng, and D. Zhang (2011). Fisher discrimination dictionary learning for sparse representation. *In ICCV.*

M. Yang, L. Zhang, J. Yang, and D. Zhang (2010). Metaface learning for sparse representation based face recognition. *In ICIP.*

Q. Zhang, and B.X. Li (2010). Discriminative K-SVD for dictionary learning in face recognition. *In CVPR.*

M. Zhou, H. Chen, J. Paisley, L. Ren, G. Sapiro, and L. Carin (2009). Non-Parametric Bayesian Dictionary Learning for Sparse Image Representations. *In NIPS.*