

BIROn - Birkbeck Institutional Research Online

Zhang, X. and Liu, Q. and Wang, D. and Zhao, L. and Gu, N. and Maybank, Stephen J. (2019) Self-taught semi-supervised dictionary learning with non-negative constraint. *IEEE Transactions on Industrial Informatics* 16 (1), pp. 532-543. ISSN 1551-3203.

Downloaded from: <https://eprints.bbk.ac.uk/id/eprint/28121/>

Usage Guidelines:

Please refer to usage guidelines at <https://eprints.bbk.ac.uk/policies.html>
contact lib-eprints@bbk.ac.uk.

or alternatively

Self-Taught Semi-Supervised Dictionary Learning with Non-Negative Constraint

Xiaoqin Zhang, *Member, IEEE*, Qianqian Liu, Di Wang, Li Zhao, Nannan Gu, and Steve Maybank, *Fellow, IEEE*

Abstract—This paper investigates classification by dictionary learning. A novel unified framework termed self-taught semi-supervised dictionary learning with non-negative constraint (NNST-SSDL) is proposed for simultaneously optimizing the components of a dictionary and a graph Laplacian. Specifically, an atom graph Laplacian regularization is built by using sparse coefficients to effectively capture the underlying manifold structure. It is more robust to noisy samples and outliers because atoms are more concise and representative than training samples. A non-negative constraint imposed on the sparse coefficients guarantees that each sample is in the middle of its related atoms. In this way the dependency between samples and atoms is made explicit. Furthermore, a *self-taught* mechanism is introduced to effectively feed back the manifold structure induced by atom graph Laplacian regularization and the supervised information hidden in unlabeled samples in order to learn a better dictionary. An efficient algorithm, combining a block coordinate descent method with the alternating direction method of multipliers is derived to optimize the unified framework. Experimental results on several benchmark datasets show the effectiveness of the proposed model.

Index Terms—Semi-supervised dictionary learning, non-negative constraint, atom graph regularization, self-taught

I. INTRODUCTION

CLASSIFICATION is widely used in the fields of industrial informatics for fault diagnosis, face recognition, visual tracking, action recognition, etc. [1], [2], [3], [4], [5]. Deep learning [6], [7], [8] and sparse coding [9], [10] are the two most popular methods for classification in the last decade. The former method, which is characterized by training deep neural networks to extract data features and learn functional relationship, has demonstrated a great potential in classification tasks. The success of deep learning is due to the powerful expressivity of deep neural networks. However, this also implies a very large hypothesis space which makes deep learning algorithms time consuming, uninterpretable and unstable because of overfitting. The latter method is based on the well-established theory of compressed sensing, which has good interpretability because many natural signals are sparse. These sparse signals can be approximated or even fully recovered. Sparse coding can handle small-sized training datasets and save a great deal of time in the training stage.

X. Zhang, Q. Liu, D. Wang and L. Zhao are with the College of Computer Science and Artificial Intelligence, Wenzhou University, 325035, China (e-mail: zhangxiaoqin@stu.wzu.edu.cn; liuqianqian@stu.wzu.edu.cn; wangdi@wzu.edu.cn; lizhao@wzu.edu.cn).

N. Gu is with the School of Statistics, Capital University of Economics and Business, Beijing 100070, China (e-mail: gu_nannan@126.com).

S. Maybank is with the School of Computer Science and Information Systems the School of Computer Science and Information Systems, Birkbeck College, London WC1E7HX, U.K. (e-mail: sjmaybank@dcs.bbk.ac.uk).

In this paper, we continue the fruitful investigation of the sparse coding for classification applications. Wright et al. [11] present a sparse representation-based classification (SRC) algorithm, and apply it to face recognition. However, SRC directly uses all training samples to form the dictionary, thus the representative power of dictionary will degenerate if the training samples are contaminated by noise. To deal with noisy samples, methods for learning a more compact and robust dictionary are proposed in [12], [13]. But these methods do not use the label information in the training process and hence are not suitable for classification tasks.

To enhance the discriminative capability of the learned dictionary, researchers propose a series of supervised dictionary learning (SDL) methods. A common technique is to add some discriminative terms to the dictionary learning framework. The discriminative terms include softmax discriminative cost function [14], Fisher discrimination function [15], linear classification errors [9], [16], [17] and hinge loss function [18]. With the assumption that samples in the same class tend to share some atoms, the structural sparsity information is explored to learn discriminative dictionaries [19], [20]. The above dictionary learning methods are linear and thus are inadequate for dealing with highly nonlinear datasets. To address this issue, several kernel dictionary learning (KDL) methods [21], [22], [23], [24], [25] are proposed for capturing sparse representation of nonlinear features. They achieve a better classification performance than the linear methods.

SDL methods generally need enough labeled samples to guarantee a good generalization performance by the learned dictionary. However, it is difficult to obtain sufficient labeled samples in practice. Hence, semi-supervised dictionary learning (SSDL) methods are proposed to deal with an insufficient number of labeled samples. Zhang *et al.* [26] develop an online semi-supervised discriminative dictionary learning (OSSDL) method, which incorporates reconstruction errors of labeled and unlabeled samples, label consistency and linear classification errors into a unified framework. Wang *et al.* [27] introduce an $\ell_{2,p}$ -norm regularization to sparse coefficients, and design a semi-supervised robust dictionary learning (SSR-DL) model. Wang *et al.* [28] propose a unified semi-supervised dictionary learning (USSDL) method, which jointly learns dictionary and classifier by adaptively estimating the confidence of unlabeled training samples. By using the structural sparse regularization on samples which potentially have the same class label, a novel model termed semi-supervised dictionary learning via structural sparse preserving (SSP-DL) [29] is presented. Since SSDL methods use labeled and unlabeled samples in the training process, they generally provide more competitive

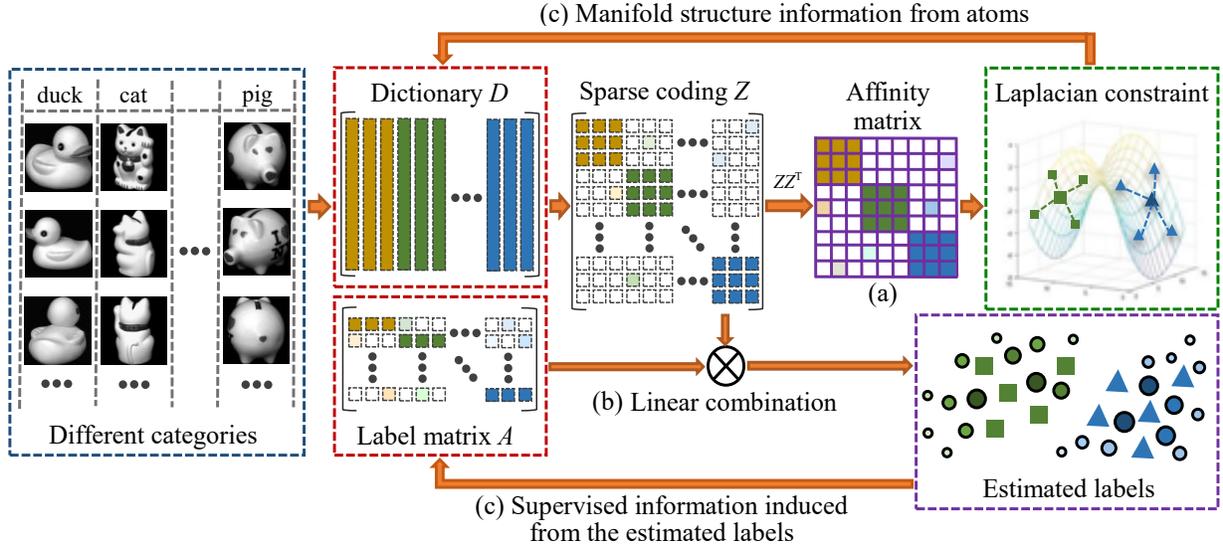


Fig. 1. The flowchart of our method. (a) The affinity matrix constructed from the sparse coefficients. (b) Labels estimated by linear combination of soft labels of atoms. (c) The feedback of supervised information hidden in unlabeled samples and manifold structure information from atoms.

performance than SDL methods. However, the aforementioned SSDL methods still have the following two limitations: 1) The intrinsic manifold structure of the training samples is not fully explored. 2) The supervised information hidden in unlabeled samples is not utilized in the training process.

Several recent methods [30], [31], [32] introduce a confidence matrix to estimate the relationship between the unlabeled samples and the classes, and iteratively feed back the estimated relationship to the training process to increase the discriminative capability of the learned dictionary. They achieve more impressive performance than the previous SSDL approaches for classification tasks. However, they still do not fully exploit the intrinsic manifold structure within the training samples. Moreover, these approaches learn multiple category-specific dictionaries and are not suitable for handling datasets with many classes.

This paper develops a new unified framework termed self-taught semi-supervised dictionary learning with non-negative constraint (NNST-SSDL), which integrates predictive errors of labeled samples and the graph Laplacian regularization into a common dictionary learning formulation. The main features of the proposed method are described as follows.

- We construct an atom graph Laplacian regularization to exploit the intrinsic manifold structure of training data. As atoms are more concise and representative than training samples, the atom graph Laplacian regularization is more robust to noise and outliers. Moreover, the atom graph Laplacian regularization can be iteratively refined by updating the sparse coefficients in the training process.
- The non-negative constraint is imposed on the sparse coefficients. This ensures that each sample is in the middle of its related atoms and thus facilitates learning local manifold structures. Based on this geometric relation, the label of a sample predicted by the linear combination of the atom's soft labels is reliable.
- The *self-taught* mechanism guarantees that the dictio-

nary and the graph Laplacian are learned in parallel. More specifically, the supervised information hidden in unlabeled samples and the manifold information induced by atom graph Laplacian regularization are efficiently fed back to improve the discriminative capability of the learned dictionary.

- An efficient algorithm combining a block coordinate descent method with the alternating directions method of multipliers is proposed to solve the unified framework. Experimental results on several benchmark datasets demonstrate that the proposed method is significantly better than the state-of-the-art approaches.

The remainder of this paper is organized as follows. Section II gives the motivation of the proposed method. Section III introduces the proposed NNST-SSDL framework, which includes the optimization algorithm and the computational complexity analysis. The experiments are reported in Section IV. Section V concludes the paper.

II. MOTIVATION

The task of semi-supervised dictionary learning is to learn an m atoms dictionary $D = [\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_m] \in \mathbb{R}^{d \times m}$ based on a K -class training dataset containing l labeled samples and u unlabeled samples $\mathcal{S} = \{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \dots, (\mathbf{x}_l, \mathbf{y}_l), \mathbf{x}_{l+1}, \mathbf{x}_{l+2}, \dots, \mathbf{x}_{l+u}\}$, where \mathbf{x}_i ($i = 1, 2, \dots, l+u$) are the samples and \mathbf{y}_i ($i = 1, 2, \dots, l$) is the label of sample \mathbf{x}_i . Here, \mathbf{y}_i is a 1-hot vector paradigm, i.e., if sample \mathbf{x}_i belongs to the k th class, then the k th element of \mathbf{y}_i is 1 and the remaining elements are 0. We stack the training samples as a matrix $X = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{l+u}] \in \mathbb{R}^{d \times (l+u)}$, where d is the feature dimension of samples. Accordingly, X_l and X_u are the data matrices of the labeled and unlabeled training samples. We also stack the labels as a matrix $Y_l = [\mathbf{y}_1, \dots, \mathbf{y}_l] \in \mathbb{R}^{K \times l}$. Denote $Z = [\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_{l+u}] \in \mathbb{R}^{m \times (l+u)}$ as the sparse coefficient matrix, where \mathbf{z}_i is the sparse coefficient vector of the training sample \mathbf{x}_i on dictionary D .

Specially, the i th row of the sparse coefficient matrix Z : \hat{z}^i , is called the profile of the atom \mathbf{d}_i [33]. Based on this concept, the relationship between samples and atoms can be re-written as

$$X \approx \mathbf{d}_1 \hat{z}^1 + \cdots + \mathbf{d}_i \hat{z}^i + \cdots + \mathbf{d}_j \hat{z}^j + \cdots + \mathbf{d}_m \hat{z}^m. \quad (1)$$

In this way, we can find that the influence of the atom \mathbf{d}_i for reconstructing the training samples can be measured by the corresponding profile \hat{z}^i .

The motivation of this work is that we not only use the label information but also the manifold structure for dictionary learning. Accordingly, the first task is to use the label information for dictionary learning. Different from the traditional multiple category-specific dictionary learning methods which use hard labels for atoms, we assign soft labels to atoms for the following reasons 1) The soft labels are more suited to the manifold assumption than hard labels; 2) atoms cannot be assigned to specific classes because samples from different classes may share atoms. The soft label matrix for the dictionary is denoted by $A = [\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_m] \in \mathbb{R}^{K \times m}$, in which \mathbf{a}_j is the soft label of the atom \mathbf{d}_j and K is the class number. Based on the assumption that the way the atoms reconstruct the sample should resemble the way the labels of atoms represent the label of the sample, the label \mathbf{y}_i of sample \mathbf{x}_i is estimated as a linear combination of the atom's soft labels as follows: $\mathbf{y}_i = A\mathbf{z}_i$. Therefore, as shown in the bottom of Fig. 1, a label constraint for labeled samples is formulated as $\|AZ_i - Y_i\|_F^2$.

Since similar profiles force the corresponding atoms to be similar, and the similar atoms in turn tend to have similar profiles, we define an affinity matrix as $W = ZZ^\top$. Obviously, the affinity matrix W can be iteratively refined by the updated Z during the training process. Inspired by manifold learning, an atom graph Laplacian regularization based on W is defined as follows

$$\frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m w_{ij} \|\mathbf{a}_i - \mathbf{a}_j\|_2^2 = \text{tr}(ALA^\top) \quad (2)$$

where $L = \Lambda - W$ is the Laplacian matrix, and Λ is a diagonal matrix whose i th diagonal element is the summation of the i th row of W , i.e., $\Lambda_{ii} = \sum_{j=1}^m w_{ij}$. The element L_{ij} corresponding to nearby atoms $\mathbf{d}_i, \mathbf{d}_j$ should be negative, which means these atoms tend to have similar labels; the elements corresponding to dissimilar pairwise atoms should be zero, which means their labels are independent. In our work, the atom graph regularization is used instead of the sample graph regularization because atoms are more concise and thus more robust to noise and outliers than samples, and meanwhile they can also inherit the manifold structure of the samples.

We constrain the sparse coefficients to be non-negative to ensure that each sample is in the middle of its related atoms. This better embodies the dependency between samples and atoms and facilitates learning local manifold structures. To more clearly illustrate the geometric relation between samples and their corresponding atoms, we present a synthetic example in \mathbb{R}^3 . As shown in Fig. 2, suppose a sample \mathbf{x} is approximately and sparsely represented by the atoms $\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3$ and \mathbf{d}_4 with non-negative coefficients, i.e., \mathbf{x} is in

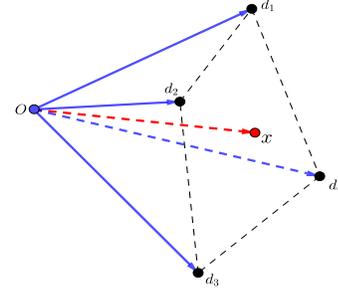


Fig. 2. Illustration for the geometric relation between samples and their corresponding atoms with the non-negative constraint.

the convex cone of set $\{\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3, \mathbf{d}_4\}$, then the sample \mathbf{x} can be enclosed by the atoms $\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3$ and \mathbf{d}_4 . Based on this geometric relation, the label of \mathbf{x} , as predicted by the linear combination of the atom's soft labels, is reasonable and reliable. Therefore, the sparse coefficient matrix Z is required to be non-negative.

As illustrated in Fig. 1, it is remarkable that the manifold structure and the estimated labels of unlabeled samples can be viewed as useful information. If we feed back this useful information to the dictionary learning process, it is possible to improve the discriminative capability of the learned dictionary, this feedback mechanism is called *self-taught*.

III. NNST-SSDL ALGORITHM

Motivated by the above idea, we propose a unified framework which fully exploits the latent manifold structure among samples in dictionary learning. The formulation is

$$\begin{aligned} \min_{D \in \mathcal{C}, A, Z} \quad & \frac{1}{2} \|X - DZ\|_F^2 + \lambda \|Z\|_1 + \frac{\beta}{2} \|AZ_l - Y_l\|_F^2 + \gamma \text{tr}(ALA^\top) \\ \text{s.t.} \quad & Z \geq 0. \end{aligned} \quad (3)$$

Here, $X = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{l+u}]$, $Y_l = [\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_l]$, $Z = [Z_l, Z_u]$ where Z_l and Z_u are respectively the sparse coefficient matrices of labeled and unlabeled samples, $\mathcal{C} = \{D = [\mathbf{d}_1, \mathbf{d}_2, \cdots, \mathbf{d}_m] \in \mathbb{R}^{d \times m}, \text{s.t. } \mathbf{d}_j^\top \mathbf{d}_j \leq 1, \forall j = 1, 2, \cdots, m\}$. λ, β and γ are trade-off parameters.

In model (3), the supervised term $\|AZ_l - Y_l\|_F^2$ enforces labeled samples in the same class to have similar sparse coefficients. The graph Laplacian regularization $\text{tr}(ALA^\top)$ enforces the atoms with similar soft labels to have similar profiles. This propagates the supervised information from labeled sparse coefficients to unlabeled sparse coefficients. In the training iterations, on one hand, the updated sparse coefficients are fed back to refine the affinity matrix and thus improve the estimation of the latent manifold structure, on the other hand, the accurate estimation of the manifold further increases the discriminative capability of the learned dictionary and the sparse coefficients. In other words, the feedback of sparse coefficients and manifold structure can be viewed as a process of information gain by self-training, which is the *self-taught* mechanism hidden in the proposed framework (3).

Compared with existed works, the proposed framework has the following differences.

- The atom's labels are soft labels, which are obtained by optimization and thus are more suitable for the manifold assumption than hard labels. The atoms are not allocated to any single class because samples from different classes may share atoms. In some other multiple category-specific dictionary learning methods (such as FDDL [15], LCKSVD [17] and S2D2 [30]), each atom is allocated to a specific class. The label of each atom is not changed during training process.
- The graph Laplacian regularization $\text{tr}(ALA^\top)$ constructed by atom's labels and sparse coefficients explicitly exploits the underlying manifold structure of the samples. It propagates the supervised information from labeled samples to unlabeled samples. Furthermore, the affinity matrix can be iteratively refined to give a more accurate estimate of the manifold structure in the training process. In contrast, earlier works scarcely consider the structural relationships among samples and the supervised information hidden in unlabeled samples.
- In the proposed framework, the labels of samples are reconstructed by the linear combination of atom's soft labels via the corresponding sparse coefficients. The supervised term $\|AZ_l - Y_l\|_F^2$ measures the reconstruction error for labels. This is different from the classification error for labeled samples in some earlier works (such as DKSVL [16], LCKSVD [17], OSSDL [26] and USSDL [32]) with linear classifiers.

A. Optimization of NNST-SSDL

The minimization (3) is non-convex and non-smooth. Optimizing over the variables Z, A, D simultaneously could be expensive in practice. In the following, we iteratively optimize the three variables using the block coordinate descent (BCD) method.

1) *Computation of Z* : On fixing D and A , the subproblem for the sparse coefficient matrix Z is formulated as

$$\begin{aligned} \min_Z \quad & \frac{1}{2} \|X - DZ\|_F^2 + \lambda \|Z\|_1 + \frac{\beta}{2} \|AZ_l - Y_l\|_F^2 + \gamma \|(ZZ^\top) \odot \Theta\|_1 \\ \text{s.t.} \quad & Z \geq 0, \end{aligned} \quad (4)$$

since

$$\text{tr}(ALA^\top) = \sum_{i,j} w_{ij} (\|\mathbf{a}_i - \mathbf{a}_j\|_2^2 / 2) = \|(ZZ^\top) \odot \Theta\|_1, \quad (5)$$

where $\Theta_{ij} = \frac{1}{2} \|\mathbf{a}_i - \mathbf{a}_j\|_2^2$, and \odot is the Hadamard product.

It is difficult to directly optimize (4) due to the interdependent terms with respect to the variable Z . To remove these interdependencies and optimize these terms independently, we introduce four auxiliary matrices P, Q_l, Q_u and B , and rewrite (4) as:

$$\begin{aligned} \min_{Z,P,Q_l,Q_u,B} \quad & \frac{1}{2} \|X_l - DQ_l\|_F^2 + \frac{1}{2} \|X_u - DQ_u\|_F^2 \\ & + \lambda \|Z\|_1 + \frac{\beta}{2} \|AQ_l - Y_l\|_F^2 + \gamma \|B \odot \Theta\|_1 \\ \text{s.t.} \quad & B = PQ^\top, \quad Z = P, \quad Z = Q, \\ & P = [P_l, P_u], Q = [Q_l, Q_u], Z \geq 0. \end{aligned} \quad (6)$$

The augmented Lagrangian function is

$$\begin{aligned} L_\mu(Z, P, Q_l, Q_u, B, \Lambda_1, \Lambda_2, \Lambda_3) \\ = \quad & \frac{1}{2} \|X_l - DQ_l\|_F^2 + \frac{1}{2} \|X_u - DQ_u\|_F^2 + \lambda \|Z\|_1 \\ & + \frac{\beta}{2} \|AQ_l - Y_l\|_F^2 + \gamma \|B \odot \Theta\|_1 + \langle B - PQ^\top, \Lambda_1 \rangle \\ & + \langle Z - P, \Lambda_2 \rangle + \langle Z - Q, \Lambda_3 \rangle + \frac{\mu}{2} \|B - PQ^\top\|_F^2 \\ & + \frac{\mu}{2} \|Z - P\|_F^2 + \frac{\mu}{2} \|Z - Q\|_F^2. \end{aligned} \quad (7)$$

Here, the matrices $\Lambda_1, \Lambda_2, \Lambda_3$ are Lagrange multipliers, and $\mu > 0$ is a penalty parameter which is adjusted by the adaptive strategy in [34].

The alternating direction method of multipliers (ADMM) [35], [36] is effective for solving optimization problems with multiple terms. Based on the scheme of ADMM, problem (6) can be iteratively solved by the following steps.

- Update Z :

$$\begin{aligned} Z^{k+1} &= \arg \min_{Z \geq 0} L_\mu(Z, P^k, Q_l^k, Q_u^k, B^k, \Lambda_1^k, \Lambda_2^k, \Lambda_3^k) \\ &= \arg \min_{Z \geq 0} \frac{1}{2} \|Z - S^k\|_F^2 + \frac{\lambda}{2\mu} \|Z\|_1, \end{aligned} \quad (8)$$

where $S^k = (\mu P^k + \mu Q^k - \Lambda_2^k - \Lambda_3^k) / (2\mu)$. For Eq. (8) without constraint $Z \geq 0$, the optimal solution is $D_{\frac{\lambda}{2\mu}}(S^k)$, in which $D_\eta(\cdot)$ is the shrinkage operator,

$$D_\eta(x) = \text{sgn}(x) \max(|x| - \eta, 0). \quad (9)$$

Then the solution of (8) can be formulated as

$$Z^{k+1} = \max\left(0, D_{\frac{\lambda}{2\mu}}(S^k)\right). \quad (10)$$

- Update P :

$$\begin{aligned} P^{k+1} &= \arg \min_P L_\mu(Z^{k+1}, P, Q_l^k, Q_u^k, B^k, \Lambda_1^k, \Lambda_2^k, \Lambda_3^k) \\ &= \arg \min_P \langle \Lambda_1^k, B^k - P(Q^k)^\top \rangle + \langle \Lambda_2^k, Z^{k+1} - P \rangle \\ & \quad + \frac{\mu}{2} \|Z^{k+1} - P\|_F^2 + \frac{\mu}{2} \|B^k - P(Q^k)^\top\|_F^2 \\ &= \left[\left(B^k + \frac{1}{\mu} \Lambda_1^k \right) Q^k + Z^{k+1} + \frac{1}{\mu} \Lambda_2^k \right] \left((Q^k)^\top Q^k + I \right)^{-1}. \end{aligned} \quad (11)$$

- Update Q :

$$\begin{aligned} Q_l^{k+1} &= \arg \min_{Q_l} L_\mu(Z^{k+1}, P^{k+1}, Q_l, Q_u^k, B^k, \Lambda_1^k, \Lambda_2^k, \Lambda_3^k) \\ &= \arg \min_{Q_l} \frac{1}{2} \|X_l - DQ_l\|_F^2 + \frac{\beta}{2} \|AQ_l - Y_l\|_F^2 \\ & \quad + \frac{\mu}{2} \left\| B^k - P_l^{k+1} Q_l^\top - P_u^{k+1} (Q_u^k)^\top + \frac{\Lambda_1^k}{\mu} \right\|_F^2 \\ & \quad + \frac{\mu}{2} \left\| Z_l^{k+1} - Q_l + \frac{\Lambda_{3(l)}^k}{\mu} \right\|_F^2 \\ &= M^\top \left[(MCL^\top) ./ (\xi \mathbf{1}_l^\top + \mathbf{1}_m \sigma^\top) \right] L. \end{aligned} \quad (12)$$

In (12), $C = D^\top X_l + \beta A^\top Y_l + \mu (B^k - P_u^{k+1} (Q_u^k)^\top)^\top P_l^{k+1} + \mu Z_l^{k+1} + (\Lambda_1^k)^\top P_l^{k+1} + \Lambda_{3(l)}^k$, and $\Lambda_{3(l)}^k$ is the sub-matrix of Λ_3^k corresponding to the labeled samples. Let $F = \mu (P_l^{k+1})^\top P_l^{k+1} + \mu I$

and $E = D^\top D + \beta A^\top A$, then there are orthogonal matrices M , L and diagonal matrices Ξ , Σ such that $E = M^\top \Xi M$ and $F = L^\top \Sigma L$ because of the positive semi-definiteness of E and F . ξ and σ are column vectors whose elements are the diagonal elements of the matrices Ξ and Σ respectively. $\mathbf{1}_l \in \mathbb{R}^{l \times 1}$ and $\mathbf{1}_m \in \mathbb{R}^{m \times 1}$ are vectors whose elements are equal to 1. ‘./’ stands for the operator of the element-by-element division. The optimization of Q_u is similar to that of Q_l , and it is skipped due to space limits. The detailed derivation is given in the ‘Supplementary Material’.

- Update B :

$$\begin{aligned} B^{k+1} &= \arg \min_B L_\mu(Z^{k+1}, P^{k+1}, Q_l^{k+1}, Q_u^{k+1}, B, \Lambda_1^k, \Lambda_2^k, \Lambda_3^k) \\ &= \arg \min_B \frac{1}{2} \|B - O^k\|_F^2 + \frac{\gamma}{\mu} \|B \odot \Theta\|_1 \\ &= \arg \min_{b_{ij}} \left\{ \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m (b_{ij} - o_{ij}^k)^2 + \frac{\gamma}{\mu} \sum_{i=1}^m \sum_{j=1}^m |\Theta_{ij} b_{ij}| \right\} \\ &= \arg \min_{b_{ij}} \left\{ \sum_{i=1}^m \sum_{j=1}^m \left[\frac{1}{2} (b_{ij} - o_{ij}^k)^2 + \frac{\gamma \Theta_{ij}}{\mu} |b_{ij}| \right] \right\}, \end{aligned} \quad (13)$$

where $O^k = P^{k+1}(Q^{k+1})^\top - \frac{1}{\mu} \Lambda_1^k$. The last equation follows from $\Theta_{ij} = \frac{1}{2} \|\mathbf{a}_i - \mathbf{a}_j\|_2^2 > 0$. It can be seen that the elements b_{ij} of B in optimization (13) are independent, and then B can be obtained by individually optimizing the following subproblem with respect to each element b_{ij} ,

$$b_{ij}^{k+1} = \arg \min_{b_{ij}} \frac{1}{2} (b_{ij} - o_{ij}^k)^2 + \frac{\gamma \Theta_{ij}}{\mu} |b_{ij}| = D_{\frac{\gamma \Theta_{ij}}{\mu}}(o_{ij}^k), \quad (14)$$

where $D_\eta(\cdot)$ is the shrinkage operator as shown in (9). Accordingly, variable B can be updated by

$$B^{k+1} = D_{\frac{\gamma \Theta}{\mu}}(O^k). \quad (15)$$

- Update the Lagrange multipliers matrices:

$$\begin{cases} \Lambda_1^{k+1} = \Lambda_1^k + \mu (B^{k+1} - P^{k+1}(Q^{k+1})^\top), \\ \Lambda_2^{k+1} = \Lambda_2^k + \mu (Z^{k+1} - P^{k+1}), \\ \Lambda_3^{k+1} = \Lambda_3^k + \mu (Z^{k+1} - Q^{k+1}). \end{cases} \quad (16)$$

2) *Computation of A* : When D and Z are fixed, the subproblem for A is

$$\min_A \frac{\beta}{2} \|AZ_l - Y_l\|_F^2 + \gamma \text{tr}(ALA^\top). \quad (17)$$

By setting the derivative of (17) to zero, the closed-form solution for A can be obtained by

$$A = \beta Y_l Z_l^\top (\beta Z_l Z_l^\top + 2\gamma L)^{-1}. \quad (18)$$

3) *Computation of D* : When A and Z are fixed, the optimization problem for D can be formulated as

$$\min_{D \in \mathcal{C}} \frac{1}{2} \|X - DZ\|_F^2. \quad (19)$$

By introducing the auxiliary variable G and defining the function $I_{\mathcal{C}}(G)$ as

$$I_{\mathcal{C}}(G) = \begin{cases} 0 & \text{if } G \in \mathcal{C}, \\ +\infty & \text{otherwise,} \end{cases} \quad (20)$$

Algorithm 1: BCD scheme for solving the problem (3).

Input: Samples matrix $X = [X_l, X_u]$, label matrix Y_l , and parameters $\lambda, \beta, \gamma > 0$.

- 1: Initialize the dictionary $D^{(0)}$, the label matrix $A^{(0)}$ of atoms and the sparse coefficient matrix $Z^{(0)}$. Let $t = 0$.
- 2: **while** not converged **do**
- 3: Compute $Z^{(t+1)}$ by using ADMM steps (8)-(16).
- 4: Compute $A^{(t+1)}$ by using (18).
- 5: Compute $D^{(t+1)}$ by using ADMM steps (23)-(25).
- 6: Let $t = t + 1$.
- 7: **end while**

Output: $D = D^{(t)}$, $Z = Z^{(t)}$, $A = A^{(t)}$.

the problem (19) can be transferred to

$$\begin{aligned} \min_{D, G} \quad & \frac{1}{2} \|X - DZ\|_F^2 + I_{\mathcal{C}}(G) \\ \text{s.t.} \quad & D = G. \end{aligned} \quad (21)$$

Next, we form the augmented Lagrangian function

$$f_\mu(D, G, \Lambda) = \frac{1}{2} \|X - DZ\|_F^2 + I_{\mathcal{C}}(G) + \langle \Lambda, D - G \rangle + \frac{\mu}{2} \|D - G\|_F^2, \quad (22)$$

where the matrix Λ is a Lagrange multiplier and μ is a penalty parameter. We also use ADMM to iteratively solve the problem (21) by the following steps.

- For term D :

$$\begin{aligned} D^{(k+1)} &= \arg \min_D f_\mu(D, G^{(k)}, \Lambda^{(k)}) \\ &= (XZ^\top - \Lambda^{(k)} + \mu * G^{(k)})(ZZ^\top + \mu I)^{-1}. \end{aligned} \quad (23)$$

- For term G :

$$\begin{aligned} G^{(k+1)} &= \arg \min_H f_\mu(D^{(k+1)}, G, \Lambda^{(k)}) \\ &= \Pi_{\mathcal{C}} \left(D^{(k+1)} + \frac{1}{\mu} \Lambda^{(k)} \right), \end{aligned} \quad (24)$$

where $\Pi_{\mathcal{C}}$ is the projection operator on \mathcal{C} .

- Update the Lagrange multiplier matrix:

$$\Lambda^{(k+1)} = \Lambda^{(k)} + \mu(D^{(k+1)} - G^{(k+1)}). \quad (25)$$

Since problem (21) only has two block variables D and G , the global convergence can be guaranteed [35], [36].

B. Algorithm Description

Algorithm 1 summarizes the BCD scheme for solving the unified optimization problem (3). The given labels Y_l and the estimated soft labels A of atoms are combined together to update the sparse coefficient matrix Z which is used to refine the affinity matrix in the next iteration. In general, if the sparse coefficient matrix Z is distinguishable, it tends to yield a discriminative dictionary D and a good affinity matrix which makes the estimation of soft label matrix A more precise. The dictionary D and the precisely estimated A can together yield a more accurate sparse coefficient matrix Z . As a consequence, the performances of the sparse coefficient matrix Z , the soft label matrix A and the dictionary D can be boosted mutually during the training process.

Algorithm 2: NNST-SSDL

Input: Samples matrix $X = [X_l, X_u]$, label matrix Y_l , the integers ι, N , and parameters $\lambda, \beta, \gamma > 0$.

- 1: Initialize the dictionary $D^{(0)}$, soft label matrix $A^{(0)}$ of atoms and sparse coefficient matrix $Z^{(0)}$.
- 2: Initialize $X_l^{(0)} = X_l, X_u^{(0)} = X_u, Y_l^{(0)} = Y_l$, and $\kappa = 1$.
- 3: **while** $\kappa < N$ **do**
- 4: Calculate $D^{(\kappa)}, A^{(\kappa)}$ and $Z^{(\kappa)} = [Z_l^{(\kappa)}, Z_u^{(\kappa)}]$ via Algorithm 1.
- 5: Calculate the soft label matrix H of unlabeled samples via $H = A^{(\kappa)} Z_u^{(\kappa)}$.
- 6: Use H to calculate the entropy of the estimated labels and predict the classes of the unlabeled samples in $X_u^{(\kappa)}$.
- 7: From each predicted class, select ι unlabeled samples with minimal entropy values. Denote the selected sample matrix and its corresponding predicted label matrix by $X_s^{(\kappa)}$ and $Y_s^{(\kappa)}$ respectively.
- 8: Let $X_l^{(\kappa+1)} = [X_l^{(\kappa)}, X_s^{(\kappa)}], Y_l^{(\kappa+1)} = [Y_l^{(\kappa)}, Y_s^{(\kappa)}], X_u^{(\kappa+1)} = X_u^{(\kappa)} \setminus X_s^{(\kappa)}$, and $\kappa = \kappa + 1$.
- 9: **end while**

Output: $D = D^{(N)}, Z = Z^{(N)}, A = A^{(N)}$.

The scheme of the self-taught semi-supervised dictionary learning is described in Algorithm 2. The core idea is to select the most confident unlabeled samples with their predicted labels to augment the labeled set in the current iteration, and thus further improve the discriminative capability of the learned dictionary in the next iteration. In Algorithm 2, Step 2 initializes current labeled and unlabeled sets to the given labeled set X_l and unlabeled set X_u respectively. The iteration for incorporating the given labels and the estimated labels is presented from Step 3 to Step 9. Specifically, Step 4 utilizes the current labeled and unlabeled sets to train sparse coefficients $Z^{(\kappa)}$, dictionary $D^{(\kappa)}$ and soft labels $A^{(\kappa)}$ via Algorithm 1. The soft labels $H_u^{(\kappa)}$ of the current unlabeled samples are then estimated by the linear combination of the soft labels $A^{(\kappa)}$ via the corresponding sparse coefficients $Z_u^{(\kappa)}$ in Step 5. Step 6 utilizes the estimated soft labels $H_u^{(\kappa)}$ to calculate the entropy and predict the classes of unlabeled samples. From Step 7 to Step 8, we select ι unlabeled samples with minimal entropy values (i.e., the ι most confident unlabeled samples) to augment the current labeled set. These selected samples are removed from the current unlabeled set. It is remarkable that from the second iteration, the number of labeled samples keeps increasing. On average, an increase of the number of labeled samples is associated with an increase in the ability of the dictionary to discriminate between the different classes. Moreover, the sparse coefficients and soft labels of atoms in the current iteration are fed back to revise the graph Laplacian regularization in the next iteration, and hence can further enhance the discriminative capability of the learned dictionary.

From the inner loop (Algorithm 1) and the outer loop (Algorithm 2), the manifold information and the estimated labels of unlabeled samples can be iteratively fed back to boost the performance of the learned dictionary in training process. This is why the proposed method is said to be *self-taught*.

The BCD scheme used in Algorithm 1 guarantees that the objective value of (3) is decreasing as the number of iterations

increases. However, the convergence properties of the ADMM for minimizing an objective function that includes N ($N > 3$) block variables have remained unclear [35], [36]. Since the subproblem (6) has five block variables $\{Z, P, Q_l, Q_u, B\}$ and its objective function is non-smooth, it might be difficult to prove the convergence theoretically. Fortunately, the proposed method performs well and converges quickly in practice, as is shown in the experiments reported in Section IV.

Remark 1. *Traditional dictionary learning methods could also use the predicted labels to boost the classification performance. In fact, several SSDL methods have utilized techniques similar to the self-taught mechanism. For example, OSSDL [26] first introduces a probabilistic model over the sparse coefficients, and then computes the entropy based on the probabilistic model to quantify the confidence level of the discriminability for unlabeled samples. The samples whose entropy values are smaller than a predefined lower bound are automatically added to the labeled set for dictionary learning. The methods such as S2D2 [30], DSSDL [31] and SSD-LP [32] adopt a confidence matrix to estimate the probabilities of unlabeled samples over all classes, and iteratively feed back the estimated probabilities to the learning process and thus boost the classification accuracy. Of course, SDL methods could also benefit from the predicted labels of unlabeled samples. This will be verified by applying the feedback mechanism of predicted labels to the LC-KSVD method, as described in Section IV-E.*

C. Classification Strategy

Once D and A are obtained, the class of a test sample \mathbf{x} can be predicted by the following two steps:

Step 1: Calculate the sparse coefficients of \mathbf{x} over the learned dictionary D , i.e.,

$$\hat{\mathbf{z}} = \arg \min_{\mathbf{z}} \frac{1}{2} \|\mathbf{x} - D\mathbf{z}\|_2^2 + \lambda \|\mathbf{z}\|_1. \quad (26)$$

Step 2: Calculate $\mathbf{y} = A\hat{\mathbf{z}}$, and predict the class of the test sample \mathbf{x} via

$$i_0 = \arg \max_{i \in \{1, 2, \dots, K\}} \mathbf{y}_i. \quad (27)$$

D. Computational Complexity

We first review the symbols used below: d, m, l, u and K are the dimension of samples, the number of atoms, the numbers of labeled and unlabeled training samples, and the number of classes respectively. $n = l + u$ is the number of all training samples. The maximal iteration numbers in ADMM and BCD are respectively denoted by s and t . In general, $n > m > K, n > l$ and $n > u$.

In Algorithm 1, the computational complexity of Step 3 is $O(s(n^3 + dmn) + m^2d)$. Step 4 and Step 5 take $O(m^2n)$ and $O(s(dnm + m^2n))$ operations to update the atom's labels A and the dictionary D respectively. In summary, the overall computational complexity of Algorithm 1 is $O(ts(n^3 + dmn))$.

For the testing phase in Section III-C, calculating the sparse coefficients and the soft label take $O(sm^3 + sm^2d)$ and $O(Km)$ operations respectively. Thus, the total computational complexity for classification is $O(s(m^3 + m^2d))$ due to the fact that $K \leq m$.

TABLE I
OVERALL DESCRIPTION OF THE DATASETS AND THE OPTIMAL PARAMETERS USED IN THE PROPOSED METHODS.

Datasets	DIM	Data#	Class#	τ	ν	m	ST-SSDL			NNST-SSDL		
							λ	β	γ	λ	β	γ
ORL	1024	400	40	2	5	80	0.05	0.1	0.001	0.05	0.1	0.001
PIE	1024	2040	12	20	60	200	0.01	0.01	0.0001	0.005	0.01	0.0001
TDT2	500	1560	30	3/5/7	28/26/24	210	0.005	0.005	0.001	0.001	0.01	0.00005
UMIST	750	575	20	5	10	200	0.001	0.001	0.00005	0.001	0.001	0.0001
COIL-20	1521	1440	20	2~10	38~30	200	0.01	0.01	0.0001	0.01	0.01	0.0001
SBDdata	638	2000	40	5	15	200	0.05	0.005	0.0001	0.05	0.005	0.0001
E-YaleB	1024	2414	38	2/5/10	18/15/10	380	0.001	0.1	0.0001	0.001	0.01	0.001
UCF50	1500	6679	50	10/15/20	30/25/20	1000	0.001	0.001	0.0001	0.005	0.005	0.0001

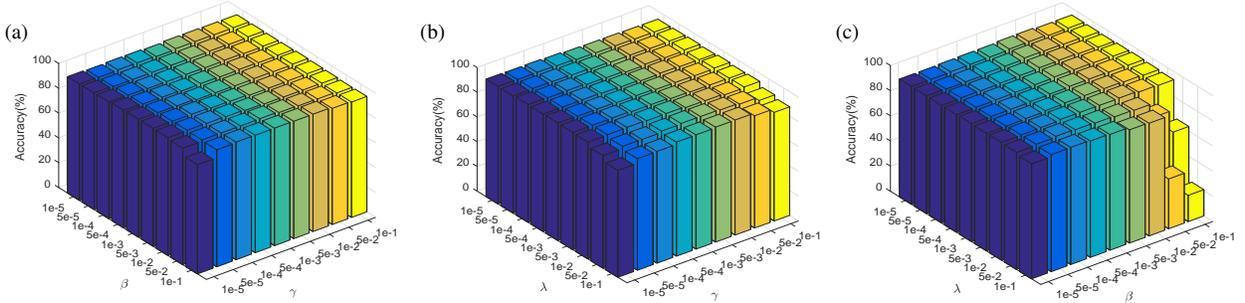


Fig. 3. Parameter sensitivity analysis of our method on TDT2 database, where (a) tune β and γ utilizing grid searching with fixed λ ; (b) tune λ and γ utilizing grid searching with fixed β ; (c) tune λ and β utilizing grid searching with fixed γ .

IV. EXPERIMENTAL RESULTS AND ANALYSIS

The experiments are conducted on seven image datasets ORL, PIE, UMIST, COIL-20, SBDdata, TDT2, Extended YaleB [29], [37], [38] and one action recognition dataset UCF50 [39]. Table I gives an overall description for these datasets. The competing methods include DKSVD [16], FDDL [15], LCKSVD [17], OSSDL [26], S2D2 [30], SSR-D [27], as well as three recently proposed SSDL methods SSP-DL [29], DSSDL [31] and SSD-LP [32]. To demonstrate the advantage of the non-negative constraint on sparse coefficients, we also perform experiments for the model of framework (3) without non-negative constraint, which is denoted by ST-SSDL. For the proposed methods (ST-SSDL and NNST-SSDL), K-means algorithm is utilized to initialize the dictionary. The parameters of all methods are selected by cross-validation. The optimal parameters of the proposed methods and the numbers of atoms are listed in Table I. For each class of a dataset, τ and ν samples are randomly selected as the labeled and unlabeled samples for training, while the rest samples are used for testing. The experiments are repeated 10 times with different random splits for each dataset, and the average accuracy together with the standard deviation are recorded. The best classification accuracies are shown in boldface (For more experiments, please refer to ‘*Supplementary Material*’).

A. Parameter Sensitivity

The sensitivity of the NNST-SSDL model to variations in the parameter values is investigated. There are three parameters in our model, so we fix one of them and explore the effects of the other two on the classification accuracy by grid search from

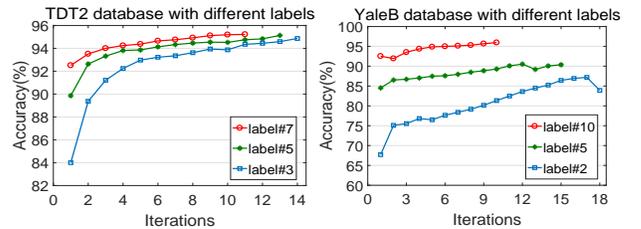


Fig. 4. Classification accuracy versus iteration number on the datasets TDT2 and Extended YaleB.

the candidate set $\{1e-5, 5e-5, \dots, 1e-1\}$ ($1e-5$ and $5e-5$ represent 10^{-5} and 5×10^{-5} respectively. The same notation is used below). We perform the experiment on the TDT2 dataset as an example. For each class of TDT2, we randomly select 7 samples as labeled samples, 24 samples as unlabeled samples, and the rest are left for testing. According to the optimal parameters for TDT2, we first fix $\lambda = 1e-3$, and tune β and γ by grid search from the candidate set. The classification accuracies are shown in Fig. 3 (a), where NNST-SSDL with $\beta \leq 5e-2$ and all γ can achieve desirable and stable results. Then, we fix $\beta = 1e-2$ to investigate the effects of λ and γ in Fig. 3 (b), from which we know that NNST-SSDL with $\lambda \leq 1e-2$ and all γ has satisfactory performance. Finally, we illustrate the effects of λ and β with fixed $\gamma = 5e-5$ in Fig. 3 (c). It can be seen that NNST-SSDL with $\beta \leq 1e-2$ and all λ yields satisfactory results. From the above analysis, it can be concluded that the classification performance of NNST-SSDL is robust to the three parameters over a wide range of values.

TABLE II
CLASSIFICATION ACCURACIES OF DIFFERENT DATASETS.

Datasets	DKSVD	FDDL	LCKSVD1	LCKSVD2	OSSDL	S2D2	SSR-D	SSP-DL	ST-SSDL	NNST-SSDL
ORL	80.17±4.59	84.92±1.69	78.92±3.09	80.50±2.81	67.50±7.69	82.58±3.10	77.50±3.14	82.33±3.09	85.92±3.04	86.83±5.06
PIE	74.58±2.73	96.94±0.94	90.53±5.14	92.67±2.79	94.68±4.77	79.43±1.80	94.70±1.37	94.85±1.03	97.56±1.27	98.10±0.73
UMIST	80.78±3.70	85.00±2.30	86.44±2.70	86.76±2.60	85.02±2.90	85.18±3.20	87.25±2.70	88.73±2.50	89.27±2.01	89.38±6.52
SBDData	49.71±5.71	58.38±4.94	56.82±5.21	56.82±5.21	53.86±2.20	56.97±1.90	63.21±6.69	64.56±5.80	65.21±4.74	65.33±4.48

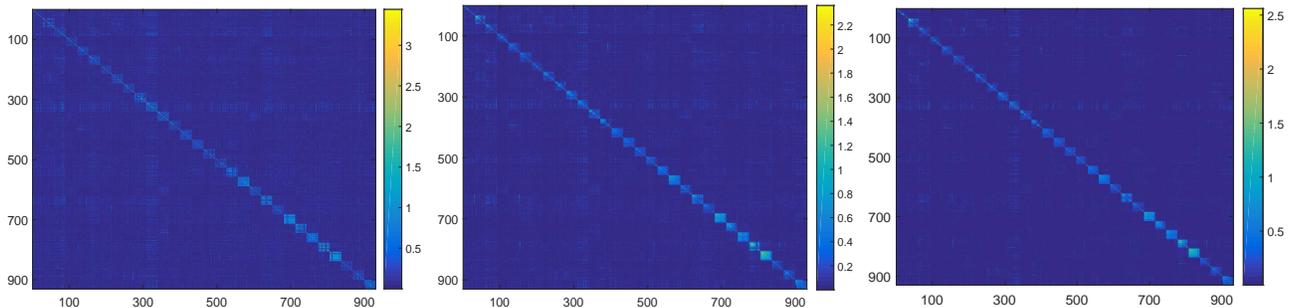


Fig. 5. The visualization of the affinity matrix on the dataset TDT2.

B. Image Classification

The classification accuracies on datasets ORL, PIE, UMIST and SBDData are reported in Table II, from which we conclude the following assertions: 1) SSDL methods usually achieve significantly better performance than SDL methods because of the utilization of the unlabeled samples. 2) The proposed ST-SSDL and NNST-SSDL outperform all other methods. This verifies that the performance of the learned dictionary can be improved by the feedback of the supervised information hidden in unlabeled samples and the manifold structure from the atom graph Laplacian regularization. 3) Although the classification performance of ST-SSDL is excellent, it can still be improved by NNST-SSDL, which confirms the advantage of imposing the non-negative constraint on the sparse coefficients.

Subsequently, classification accuracy versus iteration number on the datasets TDT2 and Extended YaleB with different numbers of labeled samples are shown in Fig. 4. It can be seen that the effectiveness of the feedback of the predicted labels is convincing, especially for small size of labeled set. We visualize the affinity matrix constructed by $Z^T Z$ on the dataset TDT2. From left column to right column in Fig. 5, the visualizations respectively represent the affinity matrix in the first iteration, the 9-th iteration and the last iteration. It can be seen that the block diagonal structure becomes more apparent as iterations proceed. This also validates that the discriminative capability of the learned dictionary can be boosted by the feedback of the predicted labels and the refined manifold structure.

To investigate the effect of the number of labeled samples on the classification accuracy, we conduct the experiments on COIL-20, TDT2 and Extended YaleB datasets with varying number of labeled samples. The classification results are

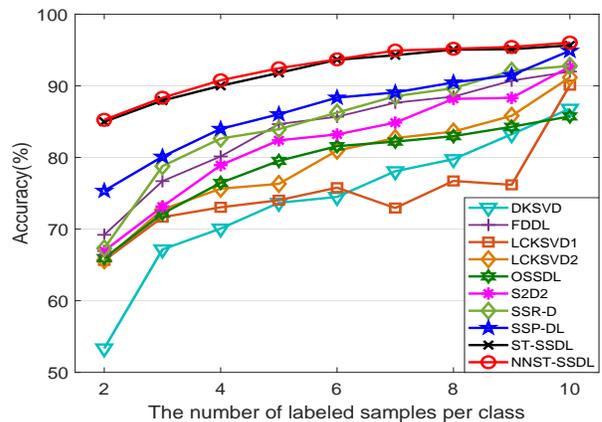


Fig. 6. Classification accuracies on the dataset COIL-20 for different numbers of labeled samples.

reported in Fig. 6, Table III and Table IV¹. We see that the classification accuracies increase with the number of labeled samples increasing for all methods. The proposed methods significantly outperform all other competitors, particularly when there are only a few labeled samples. This is another evidence that the learned dictionary can become more powerful through the *self-taught* mechanism.

Moreover, we study the classification performance on the datasets COIL-20 and Extended YaleB as the number of atoms varies. We randomly select 10 labeled samples from 40 training samples in each class of the COIL-20 dataset and vary the number of atoms in set $\{100, 120, \dots, 280\}$. For the Extended Yale-B dataset, we randomly select 10 labeled samples from 20 training samples in each class and vary the number of atoms in set $\{190, 228, \dots, 494\}$. The experimental

¹The results of DSSDL and SSD-LP are respectively referenced in [31] and [32].

TABLE III
CLASSIFICATION ACCURACIES ON THE TDT2 DATASET.

labeled #	3	5	7
DKSVD	89.79±1.66	91.65±1.24	92.65±0.94
FDDL	88.84±0.87	91.54±0.77	91.81±1.04
LCKSVD1	89.73±1.75	91.81±1.06	92.79±0.80
LCKSVD2	89.81±1.65	91.95±0.93	92.86±0.75
OSSDL	89.84±1.07	90.17±1.92	90.78±1.14
S2D2	88.35±1.69	90.24±1.20	90.86±1.28
SSR-D	83.97±1.95	87.37±1.84	87.98±1.36
SSP-DL	85.11±2.09	87.54±1.68	88.52±1.27
ST-SSDL	94.75±1.10	95.00±0.47	95.11±0.71
NNST-SSDL	94.86±0.75	95.13±0.74	95.22±0.72

TABLE IV
CLASSIFICATION ACCURACIES ON THE EXTENDED YALEB DATASET.

labeled #	2	5	10
DKSVD	52.57±8.64	67.98±6.58	88.33±4.68
FDDL	54.40±7.50	79.93±5.41	88.93±4.88
LCKSVD1	55.54±9.22	75.71±6.51	89.17±4.99
LCKSVD2	55.55±9.22	75.71±6.51	89.17±4.99
OSSDL	25.43±9.13	53.38±7.05	80.89±5.29
S2D2	58.24±1.68	84.92±2.77	92.26±2.68
SSR-D	45.96±9.89	75.26±5.75	88.04±4.54
SSP-DL	60.76±5.50	78.97±5.37	87.84±4.30
DSSDL	62.10±3.00	87.50±0.30	94.50±0.30
SSD-LP	67.20±2.90	89.80±0.90	95.20±0.20
ST-SSDL	85.21±5.99	90.45±1.53	95.52±0.91
NNST-SSDL	87.19±4.83	90.52±4.67	95.94±1.88

results are shown in Fig. 7. The number of atoms for DKSVD, LCKSVD, OSSDL and S2D2 is no more than the number of the labeled samples because their dictionaries are multiple category-specific. We can see that our methods achieve the best classification accuracies for all numbers of atoms. Besides, an interesting phenomenon is that the accuracies of our methods on the Extended YaleB dataset fall slightly when the number of atoms is larger than 380. The reason is that when the number of atoms becomes close to the number of training samples, the atoms inherit more information from the original images which contain noise (such as shadows and strips occlusion) in the Extended YaleB dataset. This experiment well validates the claim “the atom graph Laplacian regularization is more robust to noisy samples and outliers”.

C. Action Recognition

Action recognition is important for a wide range of applications, such as video surveillance, intelligent interface, sport video annotation, etc. [2]. In this subsection, we conduct experiments on one large action recognition dataset UCF50, which contains 50 action categories with a total of 6679 realistic videos taken from YouTube. Because of the high dimension of action features [39], we use PCA to reduce the feature dimension to 1500. The classification results are reported in Table V. We can see that NNST-SSDL also

TABLE V
CLASSIFICATION ACCURACIES ON THE UCF50 DATASET.

labeled #	10	15	20
DKSVD	35.40±2.57	39.00±1.35	43.45±1.78
LCKSVD1	36.03±6.35	41.10±5.39	45.48±1.45
LCKSVD2	36.14±1.30	42.65±1.28	45.92±1.23
FDDL	40.19±2.35	47.23±1.52	50.33±1.17
OSSDL	30.24±1.47	37.86±1.26	44.15±0.89
S2D2	28.34±0.87	28.86±1.09	30.68±0.66
SSR-D	38.93±1.09	46.14±1.21	49.11±1.28
SSP-DL	37.52±0.51	43.51±1.22	46.33±1.15
ST-SSDL	43.54±0.90	49.63±0.78	53.47±0.72
NNST-SSDL	43.63±1.25	49.83±1.11	54.42±1.05

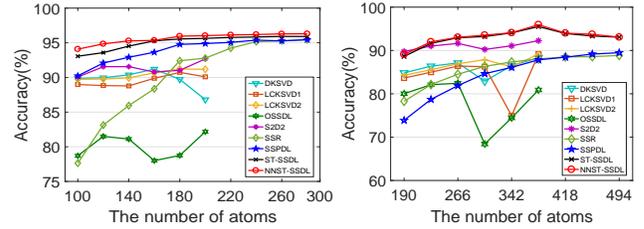


Fig. 7. Classification accuracies with different numbers of atoms on the COIL-20 dataset (left figure) and the Extended YaleB dataset (right figure).

achieves significant improvement compared with the other methods, which further demonstrates the effectiveness of the proposed method.

D. Analysis of Optimization Algorithm

In this part, we investigate the convergence properties of the BCD and ADMM methods used in Algorithm 1. Here we take the COIL-20 dataset as an example. Fig. 8 (a) shows that the value of the objective function for problem (3) drops as iterations proceed via the BCD method. A satisfying convergence trend is achieved within 30 iterations. We also define the following error terms which are related to the convergence conditions of the ADMM method for solving the sub-problem (6):

$$E_1 = \frac{\|B - PQ^\top\|_F^2}{\|X\|_F^2}, \quad E_2 = \frac{\|Z - P\|_F^2}{\|X\|_F^2}, \quad E_3 = \frac{\|Z - Q\|_F^2}{\|X\|_F^2}$$

From Fig. 8(b), it can be seen that all error curves drop quickly as the number of iterations increases. At most 30 iterations are required in all cases. This is a strong validation for the convergence of the proposed optimization algorithm. More convergence results for other datasets can be found in the ‘supplementary material’.

E. Benefit from the Predicted Labels

To more clearly demonstrate the benefit from the predicted labels of unlabeled samples for SDL methods, we also feed back the most confident unlabeled samples with their predicted labels to augment the labeled set for the training of LCKSVD2. This method is denoted by ST-LCKSVD2. The experimental results are recorded in Table VI. It is apparent that feedback

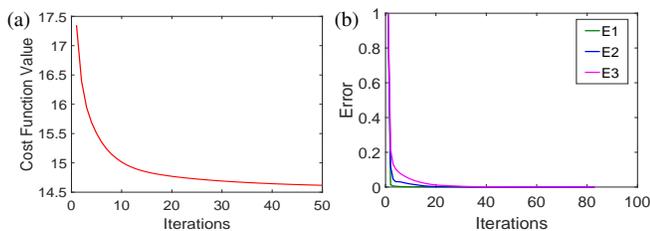


Fig. 8. The convergence of the (a) BCD and (b) ADMM methods.

TABLE VI
BENEFIT FROM THE PREDICTED LABELS.

Dataset	τ	LCKSVD2	ST-LCKSVD2	NNST-SSDL
ORL	2	80.50±2.81	82.50±4.32	86.83±5.06
PIE	20	92.67±2.79	96.94±1.00	98.10±0.73
UMIST	5	86.76±2.60	87.52±4.47	89.38±6.52
SBDData	5	56.82±5.21	57.48±4.92	65.33±4.48
COIL-20	2	65.64±6.80	72.36±6.45	85.25±3.32
	5	76.31±2.97	85.42±4.47	92.45±3.74
	10	91.17±3.89	91.92±3.49	96.03±3.83
TDT2	3	89.81±1.65	93.62±0.67	94.86±0.75
	5	91.95±0.93	93.75±0.52	95.13±0.74
	7	92.86±0.75	93.83±0.66	95.22±0.72
E-YaleB	2	55.55±9.22	57.92±5.59	87.19±4.83
	5	75.71±6.51	77.42±5.17	90.52±4.67
	10	89.17±4.99	89.43±4.93	95.94±1.88
UCF50	10	36.14±1.30	37.07±1.19	43.63±1.25
	15	42.65±1.28	44.34±1.15	49.83±1.11
	20	45.92±1.23	48.45±1.29	54.42±1.05

of predicted labels improves the classification performance of LC-KSVD2. However, NNST-SSDL still achieves the best classification accuracies. This is because NNST-SSDL fully exploits the intrinsic manifold structure of samples by constructing an atom graph Laplacian regularization which can be iteratively refined in the training process. In other words, the success of the proposed method is from two aspects. One is the “self-taught” mechanism, the other is the exploration of the manifold structure.

V. CONCLUSION

We propose a unified framework termed self-taught semi-supervised dictionary learning with non-negative constraint. In the framework, an atom graph Laplacian regularization is constructed to characterize the manifold structure of samples. The non-negative constraint imposed on the sparse coefficients ensures that each sample is in the middle of its related atoms, which is more consistent with the modeling of visual data and can facilitate learning local manifold structures. We propose the *self-taught* mechanism, which guarantees that the manifold structure induced by the atom Laplacian regularization and the supervised information hidden in unlabeled samples can be efficiently fed back to boost the discriminative capability of the learned dictionary. To solve the unified framework, we also derive an efficient algorithm by combining the block coordinate descent method with the alternating direction method of multipliers. Experiments on image classification and action

recognition show that the proposed model is a clear advance over the existing DL approaches.

ACKNOWLEDGMENTS

This work was supported in part by the National Key Research and Development Program of China [grant no. 2018YFB1004904], in part by the National Natural Science Foundation of China [grant no. 61772374], in part by the Natural Science Foundation of Zhejiang Province [grant nos. LY17F030004, LR17F030001, LQ19F020005], in part by the Project of Science and Technology Plans of Wenzhou City [grant nos. C20170008, G20160002, ZG2017016].

REFERENCES

- [1] D. Wu, X. Luo, G. Wang, M. Shang, Y. Yuan, and H. Yan, “A highly accurate framework for self-labeled semisupervised classification in industrial applications,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 3, pp. 909–920, 2018.
- [2] X. Cao, B. Ning, P. Yan, and X. Li, “Selecting key poses on manifold for pairwise action recognition,” *IEEE Transactions on Industrial Informatics*, vol. 8, no. 1, pp. 168–177, Feb 2012.
- [3] T. Bai and Y. Li, “Robust visual tracking using flexible structured sparse representation,” *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 538–547, Feb 2014.
- [4] X. Chang, Y. L. Yu, Y. Yang, and E. P. Xing, “Semantic pooling for complex event analysis in untrimmed videos,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 8, pp. 1617–1632, 2017.
- [5] X. Chang, F. Nie, S. Wang, Y. Yang, X. Zhou, and C. Zhang, “Compound rank-k projections for bilinear analysis,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 7, pp. 1502–1513, 2016.
- [6] G. E. Hinton, S. Osindero, and Y. W. Teh, “A fast learning algorithm for deep belief networks,” *Neural Computation*, vol. 18, pp. 1527–1554, 2006.
- [7] Z. Chen, L. Zhang, Z. Cao, and J. Guo, “Distilling the knowledge from handcrafted features for human activity recognition,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4334–4342, 2018.
- [8] X. Jiang, J. Sun, C. Li, and H. Ding, “Video image defogging recognition based on recurrent neural network,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 7, pp. 3281–3288, 2018.
- [9] Z. Zhang, W. Jiang, F. Li, M. Zhao, B. Li, and L. Zhang, “Structured latent label consistent dictionary learning for salient machine faults representation-based robust classification,” *IEEE Transactions on Industrial Informatics*, vol. 13, no. 2, pp. 644–656, April 2017.
- [10] W. Jiang, Z. Zhang, F. Li, L. Zhang, M. Zhao, and X. Jin, “Joint label consistent dictionary learning and adaptive label prediction for semisupervised machine fault classification,” *IEEE Transactions on Industrial Informatics*, vol. 12, no. 1, pp. 248–256, 2016.
- [11] J. Wright, M. Yang, A. Ganesh, S. Sastry, and Y. Ma, “Robust face recognition via sparse representation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 210–227, 2009.
- [12] M. Aharon, M. Elad, and A. Bruckstein, “K-svd: An algorithm for designing over-complete dictionaries for sparse representation,” *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.
- [13] J. Marial, F. Bach, J. Ponce, and G. Sapiro, “Online dictionary learning for sparse coding,” in *Proceedings of Annual International Conference on Machine Learning*, New York, NY, 2009, pp. 689–696.
- [14] J. Mairal, F. Bach, and J. Ponce, “Task-driven dictionary learning,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 791–804, 2012.
- [15] M. Yang, L. Zhang, X. Feng, and D. Zhang, “Fisher discrimination dictionary learning for sparse representation,” in *IEEE International Conference on Computer Vision*, Providence, RI, 2011, pp. 543–550.
- [16] Q. Zhang and B. Li, “Discriminative k-svd for dictionary learning in face recognition,” in *IEEE Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, 2010, pp. 2691–2698.
- [17] Z. Jiang, Z. Lin, and L. S. Davis, “Learning a discriminative dictionary for sparse coding via label consistent k-svd,” in *IEEE International Conference on Computer Vision*, Providence, RI, 2011, pp. 1697–1704.
- [18] X. Lian, Z. Li, B. Lu, and L. Zhang, “Max-margin dictionary learning for multiclass image categorization,” in *The eleventh European Conference on Computer Vision*, Crete, Greece, 2010, pp. 157–170.

