Adaptively Unified Semi-Supervised Dictionary Learning with Active Points

Xiaobo Wang[1,2] Xiaojie Guo[1*] Stan Z. Li[2]
[1]State Key Laboratory of Information Security, IIE, Chinese Academy of Sciences
[2]Center for Biometrics and Security Research & National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences
{xiaobo.wang, szli}@nlpr.ia.ac.cn xj.max.guo@gmail.com

Abstract

Semi-supervised dictionary learning aims to construct a dictionary by utilizing both labeled and unlabeled data. To enhance the discriminative capability of the learned dictionary, numerous discriminative terms have been proposed by evaluating either the prediction loss or the class separation criterion on the coding vectors of labeled data, but with rare consideration of the power of the coding vectors corresponding to unlabeled data. In this paper, we present a novel semi-supervised dictionary learning method, which uses the informative coding vectors of both labeled and unlabeled data, and adaptively emphasizes the high confidence coding vectors of unlabeled data to enhance the dictionary discriminative capability simultaneously. By doing so, we integrate the discrimination of dictionary, the induction of classifier to new testing data and the transduction of labels to unlabeled data into a unified framework. To solve the proposed problem, an effective iterative algorithm is designed. Experimental results on a series of benchmark databases show that our method outperforms other state-of-the-art dictionary learning methods in most cases.

1. Introduction

Semi-Supervised Dictionary Learning (SSDL) is a learning paradigm that usually suits for the situation when labeled data $\mathbf{X}^l$ are scarce while unlabeled data $\mathbf{X}^u$ are abundant. To take advantages of the abundant unlabeled data in real-world applications such as classification, a variety of semi-supervised discriminative dictionary learning approaches [2, 19, 20, 27, 9] have been proposed recently, the general model of which can be summarized as follows:

$$\min_{\mathbf{D},\mathbf{Z}^l,\mathbf{Z}^u,\mathbf{W}} \mathscr{R}(\mathbf{X}^l,\mathbf{X}^u,\mathbf{D},\mathbf{Z}^l,\mathbf{Z}^u) + \lambda_1\mathscr{S}(\mathbf{Z}^l,\mathbf{Z}^u) + \lambda_2\mathscr{D}(\mathbf{Z}^l), \quad (1)$$

where $\mathscr{R}(\mathbf{X}^l,\mathbf{X}^u,\mathbf{D},\mathbf{Z}^l,\mathbf{Z}^u)$ denotes the reconstruction error term for both labeled and unlabeled data over the desired dictionary $\mathbf{D}$. $\mathscr{S}(\mathbf{Z}^l,\mathbf{Z}^u)$ stands for the regularizer on the coding matrix of labeled data $\mathbf{Z}^l$ and that of unlabeled $\mathbf{Z}^u$. $\mathscr{D}(\mathbf{Z}^l)$ represents the discriminative term on $\mathbf{Z}^l$. In addition, λ_1 and λ_2 are the trade-off parameters.[1]

From the perspective of different evaluations on the discriminative term $\mathscr{D}(\mathbf{Z}^l)$, existing methods can be mainly divided into two categories. One type of SSDL methods focuses on the class separation criterion on the coding vectors. Specifically, Pedro *et al.* [9] employed the Laplacian graph to encourage the closeness of the link counts, if two coding vectors are connected in the graph. Shrivastava *et al.* [20] used Fisher discriminant analysis on the coding vectors of labeled data to train the semi-supervised dictionary, which, however, very likely leads to overfitting due to the fact that the number of the labeled data is typically much smaller than that of the unlabeled. While the other type of SSDL evaluates the prediction loss on the coding vectors. Pham *et al.* [19] combined the classification error on the coding vectors of labeled data with the sparse reconstruction error of both labeled and unlabeled data into a joint framework. Zhang *et al.* [27] incorporated the reconstruction error, label consistency, and classification error of the coding vectors of labeled data into a joint objective function for online semi-supervised dictionary learning. Mohamadabadi *et al.* [2] proposed a probabilistic framework for semi-supervised dictionary learning and exploited a geometrical preserving term on both labeled and unlabeled data. Similarly, they also added a classification error term on the coding vectors of labeled data.

Although the above approaches generally provide more promising results than most discriminative Supervised Dictionary Learning (SDL) methods like [25, 10, 28], they have two main shortcomings. 1) Due to the lack of labeled data, only the coding vectors of labeled data to enhance the dictionary discriminative capability are insufficient. 2) When learning the dictionary by evaluating the prediction loss,

*Corresponding Author

[1]The specific meaning of other symbols will be described in Section 3.

1550-5499/15 $31.00 © 2015 IEEE
DOI 10.1109/ICCV.2015.208

they only take care of the induction of classifier to new testing data, without considering the transduction of labels to unlabeled data. In other words, they require an extra step to predict the labels of unlabeled data.

To overcome the above shortcomings, this paper proposes a novel semi-supervised dictionary learning algorithm by introducing an adaptive discriminative term. Compared with the coding vectors of labeled data, we argue that some coding vectors of unlabeled data also consist of discriminative information, although the coding vectors of unlabeled data are with no labels. And also, not every coding vectors of labeled data are helpful to determine the classifier hyperplane. Thus, only the "informative" coding vectors of labeled data are (adaptively) employed for enhancing the dictionary discriminative capability. We call all these involved coding vectors the *active points* in this paper. As a consequence, our discriminative term can be rewritten as $\mathscr{D}(\mathbf{Z}^l, \mathbf{Z}^u)$. To consider both the induction of classifier to new testing data and the transduction of labels to unlabeled data, we develop a unified dictionary learning model.

2. Related work

Our method is closely relevant to supervised dictionary learning, which can be described as:

$$\min_{\mathbf{D},\mathbf{Z}^l,\mathbf{W}} \mathscr{R}(\mathbf{X}^l, \mathbf{D}, \mathbf{Z}^l) + \lambda_1 \mathscr{S}(\mathbf{Z}^l) + \lambda_2 \mathscr{D}(\mathbf{Z}^l). \quad (2)$$

In last decades, a large body of research on SDL has been carried out [25, 10, 14, 6, 28, 15, 24]. The main difference among existing methods lies in the discrimination term $\mathscr{D}(\mathbf{Z}^l)$, which concentrates on the discriminative capability of the desired dictionary. Yang *et al*. [25] proposed a Fisher discrimination dictionary learning method called FDDL, which applies the Fisher discrimination criterion on the coding vectors to improve the discrimination. Mairal *et al*. [14] incorporated a logistic loss function into the problem and jointly learned a classifier and a dictionary. Zhang *et al*. [28] extended the original K-SVD [1] algorithm to learn a linear classifier for face recognition. Mairal *et al*. [15] proposed a task-driven SDL framework, which minimizes different risk functions of the coding vectors of labeled data for different tasks. Jiang *et al*. [10] introduced a label consistent regularization to enforce the discrimination of coding vectors, which is named LC-KSVD algorithm. Yang *et al*. [24] introduced a latent vector to build the relationship between dictionary atoms and class labels, which can reduce the correlation of dictionary atoms between different classes. Cai *et al*. [6] used the support vector to guide the dictionary learning (SVGDL). The SVGDL exhibits good classification results. Most of these supervised learning methods can achieve promising classification performance when there are sufficient and diverse labeled samples for training.

3. Proposed Model

3.1. SSDL for Sparse Reconstruction

The training data consist of both labeled $\{(\mathbf{x}_i^l, \mathbf{y}_i)\}_{i=1}^{n_l}$ and unlabeled data $\{\mathbf{x}_j^u\}_{j=1}^{n_u}$, where n_l and n_u are the numbers of labeled and unlabeled samples, respectively. $\mathbf{x}_i^l \in \mathbb{R}^d$ denotes the d dimensional feature vector extracted from the i^{th} labeled data points, $\mathbf{y}_i \in \{1, -1\}^C$ is its corresponding class label, which forms the label matrix $\mathbf{Y}$ as a column. C is the number of classes. $\mathbf{x}_j^u \in \mathbb{R}^d$ is the j^{th} unlabeled feature vector. Moreover, $n = n_l + n_u$ is the total number of available training samples and usually n_l is much smaller than n_u. $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, ..., \mathbf{d}_m] \in \mathbb{R}^{d\times m}$ is the desired dictionary containing m atoms. $\mathbf{Z}^l = [\mathbf{z}_1^l, \mathbf{z}_2^l, ..., \mathbf{z}_{n_l}^l]$ is composed of the coding vectors with respect to $\mathbf{X}^l = [\mathbf{x}_1^l, \mathbf{x}_2^l, ..., \mathbf{x}_{n_l}^l]$ over $\mathbf{D}$ and $\mathbf{Z}^u = [\mathbf{z}_1^u, \mathbf{z}_2^u, ..., \mathbf{z}_{n_u}^u]$ consists of the coding vectors of $\mathbf{X}^u = [\mathbf{x}_1^u, \mathbf{x}_2^u, ..., \mathbf{x}_{n_u}^u]$ over $\mathbf{D}$.

We first focus on the sparse reconstruction term. It is expected that both the labeled data $\mathbf{X}^l$ and unlabeled data $\mathbf{X}^u$ can be well represented by the items of the learned dictionary. Formally, the reconstruction error can be formulated as follows:

$$\mathscr{R}(\mathbf{X}^l, \mathbf{X}^u, \mathbf{D}, \mathbf{Z}^l, \mathbf{Z}^u) = ||\mathbf{X}^l - \mathbf{D}\mathbf{Z}^l||_F^2 + ||\mathbf{X}^u - \mathbf{D}\mathbf{Z}^u||_F^2, \quad (3)$$

where $\|\cdot\|_F$ designates the Frobenius norm. As regard the regularizer on the representation (coding vectors), without loss of generality, we impose the ℓ_1 norm on the coding vectors of both labeled and unlabeled data to achieve the sparsity. Consequently, the regularization can be written as:

$$\mathscr{S}(\mathbf{Z}^l, \mathbf{Z}^u) = ||\mathbf{Z}^l||_1 + ||\mathbf{Z}^u||_1. \quad (4)$$

Please note that, for brevity, we rename the coding vectors of labeled data the labeled coding vectors, and those of unlabeled the unlabeled coding vectors for the rest of this paper.

3.2. SSDL for Adaptively Unified Classification

For the optimality of the learned dictionary to classification, it would be helpful to jointly learn the dictionary and the classification model, as in [27, 19], which attempts to optimize the learned dictionary for classification tasks. And a simple quadratic loss function on labeled coding vectors is used to measure the classification loss. Our model follows this line due to its strong motivation, but differently, further makes uses of the unlabeled data.

Intuition says that considering the coding vectors where the classifier is fairly certain usually has no effect on the learning problem. That means the coding vectors near the boundary are more crucial for learning the classifying hyperplane. So we only punish the coding vectors within the (temporary) margin to rectify the hyperplane. For the labeled coding vectors within the margin, we rename them the clear points (solid triangles shown in Fig. 1 (a)). For the

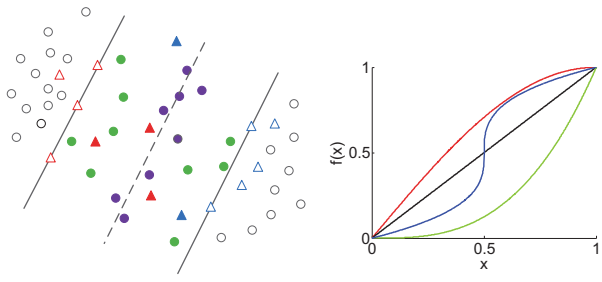
unlabeled ones, we call them the confidence points (solid circles in Fig. 1 (a)). Because of the lack of label information, the confidence points around the decision boundary should be paid less attention to (the low confidence points, solid purple circles in Fig. 1 (a)), as they may hurt the classification performance. In contrast, for the confidence points with high probabilities (high confidence points) of belonging to a certain class, we should pay more attention to them (solid green circles in Fig. 1 (a)). We call clear points and confidence points *active points* in this paper.

Unfortunately, in practice, we do not know which points are active in advance, let along which are of high confidence. So it is desirable to develop an algorithm that can automatically pick out the active points and adaptively weight those unlabeled according to their confidence. More precisely, in each iteration, to automatically select the active points, we jointly learn a dictionary and a classifier by incorporating squared hinge losses on the labeled and unlabeled coding vectors. Furthermore, to adaptively emphasize the high confidence points, we introduce a probability matrix $\mathbf{P} \in \mathbb{R}^{n_u \times C}$, whose $(j,k)^{th}$ entry p_{jk} corresponds to the probability of the j^{th} unlabeled coding vector that belongs to the k^{th} class, and satisfies $0 \leq p_{jk} \leq 1$ and $\sum_{k=1}^{C} p_{jk} = 1$. We use a function f on the element p_{jk} to adaptively assign the weights to confidence points. So our discriminative term can be naturally defined as follows:

$$\begin{aligned}\mathscr{D}(\mathbf{Z}^l, \mathbf{Z}^u) = ||\mathbf{W}||_F^2 + \theta(\sum_{i=1}^{n_l}\sum_{c=1}^{C} ||y_i^c(\mathbf{w}_c^T\mathbf{z}_i^l + b_c) - 1||_2^2 \\ + \sum_{j=1}^{n_u}\sum_{k=1}^{C} f(p_{jk}) \sum_{c=1}^{C} ||y_j^c(k)(\mathbf{w}_c^T\mathbf{z}_j^u + b_c) - 1||_2^2),\end{aligned} \tag{5}$$

where $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_C] \in \mathbb{R}^{m \times C}$ is the model parameter matrix that needs to be learned; $\mathbf{b} = [b_1, b_2, ..., b_C] \in \mathbb{R}^C$ is the corresponding bias; $y_i^c = 1$ if $y_i = c$, and otherwise $y_i^c = -1$; $y_j^c(k) = 1$ if $k = c$, otherwise $y_j^c(k) = -1$; θ is a regularization parameter; $(.)^T$ denotes the transposition operation.

For the function f, on one hand, it should be able to adaptively emphasize more on the high confidence points and less on the low confidences. On the other hand, the function could balance the two classification error terms. Since the probability is employed to reflect how possible an unlabeled coding vector belongs to a certain class, its weight should be never larger than that of a labeled coding vector. In other words, the range of f is in [0,1]. Different choices of the function f that satisfy the above two properties, can be found in Fig. 1 (b). For simplicity, we choose the power function (as green curve in Fig. 1 (b)) $f(p_{jk}) = p_{jk}^r$ with $r \in [1, \infty]$, which has been used in [5, 21] for other tasks like fuzzy clustering. It is easy to see



(a) (b)

Figure 1. (a) shows an illustrative two-class example, in which the black dotted line depicts the decision boundary, the margin is determined by two solid lines. The triangles and circles denote the labeled and unlabeled coding vectors, respectively. The solid triangles are the clear points, the green solid circles are of high confidence, while the purple circles of low confidence. In (b), four example strategies of f, the green curve $f(x) = x^3$, the red curve $f(x) = \sin(\frac{\pi}{2}x)$, the blue curve $f(x) = 0.5(2x-1)^{\frac{1}{3}} + 0.5$ and the black curve $f(x) = x$ are given for two-class cases.

that as r increases, p_{jk}^r decreases for both high and low confidence points. But, the weights of low confidence points are suppressed much faster than those of high confidences.

Putting every concern together, say Eqs. (3), (4) and (5), the final USSDL model turns out to be like:

$$\begin{aligned}&\min_{\Theta} \mathscr{R}(\mathbf{X}^l, \mathbf{X}^u, \mathbf{D}, \mathbf{Z}^l, \mathbf{Z}^u) + \lambda_1 \mathscr{S}(\mathbf{Z}^l, \mathbf{Z}^u) + \lambda_2 \mathscr{D}(\mathbf{Z}^l, \mathbf{Z}^u) \\ &\text{s.t.} \quad \forall j, \ \ p_{jk} \in [0,1], \quad \sum_{k=1}^{C} p_{jk} = 1,\end{aligned} \tag{6}$$

where $\Theta = \{\mathbf{D}, \mathbf{Z}^l, \mathbf{Z}^u, \mathbf{W}, \mathbf{b}, \mathbf{P}\}$ and, λ_1 and λ_2 are the trade-off parameters to balance the relative importance of the reconstruction error, the sparsity of the coding vectors, and the discrimination of both the labeled and unlabeled coding vectors.

4. Optimization

4.1. Solver of Problem (6)

The objective of USSDL model (6) is not jointly convex in Θ, but is convex with respect to each variable when the others are fixed. Thus we derive an effective iterative algorithm that alternatively optimizes the objective function (6) with respect to $\mathbf{P}, \mathbf{W}, \mathbf{b}, \mathbf{Z}^l, \mathbf{Z}^u$ and $\mathbf{D}$.

Update P. With $\{\mathbf{W}, \mathbf{b}, \mathbf{D}, \mathbf{Z}^l, \mathbf{Z}^u\}$ fixed, the objective associated with $\mathbf{P}$ reduces to:

$$\begin{aligned}&\min_{\mathbf{P}} \sum_{j=1}^{n_u}\sum_{k=1}^{C} p_{jk}^r \sum_{c=1}^{C} ||y_j^c(k)(\mathbf{w}_c^T\mathbf{z}_j^u + b_c) - 1||_2^2 \\ &\text{s.t.} \ \ \forall j, p_{jk} \in [0,1], \quad \sum_{k=1}^{C} p_{jk} = 1.\end{aligned} \tag{7}$$

Denote $e_{jk} = \sum_{c=1}^{C} ||y_j^c(k)(\mathbf{w}_c^T \mathbf{z}_i^u + b_c) - 1||_2^2$, which is the distance of the j^{th} unlabeled data to the k^{th} class. If $y_j^c(k)(\mathbf{w}_c^T \mathbf{z}_i^u + b_c) - 1 > 0$ in the previous iteration, we use 0 to replace the loss and remain others. Obviously, Eq.(7) can be decoupled between samples, so it is equivalent to:

$$\begin{aligned} &\min_{p_{jk}} \sum_{k=1}^{C} p_{jk}^r e_{jk} \\ &\text{s.t. } \forall j, p_{jk} \in [0,1], \quad \sum_{k=1}^{C} p_{jk} = 1. \end{aligned} \tag{8}$$

When r is chosen to 1, the optimal solution of Eq.(8) is:

$$p_{jk} = \begin{cases} 1, & k = k^* \\ 0, & otherwise \end{cases} \tag{9}$$

where $k^* = \underset{k \in \{1,2,...,C\}}{\operatorname{argmin}} e_{ik}$. It indicates that the possible class label of an unlabeled point is determined by the minimum distance to a certain class. Alternatively, when $r > 1$, the Lagrangian function of Eq.(8) gives:

$$\sum_{k=1}^{C} p_{jk}^r e_{jk} - \beta(\sum_{k=1}^{C} p_{jk} - 1), \tag{10}$$

where β is the Lagrangian multiplier. Setting the derivative of Eq.(10) with respect to p_{jk} to zero leads to the optimal solution of the subproblem, *i.e.*

$$p_{jk} = \left(\frac{\beta}{r(e_{jk} + \epsilon)} \right)^{\frac{1}{r-1}}, \tag{11}$$

where ϵ is a very small constant, which is to prevent from the invalid solution. Substituting (11) into the constraint $\sum_{k=1}^{C} p_{jk} = 1$ provides the following:

$$p_{jk} = (\frac{1}{e_{jk} + \epsilon})^{\frac{1}{r-1}} / \sum_{k=1}^{C} (\frac{1}{e_{jk} + \epsilon})^{\frac{1}{r-1}}. \tag{12}$$

Update $(\mathbf{Z}^l, \mathbf{Z}^u)$. When $\{\mathbf{D}, \mathbf{P}, \mathbf{W}, \mathbf{b}\}$ are fixed, we solve $\mathbf{Z}^l$ and $\mathbf{Z}^u$ by columns. For each $\mathbf{z}_i^l$, the optimization problem is as follows:

$$\begin{aligned} \min_{\mathbf{z}_i^l} ||\mathbf{x}_i^l - \mathbf{D}\mathbf{z}_i^l||_F^2 + \lambda_1 ||\mathbf{z}_i^l||_1 + \\ \lambda_2 \theta \sum_{c \in C_1} ||y_i^c(\mathbf{w}_c^T \mathbf{z}_i^l + b_c) - 1||_2^2, \end{aligned} \tag{13}$$

with the definition $C_1 = \{c | y_i^c(\mathbf{w}_c^T \mathbf{z}_i^l + b_c) - 1 < 0, c \in \{1, ..., C\}\}$. We can write Eq.(13) into a compact representation form in the following way:

$$\min_{\mathbf{z}_i^l} \left|\left| \begin{bmatrix} \mathbf{x}_i^l \\ \widetilde{\mathbf{x}}_i^l \end{bmatrix} - \begin{bmatrix} \mathbf{D} \\ \widetilde{\mathbf{D}}_1 \end{bmatrix} \mathbf{z}_i^l \right|\right|_2^2 + \lambda_1 ||\mathbf{z}_i^l||_1, \tag{14}$$

where $\widetilde{\mathbf{x}}_i^l = \sqrt{(\lambda_2 \theta)}[1 - y_i^1 b_1, ..., 1 - y_i^{|C_1|} b_{|C_1|}]^T$ and $\widetilde{\mathbf{D}}_1 = \sqrt{(\lambda_2 \theta)}[y_i^1 \mathbf{w}_1, ..., y_i^{|C_1|} \mathbf{w}_{|C_1|}]^T$.

For each $\mathbf{z}_j^u$, the optimization problem can be written as:

$$\begin{aligned} \min_{\mathbf{z}_j^u} ||\mathbf{x}_j^u - \mathbf{D}\mathbf{z}_j^u||_2^2 + \lambda_1 ||\mathbf{z}_j^u||_1 + \\ \lambda_2 \theta \sum_{k=1}^{C} p_{jk}^r \sum_{c \in C_2^k} ||y_j^c(k)(\mathbf{w}_c^T \mathbf{z}_j^u + b_c) - 1||_2^2, \end{aligned} \tag{15}$$

where $C_2^k = \{c | y_j^c(k)(\mathbf{w}_c^T \mathbf{z}_j^u + b_c) - 1 < 0, c \in \{1, ..., C\}\}$, $k = 1, ..., C$, which is equivalent to:

$$\min_{\mathbf{z}_j^u} \left|\left| \begin{bmatrix} \mathbf{x}_j^u \\ \widetilde{\mathbf{x}}_j^u \end{bmatrix} - \begin{bmatrix} \mathbf{D} \\ \widetilde{\mathbf{D}}_2 \end{bmatrix} \mathbf{z}_j^u \right|\right|_2^2 + \lambda_1 ||\mathbf{z}_j^u||_1, \tag{16}$$

where $\widetilde{\mathbf{x}}_j^u = [\sqrt{(\lambda_2 \theta p_{j1}^r)}[1 - y_j^1(1) b_1, ..., 1 - y_j^{|C_2^1|}(1) b_{|C_2^1|}], ..., \sqrt{(\lambda_2 \theta p_{jC}^r)}[1 - y_j^1(C) b_1, ..., 1 - y_j^{|C_2^C|}(C) b_{|C_2^C|}]]^T$, $\widetilde{\mathbf{D}}_2 = [\sqrt{(\lambda_2 \theta p_{j1}^r)}[y_j^1(1)\mathbf{w}_1, ..., y_j^{|C_2^1|}(1)\mathbf{w}_{|C_2^1|}], ..., \sqrt{(\lambda_2 \theta p_{jC}^r)}[y_j^1(C)\mathbf{w}_1, ..., y_j^{|C_2^C|}(C)\mathbf{w}_{|C_2^C|}]]^T$.

Obviously, the objective functions of Eq.(14) and (16) are exactly the Lasso problem, which can be solved by the standard l_1 algorithm [8, 23].

Update D. With $\{\mathbf{P}, \mathbf{W}, \mathbf{b}, \mathbf{Z}^l, \mathbf{Z}^u\}$ fixed, we solve the subproblem of $\mathbf{D}$ through optimizing:

$$\begin{aligned} \min_{\mathbf{D}} \ &||\mathbf{X}^l - \mathbf{D}\mathbf{Z}^l||_F^2 + ||\mathbf{X}^u - \mathbf{D}\mathbf{Z}^u||_F^2 \\ &\text{s.t. } \forall m \quad ||\mathbf{d}_m||^2 \leq 1, \end{aligned} \tag{17}$$

where the additional constraints are introduced to avoid the scaling issue of the atoms. The $\mathbf{D}$ subproblem can be solved effectively by the Lagrange dual method [12].

Update $(\mathbf{W}, \mathbf{b})$. With the updated $\{\mathbf{P}, \mathbf{Z}^l, \mathbf{Z}^u, \mathbf{D}\}$, we solve the subproblem of the classifier $(\mathbf{W}, \mathbf{b})$ by optimizing:

$$\begin{aligned} \min_{\mathbf{W}, \mathbf{b}} \ &\frac{1}{\theta} \|\mathbf{W}\|_F^2 + \sum_{i=1}^{n_l} \sum_{c=1}^{C} ||y_i^c(\mathbf{w}_c^T \mathbf{z}_i^l + b_c) - 1||_2^2 \\ &+ \sum_{j=1}^{n_u} \sum_{k=1}^{C} p_{jk}^r \sum_{c=1}^{C} ||y_j^c(k)(\mathbf{w}_c^T \mathbf{z}_j^u + b_c) - 1||_2^2, \end{aligned} \tag{18}$$

For each c, if $y_i^c(\mathbf{w}_c^T \mathbf{z}_i^l + b_c) - 1 > 0$ or $y_j^c(k)(\mathbf{w}_c^T \mathbf{z}_j^u + b_c) - 1 > 0$ in the previous iteration, we use 0 to replace the loss. As a result, the objective function becomes:

$$\begin{aligned} \min_{\mathbf{w}_c, b_c} \ &\frac{1}{\theta} ||\mathbf{w}_c||_2^2 + \sum_{i \in CP_1} ||y_i^c(\mathbf{w}_c^T \mathbf{z}_i^l + b_c) - 1||_2^2 \\ &+ \sum_{j \in CP_2} \sum_{k=1}^{C} p_{jk}^r ||y_j^c(k)(\mathbf{w}_c^T \mathbf{z}_j^u + b_c) - 1||_2^2, \end{aligned} \tag{19}$$

Algorithm 1: USSDL

Input: Labeled data $\{(\mathbf{x}_i^l, \mathbf{y}_i)\}_{i=1}^{n_l}$, unlabel data $\{\mathbf{x}_j^u\}_{j=1}^{n_u}$, $\mathbf{D}_{init}$, $\mathbf{Z}_{init}^l$, $\mathbf{Z}_{init}^u$, $\mathbf{W}_{init}$, $\mathbf{b}_{init}$, $\lambda_1 > 0$, $\lambda_2 > 0$, $\theta > 0$ and $r \geq 1$.

while *not converged* **do**

1. Update the probability matrix $\mathbf{P}$ via Eq.(9) or Eq.(12).
2. Update the coding matrix $(\mathbf{Z}^l, \mathbf{Z}^u)$ by columns, the Eq.(14) and Eq.(16) can be solved by the standard l_1 algorithm [8, 23].
3. Update the dictionary $\mathbf{D}$ by solving the Eq.(17) using Lagrange dual method [12].
4. Update the classifier $(\mathbf{W}, \mathbf{b})$ via Eq.(20) and Eq.(21).

end

Output: The optimal dictionary $\mathbf{D}$, the classifier $(\mathbf{W}, \mathbf{b})$, the probability matrix $\mathbf{P}$, and the coding matrix $(\mathbf{Z}^l, \mathbf{Z}^u)$.

where CP_1 and CP_2 denote the clear points set and the confidence points set, respectively. Setting the derivative of Eq.(19) with respect to $\mathbf{w}_c$ to 0 directly offers:

$$\mathbf{w}_c = \left(\sum_{i \in CP_1} \mathbf{z}_i^l (\mathbf{z}_i^l)^T + \sum_{j \in CP_2} \sum_{k=1}^{C} p_{jk}^r \mathbf{z}_j^u (\mathbf{z}_j^u)^T + \frac{I}{\theta} \right)^{\dagger} \left(\sum_{i \in CP_1} \mathbf{z}_i^l (y_i^c - b_c) + \sum_{j \in CP_2} \sum_{k=1}^{C} p_{jk}^r \mathbf{z}_j^u (y_j^c(k) - b_c) \right), \tag{20}$$

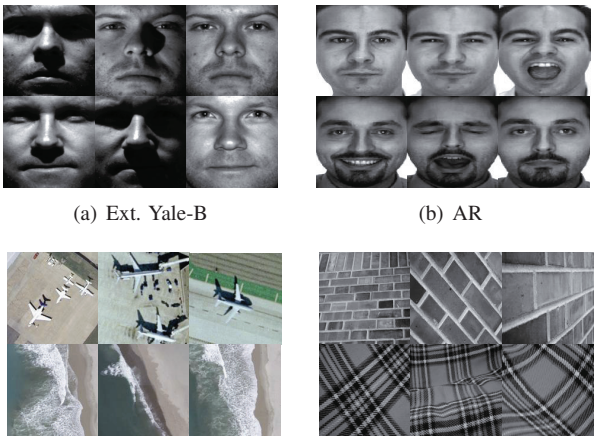
where $(\cdot)^{\dagger}$ denotes the pseudo-inverse.

Similarly, setting the derivative of Eq.(19) with respect to b_c to zero, we obtain:

$$b_c = q_1 / q_2, \tag{21}$$

where $q_1 = \sum_{i \in CP_1} (y_i^c - \mathbf{z}_i^{l\,T} \mathbf{w}_c) + \sum_{j \in CP_2} \sum_{k=1}^{C} p_{jk}^r (y_j^c(k) - \mathbf{z}_j^{u\,T} \mathbf{w}_c)$, $q_2 = (|CP_1| + \sum_{j \in CP_2} \sum_{k=1}^{C} p_{jk}^r)$, and $|\cdot|$ designates the cardinality of a set.

For clarity, the entire algorithm of solving the problem (6) is summarized in Algorithm 1, which terminates when the change of objective function value between two neighboring iterations is sufficiently small ($< 10^{-3}$) or the maximal number (20) of iterations is reached. The initialization of the dictionary is executed as follows. For each class, if the required size of sub-dictionary is over the number of corresponding labeled data, all the labeled are embraced and the excess is filled up with randomly selected unlabeled points. Otherwise, the sub-dictionary is composed of the randomly selected labeled data. Then we concatenate all the sub-dictionaries together as the final $\mathbf{D}_{init}$. $(\mathbf{Z}_{init}^l \ \mathbf{Z}_{init}^u)$ are the corresponding coding matrices. The initialization of the classifier $(\mathbf{W}_{init}, \mathbf{b}_{init})$ is calculated by [22].



(a) Ext. Yale-B (b) AR

(c) LandUse-21 (d) Texture-25

Figure 2. Sample images from each data set.

4.2. Unified Classification Strategy

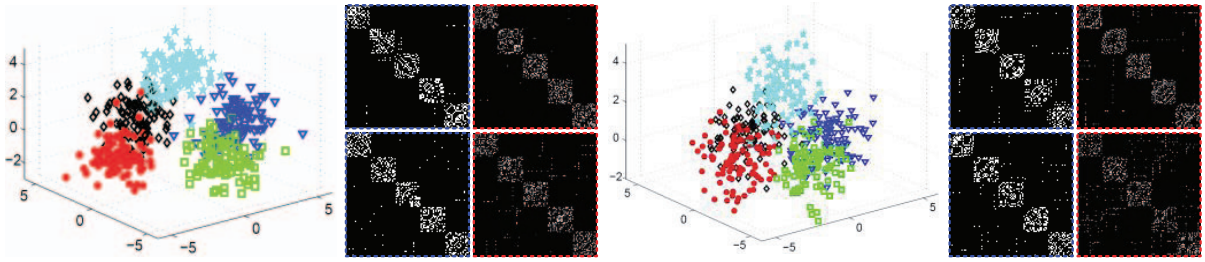
In the USSDL model, once the dictionary $\mathbf{D}$, the probability matrix $\mathbf{P}$ and the classifier $(\mathbf{W}, \mathbf{b})$ are learned, we can perform the classification task as follows. For the transduction of unlabeled data in the training process, we adopt a voting strategy on $\mathbf{P}$ to tag the unlabel data. For the induction of classifier on the new testing data $\mathbf{x}^t$, we first compute the coding vector $\mathbf{z}^t$ over the learned $\mathbf{D}$ by solving the lasso problem: $\mathbf{z}^t = \underset{\mathbf{z}^t}{\operatorname{argmin}} ||\mathbf{x}^t - \mathbf{D}\mathbf{z}^t||_F^2 + \lambda_1 ||\mathbf{z}^t||_1$. Then we simply apply the C classifier $(\mathbf{w}_c, b_c)$ $(c \in \{1, 2, ..., C\})$ on $\mathbf{z}^t$ to predict the label y of the test data $\mathbf{x}^t$ by:

$$y = \underset{c \in \{1,2,\ldots,C\}}{\operatorname{argmax}} \ \mathbf{w}_c{}^T \mathbf{z}^t + b_c. \tag{22}$$

5. Experiments

For clearly viewing the property of our model, we first conduct two tests on synthesized data. To show the efficacy of our method, we then apply our USSDL on two tasks including face recognition and object classification. For face recognition, two representative face datasets, *i.e.* Extended Yale-B [13] and AR [16] are employed. As for object classification, we adopt LandUse-21 [26] and Texture-25 [11]. Figure 2 shows sample images from these four databases. Several most representative SDL methods including FDDL [25], LC-KSVD [10], SVGDL [6], and SSDL approaches including S2D2 [20] and JDL [19] and our USSDL, are involved for performance comparison. In addition, we also compare the proposed USSDL with the classic classification method, *i.e.* the multi-class linear SVM (M-SVM) [22].

Please notice that, for the compared methods in terms of transduction, they treat the unlabeled data as new testing data and use the learned dictionary or classifier to predict the labels of unlabeled data. To verify the effectiveness of our method to semi-supervised problem, different numbers of



(a) (b) (c) (d)

Figure 3. (a) and (c) Case $\mathbf{I}_c$ and $\mathbf{II}_c$ data, respectively. (b) and (d) Affinities of $\mathbf{Z}^l$ (left) and $\mathbf{Z}^u$ (right) on clean (upper) and noisy (lower) data, respectively.

Ext. Yale-B		M-SVM	FDDL	LC-KSVD	SVGDL	S2D2	JDL	USSDL
$k_l = 2$	Unlabeled	34.7±3.3	51.0±3.4	47.2±3.2	51.9±3.0	53.6±2.9	54.2±2.6	**59.4±2.7**
	Testing	38.0±2.6	52.4±2.5	48.5±2.8	53.4±2.2	53.4±2.1	55.2±1.8	**60.5±2.1**
$k_l = 5$	Unlabeled	62.9±1.8	80.1±1.0	68.4±3.6	76.7±1.9	78.3±1.0	76.9±3.1	**85.9±2.8**
	Testing	66.6±1.1	82.3±0.7	69.6±3.6	81.1±1.0	76.1±1.3	77.4±2.8	**86.5±2.1**
$k_l = 10$	Unlabeled	80.6±1.2	91.7±1.0	83.2±4.0	90.5±1.1	85.9±1.7	84.4±2.1	**92.1±1.3**
	Testing	83.8±0.8	92.1±0.3	84.6±3.8	91.7±5.8	83.2±1.9	85.3±1.6	**93.6±0.8**

AR		M-SVM	FDDL	LC-KSVD	SVGDL	S2D2	JDL	USSDL
$k_l = 2$	Unlabeled	75.3±3.4	92.0±1.3	68.5±3.6	90.7±2.1	89.3±3.0	87.4±1.4	**92.8±2.3**
	Testing	60.1±2.2	83.6±1.8	67.4±4.2	82.1±1.8	85.3±3.1	87.2±2.0	**89.1±2.3**
$k_l = 3$	Unlabeled	88.2±1.6	93.4±1.3	89.5±2.8	94.1±1.5	92.6±1.4	89.1±1.6	**94.8±1.6**
	Testing	74.3±1.2	89.7±2.1	89.2±3.6	90.3±2.0	89.2±1.9	88.2±1.8	**91.3±1.4**
$k_l = 5$	Unlabeled	93.7±2.1	96.2±1.1	92.8±1.6	**97.6±0.6**	94.5±1.4	91.2±1.8	96.9±1.0
	Testing	84.9±2.0	93.6±0.9	91.5±2.1	93.8±1.5	92.1±1.1	90.7±1.2	**94.1±1.3**

Table 1. Classification accuracy and standard deviation (%) for running different compared methods with different number (k_l) of labeled data per class on Extended Yale-B (upper) and AR (lower) datasets, respectively.

labeled data from each class are used for training. And for those SDL methods, the dictionary size is set corresponding to the number of labeled data. For all the involved approaches, the best possible results obtained by tuning their parameters are finally reported.

5.1. Synthetic Data

Toy datasets are randomly generated from 5 Gaussian components with 3-D, *i.e.* $\mathcal{N}_3(\mu, \mathbf{I})$. There are 100 points (50 for training, 50 for testing) from each component. Furthermore, 20 training points perform as labeled data while the remaining as unlabeled. It is worth noting that we employ the diagonal-block property of affinities formed by the representation matrices $\mathbf{Z^l}$ and $\mathbf{Z^u}$ with 5 nearest neighbors to reflect the performance, as it is the key to classifier training and dictionary learning. **Case $\mathbf{I}_c$**: Figure 3 (a) shows the clean data produced by $\{\mu_1 = [4,0,0]^T, \mu_2 = [-4,0,0]^T, \mu_3 = [0,4,0]^T, \mu_4 = [0,-4,0]^T, \mu_5 = [0,0,4]^T\}$, in which the overlap between different components is relatively small. We can see from the upper row given in Fig. 3 (b), the diagonal-block property of the affinities constructed from both $\mathbf{Z^l}$ and $\mathbf{Z^u}$ is almost perfect. **Case $\mathbf{I}_n$**: We introduce Gaussian white noise into $\mathbf{I}_c$ to see the robustness of our method. From the lower row, the diagonal blockness is well preserved by both $\mathbf{Z^l}$ and $\mathbf{Z^u}$. **Case $\mathbf{II}_c$**: As displayed in Fig. 3 (c), the dataset generated by $\{\mu_1 = [2,0,0]^T, \mu_2 = [-2,0,0]^T, \mu_3 = [0,2,0]^T, \mu_4 = [0,-2,0]^T, \mu_5 = [0,0,2]^T\}$ is no longer as separable as the previous, say the overlap between different components grows. **Case $\mathbf{II}_n$**: Again, the Gaussian white noise is added to the corresponding clean data. The affinities shown in Fig. 3 (d) are still of high quality, although a few more non-zero elements are at wrong positions, compared with those in Fig. 3 (b).

5.2. Face Recognition

In this part, we evaluate the performance of the proposed algorithm on Extended Yale-B [13] and AR [16] datasets. In both the two face recognition experiments, each sample has a reduced dimension of 300.

The Extended Yale-B database consists of 2414 frontal face images of 38 individuals. Each individual has 64 images and we randomly select 20 images as training set and use the rest as testing set. The number of dictionary atoms for our algorithm is fixed as 570 in this experiment. We

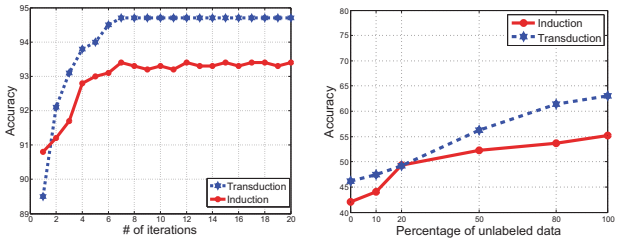


Figure 4. **Left** : the effectiveness of using unlabeled data on Texture-25 dataset. **Right** : the accuracy vs. iteration number curves of induction and transduction on Extended Yale-B dataset.

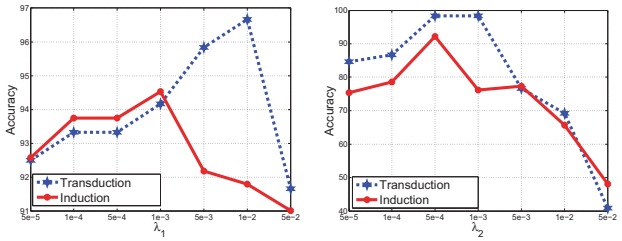


Figure 5. **Left** : the effect of λ_1 with λ_2 fixed. **Right** : the effect of λ_2 with λ_1 fixed.

randomly select $\{2, 5, 10\}$ samples from each class in the training set as labeled data, and the remaining as unlabeled data. The mean accuracies and standard deviations over 5 independent trials are summarized in the upper part of Table 1. We can observe that the merit of USSDL is more apparent when less data are labeled, compared with SDL methods. In addition, to see the performance of our algorithm can be iteratively improved, we give the accuracy *vs.* iteration number curves of induction and transduction in the first picture of Fig. 4. Evidently, both of them rapidly go up within only 7 iterations and almost keep afterwards.

The AR database consists of over 4000 images of 126 individuals. For each individual, 26 face images are collected from two separated sessions. We select 50 male individuals and 50 female individuals for our standard evaluation procedure. Focusing on the illumination and expression condition, we choose 7 images from Session 1 for training, and 7 images from Session 2 for testing. The number of dictionary atoms for our algorithm is fixed as 500 in this experiment. We randomly select {2,3,5} samples from each class in the training set as labeled data, and the remaining as unlabeled data. The lower part of Table 1 reports the numbers with regard to the cases with different settings. From Table 1, we can see that USSDL outperforms the others in most cases. This is consistent with the fact that when the labeled data are statically sufficient, they are already able to construct dictionaries well enough.

5.3. Object Classification

We here assess the performance of our USSDL on more complex datasets including LandUse-21 [26] and Texture-25 [11] for object classification. Different to the experiments on face recognition, we use the low-level features [7, 3], including PHOG [4], GIST [18] and LBP [17]. PHOG is computed with a 2-layer pyramid in 8 directions. GIST is computed on the rescaled images of 256×256 pixels, in 4, 8 and 8 orientations at 3 scales from coarse to fine. As for LBP, the uniform LBP is used. All the features are concatenated into a single 119-dimensional vector. We then reduce the dimension to 100. LandUse-21 consists of satellite images from 21 categories, 100 images each. We randomly select 10 categories as the dataset. Texture-25 dataset contains 25 texture categories, 40 samples each. In all our experiments, the dictionary sizes are fixed to 300 and 500 for LandUse-21 and Texture-25, respectively.

Different numbers of labeled samples per class are evaluated: LandUse-21 with $\{2, 5, 10, 30\}$, Texture-25 with $\{2, 5, 10, 15\}$. In all cases, 33% of the datasets are randomly left out for testing, and the remaining data are randomly split into labeled and unlabeled. The numbers shown in Table 2 is the average accuracies and standard deviations over 5 independent runs with random labeled-unlabeled splits for LandUse-21 and Texture-25 datasets, respectively. It is clear to see that our method outperforms the other compared approaches.

5.4. Effects of λ_1 and λ_2

This part attempts to reveal the influence of the main parameters appeared in our algorithm, including λ_1 and λ_2. Due to the complex dependence of the algorithm on these parameters, we test them separately, say varying one parameter with the other fixed. Figure 5 depicts the changes of accuracy on Extended Yale-B database (10 classes, 5 labeled samples from each class, and 100 dimension). From the curves, we can see that each parameter has a certain effect on the performance. In particular, for testing λ_1, we set $\lambda_2 = 5e{-}4$. From the first picture in Fig. 5, we can see that the accuracy in general increases in terms of both induction and transduction when tuning λ_1 from $5e{-}5$ up to $1e{-}3$, and drops afterward. For evaluating the effect of λ_2, we fix λ_1 as $1e{-}4$. The second picture in Fig. 5 tells that the best performance is obtained at $\lambda_2 = 5e{-}4$ in terms of both induction and transduction. With the value of λ_2 departed from $1e{-}3$, the performance degrades. [2]

5.5. Effectiveness of Using Unlabeled Data

We evaluate the performance gain from using extra unlabeled data in our unified semi-supervised dictionary learning. Various percentages $\{10\%, 20\%, 50\%, 80\%, 100\%\}$ of unlabeled data are emerged into labeled data for training. In the second picture of Fig. 4, we draw the curve of accuracy

[2]The parameter θ is used to avoid overfitting. We empirically set $\theta = 5$, which works sufficiently well for all the experiments shown in this paper.

LandUse-21		M-SVM	FDDL	LC-KSVD	SVGDL	S2D2	JDL	USSDL
$k_l = 2$	Unlabeled	40.7±1.9	42.9±3.1	40.3±3.2	40.9±2.8	44.6±2.1	43.7±2.4	**47.8±2.4**
	Testing	35.7±2.1	36.9±2.3	34.0±2.9	37.5±2.1	40.3±2.3	39.3±1.8	**43.9±1.3**
$k_l = 5$	Unlabeled	47.6±1.4	50.5±1.8	44.3±2.5	45.1±1.3	49.1±1.3	48.9±2.8	**55.8±2.1**
	Testing	41.3±3.0	40.5±4.5	39.4±4.8	40.2±2.2	44.7±2.1	43.6±1.9	**51.3±2.7**
$k_l = 10$	Unlabeled	59.2±1.8	60.6±2.0	50.1±2.1	51.8±3.6	53.5±1.3	55.7±1.6	**64.2±2.8**
	Testing	47.9±1.9	49.9±2.2	42.3±1.7	45.9±1.3	47.8±1.3	46.4±1.9	**54.2±2.2**
$k_l = 30$	Unlabeled	69.1±1.6	66.9±1.3	55.8±2.9	56.0±3.4	59.3±1.3	57.9±1.7	**71.9±1.9**
	Testing	53.0±2.9	54.0±4.2	48.9±2.7	50.8±1.3	50.9±1.7	49.3±0.8	**63.5±1.6**

Texture-25		M-SVM	FDDL	LC-KSVD	SVGDL	S2D2	JDL	USSDL
$k_l = 2$	Unlabeled	32.8±3.2	38.8±3.4	33.6±2.5	35.6±4.1	36.5±2.7	32.5±2.1	**39.5±3.1**
	Testing	24.9±3.4	31.4±4.0	28.0±4.1	29.8±3.9	31.7±2.3	27.6±2.1	**34.2±3.7**
$k_l = 5$	Unlabeled	47.2±1.5	54.6±1.8	40.6±1.7	42.1±4.5	51.5±1.3	46.7±1.6	**56.9±1.8**
	Testing	41.6±1.7	48.9±1.7	38.2±1.3	37.9±1.3	43.8±1.4	39.2±1.9	**51.1±2.2**
$k_l = 10$	Unlabeled	60.0±2.8	57.8±2.4	49.6±3.1	45.8±2.4	56.3±2.2	49.7±2.7	**64.0±1.9**
	Testing	52.9±2.7	52.6±3.1	48.6±2.9	40.3±2.3	47.9±2.4	43.3±0.8	**54.6±1.6**
$k_l = 15$	Unlabeled	62.6±0.8	63.6±1.7	57.8±3.2	63.9±1.4	59.3±1.3	54.9±1.7	**68.3±1.9**
	Testing	55.3±1.2	56.7± 1.4	54.1±2.9	56.8±1.3	50.9±1.7	50.3±0.8	**57.7±1.6**

Table 2. Classification accuracy and standard deviation (%) for running different compared methods with different number (k_l) of labeled data per class on LandUse-21 (upper) and Texture-25 (lower) dataset, respectively.

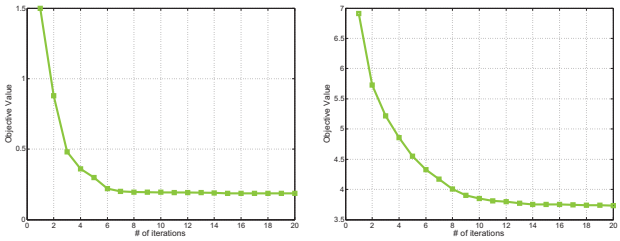


Figure 6. The curve of the objective value versus the number of iterations on LandUse-21 (left) and AR (right) dataset, respectively.

for the case with a fixed number (10) of labeled data from each class by varying amounts of used unlabeled data on Texture-25 database as an example. As can be seen from the curve, the classification accuracy goes up as the amount of unlabeled data increases. So we can conclude that our unified dictionary learning algorithm benefits from unlabeled data, which indicates that our dictionary learning model is suitable for the task of semi-supervised classification.

5.6. Convergence Speed

In this experiment, we show the converge speed of the proposed algorithm empirically. Similar to the analysis in [25], the four alternative optimizations in our USSDL model are convex, so the USSDL algorithm is guaranteed to converge at least at a stationary point. For convergence speed of the proposed dictionary learning method, without loss of generality, Figure. 6 reveals that the objective value drops as the number of iterations on LandUse-21 [26] and AR dataset [16] increases, respectively. It can be seen that the objective value drops quickly and becomes very small. In all the experiments shown in this paper, our algorithm empirically converges in less than 20 iterations.

6. Conclusion

In this paper, we have proposed a novel adaptively unified semi-supervised dictionary learning model named USSDL. Differently to the previous SSDL methods, our model merges the coding vectors of unlabeled data to the classification error term for enhancing the dictionary discriminative capability. Moreover, we automatically distinguish active points and adaptively assign weights to confidence points. We have integrated the discrimination of dictionary, the induction of classifier to new testing data and the transduction of labels to unlabeled data into a unified optimization framework and designed an effective iterative algorithm to seek the solution. The experiment results on several benchmark datasets have demonstrated that the active points are very helpful to learn better dictionaries for classification tasks, and shown the superior performance over the state-of-the-art supervised and semi-supervised dictionary learning methods in most cases.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (No. 61402467, 61375037, 61473291, 61572501, 61572536) and in part by the Excellent Young Talent Programme through the Institute of Information Engineering, Chinese Acadamy of Sciences.

References

[1] M. Aharon, E. Michael, and B. Alfred. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. on Signal Processing*, pages 4311–4322, 2006. 2

[2] B. Babagholami-Mohamadabadi, A. Zarghami, and M. Zolfaghari. Pssdl: Probabilistic semi-supervised dictionary learning. *In ECML/PKDD*, pages 192–207, 2013. 1

[3] X. Boix, G. Roig, and L. V. Gool. Comment on" ensemble projection for semi-supervised image classification". *arXiv preprint*, 2014. 7

[4] A. Bosch, A. Zisserman, and X. Muoz. Image classification using random forests and ferns. *In ICCV*, pages 1–8, 2007. 7

[5] D. J. C. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Taylor & Francis*, pages 32–57, 1973. 3

[6] S. Cai, W. Zuo, L. Zhang, X. Feng, and P. Wang. Support vector guided dictionary learning. *In ECCV*, pages 624–639, 2014. 2, 5

[7] D. Dai and L. V. Gool. Ensemble projection for semi-supervised image classification. *In ICCV*, pages 2072–2079, 2013. 7

[8] D. David and Y. Tsaig. Fast solution of l1-norm minimization problems when the solution may be sparse. *IEEE Trans. on Information Theory*, pages 4789–4812, 2008. 4, 5

[9] A. Forero, Pedro, R. Ketan, and B. G. Georgios. Semi-supervised dictionary learning for network-wide link load prediction. *In Cognitive Incromation Processing*, pages 1–5, 2012. 1

[10] Z. Jiang, Z. Lin, and L. S. Davis. Label consistent k-svd: learning a discriminative dictionary for recognition. *TPAMI*, pages 2651–2664, 2013. 1, 2, 5

[11] S. Lazebnik, C. Schmid, and J. Ponce. A sparse texture representation using local affine regions. *TPAMI*, pages 1265–1278, 2005. 5, 7

[12] H. Lee, A. Battle, R. Raina, and A. Ng. Efficient sparse coding algorithms. *In NIPS*, pages 801–808, 2006. 4, 5

[13] K.-C. Lee, H. Jeffrey, and K. David. Acquiring linear subspaces for face recognition under variable lighting. *TPAMI*, pages 684–698, 2005. 5, 6

[14] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Supervised dictionary learning. *In NIPS*, pages 1033–1040, 2008. 2

[15] J. Mairal, B. Francis, and P. Jean. Task-driven dictionary learning. *TPAMI*, pages 791–804, 2012. 2

[16] A. Martnez and B. Robert. The ar face database. *CVC Technical Report*, 1998. 5, 6, 8

[17] T. Ojala, P. Matti, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *TPAMI*, pages 971–987, 2002. 7

[18] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, pages 145–175, 2001. 7

[19] D.-S. Pham and V. Svetha. Joint learning and dictionary construction for pattern recognition. *In CVPR*, pages 1–8, 2008. 1, 2, 5

[20] A. Shrivastava, J. K. Pillai, V. M. Patel, and R. Chellappa. Learning discriminative dictionaries with partially labeled data. *In ICIP*, pages 3113–3116, 2012. 1, 5

[21] D. Wang, F. Nie, and H. Huang. Large-scale adaptive semi-supervised learning via unified inductive and transductive model. *In KDD*, pages 482–491, 2014. 3

[22] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. *In CVPR*, pages 1794–1801, 2009. 5

[23] J. Yang, Z. Zhou, A. Ganesh, S. S. Sastry, and Y. Ma. Fast l1-minimization algorithms for robust face recognition. *In ICIP*, pages 3234–3246, 2010. 4, 5

[24] M. Yang, D. Dai, L. Shen, and L. V. Gool. Latent dictionary learning for sparse representation based classification. *In CVPR*, pages 4138–4145, 2014. 2

[25] M. Yang, Z. Lei, Z. David, and X. Feng. Fisher discrimination dictionary learning for sparse representation. *In ICCV*, pages 543–550, 2011. 1, 2, 5, 8

[26] Y. Yang and S. Newsam. Bag-of-visual-words and spatial extensions for land-use classification. *In ACM GIS*, pages 270–279, 2010. 5, 7, 8

[27] G. Zhang, Z. Jiang, and L. S. Davis. Online semi-supervised discriminative dictionary learning for sparse representation. *In ACCV*, pages 259–273, 2012. 1, 2

[28] Q. Zhang and B. Li. Discriminative k-svd for dictionary learning in face recognition. *In CVPR*, pages 2691–2698, 2010. 1, 2