Ambiguously Labeled Learning Using Dictionaries

Yi-Chen Chen, *Student Member, IEEE*, Vishal M. Patel, *Member, IEEE*,
Rama Chellappa, *Fellow, IEEE*, and P. Jonathon Phillips, *Fellow, IEEE*

***Abstract*— We propose a dictionary-based learning method for ambiguously labeled multiclass classification, where each training sample has multiple labels and only one of them is the correct label. The dictionary learning problem is solved using an iterative alternating algorithm. At each iteration of the algorithm, two alternating steps are performed: 1) a confidence update and 2) a dictionary update. The confidence of each sample is defined as the probability distribution on its ambiguous labels. The dictionaries are updated using either soft or hard decision rules. Furthermore, using the kernel methods, we make the dictionary learning framework nonlinear based on the soft decision rule. Extensive evaluations on four unconstrained face recognition datasets demonstrate that the proposed method performs significantly better than state-of-the-art ambiguously labeled learning approaches.**

***Index Terms*— Semi-supervised clustering, ambiguously labeled learning, multiclass classification, dictionary learning, kernel methods.**

I. Introduction

IN MANY practical image and video applications, one has access only to ambiguously labeled data. For example, given a picture with multiple faces and a caption specifying who is in the picture, the reader may not know which face goes with the names in the caption (See Fig. 1). Another example is that people may sometimes want to label humans or objects of interest in an image based on their partial knowledge of these objects. For instance, one may be asked to name the recently introduced friends or colleagues he met at a meeting or name the categories of plants he saw in a field trip. In these cases, one may not be able to give the ground truth label of an object or a person but only a set of possible labels. Data labeling by humans can be time consuming and inaccurate. This highlights the significance of learning from ambiguously labeled samples. The ambiguously labeled learning has a wide range of applications including recognition of human faces, fingerprints, actions and behaviors.

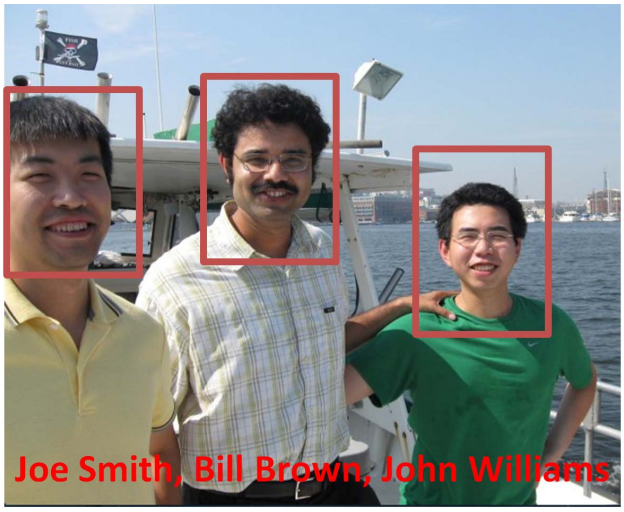


Fig. 1. Each face is associated with three names out of which only one is the true name.

Several papers have been published in the literature that address the ambiguous label problem. In [1], a discriminative framework was proposed based on the Expectation Maximization (EM) algorithm [2], with a maximum likelihood approach to disambiguate correct labels from incorrect ones. A semi-supervised dictionary-based learning method was proposed in [3] under the assumption that there are either labeled samples or totally unlabeled samples available for training. The method iteratively estimates the confidence of unlabeled samples belonging to each class and uses it to refine the learned dictionaries. In [4] and [5], a method was presented to determine the label using a multi-linear classifier that minimizes a convex loss function. The loss function used in [4] and [5] was shown to be a tighter convex upper bound on 0/1 loss when compared to an un-normalized 'naive' method that treats each example as if it took on multiple correct labels. Several non-parametric, instance-based algorithms for partially labeled learning were proposed in [6].

In recent years, sparse and redundant signal representations have generated interest in the image processing, vision and machine learning communities [7]–[11]. This is due in part to the fact that objects and images of interest can be represented sparsely in an appropriately chosen dictionary. We say a signal $\mathbf{x}$ (in the column-vectorized form) is sparse in dictionary $\mathbf{D}$ if it can be approximated by $\mathbf{x} \approx \mathbf{Dt}$,

Manuscript received February 19, 2014; revised July 16, 2014 and September 3, 2014; accepted September 3, 2014. Date of publication September 22, 2014; date of current version November 10, 2014. This work was supported by the Office of Naval Research, Arlington, VA, USA. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Aly A. Farag.

Y.-C. Chen and R. Chellappa are with the Center for Automation Research, Department of Electrical and Computer Engineering, University of Maryland Institute for Advanced Computer Studies, College Park, MD 20742 USA, and also with the University of Maryland, College Park, MD 20742 USA (e-mail: chenyc08@umd.edu; rama@umiacs.umd.edu).

V. M. Patel is with the Center for Automation Research, Department of Electrical and Computer Engineering, University of Maryland Institute for Advanced Computer Studies, College Park, MD 20742 USA (e-mail: pvishalm@umd.edu).

P. J. Phillips is with the National Institute of Standards and Technology, Gaithersburg, MD 20899 USA (e-mail: jonathon@nist.gov).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2014.2359642

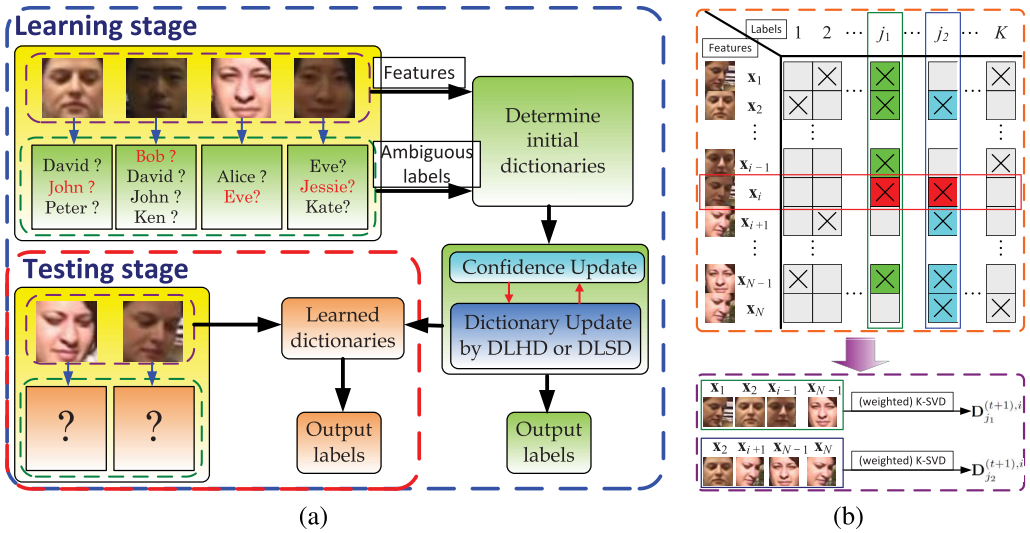


Fig. 2. The proposed dictionary learning method. (a) Block diagram - In the learning stage, given ambiguously labeled training samples (e.g. faces), the algorithm iterates with confidence update and dictionary update steps. In the testing stage, the learned dictionaries are used to determine the label of test images. (b) An illustration of how common label samples are collected to learn intermediate dictionaries, which are used to update the confidence for sample x_i.

where $\mathbf{t}$ is a sparse vector and $\mathbf{D}$ is a dictionary. Columns of $\mathbf{D}$ have the same dimension as $\mathbf{x}$ and they are called atoms. The dictionary $\mathbf{D}$ can be analytic such as a redundant Gabor dictionary or it can be trained directly from data. It has been observed that learning a dictionary directly from training data rather than using a predetermined dictionary usually leads to better representation. Thus, learned dictionaries generally have superior results in many practical image processing applications such as restoration and classification. This has motivated researchers to develop dictionary learning algorithms for supervised [12]–[16] semi-supervised [3] and unsupervised [17]–[19] learning. In this paper, we consider a dictionary learning problem where each training sample is provided with a set of possible labels and only one label among them is the true one. We develop dictionary learning algorithms that process ambiguously labeled data.

Fig. 2(a) shows the block diagram of the proposed dictionary learning method. Given ambiguously labeled training samples (e.g. faces), the algorithm consists of two main steps: confidence update and dictionary update. The confidence for each sample is defined as the probability distribution on its ambiguous labels. In the confidence update phase, the confidence is updated for each sample according to its residuals when the sample is projected onto different class dictionaries. Then, the dictionary is updated using a fixed confidence. We propose two effective approaches for updating the dictionary: dictionary learning with hard decision (DLHD), and dictionary learning with soft decision (DLSD). The DLSD is shown to be an EM-based dictionary learning approach, where class dictionaries are learned using a weighted K-SVD algorithm with weighting parameters computed by soft decision on the given confidence. In the testing stage, a novel test image is projected onto the span of the atoms in each learned dictionary. The resulting residual is then used for classification. Furthermore, to handle the non-linearities present in the data, we kernelize the proposed dictionary learning algorithm. We evaluate our approaches on four face recognition datasets: Labeled Faces in the Wild (LFW) [20], CMU PIE dataset [21], TV series 'LOST' dataset [5] and a dataset collected at the University of Maryland (UMD) [22].

The key contributions of our work are[1]:

1. We propose a dictionary-based learning method when ambiguously labeled data are provided for training.
2. We present two effective approaches for updating the dictionary.
3. We extend our method from linear to non-linear cases by kernelizing dictionary learning in the high-dimensional feature space.

The rest of the paper is organized as follows. In Section II, we formulate the ambiguously labeled learning problem and present the details of the proposed dictionary learning algorithms. In Section III, we present the non-linear dictionary learning in the kernel space. In Section IV, we demonstrate experimental results with discussions. We conclude this paper in Section V.

II. Dictionary Learning From Ambiguously Labeled Data

Let $\mathcal{L} = \{(x_i, L_i), i = 1, \ldots, N\}$ be the training data. Here x_i denotes the i^{th} training sample, $L_i \subset \{1, 2, \ldots, K\}$ the corresponding multiple label set, and N the number of training samples. There are a total of K classes. The true label z_i of the i^{th} training sample is in the multi-label set L_i. Let $\mathbf{x}_i \in \mathbb{R}^d$ denote the lexicographically ordered vector representing the sample x_i. For each feature vector $\mathbf{x}_i$ and for each class j, we define a latent variable $p_{i,j}$, which represents the confidence of $\mathbf{x}_i$ belonging to the j^{th} class. By definition, we have $\sum_j p_{i,j} = 1$, and

$$\begin{aligned} p_{i,j} &= 0 \quad \text{if } j \notin L_i, \quad i = 1, \ldots, N, \\ p_{i,j} &\in (0, 1] \quad \text{if } j \in L_i, \quad i = 1, \ldots, N. \end{aligned} \tag{1}$$

[1]Items **1** and **2** summarize the preliminary version of this work that appeared in [23]. Item **3** and experiments on the challenging UMD video dataset [22] are extensions to [23].

Let $\mathbf{P}$ be the confidence matrix with entry $p_{i,j}$ in the i-th row and j-th column. Define $\mathbf{C}_j$ to be the collection of samples in class j represented as a matrix and $\mathbf{C} = [\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_K]$ be the concatenation of all samples from different classes. Similarly, let $\mathbf{D}_j$ be the dictionary that is learned from the data in $\mathbf{C}_j$ and $\mathbf{D} = [\mathbf{D}_1, \mathbf{D}_2, \ldots, \mathbf{D}_K]$ be the concatenation of all dictionaries. Equipped with the above notation, the problem we study can be formally stated as follows:

For each feature vector available during training, we are given a set of labels, only one of which is correct. Given this ambiguously labeled data, how can one learn dictionaries to represent each class?

We solve the dictionary learning problem using an iterative alternating algorithm. At each iteration, two major steps are performed: confidence update and dictionary update. We demonstrate that both soft and hard decision rules produce robust dictionaries.

A. The Dictionary Learning Hard Decision Approach

The dictionary learning hard decision (DLHD) approach learns dictionaries directly from class matrices,[2] $\{\mathbf{C}_i\}_{i=1}^K$, that are determined using a hard decision for class labels for each sample x_i by selecting the classes with the maximum $p_{i,c}$ among all c's belonging to L_i. One iteration of the algorithm consists of the following stages.

1) Confidence Update: We use the notation $\mathbf{D}^{(t)}, \mathbf{P}^{(t)}$ to denote the dictionary matrix and confidence matrix respectively, in the t^{th} iteration. Keeping the dictionary $\mathbf{D}^{(t)}$ fixed, the confidence of a feature vector belonging to classes outside its label set is fixed to 0 and is not updated. To update the confidence of a sample belonging to classes in its label set, we first make the observation that a sample $\mathbf{x}_i$ which is well represented by the dictionary of class j, should have high confidence. In other words, the confidence of a sample $\mathbf{x}_i$ belonging to a class j should be inversely proportional to the reconstruction error that results when $\mathbf{x}_i$ is projected onto $\mathbf{D}_j$. This can be done by updating the confidence matrix $\mathbf{P}^{(t)}$ as follows

$$p_{i,j}^{(t)} = \frac{\beta_j^{(t)} \exp\left(-\frac{e_{i,j}^{(t)}}{2\sigma_j^{(t)}}\right)}{\sum\limits_{k \in L_i} \beta_k^{(t)} \exp\left(-\frac{e_{i,k}^{(t)}}{2\sigma_k^{(t)}}\right)}, \tag{2}$$

where $\beta_j^{(t)}$ and $\sigma_j^{(t)}$ are parameters (given in section II-C), and

$$e_{i,j}^{(t)} = \|\mathbf{x}_i - \mathbf{D}_j^{(t)} \overline{\mathbf{D}_j^{(t)}} \mathbf{x}_i\|_2^2 \tag{3}$$

is the reconstruction error, when $\mathbf{x}_i$ is projected onto $\mathbf{D}_j^{(t)}, \forall j \in L_i$ and

$$\overline{\mathbf{D}_j^{(t)}} \triangleq ((\mathbf{D}_j^{(t)})^T \mathbf{D}_j^{(t)})^{-1} (\mathbf{D}_j^{(t)})^T$$

is the pseudo-inverse of $\mathbf{D}_j^{(t)}$. As shown in section II-C, we derive (2) under the assumption that the likelihood of each sample $\mathbf{x}_i$ is a mixture of Gaussian densities, and $\beta_j^{(t)}$ is the weight associated with the density of label j.

[2]We refer to class matrices and clusters interchangeably.

2) Cluster Update[3]: Once the confidence matrix $\mathbf{P}^{(t)}$ is updated, we use it to update the class matrix $\mathbf{C}^{(t+1)}$. For each training sample $\mathbf{x}_i$, we assign it to the class j^i which gives the maximum confidence. That is,

$$j^i = \underset{k \in L_i}{\operatorname{argmax}}\, p_{i,k}^{(t)}. \tag{4}$$

3) Dictionary Update: The updated class matrices $\mathbf{C}^{(t+1)}$ are then used to train class-specific dictionaries. Given a class matrix $\mathbf{C}_j^{(t+1)}$, we seek a dictionary $\mathbf{D}_j^{(t+1)}$ that provides the sparsest representation for each example feature in this matrix, by solving the following optimization problem

$$(\mathbf{D}_j^{(t+1)},\ \mathbf{\Gamma}_j^{(t+1)}) = \underset{\mathbf{D},\mathbf{\Gamma}}{\operatorname{argmin}} \|\mathbf{C}_j^{(t+1)} - \mathbf{D}\mathbf{\Gamma}\|_F^2, \quad \text{subject to } \|\boldsymbol{\gamma}_i\|_0 \le T_0,\ \forall i, \tag{5}$$

where $\boldsymbol{\gamma}_i$ represents the i-th column of $\mathbf{\Gamma}$, $\mathbf{C}_j^{(t+1)}$ has a matrix representation whose columns are feature vectors assigned to the j-th class at iteration $(t+1)$, and T_0 is the sparsity parameter. Here, $\|\cdot\|_F$ denotes the Frobenius norm and $\|\cdot\|_0$ represents the ℓ_0 norm which counts the number of nonzero elements in a vector. Many approaches have been proposed in the literature for solving such optimization problem. We adapt the K-SVD algorithm [24] for solving (5) due to its simplicity and fast convergence. The K-SVD algorithm alternates between sparse-coding and dictionary update steps. In the sparse-coding step, $\mathbf{D}$ is fixed and the representation vectors $\boldsymbol{\gamma}_i$s are found for the i-th column in $\mathbf{C}_j^{(t+1)}$ as follows

$$\min_{\boldsymbol{\gamma}_i} \|\mathbf{c}_i - \mathbf{D}\boldsymbol{\gamma}_i\|_2^2, \text{ subject to } \|\boldsymbol{\gamma}_i\|_0 \le T_0 \ \ \forall i,$$

where $\mathbf{c}_i$ is the i-th column of $\mathbf{C}_j^{(t+1)}$. Then, the dictionary is updated atom-by-atom in an efficient way. For a given atom k, the quadratic term in (5) can be rewritten as

$$\left\|\mathbf{C}_j^{(t+1)} - \sum_{i \ne k} \mathbf{d}_i \boldsymbol{\gamma}_T^k - \mathbf{d}_k \boldsymbol{\gamma}_T^k\right\|_F^2 = \|\mathbf{E}_k - \mathbf{d}_k \boldsymbol{\gamma}_T^k\|_F^2, \tag{6}$$

where $\mathbf{E}_k$ is the residual matrix, $\mathbf{d}_k$ is the k-th atom of the dictionary $\mathbf{D}$ and $\boldsymbol{\gamma}_T^k$ is the k-th row of $\mathbf{\Gamma}$. The atom update is obtained by minimizing (6) for $\mathbf{d}_k$ and $\boldsymbol{\gamma}_T^k$ through a simple rank-1 approximation of $\mathbf{E}_k$ [24].

The entire approach for learning dictionaries from ambiguously labeled data using hard decisions is summarized in Algorithm 1.

B. The Dictionary Learning Soft Decision Approach

The dictionary learning soft decision (DLSD) approach learns dictionaries that are used to update the confidence for each sample $\mathbf{x}_i$, based on the weighted distribution of other samples that share the same candidate label belonging to L_i. The weighted distribution of other samples sharing a given candidate label c is computed through the normalization of all $p_{l,c}$'s with $l \ne i$. In what follows, we describe the different steps of the algorithm.

[3]This step is necessary only for the DLHD approach.

Algorithm 1 Iteratively Learning Dictionaries Using Hard Decision and Updating Confidence

Input: Training samples $\mathcal{L} = \{(x_i, L_i)\}$ and initial dictionaries $\mathbf{D}^{(0)} = [\mathbf{D}_1^{(0)}|\mathbf{D}_2^{(0)}|\dots|\mathbf{D}_K^{(0)}]$.
Output: Dictionary $\mathbf{D}^* = [\mathbf{D}_1^*|\mathbf{D}_2^*|\dots|\mathbf{D}_K^*]$.
Algorithm:
1. Repeat the following steps to refine the confidence until the maximum iteration number $\mathrm{T_c}$ is reached:
1.1 **Confidence Update**: For each feature vector $\mathbf{x}_i$, calculate the residuals $e_{i,j}^{(t)}$ using (3). Then use $e_{i,j}^{(t)}$ to update confidence $p_{i,j}^{(t)}$ using (2).
1.2 **Cluster Update**: Assign each feature vector $\mathbf{x}_i$ to $\mathbf{C}_{j^i}^{(t+1)}$ according to (4).
1.3 **Dictionary Update**: When the class assignment for all $\mathbf{x}_i$'s is completed, build dictionary $\mathbf{D}_j^{(t+1)}$ from $\mathbf{C}_j^{(t+1)}, \forall j \in \{1, 2, \dots, K\}$ using the K-SVD algorithm and obtain $\mathbf{D}^{(t+1)} = [\mathbf{D}_1^{(t+1)}|\mathbf{D}_2^{(t+1)}|\dots|\mathbf{D}_K^{(t+1)}]$.
2. Return $\mathbf{D}^* = \mathbf{D}^{(\mathrm{T_c})}$, where $\mathrm{T_c}$ is the iteration number at which the learning algorithm converges.

Algorithm 2 Iteratively Learning Dictionaries Using Soft Decision and Updating Confidence

Input: Training samples $\mathcal{L} = \{(x_i, L_i)\}$.
Output: Dictionary $\mathbf{D}^* = [\mathbf{D}_1^*|\mathbf{D}_2^*|\dots|\mathbf{D}_K^*]$.
Algorithm:
1. Repeat the following iterations to refine confidence until the maximum iteration number $\mathrm{T_c}$ is reached:
1.1 **Confidence Update**: Use (7) to calculate residuals $e_{i,n_l}^{(t),i},\ \forall n_l \in L_i$. Then, use $e_{i,n_l}^{(t)}$ to update confidence $p_{i,n_l}^{(t)}$ by (2) to obtain the confidence matrix $\mathbf{P}^{(t+1)}$.
1.2 **Dictionary Update**: Based on $\mathbf{P}^{(t)}$, do the following for each $\mathbf{x}_i$ with $L_i = \{n_1, n_2, \dots n_{|L_i|}\}$: Construct the weighting matrix $\mathbf{W}$ and use (8) to build $\mathbf{D}_{n_l}^{(t+1),i}$ from which the dictionary
$$\mathbf{D}^{(t+1),i} = [\mathbf{D}_{n_1}^{(t+1),i}|\mathbf{D}_{n_2}^{(t+1),i}|\dots|\mathbf{D}_{n_{|L_i|}}^{(t+1),i}]$$
is obtained.
2. When $t = \mathrm{T_c}$, follow 1.2 and 1.3 in Algorithm 1 to build the final dictionary $\mathbf{D}^* = \mathbf{D}_c^{(\mathrm{T_c})}$.

1) Confidence Update: In this step, given the intermediate dictionary $\mathbf{D}^{(t),i}$ learned from the previous iteration for each sample $\mathbf{x}_i$, we calculate the residuals $e_{i,n_l}^{(t),i}$ using $\mathbf{D}_{n_l}^{(t),i}$ for all n_l in L_i as

$$e_{i,n_l}^{(t),i} = \|\mathbf{x}_i - \mathbf{D}_{n_l}^{(t),i}\overline{\mathbf{D}_{n_l}^{(t),i}}\mathbf{x}_i\|_2^2. \tag{7}$$

We then use (2) to update the confidence $p_{i,n_l}^{(t)}$, with $e_{i,j}^{(t)}$ replaced by $e_{i,n_l}^{(t),i}$.

2) Dictionary Update: In this step, the confidence matrix $\mathbf{P}^{(t)}$ is given. For each $\mathbf{x}_i$, we build the intermediate dictionaries for all labels in $L_i = \{n_1, n_2, \dots n_{|L_i|}\}$. In particular, we learn

$$\mathbf{D}^{(t+1),i} = [\mathbf{D}_{n_1}^{(t+1),i}|\mathbf{D}_{n_2}^{(t+1),i}|\dots|\mathbf{D}_{n_{|L_i|}}^{(t+1),i}],$$

where each $\mathbf{D}_{n_l}^{(t+1),i}$ is built using soft decision from samples $\mathbf{x}_l$'s, where $l \neq i$ and $p_{l,n_l}^{(t+1)} > 0$. Fig. 2(b) shows an example of how these common ambiguous label samples are collected to learn the intermediate dictionaries $\mathbf{D}_{n_l}^{(t+1),i}$. The cell marked with '$\times$' at the (i, j) entry indicates a non-zero $p_{i,j}^{(t)}$. All the other empty cells indicate zero confidence. As shown in this example, only samples corresponding to green and blue cells are used to learn $\mathbf{D}_{n_1}^{(t+1),i}$ and $\mathbf{D}_{n_2}^{(t+1),i}$, respectively. To learn the intermediate dictionaries for $\mathbf{x}_i$, exclusion of $\mathbf{x}_i$ (corresponding to red cells) is necessary to enhance discriminative learning. Let $\{\mathbf{x}_{i_m}\}_{m=1}^{N(i,n_l)}$ be the collection of these samples. Its matrix form is denoted by $\mathbf{Y} = [\mathbf{y}_1\ \mathbf{y}_2 \dots\ \mathbf{y}_{N(i,n_l)}]$, where $\mathbf{y}_m$, $m \in \{1, \dots, N(i, n_l)\}$, is a column vectorized form of some collected sample $\mathbf{x}_{i_m}$. Let

$$\mathbf{w} = [w_1\ w_2 \dots\ w_{N(i,n_l)}] = [p_{i_1,n_l}^{(t)}\ p_{i_2,n_l}^{(t)} \dots\ p_{i_{N(i,n_l)},n_l}^{(t)}],$$

where the weight w_m reflects the relative amount of contribution from $\mathbf{x}_{i_m}$ when learning the dictionary. The objective of the weighted K-SVD algorithm can then be formulated as

$$\begin{aligned}(\mathbf{D}_{n_l}^{(t+1),i},\ \mathbf{\Gamma}_{n_l}^{(t+1),i}) &= \underset{\mathbf{D},\mathbf{\Gamma}}{\operatorname{argmin}} \sum_{m=1}^{N(i,n_l)} w_m\|\mathbf{y}_m - \mathbf{D}\boldsymbol{\gamma}_m\|_2^2,\\ &\quad \text{subject to } \|\boldsymbol{\gamma}_m\|_0 \le T_0, \forall m,\\ &= \underset{\mathbf{D},\mathbf{\Gamma}}{\operatorname{argmin}} \|(\mathbf{Y} - \mathbf{D}\mathbf{\Gamma})\mathbf{W}\|_F^2,\\ &\quad \text{subject to } \|\boldsymbol{\gamma}_m\|_0 \le T_0, \forall m,\end{aligned} \tag{8}$$

where $\mathbf{W}$ is a square weighting matrix with its diagonal filled with $\{\sqrt{w_m}\}_{m=1}^{N(i,n_l)}$, and zeros elsewhere. One can solve the above weighted optimization problem by modifying the K-SVD algorithm as follows:

- *Sparse Coding Stage:* For $m = 1, 2, \dots, N(i, n_l)$, compute $\boldsymbol{\gamma}_m$ for $\mathbf{y}_m$ by solving
$$\min_{\boldsymbol{\gamma}} \|(\mathbf{y}_m - \mathbf{D}\boldsymbol{\gamma})\|_2^2,\ \text{subject to } \|\boldsymbol{\gamma}\|_0 \le T_0.$$
- *Codebook Update Stage:* This step remains the same as the original K-SVD algorithm except that the overall error representation matrix $\mathbf{E}_k$ is changed to
$$\mathbf{E}_k = (\mathbf{Y} - \sum_{j \neq k} \mathbf{d}_j \boldsymbol{\gamma}_T^j)\mathbf{W},$$
where $\mathbf{d}_j$ is the j-th column of $\mathbf{D}$ and $\boldsymbol{\gamma}_T^j$ is the j-th row of $\mathbf{\Gamma}$ found in the previous sparse coding stage.

After $\mathrm{T_c}$ soft decision iterations, for each training sample, we assign the label with the maximum confidence. The labeled class matrices are used to learn the final dictionary

$$\mathbf{D}^* = \mathbf{D}^{(\mathrm{T_c})} = [\mathbf{D}_1^{(\mathrm{T_c})}|\mathbf{D}_2^{(\mathrm{T_c})}|\dots|\mathbf{D}_K^{(\mathrm{T_c})}]$$

via the K-SVD algorithm. This step is the same as 1.2 and 1.3 in Algorithm 1 with t set equal to $\mathrm{T_c}$. The entire DLSD approach is summarized in Algorithm 2.

For our current implementation of DLSD, up to $|L_i|$ class dictionaries need to be learned because of the exclusion of sample x_i. If x_i remains in the training set to train all the $|L_i|$ class dictionaries and the normalized confidences of x_i with respect to other samples assigned with the same label as x_i are similar among these $|L_i|$ classes then the resulting dictionaries may not be discriminative for x_i. Therefore, unlike DLHD, in each iteration we don't simply learn a common set of class dictionaries that are used to compute the confidences for all x_i's.

C. DLSD is an EM-Based Approach

The proposed DLSD is indeed an EM [25]– [27] dictionary learning approach. In particular, to find $\mathbf{D}^{(t+1),i}$ given $\mathbf{x}_i$ and $\mathbf{D}^{(t),i}$, in the E-step we first compute the following conditional expectation[4]

$$E\left[\log p(\{\mathbf{x}_l\}_{l=1,l\neq i}^{N}, \{Z_l\}_{l=1,l\neq i}^{N}|\mathbf{D}^i)|\mathbf{x}_i, \mathbf{D}^{(t),i}\right], \tag{9}$$

where Z_l is the random variable that corresponds to the true label z_l of the observed sample $\mathbf{x}_l$. We assume the likelihood of sample $\mathbf{x}_l$ given $\mathbf{D}^i$ is a mixture of Gaussian densities expressed by

$$p(\mathbf{x}_l|\mathbf{D}^i) = \sum_{j=1}^{K} \alpha_j\, p_j(\mathbf{x}_l|\mathbf{D}_j^i),$$

where $\alpha_j's$ are normalized weights associated with the density of label $j's$ with $\sum_{j=1}^{K}\alpha_j = 1$, and

$$p_j(\mathbf{x}_l|\mathbf{D}_j^i) = \frac{1}{\sqrt{2\pi}\,\sigma_j}\exp\left(-\frac{\|\mathbf{x}_l - \mathbf{D}_j^i\boldsymbol{\gamma}_l\|_2^2}{2\sigma_j}\right)$$

for some σ_j. Moreover, $\boldsymbol{\gamma}_l$ is a coefficient vector for representing $\mathbf{x}_l$ using $\mathbf{D}_j^i$. For independent $\mathbf{x}_l's$, it can be shown that (9) equals

$$\sum_{j=1}^{K}\sum_{l=1,l\neq i}^{N} p_{l,j}^{(t)}\big(\log(\alpha_j) + \log(p_j(\mathbf{x}_l|\mathbf{D}_j^i))\big), \tag{10}$$

where

$$p_{l,j}^{(t)} \triangleq p(Z_l = j|\mathbf{x}_l, \mathbf{D}^{(t),i}) = \frac{\alpha_j\, p_j(\mathbf{x}_l|\mathbf{D}_j^{(t),i})}{\sum_{k=1}^{K}\alpha_k\, p_k(\mathbf{x}_l|\mathbf{D}_k^{(t),i})}. \tag{11}$$

In the M-step, we maximize (10) by finding

$$\boldsymbol{\alpha}^{(t+1)} \triangleq [\alpha_1^{(t+1)}, \ldots, \alpha_K^{(t+1)}]$$

and

$$\mathbf{D}^{(t+1),i} = [\mathbf{D}_1^{(t+1),i}|\ldots|\mathbf{D}_K^{(t+1),i}]$$

such that

$$\begin{aligned}
\boldsymbol{\alpha}^{(t+1)} &= \underset{\boldsymbol{\alpha}=[\alpha_1,\alpha_2,\ldots,\alpha_K]}{\operatorname{argmax}} \sum_{j=1}^{K}\sum_{l=1,l\neq i}^{N} p_{l,j}^{(t)}\log(\alpha_j),\\
&= \underset{\alpha_j}{\operatorname{argmax}} \sum_{l=1,l\neq i}^{N} p_{l,j}^{(t)}\log(\alpha_j),\quad \forall j\in\{1,\ldots,K\}, \qquad (12)\\
\mathbf{D}^{(t+1),i} &= \underset{\mathbf{D}=[\mathbf{D}_1^i|\mathbf{D}_2^i|\ldots|\mathbf{D}_K^i]}{\operatorname{argmax}} \sum_{j=1}^{K}\sum_{l=1,l\neq i}^{N} p_{l,j}^{(t)}\log(p_j(\mathbf{x}_l|\mathbf{D}_j^i)),\\
&= \underset{\mathbf{D}_j^i}{\operatorname{argmax}} \sum_{l=1,l\neq i}^{N} p_{l,j}^{(t)}\log(p_j(\mathbf{x}_l|\mathbf{D}_j^i)),\\
&= \underset{\mathbf{D}_j^i}{\operatorname{argmax}} \sum_{l=1,l\neq i}^{N} p_{l,j}^{(t)}\left(-\frac{\log(\sigma_j)}{2} - \frac{\|\mathbf{x}_l - \mathbf{D}_j^i\boldsymbol{\gamma}_l\|_2^2}{2\sigma_j}\right),\\
&= \underset{\mathbf{D}_j^i}{\operatorname{argmin}} \sum_{l=1,l\neq i}^{N} p_{l,j}^{(t)}\|\mathbf{x}_l - \mathbf{D}_j^i\boldsymbol{\gamma}_l\|_2^2,\quad \forall j\in\{1,\ldots,K\},\\
&= \underset{\mathbf{D}_{n_l}^i}{\operatorname{argmin}} \sum_{m=1}^{N(i,n_l)} w_m\|\mathbf{y}_m - \mathbf{D}_{n_l}^i\boldsymbol{\gamma}_m\|_2^2,\quad \forall n_l\in L_i. \qquad (13)
\end{aligned}$$

The optimization problem in (13) can be solved by the weighted K-SVD algorithm in (8). $\sigma_{n_l}^{(t+1)}$ can be approximated by the average residual over $\{\mathbf{y}_m\}_{m=1}^{N(i,n_l)}$. That is,

$$\sigma_{n_l}^{(t+1)} = \frac{1}{\eta(i,n_l)}\sum_{m=1}^{N(i,n_l)} w_m\|\mathbf{y}_m - \mathbf{D}_{n_l}^{(t+1),i}\boldsymbol{\gamma}_m\|_2^2,\quad \forall n_l\in L_i,$$

where $\eta(i,n_l) = \sum_{m=1}^{N(i,n_l)} w_m$. Moreover, as α_{n_l} sums to one over n_l, (12) leads to

$$\alpha_{n_l}^{(t+1)} = \frac{\eta(i,n_l)}{N(i,n_l)}.$$

We then compute

$$\beta_{n_l}^{(t+1)} = \frac{\alpha_{n_l}^{(t+1)}}{\sqrt{2\pi\,\sigma_{n_l}^{(t+1)}}}$$

and update $p_{i,n_l}^{(t+1)}$ by (2).

D. Determining Initial Dictionaries

The performance of both DLSD and DLHD will depend on the initial dictionaries as they determine how well the final dictionaries are learned through successive alternating iterations. As a result, initializing our method with proper dictionaries is critical. In this section, we propose an algorithm that uses both ambiguous labels and features to determine the initial dictionaries.

For the i-th sample, we initialize the corresponding row of $\mathbf{P}$ uniformly for all $j\in L_i$. Hence,

$$\mathbf{P}^{(0)} \triangleq \left[p_{i,j}^{(0)}\right],\ \text{where}\ p_{i,j}^{(0)} = \frac{1}{|L_i|},\ \ \text{if}\ j\in L_i, i = 1,\ldots,N.$$

At iteration $t = 0$, we build dictionaries for the sample $\mathbf{x}_i$, denoted by

$$\mathbf{D}^{(0),i} = [\mathbf{D}_{n_1}^{(0),i}|\mathbf{D}_{n_2}^{(0),i}|\ldots|\mathbf{D}_{n_{|L_i|}}^{(0),i}],$$

[4]Here our interpretation of the DLSD as an EM-based approach is conditioned on the training sample and the corresponding class dictionaries. We have not been able to show that the DLSD is an EM algorithm by minimizing some global cost function.

Algorithm 3 Using Initial Confidence to Learn Initial Dictionaries

Input: Training samples $\mathcal{L} = \{(x_i, L_i)\}$ and the initial confidence, $\mathbf{P}^{(0)}$.

Output: Initial dictionaries $\mathbf{D}^{(0)} = [\mathbf{D}_1^{(0)}|\mathbf{D}_2^{(0)}|\dots|\mathbf{D}_K^{(0)}]$.

Algorithm:

1. Initialization: $i \leftarrow 1$; $\mathbf{C}_j^{(0)} \leftarrow \{\}, \forall j \in \{1, 2, \dots, K\}$.
2. Repeat the following for every $\mathbf{x}_i$:
   - 2.1 Construct $\mathbf{D}^{(0),i} = [\mathbf{D}_{n_1}^{(0),i}|\mathbf{D}_{n_2}^{(0),i}|\dots|\mathbf{D}_{n_{|L_i|}}^{(0),i}]$, where $\mathbf{D}_{n_k}^{(0),i}$ is built from $\mathbf{x}_l$'s such that $l \neq i$.
   - 2.2 Augment $\mathbf{C}_{\hat{j}^i}^{(0)}$ with $\mathbf{x}_i$, where $\hat{j}^i$ is obtained from (14).
3. Establish initial dictionaries $\mathbf{D}^{(0)} = [\mathbf{D}_1^{(0)}|\mathbf{D}_2^{(0)}|\dots|\mathbf{D}_K^{(0)}]$, where $\mathbf{D}_j^{(0)}$ is learned from $\mathbf{C}_j^{(0)}$ using the K-SVD algorithm.

where the intermediate dictionary $\mathbf{D}_{n_k}^{(0),i}$ is learned from samples other than $\mathbf{x}_i$ with ambiguous label $n_k \in L_i$. These samples are collected in the same way as described in section II-B. Next, $\mathbf{x}_i$ is assigned to class $\hat{j}^i$ such that it gives the lowest residual. In other words,

$$\hat{j}^i = \underset{n_k \in L_i}{\operatorname{argmin}} \|\mathbf{x}_i - \mathbf{D}_{n_k}^{(0),i}\overline{\mathbf{D}_{n_k}^{(0),i}}\mathbf{x}_i\|_2^2. \tag{14}$$

Initial clusters are obtained after the class assignment for all samples is completed. Each initial dictionary is then learned from the corresponding cluster using the K-SVD algorithm [24]. We summarize this initialization approach in Algorithm 3.

Note that our method is very different from the approach of learning dictionaries from partially labeled data [3]. The approach presented in [3] learns class discriminative dictionaries while our work learns class reconstructive dictionaries. In addition, from the formulation in [3] we see there are either labeled samples or totally unlabeled samples available for training. In contrast, in our formulation, all samples are ambiguously labeled according to three controlled parameters. In fact, formulations in [3] and [17] (for totally unlabeled samples) are special cases of the ambiguously labeled formulation presented in this paper.

III. Non-Linear Kernel Dictionary Learning

The class identities in the face dataset may not be linearly separable. This essentially requires the dictionary learning model to be non-linear [28], [29]. In this section, we formulate the problem of kernel dictionary learning with soft decision. The kernel dictionary learning with hard decision can easily be obtained by replacing the weight matrix with the one determined by the hard-threshold version of $\mathbf{P}$, where $p_{i,j^i} = 1$, and $p_{i,j} = 0, \forall j \neq j^i \in L_i, \forall i$. Note that j^i is computed by (4).

Let $\mathbf{\Phi} : \mathbb{R}^d \rightarrow \mathcal{H}$ be a non-linear mapping from d dimensional space into a dot product space $\mathcal{H}$. A non-linear dictionary can be trained in the feature space $\mathcal{H}$. Using the same notations in (8), we formulate the kernel dictionary learning as the following optimization problem

$$\begin{aligned}(\mathbf{U}_{n_l}^{(t+1),i}, \mathbf{\Lambda}_{n_l}^{(t+1),i}) &\triangleq (\mathbf{U}, \mathbf{\Lambda}) \\ &= \underset{\hat{\mathbf{U}},\hat{\mathbf{\Lambda}}}{\operatorname{argmin}} \sum_m w_m \|\mathbf{\Phi}(\mathbf{y}_m) - \mathbf{\Phi}(\mathbf{y}_m)\hat{\mathbf{U}}\hat{\boldsymbol{\lambda}}_m\|_2^2 \\ &\quad \text{subject to } \|\hat{\boldsymbol{\lambda}}_m\|_0 \leq T_0, \quad \forall m \\ &= \underset{\hat{\mathbf{U}},\hat{\mathbf{\Lambda}}}{\operatorname{argmin}} \|(\mathbf{\Phi}(\mathbf{Y}) - \mathbf{\Phi}(\mathbf{Y})\hat{\mathbf{U}}\hat{\mathbf{\Lambda}})\mathbf{W}\|_F^2, \\ &\quad \text{subject to } \|\hat{\boldsymbol{\lambda}}_m\|_0 \leq T_0, \quad \forall m, \end{aligned} \tag{15}$$

where

$$\mathbf{\Phi}(\mathbf{Y}) = [\mathbf{\Phi}(\mathbf{y}_1), \mathbf{\Phi}(\mathbf{y}_2), \dots, \mathbf{\Phi}(\mathbf{y}_{N(i,n_l)})],$$

$\hat{\boldsymbol{\lambda}}_m$ are the columns of $\hat{\mathbf{\Lambda}}$ and $\mathbf{W}$ is a square weighting matrix with its diagonal filled with $\{\sqrt{w_m}\}_{m=1}^{N(i,n_l)}$, and zeros elsewhere. In (15), since the dictionary lies in the linear span of the samples $\Phi(\mathbf{Y})$, we have used the following model for the dictionary in the feature space,

$$\mathbf{\Phi}(\mathbf{D}) = \mathbf{\Phi}(\mathbf{Y})\hat{\mathbf{U}},$$

where $\hat{\mathbf{U}} \in \mathbb{R}^{d \times K_0}$ is a matrix with K_0 atoms [28], [29]. This model provides adaptivity via modification of the matrix $\hat{\mathbf{U}}$. Through some algebraic manipulations, the cost function in (15) can be rewritten as

$$\begin{aligned}&\|(\mathbf{\Phi}(\mathbf{Y}) - \mathbf{\Phi}(\mathbf{Y})\hat{\mathbf{U}}\hat{\mathbf{\Lambda}})\mathbf{W}\|_F^2 \\ &\quad = \mathbf{tr}\left(((\mathbf{I} - \hat{\mathbf{U}}\hat{\mathbf{\Lambda}})\mathbf{W})^T \mathcal{K}(\mathbf{Y}, \mathbf{Y})(\mathbf{I} - \hat{\mathbf{U}}\hat{\mathbf{\Lambda}})\mathbf{W}\right), \end{aligned} \tag{16}$$

where $\mathcal{K}(\mathbf{Y}, \mathbf{Y})$ is a kernel matrix whose elements are computed from

$$\kappa(r, s) = \mathbf{\Phi}(\mathbf{y}_r)^T \mathbf{\Phi}(\mathbf{y}_s).$$

It is apparent that the objective function is feasible since it only involves a matrix of finite dimension $\mathcal{K} \in \mathbb{R}^{N(i,n_l) \times N(i,n_l)}$, instead of dealing with a possibly infinite dimensional dictionary.

An important property of this formulation is that the computation of $\mathcal{K}$ only requires dot products. Therefore, we are able to employ Mercer kernel functions to compute these dot products without carrying out the mapping $\mathbf{\Phi}$. Some commonly used kernels include polynomial kernels

$$\kappa(\mathbf{x}, \mathbf{y}) = (\langle \mathbf{x}, \mathbf{y} \rangle + a_1)^{a_2}$$

and Gaussian kernels

$$\kappa(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{y}\|^2}{a_3}\right),$$

where a_1, a_2 and a_3 are the parameters.

Similar to the the linear K-SVD [24] algorithm, the optimization of (15) involves sparse coding and dictionary update steps in the feature space which results in the kernel K-SVD algorithm [29]. Details of the optimization can be found in [29].

A. Computing Residuals in the Feature Space

Given kernel dictionary $\mathbf{\Phi(D)} = \mathbf{\Phi(Y)U}$ obtained by solving (15), we compute the residual e of each training sample $\mathbf{x}$ in the feature space as follows.[5]

$$e = \|\mathbf{\Phi(x)} - \mathbf{\Phi(D)}\overline{\mathbf{\Phi(D)}}\mathbf{\Phi(x)}\|_2^2, \tag{17}$$

where $\overline{\mathbf{\Phi(D)}}$ is the pseudo-inverse of $\mathbf{\Phi(D)}$ and computed as

$$\begin{aligned}\overline{\mathbf{\Phi(D)}} &= (\mathbf{\Phi(D)}^T\mathbf{\Phi(D)})^{-1}\mathbf{\Phi(D)}^T \\ &= (\mathbf{U}^T\mathcal{K}(\mathbf{Y},\mathbf{Y})^{-1}\mathbf{U})^{-1}\mathbf{U}^T\mathbf{\Phi(Y)}^T.\end{aligned}$$

Hence, it can be shown the residual e has the following close form

$$\mathcal{K}(\mathbf{x},\mathbf{x}) + \mathcal{K}(\mathbf{x},\mathbf{Y})((\mathcal{K}(\mathbf{Y},\mathbf{Y})^{-1})^T - 2\mathcal{K}(\mathbf{Y},\mathbf{Y})^{-1})\mathcal{K}(\mathbf{Y},\mathbf{x}), \tag{18}$$

where

$$\mathcal{K}(\mathbf{x},\mathbf{Y}) = \mathcal{K}(\mathbf{Y},\mathbf{x}) = [\mathcal{K}(\mathbf{y}_1,\mathbf{x}),\ldots,\mathcal{K}(\mathbf{y}_{N(i,n_l)},\mathbf{x})].$$

Using the result of (18), we update the confidence in each iteration for DLHD and DLSD via (2).

IV. Experiments

To evaluate the performance of our dictionary method, we performed two sets of experiments defined in [4] and [5]: inductive experiments and transductive experiments. We report the average test error rates (for inductive experiments) and the average labeling error rates (for transductive experiments), which were computed over 5 trials.

In an inductive experiment, samples are split in half into a training set and a test set. Each sample in the training set is ambiguously labeled according to controlled parameters, while each sample in the test set is unlabeled. In each trial, using the learned dictionaries from the training set, the test error rate is calculated as the ratio of the number of test samples that are erroneously labeled, to the total number of test samples. In a transductive experiment, all samples with ambiguous labels are used to train the dictionaries. In each trial, the labeling error rate is calculated as the ratio of the number of training samples that are erroneously labeled, to the total number of training samples.

Following the notations in [5], the controlled parameters are: p (proportion of ambiguously labeled samples), q (the number of extra labels for each ambiguously labeled sample) and ϵ (the degree of ambiguity - the maximum probability of an extra label co-occurring with a true label, over all labels and inputs [5]). We selected the following four datasets for the performance evaluations: Labeled Faces in the Wild (LFW) [20], CMU PIE dataset [21], TV series 'LOST' dataset [5] and the UMD dataset [22]. For the experiments implemented using the dictionary-based methods, we set the sparsity level T0 to be 5, and number of dictionary atoms per class to be 20.

[5]For simplicity and clarity, here we omitted superscripts and subscripts of the training samples, residuals and kernel dictionaries.

A. Labeled Faces in the Wild Dataset

The LFW database [20] was originally designed to address pair matching problems. Cropped and resized images of the LFW database were provided by the authors of [5]. In our experiment, we use one of the resulting subsets, FIW(10b), a balanced subset which contains the first 50 images for each of the top 10 most frequent subjects [5]. Fig. 3(a) shows this dataset, where faces of the same subject are shown in one row. We resized each image to 55×45 pixels,[6] and took the histogram equalized column-vector (2475×1) as input features. The size of dictionary per class is 2475×20.

Fig. 5(a) and (b) show average test error rates (for inductive experiments) of the proposed dictionary method (DLHD and DLSD) versus p and ϵ, respectively. For comparison, in the same figure we show the average test error rates of the other existing baseline methods[7] reported in [4] and [5]. Both dictionary methods are comparable to the Convex Learning from Partial Labels (CLPL) method (denoted as 'mean') [5]. Fig. 6 shows the average labeling error rates (for transductive experiments) versus q curves. The DLHD method outperforms the other compared methods when the number of extra labels is less than or equal to 5. For kernel dictionary learning, we denote the corresponding soft decision approach by KDLSD. We used Gaussian kernels in our experiments. It is observed that both soft decision approaches (DLSD and KDLSD) give better performance than the DLHD approach.

B. CMU PIE Dataset

The CMU PIE dataset [21] was designed for illumination challenges. The dataset contains 21 images under varying illumination conditions of 68 subjects. We took the first 18 subjects for our experiments and the resulting dataset is shown in Fig. 3(b), where each row presents images of the same subject under various illumination conditions. All images are resized to 48×40 and projected onto a 181-dimension subspace that is spanned by the 5th to the 185th eigenvectors obtained through the principle component analysis (PCA). The size of dictionary per class is 181×20. Figures 7(a) and (b) show the average labeling error rates versus p and q in transductive experiments. We compare the proposed method with the CLPL method (denoted as ('mean') and 'naive' methods) [4], [5][8] and two baseline methods (no dictionary learning - 'no DL', and standard K-SVD on DLSD - 'K-SVD SD'). Clearly, when either p or q is zero in transductive experiments, there exist no ambiguous labels and hence the labeling errors are zero. In Fig. 7(a), all compared

[6]We experimentally chose the feature dimension for our dictionary-based methods. Experiments have shown that the recognition accuracy does not degrade much when the image size is above 30×30 pixels. As a results, the performance of our dictionary-based methods (currently 55×45 pixels on the LFW dataset) is certainly able to remain the same with a higher dimension of 60×90 (i.e. the cropped image size used in [5] without PCA) as well.

[7]As definitions of these baselines can be found in [4] and [5], these definitions are not described again here due to space limitation.

[8]We obtained the code for CLPL ('mean') and 'naive' methods from http://www.timotheecour.com/. Both the 'naive' method and the normalized 'naive' method [1] give very similar results [5].

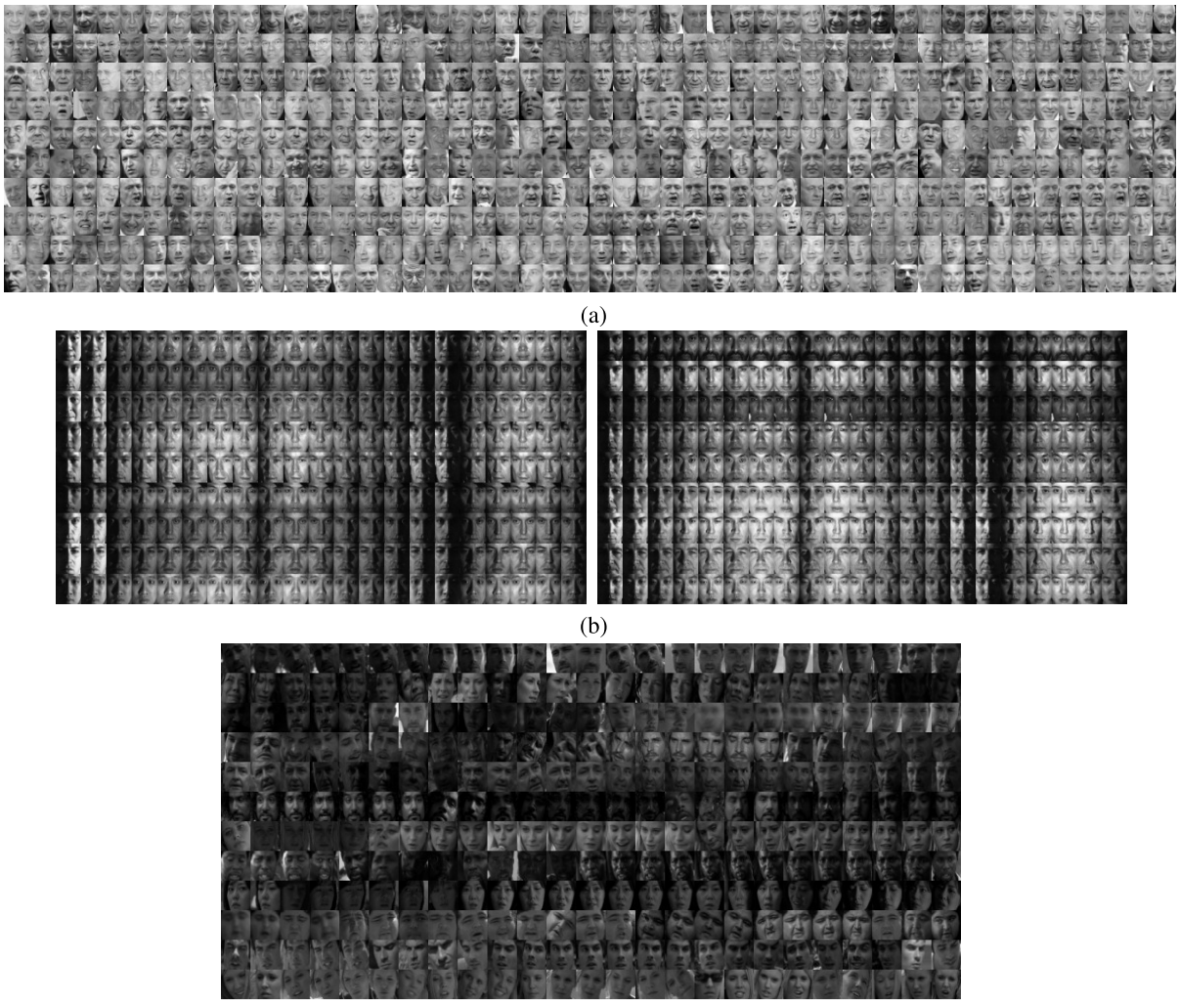


Fig. 3. (a) FIW(10b) 10-class dataset. (b) CMU PIE 18-class dataset - left: first 9 classes, right: second 9 classes. (c) TV series 'LOST' 12-class face dataset. In each dataset, face images belonging to the same class are shown in a row.

Fig. 4. Example frames from the UMD dataset. (a) Standing sequences. (b) Walking sequences. (c) Frames with blurred subjects due to the camera motion. Faces in standing sequences were sometimes non-frontal or partially occluded, while faces in walking sequences were frontal most of the time. Camera movements raise the additional difficulty for face tracking and recognition. The subjects were at a distance of several tens of meters from the camera.

methods provides good labeling performances. When 95% of samples are ambiguously labeled, the lowest average labeling error rate, 0.05%, is achieved by the DLSD approach. As shown in Fig. 7(b), both DLHD, DLSD and KDLSD outperform other compared methods for all numbers of extra labels.

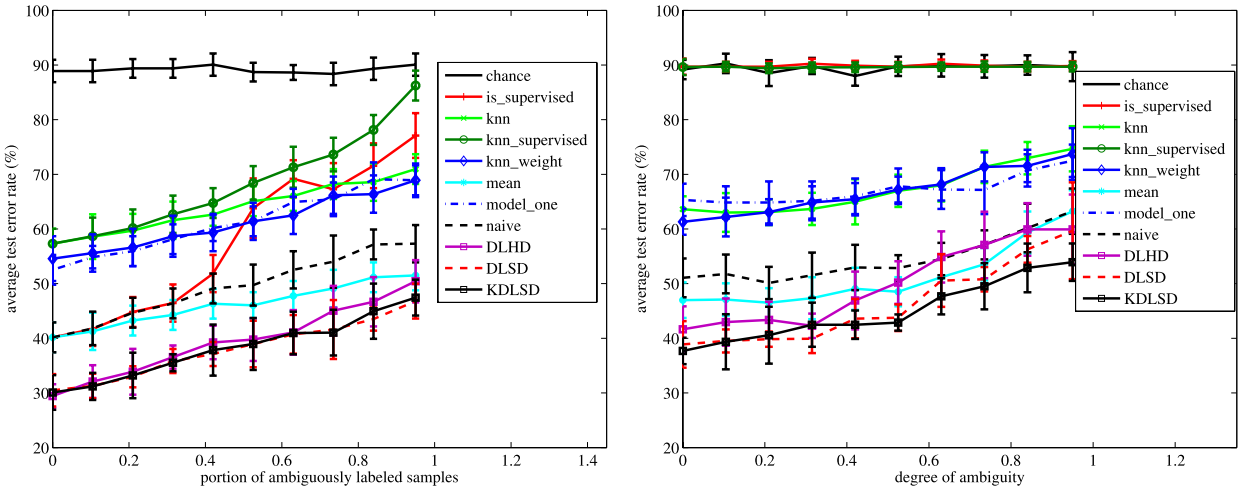


Fig. 5. Performance of the proposed dictionary methods and other baselines [4], [5] on the LFW dataset. (a) Average test error rates versus the proportion of ambiguously labeled samples ($p \in [0, 0.95]$, $q = 2$, inductive). (b) Average test error rates versus the degree of ambiguity for each ambiguously labeled sample ($p = 1, q = 1, \epsilon \in [1/(L-1), 1]$, inductive).

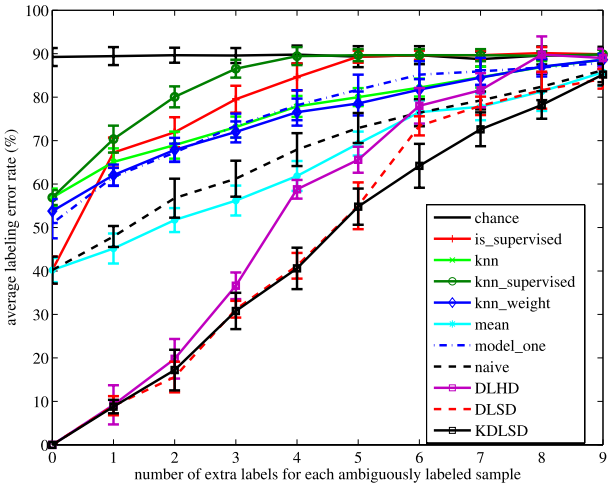


Fig. 6. Performance of the proposed dictionary methods and other baselines [4], [5] on the LFW dataset - average labeling error rates versus the number of extra labels for each ambiguously labeled sample ($p = 1, q \in [0, 1, \ldots, 9]$, transductive).

C. TV Series 'LOST' Dataset

We obtained cropped face images of TV series 'LOST' that were provided on-line by the authors of [5]. The original dataset contains 1122 registered face images across 14 subjects, and each subject contains from 18 up to 204 face images. In our experiment, we chose 12 subjects with at least 25 faces images per subject and for each chosen subject, we collected his/her first 25 face images. Fig. 3(c) shows the resulting dataset where faces from the same subject are shown in one row. We resized each image to 30×30 pixels, and took the histogram equalized column-vector (900×1) as input features. The size of dictionary per class is 900×20. Fig. 8 show the average labeling error rates versus p curves in transductive experiments. It is observed that when 95% of samples are ambiguously labeled, DLSD achieves the lowest labeling error rate, of 14.33%. KDLSD ranks the second, giving labeling error rate of 14.68%. On the other hand, for the overall performance averaged over p, KDLSD achieves the lowest labeling error rate of 6.07%, and it outperforms 6.32% given by DLSD.

D. UMD Video Dataset

The UMD video dataset [22] contains 12 videos recorded of a group of 16 subjects positioned several tens of meters from the camera. The videos were collected in a high definition format (1920×1080 pixels). They contain sequences of subjects standing without walking toward the camera, which we refer to as standing sequences, and sequence of each subject walking toward the camera, which we refer to as walking sequences. After segmenting the videos according to subjects and sequence types, we obtained 93 sequences in total: 70 standing sequences and 23 walking sequences. Figure 4(a) shows example frames from four different standing sequences, where most subjects are standing in a group. As some subjects were having conversations and others were looking elsewhere, their faces were sometimes non-frontal or partially occluded. Figure 4(b) shows example frames from four different walking sequences, in each of which a single subject was walking toward the camera, with a frontal face for most of the time. However, the walking subject's head sometimes turned to the right or left showing a profile face. Furthermore, for both types of sequences, the camera was not always static. Figure 4(c) shows example frames with blurred subjects due to the camera motion. Same as IV-C, we resized each image to 30×30 pixels, and took the histogram equalized column-vector (900×1) as input features. The size of dictionary per class is 900×20.

Fig. 9(a) shows the average labeling error rates versus p (with $q = 2$) curves in transductive experiments. It is observed that when 95% of samples are ambiguously labeled, KDLSD achieves the lowest labeling error rate, of 12.90%.

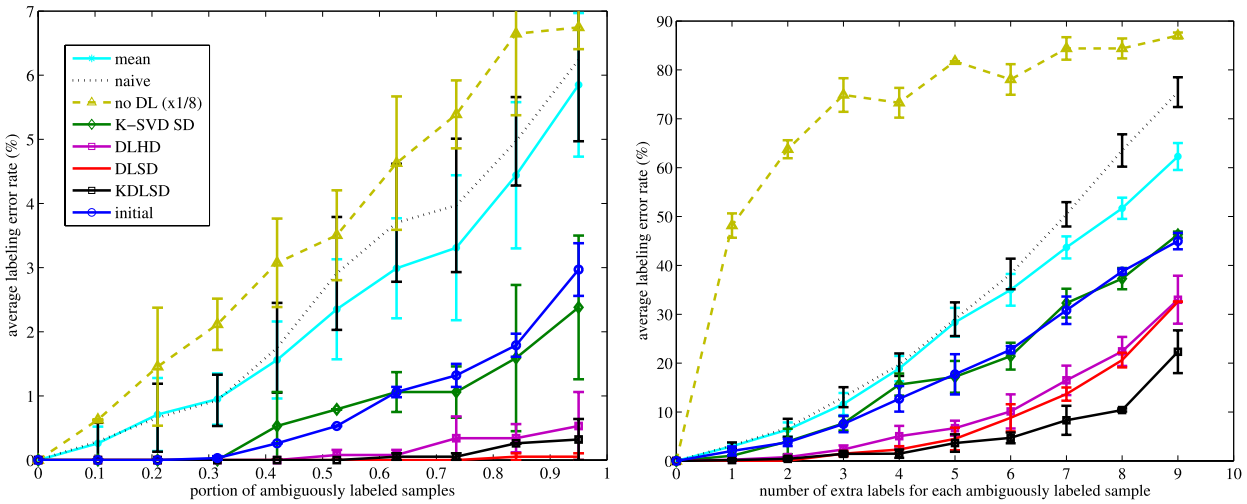


Fig. 7. Performance of the proposed dictionary methods, two baseline methods (no dictionary learning - 'no DL', and standard K-SVD on DLSD - 'K-SVD SD'), CLPL ('mean') and 'naive' methods [4], [5] on the PIE dataset. (a) Average labeling error rates versus the proportion of ambiguously labeled samples ($p \in [0, 0.95]$, $q = 2$). (b) Average labeling error rates versus the number of extra labels for each ambiguously labeled sample ($p = 1, q \in [0, 1, \ldots, 9]$).

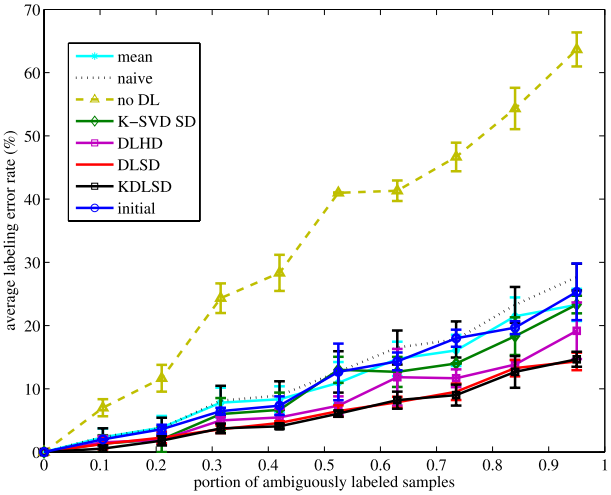


Fig. 8. Performance of the proposed dictionary methods, two baseline methods (no dictionary learning - 'no DL', and standard K-SVD on DLSD - 'K-SVD SD'), CLPL ('mean') and 'naive' methods [4], [5] on the LOST dataset - average labeling error rates versus the proportion of ambiguously labeled samples ($p \in [0, 0.95]$, $q = 2$).

Fig. 9(b) shows the average labeling error rates versus q (with $p = 1$) curves in transductive experiments. It is observed that when the number of extra labels is 9, KDLSD achieves the lowest labeling error rate, of 58.78%.

E. Discussions

To explain the performance gain of our dictionary learning approach, in plots of Figs. 7, 8 and 9, we show curves of two additional baseline methods: no dictionary learning ('no DL') and standard K-SVD on DLSD ('K-SVD SD') methods. The 'no DL' method utilizes features and ambiguous labels only, without learning dictionaries. This baseline collects for each class c, all its possible samples (i.e, $\mathbf{x}_i$'s with $p_{i,c}^{(t)} > 0$) at each iteration t, and uses them directly as a set of basis atoms. The 'K-SVD SD' method contrasts the DLSD method by simply using equal weights among possible samples of each label for dictionary learning. In other words, it ignores the weighting matrix $\mathbf{W}$ in (8) and learns dictionaries by the standard K-SVD algorithm. Reconstruction errors for both baseline methods are computed using the same L-2 norm as in (7) to update the confidence. As can be seen from these figures that the 'no DL' method was not able to obtain satisfactory results, while the 'K-SVD SD' method did not perform as well as DLHD and DLSD either. In particular, the performance degradation of the 'K-SVD SD' method highlights the importance of $\mathbf{W}$ computed from the DLSD method. Comparing DLHD and DLSD, we observe that DLHD performs not as well as the DLSD in that the hard-threshold confidence in DLHD is locally constrained, and hence it may not give the global optimal $\mathbf{W}$ for the dictionary learning. We further observed that the KDLSD outperforms the DLSD not for every case shown in Figs. 5 - 9. We experimentally set parameters of the Gaussian kernel for KDLSD, which may not be the optimal. This explains those few cases where KDLSD did not obtain better performance than the DLSD. In addition, while the state-of-the-art CLPL ('mean') method may be sensitive to face images with certain within-class variation due to illumination changes (e.g., in Fig. 3(b), (c)) and noise, the learned dictionary atoms in our method are able to account for these variations to some degree. Therefore, the performance of our dictionary-based approach is better than those of the CLPL ('mean') and other compared baseline methods.

When the proportion of ambiguously labeled is smaller than one, there are samples assigned to the correct label only. It is interesting to compare the proposed approaches to a semi-supervised method that ignores samples with extra ambiguous labels. We implemented a dictionary-based semi-supervised method in this setting. In particular, we used samples only with exactly one given label (based on our assumption, this label is the ground truth one) to learn the class specific dictionaries while ignoring the other samples with more than one labels. We then used the learned dictionaries to label all

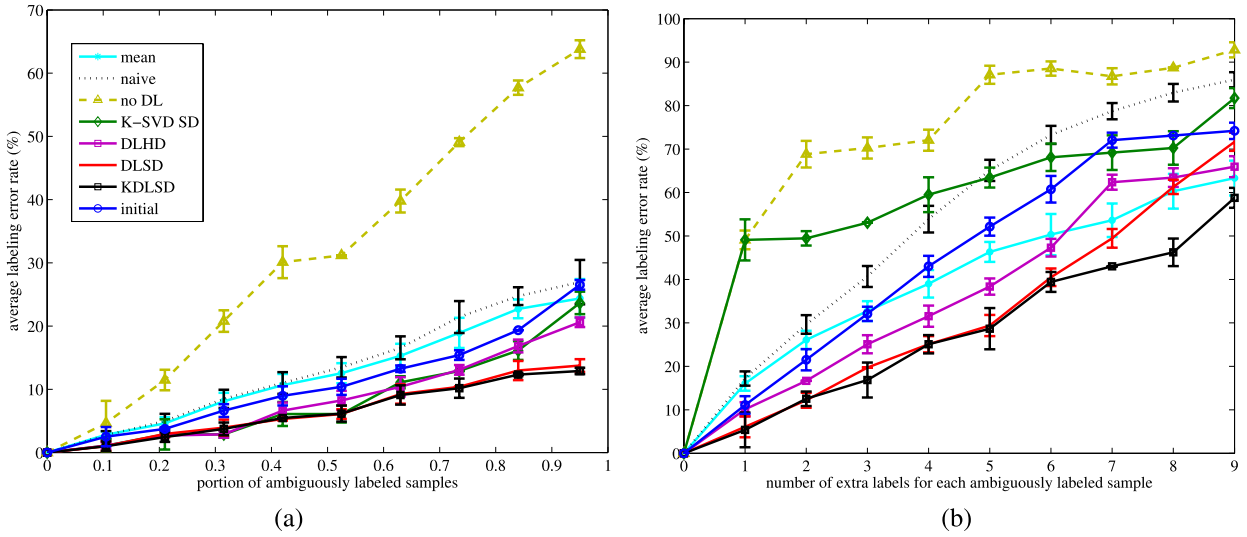


Fig. 9. Performance of the proposed dictionary methods, two baseline methods (no dictionary learning - 'no DL', and standard K-SVD on DLSD - 'K-SVD SD'), CLPL ('mean') and 'naive' methods [4], [5] on the UMD video. (a) Average labeling error rates versus the proportion of ambiguously labeled samples ($p \in [0, 0.95]$, $q = 2$). (b) Average labeling error rates versus the number of extra labels for each ambiguously labeled sample ($p = 1, q \in [0, 1, \ldots, 9]$).

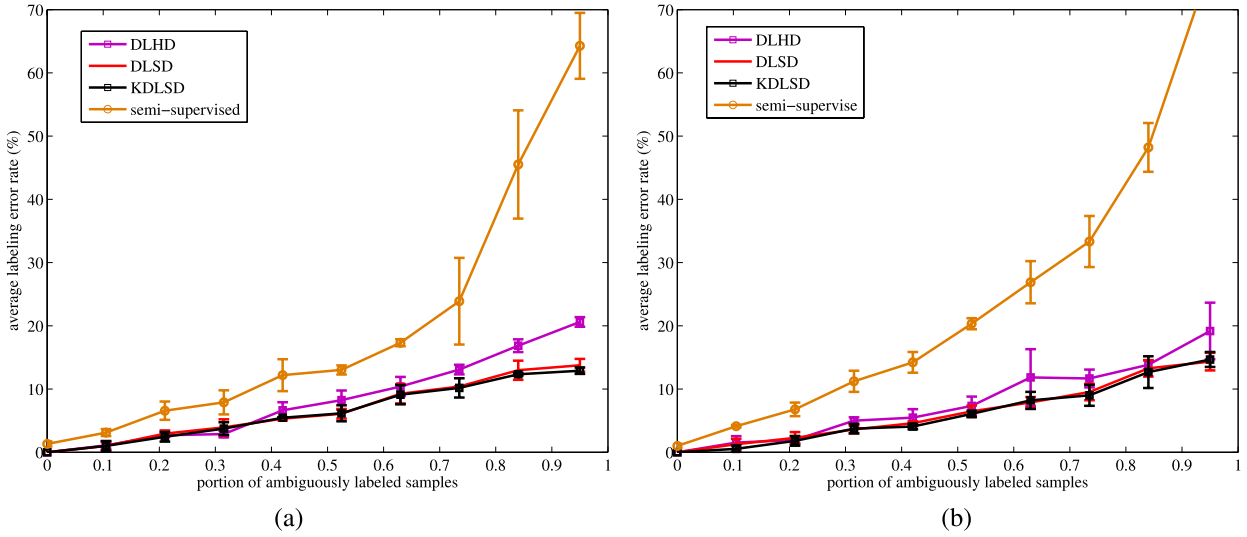


Fig. 10. Average labeling error rates versus the proportion of ambiguously labeled samples ($p \in [0, 0.95]$, $q = 2$) of our methods and a semi-supervised method on (a) the UMD video dataset and (b) the TV series "LOST" dataset.

samples based on the minimum distance criterion so that we can compute the average labeling error rates. Fig. 10 shows the average labeling error rates for $p < 1$ on the UMD video and TV series "LOST" datasets. The semi-supervised method is found to perform close to the proposed methods when p is low. On the other hand, when p is high, the gap becomes larger because the number of correctly labeled samples are insufficient to represent classes in a generative and discriminative way.

In Figs. 8, 7 and 9, we also included the performance curves of initial dictionary without subsequent iterations ('initial'). We can see the improvement obtained with the proposed iterative methods is significant. Initial dictionaries (not dictionaries that are randomly assigned) are important to the subsequent iterations. Based on good initial dictionaries with additional iterations, the proposed dictionary-based methods are able to boost the final performance. To further look into this behavior, we plot the average labeling errors over iterations in Fig. 11 for the proposed DLHD, DLSD and KDLSD methods on the UMD video when the portion of ambiguously labeled samples is $p = 0.42$, and the number of additional labels for each ambiguously labeled sample is $q = 2$. As can be seen from this figure, our errors decrease with increase in iterations and become stable after seven iterations.

Moreover, in order to examine the updates of the confidence matrices, in Fig. 12, we further show the initial (at $t = 0$) and updated (using DLSD at $t = 20$) confidence matrices corresponding to this experiment, where samples and labels are indexed vertically and horizontally, respectively. Without any prior knowledge, ambiguously labeled samples have equally probable initial confidences. At $t = 20$, we observe that the updated confidences for most samples tend to converge as they become impulse-shape where the confidence value is 1 for one label, and zero for the other labels.

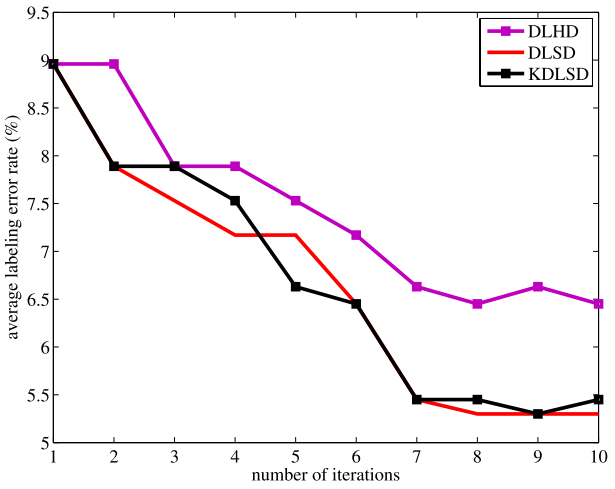


Fig. 11. Average labeling error rates versus the number of iterations of our methods: DLHD, DLSD and KDLSD ($p = 0.42, q = 2$) on the UMD video dataset. The error curves tend to be stable for more than 7 iterations.

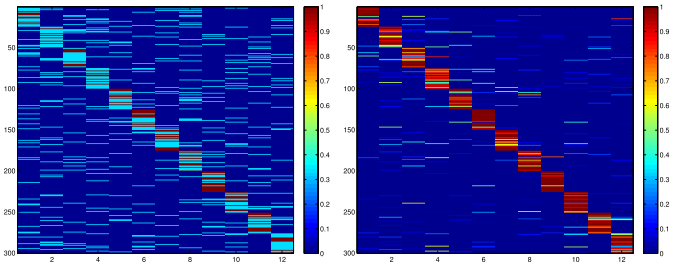


Fig. 12. Initial and updated confidence matrices on the TV series 'LOST' (12-class) dataset. (a) Initial confidence, $\mathbf{P}^{(0)}$. (b) $\mathbf{P}^{(20)}$ (using DLSD at $t = 20$).

The computation complexity of our current DLSD/KDLSD indeed increases with the number of training samples and the number of additional labels. Solving the scalability problem of DLSD/KDLSD is one of our future research directions. One possible solution to this problem is to consider online dictionary learning methods [30]. The other solution we consider is that, like DLHD, we first build class dictionaries (by applying weighted K-SVD algorithm on soft-thresholded classes). Next, prior to computing the confidence of sample x_i, we use the eigen-space updating algorithm proposed in [31] and [32] to dynamically update each of the $|L_i|$ class dictionaries such that the corresponding component of x_i itself under the dictionary's eigen-space is removed. Another possibility is that we optimize the number of dictionary atoms via the online singular value decomposition (SVD) algorithm when learning the class dictionaries. These dictionaries are then directly used to compute confidences for all x_i's. We consider this way because optimizing the number of dictionary atoms can improve the discriminative power of dictionaries.

V. Conclusion

Dictionary learning methods have been shown to be state-of-the-art in many supervised, unsupervised and semi-supervised classification problems. We have extended the dictionary learning to the case of ambiguously labeled learning, where each example is supplied with multiple labels, only one of which is correct. The proposed method iteratively estimates the confidence of samples belonging to each of the classes and uses it to refine the learned dictionaries. To enhance the performance, we further extended our work to handle the non-linearities in the data by learning kernel dictionaries. Experiments using four datasets demonstrate the improved accuracy of the proposed method compared to state-of-the-art ambiguously labeled learning techniques.

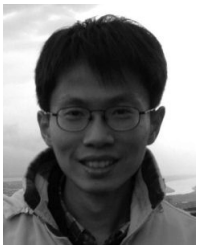
Acknowledgment

The authors would like to thank Dr. Jaishanker K. Pillai for his valuable discussions.

References

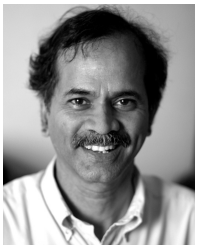
[1] R. Jin and Z. Ghahramani, "Learning with multiple labels," in *Proc. Neural Inf. Process. Syst. (NIPS)*, 2002, pp. 897–904.
[2] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Statist. Soc., Ser. B*, vol. 39, no. 1, pp. 1–38, 1977.
[3] A. Shrivastava, J. K. Pillai, V. M. Patel, and R. Chellappa, "Learning discriminative dictionaries with partially labeled data," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep./Oct. 2012, pp. 3113–3116.
[4] T. Cour, B. Sapp, C. Jordan, and B. Taskar, "Learning from ambiguously labeled images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 919–926.
[5] T. Cour, B. Sapp, and B. Taskar, "Learning from partial labels," *J. Mach. Learn. Res.*, vol. 12, pp. 1501–1536, May 2011.
[6] E. Hüllermeier and J. Beringer, "Learning from ambiguously labeled examples," *Intell. Data Anal.*, vol. 10, no. 5, pp. 419–439, Sep. 2006.
[7] J. Wright, A. Y. Yang, A. A. Ganesh, S. 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.
[8] J. K. Pillai, V. M. Patel, R. Chellappa, and N. K. Ratha, "Secure and robust iris recognition using random projections and sparse representations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 9, pp. 1877–1893, Sep. 2011.
[9] R. Rubinstein, A. M. Bruckstein, and M. Elad, "Dictionaries for sparse representation modeling," *Proc. IEEE*, vol. 98, no. 6, pp. 1045–1057, Jun. 2010.
[10] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. S. Huang, and S. Yan, "Sparse representation for computer vision and pattern recognition," *Proc. IEEE*, vol. 98, no. 6, pp. 1031–1044, Jun. 2010.
[11] V. M. Patel and R. Chellappa, "Sparse representations, compressive sensing and dictionaries for pattern recognition," in *Proc. 1st Asian Conf. Pattern Recognit. (ACPR)*, Nov. 2010, pp. 325–329.
[12] 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.
[13] K. Huang and S. Aviyente, "Sparse representation for signal classification," in *Proc. Neural Inf. Process. Syst. (NIPS)*, vol. 19. 2007, pp. 609–616.
[14] M. Ranzato, F. J. Huang, Y.-L. Boureau, and Y. LeCun, "Unsupervised learning of invariant feature hierarchies with applications to object recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2007, pp. 1–8.
[15] V. M. Patel, T. Wu, S. Biswas, P. J. Phillips, and R. Chellappa, "Dictionary-based face recognition under variable lighting and pose," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 3, pp. 954–965, Jun. 2012.
[16] Q. Qiu, V. Patel, and R. Chellappa, "Information-theoretic dictionary learning for image classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published.
[17] P. Sprechmann and G. Sapiro, "Dictionary learning and sparse coding for unsupervised clustering," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Mar. 2010, pp. 2042–2045.
[18] Y.-C. Chen, C. S. Sastry, V. M. Patel, P. J. Phillips, and R. Chellappa, "Rotation invariant simultaneous clustering and dictionary learning," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Mar. 2012, pp. 1053–1056.

[19] E. Elhamifar and R. Vidal, "Sparse subspace clustering," in *Proc. Conf. IEEE Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 2790–2797.
[20] G. B. Huang, V. Jain, and E. Learned-Miller, "Unsupervised joint alignment of complex images," in *Proc. IEEE 11th Int. Conf. Comput. Vis. (ICCV)*, Oct. 2007, pp. 1–8.
[21] T. Sim, S. Baker, and M. Bsat, "The CMU pose, illumination, and expression database," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 12, pp. 1615–1618, Dec. 2003.
[22] R. Chellappa, J. Ni, and V. M. Patel, "Remote identification of faces: Problems, prospects, and progress," *Pattern Recognit. Lett.*, vol. 33, no. 14, pp. 1849–1859, Oct. 2012.
[23] Y.-C. Chen, V. M. Patel, J. K. Pillai, R. Chellappa, and P. J. Phillips, "Dictionary learning from ambiguously labeled data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 353–360.
[24] 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. 11, pp. 4311–4322, Nov. 2006.
[25] J. A. Bilmes, "A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models," Dept. Elect. Eng. Comput. Sci., International Computer Science Inst., Berkeley, CA, USA, Tech. Rep. TR-97-021, Apr. 1998.
[26] F. Dellaert, "The expectation maximization algorithm," Georgia Inst. Technol., Atlanta, GA, USA, Tech. Rep. GIT-GVU-02-20, Feb. 2002.
[27] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer-Verlag, Oct. 2007.
[28] H. Van Nguyen, V. M. Patel, N. M. Nasrabadi, and R. Chellappa, "Design of non-linear kernel dictionaries for object recognition," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 5123–5135, Dec. 2013.
[29] H. V. Nguyen, V. M. Patel, N. M. Nasrabadi, and R. Chellappa, "Kernel dictionary learning," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Mar. 2012, pp. 2021–2024.
[30] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *Proc. Int. Conf. Mach. Learn.*, Montreal, QC, Canada, Jun. 2009, pp. 689–696.
[31] S. Chandrasekaran, B. S. Manjunath, Y. F. Wang, J. Winkeler, and H. Zhang, "An eigenspace update algorithm for image analysis," *Graph. Models Image Process.*, vol. 59, no. 5, pp. 321–332, Sep. 1997.
[32] B. S. Manjunath, S. Chandrasekaran, and Y. F. Wang, "An eigenspace update algorithm for image analysis," in *Proc. Int. Symp. Comput. Vis. (ISCV)*, Nov. 1995, pp. 551–556.



Yi-Chen Chen (S'08) received the B.S. degree in electrical engineering from National Tsing Hua University, Hsinchu, Taiwan, the M.S. degree in communication engineering from National Taiwan University, Taipei, Taiwan, the M.S. degree in electrical engineering from the University of Washington, Seattle, WA, USA, and the Ph.D. degree in electrical engineering from the University of Maryland, College Park, MD, USA. His research interests are in computer vision and pattern recognition, with primary focuses on face recognition, salient view selection, clustering, and content-based image retrieval. He also has research interests in communications.

Vishal M. Patel (M'01) is currently a Research Faculty Member with the University of Maryland Institute for Advanced Computer Studies, College Park, MD, USA. He received the B.S. (Hons.) degrees in electrical engineering and applied mathematics, and the M.S. degree in applied mathematics from North Carolina State University, Raleigh, NC, USA, in 2004 and 2005, respectively, and the Ph.D. degree in electrical engineering from the University of Maryland, College Park, in 2010. He was a recipient of the ORAU Postdoctoral Fellowship in 2010. His research interests are in signal processing, computer vision, and pattern recognition with applications to object recognition, biometrics, and imaging. He is a member of Eta Kappa Nu, Pi Mu Epsilon, and Phi Beta Kappa.



Rama Chellappa (S'78–M'79–SM'83–F'92) received the B.E. (Hons.) degree in electronics and communication engineering from the University of Madras, Chennai, India, the M.E. (Hons.) degree from the Indian Institute of Science, Bangalore, India, and the M.S.E.E. and Ph.D. degrees in electrical engineering from Purdue University, West Lafayette, IN, USA. From 1981 to 1991, he was a faculty member with the Department of Electrical Engineering-Systems, University of Southern California (USC), Los Angeles, CA, USA. Since 1991, he has been a Professor of Electrical and Computer Engineering and an Affiliate Professor of Computer Science with the University of Maryland (UMD), College Park, MD, USA. He is also affiliated with the Center for Automation Research and the Institute for Advanced Computer Studies (permanent member), and serves as the Chair of the Department of Electrical and Computer Engineering. In 2005, he was named as a Minta Martin Professor of Engineering. His current research interests span many areas in image processing, computer vision, and pattern recognition. He was a recipient of the NSF Presidential Young Investigator Award, four IBM Faculty Development Awards, two Paper Awards, the K.S. Fu Prize from the International Association of Pattern Recognition, the Society, Technical Achievement, and Meritorious Service Awards from the IEEE Signal Processing Society, the Technical Achievement and Meritorious Service Awards from the IEEE Computer Society, and the Excellence in Teaching Award from the School of Engineering at USC. At UMD, he received the college and university level recognitions for research, teaching, innovation, and mentoring undergraduate students. In 2010, he was recognized as an Outstanding Electrical and Computer Engineer by Purdue University. He served as the Editor-in-Chief of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, and the General and Technical Program Chair/Cochair of several IEEE international and national conferences and workshops. He is a Golden Core Member of the IEEE Computer Society, and served as a Distinguished Lecturer of the IEEE Signal Processing Society and the President of IEEE Biometrics Council. He is a fellow of the International Association for Pattern Recognition, the Optical Society of America, the American Association for the Advancement of Science, and the Association for Computing Machinery, and holds four patents.

P. Jonathon Phillips (M'96–SM'06–F'10) is currently a leading Researcher in computer Vision, Biometrics, Face Recognition, and Human Identification. He is with the National Institute of Standards and Technology (NIST), Gaithersburg, MD, USA, where he works on designing grand challenges for advancing face recognition and visual biometric technology and science. His previous efforts include the Iris Challenge Evaluations, the Face Recognition Vendor Test (FRVT) 2006, the Face Recognition Grand Challenge, and the Face Recognition Technology. From 2000 to 2004, he was assigned to the Defense Advanced Projects Agency, Arlington, VA, USA, as a Program Manager of the Human Identification at a Distance Program. He was the Test Director of the FRVT 2002. For his work on the FRVT 2002, he was awarded the Department of Commerce Gold Medal. His work has been reported in print media of record including the *New York Times* and *Economist.* He has appeared on NPR's Science Friday. Prior to joining NIST, he was with the U.S. Army Research Laboratory, Adelphi, MD, USA. He received the Ph.D. degree in operations research from Rutgers University, New Brunswick, NJ, USA. From 2004 to 2008, he was an Associate Editor of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE and a Guest Editor of an issue of the PROCEEDINGS OF THE IEEE on biometrics. In an Essential Science Indicators analysis of face recognition publication over the past decade, his work ranks at #2 by total citations and #1 by cites per paper. He received the inaugural Mark Everingham Prize. He is a fellow the International Association for Pattern Recognition.