Joint Discriminative Bayesian Dictionary and Classifier Learning

Naveed Akhtar[1], Ajmal Mian[2] and Fatih Porikli[3]
Australian National University[1,3], University of Western Australia[2]
{naveed.akhtar,fatih.porikli}@anu.edu.au, ajmal.mian@uwa.edu.au

Abstract

*We propose to jointly learn a Discriminative Bayesian dictionary along a linear classifier using coupled Beta-Bernoulli Processes. Our representation model uses separate base measures for the dictionary and the classifier, but associates them to the class-specific training data using the same Bernoulli distributions. The Bernoulli distributions control the frequency with which the factors (*e.g.* dictionary atoms) are used in data representations, and they are inferred while accounting for the class labels in our approach. To further encourage discrimination in the dictionary, our model uses separate (sets of) Bernoulli distributions to represent data from different classes. Our approach adaptively learns the association between the dictionary atoms and the class labels while tailoring the classifier to this relation with a joint inference over the dictionary and the classifier. Once a test sample is represented over the dictionary, its representation is accurately labeled by the classifier due to the strong coupling between the dictionary and the classifier. We derive the Gibbs Sampling equations for our joint representation model and test our approach for face, object, scene and action recognition to establish its effectiveness.*

1. Introduction

Dictionary Learning [25] is a well-established signal representation technique, employed in compressive sensing [4], image restoration [5] and morphological component analysis [3]. A *dictionary* is a set of basis vectors (a.k.a. *atoms*), learned to represent training data. More recently, this technique has also shown great potential for multiple classification tasks [9], [10], [13], [30], [31], [32]. Dictionaries learned for classification (a.k.a. *discriminative dictionaries*) not only represent the training data from different classes accurately, but they also render their representations easily classifiable by a suitable classifier.

Generally, discriminative dictionary learning approaches either restrict subsets of the dictionary atoms to represent training data of specific classes only [15], [19], [24], [28]; or they force the representations of the data over the entire dictionary to become discriminative [10], [13], [29], [32]. In some instances, a dictionary is also learned as a concatenation of class-specific atoms and atoms to jointly represent the training data from all classes [19], [26]. In any case, the relationship between the dictionary atoms and the class labels remains the key for effective discriminative dictionaries [11]. Nevertheless, adaptive learning of this relation is still a largely open research problem [2], [30]. Subsequently, tailoring a classifier to the adaptively learned relation generally remains unaddressed.

In this work, we present a Bayesian approach[1] to address the above problems. We propose a Beta-Bernoulli process [17] based representation model that relates the dictionary atoms with the class labels using Bernoulli distributions, learned adaptively in our approach. The same distributions also associate parameters of a classifier to the class label vectors of the training data. The dictionary and the classifier are inferred simultaneously under a joint inference process, however using separate base measures. This gives our approach the flexibility to learn both the dictionary and the classifier accurately while keeping them strongly coupled under the Bernoulli distributions. For the underlying Beta-Bernoulli processes, the Bernoulli distributions signify the frequency of the factor (*e.g.* dictionary atoms) usage in data representation. We use separate sets of Bernoulli distributions for representing data from different classes, promoting frequent use of its own popular factors for each class. This further improves the discriminability of the dictionary learned under a Beta-Bernoulli process [2].

When test samples are encoded over the dictionary, they use the popular atoms for their correct class more frequently. Since the classifier has a strong coupling with the dictionary and it is already tailored to the popularity of the atoms, it accurately predicts the class labels of the test representations. We derive Gibbs Sampling equations for our model and test our approach on benchmark datasets for face [8], [16], object [6], scene [12] and action recognition [20]. Experiments show that our approach consistently improves the classification accuracy over the existing state-of-the-art dictionary learning and sparse representation based classification approaches.

[1] Download Matlab code from http://staffhome.ecm.uwa.edu.au/~00053650/code.html

2. Problem settings and preliminaries

Let the i^{th} training sample of the c^{th} class be expressed as $\mathbf{y}_i^c = \boldsymbol{\Phi}\boldsymbol{\alpha}_i^c + {}^{\mathrm{y}}\boldsymbol{\epsilon}_i$, where $\boldsymbol{\Phi} \in \mathbb{R}^{L \times |\mathcal{K}|}$ is an unknown dictionary, $\boldsymbol{\alpha}_i^c \in \mathbb{R}^{|\mathcal{K}|}$ is the representation of the sample over the dictionary and ${}^{\mathrm{y}}\boldsymbol{\epsilon}_i \in \mathbb{R}^L$ denotes noise. The k^{th} atom $\boldsymbol{\varphi}_k$ of the dictionary is indexed in the set $\mathcal{K} = \{1, ..., K\}$, whose cardinality $|\mathcal{K}|$ is also not known beforehand. Note that, $|\mathcal{K}|$ defines the dictionary size. The training samples of the c^{th} class are indexed in a set $\mathcal{I}_c$; and $\sum_{c=1}^{C} |\mathcal{I}_c| = N$, where C denotes the total number of classes. We follow the convention that absence of the superscript 'c' implies that no distinction is being made between different classes for the variable under consideration. For instance, ${}^{\mathrm{y}}\boldsymbol{\epsilon}_i$ is not annotated with 'c' because the training samples from all the classes are considered to have the same noise statistics[2].

A dictionary learning approach generally solves the following optimization problem to learn sparse representation:

$$< \boldsymbol{\Phi}, \boldsymbol{\alpha}_i >= \min_{\boldsymbol{\Phi}, \boldsymbol{\alpha}} ||\mathbf{y}_i - \boldsymbol{\Phi}\boldsymbol{\alpha}_i||_2^2 \;\; s.t. \;\; \forall i, ||\boldsymbol{\alpha}_i||_p \leq t, \quad (1)$$

where, $||.||_p$ denotes the ℓ_p-norm and 't' is a predefined constant. Using thus computed $\boldsymbol{\alpha}_i$, it is also possible to learn a linear classifier $\boldsymbol{\Psi} \in \mathbb{R}^{C \times |\mathcal{K}|}$ by solving:

$$< \boldsymbol{\Psi} >= \min_{\boldsymbol{\Psi}} \sum_{i=1}^{N} \mathcal{L}\{\mathbf{h}_i, f(\boldsymbol{\alpha}_i, \boldsymbol{\Psi})\} + \lambda ||\boldsymbol{\Psi}||_F^2, \quad (2)$$

where $\mathcal{L}$ is the loss function, λ is the regularizer, $\mathbf{h}_i \in \mathbb{R}^C$ denotes the class label for $\mathbf{y}_i$ and $f(.)$ results in the predicted label. In these settings, a test sample can be classified by first computing its representation $\widehat{\boldsymbol{\alpha}}$ over the learned dictionary and then classifying $\widehat{\boldsymbol{\alpha}}$ using $\boldsymbol{\Psi}$. However, since the dictionary is learned in an unsupervised manner, the classification performance is expected to remain sub-optimal. To overcome this issue, Zhang and Li [32], followed by Jiang et al. [10], proposed to learn the classifier jointly with the dictionary in a supervised fashion. However, the performance of their approaches strongly depend on the used dictionary size, because the accuracy of data representation actively depends on this parameter [27]. Moreover, those approaches must prefix the relationship between the dictionary atoms and the class labels, which is not an attractive machine learning strategy.

Paisley and Carin [17] proposed a Beta-Bernoulli process that can be used to learn a dictionary in a non-parametric manner, thereby automatically inferring the appropriate dictionary size. With its base measure $\hbar_0$ and parameters $a, b > 0$, a finite representation of Beta Process is given as follows [17]:

$$\hbar = \sum_{k \in \mathcal{K}} \pi_k \delta_{\boldsymbol{\varphi}_k}(\boldsymbol{\varphi}); \quad (3)$$

$$\pi_k \sim \text{Beta}\left(\pi_k \middle| \frac{a}{K}, \frac{b(K-1)}{K}\right); \;\; \boldsymbol{\varphi}_k \sim \hbar_0,$$

where $\delta_{\boldsymbol{\varphi}_k}(\boldsymbol{\varphi}) = 1$ when $\boldsymbol{\varphi} = \boldsymbol{\varphi}_k$ and 0 otherwise. A draw $\hbar$ from the process is a set of K probabilities $\pi_{k \in \mathcal{K}}$, each associated with a $\boldsymbol{\varphi}_{k \in \mathcal{K}}$ that is drawn *i.i.d.* from the base measure $\hbar_0$. Considering π_k to be a Bernoulli distribution parameter, we can use $\hbar$ to draw a binary vector $\mathbf{z} \in \mathbb{R}^{|\mathcal{K}|}$ such that its k^{th} coefficient follows Bernoulli(π_k).

Drawing N binary vectors $\mathbf{z}_{i \in \{1,...,N\}}$ under $\mathcal{B} = \{\text{Bernoulli}(\pi_k) : k \in \mathcal{K}\}$ using the Beta-Bernoulli Process, the training data may be factorized as: $\mathbf{y}_i \approx \boldsymbol{\Phi}\mathbf{z}_i, \forall i$, where the atoms $\boldsymbol{\varphi}_k$ of the dictionary $\boldsymbol{\Phi}$ are the base measure draws. In the limit $|\mathcal{K}| \to \infty$, the number of the non-zero elements in $\mathbf{z}_i$ is itself a draw from Poisson$(\frac{a}{b})$ [17] that controls the dictionary size. Notice that the vectors $\mathbf{z}_i$ relate the dictionary atoms to the training data following the set $\mathcal{B}$, such that, the k^{th} distribution in this set signifies the frequency of the k^{th} atom usage in the data expansion. Recently, this relation was exploited to induce discriminability in a Bayesian dictionary [2]. In that approach, for each class, $\mathbf{z}_i$ was sampled under a separate $\mathcal{B}$ for the factorization, that encouraged the dictionary atoms to become more discriminative. However, in that work, the classifier to be used with the dictionary must be learned separately because the model does not support the joint learning of the two. This weakens the coupling between the dictionary and the classifier. The separately learned classifier also fails to influence the adaptively learned association between the dictionary atoms and the class labels. Moreover, the inference process is unable to benefit from the class label vectors, thereby falling short on exploiting the full potential of a supervised learning process.

3. Proposed approach

In this paper, we propose to jointly learn a discriminative Bayesian dictionary with a linear classifier in a fully supervised manner, using two coupled Beta-Bernoulli Processes [17]. To learn the dictionary, we use separate draws of a finite Beta-Bernoulli Process for each class but use the same base measure. To jointly learn the classifier with the dictionary, we employ a second Beta-Bernoulli Process that uses the same sets of the Bernoulli distributions as used by the dictionary learning process, but its own base measure to draw the classifier parameters. The Bernoulli distributions are adaptively learned in our approach while accounting for the class labels of the training data. Moreover, they directly influence the dictionary and the classifier parameters alike during the joint inference, which results in learn-

[2] We also tested the approach for different noise statistics for each class but the performance generally remained very similar. We avoid unnecessary modeling complexity by assuming the same noise statistics.

ing of an accurate dictionary coupled with an effective classifier. We use Gibbs Sampling[3] to perform the Bayesian inference. This work addresses the problem of joint dictionary and classifier learning for Beta-Bernoulli Process for the first time.

3.1. The model

Representing data only as binary combinations of the basis vectors is restrictive. Therefore, we factorize the i^{th} training sample of the c^{th} class as: $\mathbf{y}_i^c = \mathbf{\Phi}(\mathbf{z}_i^c \odot \mathbf{s}_i^c) + {}^{\text{y}}\boldsymbol{\epsilon}_i$, where $\mathbf{s}_i^c \in \mathbb{R}^{|\mathcal{K}|}$ denotes a weight vector such that $\mathbf{z}_i^c \odot \mathbf{s}_i^c = \boldsymbol{\alpha}_i^c$, here, $\odot$ represents the Kronecker product. This factorization is possible under a weighted Beta-Bernoulli Process, as shown below. For each training sample, we also factorize the corresponding class label vector $\mathbf{h}_i^c$ as follows: $\mathbf{h}_i^c = \mathbf{\Psi}(\mathbf{z}_i^c \odot \mathbf{t}_i^c) + {}^{\text{h}}\boldsymbol{\epsilon}_i$. We propose the following joint hierarchical Bayesian model for simultaneous dictionary and classifier learning:

$$\begin{gathered}
\forall i \in \mathcal{I}_c \text{ and } \forall k \in \mathcal{K} = \{1, ..., K\}: \\
\mathbf{y}_i^c = \mathbf{\Phi}(\mathbf{z}_i^c \odot \mathbf{s}_i^c) + {}^{\text{y}}\boldsymbol{\epsilon}_i \qquad \mathbf{h}_i^c = \mathbf{\Psi}(\mathbf{z}_i^c \odot \mathbf{t}_i^c) + {}^{\text{h}}\boldsymbol{\epsilon}_i \\
z_{ik}^c \sim \text{Bernoulli}(z_{ik}^c | \pi_{k_o}^c) \\
\pi_k^c \sim \text{Beta}(\pi_k^c | a_o/K, b_o(K-1)/K) \\
s_{ik}^c \sim \mathcal{N}(s_{ik}^c | 0, 1/\lambda_{s_o}^c) \qquad t_{ik}^c \sim \mathcal{N}(t_{ik}^c | 0, 1/\lambda_{t_o}^c) \\
\boldsymbol{\varphi}_k \sim \mathcal{N}(\boldsymbol{\varphi}_k | \mathbf{0}, \mathbf{\Lambda}_{\varphi_o}^{-1}) \qquad \boldsymbol{\psi}_k \sim \mathcal{N}(\boldsymbol{\psi}_k | \mathbf{0}, \mathbf{\Lambda}_{\psi_o}^{-1}) \\
{}^{\text{y}}\boldsymbol{\epsilon}_i \sim \mathcal{N}({}^{\text{y}}\boldsymbol{\epsilon}_i | \mathbf{0}, 1/\lambda_{y_o} \mathbf{I}_L) \quad {}^{\text{h}}\boldsymbol{\epsilon}_i \sim \mathcal{N}({}^{\text{h}}\boldsymbol{\epsilon}_i | \mathbf{0}, 1/\lambda_{h_o} \mathbf{I}_C).
\end{gathered} \tag{4}$$

In Eq. (4), λ and $\mathbf{\Lambda}$ represent the Gaussian distribution precision parameters, $\boldsymbol{\psi}_k \in \mathbb{R}^C$ is the k^{th} column of $\mathbf{\Psi}$, $\mathbf{I}_Q$ denotes an identity matrix in $\mathbb{R}^{Q \times Q}$, $\mathbf{0}$ represents a vector of zeros with an appropriate dimension and the subscript 'o' indicates that the associated parameter belongs to a prior distribution.

In the proposed model, both $\mathbf{y}_i^c$ and $\mathbf{h}_i^c$ use the same $\mathbf{z}_i^c$, whose k^{th} coefficient z_{ik}^c is drawn from a Bernoulli distribution, with a conjugate Beta prior. On the other hand, the coefficients of the weight vectors $\mathbf{s}_i^c$, $\mathbf{t}_i^c \in \mathbb{R}^{|\mathcal{K}|}$ are drawn from separate Gaussian distributions. Similarly, the dictionary atoms and the classifier parameters are also drawn from distinct multivariate Gaussians. This allows the factorization of $\mathbf{y}_i^c$ and $\mathbf{h}_i^c$ to remain accurate while being strongly coupled. The model is also flexible to allow the additive noise/modeling error for $\mathbf{y}_i^c$ and $\mathbf{h}_i^c$ to be the samples of different distributions. We further place the following non-informative Gamma hyper-priors over the precision parameters of the distributions: $\lambda_s^c, \lambda_t^c \sim \text{Gam}(c_o, d_o)$ and $\lambda_y, \lambda_h \sim \text{Gam}(e_o, f_o)$. The graphical representation of the proposed model is given in Fig. 1. We also provide the analytical expression for the joint probability distribution of the model in the supplementary material of the paper.

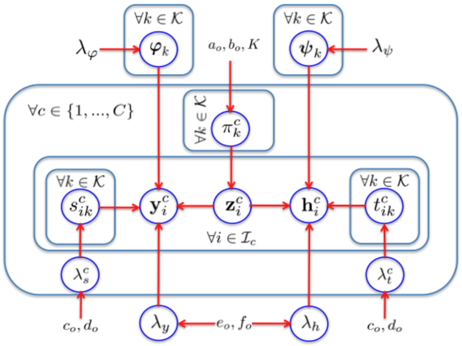

Figure 1. Factor graph representation of the proposed Bayesian model for joint learning of a classifier with a dictionary.

According to the proposed model, the covariances of $\mathbf{y}_i^c$ and $\mathbf{h}_i^c$ are given as $\mathbb{E}[\mathbf{y}_i^c \mathbf{y}_i^{c\intercal}] = \frac{aK}{a+b(K-1)} \frac{\mathbf{\Lambda}_\varphi^{-1}}{\lambda_s} + \frac{\mathbf{I}_L}{\lambda_y}$ and $\mathbb{E}[\mathbf{h}_i^c \mathbf{h}_i^{c\intercal}] = \frac{aK}{a+b(K-1)} \frac{\mathbf{\Lambda}_\psi^{-1}}{\lambda_t} + \frac{\mathbf{I}_C}{\lambda_h}$, respectively. Recall, that K signifies the number of factors $\boldsymbol{\varphi}_k$ or $\boldsymbol{\psi}_k$ in our settings. By letting $K \rightarrow \infty$, $\mathbb{E}[\mathbf{y}_i^c \mathbf{y}_i^{c\intercal}] \rightarrow \frac{a}{b} \frac{\mathbf{\Lambda}_\varphi^{-1}}{\lambda_s} + \frac{\mathbf{I}_L}{\lambda_y}$ and $\mathbb{E}[\mathbf{h}_i^c \mathbf{h}_i^{c\intercal}] \rightarrow \frac{a}{b} \frac{\mathbf{\Lambda}_\psi^{-1}}{\lambda_t} + \frac{\mathbf{I}_C}{\lambda_h}$; which shows that the joint model remains well defined in the infinite limit. However, when K is truly infinite, the model would naturally use different sets of basis vectors to factorize data from different classes. Whereas atom sharing between different classes is not a necessary requirement for discriminative dictionaries, for practically large values of K, our model results in dictionaries that share atoms among different classes. We note that it is a common practice to use large values of K (instead of true infinity) in practical applications of Beta-Bernoulli process [1], [2], [33].

In our approach, the dictionary atom sharing among the classes is such that besides a group of popular atoms for a given class, few other atoms also have non-zero probability of selection for representing the data of that class. Fig. 2 illustrates this phenomenon with a representative example of an object recognition task. The figure plots the inferred Bernoulli distribution parameters $\pi_{k \in \mathcal{K}}^{c \in \{10,50,90\}}$ for an arranged dictionary learned using the proposed approach. For each class, a cluster of large π_k^c values is observable at a distinct location, emphasizing the discriminative nature of the dictionary. However, few non-zero values can also be observed far from each cluster. These values are indicative of the atom sharing between different classes.

In the aforementioned covariances of $\mathbf{y}_i^c$ and $\mathbf{h}_i^c$, the fraction $\frac{a}{a+b(K-1)}$ appeared due to the presence of $\mathbf{z}_i^c$ in Eq. (4). Ignoring this fraction and comparing the remaining expressions with those when $K \rightarrow \infty$ shows that the value $\frac{a}{b}$ signifies the expected number of factors required to represent $\mathbf{y}_i^c$ and $\mathbf{h}_i^c$ according to the proposed model. This result is similar to the original model of the Beta-Bernoulli process [17], except that the data under consideration is class-

[3] A variational algorithm was developed for the Beta-Bernoulli Process in [17]. Later, Zhou et al. [33] showed Gibbs Sampling to be equally effective for Bayesian dictionary learning. As the latter is intuitively more related to the optimization based algorithms for learning discriminative dictionaries, we developed a Gibbs Sampler for our model in this paper.

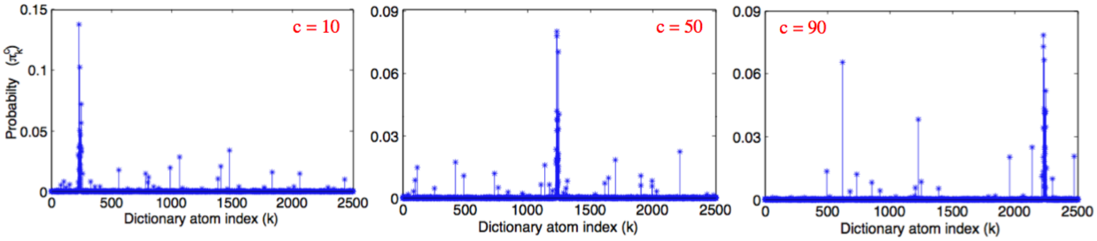

Figure 2. Dictionary atom usage for the data representation of three classes of Caltech-101 [6]: For classes 10, 50 and 90, the Bernoulli distribution parameters π_k^c are plotted against the indices of the arranged dictionary atoms. A larger π_k^c indicates a more frequent use of the atom in data representation. Due to the discriminative nature of the dictionary, distinct clusters of popular atoms appear. Due to sharing of the atoms by different classes, non-zero values also appear far from the main clusters. The shown relationships between the atoms and the class labels are learned adaptively by our approach.

specific in our approach. We use this result to automatically infer the desired dictionary size for our approach during Bayesian inference. Further details in this regard are provided below. We apply our model with $K = 1.25 \times N$ for the initialization and the dictionary is pruned to the desired size during the Bayesian inference.

3.2. Inference

We perform Gibbs Sampling for inference over the proposed model. Due to the conjugacy of the used distributions, we are able to derive all the sampling equations analytically. Below, we briefly present the derivations in the sequence followed by the sampling process to sample the respective parameters. Please refer to the supplementary material of the paper for the detailed derivations. In these derivations, we let $\mathbf{\Lambda}_{\varphi_o} = \mathbf{I}_L/\lambda_{\varphi_o}$ and $\mathbf{\Lambda}_{\psi_o} = \mathbf{I}_C/\lambda_{\psi_o}$. In our experiments, this simplification lead to considerable computational advantage, without significantly affecting the classification performance.

Sample $\boldsymbol{\varphi}_k$: According to the proposed model, we can write the following regarding the posterior probability distribution $p(\boldsymbol{\varphi}_k|-)$ over the k^{th} dictionary atom:

$$p(\boldsymbol{\varphi}_k|-) \propto \prod_{i=1}^{N} \mathcal{N}(\mathbf{y}_{i_{\varphi_k}}|\boldsymbol{\varphi}_k(z_{ik}.s_{ik}), \lambda_{y_o}^{-1}\mathbf{I}_L)\; \mathcal{N}(\boldsymbol{\varphi}_k|\mathbf{0}, \lambda_{\varphi_o}^{-1}\mathbf{I}_L),$$

where, $\mathbf{y}_{i_{\varphi_k}} = \mathbf{y}_i - \mathbf{\Phi}(\mathbf{z}_i \odot \mathbf{s}_i) + \boldsymbol{\varphi}_k(z_{ik} \odot s_{ik})$ denotes the contribution of the k^{th} atom in representing $\mathbf{y}_i$. Note the absence of the superscript 'c' that indicates the atom being treated alike for all the classes and being updated using the complete training data. Exploiting the conjugacy between the Gaussian distributions, $\boldsymbol{\varphi}_k$ can be sampled from $\mathcal{N}(\boldsymbol{\varphi}_k|\boldsymbol{\mu}_k, \lambda_\varphi^{-1}\mathbf{I}_L)$, where:

$$\lambda_\varphi = \lambda_{\varphi_o} + \lambda_{y_o}\sum_{i=1}^{N}(z_{ik}.s_{ik})^2, \;\; \boldsymbol{\mu}_k = \lambda_{y_o}\lambda_\varphi^{-1}\sum_{i=1}^{N}(z_{ik}.s_{ik})\mathbf{y}_{i_{\varphi_k}}.$$

Sample $\boldsymbol{\psi}_k$: Similarly, we can sample $\boldsymbol{\psi}_k$ from $\mathcal{N}(\boldsymbol{\psi}_k|\boldsymbol{\mu}_k, \lambda_\psi^{-1}\mathbf{I}_C)$, such that:

$$\lambda_\psi = \lambda_{\psi_o} + \lambda_{h_o}\sum_{i=1}^{N}(z_{ik}.t_{ik})^2, \;\; \boldsymbol{\mu}_k = \lambda_{h_o}\lambda_\psi^{-1}\sum_{i=1}^{N}(z_{ik}.t_{ik})\mathbf{h}_{i_{\psi_k}}.$$

Sample z_{ik}^c: Once the dictionary and the classifier have been sampled, we sample z_{ik}^c using their updated versions. The posterior probability distribution over z_{ik}^c can be written as, $\forall i \in \mathcal{I}_c, \forall k \in \mathcal{K}$:

$$\begin{aligned} p(z_{ik}^c|-) \propto\; & \mathcal{N}(\mathbf{y}_{i_{\varphi_k}}^c|\boldsymbol{\varphi}_k(z_{ik}^c.s_{ik}^c), \lambda_{y_o}^{-1}\mathbf{I}_L) \\ & \mathcal{N}(\mathbf{h}_{i_{\varphi_k}}^c|\boldsymbol{\psi}_k(z_{ik}^c.t_{ik}^c), \lambda_{h_o}^{-1}\mathbf{I}_C)\ \text{Bernoulli}(z_{ik}^c|\pi_{k_o}^c). \end{aligned}$$

Based on the above expression, it can be shown that z_{ik}^c should be sampled from the following:

$$z_{ik}^c \sim \text{Bernoulli}\left(\frac{\pi_{k_o}^c \xi_1 \xi_2}{1 - \pi_{k_o}^c + \xi_1\xi_2\pi_{k_o}^c}\right), \text{where}$$

$\xi_1 = \exp\left(-\frac{\lambda_{y_o}}{2}(\boldsymbol{\varphi}_k^{\intercal}\boldsymbol{\varphi}_k s_{ik}^{c\,2} - 2s_{ik}^c\mathbf{y}_{i_{\varphi_k}}^{c\intercal}\boldsymbol{\varphi}_k)\right)$ and $\xi_2 = \exp\left(-\frac{\lambda_{h_o}}{2}(\boldsymbol{\psi}_k^{\intercal}\boldsymbol{\psi}_k t_{ik}^{c\,2} - 2t_{ik}\mathbf{h}_{i_{\psi_k}}^{c\intercal}\boldsymbol{\psi}_k)\right)$.

Sample s_{ik}^c: We can write the following regarding the posterior probability distribution over s_{ik}^c:

$$p(s_{ik}^c|-) \propto \mathcal{N}(\mathbf{y}_{i_{\varphi_k}}^c|\boldsymbol{\varphi}_k(z_{ik}^c.s_{ik}^c), \lambda_{y_o}^{-1}\mathbf{I}_L)\mathcal{N}(s_{ik}^c|0, \lambda_{s_o}^{-1}).$$

Exploiting the conjugacy between the distributions, s_{ik}^c can be sampled from $\mathcal{N}(s_{ik}^c|\mu_s, \lambda_s^{-1})$, where:

$$\lambda_s = \lambda_{s_o} + \lambda_{y_o} z_{ik}^{c\,2}\boldsymbol{\varphi}_k^{\intercal}\boldsymbol{\varphi}_k, \quad \mu_s = \lambda_s^{-1}\lambda_{y_o} z_{ik}^c \boldsymbol{\varphi}_k^{\intercal}\mathbf{y}_{i_{\varphi_k}}^c.$$

Sample t_{ik}^c: Correspondingly, we can sample t_{ik}^c from $\mathcal{N}(t_{ik}^c|\mu_t, \lambda_t^{-1})$, where:

$$\lambda_t = \lambda_{t_o} + \lambda_{h_o} z_{ik}^{c\,2}\boldsymbol{\psi}_k^{\intercal}\boldsymbol{\psi}_k, \qquad \mu_t = \lambda_t^{-1}\lambda_{h_o} z_{ik}^c \boldsymbol{\psi}_k^{\intercal}\mathbf{h}_{i_{\psi_k}}^c.$$

Sample π_k^c: According to our model, the posterior probability distribution over π_k^c can be written as follows:

$$p(\pi_k^c|-)\propto\prod_{i\in\mathcal{I}_c}\text{Bernoulli}(z_{ik}^c|\pi_{k_o}^c)\text{Beta}\left(\pi_{k_o}^c|\frac{a_o}{K},\frac{b_o(K-1)}{K}\right),$$

$$\propto \text{Beta}\left(\frac{a_o}{K}+\sum_{i=1}^{|\mathcal{I}_c|} z_{ik}^c,\frac{b_o(K-1)}{K}+|\mathcal{I}_c|-\sum_{i=1}^{|\mathcal{I}_c|} z_{ik}^c\right).$$

Hence, we sample π_k^c from the above mentioned Beta distribution.

Lemma 3.1 *When $\sum\limits_{c=1}^{C}\pi_k^c\rightarrow 0$ in a sampling iteration, the k^{th} factors become unlikely to contribute to the final data representations, given $a_o, b_o<|\mathcal{I}_c|\ll K$.*

Proof: Once $\sum_{c=1}^{C}\pi_k^c \rightarrow 0$ in a given sampling iteration, $\sum_{i=1}^{|\mathcal{I}_c|} z_{ik}^c \rightarrow 0, \forall c$ in the next iteration. This results in approximating the posterior distribution over π_k^c as Beta$\left(\pi_k^c|\frac{a_o}{K},\frac{b_o(K-1)}{K}+|\mathcal{I}_c|\right)$, for which, $\mathbb{E}[\pi_k^c] = \frac{a_o}{a_o+b_o(K-1)+K|\mathcal{I}_c|}$. Under the condition $0<a_o,b_o<|\mathcal{I}_c|\ll K$, $\mathbb{E}[\pi_k^c]\rightarrow 0$. This further results in $\sum\limits_{c=1}^{C}\pi_k^c \rightarrow 0$ in the subsequent iteration. Since π_k represents the probability of selection of the k^{th} factors in data representations, the k^{th} factors become unlikely to contribute to the final representations.

In our approach, the desired dictionary size $|\mathcal{K}|$ is determined by monitoring the sampled values of $\pi_k^c, \forall c$. According to Lemma 3.1, the k^{th} factors $\boldsymbol{\varphi}_k$ and $\boldsymbol{\psi}_k$ can be ignored in the subsequent iterations of the sampling process if $\sum_{c=1}^{C}\pi_k^c\rightarrow 0$ in the current iteration. It happens because when the probability of using a particular factor in representing the data of all classes becomes extremely small in an iteration, that factor also becomes unlikely to contribute in the subsequent iterations of the sampling process. Thus, such factors can be safely ignored in the final representation. Our inference process keeps monitoring such factors and drops them off during the iterative sampling, resulting in an automatic adjustment of the dictionary/classifier size according to the available training data.

Sample λ_s^c: To compute λ_s^c, we treat s_{ik}^c for all the dictionary atoms simultaneously (we do the same for λ_t^c below). We consider $\mathbf{s}_i^c\in\mathbb{R}^{|\mathcal{K}|}$ to be drawn from a multivariate Gaussian with isotropic precision. This allows us to efficiently infer the posterior distribution over λ_s^c. The posterior over λ_s^c can be given as:

$$p(\lambda_s^c|-)\propto\prod_{i\in\mathcal{I}_c}\mathcal{N}(\mathbf{s}_i^c|\mathbf{0},1/\lambda_{s_o}^c\mathbf{I}_{|\mathcal{K}|})\text{Gam}(\lambda_s^c|c_o,d_o).$$

Exploiting the conjugacy between the Gaussian and the Gamma distributions, we sample λ_s^c as:

$$\lambda_s^c\sim\text{Gam}\left(\frac{|\mathcal{I}_c||\mathcal{K}|}{2}+c_o,\frac{1}{2}\sum_{i=1}^{|\mathcal{I}_c|}||\mathbf{s}_i^c||_2^2+d_o\right).$$

Sample λ_t^c: Correspondingly, we also sample λ_t^c from the Gamma probability distribution mentioned above, with $\mathbf{t}_i^c$ replacing $\mathbf{s}_i^c$ in the expression.

Sample λ_y: The posterior probability distribution over λ_y can be written as:

$$p(\lambda_y|-)\propto\prod_{i=1}^{N}\mathcal{N}(\mathbf{y}_i|\boldsymbol{\Phi}(\mathbf{z}_i\odot\mathbf{s}_i),\lambda_{y_o}^{-1}\mathbf{I}_L)\text{Gam}(\lambda_y|e_o,f_o).$$

Again, we do not use the superscript 'c' because λ_y is sampled utilizing the training data from all the classes. Similar to the case of λ_s^c, we can show that λ_y must be sampled as follows:

$$\lambda_y\sim\text{Gam}\left(\frac{LN}{2}+e_o,\frac{1}{2}\sum_{i=1}^{N}||\mathbf{y}_i-\boldsymbol{\Phi}(\mathbf{z}_i\odot\mathbf{s}_i)||_2^2+f_o\right).$$

Sample λ_h: Analogously, λ_h is sampled using the following Gamma distribution:

$$\lambda_h\sim\text{Gam}\left(\frac{CN}{2}+e_o,\frac{1}{2}\sum_{i=1}^{N}||\mathbf{h}_i-\boldsymbol{\Psi}(\mathbf{z}_i\odot\mathbf{t}_i)||_2^2+f_o\right).$$

As a result of the sampling process we infer posterior probability distributions over the dictionary atoms and the classifier parameters. We sample these distributions to obtain the dictionary $\boldsymbol{\Phi}$ and the classifier $\boldsymbol{\Psi}$. To classify a test sample, we first compute its representation $\widehat{\boldsymbol{\alpha}}$ over $\boldsymbol{\Phi}$ and then predict the label by classifying $\widehat{\boldsymbol{\alpha}}$ with the classifier. The class label of the test sample is decided as the index of the largest coefficient of $\boldsymbol{\ell}\in\mathbb{R}^C=\boldsymbol{\Phi}\widehat{\boldsymbol{\alpha}}$. Following the standard practice [2],[10], we use the Orthogonal Matching Pursuit (OMP) algorithm [18] to compute $\widehat{\boldsymbol{\alpha}}$. A Beta-Bernoulli process makes a representation vector sparse by forcing most of its coefficients to become exactly zero, similar to OMP. Therefore, OMP is a natural choice for computing $\widehat{\boldsymbol{\alpha}}$ in our approach.

To start the sampling process, we initialize the dictionary atoms by randomly selecting samples from the training data with replacement. We compute the sparse codes of the training data over the initial $\boldsymbol{\Phi}$ with OMP and use them as the initial values of $\mathbf{s}_i^c$ and $\mathbf{t}_i^c$. The vectors $\mathbf{z}_i^c$ are computed by replacing the non-zero coefficients of the initial $\mathbf{s}_i^c$ with ones. The initial value of $\boldsymbol{\Psi}$ is computed with the help of ridge regression, using $\mathbf{t}_i^c$ and the training labels. This initialization procedure is inspired by the popular discriminative dictionary learning approaches [2], [10], [32].

Table 1. Face recognition results on Extended YaleB database [8]. Results are averaged over ten experiments. The time is given for classifying a single test sample.

Method	Accuracy %	Average Time (ms)
DL-COPAR [26]	86.47 ± 0.69	31.11
LC-KSVD1 [10]	87.76 ± 0.60	0.61
LC-KSVD [10]	89.73 ± 0.59	0.60
D-KSVD [32]	89.77 ± 0.57	0.61
SRC [27]	89.71 ± 0.45	50.19
FDDL [31]	90.01 ± 0.69	42.82
DBDL [2]	91.09 ± 0.59	1.07
JBDC (Proposed)	**92.14 ± 0.52**	1.02

4. Experiments

We evaluated our approach for face, object, scene, and action recognition tasks using standard data sets. The performance is compared to the Label Consistent K-SVD (LC-KSVD) [10], Sparse Representation based Classification (SRC) [27], Discriminative Bayesian Dictionary Learning (DBDL) [2], Discriminative K-SVD (D-KSVD) [32], Fisher Discrimination Dictionary Learning (FDDL) [31] and the Dictionary Learning based on Commonalities and Particularities of the data (DL-COPAR) [19]. These are the state-of-the-art methods in the area of discriminative dictionary learning/sparse representation. We also include the results of an LC-KSVD variant LC-KSVD1, that computes the classifier separately from the dictionary [10].

We used the author-provided implementations of all the methods, except for SRC and D-KSVD. We implemented SRC using the SPAMS toolbox [14], whereas the public code of LC-KSVD [10] was modified for D-KSVD, as recommended by Jiang et al [10]. For all the methods that use OMP to compute the sparse codes, we used the implementation of OMP provided by Elad et al. [21]. In our experiments, all the approaches use the same training and test data. The reported results have been computed after careful optimization of the parameters for each method using cross-validation. We followed the guidelines provided in the original works for selecting the parameter ranges. Discussion on parameter value selection of the proposed approach is provided in Section 5. Experiments were conducted on an Intel processor at 3.4 GHz, with 16 GB RAM.

4.1. Face recognition

We experimented with two commonly used face databases: Extended YaleB [8] and the AR database [16].

4.1.1 Extended YaleB database

This database comprises $2,414$ images of 38 subjects. The images have large variations in terms of illumination conditions and expressions for each subject. To use the images in our experiments, we first created 504-dimensional random face features [27] from the 192×168 cropped face images. For each experiment, we randomly selected 15 features per subject for training and the remaining samples were used for testing. We conducted 10 experiments by randomly selecting the training and testing samples. The means±std.dev of the resulting recognition rates are reported in Table 2. We abbreviate the proposed approach as JBDC for Joint discriminative Bayesian Dictionary and Classifier learning.

The proposed approach resulted in 11.8% reduction in the error rate in Table 2. This reduction is achieved over a recently proposed Bayesian discriminative dictionary learning technique [2]. In our opinion, the better performance of our approach over DBDL is attributed to the stronger coupling between the dictionary and the classifier, and to the ability of JBDC to exploit the relation between the class labels and the factors for the dictionary and the classifier alike. The recognition time of JBDC is comparable to those of the efficient approaches. The low recognition time owes to the joint learning of the classifier along the dictionary. The dictionary/classifier size inferred by JBDC is generally smaller than the dictionary size computed by DBDL which also gives a slight computational advantage to our approach over DBDL. However, the final dictionary size of JBDC is generally larger than the optimal dictionary sizes for D-KSVD and LC-KSVD, which benefits these approaches computationally. Nevertheless, the accuracy of the proposed approach remains significantly better these approaches. In our experiments, the average dictionary size computed by JBDC was 567 atoms, whereas this value was 574 for DBDL. LC-KSVD and D-KSVD used 375 dictionary atoms, which resulted in their best performance.

4.1.2 AR face database

This database consists of over $4,000$ face images of 126 subjects. For each subject, 26 images are taken during two different sessions such that they have large variations in facial disguise, illumination and expressions. We projected 165×120 cropped face images onto 540-dimensional vectors using a random projection matrix, thereby extracting Random-Face features [27]. Following a common evaluation protocol, we selected a subset of $2,600$ images of 50 male and 50 female subjects from the database. For each subject, we used 7 randomly selected images for training and the remaining images were used for testing. Results of our recognition experiments are summarized in Table 2. Similar to the Extended YaleB data set, the proposed approach is also able to generally perform better than the existing approaches on AR database. On average, as compared to 705 dictionary atoms learned by DBDL, JBDC inferred 697 atoms for the training data.

Table 2. Face recognition on AR database [16]. Results are averaged over ten experiments. The time is for a single test sample.

Method	Accuracy %	Average Time (ms)
DL-COPAR [26]	83.29 ± 1.23	36.49
SRC [27]	84.60 ± 1.37	59.91
LC-KSVD1 [10]	84.61 ± 1.44	0.91
LC-KSVD [10]	85.37 ± 1.34	0.91
D-KSVD [32]	85.41 ± 1.49	0.92
FDDL [31]	85.97 ± 1.23	50.03
DBDL [2]	86.15 ± 1.19	1.20
JBDC (Proposed)	$\mathbf{87.17 \pm 0.99}$	1.18

4.2. Object classification

For object classification, we used the Caltech-101 database [6], which contains $9,144$ image samples from 101 object categories and a class of background images. The number of samples per class in this database vary between 31 and 800. For classification, we first created 4096-dimensional feature vectors of the images using the 16-layer deep convolutional neural networks for large scale visual recognition [23]. These features were used to create the training and the testing data sets. Following a common evaluation protocol, we used $5, 10, 15, 20, 25$ and 30 randomly chosen samples per class for training and the remaining samples were used for testing. Results of our experiments are summarized in Table 3. From the table, it is clear that the proposed approach consistently improves the classification accuracy over the existing techniques. The average reduction in the error rate for these experiments is 7.85%. The overall time for classifying 30 samples per class by our approach was 18.77 seconds, whereas DBDL, LC-KSVD and D-KSVD required 18.80, 18.78 and 18.79 seconds, respectively. For JBDC, the final dictionary size was 3001, whereas this value was 3033 for DBDL. Similarly, LC-KSVD and D-KSVD required 3030 atoms for their best performance.

4.3. Scene categorization

The Fifteen Scene Category database [12] consists of images from fifteen natural scene categories. The average image size in the database is 250×300 pixels and the number of sample per class vary between 200 to 400. For this data set, we directly used the 3000-dimensional Spatial Pyramid Features of the images provided by Jiang et al. [10]. From these features, we selected 50 random samples per class for training and used the remaining samples for testing. We summarize the results of our experiments with this data set in Table 4. As evident form the table, the proposed approach is also able to improve results for categorizing the natural scenes.

Table 3. Object classification on Caltech-101 [6].

Training samples	5	10	15	20	25	30
SRC [27]	76.23	79.99	81.27	83.48	84.00	84.51
DL-COPAR [26]	76.11	80.40	83.44	84.01	84.85	85.03
FDDL [31]	78.31	81.37	83.37	84.76	85.66	85.98
LC-KSVD1 [10]	79.03	82.86	84.13	84.65	86.10	86.94
D-KSVD [32]	79.69	83.11	84.99	86.01	86.80	87.72
LC-KSVD [10]	79.74	83.13	85.20	85.98	86.77	87.81
DBDL [2]	80.11	84.03	85.99	86.71	87.97	88.81
JBDC (Proposed)	**81.64**	**85.70**	**86.96**	**87.88**	**88.72**	**89.59**

Table 4. Classification accuracies $(\%)$ on Fifteen Scene Category dataset [12] using Spatial Pyramid Features. The time for computing a single test sample is given in milliseconds.

Method	Accuracy %	Time
FDDL[31]	94.08 ± 0.43	57.99
D-KSVD [32]	95.12 ± 0.18	0.58
LC-KSVD1[10]	95.37 ± 0.28	0.59
SRC [27]	95.41 ± 0.13	78.33
DL-COPAR [26]	96.02 ± 0.28	55.67
LC-KSVD[10]	96.38 ± 0.29	0.59
DBDL[2]	96.98 ± 0.28	0.71
JBDC (Proposed)	$\mathbf{97.73 \pm 0.21}$	0.70

4.4. Action recognition

We used UCF sports action database [20] for action recognition. The database consists of 10 classes of varied sports actions, having a total of 150 clips @ 10 fps. We used the action bank features [22] for this database to train and test the approaches. Following a common evaluation protocol, we performed a five-fold cross validation. The mean recognition rates of the resulting five experiments are reported in Table 5. For FDDL and DL-COPAR we report the results directly from [30], as our parameter optimization for these algorithms could not achieve these accuracies. Results of LDL [30] are also taken from the original work. The proposed joint Bayesian dictionary and classifier learning approach is able to show an average reduction of 12.2% in the error rate for action recognition.

5. Discussion

The choice of the parameter values for our approach is intuitive due to its Bayesian nature. In all the experiments, we set c_o, d_o, e_o and f_0 to 10^{-6}. A wide range of similar small values of these parameters (of the non-informative Gamma hyper-priors) results in a very similar performance of the approach. Considering that the data used in our experiments is mainly clean in terms of white noise, we selected $\lambda_{y_o} = \lambda_{h_o} = 10^6$ for the face, object and scene recognition experiments. The values of these precision parameters were set to 10^9 for the action recognition task due to the less amount of the available training data. Follow-

Table 5. Action recognition on UCF Sports Action database [20].

Method	Accuracy %	Method	Accuracy %
D-KSVD [32]	89.1	SRC [27]	92.6
LC-KSVD1 [10]	89.6	FDDL [31]	93.6
DL-COPAR [26]	90.7	LDL [30]	95.0
LC-KSVD [10]	91.7	DBDL [2]	95.1
JBDC(Proposed)	**95.7**		

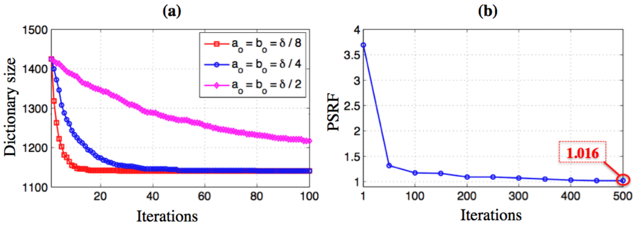

Figure 3. (a) Dictionary size as a function of Gibbs Sampling iterations for Extended YaleB. The first 100 iterations are shown for different values of a_o and b_o. (b) Worst-case Point Scale Reduction Factor (PSRF) [7] for $\pi_c^k, \forall c, \forall k$ as a function of the sampling iterations for Extended YaleB.

ing the common practice in the Beta Process based factor analysis [33], we let $\lambda_{\varphi_o} = 1/L$ and $\lambda_{\psi_o} = 1/C$; and chose $\lambda_{s_o} = \lambda_{t_o} = 1$ for all the experiments. We chose a non-informative initial value for the Bernoulli parameters, i.e. $\pi_{k_o}^c = 0.5, \forall k, \forall c$, implying each dictionary atom has a 50% chance of being used by the representation of a sample of any class at the start of the sampling process.

In the light of Lemma 3.1, we chose $a_o = b_o = \delta/4$ in our experiments, such that $\delta = \min_c |\mathcal{I}_c|$. Here, $a_o = b_o$ indicates that we let the final dictionary size to be roughly around the training data size. This rule was empirically derived and it generally worked well for all the recognition tasks in our experiments. The value $\delta/4$ controls the rate at which the dictionary is pruned to its final size - as illustrated in Fig. 3 (a). In the figure, we plot the dictionary size obtained after each sampling iteration for a face recognition experiment with Extended YaleB database, where 32 samples per class were used for training. The plot is provided for the first 100 iterations for a better visualization. After around 500 iterations, the recognition rates for all the three curves in Fig. 3 (a) were found to be very similar, which also indicates good convergence of the sampler.

To quantitatively analyze the convergence of the sampling process, we followed Gelman and Rubin [7]. For that, the Potential Scale Reduction Factors (PSRFs) for the key parameters of our model, *i.e.* $\pi_k^c, \forall k, \forall c$, were monitored with the increasing number of the sampling iterations for each recognition task. To compute the PSRF values, we ran 10 sampling processes for each database. Each sampling process was initialized by randomly sampling the parameters π_k^c from the standard uniform distribution on the open interval $(0, 1)$. In each experiment, the processes were run for $2n$ iterations and the last n iterations were used to compute the PSRFs. For the details on computing the PSRF values, we refer to [7]. According to Gelman and Rubin, the sampler can be considered converged when PSRF values of the parameters approach to 1. In Fig. 3 (b), we show the *worst-case* values for the Extended YaleB database against the increasing number of the sampler iterations. The worst-case PSFRs are the maximum values among the $C \times |\mathcal{K}|$ values for $\pi_k^c, \forall k, \forall c$. In the figure, these values become very close to 1 after five hundred iterations of the sampler. Since the shown values are for the worst cases, we can conjecture that the performed Gibbs sampler converges reasonably well. The mean PSRFs values for all the five data sets used in our experiments were observed to be very close to 1 after five hundred iterations. Note that, the analysis has been done using those values of the remaining parameters that are mentioned in the preceding paragraphs and using the initialization procedure discussed in Section 3.2. The sampling process took around 8 and 23 minutes to converge for a single experiment of face recognition with Extended YaleB and AR database, respectively. It took around $26, 8$ and 3 minutes respectively for a single object, scene and action recognition experiment. For the object recognition, the reported time is for 5 training samples per class.

6. Conclusion

We proposed a Bayesian approach to jointly infer a discriminative dictionary and a linear classifier under coupled Beta-Bernoulli processes. Our representation model places separate probability distributions over the dictionary and the classifier, but associates them to the training data using the same Bernoulli distributions. The Bernoulli distributions represent the frequency of the dictionary atom usage in data representation and they are learned adaptively under a Bayesian inference. The inference process also accounts for the class labels and the classifier is tailored according to the learned Bernoulli distributions. The joint inference promotes discriminability in the dictionary, which is further encouraged by using separate Bernoulli distributions to represent the training data of each class in our approach. To classify a test sample, we first compute its representation over the dictionary and then predict its label using the representation with the classifier. The classifier accurately predicts the class label due to its strong coupling with the dictionary. We compared our approach with the state-of-the-art discriminative dictionary learning approaches for face, object, scene and action classification tasks. Experiments demonstrate the effectiveness of the proposed approach across the board.

Acknowledgments This research was supported by ARC Discovery grant DP160101458 and it received funding from ARENA as part of ARENA's Research and Development Programme.

References

[1] N. Akhtar, F. Shafait, and A. Mian. Hierarchical beta process with gaussian process prior for hyperspectral image super resolution. In *European Conference on Computer Vision*, pages 103–120. Springer, 2016. 3

[2] N. Akhtar, F. Shafait, and A. Mian. Discriminative bayesian dictionary learning for classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, In Press. 1, 2, 3, 5, 6, 7, 8

[3] J. Bobin, J.-L. Starck, J. M. Fadili, Y. Moudden, and D. L. Donoho. Morphological component analysis: An adaptive thresholding strategy. *IEEE Transactions on Image Processing*, 16(11):2675–2681, 2007. 1

[4] D. L. Donoho. Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006. 1

[5] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image processing*, 15(12):3736–3745, 2006. 1

[6] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding*, 106(1):59–70, 2007. 1, 4, 7

[7] A. Gelman and D. B. Rubin. Inference from iterative simulation using multiple sequences. *Statistical science*, pages 457–472, 1992. 8

[8] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE transactions on pattern analysis and machine intelligence*, 23(6):643–660, 2001. 1, 6

[9] T. Guha and R. K. Ward. Learning sparse representations for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(8):1576–1588, 2012. 1

[10] Z. Jiang, Z. Lin, and L. S. Davis. Label consistent k-svd: Learning a discriminative dictionary for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2651–2664, 2013. 1, 2, 5, 6, 7, 8

[11] I. Kviatkovsky, M. Gabel, E. Rivlin, and I. Shimshoni. On the equivalence of the lc-ksvd and the d-ksvd algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016 (In Press). 1

[12] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 2169–2178, 2006. 1, 7

[13] J. Mairal, F. Bach, and J. Ponce. Task-driven dictionary learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(4):791–804, 2012. 1

[14] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11(Jan):19–60, 2010. 6

[15] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Discriminative learned dictionaries for local image analysis. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. 1

[16] A. M. Martinez and R. Benavente. The AR Face Database. Technical report, CVC, June 1998. 1, 6, 7

[17] J. Paisley and L. Carin. Nonparametric factor analysis with beta process priors. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 777–784. ACM, 2009. 1, 2, 3

[18] Y. C. Pati, R. Rezaiifar, and P. Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on*, pages 40–44. IEEE, 1993. 5

[19] I. Ramirez, P. Sprechmann, and G. Sapiro. Classification and clustering via dictionary learning with structured incoherence and shared features. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3501–3508. IEEE, 2010. 1, 6

[20] M. D. Rodriguez, J. Ahmed, and M. Shah. Action mach a spatio-temporal maximum average correlation height filter for action recognition. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008. 1, 7, 8

[21] R. Rubinstein, M. Zibulevsky, and M. Elad. Efficient implementation of the k-svd algorithm using batch orthogonal matching pursuit. *CS Technion*, 40(8):1–15, 2008. 6

[22] S. Sadanand and J. J. Corso. Action bank: A high-level representation of activity in video. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1234–1241. IEEE, 2012. 7

[23] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 7

[24] P. Sprechmann and G. Sapiro. Dictionary learning and sparse coding for unsupervised clustering. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2042–2045. IEEE, 2010. 1

[25] I. Tosic and P. Frossard. Dictionary learning. *IEEE Signal Processing Magazine*, 28(2):27–38, 2011. 1

[26] D. Wang and S. Kong. A classification-oriented dictionary learning model: Explicitly learning the particularity and commonality across categories. *Pattern Recognition*, 47(2):885–898, 2014. 1, 6, 7, 8

[27] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE transactions on pattern analysis and machine intelligence*, 31(2):210–227, 2009. 2, 6, 7, 8

[28] Y. N. Wu, Z. Si, H. Gong, and S.-C. Zhu. Learning active basis model for object detection and recognition. *International journal of computer vision*, 90(2):198–235, 2010. 1

[29] J. Yang, K. Yu, and T. Huang. Supervised translation-invariant sparse coding. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3517–3524. IEEE, 2010. 1

[30] M. Yang, D. Dai, L. Shen, and L. Van Gool. Latent dictionary learning for sparse representation based classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4138–4145, 2014. 1, 7, 8

[31] M. Yang, L. Zhang, X. Feng, and D. Zhang. Sparse representation based fisher discrimination dictionary learning for image classification. *International Journal of Computer Vision*, 109(3):209–232, 2014. 1, 6, 7, 8

[32] Q. Zhang and B. Li. Discriminative k-svd for dictionary learning in face recognition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2691–2698. IEEE, 2010. 1, 2, 5, 6, 7, 8

[33] M. Zhou, H. Chen, L. Ren, G. Sapiro, L. Carin, and J. W. Paisley. Non-parametric bayesian dictionary learning for sparse image representations. In *Advances in neural information processing systems*, pages 2295–2303, 2009. 3, 8