Dictionary Learning from Ambiguously Labeled Data

Yi-Chen Chen, Vishal M. Patel, Jaishanker K. Pillai, Rama Chellappa * and P. Jonathon Phillips †

Abstract

We propose a novel 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: a confidence update and 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 (EM-based) or hard decision rules. Extensive evaluations on existing datasets demonstrate that the proposed method performs significantly better than state-of-the-art ambiguously labeled learning approaches.

1. 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 are in the picture, the reader may not know which face goes with the names in the caption. The problem of learning identities where each example is associated with multiple labels, when only one of which is correct is often known as ambiguously labeled learning.

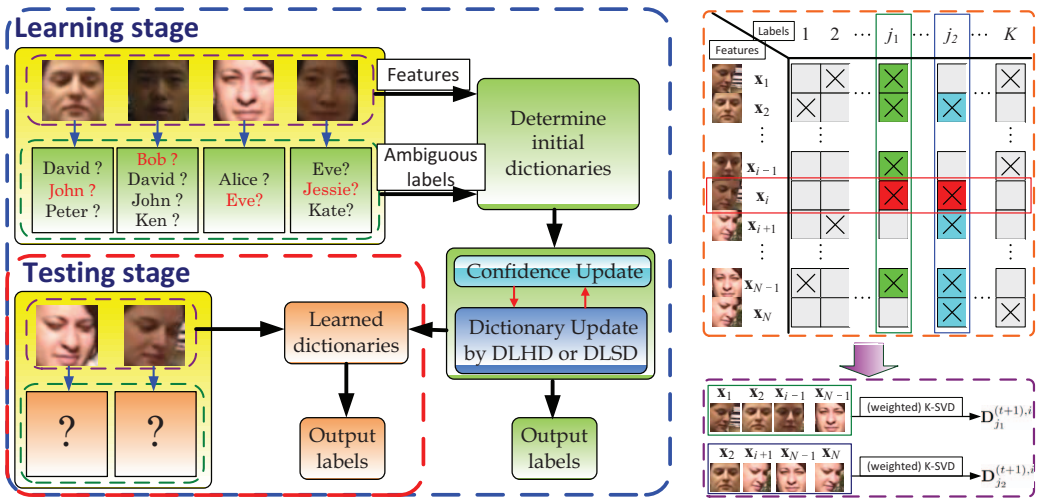
Several papers have been published in the literature that address the ambiguous label problem. In [13], a discriminative framework was proposed based on the Expectation Maximization (EM) algorithm [8], with a maximum likelihood approach to disambiguate the correct labels from incorrect ones. A semi-supervised dictionary-based learning method was proposed in [18] under the formulation where 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 [5] and [6], a method was presented to determine the label using a multilinear classifier that minimizes a convex loss function. The loss function used in [5] and [6] was shown to be a tighter convex upper bound on the 0/1 loss function 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 [12].

In recent years, sparse and redundant signal representations have generated interest in image processing, vision and machine learning communities. 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 $\mathbf{x}$ is sparse in dictionary $\mathbf{D}$ if it can be approximated by $\mathbf{x} = \mathbf{D}\mathbf{t}$, where $\mathbf{t}$ is a sparse vector and $\mathbf{D}$ is a dictionary that contains atoms as its columns. 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 [15], [11], [17], [14], [16], semi-supervised [18] and unsupervised [20], [4], [9] 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. 1(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. 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.

*Yi-Chen Chen, Vishal M. Patel, Jaishanker K. Pillai and Rama Chellappa are with the Department of Electrical and Computer Engineering and the Center for Automation Research, UMIACS, University of Maryland, College Park, MD. {chenyc08, pvishalm, jsp, rama}@umiacs.umd.edu

†P. Jonathon Phillips is with National Institute of Standards and Technology, Gaithersburg, MD. jonathon.phillips@nist.gov



(a) (b)

Figure 1. The proposed dictionary learning method. (a) Block diagram. (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.

Our paper makes the following contributions:

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 show that our dictionary learning with soft decision rule is an EM–based dictionary learning method.

4. We propose a weighted K–SVD [1] algorithm to weigh the importance of samples according to their confidences during the learning process.

2. 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

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

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 $\mathbf{C} = [\mathbf{C}_1, \mathbf{C}_2, \cdots, \mathbf{C}_K]$ 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 $\mathbf{D} = [\mathbf{D}_1, \mathbf{D}_2, \cdots, \mathbf{D}_K]$ 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.

2.1. The Dictionary Learning Hard Decision approach

The dictionary learning hard decision (DLHD) approach learns dictionaries directly from class matrices[1], $\{\mathbf{C}_i\}_{i=1}^K$, 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.

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 at 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

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

$\mathbf{x}_i$ 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

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

where $\beta_j^{(t)}$ and $\sigma_j^{(t)}$ are parameters (given in section 2.3), and

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

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 2.3, 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.

Cluster Update:[2] 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,

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

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. Many approaches have been proposed in the literature for solving such optimization problem. In this paper, we adapt the K-SVD algorithm [1] 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 $\mathbf{C}_j^{(t+1)}$. Then, the dictionary is updated atom-by-atom in an efficient way. The entire approach for learning dictionaries from ambiguously labeled data using hard decisions is summarized in Algorithm 1.

[2]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 $e_{ij}^{(t)}$ using (3). Then use $e_{ij}^{(t)}$ 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.

2.2. 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 $l \neq i$. In what follows, we describe the different steps of the algorithm.

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 $e_{ij_l}^{(t),i}$ using $\mathbf{D}_{j_l}^{(t),i}$ for all j_l in L_i as

$$e_{ij_l}^{(t),i} = \|\mathbf{x}_i - \mathbf{D}_{j_l}^{(t),i}\overline{\mathbf{D}_{j_l}^{(t),i}}\mathbf{x}_i\|_2^2. \tag{6}$$

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

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 $L_i = \{j_1, j_2, \dots j_{|L_i|}\}$. In particular, we learn $\mathbf{D}^{(t+1),i} = [\mathbf{D}_{j_1}^{(t+1),i}|\mathbf{D}_{j_2}^{(t+1),i}|\dots|\mathbf{D}_{j_{|L_i|}}^{(t+1),i}]$, where

each $\mathbf{D}_{j_l}^{(t+1),i}$ is built using the soft decision rules and samples $\mathbf{x}_l$'s, where $l \neq i$ and $p_{l,j_l}^{(t+1)} > 0$. Fig. 1(b) shows an example of how these common ambiguous label samples are collected to learn the intermediate dictionaries $\mathbf{D}_{j_l}^{(t+1),i}$. 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 $\mathbf{D}_{j_1}^{(t+1),i}$ and $\mathbf{D}_{j_2}^{(t+1),i}$, 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 $\{\mathbf{x}_{i_m}\}_{m=1}^{N(i,j_l)}$ be the collection of these samples. Its matrix form is denoted by $\mathbf{Y} = [\mathbf{y}_1\ \mathbf{y}_2 ... \ \mathbf{y}_{N(i,j_l)}]$, where $\mathbf{y}_m$, $m \in \{1,\ldots,N(i,j_l)\}$, is a column vectorized form of some collected sample $\mathbf{x}_{i_m}$. Let $\mathbf{w} = [w_1\ w_2 ... \ w_{N(i,j_l)}] = [p_{i_1,j_l}^{(t)}\ p_{i_2,j_l}^{(t)} ... \ p_{i_{N(i,j_l)},j_l}^{(t)}]$, 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

$$
\begin{aligned}
[\mathbf{D}_{j_l}^{(t+1),i}\ \mathbf{\Gamma}_{j_l}^{(t+1),i}] &= \underset{\mathbf{D},\mathbf{\Gamma}}{\operatorname{argmin}} \sum_{m=1}^{N(i,j_l)} w_m \|\mathbf{y}_m - \mathbf{D}\boldsymbol{\gamma}_m\|_2^2, \\
&\quad \text{subject to } \|\boldsymbol{\gamma}_m\|_0 \leq T_0, \forall m, \\
&= \underset{\mathbf{D},\mathbf{\Gamma}}{\operatorname{argmin}} \|(\mathbf{Y} - \mathbf{D}\mathbf{\Gamma})\mathbf{W}\|_F^2, \\
&\quad \text{subject to } \|\boldsymbol{\gamma}_m\|_0 \leq T_0, \forall m, \qquad (7)
\end{aligned}
$$

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

- *Sparse Coding Stage*: For $m = 1, 2, \ldots, N(i,j_l)$, compute $\boldsymbol{\gamma}_m$ for $\mathbf{y}_m$ by solving

$$\min_{\boldsymbol{\gamma}} \|(\mathbf{y}_m - \mathbf{D}\boldsymbol{\gamma})\sqrt{w_m}\|_2^2, \text{ subject to } \|\boldsymbol{\gamma}\|_0 \leq T_0.$$

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

Algorithm 2: Iteratively learning dictionaries using soft decision and updating confidence.

Input: Training samples $\mathcal{L} = \{(x_i, L_i)\}$.
Output: Dictionary $\mathbf{D}^* = [\mathbf{D}_1^*|\mathbf{D}_2^*|\ldots|\mathbf{D}_K^*]$.
Algorithm:

1. Repeat the following iterations to refine confidence until the maximum iteration number $\mathrm{T_c}$ is reached:
   - 1.1 **Confidence Update**: Use (6) to calculate residuals $e_{ij_l}^{(t),i}$, $\forall j_l \in L_i$. Then, use $e_{ij_l}^{(t)}$ to update confidence $p_{i,j_l}^{(t)}$ 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 $L_i = \{j_1, j_2, \ldots j_{|L_i|}\}$: Construct the weighting matrix $\mathbf{W}$ and use (7) to build $\mathbf{D}_{j_l}^{(t+1),i}$ from which the dictionary
   $$\mathbf{D}^{(t+1),i} = [\mathbf{D}_{j_1}^{(t+1),i}|\mathbf{D}_{j_2}^{(t+1),i}|\ldots|\mathbf{D}_{j_{|L_i|}}^{(t+1),i}]$$
   is obtained.
2. When $t = \mathrm{T_c}$, follow 1.2 and 1.3 in Algorithm 1 to build the final dictionary $\mathbf{D}^* = \mathbf{D}_c^{(\mathrm{T_c})}$.

2.3. DLSD is an EM-based approach

The proposed DLSD is indeed an EM [2][7][3] 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

$$E\left[\log p(\{\mathbf{x}_l\}_{l=1,l\neq i}^N, \{Z_l\}_{l=1,l\neq i}^N|\mathbf{D}^i)|\mathbf{x}_i, \mathbf{D}^{(t),i}\right], \qquad (8)$$

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 (8) equals

$$\sum_{j=1}^K \sum_{l=1,l\neq i}^N p_{l,j}^{(t)} \left(\log(\alpha_j) + \log(p_j(\mathbf{x}_l|\mathbf{D}_j^i))\right), \qquad (9)$$

where

$$p_{l,j}^{(t)} \triangleq p(Z_l = j|\mathbf{x}_l, \mathbf{D}^{(t),i}) = \frac{\alpha_j p_j(\mathbf{x}_l|\mathbf{D}_j^{(t),i})}{\sum_{k=1}^K \alpha_k p_k(\mathbf{x}_l|\mathbf{D}_k^{(t),i})}. \qquad (10)$$

In the M–step, we maximize (9) by finding $\boldsymbol{\alpha}^{(t+1)} \triangleq [\alpha_1^{(t+1)}, ..., \alpha_K^{(t+1)}]$ and $\mathbf{D}^{(t+1),i} = [\mathbf{D}_1^{(t+1),i}|...|\mathbf{D}_K^{(t+1),i}]$

such that

$$\boldsymbol{\alpha}^{(t+1)} = \underset{\boldsymbol{\alpha}=[\alpha_1,\alpha_2,\ldots,\alpha_K]}{\operatorname{argmax}} \sum_{j=1}^{K}\sum_{l=1,l\neq i}^{N} p_{l,j}^{(t)} \log(\alpha_j),$$

$$= \underset{\alpha_j}{\operatorname{argmax}} \sum_{l=1,l\neq i}^{N} p_{l,j}^{(t)} \log(\alpha_j), \forall j, \tag{11}$$

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

The optimization problem in (12) can be solved by the weighted K-SVD algorithm in (7). $\sigma_{j_l}^{(t+1)}$ can be approximated by the average residual over $\{\mathbf{y}_m\}_{m=1}^{N(i,j_l)}$. That is, $\sigma_{j_l}^{(t+1)} = \frac{1}{\eta(i,j_l)}\sum_{m=1}^{N(i,j_l)} w_m \|\mathbf{y}_m - \mathbf{D}_{j_l}^{(t+1),i}\boldsymbol{\gamma}_m\|_2^2, \forall j_l \in L_i$, where $\eta(i,j_l) = \sum_{m=1}^{N(i,j_l)} w_m$. Moreover, as α_{j_l} sums to one over j_l, (11) leads to $\alpha_{j_l}^{(t+1)} = \frac{\eta(i,j_l)}{N(i,j_l)}$. We then compute $\beta_{j_l}^{(t+1)} = \frac{\alpha_{j_l}^{(t+1)}}{\sqrt{2\pi\sigma_{j_l}^{(t+1)}}}$, and update $p_{i,j_l}^{(t+1)}$ by (2).

2.4. 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 $\mathbf{D}^{(0),i} = [\mathbf{D}_{j_1}^{(0),i}|\mathbf{D}_{j_2}^{(0),i}|\ldots|\mathbf{D}_{j_{|L_i|}}^{(0),i}]$, where the intermediate dictionary $\mathbf{D}_{j_k}^{(0),i}$ is learned from samples other than $\mathbf{x}_i$ with ambiguous label $j_k \in L_i$. These samples are collected in the same way as described in section 2.2. Next, $\mathbf{x}_i$ is assigned to class $\hat{j}^i$ such that it gives the lowest residual. In other words,

$$\hat{j}^i = \underset{j_k\in L_i}{\operatorname{argmin}} \|\mathbf{x}_i - \mathbf{D}_{j_k}^{(0),i}\overline{\mathbf{D}_{j_k}^{(0),i}}\mathbf{x}_i\|_2^2. \tag{13}$$

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 [1]. We summarize this initialization approach in Algorithm 3.

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 $\mathbf{D}^{(0)} = [\mathbf{D}_1^{(0)}|\mathbf{D}_2^{(0)}|\ldots|\mathbf{D}_K^{(0)}]$.
Algorithm:

1. Initialization: $i \leftarrow 1$; $\mathbf{C}_j^{(0)} \leftarrow \{\}, \forall j \in \{1,2,\ldots,K\}$.
2. Repeat the following for every $\mathbf{x}_i$:
   - 2.1 Construct $\mathbf{D}^{(0),i} = [\mathbf{D}_{j_1}^{(0),i}|\mathbf{D}_{j_2}^{(0),i}|\ldots|\mathbf{D}_{j_{|L_i|}}^{(0),i}]$, where $\mathbf{D}_{j_k}^{(0),i}$ 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 (13).
3. Establish initial dictionaries $\mathbf{D}^{(0)} = [\mathbf{D}_1^{(0)}|\mathbf{D}_2^{(0)}|\ldots|\mathbf{D}_K^{(0)}]$, where $\mathbf{D}_j^{(0)}$ is learned from $\mathbf{C}_j^{(0)}$ using the K-SVD algorithm.

Note that our method is very different from the approach of learning dictionaries from partially labeled data [18]. The work in [18] learns class discriminative dictionaries while our work learns class reconstructive dictionaries. In addition, from the formulation in [18] 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 [18] and [20] (for totally unlabeled samples) are special cases of the ambiguously labeled formulation presented in this paper.

3. Experiments

To evaluate the performance of our dictionary method, we performed two sets of experiments defined in [5][6]: 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 [6], 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 [6]). We selected the following three datasets for the performance evaluations: Labeled Faces in the Wild (LFW) [10], CMU PIE dataset [19] and TV series 'LOST' dataset [6].

3.1. Labeled Faces in the Wild dataset

The LFW database [10] was originally designed to address pairwise matching problems. Cropped and resized images of the LFW database were provided by the authors of [6]. 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 [6]. Fig. 2(a) shows this dataset, where faces of the same subject are shown in one row. We resized each image to 55×45 pixels, and took the histogram equalized column-vector (2475×1) as input features. Figures 3(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 [3] reported in [5], [6]. Both dictionary methods are comparable to the Convex Learning from Partial Labels (CLPL) method (denoted as 'mean') [6]. Fig. 3(c) 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. The DLSD approach gives slightly better performance than the DLHD approach.

3.2. CMU PIE dataset

The PIE dataset was designed for addressing illumination and pose 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. 2(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). Figures 4(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) [5], [6] [4]. 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. 4(a), all compared methods provides good labeling performances. When 95% of samples are ambiguously labeled, the lowest average error labeling rate, 0.05%, is achieved by the DLSD approach. As shown in Fig. 4(b), both DLHD and DLSD outperform other compared methods for all values of extra labels.

3.3. TV series 'LOST' dataset

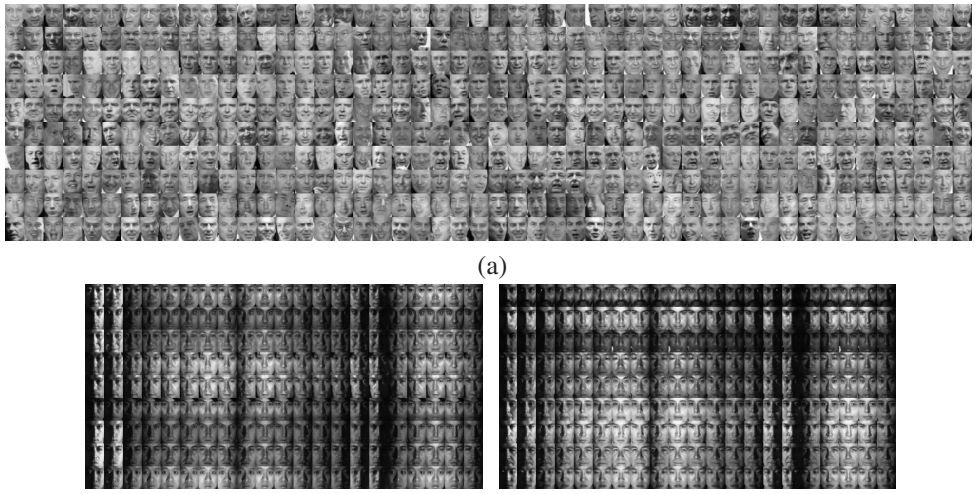
We obtained the cropped face images of TV series 'LOST' provided by the authors of [6]. The original dataset contains 1122 registered face images across 14 subjects, and each subject contains from 18 up to 204 face images. In our experiment, we chose 12 subjects with at least 25 faces images per subject and for each chosen subject, we collected his/her first 25 face images. We resized each image to 30×30 pixels, and took the histogram equalized column-vector (900×1) as input features. Fig. 4(c) 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 error labeling rate, of 14.33%.

3.4. Discussions

To explain the performance gain of our dictionary learning approach, in Fig. 4, we show curves of two additional baseline methods: 'no dictionary learning (DL)' and 'equally-weighted K-SVD'. 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 'equally-weighted K-SVD' 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 weight matrix $\mathbf{W}$ in (7) 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 (6) to update the confidence. These figures show that the 'no DL' method was not able to obtain satisfactory results. The 'equally-weighted K-SVD' method did not perform as well as DLHD and DLSD. In particular, the performance degradation of the 'equally-weighted K-SVD' method highlights the importance of $\mathbf{W}$ computed from the DLSD method.

[3] As definitions of these baselines can be found in [5], [6], these definitions are not described again here due to space limitation.

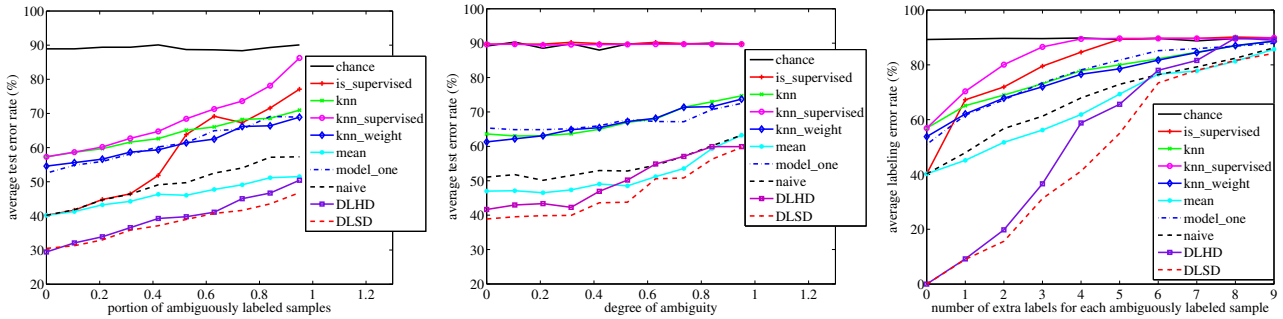
[4] We obtained the code for CLPL ('mean') and 'naive' methods from http://www.timotheecour.com/. Both the 'naive' method and the normalized 'naive' method [13] give very similar results [6].



(a)

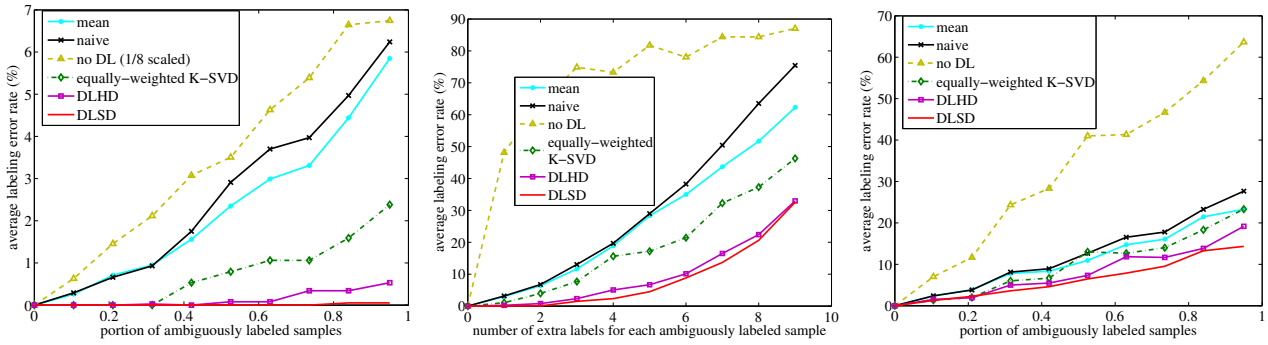
(b)

Figure 2. (a) FIW(10b) 10–class dataset. (b) CMU PIE 18–class dataset –left: first 9 classes, right: second 9 classes. In each dataset, face images belonging to the same class are shown in a row.



(a) (b) (c)

Figure 3. Performance of the proposed dictionary methods and other baselines [5], [6] 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). (c) Average labeling error rates versus the number of extra labels for each ambiguously labeled sample ($p = 1, q \in [0, 1, ..., 9]$, transductive). The proposed dictionary methods are comparable to the CLPL method ('mean').

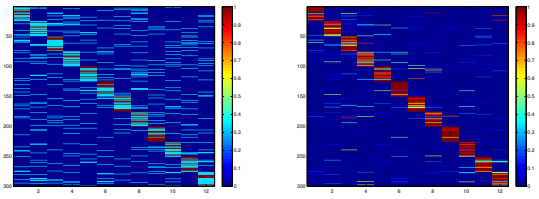


(a) (b) (c)

Figure 4. Performance of the proposed dictionary methods, two baseline methods (no dictionary learning –'no DL', and standard K–SVD – 'equally–weighted K–SVD'), CLPL ('mean') and 'naive' methods [5], [6] on transductive experiments. (a) and (c) Average labeling error rates versus the proportion of ambiguously labeled samples ($p \in [0, 0.95]$, $q = 2$) on the PIE and LOST datasets, respectively. (b) Average labeling error rates versus the number of extra labels for each ambiguously labeled sample ($p = 1, q \in [0, 1, ..., 9]$) on the PIE dataset.

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. 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. 2(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.

Moreover, in order to examine the updates of the confidence matrices, in Fig. 5, 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.



(a) (b)

Figure 5. 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 $t = 20$).

4. Conclusion

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. Experiments using three publicly available datasets demonstrate the improved accuracy of the proposed method compared to state-of-the-art ambiguously labeled learning techniques.

Acknowledgment

This work was partially supported by a Co-operative Agreement from the National Institute of Standards and Technology (NIST) under the Grant 70NANB11H023. PJP was supported by the Federal Bureau of Investigation. The identification of any commercial product or trade name does not imply endorsement or recommendation by NIST.

References

[1] M. Aharon, M. Elad, and A. M. Bruckstein. The K-SVD: an algorithm for designing of overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, 2006. 2, 3, 5

[2] J. A. Bilmes. A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. *ICSI and U.C. Berkeley, TR-97-021*, April, 1998. 4

[3] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, October, 2007. 4

[4] Y.-C. Chen, C. S. Sastry, V. M. Patel, P. J. Phillips, and R. Chellappa. Rotation invarient simultaneous clustering and dictionary learning. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2012. 1

[5] T. Cour, B. Sapp, C. Jordan, and B. Taskar. Learning from ambiguously labeled images. *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 919–926, June 2009. 1, 5, 6, 7

[6] T. Cour, B. Sapp, and B. Taskar. Learning from partial labels. *Journal of Machine Learning Research (JMLR)*, 12:1225–1261, 2011. 1, 5, 6, 7

[7] F. Dellaert. The expectation maximization algorithm. *Georgia Institute of Technology, GIT-GVU-02-20*, February, 2002. 4

[8] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B*, 39(1):1–38, 1977. 1

[9] E. Elhamifar and R. Vidal. Sparse subspace clustering. *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 2790–2797, June 2009. 1

[10] G. Huang, V. Jian, and E. Learned-Miller. Unsupervised joint alignment of complex images. *International Conference on Computer Vision (ICCV)*, 2007. 6

[11] K. Huang and S. Aviyente. Sparse representation for signal classification. *Proceedings of Neural Information Processing Systems (NIPS)*, 19:609–616, 2007. 1

[12] E. Hüllermeier and J. Beringer. Learning from ambiguously labeled examples. *Intell. Data Anal.*, 10(5):419–439, Sept. 2006. 1

[13] R. Jin and Z. Ghahramani. Learning with multiple labels. *Proceedings of Neural Information Processing Systems (NIPS)*, pages 897–904, 2002. 1, 6

[14] J. Mairal, F. Bach, J. Pnce, G. Sapiro, and A. Zisserman. Discriminative learned dictionaries for local image analysis. *IEEE Computer Vision and Pattern Recognition (CVPR)*, Anchorage, AL, June 2008. 1

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

[16] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Supervised dictionary learning. *Advances in Neural Information Processing Systems*, Vancouver, B.C., Canada, Dec. 2008. 1

[17] M. Ranzato, F. Haung, Y. Boureau, and Y. LeCun. Unsupervised learning of invariant feature hierarchies with applications to object recognition. *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007. 1

[18] A. Shrivastava, J. K. Pillai, V. M. Patel, and R. Chellappa. Learning discriminative dictionaries with partially labeled data. *International Conference on Image Processing (ICIP)*, 2012. 1, 5

[19] T. Sim, S. Baker, and M. Bsat. The cmu pose, illumination, and expression database. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(12):1615–1618, Dec. 2003. 6

[20] P. Sprechmann and G. Sapiro. Dictionary learning and sparse coding for unsupervised clustering. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 2042–2045, March 2010. 1, 5