Discriminative Bayesian Dictionary Learning for Classification

Naveed Akhtar, Faisal Shafait, and Ajmal Mian

Abstract—We propose a Bayesian approach to learn discriminative dictionaries for sparse representation of data. The proposed approach infers probability distributions over the atoms of a discriminative dictionary using a finite approximation of Beta Process. It also computes sets of Bernoulli distributions that associate class labels to the learned dictionary atoms. This association signifies the selection probabilities of the dictionary atoms in the expansion of class-specific data. Furthermore, the non-parametric character of the proposed approach allows it to infer the correct size of the dictionary. We exploit the aforementioned Bernoulli distributions in separately learning a linear classifier. The classifier uses the same hierarchical Bayesian model as the dictionary, which we present along the analytical inference solution for Gibbs sampling. For classification, a test instance is first sparsely encoded over the learned dictionary and the codes are fed to the classifier. We performed experiments for face and action recognition; and object and scene-category classification using five public datasets and compared the results with state-of-the-art discriminative sparse representation approaches. Experiments show that the proposed Bayesian approach consistently outperforms the existing approaches.

Index Terms—Bayesian sparse representation, discriminative dictionary learning, supervised learning, classification

1 Introduction

Sparse representation encodes a signal as a sparse linear combination of redundant basis vectors. With its inspirational roots in human vision system [16], [17], this technique has been successfully employed in image restoration [18], [19], [20], compressive sensing [21], [22] and morphological component analysis [23]. More recently, sparse representation based approaches have also shown promising results in face recognition and gender classification [9], [8], [10], [13], [24], [25], [26], texture and handwritten digit classification [14], [29], [30], [31], natural image and object classification [9], [11], [32] and human action recognition [33], [34], [35], [36]. The success of these approaches comes from the fact that a sample from a class can generally be well represented as a sparse linear combination of the other samples from the same class, in a lower dimensional manifold [8].

For classification, a discriminative sparse representation approach first encodes the test instance over a dictionary, i.e., a redundant set of basis vectors, known as atoms. Therefore, an effective dictionary is critical for the performance of such approaches. It is possible to use an off-the-shelf basis (e.g., fast Fourier transform [41] or wavelets [42]) as a generic dictionary to represent data from different domains/classes. However, research in the last decade ([6], [9], [10], [11], [18], [43], [44], [45]) has provided strong evidence in favor of learning dictionaries using the domain/class-specific training data, especially for classification and recognition tasks [10], where class label information of the training data can be exploited in the supervised learning of a dictionary.

Whereas unsupervised dictionary learning approaches (e.g., K-SVD [6], Method of Optimal Directions [46]) aim at learning faithful signal representations, supervised sparse representation additionally strives for making the dictionaries discriminative. For instance, in Sparse Representation based Classification (SRC) scheme, Wright et al. [8] constructed a discriminative dictionary by directly using the training data as the dictionary atoms. With each atom associated to a particular class, the query is assigned the label of the class whose associated atoms maximally contribute to the sparse representation of the query. Impressive results have been achieved for recognition and classification using SRC, however, the computational complexity of this technique becomes prohibitive for large training data. This has motivated considerable research on learning discriminative dictionaries that would allow sparse representation based classification with much lower computational cost.

In order to learn a discriminative dictionary, existing approaches either force subsets of the dictionary atoms to represent data from only specific classes [12], [26], [47] or they associate the complete dictionary to all the classes and constrain their sparse coefficients to be discriminative [7], [9], [28]. A third category of techniques learns exclusive sets of class specific and common dictionary atoms to separate the common and particular features of the data from different classes [11], [54]. Establishing association between the atoms and the corresponding class labels is a key step of the existing methods. However, adaptively building this association is still an open research problem [13]. Moreover, the strategy of assigning different number of dictionary atoms

- *N. Akhtar and A. Mian are with the School of Computer Science and Software Engineering, The University of Western Australia, 35 Stirling Highway Crawley, 6009, WA, Australia. E-mail: navid.915@gmail.com, ajmal.mian@uwa.edu.au.*
- *F. Shafait is with the School of Electrical Engineering and Computer Science at the National University of Sciences and Technology, Islamabad, Pakistan. E-mail: faisal.shafait@uwa.edu.au.*

Manuscript received 16 Mar. 2015; revised 25 Jan. 2016; accepted 1 Feb. 2016. Date of publication 10 Feb. 2016; date of current version 10 Nov. 2016. Recommended for acceptance by S. Nowozin. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TPAMI.2016.2527652

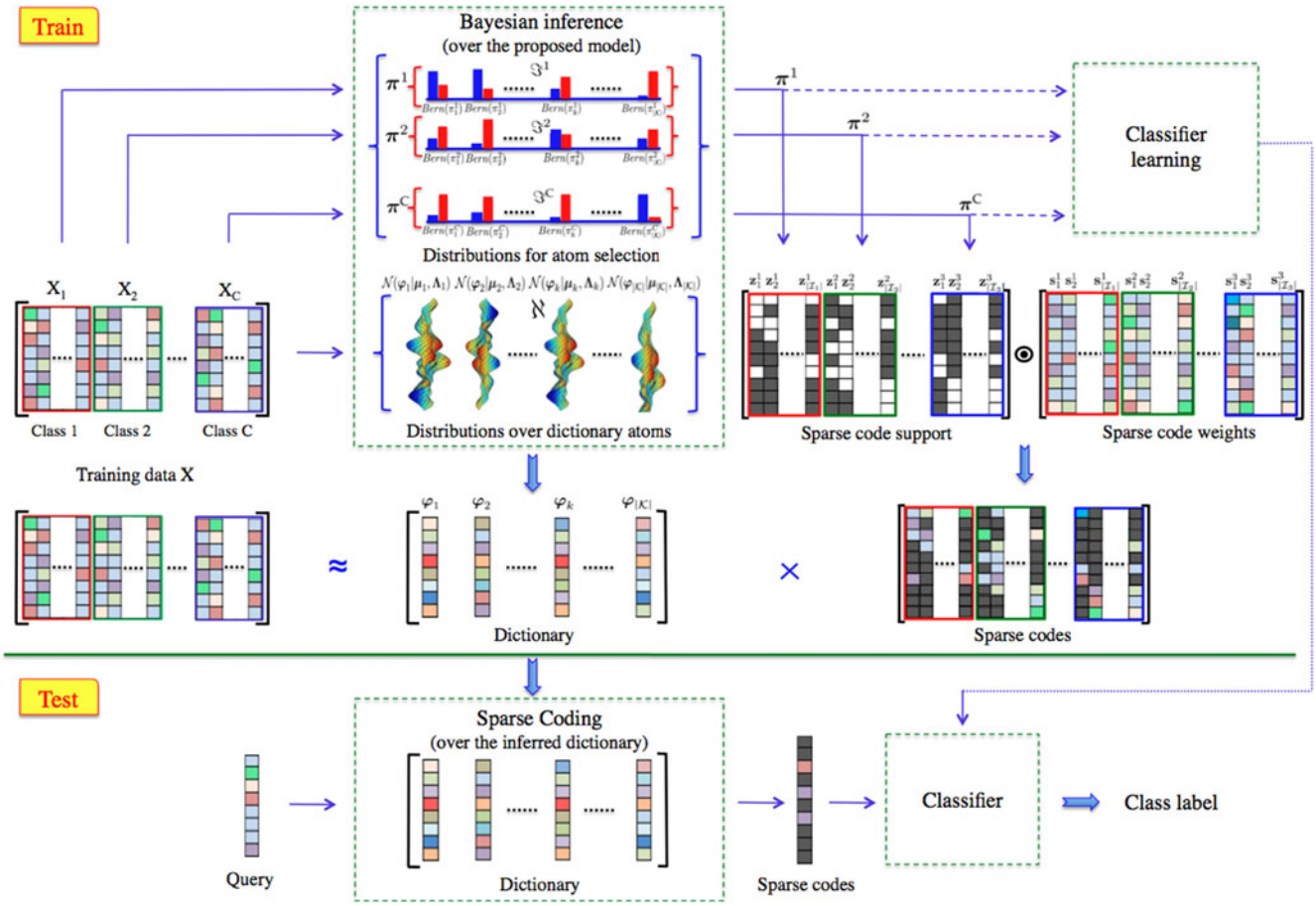


Fig. 1. Schematics of the proposed approach: For training, a set of probability distributions over the dictionary atoms, i.e., $\aleph$, is learned. We also infer sets of Bernoulli distributions indicating the probabilities of selection of the dictionary atoms in the expansion of data from each class. These distributions are used for inferring the support of the sparse codes. The (parameters of) Bernoulli distributions are later used for learning a classifier. The final dictionary is learned by sampling the distributions in $\aleph$, whereas the sparse codes are computed as element-wise product of the support and the weights (also inferred by the approach) of the codes. Combined, the dictionary and the codes faithfully represent the training data. For testing, sparse codes of the query over the dictionary are computed and fed to the classifier for labeling.

to different classes and adjusting the overall size of the dictionary become critical for the classification accuracy of the existing approaches, as no principled approach is generally provided to predetermine these parameters.

In this work, we propose a solution to this problem by approaching the sparse representation based classification from a non-parametric Bayesian perspective. We propose a Bayesian sparse representation technique that infers a discriminative dictionary using a finite approximation of the Beta Process [56]. Our approach adaptively builds the association between the dictionary atoms and the class labels such that this association signifies the probability of selection of the dictionary atoms in the expansion of class-specific data. Furthermore, the non-parametric character of the approach allows it to automatically infer the correct size of the dictionary. The scheme employed by our approach is shown in Fig. 1. We perform Bayesian inference over a model proposed for the discriminative sparse representation of the training data. The inference process learns distributions over the dictionary atoms and sets of Bernoulli distributions associating the dictionary atoms to the labels of the data. The Bernoulli distributions govern the support of the final sparse codes and are later utilized in learning a multi-class linear classifier. The final dictionary is learned by sampling the distributions over the dictionary atoms and the corresponding sparse codes are computed by an element-wise product of the support and the inferred weights of the codes. The learned dictionary and the sparse codes also represent the training data faithfully.

A query is classified in our approach by first sparsely encoding it over the inferred dictionary and then classifying its sparse code with the learned classifier. In this work, we learn the classifier and the dictionary using the same hierarchical Bayesian model. This allows us to exploit the aforementioned Bernoulli distributions in the accurate estimate of the classifier. We present the proposed Bayesian model along its inference equations for Gibbs sampling. Our approach has been tested on two face-databases [1], [2], an object-database [3], an action-database [5] and a scene-database [4]. The classification results are compared with the state-of-the-art discriminative sparse representation approaches. The proposed approach not only outperforms these approaches in terms of accuracy, its computational efficiency for the classification stage is also comparable to the most efficient existing approaches.

This paper is organized as follows. We review the related work in Section 2. In Section 3, we formulate the problem and briefly explain the relevant concepts that clarify the rationale behind our approach. The proposed approach is presented in Section 4. Experimental results are reported in

Section 5 and a discussion on the parameter settings is provided in Section 6. We draw conclusions in Section 7.

2 Related Work

There are three main categories of the approaches that learn discriminative sparse representation. In the first category, the learned dictionary atoms have a direct correspondence to the labels of the classes [12], [26], [35], [36], [47], [48], [49]. Yang et al. [26] proposed an SRC like framework for face recognition, where the atoms of the dictionary are learned from the training data instead of directly using the training data as the dictionary. In order to learn a dictionary that is simultaneously discriminative and reconstructive, Mairal et al. [47] used a discriminative penalty term in the K-SVD model [6], achieving state-of-the-art results on texture segmentation. Sprechmann and Sapiro [48] also proposed to learn dictionaries and sparse codes for clustering. In [36], Castrodad and Sapiro computed class-specific dictionaries for actions. The dictionary atoms and their sparse coefficients also exploited the non-negativity of the signals in their approach. Active basis models are learned from the training images of each class and applied to object detection and recognition in [49]. Ramirez et al. [12] have used an incoherence promoting term for the dictionary atoms in their learning model. Encouraging incoherence among the class-specific sub-dictionaries allowed them to represent samples from the same class better than the samples from the other classes. Wang et al. [35] have proposed to learn class-specific dictionaries for modeling individual actions for action recognition. Their model incorporated a similarity constrained term and a dictionary incoherence term for classification. The above mentioned methods mainly associate a dictionary atom directly to a single class. Therefore, a query is generally assigned the label of the class whose associated atoms result in the minimum representational error for the query. The classification stages of the approaches under this category often require the computation of representations of the query over many sub-dictionaries.

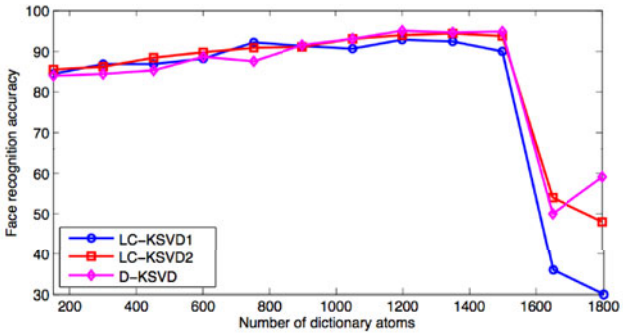
In the second category, a single dictionary is shared by all the classes, however the representation coefficients are forced to be discriminative ([7], [9], [28], [29], [30], [31], [33], [45], [50], [51]). Jiang et al. [9] proposed a dictionary learning model that encourages the sparse representation coefficients of the same class to be similar. This is done by adding a 'discriminative sparse-code error' constraint to a unified objective function that already contains reconstruction error and classification error constraints. A similar approach is taken by Rodriguez and Sapiro [30] where the authors solve for a simultaneous sparse approximation problem [52] while learning the coefficients. It is common to learn dictionaries jointly with a classifier. Pham and Venkatesh [45] and Mairal et al. [28] proposed to train linear classifiers along the joint dictionaries learned for all the classes. Zhang and Li [7] enhanced the K-SVD algorithm [6] to learn a linear classifier along the dictionary. A task driven dictionary learning framework has also been proposed [31]. Under this framework, different risk functions of the representation coefficients are minimized for different tasks. Broadly speaking, the above mentioned approaches aim at learning a single dictionary together with a classifier. The query is classified by directly feeding its sparse codes over the learned single dictionary to the classifier. Thus, in comparison to the approaches in the first category, the classification stage of these approaches is computationally more efficient. In terms of learning a single dictionary for the complete training data and the classification stage, the proposed approach also falls under this category of discriminative sparse representation techniques.

The third category takes a hybrid approach for learning the discriminative sparse representation. In these approaches, the dictionaries are designed to have a set of shared atoms in addition to class-specific atoms. Deng et al. [53] extended the SRC algorithm by appending an intra-class face variation dictionary to the training data. This extension achieves promising results in face recognition with a single training sample per class. Zhou and Fan [54] employ a Fisher-like regularizer on the representation coefficients while learning a hybrid dictionary. Wang and Kong [11] learned a hybrid dictionary to separate the common and particular features of the data. Their approach also encouraged the class-specific dictionaries to be incoherent. Shen et al. [55] proposed to learn a multi-level dictionary for hierarchical visual categorization. To some extent, it is possible to reduce the size of the dictionary using the hybrid approach, which also results in reducing the classification time in comparison to the approaches that fall under the first category. However, it is often non-trivial to decide on how to balance between the shared and the class-specific parts of the hybrid dictionary [10], [13].

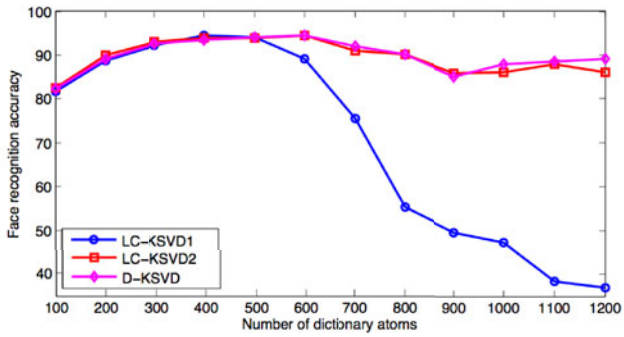
The above mentioned approaches make the dictionaries discriminative by controlling the extent of their atom-sharing among class-specific representations. In this regard, latent variable models [67], [68], [69], [70] are also related to the discriminative dictionary learning framework. Damianou et al. [67] presented a Bayesian model that factorizes the latent variable space to represent shared and private information from multiple data views. They kept the segmentation of the latent space soft, such that a latent variable is even allowed to be more important to the shared space than the private space. Andrade-Pacheco et al. [68] later extended their approach to non-Gaussian data. Lu and Tang [69] also extended the Relevance Manifold Determination (RMD) [67] to learn face prior for Bayesian face recognition. Their approach first learned identity subspaces for each subject using RMD and later used the structure of the subspaces to estimate the Gaussian mixture densities in the observation space. Klami et al. [70] proposed a model for group factor analysis and formulated its solution as a variational inference of a latent variable model with structural sparsity.

3 Problem Formulation and Background

Let $\mathbf{X} = [\mathbf{X}^1, \ldots, \mathbf{X}^c, \ldots, \mathbf{X}^C] \in \mathbb{R}^{m \times N}$ be the training data comprising N instances from C classes, wherein $\mathbf{X}^c \in \mathbb{R}^{m \times N_c}$ represents the data from the cth class and $\sum_{c=1}^{C} N_c = N$. The columns of $\mathbf{X}^c$ are indexed in $\mathcal{I}_c$. We denote a dictionary by $\mathbf{\Phi} \in \mathbb{R}^{m \times |\mathcal{K}|}$ with atoms $\boldsymbol{\varphi}_k$, where $k \in \mathcal{K} = \{1, \ldots, K\}$ and $|.|$ represents the cardinality of the set. Let $\mathbf{A} \in \mathbb{R}^{|\mathcal{K}| \times N}$ be the sparse code matrix of the data, such that $\mathbf{X} \approx \mathbf{\Phi}\mathbf{A}$. We can write $\mathbf{A} = [\mathbf{A}^1, \ldots, \mathbf{A}^c, \ldots, \mathbf{A}^C]$, where $\mathbf{A}^c \in \mathbb{R}^{|\mathcal{K}| \times |\mathcal{I}_c|}$ is the sub-matrix related to the cth class. The ith column of $\mathbf{A}$ is denoted as $\boldsymbol{\alpha}_i \in \mathbb{R}^{|\mathcal{K}|}$. To learn a sparse representation of the data, we can solve the following optimization problem:



(a) AR database [1]



(b) Extended YaleB database [2]

Fig. 2. Examples of how recognition accuracy is affected with varying dictionary size: $\kappa = 0$ for LC-KSVD1 and $\upsilon = 0$ for D-KSVD in Eq. (3). All other parameters are kept constant at optimal values reported in [9]. For the AR database, 2,000 training instances are used and testing is performed with 600 instances. For the Extended YaleB, half of the database is used for training and the other half is used for testing. The instances are selected uniformly at random.

$$<\mathbf{\Phi}, \mathbf{A}> = \min_{\mathbf{\Phi},\mathbf{A}} ||\mathbf{X} - \mathbf{\Phi A}||_F^2 \quad s.t. \quad \forall i, ||\boldsymbol{\alpha}_i||_p \leq t, \tag{1}$$

where t is a predefined constant, $||.||_F$ computes the Frobenius norm and $||.||_p$ denotes the ℓ_p-norm of a vector. Generally, p is chosen to be 0 or 1 for sparsity [57]. The non-convex optimization problem of Eq. (1) can be iteratively solved by fixing one parameter and solving a convex optimization problem for the other parameter in each iteration. The solution to Eq. (1), factors the training data $\mathbf{X}$ into two complementary matrices, namely the dictionary and the sparse codes, without considering the class label information of the training data. Nevertheless, we can still exploit this factorization in classification tasks by using the sparse codes of the data as features [9], for which, a classifier can be obtained as

$$\mathbf{W} = \min_{\mathbf{W}} \sum_{i=1}^{N} \mathcal{L}\{h_i, f(\boldsymbol{\alpha}_i, \mathbf{W})\} + \lambda ||\mathbf{W}||_F^2, \tag{2}$$

where $\mathbf{W} \in \mathbb{R}^{C \times |\mathcal{K}|}$ contains the model parameters of the classifier, $\mathcal{L}$ is the loss function, h_i is the label of the ith training instance $\mathbf{x}_i \in \mathbb{R}^m$ and λ is the regularizer.

It is usually suboptimal to perform classification based on sparse codes learned by an unsupervised technique. Considering this, existing approaches [7], [28], [29], [45], proposed to jointly optimize a classifier with the dictionary while learning the sparse representation. One intended ramification of this approach is that the label information also gets induced into the dictionary. This happens when the information is utilized in computing the sparse codes of the data, which in turn are used for computing the dictionary atoms. This results in improving the discriminative abilities of the learned dictionary. Jiang et al. [9] built further on this concept and encouraged explicit correspondence between the dictionary atoms and the class-labels. More precisely, the following optimization problem is solved by the Label-Consistent K-SVD (LC-KSVD2) algorithm [9]:

$$<\mathbf{\Phi}, \mathbf{W}, \mathbf{T}, \mathbf{A}> = \min_{\mathbf{\Phi},\mathbf{W},\mathbf{T},\mathbf{A}} \left\| \begin{pmatrix} \mathbf{X} \\ \sqrt{\upsilon}\mathbf{Q} \\ \sqrt{\kappa}\mathbf{H} \end{pmatrix} - \begin{pmatrix} \mathbf{\Phi} \\ \sqrt{\upsilon}\mathbf{T} \\ \sqrt{\kappa}\mathbf{W} \end{pmatrix} \mathbf{A} \right\|_F^2 \tag{3}$$
$$s.t.\ \forall i\ ||\boldsymbol{\alpha}_i||_0 \leq t,$$

where υ and κ are the regularization parameters, the binary matrix $\mathbf{H} \in \mathbb{R}^{C \times N}$ contains the class label information,[1] $\mathbf{T} \in \mathbb{R}^{|\mathcal{K}| \times |\mathcal{K}|}$ is the transformation between the sparse codes and the *discriminative sparse codes* $\mathbf{Q} \in \mathbb{R}^{|\mathcal{K}| \times N}$. Here, for the ith training instance, the ith column of the fixed binary matrix $\mathbf{Q}$ has 1 appearing at the kth index only if the kth dictionary atom has the same class label as the training instance. Thus, the discriminative sparse codes form a pre-defined relationship between the dictionary atoms and the class labels. This brings improvement to the discriminative abilities of the dictionary learned by solving Eq. (3).

It is worth noting that in Label-Consistent K-SVD algorithm [9], the relationship between class-specific subsets of dictionary atoms and class labels is pre-defined. However, regularization allows flexibility in this association during optimization. We also note that using $\upsilon = 0$ in Eq. (3) reduces the optimization problem to the one solved by Discriminative K-SVD (D-KSVD) algorithm [7]. Successful results are achievable using the above mentioned techniques for recognition and classification. However, like any discriminative sparse representation approach, these results are obtainable only after careful optimization of the algorithm parameters, including the dictionary size. In Fig. 2, we illustrate the behavior of recognition accuracy under varying dictionary sizes for [7] and [9] for two face databases.

Paisley and Carin [56] developed a *Beta Process* for non-parametric factor analysis, which was later used by Zhou et al. [44] in successful image restoration. Exploiting the non-parametric Bayesian framework, a Beta Process can automatically infer the factor/dictionary size from the training data. With the base measure $\hbar_0$ and parameters $a_o > 0$ and $b_o > 0$, a Beta Process is denoted as $\mathrm{BP}(a_o, b_o, \hbar_0)$. Paisley and Carin [56] noted that a finite representation of the process can be given as:

$$\begin{aligned} \hbar &= \sum_k \pi_k \delta_{\boldsymbol{\varphi}_k}(\boldsymbol{\varphi}), \quad k \in \mathcal{K} = \{1, \ldots, K\}, \\ \pi_k &\sim \mathrm{Beta}(\pi_k | a_o/K, b_o(K-1)/K), \\ \boldsymbol{\varphi}_k &\sim \hbar_0. \end{aligned} \tag{4}$$

1. For the ith training instance, the ith column of $\mathbf{H}$ has 1 appearing only at the index corresponding to the class label.

In Eq. (4), $\delta_{\boldsymbol{\varphi}_k}(\boldsymbol{\varphi})$ is 1 when $\boldsymbol{\varphi} = \boldsymbol{\varphi}_k$ and 0 otherwise. Therefore, a draw $\hbar$ from the process can be represented as a set of $|\mathcal{K}|$ probabilities, each having an associated vector $\boldsymbol{\varphi}_k$, drawn $i.i.d.$ from the base measure $\hbar_0$. Using $\hbar$, we can draw a binary vector $\mathbf{z}_i \in \{0,1\}^{|\mathcal{K}|}$, such that the k^{th} component of $\mathbf{z}_i$ is drawn $z_{ik} \sim \text{Bernoulli}(\pi_k)$. By independently drawing N such vectors, we may construct a matrix $\mathbf{Z} \in \{0,1\}^{|\mathcal{K}| \times N}$, where $\mathbf{z}_i$ is the ith column of this matrix.

Using the above mentioned finite approximation of the Beta Process, it is possible to factorize $\mathbf{X}$ as follows:

$$\mathbf{X} = \boldsymbol{\Phi}\mathbf{Z} + \mathbf{E}, \tag{5}$$

where, $\boldsymbol{\Phi} \in \mathbb{R}^{m \times |\mathcal{K}|}$ has $\boldsymbol{\varphi}_k$ as its columns and $\mathbf{E} \in \mathbb{R}^{m \times N}$ is the error matrix. In Eq. (5), the number of non-zero components in a column of $\mathbf{Z}$ can be controlled by the parameters a_o and b_o in Eq. (4). The components of the kth row of $\mathbf{Z}$ are independent draws from Bernoulli(π_k). Let $\boldsymbol{\pi} \in \mathbb{R}^{|\mathcal{K}|}$ be a vector with $\pi_{k \in \mathcal{K}}$, as its kth element. This vector governs the probability of selection of the columns of $\boldsymbol{\Phi}$ in the expansion of the data. Existence of this physically meaningful latent vector in the Beta Process based matrix factorization plays a central role in the proposed approach.

4 Proposed Approach

We propose a Discriminative Bayesian Dictionary Learning approach for classification. For the cth class, our approach draws $|\mathcal{I}_c|$ binary vectors $\mathbf{z}_i^c \in \mathbb{R}^{|\mathcal{K}|}$, $\forall i \in \mathcal{I}_c$ using a finite approximation of the Beta Process. For each class, the vectors are sampled using separate draws with the same base. That is, the matrix factorization is governed by a set of C probability vectors $\boldsymbol{\pi}^{c \in \{1,\ldots,C\}}$, instead of a single vector, however the inferred dictionary is shared by all the classes. An element of the aforementioned set, i.e., $\boldsymbol{\pi}^c \in \mathbb{R}^{|\mathcal{K}|}$, controls the probability of selection of the dictionary atoms for a single class data. This promotes discrimination in the inferred dictionary.

4.1 The Model

Let $\boldsymbol{\alpha}_i^c \in \mathbb{R}^{|\mathcal{K}|}$ denote the sparse code of the ith training instance of the cth class, i.e., $\mathbf{x}_i^c \in \mathbb{R}^m$, over a dictionary $\boldsymbol{\Phi} \in \mathbb{R}^{m \times |\mathcal{K}|}$. Mathematically, $\mathbf{x}_i^c = \boldsymbol{\Phi}\boldsymbol{\alpha}_i^c + \boldsymbol{\epsilon}_i$, where $\boldsymbol{\epsilon}_i \in \mathbb{R}^m$ denotes the modeling error. We can directly use the Beta Process discussed in Section 3 for computing the desired sparse code and the dictionary. However, the model employed by the Beta Process is restrictive, as it only allows the code to be binary. To overcome this restriction, let $\boldsymbol{\alpha}_i^c = \mathbf{z}_i^c \odot \mathbf{s}_i^c$, where $\odot$ denotes the Hadamard/element-wise product, $\mathbf{z}_i^c \in \mathbb{R}^{|\mathcal{K}|}$ is the binary vector and $\mathbf{s}_i^c \in \mathbb{R}^{|\mathcal{K}|}$ is a weight vector. We place a standard normal prior $\mathcal{N}(s_{ik}^c|0, 1/\lambda_{s_o}^c)$ on the kth component of the weight vector s_{ik}^c, where $\lambda_{s_o}^c$ denotes the precision of the distribution. In here, as in the following text, we use the subscript 'o' to distinguish the parameters of the prior distributions. The prior distribution over the kth component of the binary vector is Bernoulli($z_{ik}^c|\pi_{k_o}^c$). We draw the atoms of the dictionary from a multivariate Gaussian base, i.e.. $\boldsymbol{\varphi}_k \sim \mathcal{N}(\boldsymbol{\varphi}_k|\boldsymbol{\mu}_{k_o}, \boldsymbol{\Lambda}_{k_o}^{-1})$, where $\boldsymbol{\mu}_{k_o} \in \mathbb{R}^m$ is the mean vector and $\boldsymbol{\Lambda}_{k_o} \in \mathbb{R}^{m \times m}$ is the precision matrix for the kth atom of the dictionary. We model the error as zero mean Gaussian in $\mathbb{R}^m$. Thus, we arrive at the following representation model:

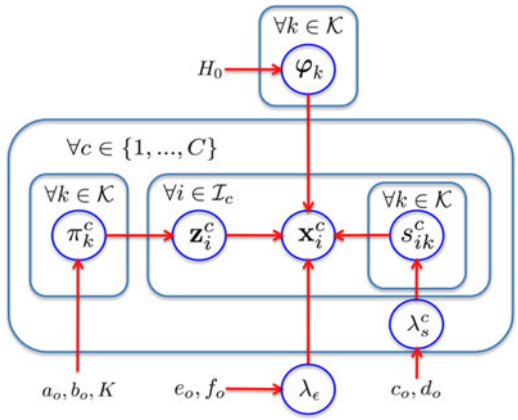


Fig. 3. Graphical representation of the proposed discriminative Bayesian dictionary learning model.

$$\begin{aligned}
&\mathbf{x}_i^c = \boldsymbol{\Phi}\boldsymbol{\alpha}_i^c + \boldsymbol{\epsilon}_i && \forall i \in \mathcal{I}_c, \forall c \\
&\boldsymbol{\alpha}_i^c = \mathbf{z}_i^c \odot \mathbf{s}_i^c \\
&z_{ik}^c \sim \text{Bernoulli}(z_{ik}^c|\pi_{k_o}^c) \\
&s_{ik}^c \sim \mathcal{N}(s_{ik}^c|0, 1/\lambda_{s_o}^c) \\
&\pi_k^c \sim \text{Beta}(\pi_k^c|a_o/K, b_o(K-1)/K) \\
&\boldsymbol{\varphi}_k \sim \mathcal{N}(\boldsymbol{\varphi}_k|\boldsymbol{\mu}_{k_o}, \boldsymbol{\Lambda}_{k_o}^{-1}) \quad \forall k \in \mathcal{K} \\
&\boldsymbol{\epsilon}_i \sim \mathcal{N}(\boldsymbol{\epsilon}_i|\mathbf{0}, \boldsymbol{\Lambda}_{\epsilon_o}^{-1}) \forall i \in \{1, \ldots, N\}.
\end{aligned} \tag{6}$$

Notice, in the above model a conjugate Beta prior is placed over the parameter of the Bernoulli distribution, as mentioned in Section 3. Hence, a latent probability vector $\boldsymbol{\pi}^c$ (with π_k^c as its components) is associated with the dictionary atoms for the representation of the data from the cth class. The common dictionary $\boldsymbol{\Phi}$ is inferred from C such vectors. In the above model, this fact is notationally expressed by showing the dictionary atoms being sampled from a common set of $|\mathcal{K}|$ distributions, while distinguishing the class-specific variables in the other notations with a superscript 'c'. We assume the same statistics for the modeling error over the complete training data.[2] We further place non-informative Gamma hyper-priors over the precision parameters of the normal distributions. That is, $\lambda_s^c \sim \Gamma(\lambda_s^c|c_o, d_o)$ and $\lambda_\epsilon \sim \Gamma(\lambda_\epsilon|e_o, f_o)$, where c_o, d_o, e_o and f_o are the parameters of the respective Gamma distributions. Here, we allow the error to have an isotropic precision, i.e., $\boldsymbol{\Lambda}_\epsilon = \lambda_\epsilon \mathbf{I}_m$, where $\mathbf{I}_m$ denotes the identity matrix in $\mathbb{R}^{m \times m}$. The graphical representation of the complete model is shown in Fig. 3.

4.2 Inference

Gibbs sampling is used to perform Bayesian inference over the proposed model.[3] Starting with the dictionary, below we derive analytical expressions for the posterior distributions

2. It is also possible to use different statistics for different classes, however, in practice the assumption of similar noise statistics works well. We adopt the latter to avoid unnecessary complexity.

3. Paisley and Carin [56] derived variational Bayesian algorithm [58] for their model. It was shown by Zhou et al. [44] that Gibbs sampling is an equally effective strategy in data representation using the same model. Since it is easier to relate the Gibbs sampling process to the learning process of conventional optimization based sparse representation (e.g., K-SVD [6]), we derive expressions for the Gibbs sampler for our approach. Due to the conjugacy of the model, these expressions can be derived analytically.

over the model parameters for the Gibbs sampler. The inference process performs sampling over these posterior distributions. The expressions are derived assuming zero mean Gaussian prior over the dictionary atoms, with isotropic precision. That is, $\boldsymbol{\mu}_{k_o} = \mathbf{0}$ and $\boldsymbol{\Lambda}_{k_o} = \lambda_{k_o}\mathbf{I}_m$. This simplification leads to faster sampling, without significantly affecting the accuracy of the approach. The sampling process samples the atoms of the dictionary one-by-one from their respective posterior distributions. This process is analogous to the atom-by-atom dictionary update step of K-SVD [6], however the sparse codes remain fixed during our dictionary update.

Sampling $\boldsymbol{\varphi}_k$. For our model, we can write the following about the posterior distribution over a dictionary atom:

$$p(\boldsymbol{\varphi}_k|-) \propto \prod_{i=1}^{N} \mathcal{N}(\mathbf{x}_i|\boldsymbol{\Phi}(\mathbf{z}_i \odot \mathbf{s}_i), \lambda_{\epsilon_o}^{-1}\mathbf{I}_m)\mathcal{N}\big(\boldsymbol{\varphi}_k|\mathbf{0}, \lambda_{k_o}^{-1}\mathbf{I}_m\big).$$

Here, we intentionally dropped the superscript 'c' as the dictionary is updated using the complete training data. Let $\mathbf{x}_{i_{\varphi_k}}$ denote the contribution of the dictionary atom $\boldsymbol{\varphi}_k$ to the ith training instance $\mathbf{x}_i$:

$$\mathbf{x}_{i_{\varphi_k}} = \mathbf{x}_i - \boldsymbol{\Phi}(\mathbf{z}_i \odot \mathbf{s}_i) + \boldsymbol{\varphi}_k(z_{ik} \odot s_{ik}). \tag{7}$$

Using Eq. (7), we can re-write the aforementioned proportionality as

$$p(\boldsymbol{\varphi}_k|-) \propto \prod_{i=1}^{N} \mathcal{N}\big(\mathbf{x}_{i_{\varphi_k}}|\boldsymbol{\varphi}_k(z_{ik}s_{ik}), \lambda_{\epsilon_o}^{-1}\mathbf{I}_m\big)\mathcal{N}\big(\boldsymbol{\varphi}_k|\mathbf{0}, \lambda_{k_o}^{-1}\mathbf{I}_m\big).$$

Considering the above expression, the posterior distribution over a dictionary atom can be written as

$$p(\boldsymbol{\varphi}_k|-) = \mathcal{N}\big(\boldsymbol{\varphi}_k|\boldsymbol{\mu}_k, \lambda_k^{-1}\mathbf{I}_m\big), \tag{8}$$

where,

$$\boldsymbol{\mu}_k = \frac{\lambda_{\epsilon_o}}{\lambda_k}\sum_{i=1}^{N}(z_{ik}.s_{ik})\mathbf{x}_{i_{\varphi_k}},\ \ \lambda_k = \lambda_{k_o} + \lambda_{\epsilon_o}\sum_{i=1}^{N}(z_{ik}.s_{ik})^2.$$

Sampling z_{ik}^c. Once the dictionary atoms have been sampled, we sample z_{ik}^c, $\forall i \in \mathcal{I}_c$, $\forall k \in \mathcal{K}$. Using the contribution of the kth dictionary atom, the posterior probability distribution over z_{ik}^c can be expressed as

$$p\big(z_{ik}^c| - \big) \propto \mathcal{N}\big(\mathbf{x}_{i_{\varphi_k}}^c|\boldsymbol{\varphi}_k\big(z_{ik}^c.s_{ik}^c\big), \lambda_{\epsilon_o}^{-1}\mathbf{I}_m\big)\text{Bernoulli}\big(z_{ik}^c|\pi_{k_o}^c\big).$$

Here we are concerned with the cth class only, therefore $\mathbf{x}_{i_{\varphi_k}}^c$ is computed with the cth class data in Eq. (7). With the prior probability of $z_{ik}^c = 1$ given by $\pi_{k_o}^c$, we can write the following about its posterior probability:

$$p\big(z_{ik}^c = 1| - \big) \propto \pi_{k_o}^c \exp\left(-\frac{\lambda_{\epsilon_o}}{2}||\mathbf{x}_{i_{\varphi_k}}^c - \boldsymbol{\varphi}_k s_{ik}^c||_2^2\right).$$

It can be shown that the right hand side of the above proportionality can be written as:

$$p_1 = \pi_{k_o}^c \zeta_1 \zeta_2,$$

where, $\zeta_1 = \exp(-\frac{\lambda_{\epsilon_o} s_{ik}^c}{2}(||\boldsymbol{\varphi}_k||_2^2 s_{ik}^c - 2(\mathbf{x}_{i_{\varphi_k}}^c)^{\mathrm{T}}\boldsymbol{\varphi}_k))$ and $\zeta_2 = \exp(-\frac{\lambda_{\epsilon_o}}{2}||\mathbf{x}_{i_{\varphi_k}}^c||_2^2)$. Furthermore, since the prior probability of $z_{ik}^c = 0$ is given by $1 - \pi_{k_o}^c$, we can write the following about its posterior probability:

$$p\big(z_{ik}^c = 0| - \big) \propto \big(1 - \pi_{k_o}^c\big)\zeta_2.$$

Thus, z_{ik}^c can be sampled from the following normalized Bernoulli distribution:

$$\text{Bernoulli}\left(z_{ik}^c\Big|\frac{p_1}{p_1 + (1 - \pi_{k_o}^c)\zeta_2}\right).$$

By inserting the value of p_1 and simplifying, we finally arrive at the following expression for sampling z_{ik}^c:

$$z_{ik}^c \sim \text{Bernoulli}\left(z_{ik}^c\Big|\frac{\pi_{k_o}^c \zeta_1}{1 + \pi_{k_o}^c(\zeta_1 - 1)}\right). \tag{9}$$

Sampling s_{ik}^c. We can write the following about the posterior distribution over s_{ik}^c:

$$p\big(s_{ik}^c| - \big) \propto \mathcal{N}\big(\mathbf{x}_{i_{\varphi_k}}^c|\boldsymbol{\varphi}_k\big(z_{ik}^c.s_{ik}^c\big), \lambda_{\epsilon_o}^{-1}\mathbf{I}_m\big)\mathcal{N}\big(s_{ik}^c|0, 1/\lambda_{s_o}^c\big).$$

Again, notice that we are concerned with the cth class data only. In light of the above expression, s_{ik}^c can be sampled from the following posterior distribution:

$$p\big(s_{ik}^c| - \big) = \mathcal{N}\big(s_{ik}^c|\mu_s^c, 1/\lambda_s^c\big), \tag{10}$$

where, $\mu_s^c = \frac{\lambda_{\epsilon_o}}{\lambda_s^c} z_{ik}^c \boldsymbol{\varphi}_k^{\mathrm{T}} \mathbf{x}_{i_{\varphi_k}}^c$, $\lambda_s^c = \lambda_{s_o}^c + \lambda_{\epsilon_o}(z_{ik}^c)^2||\boldsymbol{\varphi}_k||_2^2$.

Sampling π_k^c. Based on our model, we can also write the posterior probability distribution over π_k^c as

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

Using the conjugacy between the distributions, it can be easily shown that the kth component of $\boldsymbol{\pi}^c$ must be drawn from the following posterior distribution during the sampling process:

$$p(\pi_k^c|-) = \text{Beta}\left(\pi_k^c\Big|\frac{a_o}{K} + \sum_{i \in \mathcal{I}_c} z_{ik}^c, \frac{b_o(K-1)}{K} + |\mathcal{I}_c| - \sum_{i \in \mathcal{I}_c} z_{ik}^c\right). \tag{11}$$

Sampling λ_s^c. In our model, the components of the weight vectors are drawn from a standard normal distribution. For a given weight vector, common priors are assumed over the precision parameters of these distributions. This allows us to express the likelihood function for λ_s^c in terms of standard multivariate Gaussian with isotropic precision. Thus, we can write the posterior over λ_s^c as the following:

$$p\big(\lambda_s^c| - \big) \propto \prod_{i \in \mathcal{I}_c} \mathcal{N}\left(\mathbf{s}_{\mathbf{i}}^{\boldsymbol{\varsigma}}\Big|\mathbf{0}, \frac{1}{\lambda_{s_o}^c}\mathbf{I}_{|\mathcal{K}|}\right)\Gamma\big(\lambda_{s_o}^c|c_o, d_o\big).$$

Using the conjugacy between the Gaussian and Gamma distributions, it can be shown that λ_s^c must be sampled as:

$$\lambda_s^c \sim \Gamma\left(\lambda_s^c \middle| \frac{|\mathcal{K}|N_c}{2} + c_o, \frac{1}{2}\sum_{i\in\mathcal{I}_c} ||\mathbf{s}_i^c||_2^2 + d_o\right). \quad (12)$$

Sampling λ_ϵ. We can write the posterior over λ_ϵ as

$$p(\lambda_\epsilon|-) \propto \prod_{i=1}^{N} \mathcal{N}\big(\mathbf{x}_i|\mathbf{\Phi}(\mathbf{z}_i \odot \mathbf{s}_i), \lambda_{\epsilon_o}^{-1}\mathbf{I}_m\big)\Gamma(\lambda_{\epsilon_o}|e_o, f_o).$$

Similar to λ_s^c, we can arrive at the following for sampling λ_ϵ during the inferencing process:

$$\lambda_\epsilon \sim \Gamma\left(\frac{mN}{2} + e_o, \frac{1}{2}\sum_{i=1}^{N} ||\mathbf{x}_i - \mathbf{\Phi}(\mathbf{z}_i \odot \mathbf{s}_i)||_2^2 + f_o\right). \quad (13)$$

As a result of Bayesian inference, we obtain sets of posterior distributions over the model parameters. We are particularly interested in two of them. Namely, the set of distributions over the dictionary atoms $\aleph \stackrel{\text{def}}{=} \{\mathcal{N}(\boldsymbol{\varphi}_k|\boldsymbol{\mu}_k, \mathbf{\Lambda}_k^{-1}) : k \in \mathcal{K}\} \subset \mathbb{R}^m$, and the set of probability distributions characterized by the vectors $\boldsymbol{\pi}^{c\in\{1,\ldots,C\}} \in \mathbb{R}^{|\mathcal{K}|}$. Momentarily, we defer the discussion on the latter. The former is used to compute the desired dictionary $\mathbf{\Phi}$. This is done by drawing multiple samples from the elements of $\aleph$ and estimating the corresponding dictionary atoms as respective means of the samples. Indeed, the mean parameters of the elements of $\aleph$ can also be chosen as the desired dictionary atoms. However, we adopt the former approach because it also accounts for the precisions of the posterior distributions while computing the final dictionary. Although the difference in the classification performance resulting from the two approaches is generally very small, the adopted approach is preferred as it also adds to the robustness of the dictionary against the noise in the training data [72].

Our model allows to estimate the desired size of the dictionary non-parametrically. We present Lemma 4.1 regarding the expected size of the dictionary according to our model. In Lemma 4.2, we make an observation that is exploited in the sampling process to estimate this size.

Lemma 4.1. *For a very large K, $\mathbb{E}[\xi] = \frac{a_o}{b_o}$, where $\xi = \sum_{k=1}^{K} z_{ik}^c$.*

Proof. [4] According to the proposed model, the covariance of a data vector from the cth class, i.e., $\mathbf{x}_i^c$ can be given by:

$$\mathbb{E}\big[\big(\mathbf{x}_i^c\big)\big(\mathbf{x}_i^c\big)^{\mathrm{T}}\big] = \frac{a_o K}{a_o + b_o(K-1)}\frac{\mathbf{\Lambda}_{k_o}^{-1}}{\lambda_{s_o}^c} + \mathbf{\Lambda}_{\epsilon_o}^{-1}. \quad (14)$$

□

In Eq. (14), fraction $\frac{a_o}{a_o+b_o(K-1)}$ appears due to the presence of $\mathbf{z}_i^c$ in the model and the equation simplifies to $\mathbb{E}[(\mathbf{x}_i^c)(\mathbf{x}_i^c)^{\mathrm{T}}] = K\frac{\mathbf{\Lambda}_{k_o}^{-1}}{\lambda_{s_o}^c} + \mathbf{\Lambda}_{\epsilon_o}^{-1}$ when we neglect $\mathbf{z}_i^c$. Here, K signifies the number of dictionary atoms required to represent the data vector. In the equation, as K becomes very large, $\mathbb{E}[(\mathbf{x}_i^c)(\mathbf{x}_i^c)^{\mathrm{T}}] \to \frac{a_o}{b_o}\frac{\mathbf{\Lambda}_{k_o}^{-1}}{\lambda_{s_o}^c} + \mathbf{\Lambda}_{\epsilon_o}^{-1}$. Thus, for a large dictionary, the expected number of atoms required to represent $\mathbf{x_i}^c$ is given by $\frac{a_o}{b_o}$. Meaning, $\mathbb{E}[\xi] = \frac{a_o}{b_o}$, where $\xi = \sum_{k=1}^{K} z_{ik}^c$.

Lemma 4.2. *Once $\pi_k^c = 0$ in a given iteration of the sampling process, $\mathbb{E}[\pi_k^c] \approx 0$ for the later iterations.*

Proof. According to Eq. (9), $\forall i \in \mathcal{I}_c$, $z_{ik}^c = 0$ when $\pi_{k_o}^c = 0$. Once this happens, the posterior distribution over π_k^c becomes $\text{Beta}\big(\pi_k^c\big|\hat{a}, \hat{b}\big)$, where $\hat{a} = \frac{a_o}{K}$ and $\hat{b} = \frac{b_o(K-1)}{K} + |\mathcal{I}_c|$ (see Eq. (11)). Thus, the expected value of π_k^c for the later iterations can be written as $\mathbb{E}[\pi_k^c] = \frac{\hat{a}}{\hat{a}+\hat{b}} = \frac{a_o}{a_o+b_o(K-1)+K|\mathcal{I}_c|}$. With $0 < a_o, b_o < |\mathcal{I}_c| \ll K$ we can see that $\mathbb{E}[\pi_k^c] \approx 0$. □

Considering Lemma 4.1, we start with a very large value of K in the Gibbs sampling process. We let $K = 1.5 \times N$ and let $0 < a_o, b_o < |\mathcal{I}_c|$ to ensure that the resulting representation is sparse. We drop the kth dictionary atom during the sampling process if $\pi_k^c = 0$, for all the classes simultaneously. According to Lemma 4.2, dropping such an atom does not bring significant changes to the final representation. Thus, by removing the redundant dictionary atoms in each sampling iteration, we finally arrive at the correct size of the dictionary, i.e., $|\mathcal{K}|$.

As mentioned above, with Bayesian inference over the proposed model we also infer a set of probability vectors $\boldsymbol{\pi}^{c\in\{1,\ldots,C\}}$. Each element of this set, i.e., $\boldsymbol{\pi}^c \in \mathbb{R}^{|\mathcal{K}|}$, further characterizes a set of probability distributions $\Im^c \stackrel{\text{def}}{=} \{\text{Bernoulli}(\pi_k^c) : k \in \mathcal{K}\} \subset \mathbb{R}$. Here, $\text{Bernoulli}(\pi_k^c)$ is jointly followed by all the kth components of the sparse codes for the cth class. If the kth dictionary atom is commonly used in representing the cth class training data, we must expect a high value of π_k^c, and $\pi_k^c \to 0$ otherwise. In other words, for an arranged dictionary, components of $\boldsymbol{\pi}^c$ having large values should generally cluster well if the learned dictionary is discriminative. Furthermore, these clusters must appear at different locations in the inferred vectors for different classes. Such clusterings would demonstrate the discriminative character of the inferred dictionary. Fig. 4 verifies this character for the dictionaries inferred under the proposed model. Each row of the figure plots six different probability vectors (i.e., $\boldsymbol{\pi}^c$) for different training datasets. A clear clustering of the high value components of the vectors is visible in each plot. In the supplementary material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2016.2527652 of the paper, we also illustrate few dictionary atoms corresponding to the largest values of π_k^c for the Extended YaleB database [2]. Detailed experiments are presented in Section 5.

Whereas clear clusters are visible in Fig. 4, we can also see few non-zero values appearing far from the main clusters. These values indicate the sharing of atoms among the data representations of different classes. We note that our model allows such sharing because it employs finite approximation of the Beta Process. Such a model is sufficient for practical classification tasks where the training data size is always finite and known a priori. Our model only requires K to be larger than the training data size. We also note that the model does not allow the atom sharing

4. We follow [56] closely in the proof, however, our analysis also takes into account the class labels of the data, whereas no such data discrimination is assumed in [56].

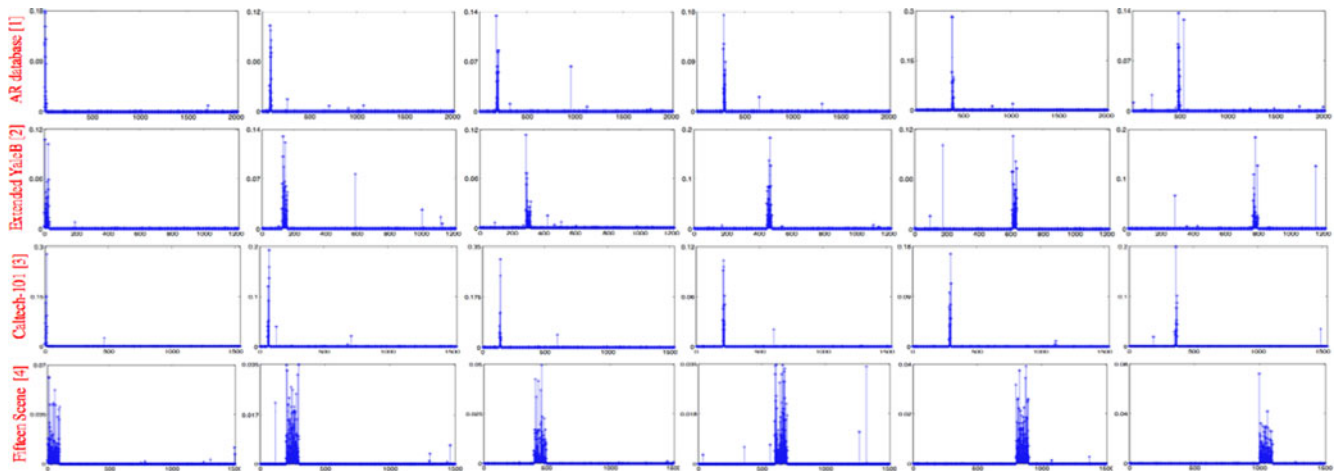


Fig. 4. Illustration of the discriminative character of the inferred dictionary: From top, the four rows present results on AR database [1], Extended YaleB [2], Caltech-101[3] and 15 Scene categories [4], respectively. In each plot, the *x*-axis represents $k \in \mathcal{K}$ and the *y*-axis shows the corresponding probability of selection of the kth dictionary atom in the expansion of the data. A plot represents a single $\boldsymbol{\pi}^c$ vector learned as a result of Bayesian inference. For the first three rows, from left to right, the value of c (i.e., class label) is 1, 5, 10, 15, 20 and 25, respectively. For the fourth row the value of c is 1, 3, 5, 7, 9 and 11 for the plots from left to right. Plots clearly show distinct clusters of high probabilities for different classes.

between the classes if K is infinite. In that case, an atom of the dictionary will correspond to only a single class. This is similar to the first category of the discriminative dictionary learning approaches discussed in Section 2.

4.3 Classification

To classify a query $\mathbf{y} \in \mathbb{R}^m$, we first compute its sparse representation $\widehat{\boldsymbol{\alpha}}$ over the learned dictionary. The label of the query is predicted by maximizing the coefficients of the vector $\boldsymbol{\ell} = \mathbf{W}\widehat{\boldsymbol{\alpha}}$, where $\mathbf{W} \in \mathbb{R}^{C \times |\mathcal{K}|}$ denotes a multi-class linear classifier. Effectiveness of learning such a classifier in accordance with the dictionary is already established for discriminative dictionary learning [7], [9]. Therefore, we also couple our classifier with the learned dictionary. Nevertheless, we keep the learning processes of the dictionary and the classifier disjoint to fully exploit the potential of our model. Further discussion in this regard is deferred to Section 6. In order to learn the classifier, we first define a vector $\mathbf{h}_i^c \in \mathbb{R}^C$ for each class. These vectors are computed using $\boldsymbol{\pi}^{c \in \{1,\ldots,C\}}$ inferred by the dictionary learning process. For the cth class, the qth coefficient h_{iq}^c of $\mathbf{h}_i^c$ is computed as $h_{iq}^c = \sum_{k \in \mathcal{C}} \pi_k^q$ where, $\mathcal{C}$ indexes the non-zero coefficients of $\boldsymbol{\pi}^c$. Considering that large non-zero coefficients of $\boldsymbol{\pi}^c$ generally appear at the locations corresponding to the cth class, h_{iq}^c is large when $q = c$ and small otherwise. After normalization, we use the computed vectors as the training data for the classifier. The training is done by solving $\mathbf{h}_i^c = \mathbf{W}\boldsymbol{\beta}_i^c + \boldsymbol{\epsilon}_i$ using the model in Eq. (6). Here, $\boldsymbol{\beta}_i^c \in \mathbb{R}^{|\mathcal{K}|}$ is a sparse coefficient vector defined over $\mathbf{W}$, just as $\boldsymbol{\alpha}_i^c$ was defined over the dictionary.

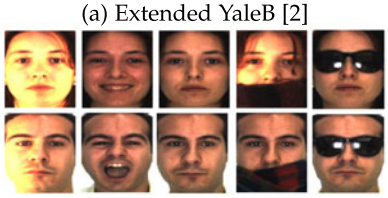
The inference process for learning the classifier is also guided by the probability vectors $\boldsymbol{\pi}^{c \in \{1,\ldots,C\}}$ computed by the dictionary learning stage. We directly use these vectors for classifier learning and keep them fixed during the complete sampling process. Notice that, our sampling process computes a basis keeping in view the support of its coefficient vectors, i.e., $\boldsymbol{\varphi}_k$ depends on z_{ik} in Eq. (8). Since the support of $\boldsymbol{\alpha}_i^c$ and $\boldsymbol{\beta}_i^c$ follow the same set of probability distributions, given by $\boldsymbol{\pi}^{c \in \{1,\ldots,C\}}$, a coupling is induced between their inferred bases, i.e., $\boldsymbol{\Phi}$ and $\mathbf{W}$. This forces the learned parameters of $\mathbf{W}$ to respect the popularity of the dictionary atoms for representing the class-specific training data. Since the popularity of the atoms is expected to remain consistent across the training and the test data for a given class, we can directly use the classifier with the sparse codes of the test data to correctly predict its class label.

To classify a query, we first find its sparse representation over the learned dictionary. Keeping in view that our dictionary learning model imposes sparsity on a coefficient vector by forcing many of its components to zero, we choose Orthogonal Matching Pursuit (OMP) algorithm [60] to efficiently compute the sparse representation of the query signal. OMP allows only a few non-zero components in the representation vector to maximally approximate the query signal using the dictionary. Therefore, the popular dictionary atoms for the correct class of the query usually contribute significantly to the representation. This helps in accurate classification using $\mathbf{W}$. Notice that, to predict the label of a K-dimensional sparse vector, our approach only has to multiply it with a $C \times K$-dimensional matrix and search for the maximum value in the resulting C-dimensional vector. This makes our classification approach much efficient compared to the alternative of using a sophisticated classifier like SVM to classify the sparse codes of the query. Since efficient classification of a query signal is one of the major goals of discriminative dictionary learning, we consider our approach highly attractive.

4.4 Initialization

For inferring the dictionary, we need to first initialize $\boldsymbol{\Phi}$, $\mathbf{z}_i^c$, $\mathbf{s}_i^c$ and π_k^c. We initialize $\boldsymbol{\Phi}$ by randomly selecting the training instances with replacement. We sparsely code $\mathbf{x}_i^c$ over the initial dictionary using OMP [60]. The codes are considered as the initial $\mathbf{s}_i^c$, whereas their support forms the initial vector $\mathbf{z}_i^c$. Computing the initial $\mathbf{s}_i^c$ and $\mathbf{z}_i^c$ with other methods, such as regularized least squares, is equally effective. We set $\pi_k^c = 0.5, \forall c, \forall k$ for the initialization. Notice, this means that all the dictionary atoms initially have equal chances of getting selected in the expansion of a training instance from

(a) Extended YaleB [2]



(b) AR database [1]

Fig. 5. Examples from the face databases.

any class. The values of $\pi_k^c, \forall c, \forall k$ finally inferred by the dictionary learning process serve as the initial values of these parameters for learning the classifier. Similarly, $\mathbf{z}_i^c$ and $\mathbf{s}_i^c$ computed by the dictionary learning stage are used for initializing the corresponding vectors for the classifier. We initialize $\mathbf{W}$ using the ridge regression [61] with the ℓ_2-norm regularizer and quadratic loss:

$$\mathbf{W} = \min_{W} ||\mathbf{H} - \mathbf{W}\boldsymbol{\alpha}_i||^2 + \lambda||\mathbf{W}||_2^2, \ \forall i \in \{1, \ldots, N\}, \qquad (15)$$

where λ is the regularization constant. The computation is done over the complete training data, therefore the superscript 'c' is dropped in the above equation. Similar to the existing approaches [7], [9], we consider the initialization procedure as an integral part of the proposed approach.

5 Experiments

We have evaluated the proposed approach on two face data sets: the Extended YaleB [2] and the AR database [1], a data set for object categories: Caltech-101 [3], a data set for scene categorization: 15 scene categories [4], and an action data set: UCF sports actions [5]. These data sets are commonly used in the literature for evaluation of sparse representation based classification techniques. We compare the performance of the proposed approach with SRC [8], the two variants of Label-Consistent K-SVD [9] (i.e., LC-KSVD1, LC-KSVD2), the Discriminative K-SVD algorithm (D-KSVD) [7], the Fisher Discrimination Dictionary Learning algorithm (FDDL) [10] and the Dictionary Learning based on separating the Commonalities and the Particularities of the data (DL-COPAR) [11]. In our comparisons, we also include results of unsupervised sparse representation based classification that uses K-SVD [6] as the dictionary learning technique and separately computes a multi-class linear classifier using Eq. (15).

For all of the above mentioned methods, except SRC and D-KSVD, we acquired the public codes from the original authors. To implement SRC, we used the LASSO [63] solver of the SPAMS toolbox [62]. For D-KSVD, we used the public code provided by Jiang et al. [9] for LC-KSVD2 algorithm and solved Eq. (3) with $\upsilon = 0$. In all experiments, our approach uses the implementation of OMP made public by Elad et al. [71]. K-SVD, D-KSVD, LC-KSVD1 and LC-KSVD2 also use the same implementation. The experiments are performed on an Intel Core i7-2600 CPU at 3.4 GHz with 8 GB RAM. We performed our own experiments using the above mentioned methods and the proposed approach using the same data. The parameters of the existing approaches were carefully optimized following the guidelines of the original works. We mention the used parameter values and, where it exists, we note the difference between our values and those used in the original works. In our experiments, these differences were made to favor the existing approaches. Results of the approaches other than those mentioned above, are taken directly from the literature, where the same experimental protocol has been followed.

TABLE 1
Recognition Accuracy with Random-Face Features on the Extended YaleB Database [2]

Method	Accuracy %	Average Time (ms)
LLC [15]	90.7	-
K-SVD [6]	93.13 ± 0.43	0.37
LC-KSVD1 [9]	93.59 ± 0.54	0.36
D-KSVD [7]	94.79 ± 0.49	0.38
DL-COPAR [11]	94.83 ± 0.52	32.55
LC-KSVD2 [9]	95.22 ± 0.61	0.39
FDDL [10]	96.07 ± 0.64	49.59
DBDL+SVM	96.10 ± 0.25	426.14
SRC [8]	96.32 ± 0.85	53.12
Proposed	$\mathbf{97.31 \pm 0.67}$	1.22

The computed average time is for classification of a single instance.

For the proposed approach, the used parameter values were as follows. In all experiments, we chose $K = 1.5N$ for initialization, whereas c_o, d_o, e_o and f_o were all set to 10^{-6}. We selected $a_o = b_o = \frac{\min_c |\mathcal{I}_c|}{2}$, whereas λ_{s_o} and λ_{k_o} were set to 1 and m, respectively. Furthermore, λ_{ϵ_o} was set to 10^6 for all the datasets except for 15 Scene Categories [4], where we used $\lambda_{\epsilon_o} = 10^9$. In each experiment, we ran 500 Gibbs sampling iterations that proved sufficient for accurate inference using our approach. We provide assessment of the inference accuracy of the performed Gibbs sampling in the supplementary material, available online, of the paper. We defer further discussion on the selection of the parameter values to Section 6.

5.1 Extended YaleB

Extended YaleB [2] contains 2,414 frontal face images of 38 different people, each having about 64 samples. The images are acquired under varying illumination conditions and the subjects have different facial expressions. This makes the database fairly challenging, see Fig. 5a for a few examples. In our experiments, we used the random face feature descriptor [8], where a cropped 192×168 pixels image was projected onto a 504-dimensional vector. For this, the projection matrix was generated from random samples of standard normal distributions. Following the common settings for this database, we chose one half of the images for training and the remaining samples were used for testing. We performed 10 experiments by randomly selecting the samples for training and testing. Based on these experiments, the mean recognition accuracies of different approaches are reported in Table 1. The results for Locality-constrained Linear Coding (LLC) [15] is directly taken from [9], where the accuracy is computed using 70 local bases.

TABLE 2
Recognition Accuracy with Random-Face Features on the AR Database [1]

Method	Accuracy %	Average Time (ms)
LLC [15]	88.7	-
DL-COPAR [11]	93.23 ± 1.71	39.80
LC-KSVD1 [9]	93.48 ± 1.13	0.98
LC-KSVD1‡	87.48 ± 1.19	0.37
K-SVD [6]	94.13 ± 1.20	0.99
LC-KSVD2 [9]	95.33 ± 1.24	1.01
LC-KSVD2‡	88.35 ± 1.33	0.41
D-KSVD [7]	95.47 ± 1.50	1.01
D-KSVD‡	88.29 ± 1.38	0.38
DBDL+SVM	95.69 ± 0.73	1040.01
FDDL [10]	96.22 ± 1.03	50.03
SRC [8]	96.65 ± 1.37	62.86
Proposed	$\mathbf{97.47 \pm 0.99}$	1.28

The computed time is for classifying a single instance. The ‡ sign denotes the results using the parameter settings reported in the original works.

Similar to Jiang et al. [9], the sparsity threshold for K-SVD, LC-KSVD1, LC-KSVD2 and D-KSVD was set to 30 in our experiments. Larger values of this parameter were found to be ineffective as they mainly resulted in slowing the algorithms without improving the recognition accuracy. Furthermore, as in [9], we used $\upsilon = 4.0$ for LC-KSVD1 and LC-KSVD2, whereas κ was set to 2.0 for LC-KSVD2 and D-KSVD in Eq. (3). Keeping these parameter values fixed, we optimized for the number of dictionary atoms for each algorithm. This resulted in selecting 600 atoms for LC-KSVD2, D-KSVD and K-SVD, whereas 500 atoms consistently resulted in the best performance of LC-KSVD1. This value is set to 570 in [9] for all of the four methods. In all techniques that learn dictionaries, we used the complete training data in the learning process. Therefore, all training samples were used as dictionary atoms for SRC. Following [8], we set the residual error tolerance to 0.05 for SRC. Smaller values of this parameter also resulted in very similar accuracies. For FDDL, we followed [10] for the optimized parameter settings. These settings are the same as those reported for AR database in the original work. We refer the reader to the original work for the list of the parameters and their exact values. The results reported in the table are obtained by the Global Classifier (GC) of FDDL, which showed better performance than the Local Classifier (LC). For the parameter settings of DL-COPAR we followed the original work [11]. We fixed 15 atoms for each class and a set of 5 atoms was chosen to learn commonalities of the classes. The reported results are achieved by LC, that performed better than GC in our experiments.

It is clear from Table 1 that our approach outperforms the above mentioned approaches in terms of recognition accuracy, with nearly 23 percent improvement over the error rate of the second best approach. Furthermore, the time required by the proposed approach for classifying a single test instance is also very low as compared to SRC, FDDL and DL-COPAR. For the proposed approach, this time is comparable to D-KSVD and LC-KSVD. Like these algorithms, the computational efficiency in the classification stage of our approach comes from using the learned multiclass linear classifier to classify the sparse codes of a test instance. To show the computational benefits of the proposed classifier over SVM, we also include the results of using SVM on the sparse code features of the query. In the table, DBDL+SVM refers to these results. Note that, our classifier also used the same features.

5.2 AR Database

This database contains more than 4,000 face images of 126 people. There are 26 images per person taken during two different sessions. The images in AR database have large variations in terms of facial expressions, disguise and illumination conditions. Samples from AR database are shown in Fig. 5b for illustration. We followed a common evaluation protocol in our experiments for this database, in which we used a subset of 2,600 images pertaining to 50 males and 50 female subjects. For each subject, we randomly chose 20 samples for training and the rest for testing. The 165×120 pixel images were projected onto a 540-dimensional vector with the help of a random projection matrix, as in Section 5.1. We report the average recognition accuracy of our experiments in Table 2, which also includes the accuracy of LLC [15] reported in [9]. The mean values reported in the table are based on 10 experiments.

In our experiments, we set the sparsity threshold for K-SVD, LC-KSVD1, LC-KSVD2 and D-KSVD to 50 as compared to 10 and 30 which was used in [7] and [9], respectively. Furthermore, the dictionary size for K-SVD, LC-KSVD2 and D-KSVD was set to 1,500 atoms, whereas the dictionary size for LC-KSVD1 was set to 750. These large values (compared to 500 used in [7], [9]) resulted in better accuracies at the expense of more computation. However, the classification time per test instance remained reasonably small. In Table 2, we also include the results of LC-KSVD1, LC-KSVD2 and D-KSVD using the parameter values proposed in the original works. These results are distinguished with the ‡ sign. For FDDL and DL-COPAR we used the same parameter settings as in Section 5.1. The reported results are for GC and LC for FDDL and DL-COPAR, respectively. For SRC we set the residual error tolerance to 10^{-6}. This small value gave the best results.

From Table 2, we can see that the proposed approach performs better than the existing approaches in terms of accuracy. The recognition accuracies of SRC and FDDL are fairly close to our approach however, these algorithms require large amount of time for classification. This fact compromises their practicality. In contrast, the proposed approach shows high recognition accuracy (i.e., 22 percent reduction in the error rate as compared to SRC) with less than 1.5 ms required for classifying a test instance. The relative difference between the classification time of the proposed approach and the existing approaches remains similar in the experiments below. Therefore, we do not explicitly note these timings for all of the approaches in these experiments.

5.3 Caltech-101

The Caltech-101 database [3] comprises $9{,}144$ samples from 102 classes. Among these, there are 101 object classes (e.g., minarets, trees, signs) and one "background" class. The number of samples per class varies from 31 to 800, and the images within a given class have significant shape variations, as can be seen in Fig. 6. To use the database, first the SIFT

Fig. 6. Examples from Caltech-101 database [3]. The proposed approach achieves 100 percent accuracy on these classes.

descriptors [64] were extracted from 16×16 image patches, which were densely sampled with a 6-pixels step size for the grid. Then, based on the extracted features, spatial pyramid features [38] were extracted with $2^l \times 2^l$ grids, where $l = 0, 1, 2$. The codebook for the spatial pyramid was trained using k-means with $k = 1{,}024$. Then, the dimension of a spatial pyramid feature was reduced to $3{,}000$ using PCA. Following the common experimental protocol, we selected 5, 10, 15, 20, 25 and 30 instances for training the dictionary and the remaining instances were used in testing, in our six different experiments. Each experiment was repeated 10 times with random selection of train and test data. The mean accuracies of these experiments are reported in Table 3.

For this dataset, we set the number of dictionary atoms used by K-SVD, LC-KSVD1, LC-KSVD2 and D-KSVD to the number of training examples available. This resulted in the best performance of these algorithms. The sparsity level was also set to 50 and υ and κ were set to 0.001. Jiang et al. [9] also suggested the same parameter settings. For SRC, the error tolerance of 10^{-6} gave the best results in our experiments. We used the parameter settings for object categorization given in [10] for FDDL. For DL-COPAR, the selected number of class-specific atoms were kept the same as the number of training instances per class, whereas the number of shared atoms were fixed to 314, as in the original work [11]. For this database GC performed better than LC for DL-COPAR in our experiments.

From Table 3, it is clear that the proposed approach consistently outperforms the competing approaches. For some cases the accuracy of LC-KSVD2 is very close to the proposed approach, however with the increasing number of training instances the difference between the results increases in favor of the proposed approach. This is an expected phenomenon since more training samples result in more precise posterior distributions in Bayesian settings. We provide further discussion on dependence of our approach's performance on training data size in the supplementary material, available online, of the paper. Here, it is also worth mentioning that being Bayesian, the proposed approach is inherently an online technique. This means, in our approach, the computed posterior distributions can be used as prior distributions for further inference if more training data is available. Moreover, our approach is able to handle a batch of large training data more efficiently than LC-KSVD [9] and D-KSVD [7]. This can be verified by comparing the training time of the approaches in Table 4. The timings are given for complete training and testing durations for Caltech-101 database, where we used a batch of 30 images per class for training and the remaining images were used for testing. We note that, like all the other approaches, good initialization (using the procedure presented in Section 4.4) also contributes towards the computational efficiency of our approach. The training time in the table also includes the initialization time for all the approaches. Note that the testing time of the proposed approach is very similar to those of the other approaches in Table 4.

TABLE 3
Classification Results Using Spatial Pyramid Features on the Caltech-101 Dataset [3]

Total training samples	5	10	15	20	25	30
Zhang et al. [37]	46.6	55.8	59.1	62.0	-	66.20
Lazebnik et al. [38]	-	-	56.4	-	-	64.6
Griffin et al. [39]	44.2	54.5	59.0	63.3	65.8	67.6
Wang et al. [15]	51.1	59.8	65.4	67.7	70.2	73.4
SRC [8]	49.9	60.1	65.0	67.5	69.3	70.9
DL-COPAR [11]	49.7	58.9	65.2	69.1	71.0	72.9
K-SVD [6]	51.2	59.1	64.9	68.7	71.0	72.3
FDDL [10]	52.1	59.8	66.2	68.9	71.3	73.1
D-KSVD [7]	52.1	60.8	66.1	69.6	70.8	73.1
LC-KSVD1 [9]	53.1	61.2	66.3	69.8	71.9	73.5
LC-KSVD2 [9]	53.8	62.8	67.3	70.4	72.6	73.9
Proposed	**53.9**	**63.1**	**68.1**	**71.0**	**73.3**	**74.6**

TABLE 4
Computation Time for Training and Testing on Caltech-101 Database

Method	Training (sec)	Testing (sec)
Proposed	1,474	19.96
D-KSVD [7]	3,196	19.90
LC-KSVD1 [9]	5,434	19.65
LC-KSVD2 [9]	5,434	19.92

5.4 Fifteen Scene Category

The 15 Scene Category dataset [4] has 200 to 400 images per category for 15 different kinds of scenes. The scenes include images from kitchens, living rooms and country sides etc. See Fig. 7 for examples. In our experiments, we used the Spatial Pyramid Features of the images, which have been made public by Jiang et al. [9]. In this data, each feature descriptor is a 3,000-dimensional vector. Using these features, we performed experiments by randomly selecting 100 training instances per class and considering the remaining as the test instances.

Classification accuracy of the proposed approach is compared with the existing approaches in Table 5. The reported values are computed over 10 experiments. We set the error tolerance for SRC to 10^{-6} and used the parameter settings suggested by Jiang et al. [9] for LC-KSVD1, LC-KSVD2 and

Fig. 7. Examples images from eight different categories in 15 Scene categories dataset [4].

TABLE 5
Classification Accuracy on 15 Scene Category Dataset [4] Using Spatial Pyramid Features

Method	Accuracy %
K-SVD [6]	93.60 ± 0.14
LC-KSVD1[9]	94.05 ± 0.17
D-KSVD [7]	96.11 ± 0.12
SRC [8]	96.21 ± 0.09
DL-COPAR [11]	96.91 ± 0.22
LC-KSVD2 [9]	97.01 ± 0.23
Proposed	$\mathbf{98.73 \pm 0.17}$

D-KSVD. Parameters of DL-COPAR were set as suggested in the original work [11] for the same database. The reported results are obtained by LC for DL-COPAR. Again, the proposed approach shows more accurate results than the existing approaches. The accuracy of the proposed approach is 1.66 percent more than LC-KSVD2 on the used dataset.

5.5 UCF Sports Action

This database comprises video sequences that are collected from different broadcast sports channels (e.g., ESPN and BBC) [5]. The videos contain 10 categories of sports actions that include: kicking, golfing, diving, horse riding, skateboarding, running, swinging, swinging highbar, lifting and walking. Examples from this dataset are shown in Fig. 8. Under the common evaluation protocol we performed five-fold cross validation over the dataset, where four folds are used in training and the remaining one is used for testing. Results, computed as the average of the five experiments, are summarized in Table 6. For D-KSVD, LC-KSVD1 and LC-KSVD2 we followed [9] for the parameter settings. Again, the value of 10^{-6} (along with similar small values) resulted in the best accuracies for SRC.

In the Table, the results for some specific action recognition methods are also included, for instance, Qui et al. [33] and action back feature with SVM [40]. These results are taken directly from [13] along the results of DLSI [12], DL-COPAR [11] and FDDL [10].[5] Following [40], we also performed leave-one-out cross validation on this database for the proposed approach. Our approach achieves 95.7 percent accuracy under this protocol, which is 0.7 percent better than the state-of-the-art results claimed in [40].

6 Discussion

In our experiments, we used large values for λ_{ϵ_o}, because this parameter represents the precision of the white noise distribution in the samples. The datasets used in our experiments are mainly clean in terms of white noise. Therefore, we achieved the best performance with $\lambda_{\epsilon_o} \geq 10^6$. In the case of noisy data, this parameter value can be adjusted accordingly. For UCF sports action dataset $\lambda_{\epsilon_o} = 10^9$ gave the best results because less number of training samples were available per class. It should be noted that the value of λ_ϵ increases as a result of Bayesian inference with the availability of more clean training samples. Therefore, we adjusted the precision parameter of the prior distribution to a larger value for UCF dataset. Among the other parameters, c_o to f_o were fixed to 10^{-6}. Similar small non-negative values can also be used without affecting the results. This fact can be easily verified by noticing the large values of the other variables involved in Eq. (12) and (13), where these parameters are used. With the above mentioned parameter settings and the initialization procedure presented in Section 4.4, the Gibbs sampling process converges quickly to the desired distributions and the correct number of dictionary atoms, i.e., $|\mathcal{K}|$. In Fig. 9, we plot the value of $|\mathcal{K}|$ as a function of Gibbs sampling iterations during dictionary training. It can be easily seen that the first few iterations of the Gibbs sampling process were generally enough to converge to the correct size of the dictionary. However, it should be mentioned that this fast convergence also owes to the initialization process adopted in this work. In our experiments, while sparse coding a test instance over the learned dictionary, we consistently used the sparsity threshold of 50 for all the datasets except for the UCF [5], for which this parameter was set to 40 because of the smaller dictionary resulting from less training samples. In all the experiments, this parameter value was also kept the same for K-SVD, LC-KSVD1, LC-KSVD2 and D-KSVD.

It is worth mentioning that our model in Eq. (6) can also be exploited for simultaneously learning the dictionary and the classifier. Therefore, we also explored this alternative for our model. For that, we used the matrix $[\mathbf{X}; \mathbf{H}] \in \mathbb{R}^{(m+C)\times N}$ as the training data, where ';' denotes the vertical concatenation of the matrices and $\mathbf{H} \in \mathbb{R}^{C\times N}$ is created by arranging the vectors $\mathbf{h}_i^c$ for the training samples. For such training data, the basis inferred by our model can be seen as $[\mathbf{\Phi}; \mathbf{W}] \in \mathbb{R}^{(m+C)\times|\mathcal{K}|}$. This approach of joint dictionary and classifier training is inspired by D-KSVD [7] and LC-KSVD [9], that exploit the K-SVD model [6] in a similar fashion. Since that model is for unsupervised dictionary learning, its extension towards supervised training by the joint learning procedure yielded improved classification performance for D-KSVD [7] and LC-KSVD [9], as compared to K-SVD [6]. However, the joint training procedure did not have

Fig. 8. Examples from UCF sports action dataset [5].

TABLE 6
Classification Rates on UCF Sports Action [5]

Method	Accuracy %	Method	Accuracy %
Qiu et al. [33]	83.6	LC-KSVD2 [9]	91.5
D-KSVD [7]	89.1	DLSI [12]	92.1
LC-KSVD1 [9]	89.6	SRC [8]	92.7
DL-COPAR [11]	90.7	FDDL [10]	93.6
Sadanand [40]	90.7	LDL [13]	95.0
Proposed	**95.1**		

5. The results of DL-COPAR [11] and FDDL [10] are taken directly from the literature because the optimized parameter values for these algorithms are not previously reported for this dataset. Our parameter optimization did not outperform the reported accuracies.

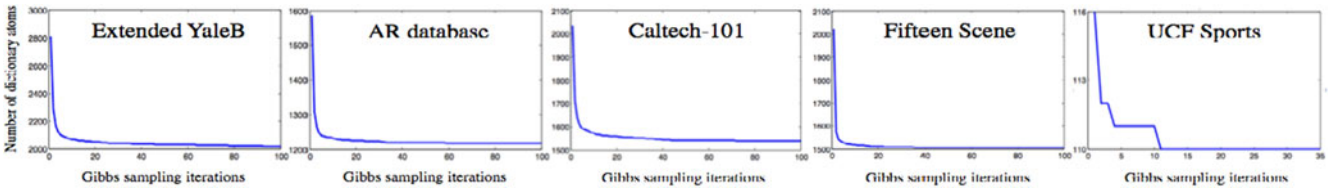


Fig. 9. Size of the inferred dictionary, i.e., $|\mathcal{K}|$, as a function of the Gibbs sampling iterations. The plots show first hundred iterations for each dataset.

a similar effect on our approach. In fact, in our experiments, the approach presented in Section 4 achieved an average percentage gain of 0.93, 0.61, 0.32 and 0.50 for AR database [1], Extended YaleB [2], Caltech-101 [3] and 15 Scene Category database [4], respectively over the joint learning procedure. This consistent performance gain is an expected phenomenon for our approach. As mentioned in Section 4.2, for computational purposes, we assume isotropic precision Gaussians over the basis vectors in our model. By jointly learning the classifier with the dictionary, its weights must also follow the same distributions, as followed by the dictionary atoms. This assumption is restrictive, which results in a slight degradation of the classification accuracy. The separate learning procedures for the dictionary and the classifier remove this restrictive assumption while keeping the inference process efficient.

7 Conclusion

We proposed a non-parametric Bayesian approach for learning discriminative dictionaries for sparse representation of data. The proposed approach employs a truncated Beta process to infer a discriminative dictionary and sets of Bernoulli distributions associating the dictionary atoms to the class labels of the training data. The said association is adaptively built during Bayesian inference and it signifies the selection probabilities of dictionary atoms in the expansion of class-specific data. The inference process also results in computing the correct size of the dictionary. For learning the discriminative dictionary, we presented a hierarchical Bayesian model and the corresponding inference equations for Gibbs sampling. The proposed model is also exploited in learning a linear classifier that finally classifies the sparse codes of a test instance that are learned using the inferred discriminative dictionary. The proposed approach is evaluated for classification using five different databases of human face, human action, scene category and object images. Comparisons with state-of-the-art discriminative sparse representation approaches show that the proposed Bayesian approach consistently outperforms these approaches and has computational efficiency close to the most efficient approach.

Whereas its effectiveness in terms of accuracy and computation is experimentally proven in this work, there are also other key advantages that make our Bayesian approach to discriminative sparse representation much more appealing than the existing optimization based approaches. First, the Bayesian framework allows us to learn an ensemble of discriminative dictionaries in the form of probability distributions instead of the point estimates that are learned by the optimization based approaches. Second, it provides a principled approach to estimate the required dictionary size and we can associate the dictionary atoms and the class labels in a physically meaningful manner. Third, the Bayesian framework makes our approach inherently an online technique. Furthermore, the Bayesian framework also provides an opportunity of using domain/class-specific prior knowledge in our approach in a principled manner. This can prove beneficial in many applications. For instance, while classifying the spectral signatures of minerals on pixel and sub-pixel level in remote-sensing hyperspectral images, the relative smoothness of spectral signatures [65] can be incorporated in the inferred discriminative bases. For this purpose, Gaussian Processes [66] can be used as a base measure for the Beta Process. Adapting the proposed approach for remote-sensing hyperspectral image classification is also our future research direction.

Acknowledgments

This research was supported by ARC Grant DP110102399.

References

[1] A. Martinez and R. Benavente, "The AR Face Database," *Comput. Vis. Center, Purdue Univ.*, West Lafayette, IN, USA, Tech. Rep. 24, Jun. 1998.

[2] A. Georghiades, P. Belhumeur, and D. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 643–660, Jun. 2001.

[3] 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," *Comput. Vis. Image Und.*, vol. 106, no. 1, pp. 59–70, 2007.

[4] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bag of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog.*, 2007, pp. 2169–2178.

[5] M. D. Rodriguez, J. Ahmed, and M. Shah, "Action MACH: A spatio-temporal maximum average correlation height filter for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2008, pp. 1–8.

[6] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 1, pp. 4311–4322, Nov. 2006.

[7] Q. Zhang and B. Li, "Discriminative K-SVD for dictionary learning in face recognition," in *Proc. Comput. Vis. Pattern Recog.*, 2010, pp. 2691–2698.

[8] J. Wright, M. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.

[9] Z. Jiang, Z. Lin, and L. S. Davis, "Label consistent K-SVD: Learning a discriminative dictionary for recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2651–2664, Nov. 2013.

[10] M. Yang, L. Zhang, X. Feng, and D. Zhang, "Sparse representation based Fisher discrimination dictionary learning for image classification," *Int. J. Comput. Vis.*, vol. 109, no. 3, pp. 209–232, Sep. 2014.

[11] D. Wang and S. Kong, "A classification-oriented dictionary learning model: Explicitly learning the particularity and commonality across categories," *Pattern Recog.*, vol. 47, no. 2, pp. 885–898, Feb. 2014.

[12] I. Ramirez, P. Sprechmann, and G. Sapiro, "Classification and clustering via dictionary learning with structured incoherence and shared features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2010, pp. 3501–3508.

[13] M. Yang, D. Dai, L. Shen, and L. V. Gool, "Latent dictionary learning for sparse representation based classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 4138–4145.
[14] Y. Sun, Q. Liu, J. Tang, and D. Tao, "Learning discriminative dictionary for group sparse representation," *IEEE Trans. Image Process.*, vol. 23, no. 9, pp. 3816–3828, Sep. 2014.
[15] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality constrained linear coding for image classification," in *Proc. Conf. Comput. Vis. Pattern Recog.*, 2010, pp. 3360–3367.
[16] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381 no. 6583, pp. 607–609, 1996.
[17] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by V1?" *Vis. Res.*, vol. 37, no. 23, pp. 3311–3325, 1997.
[18] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006.
[19] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Trans. Image Process.*, vol. 17, no. 1, pp. 53–69, Jan. 2008.
[20] N. Akhtar, F. Shafait, and A. Mian, "Sparse spatio-spectral representation for hyperspectral image super resolution," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 63–78.
[21] E. Candes, "Comressive Sampling," in *Proc. Int. Congress Math.*, 2006, pp. 1433–1452.
[22] D. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52 no. 4, pp. 1289–1306, Apr. 2006.
[23] J. Bobin, J. L. Starck, J. M. Fadili, Y. Moudden, and D. L. Donoho, "Morphological component analysis: An adaptive thresholding strategy," *IEEE Trans. Image Process.*, vol. 16, no. 11 pp. 2675–2681, Nov. 2006.
[24] M. Yang and L. Zhang, "Gabor feature based sparse representation for face recognition with Gabor occlusion dictionary," in *Proc. 11th Eur. Conf. Comput. Vis.*, 2010, pp. 448–461.
[25] M. Yang, D. Zhang, J. Yang, and D. Zhang, "Robust sparse coding for face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 625–632.
[26] M. Yang, L. Zhang, J. Yang, and D. Zhang, "Metaface learning for sparse representation based face recognition," in *Proc. IEEE 17th IEEE Int. Conf. Image Process.*, 2010, pp. 1601–1604.
[27] K. Huang and S. Aviyente, "Sparse representation for signal classification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 609–616.
[28] J. Mairal, J. Ponce, G. Sapiro, A. Zisserman, and F. R. Bach, "Supervised dictionary learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 1033–1040.
[29] J. Yang, K. Yu, and T. Huang, "Supervised translation-invariant sparse coding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2010, pp. 3517–3524.
[30] F. Rodriguez and G. Sapiro, "Sparse representations for image classification: Learning discriminative and reconstructive non-parametric dictionaries," Univ. Minnesota, Minneapolis, MN, Tech. Rep. /IMA Preprint, Dec. 2007.
[31] J. Mairal, F. Bach, and J. Ponce, "Task driven dictionary learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 791–804, Apr. 2012.
[32] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 1794–1801.
[33] Q. Qiu, Z. Jiang, and R. Chellappa, "Sparse dictionary-based representation and recognition of action attributes," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 707–714.
[34] G. Tanaya and R. K. Ward, "Learning sparse representations for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 8, pp. 1576–1588, Aug. 2012.
[35] H. Wang, Y. Chunfeng, H. Weiming, and S. Changyin, "Supervised class-specific dictionary learning for sparse modeling in action recognition," *Pattern Recog.*, vol. 45, no. 11, pp. 3902–3911, 2012.
[36] A. Castrodad and G. Sapiro, "Sparse modeling of human actions from motion imagery," *Int. J. Comput. Vis.*, vol. 100, no. 1, pp. 1–15, 2012.
[37] H. Zhang, A. Berg, M. Maire, and J. Malik, "SVM-KNN: Discriminative nearest neighbor classification for visual category recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2006, pp. 2126–2136.
[38] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2007, pp. 2169–2178.
[39] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," California Inst. Technol., Pasadena, CA, USA, Tech. Rep. 7694, 2007.
[40] S. Sadanand and J. J. Corso, "Action bank: A high-level representation of activity in video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2012, pp. 1234–1241.
[41] J. W. Cooley and J. W. Tukey, "An algorithm for machine calculation of complex Fourier series," *Math. Comput.*, vol. 19, pp. 297–301, 1965.
[42] S. Mallat, *A Wavelet tour of Signal Processing*, 2nd ed., San Diego, CA, USA: Academic, 1999.
[43] O. Bryt and M. Elad, "Compression of facial images using the K-SVD algorithm," *J. Vis. Commun. Image Representation*, vol. 19, no. 4, pp. 270–282, 2008.
[44] M. Zhou, H. Chen, L. Ren, G. Sapiro, L. Carin, and J. W. Paisley, "Non-parametric Bayesian dictionary learning for sparse image representations," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 2295–2303.
[45] D. S. Pham and S. Venkatesh, "Joint learning and dictionary construction for pattern recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2008, pp. 1–8.
[46] K. Engan, S. O. Aase, and J. H. Husoy, "Method of optimal directions for frame design," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 1999, pp. 2443–2446.
[47] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Discriminative learned dictionaries for local image analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2008, pp. 1–8.
[48] P. Sprechmann and G. Sapiro, "Dictionary learning and sparse coding for unsupervised clustering," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2010, pp. 2042–2045.
[49] Y. N. Wu, Z. Si, H. Gong, and S. C. Zhu, "Learning active basis model for object detection and recognition," *Int. J. Comput. Vis.*, vol. 90, no. 2, pp. 198–235, 2010.
[50] X. C. Lian, Z. Li, B. L. Lu, and L. Zhang, "Max-margin dictionary learning for multiclass image categorization," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 157–170.
[51] Z. Jiang, "Submodular dictionary learning for sparse coding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2012, pp. 3418–3425.
[52] J. A. Tropp, A. C. Gilbert, and M. J. Strauss, "Algorithms for simultaneous sparse approximation: Part I: Greedy pursuit," *Signal Process.*, vol. 86, no. 3, pp. 572–588, 2006.
[53] W. Deng, J. Hu, and J. Guo, "Extended SRC: Undersampled face recognition via intraclass variant dictionary," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1864–1870, Sep. 2012.
[54] N. Zhou and J. P. Fan, "Learning inter-related visual dictionary for object recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2012, pp. 3490–3497.
[55] L. Shen, S. Wang, G. Sun, S. Jiang, and Q. Huang, "Multi-level discriminative dictionary learning towards hierarchical visual categorization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2013, pp. 383–390.
[56] J. Paisley and L. Carin, "Nonparametric factor analysis with beta process prior," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 777–784.
[57] M. Elad, *Sparse and Redundant Representation: From Theory to Applications in Signal and Image processing*. New York, NY, USA: Springer, 2010.
[58] M. Beal, "Variational algorithms for approximate Bayesian inference," Doctoral dissertation, Gatsby Comput. Neurosci. Unit, Univ. College London, London, U.K., 2003.
[59] C.M. Bishop, *Pattern Recognition and Machine Learning* (Information Science and Statistics). New York, NY, USA: Springer-Verlag, 2006.
[60] T. T. Cai and Lei Wang, "Orthogonal matching pursuit for sparse signal recovery with noise," *IEEE Trans. Inf. Theory*, vol. 57, no. 7 pp. 4680–4688, Jul. 2011.
[61] G. Golub, P. Hansen, and D. O'leary, "Tikhonov regularization and total least squares," *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 1, pp. 185–194, 1999.
[62] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *The J. Mach. Learn. Res.*, vol. 11, pp. 19–60, 2010.
[63] R. Tibshirani, "Regression shrinkage and selection via Lasso," *J. Roy. Statist. Soc.*, vol. 58, pp. 267–288, 1996.

[64] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
[65] N. Akhtar, F. Shafait, and A. Mian, "Futuristic greedy approach to sparse unmixing of hypersperctral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 4, pp. 2157–2174, Apr. 2015.
[66] M. Seeger, "Gaussian processes for machine learning," *Int. J. Neural Syst.*, vol. 14, no. 2, pp. 69–104, 2004.
[67] A. Damianou, C. Ek, M. Titsias, and N. Lawrence, "Manifold relevance determination," in *Proc. Int. Conf. Mach. Learn.*, 2012, pp. 145–152.
[68] R. Andrade-Pacheco, J. Hensman, M. Zwiessele, and N. Lawrence, "Hybrid discriminative-generative approach with Gaussian processes," in *Proc. Int. Conf. Artif. Intell. Stat.*, 2014, pp. 47–56.
[69] C. Lu and X. Tang, "Learning the face prior for Bayesian face recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 119–134.
[70] A. Klami, S. Virtanen, E. Leppaaho, and S. Kaski, "Group factor analysis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 9, pp. 2136–2147, Sep. 2015.
[71] M. Elad, R. Rubinstein, and M. Zibulevsky, "Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit," Technion, Haifa, Israel, Tech. Rep. CS-2008-08, 2008.
[72] N. Akhtar, F. Shafait, and A. Mian, "Bayesian sparse representation for hyperspectral image super resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 3631–3640.

Naveed Akhtar received the BE degree with distinction in avionics from the College of Aeronautical Engineering, National University of Sciences and Technology (NUST), Pakistan, in 2007 and the MSc degree with distinction in autonomous systems from Hochschule Bonn-Rhein-Sieg (HBRS), Sankt Augustin, Germany, in 2012. He is currently working toward the PhD degree at The University of Western Australia (UWA) under the supervision of Dr. Faisal Shafait and Prof. Ajmal Mian. He received the Scholarship for International Research Fees, University International Stipend (UIS) and UIS Safety-net-top-up scholarships at UWA. He has served as a research assistant at the Research Institute for Microwaves and Millimeter-waves Studies, NUST, Pakistan, from 2007 to 2009 and as a research associate at the Department of Computer Science at HBRS, Germany in 2012. His current research is focused on sparse representation based image analysis.

Faisal Shafait is the director of the TUKL-NUST Research and Development Center and an associate professor in the School of Electrical Engineering and Computer Science at the National University of Sciences and Technology, Pakistan. He has previously held positions at The University of Western Australia, Perth, Australia; German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, Germany; and Google Inc., Mountain View, CA. His research interests include machine learning and computer vision with a special emphasis on applications in document image analysis and recognition. He has coauthored more than 100 publications in international peer-reviewed conferences and journals in this area. He is an editorial board member of the *International Journal on Document Analysis and Recognition* (IJDAR), and a program committee member of leading document analysis conferences including ICDAR, DAS, and ICFHR. Besides, he is serving on the leadership board of IAPRs Technical Committee on Computational Forensics (TC-6) and is currently the president of Pakistani Pattern Recognition Society (PPRS).

Ajmal Mian recieved the PhD degree from The University of Western Australia in 2006 with distinction and received the Australasian Distinguished Doctoral Dissertation Award from the Computing Research and Education Association of Australasia. He received two prestigious nationally competitive fellowships namely the Australian Postdoctoral Fellowship in 2008 and the Australian Research Fellowship in 2011. He received the UWA Outstanding Young Investigator Award in 2011, the West Australian Early Career Scientist of the Year award in 2012 and the Vice-Chancellors Mid-Career Research Award in 2014. He is currently with the School of Computer Science and Software Engineering, The University of Western Australia. His research interests include computer vision, machine learning, action recognition, 3D shape analysis, 3D facial morphometrics, hyperspectral image analysis, and biometrics.