Support Vector Guided Dictionary Learning

Sijia Cai[1,3], Wangmeng Zuo[2], Lei Zhang[3⋆], Xiangchu Feng[4], and Ping Wang[1]

[1]School of Science, Tianjin University
[2]School of Computer Science and Technology, Harbin Institute of Technology
[3]Dept. of Computing, The Hong Kong Polytechnic University
[4]Dept. of Applied Mathematics, Xidian University
cssjcai@gmail.com,cslzhang@comp.polyu.edu.hk

Abstract. Discriminative dictionary learning aims to learn a dictionary from training samples to enhance the discriminative capability of their coding vectors. Several discrimination terms have been proposed by assessing the prediction loss (e.g., logistic regression) or class separation criterion (e.g., Fisher discrimination criterion) on the coding vectors. In this paper, we provide a new insight on discriminative dictionary learning. Specifically, we formulate the discrimination term as the weighted summation of the squared distances between all pairs of coding vectors. The discrimination term in the state-of-the-art Fisher discrimination dictionary learning (FDDL) method can be explained as a special case of our model, where the weights are simply determined by the numbers of samples of each class. We then propose a parameterization method to adaptively determine the weight of each coding vector pair, which leads to a support vector guided dictionary learning (SVGDL) model. Compared with FDDL, SVGDL can adaptively assign different weights to different pairs of coding vectors. More importantly, SVGDL automatically selects a few critical pairs to assign non-zero weights, resulting in better generalization ability for pattern recognition tasks. The experimental results on a series of benchmark databases show that SVGDL outperforms many state-of-the-art discriminative dictionary learning methods.

Keywords: Dictionary learning, support vector machine, sparse representation, Fisher discrimination

1 Introduction

Sparsity has become an appealing concept for data representation and it has been successfully applied in a variety of fields, e.g., compressed sensing [1], image restoration [2, 3], subspace clustering [4] and image classification [5, 6], etc. In sparse representation, a signal is approximated by the linear combination of a few bases sparsely selected from an over-complete set of atoms, i.e., a dictionary. Such a sparse coding strategy can be explained from the perspective of neuroscience [7] and it brings some desirable properties for signal reconstruction [8]. In sparse representation, the dictionary can be simply predefined as some

⋆ Corresponding author.

off-the-shelf dictionaries such as wavelets [9], but it has been demonstrated that learning a dictionary from exemplar images can lead to much better signal reconstruction performance [10]. Some typical reconstructive dictionary learning methods include K-means, method of optimal direction (MOD) [11], K-SVD [10] and analysis K-SVD [12].

Sparse representation can also be used for effective pattern recognition. The sparse representation based classification (SRC) has achieved competitive performance on face recognition [13]. Moreover, sparse coding as a soft vector quantization technique [14] adopted in Bag-of-Words based image representation [15] has also been recognized as a thought-provoking idea in image classification [5, 6]. Similar to signal reconstruction, in pattern classification, a discriminative dictionary learned from given examples can also improve much the performance.

A number of discriminative dictionary learning (DDL) methods [16–25] have been proposed. One type of the DDL methods dedicate to improving the discriminative capability of signal reconstruction residual. Rather than learning a dictionary for all classes, these methods exploit structural assumption on dictionary design and impose the learned dictionary with category-specific property, e.g., learning a sub-dictionary for each class [18, 22, 23]. Ramirez *et al.* [22] introduced the structured incoherence term to promote the independence of the sub-dictionaries associated with different classes. Gao *et al.* [23] learned both the category-specific sub-dictionaries and a shared dictionary for fine-grained image categorization. However, these dictionary learning methods might not be scalable to the problems with a large number of classes.

Another type of DDL methods aim to seek the optimal dictionary to improve the discriminative capability of coding vectors. These methods learn concurrently a dictionary and a classifier by incorporating some prediction loss on the coding vectors. In this spirit, Zhang *et al.* [16] extended the original K-SVD algorithm by simultaneously learning a linear classifier. Jiang *et al.* [19, 20] introduced a label consistent regularization to enforce the discrimination of coding vectors. The so-called LC-KSVD algorithm exhibits good classification results. Mairal *et al.* [17] proposed a supervised dictionary learning scheme by exploiting logistic loss function and further presented a general task-driven dictionary learning (TDDL) framework [21]. Wang *et al.* [25] formulated the dictionary learning problem from a max-margin perspective and learned the dictionary by using a multi-class hinge loss function. By considering the discrimination from both reconstruction residual and coding vectors, Yang *et al.* [18] proposed a Fisher discrimination dictionary learning (FDDL) method, where the category-specific strategy is adopted for learning a structural dictionary and the Fisher discrimination criterion is imposed on the coding vectors to enhance class discrimination.

In most of the above DDL methods, the discrimination of the learned dictionary is enforced by either imposing structural constraints on dictionary or imposing a discrimination term on the coding vectors. Several discrimination terms have been proposed by assessing the prediction loss (e.g., logistic regression) or class separation criterion (e.g., Fisher discrimination criterion) on coding vectors [16–20, 22]. In this paper, we provide a new scheme for DDL, where the discrim-

ination term is formulated as the weighted summation of the squared distances between all pairs of coding vectors. This weighted squared distance principle has been widely adopted in unsupervised manifold regularization, where the coding vectors can preserve the geometric structure of original data samples to benefit clustering and classification. Recent advances in sparse coding, such as Graph-SC [27], LLC [26] and LSC/HLSC [28, 29], utilized the similarity between pairs of samples to assign the weight and achieved significant improvements in Bag-of-Words based image classification. Unlike these methods, we incorporate the sample label information into the design of weight. With the proposed scheme, one can either follow some paradigm to set weight to each pair of coding vectors, or design a specific parametric form for weight assignment. By this way, the design of discrimination term can be regarded as the design of a paradigm of weight assignment, which provides a new insight in developing new DDL models. Actually, the discrimination term on coding vectors in the FDDL method can be explained as a special case of our model, where the weights are deterministic and are simply determined by the numbers of samples of each class.

To make weight assignment more adaptive and flexible, we then propose a parameterization method, which consequently leads to the proposed support vector guided dictionary learning (SVGDL) model. One promising property of SVGDL is that, by incorporating the weight parameterization with the symmetry, consistency and balance constraints, the optimization problem on weight assignment is equivalent to the dual form of linear support vector machines (SVM) [30]. This property allows us to use the multi-class linear SVM [5] for efficient DDL. Another important insight from SVGDL is that, most weights will be zero and only the weights of pairs of support vectors are nonzero. Such a fact indicates that the weights are sparse and only the coding vectors near the decision boundaries play a crucial role in learning a discriminative dictionary. Compared with FDDL, SVGDL adaptively assigns weights to pairs of coding vectors in the support vector set, and is superior in terms of classification performance.

Another interesting point of SVGDL is its robustness to the regularizer of coding vectors. Almost all DDL methods impose the sparse ℓ_0-norm or ℓ_1-norm regularizers on coding vectors. However, some recent works [31, 32] argue that whether sparsity would always be helpful for classification. Mehta and Gray [33] analyzed the working mechanism and generalization error bound of predictive sparse coding, but several open problems remain on the necessity of sparsity in DDL. Furthermore, the complexity of ℓ_1-norm sparse coding generally is much higher than that of ℓ_2-norm coding, and the inefficiency would be exacerbated for DDL with ℓ_1-norm regularizer when the number of atoms or training samples is high. For SVGDL, fortunately, our experimental results show that the classification performance is insensitive to the choice of ℓ_2-norm or ℓ_1-norm regularizer. This can be owed to the fact only a few support coding vectors (with non-zero weights) will be automatically selected to guide the learning of dictionary, i.e., the sparsity lies in the weights but not the coding vectors. Consequently, the time complexity of SVGDL can be greatly reduced, especially in the testing stage where the coding step can be replaced by matrix-vector multiplication.

2 Problem Formulation

Assume that $x \in \mathbb{R}^m$ is a m dimensional signal with class label $y \in \{1, 2, \ldots, C\}$. The training set with n samples is denoted as $X = [X_1, X_2, \ldots, X_C] = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^{m \times n}$, where X_c is the subset of n_c training samples from class c. Denote the learned dictionary by $D = [d_1, d_2, \ldots, d_K] \in \mathbb{R}^{m \times K}$ ($K > m$ and $K \ll n$), where d_is are the atoms. $Z = [z_1, z_2, \ldots, z_n]$ are the coding vectors of X over D. A general DDL model can be described as:

$$< D, Z >= \arg\min_{D,Z} \mathcal{R}(X, D, Z) + \lambda_1 \|Z\|_p^p + \lambda_2 \mathcal{L}(Z), \tag{1}$$

where λ_1 and λ_2 are the trade-off parameters, $\mathcal{R}(X, D, Z)$ is the reconstruction term, p denotes the parameter of the ℓ_p-norm regularizer (e.g., ℓ_1-norm or ℓ_2-norm), and $\mathcal{L}(Z)$ denotes the discrimination term for Z.

Note that apart from the discrimination term, discrimination can also be enforced by imposing structural constraints on the learned dictionary. For example, FDDL [18] learns the structured dictionary $D = [D_1, D_2, \ldots, D_C]$, where D_c is the sub-dictionary corresponding to class c. Then $\mathcal{R}(X, D, Z)$ can be divided into the sum of the reconstruction errors under the sub-dictionaries. Although this class-customized setting for dictionary learning is effective when there are sufficient training samples for each class, it is not scalable to the problem with a great number of classes. Thus, in our formulation we only consider the discrimination term and learn a single dictionary shared between all classes.

Intuitively, the discrimination can be assessed by the similarity of pairs of coding vectors from the same class and the dissimilarity of pairs of coding vectors from the different classes. Thus, it is reasonable to use the weighted sum of the squared distances of pairs of coding vectors as an indicator of discrimination capability, resulting in the discrimination term:

$$\mathcal{L}(Z, w_{ij}) = \sum_{i,j} \|z_i - z_j\|_2^2 w_{ij}. \tag{2}$$

Next we will show that the Fisher discrimination criterion adopted in FDDL can be reformulated as a special case of the discrimination term in Eq. (2).

In FDDL, the discrimination term is defined as $\mathcal{L}(Z) = tr(S_W(Z)) - tr(S_B(Z))$, where $tr(S_W(Z))$ and $tr(S_B(Z))$ denote the within-class and between-class scatters, respectively. Based on the definitions of S_W and S_B, $\mathcal{L}(Z)$ in FDDL can be reformulated as the weighted sum of the squared distances of pairs of coding vectors. We have the following **Lemma 1**.

Lemma 1. *Denote by $\bar{z}_c$ and $\bar{z}$ the mean vectors of Z_c and Z, respectively, where Z_c is the set of coding vectors of samples from class c. Then $\mathcal{L}(Z)$ in FDDL is equivalent to the weighted sum of the squared distances of pairs of coding vectors:*

$$\mathcal{L}(Z) = \sum_{c=1}^{C} \Big(\sum_{y_i=c, y_j=c} \big(\frac{1}{n_c} - \frac{1}{2n}\big)\|z_i - z_j\|_2^2 + \sum_{y_i=c, y_j \neq c} \big(-\frac{1}{2n}\big)\|z_i - z_j\|_2^2 \Big). \tag{3}$$

Please refer to **Appendix A** for the proof of **Lemma 1**.

From Eq. (3), we can see that if two samples are from the same class, the weight $\frac{1}{n_c}-\frac{1}{2n}$ is positive, and the Fisher discrimination term would encourage to learn a dictionary that minimizes the difference between coding vectors from the same class. Meanwhile, if two samples are from different classes, the weight $-\frac{1}{2n}$ is negative, and the Fisher discrimination term would encourage to learn a dictionary that maximizes the difference between coding vectors from different classes.

Using the discrimination term in Eq. (2), we define a general model for DDL:

$$< D, Z >= \arg\min_{D,Z} \ \|X - DZ\|_F^2 + \lambda_1 \|Z\|_p^p + \lambda_2 \sum_{i,j} \|z_i - z_j\|_2^2 w_{ij}, \qquad (4)$$

where $w_{ij} \geq 0$ when x_i and x_j are of the same class, and $w_{ij} < 0$ when x_i and x_j are of different classes. One choice of the discrimination term is the Fisher discrimination criterion. However, as we show above the weight assignment adopted in the Fisher discrimination term is deterministic. The weight of pairwise coding vectors from different classes is fully determined by the number of samples n, and the weight of pairwise coding vectors from the same class is fully determined by n and the number of samples of this class n_c. Note that some pairs of coding vectors may play more important roles than other pairs in learning a discriminative dictionary. The deterministic weight assignment in Fisher discrimination term ignores this fact and thus may result in less effective classification. In the next section we propose a parameterization method for adaptive weight assignment.

3 Support Vector Guided Dictionary Learning

3.1 A Parameterized Perspective on Discrimination

Rather than directly assigning weight w_{ij} for each pair, we assume that all the weights w_{ij} can be parameterized as a function with variable β, and define the parameterized formulation of the discrimination term $\mathcal{L}(Z)$ as follows:

$$\mathcal{L}(Z, w_{ij}(\beta)) = \sum_{i,j} \|z_i - z_j\|_2^2 w_{ij}(\beta). \qquad (5)$$

In order to choose a proper manner for the parameterization of w_{ij}, we claim that the following three properties should be satisfied:

a) Symmetry: $w_{ij}(\beta) = w_{ji}(\beta)$;

b) Consistency: $w_{ij}(\beta) \geq 0$ if $y_i = y_j$, and $w_{ij}(\beta) \leq 0$ if $y_i \neq y_j$;

c) Balance: $\sum_{j=1}^{n} w_{ij}(\beta) = 0, \forall i$.

The above three properties give a specific explanation of the model in Eq. 4. The symmetry can be achieved naturally; the consistency means that the weight w_{ij} should be non-negative when z_i and z_j are from the same class while the weight w_{ij} should be non-positive when z_i and z_j are from different classes; since the number of pairs with different class labels is much larger than that with the same

class label, the balance constraint is introduced to balance the contributions of positive and negative weights.

We then give a special instance of the constructed parameterization for $w_{ij}(\beta)$. For the convenience, we consider the two-class classification problem with label $y_i \in \{-1, 1\}$. Then we can define $w_{ij}(\beta) = y_i y_j \beta_i \beta_j$ and $\sum_{j=1}^{n} y_j \beta_j = 0$, where the variable $\beta = [\beta_1, \beta_2, \ldots, \beta_n]$ is a nonnegative vector. It is obvious to see that $w_{ij}(\beta)$ satisfies all the three properties above. Based on this setting for $w_{ij}(\beta)$, we can then transform $\mathcal{L}(Z, w_{ij}(\beta))$ into the new form as described in the following **Lemma 2**.

Lemma 2. *Denote $w_{ij}(\beta) = y_i y_j \beta_i \beta_j$. If $\sum_{j=1}^{n} y_j \beta_j = 0$, then the discrimination term $\mathcal{L}(Z)$ can be written as:*

$$\mathcal{L}(Z, w_{ij}(\beta)) = -2 \sum_{i,j} y_i y_j \beta_i \beta_j z_i^T z_j = \beta^T K \beta, \tag{6}$$

where K is a negative semidefinite matrix.

Please refer to **Appendix B** for the proof of **Lemma 2**.

Since K is a negative semidefinite matrix, to obtain a extremum of β, we could maximize the objective function of $\mathcal{L}(Z, w_{ij}(\beta))$:

$$\begin{aligned} &< \beta >= \arg\max_{\beta} \ \beta^T K \beta + r(\beta) \\ &\text{s.t.} \ \ \beta_i \geq 0, \forall i, \textstyle\sum_{j=1}^{n} y_j \beta_j = 0, \end{aligned} \tag{7}$$

where $r(\beta)$ is some regularization term to avoid the trivial solution with $\beta = 0$. Overall, we have the following parameterized formulation of DDL:

$$<D, Z>= \arg\min_{D,Z} (\|X - DZ\|_F^2 + \lambda_1 \|Z\|_p^p + \lambda_2 \max_{\beta \in dom(\beta)} (\sum_{i,j} \|z_i - z_j\|_2^2 w_{ij}(\beta) + r(\beta))), \tag{8}$$

where the domain $dom(\beta)$ of variable β is $dom(\beta) : \beta \succeq 0, \sum_{j=1}^{n} y_j \beta_j = 0$ according to the previous definition. We can see that the general weight assignment in coding space falls into the appropriate selection of $dom(\beta)$, $w_{ij}(\beta)$ and $r(\beta)$. In particular, the model in Eq. (4) is a special case of Eq. (8) when β is given by a fixed matrix $[w_{ij}]$.

3.2 Dictionary Learning Model

By choosing $r(\beta) = 4 \sum_{i=1}^{n} \beta_i$ and adopting the $w_{ij}(\beta)$ and $dom(\beta)$ described above, the model in Eq. (8) can be rewritten as:

$$\begin{aligned} &< D, Z >= \arg \min_{D,Z} \ (\|X - DZ\|_F^2 + \lambda_1 \|Z\|_p^p + \lambda_2 \max_{\beta} (4 \textstyle\sum_{i=1}^{n} \beta_i \\ &\qquad\qquad - 2 \textstyle\sum_{i,j} y_i y_j \beta_i \beta_j z_i^T z_j)) \\ &\text{s.t.} \ \ \beta_i \geq 0, \forall i \text{ and } \textstyle\sum_{j=1}^{n} y_j \beta_j = 0. \end{aligned} \tag{9}$$

Note that the subproblem for β is exactly the Lagrange dual of hard-margin binary SVM, which can be solved using some classical algorithms like sequential

minimal optimization (SMO) [34]. To further reduce the adverse effect of outliers, we impose β with the additional constraint $\beta_i \leq \frac{1}{2}\theta$ for all i, where θ is a fixed constant. Thus the subproblem for β reduces to the dual formulation of soft-margin binary SVM. Then we replace the subproblem of β with its primal SVM form, leading to the support vector guided dictionary learning (SVGDL) model:

$$< D, Z, u, b >= \arg \min_{D,Z,u,b} \ \|X - DZ\|_F^2 + \lambda_1 \|Z\|_p^p + 2\lambda_2 \mathcal{L}(Z, y, u, b), \quad (10)$$

where u is the normal to the hyperplane of SVM, b is the corresponding bias, $y = [y_1, y_2, \ldots, y_n]$ is the label vector, and $\mathcal{L}(Z, y, u, b)$ is defined as:

$$\mathcal{L}(Z, y, u, b) = \|u\|_2^2 + \theta \sum\nolimits_{i=1}^{n} \ell(z_i, y_i, u, b), \quad (11)$$

where $\ell(z_i, y_i, u, b)$ is the hinge loss function.

The solution $< u, b >$ can be represented as the linear combination of a few coding vectors (support vectors), i.e., we have $\beta_i \neq 0$ only if z_i is the support vector. The sparsity of β further leads to the sparsity of weight matrix $[w_{ij}]$ based on our parameterization method. Thus, the model in Eq. (4) can be written as:

$$< D, Z >= \arg \min_{D,Z} \ \|X - DZ\|_F^2 + \lambda_1 \|Z\|_p^p + \lambda_2 \sum_{i,j \in SV} \|z_i - z_j\|_2^2 w_{ij}(\beta), \quad (12)$$

where SV is the set of support vectors. From the model in Eq. (10), there are two distinct characteristics of SVGDL. First, unlike FDDL which adopts a deterministic method for weight assignment, SVGDL adopts an adaptive weight assignment. Second, rather than assigning non-zero weights for all pairwise coding vectors, SVGDL only assigns non-zero weights for pairwise support coding vectors, which indicates that only the coding vectors near the classification hyperplane play a dominant role in learning the discriminative dictionary. These two characteristics are consistent with our intuitive understandings: the coding vectors near the boundary are more crucial for DDL.

Another noticeable advantage of the proposed model is that the classification performance of SVGDL is insensitive to the choice of ℓ_1-norm or ℓ_2-norm regularizers on the coding vectors. Note that most existing dictionary learning methods take the sparsity as a primary requirement for learning a discriminative dictionary. However, our experimental results indicate that sparsity has little impact on the discriminative capability of the learned dictionary by SVGDL, while it will greatly increase the computational burden in both the training and testing stages. Figure 1 shows the classification accuracy of SVGDL with the ℓ_1-norm regularizer and the ℓ_2-norm regularizer on the Caltech-101 database using different numbers of training samples per class. One can see that, SVGDL with the ℓ_2-norm regularizer always achieves higher accuracy than SVGDL with the ℓ_1-norm regularizer. We argue that, other than the sparsity of coding vectors, the sparsity of the weight matrix $[w_{ij}]$ seems to play a more crucial role in learning a discriminative dictionary. To verify this, we evaluate the model in Eq. (10) by utilizing the quadratic hinge loss (will be discussed later) and squared loss, which

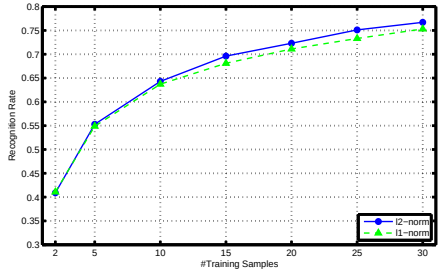

Fig. 1. Accuracy curves on the Caltech-101 database using ℓ_1-norm and ℓ_2-norm for regularization in SVGDL.

induce the sparse and non-sparse weight matrix $[w_{ij}]$ respectively, and compare the recognition results on several face databases (the detailed settings are presented in Section 4.3). As shown in Table 1, the results using quadratic hinge loss are much better than that using squared loss, which further emphasizes the importance of sparse weight matrix. Thus, we choose ℓ_2-norm regularizer on Z for SVGDL in the later discussion due to its computational efficiency.

Table 1. The recognition results via quadratic hinge loss and squared loss on different face databases.

	Extended Yale B	AR	Multi-PIE Test 1
Quadratic hinge loss	0.961	0.946	0.955
Squared loss	0.933	0.921	0.937

For multi-class classification, we simply adopt the one-vs-all strategy by leaning C hyperplanes $U = [u_1, u_2, \ldots, u_C]$ and corresponding biases $b = [b_1, b_2, \ldots, b_C]$, and formulate SVGDL as:

$$< D, Z, U, b >= \arg\min_{D,Z,U,b} \|X - DZ\|_F^2 + \lambda_1 \|Z\|_p^p + 2\lambda_2 \sum_{c=1}^{C} \mathcal{L}(Z, y^c, u_c, b_c), \quad (13)$$

where $y^c = [y_1^c, y_2^c, \ldots, y_n^c]$, $y_i^c = 1$ if $y_i = c$, and otherwise $y_i^c = -1$.

3.3 Optimization and Complexity

The SVGDL model in Eq. (13) is a not a jointly convex optimization problem for $< D, Z, U, b >$, but is convex with respect to each variable. Thus, we adopt an alternative minimization scheme for updating D, Z and $< U, b >$, respectively. The detailed procedure can be partitioned into three steps alternatingly.

When D and Z are fixed, the minimization of $< U, b >$ can be formulated as a multi-class linear SVM problem, which can be further divided into C linear

Algorithm 1: Algorithm of Support Vector Guided Dictionary Learning (SVGDL)

Input: $D_{init}, Z_{init}, U_{init}, b_{init}, X \in \mathbb{R}^{m\times n}, \lambda_1, \lambda_2, \theta$.

Output: D, U, b.

1:**do until the terminal condition**

2: **for** $c = 1$ to C **do**

3: $u_c, b_c \leftarrow$ by one-vs-all linear SVM

4: **end for**

5: **for** $i = 1$ to n **do**

6: $z_i \leftarrow \arg\min_z \ \|x_i - Dz\|_2^2 + \lambda_1\|z\|_2^2 + 2\lambda_2 \cdot \theta \cdot \sum_{c=1}^{C} \ell(z_i, y_i, u_c, b_c)$

7: **end for**

8: $D \leftarrow \arg\min_D \ \|X - DZ\|_F^2 \quad s.t.\ \|d_i\|^2 \le 1, \forall i.$

9:**end do**

one-against-all SVM subproblems. We adopt the multi-class linear SVM solver [5] proposed by Yang to learn all u_cs and b_cs one by one based on the gradient-based optimization method. The quadratic hinge loss function $\ell(z_i, y_i^c, u_c, b_c) = [\max(0, y_i^c[u_c; b_c]^T[z_i; 1]-1)]^2$ in [5] is used in our implementation to approximate the hinge loss due to its computational simplicity and the better smooth property than hinge loss function.

When D, U and b are fixed, the coefficient matrix Z can be optimized by columns. The optimization problem related to each z_i is formulated as follows:

$$< z_i >= \arg\min_{z_i} \ \|x_i - Dz_i\|_2^2 + \lambda_1\|z_i\|_2^2 + 2\lambda_2 \cdot \theta \cdot \sum\nolimits_{c=1}^{C} \ell(z_i, y_i^c, u_c, b_c). \quad (14)$$

In each iteration, for each c, if $y_i^c(u_c^T z_i + b_c)-1>0$ in the previous iteration, we use $\|y_i^c(u_c^T z_i + b_c)-1\|^2$ to replace the squared hinge loss, and use 0 else. We repeat this until convergence. Thus the optimization of each z_i has a closed-form solution.

When Z, U and b are fixed, the optimization problem with respect to D can be written as:

$$< D >= \arg\min_D \ \|X - DZ\|_F^2 \quad \text{s.t.} \ \ \|d_k\|^2 \le 1, \ \forall k \in \{1, 2, \ldots, K\}, \quad (15)$$

where the additional constraints are introduced to avoid the scaling issue of the atoms. The subproblem in Eq.(15) can be solved effectively by the Lagrange dual method [35].

We use PCA to initialize the dictionary of each class, and concatenate these sub-dictionaries as the initialized D. The initialized Z, U and b are set as zero matrices and zero vector, respectively. The stopping criterion is the relative difference between D in 2 successive iterations with a maximum iteration number. The overall optimization procedure of SVGDL is summarized in Algorithm 1.

In the training stage, the computational cost of the SVGDL algorithm comes from three parts: $O(Cmn)$ for linear SVM, $O(K^3mn)$ for updating the coding vectors and $O(K^3mn)$ for updating the learned dictionary. Since the optimization model is non-convex, the algorithm can not converges to the global minimum. Empirically, satisfactory solutions to the desired dictionary D and the

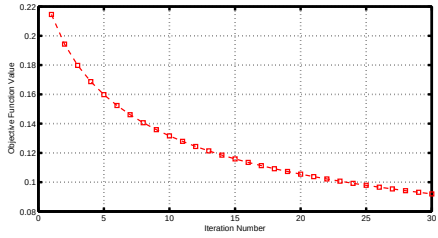

Fig. 2. The convergence of SVGDL on the AR database.

SVM classifier $< U, b >$ can be obtained with the decreasing of the objective function. Figure 2 shows an example to illustrate the convergence of SVGDL.

3.4 Classification Approach

Once the dictionary D and the classifier $< U, b >$ is learned, we can perform the classification task as follows. For a test sample x, we first perform the coding step by projecting x with a fixed matrix P: $z = Px$, where $P = (D^T D + \lambda_1 I)^{-1} D^T$. Then we simply apply the C linear classifier $< u_c, b_c >$, where $c \in 1, 2, \ldots, C$, on the coding vector z to predict the label of the sample x by:

$$y = \arg \max_{c \in 1,2,\ldots,C} u_c^T z + b_c. \tag{16}$$

In the test stage, the computational complexity of SVGDL is $O(Km)$.

4 Experiments

In this section, SVGDL is evaluated on three classification tasks, i.e., face recognition, object recognition, and sport action recognition. For face recognition, we use three face datasets: Extended Yale B [36], AR [37], and Multi-PIE [38]. For object recognition, we adopt the Caltech-101 dataset [39]. For action recognition, we use the UCF sport action dataset. SVGDL is compared with both the standard sparse representation based classification (SRC) method [13] and the state-of-the-art dictionary learning methods, including DKSVD [16], LC-KSVD [19, 20], dictionary learning with structure incoherence (DLSI) [22] and FDDL [18]. For each dataset, we report the recognition accuracy, training and test time of the competing methods. (NN means training stage is not needed and "-" means the time is not available.)

4.1 Parameter Settings

We choose the parameter $\theta = 0.2$ and it works well in all of our experiments. Besides, there are two main parameters (λ_1, λ_2) to be tuned in the proposed

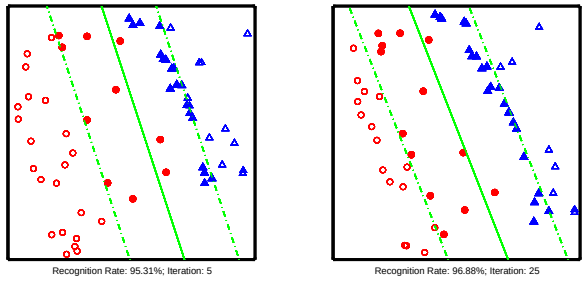

Fig. 3. The change of coding vectors and the classifying hyperplane in iterations.

SVGDL method. The parameter λ_1 and λ_2 are evaluated by 5-fold cross validation. For the face recognition tasks, we set $\lambda_1 = 0.002, \lambda_2 = 0.001$ for Extended Yale B [36], $\lambda_1 = 0.002, \lambda_2 = 0.001$ for AR [37], and $\lambda_1 = 0.002, \lambda_2 = 0.001$ for Multi-PIE [38]. We also evaluate our method on Caltech-101 dataset [39] for object recognition task. We use the 3,000 dimensional features described in [20] for fair comparison. $\lambda_1 = 0.05, \lambda_2 = 0.002$ are selected in this setup. We finally apply SVGDL on the UCF sport action dataset [40], each sample has a dimension of 29930 and the parameters are chosen as $\lambda_1 = 0.02, \lambda_2 = 0.002$.

4.2 Visual Illustration of SVGDL

Using two individuals from the Extended Yale B database, we provide a visual illustration of the influence of SVGDL training on the coding vectors and classification hyperplane. For each individual, we select 32 images for training and use the remaining 32 images for test. Figure 3 the distributions of the coding vectors obtained using SVGDL after 5 and 25 iterations. The hyperplanes and margins are also provided to illustrate the discriminative capability of coding vectors. The solid circles and triangles are the support vectors that need to be assigned the weight to update dictionary. The green solid line and dotted line depict the separating hyperplane and margin. From Figure 3, one can see that, the number of misclassified samples after 25 iterations is 2, which is less than that after 5 iterations. The margin after 25 iterations is also larger than that after 5 iterations. The recognition accuracy on the test set after 25 iterations is 96.88%, which is also higher than that after 5 iterations. All these cues indicate that SVGDL training is effective in learning a discriminative dictionary, resulting in coding vectors with better discriminative capability.

4.3 Face Recognition

We evaluate the performance of the proposed algorithm on several face recognition benchmark databases like the Extended Yale B, AR, and Multi-PIE. We

Table 2. The recognition results and running time on the Extended Yale B database.

Methods	SRC	SVM	DKSVD	LC-KSVD	DLSI	FDDL	**SVGDL**
Accuracy	0.900	0.888	0.753	0.906	0.890	0.919	**0.961**
Train(s)	NN	0.51	-	75.3	4.5e2	4.4e3	2.2e2
Test(s)	3.1e-2	3.5e-5	-	4.0e-4	4.3e-2	1.4	7.9e-6

compare the proposed SVGDL with two typical classification methods, including linear support vector machines (SVM) and SRC [13], five dictionary learning based methods, including DKSVD [16], LC-KSVD [19, 20], DLSI [22] and FDDL [18]. In all FR experiments, each face image has a reduced dimension of 300.

a) Extended Yale B: The Extended Yale B database consists of 2,414 frontal face images of 38 individuals. Each individual has 64 images and we randomly pick 20 images as training set and use the rest as testing set. The images were cropped to 54×48. The number of dictionary atoms K is fixed as 380 here. Table 2 summarizes the recognition accuracies. We can observe that SVGDL gives a significant accuracy improvement compared to other methods and it has the least testing time.

b) AR: The AR database consists of over 4,000 images of 126 individuals. For each individual, 26 face images are collected from two separated sessions. Following [18], we select 50 male individuals and 50 female individuals for the 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 face image is of size 60×43 and the learned dictionary has 500 atoms. The results are presented in Table 3. Although the experimental setting is challenging, SVGDL still has at least 2% improvement over other methods, and it has much less time consumption compared to FDDL.

c) Multi-PIE: The CMU Multi-PIE face database consists of 337 individuals including four sessions with the variations of pose, expression and illumination. We follow the same experimental setting adopt in [18]. We choose the first 60 individuals from Session 1 for training. For each training person, we use the frontal images of 14 illuminations ({0,1,3,4, 6,7,8,11,13,14,16,17,18,19}) with neutral expression (for Test 1) or smile expression (for Test 2) for training, and use the frontal images of 10 illuminations ({0,2,4,6,8,10,12,14,16,18}) from Session 3 with neutral expression (for Test 1) or smile expression (for Test 2) for testing. The images are normalized to 100×82 and $K = 840$. The recognition results and the elapsed time of Test 1 are presented in Table 4. SVGDL performs the second best in the experiment, only lags behind by FDDL. Note that FDDL trains sub-dictionaries for all individuals, while a single dictionary is enough to give good performance by SVGDL.

4.4 Objection Classification

We also evaluate SVGDL on the Caltech-101 dataset for object classification. This dataset contains 101 object categories and 29,780 images; each category

Table 3. The recognition results and running time on the AR database.

Methods	SRC	SVM	DKSVD	LC-KSVD	DLSI	FDDL	**SVGDL**
Accuracy	0.888	0.871	0.854	0.897	0.898	0.920	**0.946**
Train(s)	NN	1.24	-	53.7	4.9e2	2.1e4	7.6e2
Test(s)	3.4e-2	6.1e-5	-	4.2e-4	0.16	2.5	2.0e-5

Table 4. The recognition results and running time on the Multi-PIE database.

Methods	SRC	SVM	DKSVD	LC-KSVD	DLSI	FDDL	**SVGDL**
Test 1	0.955	0.916	0.939	93.7	0.941	**0.967**	0.955
Test 2	0.961	0.922	0.898	90.8	0.959	**0.980**	0.963
Train(s)	NN	1.74	-	64.8	6.3e2	5.1e4	2.2e3
Test(s)	3.0e-2	5.2e-5	-	3.7e-4	6.9e-2	3.1	2.6e-5

has at least 80 images. Following [20], we randomly select 5, 10, 15, 20, 25 and 30 images per object, respectively, for training and test on the rest. We also give the running time in the case of 30 images. Figure 4 shows some samples from five of all the classes. We find $K = 510$ is sufficient in this experiment.

Table 5 compares the classification accuracies of SVGDL with SRC, K-SVD, DKSVD, LC-KSVD and FDDL under the same experimental setting. As it can be observed, SVGDL outperforms the other methods in all cases. SVGDL, FDDL, LC-KSVD and DKSVD all give better results than SRC, which indicates that the better performance can be achieved by learning a discriminative dictionary. When 30 images involved in training, the improvements over LC-KSVD and FDDL by SVGDL are 2.7% and 3.6%. The shorter training and testing time also shows the superiority of SVGDL.

Fig. 4. Some sample objects from the Caltech-101 database

4.5 Action Recognition

Finally, we illustrate SVGDL on the UCF sport action dataset [40] for action recognition. There are 140 video clips in the UCF sport action dataset that are collected from various broadcast sports channels (e.g., BBC and ESPN). This dataset contains 10 sport action classes: driving, golfing, kicking, lifting, horse riding, running, skate-boarding, swinging (prommel horse and floor), swinging (high bar) and walking. We follow the common experimental settings in [20]. The number of atoms is set to $K = 50$.

Table 5. The recognition results (%) and running time on the Caltech-101 dataset

training number	5	10	15	20	25	30	Train(s)	Test(s)
SRC	48.8	60.1	64.9	67.7	69.2	70.7	NN	1.09
K-SVD	49.8	59.8	65.2	68.7	71.0	73.2	-	-
DKSVD	49.6	59.5	65.1	68.6	71.1	73.0	-	-
LC-KSVD	54.0	63.1	67.7	70.5	72.3	73.6	1.3e4	3.7e-3
FDDL	53.6	63.6	66.8	69.8	71.7	73.1	1.1e5	12.9
SVGDL	**55.3**	**64.3**	**69.6**	**72.3**	**75.1**	**76.7**	1.5e3	1.2e-5

The results of SVGDL are evaluated via five-fold cross validation, where one fold is used for testing and the remaining four folds for training. We compare SVGDL with Qiu *et. al.* [41], Yao *et. al.* [42], Sadanand *et. al.* [43], SRC, K-SVD, DKSVD, LC-KSVD and FDDL. The recognition accuracies, training and testing time are shown in Table 6. SVGDL outperforms the state-of-the-art methods. It is 200 times faster than FDDL, which has the second best accuracy in test.

Table 6. The accuracies (%) and running time on the UCF sports action dataset.

Methods	Qiu	Yao	Sadanand	SRC	K-SVD	DKSVD	LC-KSVD	FDDL	**SVGDL**
Accuracy	83.6	86.6	90.7	92.9	86.8	88.1	91.2	94.3	**94.4**
Train(s)	-	-	-	NN	-	-	2.0	8.02	15.6
Test(s)	-	-	-	1.8e-3	-	-	8.6e-4	3.4e-2	1.6e-4

5 Conclusions

This paper provided a new insight on DDL by formulating the discrimination term as the weighted summation of the squared distances between pairwise coding vectors. The proposed discrimination term not only can explain some existing discrimination term, e.g., Fisher discrimination, but also is valuable in developing novel DDL methods by designing appropriate weight assignment scheme. To overcome the limitation of Fisher discrimination, we adopt a parameterization method for adaptive weight assignment, leading to the proposed support vector guided dictionary learning (SVGDL) method. SVGDL can adaptively assign non-zero weights to a few pairwise coding vectors which play a critical role in learning a discriminative dictionary. Furthermore, in contrast to the standard ℓ_1 sparsity based dictionary learning methods, SVGDL is more efficient by using the ℓ_2-norm regularizer on coding vectors. Experimental results on several image benchmark image classification datasets showed that SVGDL outperforms many state-of-the-art DDL methods in terms of higher accuracy and faster test time.

Acknowledgements. This work is supported by the Hong Kong RGC GRF grant (PolyU 5313/13E), NSFC grant (61271093, 51275348), and the program of MoE (NCET-12-0150).

References

1. Baraniuk, R.: Compressive sensing. IEEE Signal Processing Magazine (2007)
2. Mairal, J., Elad, M., Sapiro, G.: Sparse representation for color image restoration. IEEE Transactions on Image Processing (2008)
3. Yang, J., Wright, J., Huang, T.S., Ma, Y.: Image super-resolution via sparse representation. IEEE Transactions on Image Processing (2010)
4. Elhamifar, E., Vidal, R.: Sparse subspace clustering: Algorithm, theory, and applications. IEEE Transactions on Pattern Analysis and Machine Intelligence (2013)
5. Yang, J., Yu, K., Gong, Y., Huang, T.: Linear spatial pyramid matching using sparse coding for image classification. In: CVPR. (2009)
6. Shabou, A., LeBorgne, H.: Locality-constrained and spatially regularized coding for scene categorization. In: CVPR. (2012)
7. Olshausen, B.A., Field, D.J.: Sparse coding with an overcomplete basis set: A strategy employed by v1? Vision research (1997)
8. Candes, E.J.: The restricted isometry property and its implications for compressed sensing. Comptes Rendus Mathematique (2008)
9. Mallat, S.: A wavelet tour of signal processing. (1999)
10. Aharon, M., Elad, M., Bruckstein, A.: K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. IEEE Transactions on Signal Processing (2006)
11. Engan, K., Aase, S.O., Husoy, J.: Frame based signal compression using method of optimal directions (mod). In: ISCAS. (1999)
12. Rubinstein, R., Peleg, T., Elad, M.: Analysis k-svd: a dictionary-learning algorithm for the analysis sparse model. IEEE Transactions on Signal Processing (2013)
13. Wright, J., Yang, A.Y., Ganesh, A., Sastry, S.S., Ma, Y.: Robust face recognition via sparse representation. IEEE Transactions on Pattern Analysis and Machine Intelligence (2009)
14. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Lost in quantization: Improving particular object retrieval in large scale image databases. In: CVPR. (2008)
15. Sivic, J., Zisserman, A.: Video google: A text retrieval approach to object matching in videos. In: ICCV. (2003)
16. Zhang, Q., Li, B.: Discriminative k-svd for dictionary learning in face recognition. In: CVPR. (2010)
17. Mairal, J., Bach, F., Ponce, J., Sapiro, G., Zisserman, A., et al: Supervised dictionary learning. In: NIPS. (2008)
18. Yang, M., Zhang, D., Feng, X.: Fisher discrimination dictionary learning for sparse representation. In: ICCV. (2011)
19. Jiang, Z., Lin, Z., Davis, L.S.: Learning a discriminative dictionary for sparse coding via label consistent k-svd. In: CVPR. (2011)
20. Jiang, Z., Lin, Z., Davis, L.: Label consistent k-svd: learning a discriminative dictionary for recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence (2013)
21. Mairal, J., Bach, F., Ponce, J.: Task-driven dictionary learning. IEEE Transactions on Pattern Analysis and Machine Intelligence (2012)
22. Ramirez, I., Sprechmann, P., Sapiro, G.: Classification and clustering via dictionary learning with structured incoherence and shared features. In: CVPR. (2010)
23. Gao, S., Tsang, I., Ma, Y.: Learning category-specific dictionary and shared dictionary for fine-grained image categorization. IEEE Transactions on Image Processing (2013)

24. Zhang, W., Surve, A., Fern, X., Dietterich, T.: Learning non-redundant codebooks for classifying complex objects. In: ICML. (2009)
25. Wang, Z., Yang, J., Nasrabadi, N., Huang, T.: Look into sparse representation-based classification: a margin-based perspective. In: ICCV. (2013)
26. Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., Gong, Y.: Locality-constrained linear coding for image classification. In: CVPR. (2010)
27. Zheng, M., Bu, J., Chen, C., Wang, C., Zhang, L., Qiu, G., Cai, D.: Graph regularized sparse coding for image representation. IEEE Transactions on Image Processing (2011)
28. Gao, S., Tsang, I.W., Chia, L.T., Zhao, P.: Local features are not lonely–laplacian sparse coding for image classification. In: CVPR. (2010)
29. Gao, S., Tsang, I.H., Chia, L.T.: Laplacian sparse coding, hypergraph laplacian sparse coding, and applications. IEEE Transactions on Pattern Analysis and Machine Intelligence (2013)
30. Burges, C.J.: A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery (1998)
31. Rigamonti, R., Brown, M.A., Lepetit, V.: Are sparse representations really relevant for image classification? In: CVPR. (2011)
32. Zhang, D., Yang, M., Feng, X.: Sparse representation or collaborative representation: Which helps face recognition? In: ICCV. (2011)
33. Mehta, N., Gray, A.G.: Sparsity-based generalization bounds for predictive sparse coding. In: ICML. (2013)
34. Platt, J., et al.: Sequential minimal optimization: A fast algorithm for training support vector machines. (1998)
35. Lee, H., Battle, A., Raina, R., Ng, A.Y.: Efficient sparse coding algorithms. In: NIPS. (2007)
36. Lee, K.C., Ho, J., Kriegman, D.: Acquiring linear subspaces for face recognition under variable lighting. IEEE Transactions on Pattern Analysis and Machine Intelligence (2005)
37. Martınez, A., Benavente, R.: The ar face database. CVC Technical Report (1998)
38. Gross, R., Matthews, I., Cohn, J., Kanade, T., Baker, S.: Multi-pie. Image and Vision Computing (2010)
39. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. Computer Vision and Image Understanding (2007)
40. Rodriguez, M., Ahmed, J., Shah, M.: Action mach a spatio-temporal maximum average correlation height filter for action recognition. In: Indian Conference on Computer Vision, Graphics and Image Processing. (2008)
41. Qiu, Q., Jiang, Z., Chellappa, R.: Sparse dictionary-based representation and recognition of action attributes. In: ICCV. (2011)
42. Yao, A., Gall, J., Van Gool, L.: A hough transform-based voting framework for action recognition. In: CVPR. (2010)
43. Sadanand, S., Corso, J.J.: Action bank: A high-level representation of activity in video. In: CVPR. (2012)