Online Semi-Supervised Discriminative Dictionary Learning for Sparse Representation

Guangxiao Zhang, Zhuolin Jiang, Larry S. Davis

University of Maryland, College Park, MD, 20742
{gxzhang,zhuolin,lsd}@umiacs.umd.edu

Abstract. We present an online semi-supervised dictionary learning algorithm for classification tasks. Specifically, we integrate the reconstruction error of labeled and unlabeled data, the discriminative sparse-code error, and the classification error into an objective function for online dictionary learning, which enhances the dictionary's representative and discriminative power. In addition, we propose a probabilistic model over the sparse codes of input signals, which allows us to expand the labeled set. As a consequence, the dictionary and the classifier learned from the enlarged labeled set yield lower generalization error on unseen data. Our approach learns a single dictionary and a predictive linear classifier jointly. Experimental results demonstrate the effectiveness of our approach in face and object category recognition applications.

1 Introduction

Learning dictionaries for sparse coding has recently led to state-of-art performances in many computer vision tasks [1–4]. The performance of image classification, in particular, has been further improved by learning discriminative dictionaries for sparse coding. Consider an input signal $\mathbf{x} \in \mathbb{R}^n$. It can be represented as a linear combination of a few atoms from a dictionary $D = \{d_1 ... d_K\} \in \mathbb{R}^{n \times K}$, *i.e.*, $\mathbf{x} = D\mathbf{z}$. The vector $\mathbf{z} \in \mathbb{R}^K$ is called the sparse code of $\mathbf{x}$ with respect to D. The resulting $\mathbf{z}$ is discriminative when D has discriminative power.

Some discriminative dictionary learning approaches have been proposed recently for classification [5–10]. However, most of them are based on iterative batch procedures [11, 5, 9, 12], which access the whole dataset at each iteration and optimize over all data. For large scale datasets, this becomes a big challenge due to memory requirements and computational complexity. Although some online dictionary learning algorithms [13, 14] have been proposed for image restoration purpose recently, incorporating the discriminative information in online dictionary learning for discriminative tasks has not been fully explored.

Learning a discriminative dictionary usually requires sufficient labeled training data, which is expensive and difficult to obtain. Insufficient labeled training data yields a dictionary with potentially bad generalization power. By exploiting the information provided by the vast quantity of inexpensive unlabeled data, we aim to develop an online algorithm to learn a dictionary which is more representative and discriminative than a dictionary trained using only a limited number

of labeled samples in a batch procedure [15]. More importantly, we show how to identify those 'important' unlabeled data points, such as the points located near the decision boundary in sparse feature space, or points representing items very different from those we have seen before, and manually label those points in an active learning setting [16].

In this paper, we propose an online, semi-supervised dictionary learning algorithm that integrates dictionary learning and classifier training. We introduce a novel objective function which includes terms representing the reconstruction error of both labeled and unlabeled data, the discriminative sparse-code error, and the classification error. Compared to supervised dictionary learning approaches, our approach improves the representation power of the dictionary by exploiting the unlabeled data. It takes the reconstruction error of the unlabeled data to account in the objective function, and treats the unlabeled points with high confidence in label prediction as 'labeled' points. In addition, it identifies the unlabeled points with the most uncertainty in label prediction for manually labeling. Our approach learns a single over-complete dictionary and an optimal linear classifier jointly. Our main contributions are:

- We propose an online framework of discriminative dictionary learning for classification tasks, which is suitable for large data sets or dynamic training.
- The dictionary learns from labeled samples for discrimination as well as a large number of unlabeled samples. Learning from unlabeled data further increases its representative power.
- Our approach actively identifies the hard classified samples to be manually labeled and selects the easily classified samples as labeled data, using a probabilistic model of the sparse code of an input signal. In this way, unlabeled data also contribute to learning discriminative dictionaries with minimal human supervision.

1.1 Related Work

Discriminative dictionary learning for sparse coding has received a lot of attention recently. Some approaches treat dictionary learning and classifier training as two separate processes as in [18, 8, 19–21]. The sparse codes associated with the dictionary trained in the first step are later fed into classifiers such as SVMs as feature attributes. For those methods, the discrimination power comes from either the sophisticated classifiers in the later stage, or learning multiple category-specific dictionaries [20, 22, 8], which might not be suitable when there are a large number of classes. Some other approaches incorporate category label information into the dictionary training process [6, 8, 7, 5, 12, 23, 9]. The dictionaries are learned by optimizing a unified objective function combining reconstructive and discriminative terms. In general, the optimization processes are iterative batch procedures: [6] alternates between dictionary construction and classifier design, and [8, 7, 9] alternate between supervised sparse coding and dictionary update. However these existing approaches cannot handle very large training sets.

To address these issues, several incremental learning or online learning algorithms [24, 13, 14, 17] have been proposed recently. [24] utilizes first-order stochas-

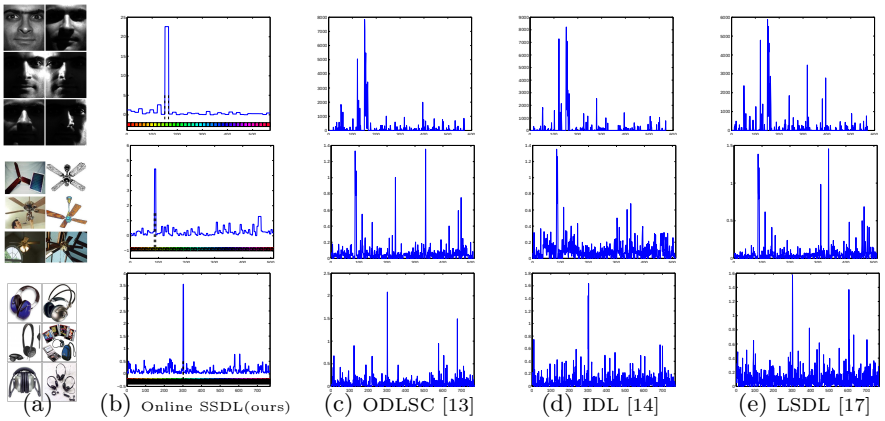


Fig. 1. Examples of sparse codes using dictionaries learned by different approaches on the Extended YaleB, Caltech101, and Caltech256 datasets. Each waveform indicates a sum of absolute sparse codes for different testing images from the same class. The 1st, 2nd, and 3nd row correspond to class 11 (28 testing frames) in Extended YaleB, class 18 (61 testing frames) in Caltech101, and class 101 (123 testing frames) in Caltech256 respectively. (a) are sample images from these classes. Each color from the color bar in (b) represents one class for a subset of dictionary items. The black dashed lines indicate that the curves are highly peaked in one class. (c) Online Dictionary Learning for sparse coding (ODLSC) [13], (d) Incremental Dictionary Learning (IDL) [14], (e) Large Scale Dictionary Learning (LSDL) [17]. The figure is best viewed in color and 600% zoom in.

tic gradient descent with projections on the constraint set for dictionary learning. [13] efficiently minimizes a quadratic surrogate function of the empirical cost over the set of constraints at each step. [14] utilizes locality constraints to project each descriptor into its local-coordinate system so that the objective function can be optimized analytically. The dictionary is then updated incrementally in a gradient descent fashion. Unfortunately, all of these techniques focus on minimizing the reconstruction error, which is good for reconstruction tasks but not for discrimination tasks such as classification. One of the major difficulties here is that we cannot afford to obtain sufficient labeled training samples. Therefore, learning a discriminative dictionary in an online fashion with minimal human supervision becomes an interesting problem.

2 Sparse Representation and Dictionary Learning

Consider a set of N input signals $X = [\mathbf{x}_1 ... \mathbf{x}_N] \in \mathbb{R}^{n \times N}$. Given a dictionary D of size K, the sparse representations $Z = [\mathbf{z}_1 ... \mathbf{z}_N] \in \mathbb{R}^{K \times N}$ for X can be obtained by:

$$Z = \arg\min_{Z} ||X - DZ||_2^2, \quad s.t. \forall i, \quad \|\mathbf{z}_i\|_0 \leq \varepsilon \tag{1}$$

where $\|\mathbf{z}_i\|_0 \leq \varepsilon$ is a sparsity constraint. The performance of sparse representation highly depends on D. Traditional dictionary learning for sparse coding is achieved by minimizing the empirical reconstruction error:

$$< D, Z >= \arg\min_{D,Z} ||X - DZ||_2^2, \quad s.t. \forall i, \quad \|\mathbf{z}_i\|_0 \leq \varepsilon \tag{2}$$

where $D = [d_1 ... d_K] \in \mathbb{R}^{n \times K}$ is the learned dictionary. In general, the number of training samples is larger than the size of D ($N \gg K$), and $\mathbf{x}_i$ only uses

a few dictionary items out of total K for its reconstruction under the sparsity constraint. K-SVD [11] is an efficient algorithms to solve (2); it alternates between dictionary construction and sparse coding while keeping the other fixed until convergence is achieved. However, K-SVD only focuses on minimizing the reconstruction error. In addition, for a large training set, batch optimization techniques may be impractical.

There are two classes of algorithms that solve the optimization problems in (2) even with large training sets. One is classical projected first-order stochastic gradient descent [24, 17]. With an appropriate selection of a learning rate, the dictionary is sequentially updated by:

$$D_t = \Pi_c \left[D_{t-1} - \frac{\rho}{t} \nabla_D l(\mathbf{x}_t, D_{t-1}) \right], \tag{3}$$

Another class of algorithms does not require explicit learning rate tuning; instead, they exploit the structure of the problem based on the second-order stochastic approximation [13]. The new dictionary D_t is computed by minimizing the following cost function over the convex set $\mathcal{C} = \{D \in \mathbb{R}^{n \times K}, s.t. \forall j = 1, ..., K, \mathbf{d_j}^T \mathbf{d_j} \leq 1\}$

$$\begin{aligned} D_t &= \arg\min_{D \in \mathcal{C}} \frac{1}{t} \sum_{i=1}^{t} \frac{1}{2} ||\mathbf{x}_i - D\mathbf{z}_i||_2^2 + \lambda ||\mathbf{z}_i||_0 \\ &= \arg\min_{D \in \mathcal{C}} \frac{1}{t} \left(\frac{1}{2} Tr(D^T D \sum_{i=1}^{t} \mathbf{z}_i \mathbf{z}_i^T) - Tr(D^T \sum_{i=1}^{t} \mathbf{x}_i \mathbf{z}_i^T) \right) \\ &= \arg\min_{D \in \mathcal{C}} \frac{1}{t} \left(\frac{1}{2} Tr(D^T D A_t) - Tr(D^T B_t) \right) \end{aligned} \tag{4}$$

With some simple algebra, it is easy to show that algorithm 1 (below) gives the solution to the convex optimization problem with respect to the j-th column while keeping the others fixed. Here matrices $A = \sum_{i=1}^{t} \mathbf{z}_i \mathbf{z}_i^T$ and $B = \sum_{i=1}^{t} \mathbf{x}_i \mathbf{z}_i^T$ propagate information from the past. This efficient online algorithm outperforms its batch counterpart in natural image experiments [13].

Unfortunately, these online algorithms are not explicitly designed for classification tasks. To further enhance the discrimination power of the dictionary, we propose an online semi-supervised dictionary learning algorithm which will be discussed in the next section.

3 Online Semi-Supervised Dictionary Learning

3.1 Problem Statement

To improve the discriminative power of a dictionary, we follow [9] and combine two discriminative term- the 'discriminative sparse-code error' and the 'classification error'- with the reconstruction error term to form an objective function for dictionary learning. In this way, the dictionary and the classifier are learned jointly. To take advantage of the large number of inexpensive unlabeled data, the reconstructive term consists of two parts: one from labeled training data and

the other from unlabeled training data. To be concrete, the objective function for our dictionary learning is defined as:

$$< D, G, W, Z >= \arg \min_{D,G,W,Z} \alpha \|X^u - DZ^u\|_2^2 + \beta ||X^l - DZ^l||_2^2 + \gamma ||Q - GZ^l||_2^2 + ||H - WZ^l||_2^2 \quad s.t. \forall i, \quad \|\mathbf{z}_i\|_0 \leq \varepsilon \tag{5}$$

The superscripts u and l specify whether the sample is from the unlabeled set or the labeled set. The first two terms are the reconstruction errors, while the last two terms are the discrimination errors. Parameters α, β, γ control the relative weight of these terms. In the $\|Q - GZ^l\|_2^2$ term, $Q = [\mathbf{q}_1^l, ..., \mathbf{q}_N^l]$ is a label-consistency matrix of size $K \times N^l$, with N^l being the number of the labeled training samples. Each dictionary item in our approach is attached to a specific class label. Each column $\mathbf{q}_j \in \mathbb{R}^K$ is a discriminative sparse code corresponding to $\mathbf{x}_j$. $\mathbf{q}_j(i) = 1$ only when dictionary item d_i and the training point $\mathbf{x}_j$ share the same class label; otherwise $\mathbf{q}_j(i) = 0, i = 1...K$. $G \in \mathbb{R}^{K \times K}$ is a linear transformation matrix that projects the sparse codes $\mathbf{z}$ to a discriminative sparse feature space $\mathbb{R}^K$.

The term $||H - WZ^l||_2^2$ measures the classification error. Suppose we have m classes in the classification task. A linear predictive classifier $f(z; W) = Wz$ is employed, where $W \in \mathbb{R}^{m \times K}$ is the classifier parameters. A column h_i of $H = [h_1, ..., h_N] \in \mathbb{R}^{m \times N}$ is the label vector for $\mathbf{x}_i$, where non-zero position indicates the category label of $\mathbf{x}_i$. The classifier W is learned jointly with the transformation matrix G and the dictionary D by solving (5).

A major consideration in choosing a suitable optimization method is that since our problem is to be solved in an online learning setting, we cannot separate the labeled set and the unlabeled set in advance. Supervised learning and the unsupervised learning interleave as new data comes in; thus we require an adaptive strategy.

3.2 Optimization

Our algorithm alternates between sparse coding and dictionary updating as the input signals arrive sequentially. We rewrite the objective function in (5) as:

$$\min_{D,G,W,Z} \sum_{i=1}^{N_u} \left\{ \alpha ||\mathbf{x}_i^u - D\mathbf{z}_i^u||_2^2 \right\} + \sum_{i=1}^{N_l} \left\{ \beta ||\mathbf{x}_i^l - D\mathbf{z}_i^l||_2^2 + \gamma ||\mathbf{q}_i - G\mathbf{z}_i^l||_2^2 + ||\mathbf{h}_i - W\mathbf{z}_i^l||_2^2 \right\}, \quad s.t. \forall i, ||\mathbf{z}_i||_0 \leq \varepsilon \tag{6}$$

where N_u and N_l are the number of unlabeled and labeled training samples respectively.

Initialization We assume that, initially, we have a small labeled data set spanning all classes. To meet the requirement that each dictionary item is associated with a class label, we learn multiple class-specific dictionaries separately using K-SVD and then combine their dictionary items together. For simplicity we allocate equal number of dictionary items to each class, and the class labels

attached to the dictionary items remain the same no matter how we update them throughout the training process. The initialization process is completely supervised.

Algorithm 1: Dictionary Update

Input: current dictionary D_{t-1};
$A_t = \sum_{i=1}^{t} \mathbf{z}_i \mathbf{z}_i^T = [\mathbf{a}_1 ... \mathbf{a}_t]$,
$B_t = \sum_{i=1}^{t} \mathbf{x}_i \mathbf{z}_i^T = [\mathbf{b}_1 ... \mathbf{b}_t]$;
Output: updated dictionary D_t.
repeat
  for $j = 1, 2, ..., K$ **do**
    Update the j-th column
      $\mathbf{u}_j \leftarrow \frac{1}{A_{j,j}}(\mathbf{b}_j - D\mathbf{a}_j) + \mathbf{d}_j$.
      $\mathbf{d}_j \leftarrow \frac{1}{\max ||\mathbf{u}_j||_2, 1} \mathbf{u}_j$.
  end for
until convergence
Return

Online sparse coding At time t, given that the dictionary D, the label-consistency transformation matrix G, and the label matrix H are all fixed, the task is to find the sparse code $\mathbf{z}_t$ for the signal $\mathbf{x}_t$.

- For unlabeled $\mathbf{x}_t$, the sparse coding problem simply takes this standard form: $\mathbf{z}_t = \arg\min_{\mathbf{z} \in \mathbb{R}^K} ||\mathbf{x}_t - D\mathbf{z}||_2^2, s.t. ||\mathbf{z}||_0 \leq \varepsilon$. The orthogonal matching pursuit (OMP) algorithm is adopted here for its efficiency.
- For labeled $\mathbf{x}_t$, first construct the label-consistency vector $\mathbf{q}_t$ and label vector $\mathbf{h}_t$. The sparse coding problem becomes:

$$\mathbf{z}_t = \arg\min_{\mathbf{z} \in \mathbb{R}^K} \beta ||\mathbf{x}_t - D\mathbf{z}||_2^2 + \gamma ||\mathbf{q}_t - G\mathbf{z}||_2^2 + ||\mathbf{h}_t - W\mathbf{z}||_2^2, s.t. ||\mathbf{z}||_0 \leq \varepsilon, \quad (7)$$

which can be rewritten as,

$$\mathbf{z}_t = \arg\min_{\mathbf{z} \in \mathbb{R}^K} \left\| \begin{pmatrix} \sqrt{\beta}\mathbf{x}_t \\ \sqrt{\gamma}\mathbf{q}_t \\ \mathbf{h}_t \end{pmatrix} - \begin{pmatrix} \sqrt{\beta}D \\ \sqrt{\gamma}G \\ W \end{pmatrix} \mathbf{z} \right\|_2^2 = \arg\min_{\mathbf{z} \in \mathbb{R}^K} ||\tilde{\mathbf{x}}_t - \tilde{D}\mathbf{z}||_2^2, \quad (8)$$

With definition of augmented input signal $\tilde{\mathbf{x}}_t = [\sqrt{\beta}\mathbf{x}_t^T, \sqrt{\gamma}\mathbf{q}_t^T, \mathbf{h}_t^T]^t$ and augmented dictionary $\tilde{D} = [\sqrt{\beta}D^T, \sqrt{\gamma}G^T, W^T]^T$, the sparse code of the labeled $\mathbf{z}_t$ can be solved by OMP as for the unlabeled case.

Dictionary update Once the sparse code for $\mathbf{x}_i$ is obtained, we perform the dictionary update motivated by [13]. First, the coefficient matrix $B_t = \sum_{i=1}^{t} \mathbf{x}_i \mathbf{z}_i^T$, which carry all the information from the past sparse codes $\mathbf{z}_1, ..., \mathbf{z}_t$, is augmented to $\tilde{B}$ as the $\mathbf{x}_i$'s are augmented to $\tilde{\mathbf{x}}_i = [\sqrt{\beta}\mathbf{x}_i^T, \sqrt{\gamma}\mathbf{q}_i^T, \mathbf{h}_i^T]^T$. Note that $\tilde{B}$ is iteratively updated by both labeled data and unlabeled data. In the latter case, only the first n rows which correspond to $\mathbf{x}_i$'s are updated. In essence, the first n rows in $\tilde{B}$ record the past information of all training data, and the remaining $K + m$ rows (the dimension of $\mathbf{q}_i$ plus $\mathbf{h}_i$) reflect only the history of the labeled data. Second, the dictionary is updated either by itself or with G

and W jointly in the augmented $\tilde{D}$, depending on whether the signal is labeled or not in that iteration. Given sparse codes $\mathbf{z}_i, i = 1...t$, the updated dictionary using algorithm 1 is the solution to (4) stated in section 2.

Note that algorithm 1 can also be applied to solve (4) with the augmented dictionary simply by replacing $\mathbf{x}_i$ with the augmented $\tilde{\mathbf{x}}_i = [\sqrt{\beta}\mathbf{x}_i^T, \sqrt{\gamma}\mathbf{q}_i^T, \mathbf{h}_i^T]^T$.

3.3 Learning From Unlabeled Data

So far we have discussed our online dictionary learning strategy with a mixture of labeled and unlabeled training samples. In practice, it still remains unclear how to choose which input data to label. After labeling the first few samples for the initial dictionary learning, we wish to keep the manual labeling effort minimum without sacrificing discriminative capability. In this section we propose a selection criterion based on a probabilistic model from the signal's sparse code.

Consider the sparse representation $\mathbf{z} = [z_1...z_K]^T$ of an input signal $\mathbf{x}$. Since once a dictionary element has its class determined, that can never change, the sparse coefficients z_j associated with item d_j can be used to compute the probability of signal $\mathbf{x}$ being in the same class as dictionary item d_j. If we sum up the absolute sparse codes associated with dictionary items from the same class and normalize them, we obtain the class probability distribution of the signal. Concretely, suppose we have an m-class classification problem, where each class is represented by k dictionary items, $k \times m = K$. The class probability of an input signal $\mathbf{x}$ with $\mathbf{z} = [z_1...z_K]^T$ being in class l, given D, is computed as:

$$p_l(\mathbf{x}) = Pr(\mathcal{L}(\mathbf{x}) = l|D) = \frac{\sum_{j:\mathcal{L}(d_j)=l} |z_j|}{\sum_j |z_j|}, \tag{9}$$

where $\mathcal{L}$ maps a data point or a dictionary item to a specific class label $l \in \{1...m\}$. The class probability distribution $P(\mathbf{x})$ for signal $\mathbf{x}$ is calculated by $P(\mathbf{x}) = [p_1(\mathbf{x})...p_m(\mathbf{x})]^T$.

The probability distribution informs us how well the dictionary discriminates the input signal. To quantify the confidence level of the discriminability of an input signal, we compute the entropy of its sparse code:

$$ent(\mathbf{x}) = -\sum_{l=1}^{m} p_l(\mathbf{x}) \log p_l(\mathbf{x}). \tag{10}$$

Intuitively if the dictionary is highly discriminative to an input signal, we expect the large values of the sparse code to concentrate at certain dictionary items, and thus the class distribution should be peaked at the most likely class. Quantitatively, we set two thresholds on the entropy of the probability distribution. Any entropy value smaller than a lower bound indicates a 'good' input signal with respect to the current dictionary, and we are fairly confident about our maximum likelihood class label prediction of this signal. Such points can thus be automatically added to the labeled set for dictionary learning with no human cost.

An entropy value higher than an upper bound tells us one of two things: it could be a difficult or uncertain input signal, or the current dictionary cannot represent Here it well. These points are critical to the dictionary learning because this highly uncertain point might be located near the decision boundary in the

feature space, or might be new data unlike any we have seen before. In both situations, manual labeling will have its greatest impact.

Parameter Selection The values of parameter ϕ_{low} and ϕ_{high} are chosen empirically. Here we use the sparse codes of the training data using the initial dictionary to approximate the class distributions of the training data, and then generate a distribution of the entropy values as a basis to determine the values of the thresholds. ϕ_{high} can be roughly estimated according to the budget of the manual labeling, while the best ϕ_{low} can be determined by five-fold cross validation on the training set. α,β,and γ are also determined via cross validation.

To summarize the discussions above, we propose the following semi-supervised learning strategy. The initial dictionary is learned under full supervision. As the unlabeled training data sequentially arrives, we compute the probability distribution of the sparse codes given the current dictionary, and evaluate the confidence level of the data. If the entropy value is lower than the lower bound, then we automatically label the point as the dominating class, and treat it as labeled data. If, in rare cases, the entropy value exceeds our upper threshold the user will be requested to label it. For those falling in between, we leave them as unlabeled data.

Algorithm 2 presents the pseudocode of our approach. The normalization step at the end of the dictionary update for the labeled data completes the iteration. Note that the columns of D, G and W are L_2-normalized in $\tilde{D}$ jointly, *i.e.*, $\forall j, \| \left[d_j^T, g_j^T, w_j^T\right]^T \|_2 = 1$. The desired dictionary $\hat{D}$, the transformation matrix $\hat{G}$, and the classifier $\hat{W}$ can be computed as [5]:

$$\hat{D} = \left[\frac{d_1}{||d_1||_2} \cdots \frac{d_K}{||d_K||_2}\right]; \hat{G} = \left[\frac{g_1}{||d_1||_2} \cdots \frac{g_K}{||d_K||_2}\right]; \hat{W} = \left[\frac{w_1}{||d_1||_2} \cdots \frac{w_K}{||d_K||_2}\right]; \quad (11)$$

3.4 Classification Approach

Once we obtain the discriminative $\hat{D}$, $\hat{G}$ and $\hat{W}$ from Algorithm 2, we need to recompute the sparse codes Z_l of the labeled data X_l to re-estimate $\hat{W}$, which includes the original labeled data, the automatically labeled data, and the manually labeled data. Given Z_l, the classifier $\hat{W}$ is estimated by using the multivariate ridge regression model with quadratic loss and L_2 norm regularization:

$$\arg\min_W \|H - WZ^l\|_2^2 + \lambda\|W\|_2^2, \quad (12)$$

which yields the analytic solution: $\hat{W} = HZ^T(ZZ^T + \lambda I)^{-1}$. When a testing point $\mathbf{x}^{test}$ comes in, we first compute its sparse code $\mathbf{z}^{test}$, and then compute $\hat{W}\mathbf{z}^{test}$. The label for $\mathbf{x}_j$ is assigned by the position corresponding to the largest value in the label vector: $\chi = \hat{W}\mathbf{z}^{test}$, where $\chi \in \mathbb{R}^m$.

4 Experiments

We evaluate our approach on three popular datasets: Extended YaleB database [25], Caltech101 [26], and Caltech256 [27]. We compare our results with two competing supervised dictionary learning algorithms: D-KSVD [8],LC-KSVD [9], as well

Algorithm 2: Online Semi-Supervised Dictionary Learning (Online SSDL)

Input: input signals $X = \{\mathbf{x}_1 ... \mathbf{x}_N\}$ and their labels, if any; regularization constant α, β and γ; lower bound ϕ_{low} and upper bound ϕ_{high}
Output: D, G, and W.
Initialization: Compute D_0, G_0, and W_0 via LC-KSVD
$A_0 \leftarrow 0; \tilde{B}_0 \leftarrow 0$
for $t = 1, 2, ..., N$ **do**
Draw $\mathbf{x}_t$ from the sequence;
Sparse coding: compute sparse code $\mathbf{z}_t$ using (1);
if $\mathbf{x}_t$ is unlabeled,
Compute the entropy $ent(\mathbf{x}_t)$ using (10);
if $ent(\mathbf{x}_t) \leq \phi_{high}$ and $ent(\mathbf{x}) \geq \phi_{low}$;
% dictionary update with unlabeled data
$A_t \leftarrow A_{t-1} + \alpha \mathbf{z}_t \mathbf{z}_t^T$;
$B_t \leftarrow \tilde{B}_{t-1}(1:n,:); B_t \leftarrow B_t + \alpha \mathbf{x}_t \mathbf{z}_t^T$;
Dictionary update by unlabeled data:
update D_t using algorithm 1 with D_{t-1}, A_t, and B_t;
continue;
elseif $ent(\mathbf{x}_t) < \phi_{low}$
% automatical labeling on the confident point
$\mathcal{L}(\mathbf{x_t}) = \arg\max_j p_j(\mathbf{x})$;
else $ent(\mathbf{x}_t) > \phi_{high}$
% manual labeling on the difficult point
$\mathcal{L}(\mathbf{x_t}) = l$;
endif
endif
% dictionary update with labeled data
Construct $\tilde{\mathbf{x}}_t = [\sqrt{\beta}\mathbf{x}_t^T; \sqrt{\gamma}\mathbf{q}_t^T; \mathbf{h}_t^T]^T$, and $\tilde{D}_{t-1} = [\sqrt{\beta}D_{t-1}^T; \sqrt{\gamma}G_{t-1}^T; W_{t-1}^T]^T$;
$A_t \leftarrow A_{t-1} + \mathbf{z}_t \mathbf{z}_t^T; \quad \tilde{B}_t \leftarrow \tilde{B}_{t-1} + \tilde{\mathbf{x}}_t \mathbf{z}_t^T$;
Dictionary update by labeled data:
update $\tilde{D}_t$ using algorithm 1 with $\tilde{D}_{t-1}$, A_t, and $\tilde{B}_t$;
obtain D, G and W from $\tilde{D}_t$ and normalize them by (11).
end for
Return D, G, and W.

as three online dictionary learning algorithms including Online Dictionary Learning for Sparse Coding (ODLSC) [13], Incremental Dictionary Learning (IDL) [14] and Large Scale Dictionary Learning (LSDL) [17], and some other benchmark algorithms such as K-SVD [11].

Since the number of labeled samples varies with our selection of ϕ_{low} and ϕ_{high} and the classification accuracy depends on the number of labeled training samples, it is tricky to do a fair comparison with other methods unless we fix our settings. To address this issue, we conducted the experiments in two folds: (1) Split the training set into labeled set and unlabeled set. We want to demonstrate the effect of the number of labeled samples on our performance in comparison with others. While our method takes advantage of both sets due to our learning strategy, the competing methods can only take the labeled set for training since the unlabeled samples are useless to them. (2) To compare our best recognition rate with the state-of-the-arts, we assumed all the training samples are labeled. We'd like to point out two facts: (a) our method adopts a simple classifier jointly learned with the dictionary, whereas other methods take advantage of sophisticated classifiers such as SVM; (2) although the advantage is not too obvious in terms of recognition rate in case of which all the training samples are labeled, the benefit of our method can be signified when the labeled samples are few,

Table 1. Recognition results using random face features on the Extended YaleB. We obtained the accuracies of LSDL, OSCDL, and IDL by running the codes, while the accuracies of the other methods are copied from the references.

Method	K-SVD [11]	D-KSVD [5]	SRC [3]	LLC [14]	LC-KSVD [9]
Acc.	93.1	94.1	80.5	82.2	94.5
Method	LSDL [17]	ODLSC [13]	IDL [14]	**Online SSDL**	
Acc.	90.5	91.4	89.6	**94.7**	

which is demonstrated at the starting points of all curves (see Fig. 2(a), 3(a), and 3(b)).

4.1 Extended YaleB Database

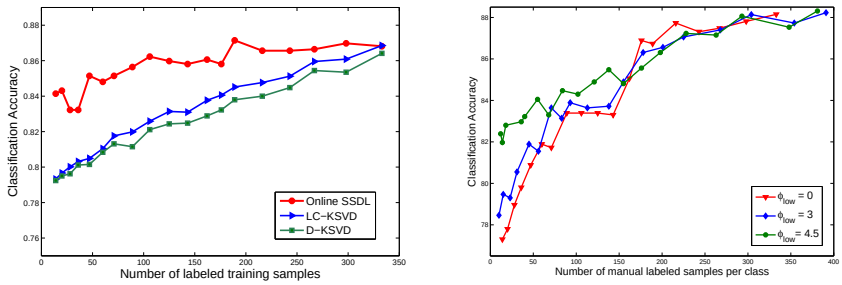
The extended YaleB database [25] contains $2,414$ images of 38 human frontal faces under about 64 illumination conditions and expressions. The images were chopped to 192×168 pixels. Each face was projected to a 504-dimensional random space by multiplying a random matrix introduced in [3, 5]. The entries of the matrix follow a zero-mean Gaussian distribution. We randomly selected 32 faces per person as training data, and the rest 32 are for testing. We report the results from the average of ten such random splits of the training and testing images.

To make the initial dictionary discriminative, we trained 38 dictionaries of six items for each person with eight samples using K-SVD, and combine them as our initial dictionary of 228 items. The remaining 24×38 training samples are randomly permutated as sequential input signals to our online algorithm. The dictionary size and the item labels are fixed during the learning process. We conducted two experiments on this dataset for the purpose discussed previously.

Experiment 1 We compare our approach with two supervised methods: LC-KSVD and D-KSVD. We fixed $\phi_{low} = 4.5$ for automatic labeling, and incrementally tune ϕ_{high}, each value corresponding to a set of selected samples for manual labeling. The same number of manually labeled samples are used as training set for D-KSVD and LC-KSVD. Figure 2(a) shows that the recognition rate goes up as the number of labeled samples increases as expected. Our approach takes all the training samples regardless of whether they are labeled or unlabeled, and thus achieves a higher recognition rate even with few manually labeled data (the left end of the curve).

To demonstrate the impact of the lower threshold, we present another set of curves in Figure 2(b). Each curve corresponds to recognition rate growing with the number of manually labeled samples for a given value of the lower threshold. All curves are obtained with the same set of parameters (α, β and γ) and the same set of higher thresholds.

From the curves we clearly see that a higher ϕ_{low}, i.e. more automatic labels, is most beneficial to the case when manual labels are scarce (the left end of the curves). When the number of manual labels increase, the recognition rates with different lower thresholds tend to converge. In addition, the curve with $\phi_{low} = 4.5$ in Figure 2(b) is different from the curve in Figure 2(a) due to different parameter settings.



(a) (b)

Fig. 2. Recognition performance on the Extended YaleB. (a) Recognition performance with varying number of labeled samples, where $K = 6 \times 38$ and $N = 24 \times 38$; (b) An illustration of the effect of the lower bound. The curves are obtained with the same set of parameters: α, β, γ and the same set of higher entropy thresholds.

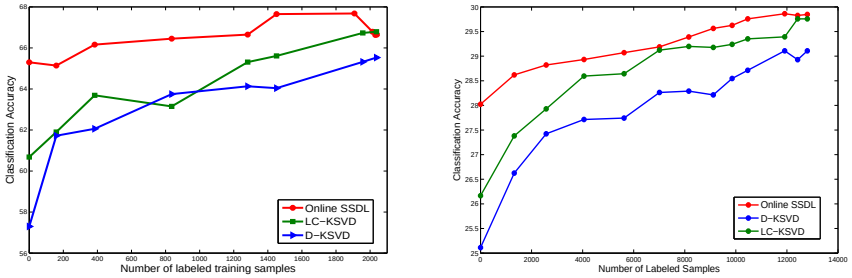
Table 2. Recognition results using spatial pyramid features on the Caltech101. The accuracies of the other results are copied from the references.

Training Images	5	10	15	20	25	30
Malik [28]	46.6	55.8	59.1	62.0	-	66.20
Lazebnik [29]	-	-	56.4	-	-	64.6
Griffin [27]	44.2	54.5	59.0	63.3	65.8	67.60
Irani [30]	-	-	65.0	-	-	70.40
Grauman [31]	-	-	61.0	-	-	69.10
Venkatesh [6]	-	-	42.0	-	-	-
Gemert [32]	-	-	-	-	-	64.16
Yang [2]	-	-	67.0	-	-	73.20
Wang [14]	51.15	59.77	65.43	67.74	70.16	73.44
SRC [3]	48.8	60.1	64.9	67.7	69.2	70.7
K-SVD [11]	49.8	59.8	65.2	68.7	71.0	73.2
D-KSVD [5]	49.6	59.5	65.1	68.6	71.1	73.0
IDL [14]	51.2	61.5	65.7	68.4	71.6	-
LSDL [17]	52.8	61.5	65.7	68.4	71.5	-
ODLSC [13]	52.8	61.5	65.6	68.5	71.3	72.4
LC-KSVD [9]	54.0	63.1	67.7	70.5	72.3	73.6
Online SSDL	**55.0**	**62.6**	**67.2**	**69.6**	**72.4**	**74.3**

Experiment 2 In the second experiment, we compare with other online dictionary learning approaches: ODLSC [13], IDL [14] and LSDL [17], and some state-of-art dictionary learning approaches [11, 5, 3, 14, 9]. Here we set $\phi_{low} = \phi_{high} = 0$, i.e. we get an online dictionary learning algorithm in which all new samples are labeled, as opposed to supervised algorithm in batch mode (LC-KSVD) and unsupervised online algorithms such as ODLSC, IDL, LSDL. As shown in Table 1, our approach (referred to as Online SSDL) has the best performance.

4.2 Caltech101 Dataset

The Caltech101 dataset [26] contains 9,144 images of 102 categories (101 categories of objects and a 'background' category). There are about 40 to 800 images per category. All images are resized to be smaller than 300×300 pixels. We extract sift descriptor with 128 dimension from 16×16 patches. Then we extract the spatial pyramid features with three grids of size 1×1, 2×2 and 4×4, and reduce them to 3,000 dimensions by PCA. Similarly, we conducted two experiments: one is the recognition versus the number of manual labels (seen in Figure 3(a)), and the other is a comparison with the state-of-art methods, using 5, 10, 15,



(a) (b)

Fig. 3. Recognition rate on Caltech101 and Caltech256 with varying number of labeled samples. (a) Caltech101 with $K = 10 \times 102$ and $N = 20 \times 102$; (b) Caltech256 with $K = 3 \times 256$ and $N = 50 \times 102$;.

20, 25 and 30 training samples per category. The results are summarized in Table 4.1. The training samples are randomly selected from each category, and the remaining images are used for testing. We repeated this sampling process to get ten splits and report their average. Following the experimental settings for other methods, we trained dictionaries of the same size as the training samples, *i.e.*, $K = 510, 1020, 1530, 2040, 2550, 3060$. Again, by setting $\phi_{low} = \phi_{high} = 0$, we essentially label all the training data, and this yields the best performance compared to the competition. As shown in Table 4.1, our approach is comparable to LC-KSVD but outperforms the other methods because we take the discriminative error into account.

4.3 Caltech256 Dataset

The Caltech256 dataset [27] contains $30,607$ images of 256 categories. There are at least 80 images per category. Compared to Caltech101 dataset, it is much more difficult due to the variability in object location, pose and size, etc. In contrast to Caltech101, here we extract HOG descriptors from each patch at three scales, 16×16, 25×25 and 31×31. The dimension of each HOG descriptor is 128. We extracted the spatial pyramid features using 4×4, 2×2 and 1×1 sub-regions. Finally we reduce the dimension of the features to 305 using PCA. We used 15, 30, 45 and 60 training samples per class for dictionary learning. Again, training images are randomly selected from each category and all are manually labeled. But unlike the common setup, where the dictionary size equals the number of training samples, we trained dictionaries that contains only 3 items per class. Also, consistent with our previous experiments, we used low-dimensional features and a simple linear classifier instead of sophisticated features and discriminative classifiers such as SVMs. As shown in Table 4.3, our approach achieves good performance even with a simple classifier and significantly smaller dictionary sizes. Note that the accuracies in the first three rows (group 1) are copied from the references, and the rest (group 2) are obtained from our implementation. The differences in experimental settings might account for the average drop in performance of group 2. The recognition performances with varying number of labeled samples perclass are presented in Figure 3(b). The advantage of our method is shown especially when the manual labels are few.

Table 3. Recognition results using spatial pyramid features on the Caltech256. The accuracies in the first three rows are copied from the references, and the rest are obtained from our implementations. In our own implementation, dictionary size is fixed to be $3\times256 = 768$)

Training Images	15	30	45	60
Griffin [27]	28.30	34.10	-	-
Gemert [32]	-	27.17	-	-
Yang [2]	27.73	34.02	37.46	40.14
IDL [14]	19.9	21.7	23.9	26.3
LSDL [17]	23.3	25.6	28.4	30.5
ODLSC [13]	19.3	21.3	23.6	26.1
LC-KSVD [9]	24.6	28.6	30.3	34.9
Online SSDL	**27.9**	**31.9**	**34.4**	**36.7**

5 Conclusion

We proposed an online semi-supervised dictionary learning approach for classification. It's particularly suitable for large scale datasets where batch mode doesn't work well. Moreover, by using a probabilistic model of the sparse codes, our algorithm actively seeks for the critical points for labeling, and identifies the easily classified points as labeled data. In this way we reduce the manual labeling effort to the minimum without sacrificing the performance too much. The fact that the dictionary and the classifier are jointly learned further enhances the discriminative power. Experimental results showed that our approach achieves state-of-art performance. Possible future work includes updating the learned discriminative dictionary for input signals from a new category.

Acknowledgement. This work was supported by the Army Research Office MURI Grant W911NF-09-1-0383

References

1. Elad, M., Aharon, M.: Image denosing via sparse and redundant representations over learned dictionaries. IEEE Trans. Img. Proc. **54** (2006) 3736–3745
2. Yang, J., Yu, K., Gong, Y., Huang, T.: Linear spatial pyramid matching using sparse coding for image classification (2009) *CVPR*.
3. Wright, J., Yang, M., Ganesh, A., Sastry, S., Ma, Y.: Robust face recognition via sparse representation. TPAMI **31** (2009) 210–227
4. Bradley, D., Bagnell, J.: Differential sparse coding (2008) *NIPS*.
5. Zhang, Q., Li, B.: Discriminative k-svd for dictionary learning in face recognition (2010) *CVPR*.
6. Pham, D., Venkatesh, S.: Joint learning and dictionary construction for pattern recognition (2008) *CVPR*.
7. Mairal, J., Bach, F., Ponce, J., Sapiro, G., Zisserman, A.: Supervised dictionary learning (2009) *NIPS*.
8. Mairal, J., Bach, F., Ponce, J., Sapiro, G., Zisserman, A.: Discriminative learned dictionaries for local image analysis (2008) *CVPR*.
9. Jiang, Z., Lin, Z., Davis, L.: Learning a distriminative dictionary for sparse coding via label consistent k-svd (2011) *CVPR*.
10. Qiu, Q., Jiang, Z., Davis, L.: Sparse dictionary-based representation and recognition of action attributes (2011) *ICCV*.
11. Aharon, M., Elad, M., Bruckstein, A.: K-svd: An algorithm for designing overcomplete dictionries for sparse representation. IEEE Trans. on Signal Processing **54** (2006) 4311–4322

12. Yang, J., Yu, K., Huang, T.: Supervised translation-invariant sparse coding (2010) *CVPR*.
13. Marial, J., Bach, F., Ponce, J., Sapiro, G.: Online dictionary learning for sparse coding (2009) *ICML*.
14. Wang, J., Yang, J., Yu, K., Lv, F., huang, T., Gong, Y.: Locality-constrained linear coding for image classification (2010) *CVPR*.
15. Raina, R., Battle, A., Lee, H., Packer, B., Ng, A.: Self-taught learning: Transfer learning from unlabeled data (2007) *ICML*.
16. Zeng, H., Wang, X., Chen, Z., Lu, H., Ma, W.: Clustering based text classification requiring minimal labeled data (2003) *ICDM*.
17. B. Xie, M. Song, D.T.: Large-scale dictionary learning for local coordinate coding (2010) *BMVC*.
18. Boureau, Y., Bach, F., LeCun, Y., Ponce, J.: Learning mid-level features for recognition (2010) *CVPR*.
19. Grosse, R., Raina, R., Kwong, H., Ng, A.Y.: Shift-invariant sparse coding for audio classification (2007) *Conf. on Uncertainty in AI*.
20. Zhang, W., Surve, A., Fern, X., Dietterich, T.: Learning non-redundant codebooks for classifying complex objects (2009) *ICML*.
21. Rodriguez, F., Sapiro, G.: Sparse representations for image classification: Learning discriminative and reconstructive non-parametric dictionaryies (2007) *IMA Preprint 2213*.
22. Yang, L., Jin, R., Sukthankar, R., Jurie, F.: Unifying discriminative visual codebook genearation with classifier training for object category recognition (2008) *CVPR*.
23. Lian, X., Li, Z., Lu, B., Zhang, L.: Max-margin dictionary learning for multiclass image categorization (2010) *ECCV*.
24. Aharon, M., Elad, M.: Sparse and redundant modeling of image content using an image-signaturedictionary. SIAM J. Imaging Sciences **1** (2008) 228–274
25. Georghiades, A., Belhumeur, P., Kriegman, D.: From few to many: Illumination cone models for face recognition under variable lighting and pose. TPAMI **23** (2001) 643–660
26. FeiFei, L., Fergus, R., Perona, P.: Learning generative visual models from few training samples: An incremental bayesian appoach tested on 101 object categories (2004) *CVPR Workshop on Generative Model Based Vision*.
27. Griffin, G., Holub, A., Perona, P.: Caltech-256 object category dataset (2007) *CIT Technical Report 7694*.
28. Zhang, H., Berg, A., Maire, M., Malik, J.: Svm-knn: Discriminative nearest neighbor classification for visual category recognition (2006) *CVPR*.
29. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories (2007) *CVPR*.
30. Boiman, O., Shechtman, E., Irani, M.: In defense of nearest-neighor based image classification (2008) *CVPR*.
31. Jain, P., Kullis, B., Grauman, K.: Fast image search for learned metrics (2008) *CVPR*.
32. Gemert, J., Geusebroek, J., Veenman, C., Smeulders, A.: Kernel codebooks for scene categorization (2008) *ECCV*.