A Max-Margin Perspective on Sparse Representation-based Classification

Zhaowen Wang[†], Jianchao Yang[‡], Nasser Nasrabadi[§], Thomas Huang[†]
[†]Beckman Institute, University of Illinois at Urbana-Champaign, Urbana, IL 61801
[‡]Adobe Systems Inc., San Jose, CA 95110 [§]U.S. Army Research Laboratory, Adelphi, MD 20783
{wang308, huang}@ifp.uiuc.edu, jiayang@adobe.com, nasser.m.nasrabadi.civ@mail.mil

Abstract

Sparse Representation-based Classification (SRC) is a powerful tool in distinguishing signal categories which lie on different subspaces. Despite its wide application to visual recognition tasks, current understanding of SRC is solely based on a reconstructive perspective, which neither offers any guarantee on its classification performance nor provides any insight on how to design a discriminative dictionary for SRC. In this paper, we present a novel perspective towards SRC and interpret it as a margin classifier. The decision boundary and margin of SRC are analyzed in local regions where the support of sparse code is stable. Based on the derived margin, we propose a hinge loss function as the gauge for the classification performance of SRC. A stochastic gradient descent algorithm is implemented to maximize the margin of SRC and obtain more discriminative dictionaries. Experiments validate the effectiveness of the proposed approach in predicting classification performance and improving dictionary quality over reconstructive ones. Classification results competitive with other state-of-the-art sparse coding methods are reported on several data sets.

1. Introduction

Since it was originally proposed for face recognition, the Sparse Representation-based Classification (SRC) [24] has received an increasing amount of attention, and it has been successfully used in the classification of various visual signals including facial expressions [6], hand written digits [25], and general images [5].

In SRC, a test signal $\mathbf{x}$ is represented as a sparse linear combination of the atoms in a dictionary $\mathbf{D}$ composed of training data from all classes, *i.e.* $\mathbf{x} = \mathbf{D}\boldsymbol{\alpha}$. If the signals in each class lie in a low-dimensional subspace and the subspaces of different classes satisfy certain incoherence conditions, it is speculated in [24] that all the nonzero coefficients in sparse code $\boldsymbol{\alpha}$ will be associated with the dictionary atoms that belong to the same class as $\mathbf{x}$. This argument has gained more theoretical support latterly from the analysis of sparse subspace clustering in [21], as classification can be regarded as clustering new data into existing clusters with known labels. However, due to noise corruption and subspace overlap, the nonzero coefficients in $\boldsymbol{\alpha}$ are usually associated with atoms from more than one class in practice. This problem is addressed in SRC by predicting the label as the class whose corresponding coefficients give the smallest reconstruction error of $\mathbf{x}$. Although such classification scheme shows effectiveness in many applications empirically, its working mechanism is obscure and there is no guarantee for the classification performance. Some attempts have been made to attribute the power of SRC to collaborative representation [28], but the analysis is quite limited.

Due to the absence of a feasible performance metric for SRC, the design of its dictionary (which serves as the parameter for both representation and classification) has been more or less heuristic so far. Originally, an SRC dictionary is constructed by directly including all the training samples [24], which is not efficient and practical when the size of training set is huge. Random sampling or clustering methods such as K-means can give a compact dictionary, but generative as well as discriminative capabilities are lost. Traditional dictionary learning methods specialized for sparse representation, such as Method of Optimal Direction (MOD) [8], K-SVD [1], and the ℓ_1-relaxed convex formulations [13, 15], all focus on minimizing signal reconstruction error and thus are not optimized for classification task. In order to promote the discriminative power of dictionaries, recent works have augmented the reconstructive objective function with additional discrimination terms; *e.g.*, fisher discriminant criterion [27], structural incoherence [20], class residual difference [16, 25] and mutual information [19]. Classification models other than SRC have also been used with sparse codes as inputs [4, 10, 14]. The discrimination metrics in all the above methods are not geared to the mechanism of SRC; moreover, the use of an extra classification model (often requiring one-versus-rest paradigm in multi-class cases) will multiply the number of parameters and increase the risk of over-fitting.

In this paper, we present a novel margin-based perspective towards SRC and propose a maximum margin performance metric that is specifically designed for learning the dictionaries of SRC. Large margin classifiers [2] are well studied by the machine learning community, and they have many desirable properties such as robustness to noise and outlier, and theoretical connection with generalization bound. Due to the complex nonlinear mapping induced by sparse coding, evaluating the margin of SRC is nontrivial. Based on the local stability of sparse code support, we show in Sec. 2 that the decision boundary of SRC is a continuous piecewise quadratic surface, and the margin of a sample is approximated as its distance to the tangent plane of the decision function in a local region where the support of sparse code is stable. Following the idea of Support Vector Machine (SVM), we propose in Sec. 3 to use the hinge loss of approximated margin as a metric for the classification performance and generalization capability of SRC. A stochastic gradient descent algorithm is then implemented to maximize the margin of SRC and obtain more discriminative dictionaries. To the best of our knowledge, we are the first to conduct margin analysis on SRC and optimize its dictionary by margin maximization. The experiments in Sec. 4 validate the effectiveness of our margin-based loss function in predicting classification performance. It is shown on several data sets that our algorithm can learn very compact dictionaries that attain much higher accuracies than the conventional dictionaries in SRC; the performance is also competitive with other state-of-the-art methods based on sparse coding. Sec. 5 draws conclusion and discusses future work.

2. Margin Analysis of SRC

2.1. Preliminary

Suppose our data sample $\mathbf{x}$ lies in the high dimensional space $\mathbb{R}^m$ and comes from one of the C classes with label $y \in \{1...C\}$. In SRC, a dictionary $\mathbf{D} \in \mathbb{R}^{m\times n}$ with n atoms is composed of C class-wise sub-dictionaries $\mathbf{D}_c \in \mathbb{R}^{m\times n_c}$ such that $\mathbf{D} = [\mathbf{D}_1, ..., \mathbf{D}_C] = [\mathbf{d}_1, ..., \mathbf{d}_n]$. Given $\mathbf{D}$, we can find the sparse code $\boldsymbol{\alpha} \in \mathbb{R}^n$ for signal $\mathbf{x}$ by solving the following LASSO problem:

$$\boldsymbol{\alpha} = \arg\min_{\mathbf{z}} \|\mathbf{D}\mathbf{z} - \mathbf{x}\|_2^2 + \lambda\|\mathbf{z}\|_1, \tag{1}$$

where $\lambda > 0$ is a constant. The sparse code can be decomposed into C sub-codes as $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1; ...; \boldsymbol{\alpha}_C]$, where each $\boldsymbol{\alpha}_c$ corresponds to the coefficients for sub-dictionary $\mathbf{D}_c$. SRC makes classification decision based on the residual of signal approximated by the sub-code of each class: $r_c = \|\mathbf{e}_c\|_2^2$, where $\mathbf{e}_c = \mathbf{D}_c\boldsymbol{\alpha}_c - \mathbf{x}$ is the reconstruction error vector for class c. The class label is then predicted as:

$$\hat{y} = \arg\min_c r_c. \tag{2}$$

More detailed explanation of SRC can be found in [24].

2.2. Local Decision Boundary for SRC

To perform margin-based analysis for SRC, we first need to find its classification decision boundary. Consider two classes c_1 and c_2, and assume the dictionary $\mathbf{D}$ is given. The decision function at sample $\mathbf{x}$ is simply defined as $f(\mathbf{x}) = r_{c_2} - r_{c_1} \gtrless 0$. $f(\mathbf{x})$ can be expanded as:

$$f(\mathbf{x}) = 2(\mathbf{D}_{c_1}\boldsymbol{\alpha}_{c_1} - \mathbf{D}_{c_2}\boldsymbol{\alpha}_{c_2})^T\mathbf{x} - \|\mathbf{D}_{c_1}\boldsymbol{\alpha}_{c_1}\|^2 + \|\mathbf{D}_{c_2}\boldsymbol{\alpha}_{c_2}\|^2. \tag{3}$$

Eq. (3) could be regarded as a linear hyper-plane in the space of data $\mathbf{x}$, if the sparse code $\boldsymbol{\alpha}$ was fixed. What complicates things here is that $\boldsymbol{\alpha}$ is also determined by $\mathbf{x}$ through the sparse coding model in (1), and the hyper-plane in (3) will change with any small change in $\mathbf{x}$. Expressing $\boldsymbol{\alpha}$ analytically as a function of $\mathbf{x}$ is not possible in general, unless we know a priori the support and sign vector of $\boldsymbol{\alpha}$. In the latter case, the non-zero part of $\boldsymbol{\alpha}$ can be found according to the optimal condition of LASSO solution [29]:

$$\boldsymbol{\alpha}_\Lambda = (\mathbf{D}_\Lambda^T\mathbf{D}_\Lambda)^{-1}(\mathbf{D}_\Lambda^T\mathbf{x} - \frac{\lambda}{2}\mathbf{s}_\Lambda), \tag{4}$$

where $\Lambda = \{j|\alpha_j \neq 0\}$ is the active set of sparse coefficients with cardinality $|\Lambda| = \|\boldsymbol{\alpha}\|_0 = s$, $\boldsymbol{\alpha}_\Lambda \in \mathbb{R}^s$ contains the sparse coefficients at these active locations, $\mathbf{D}_\Lambda \in \mathbb{R}^{m\times s}$ is composed of the columns in $\mathbf{D}$ corresponding to Λ, and $\mathbf{s}_\Lambda \in \mathbb{R}^s$ carries the signs (±1) of $\boldsymbol{\alpha}_\Lambda$. Although the active set Λ and sign vector $\mathbf{s}_\Lambda$ also depend on $\mathbf{x}$, it can be shown (in supplementary material) that they are locally stable if $\mathbf{x}$ changes by a small amount of $\Delta\mathbf{x}$ satisfying the following stability conditions:

$$\begin{cases} |\mathbf{d}_j^T\{\mathbf{e} + [\mathbf{D}_\Lambda(\mathbf{D}_\Lambda^T\mathbf{D}_\Lambda)^{-1}\mathbf{D}_\Lambda^T - \mathbf{I}]\Delta\mathbf{x}\}| \leq \frac{\lambda}{2}, \forall j \notin \Lambda \\ \mathbf{s}_\Lambda \odot [(\mathbf{D}_\Lambda^T\mathbf{D}_\Lambda)^{-1}\mathbf{D}_\Lambda^T\Delta\mathbf{x}] > -\mathbf{s}_\Lambda \odot \boldsymbol{\alpha}_\Lambda \end{cases}, \tag{5}$$

where $\odot$ denotes element-wise multiplication, and $\mathbf{e} = \mathbf{D}_\Lambda\boldsymbol{\alpha}_\Lambda - \mathbf{x}$ is the global reconstruction error. All the conditions in (5) are linear inequalities for $\Delta\mathbf{x}$. Therefore, the local neighborhood around $\mathbf{x}$ where the active set (and signs[1]) of signal's sparse code remains stable is a convex polytope.

Now substitute the sparse code terms in (3) with (4), and after some manipulations we obtain a quadratic local decision function $f_\Lambda(\mathbf{x})$ which is defined for any $\mathbf{x}$ whose sparse code corresponds to active set Λ:

$$\begin{aligned} f_\Lambda(\mathbf{x}) =& \mathbf{x}^T\boldsymbol{\Phi}_{c_2}^T\boldsymbol{\Phi}_{c_2}\mathbf{x} + 2\boldsymbol{\nu}_{c_2}^T\boldsymbol{\Phi}_{c_2}\mathbf{x} + \boldsymbol{\nu}_{c_2}^T\boldsymbol{\nu}_{c_2} \\ &- (\mathbf{x}^T\boldsymbol{\Phi}_{c_1}^T\boldsymbol{\Phi}_{c_1}\mathbf{x} + 2\boldsymbol{\nu}_{c_1}^T\boldsymbol{\Phi}_{c_1}\mathbf{x} + \boldsymbol{\nu}_{c_1}^T\boldsymbol{\nu}_{c_1}), \end{aligned} \tag{6}$$

where

$$\boldsymbol{\Phi}_c = \mathbf{D}_\Lambda\mathbf{P}_c(\mathbf{D}_\Lambda^T\mathbf{D}_\Lambda)^{-1}\mathbf{D}_\Lambda^T - \mathbf{I}, \tag{7}$$

$$\boldsymbol{\nu}_c = -\frac{\lambda}{2}\mathbf{D}_\Lambda\mathbf{P}_c(\mathbf{D}_\Lambda^T\mathbf{D}_\Lambda)^{-1}\mathbf{s}_\Lambda, \tag{8}$$

[1] In the following, the concept of sign vector $\mathbf{s}_\Lambda$ is included by default when we refer to "active set" or "Λ".

and $\mathbf{P}_c$ is an $s \times s$ diagonal matrix with 1 at positions corresponding to class c in the active set and 0 otherwise. The above analysis leads to the following proposition for the decision function of SRC.

Proposition 2.1 *The decision function of SRC is a piecewise quadratic function of input signal with the form of*

$$f(\mathbf{x}) = f_\Lambda(\mathbf{x}), \tag{9}$$

for any $\mathbf{x}$ *in the convex polytope defined by Eq.* (5) *where the active set* Λ *of its sparse code is stable.*

Since there are a set of quadratic decision functions each devoted to a local area of $\mathbf{x}$, SRC is capable of classifying data which cannot be linearly or quadratically separated in a global sense. The decision boundary of SRC can be adapted to each local area in the most discriminative and compact way, which shares a similar idea with locally adaptive metric learning [7]. On the other hand, these quadratic functions as well as the partition of local areas cannot be adjusted individually; they are all tied via a common model $\mathbf{D}$. This is crucial to reduce model complexity and enhance information sharing among different local regions, considering there could be as many as 3^n regions[2] out of the partition of the entire signal space.

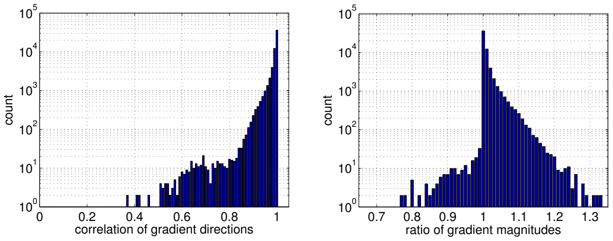
To find the decision boundary of SRC, we simply need to check at what values of $\mathbf{x}$, $f(\mathbf{x})$ will vary from positive to negative, as the decision threshold is 0. It has been show in [29] that the sparse code $\boldsymbol{\alpha}$ is a continuous function of $\mathbf{x}$. Thus we can easily see that $f(\mathbf{x})$ is also continuous over the entire domain of $\mathbf{x}$, and the points on the decision boundary of SRC have to satisfy $f(\mathbf{x}) = 0$, which is stated in the following proposition.

Proposition 2.2 *The decision boundary of SRC is a piecewise quadratic hypersurface defined by* $f(\mathbf{x}) = 0$.

2.3. Margin Approximation for SRC

For linear classifiers, the margin of a sample is defined as its distance from the decision hyperplane. In the context of SRC, we similarly define the margin of a sample $\mathbf{x}_0$ as its distance to the closest point on the decision boundary: $\min_{f(\mathbf{x})=0} \|\mathbf{x}_0 - \mathbf{x}\|_2$. Unfortunately, due to the complexity of SRC's decision function $f(\mathbf{x})$, it is difficult to evaluate the associated margin directly.

Instead, we estimate $\mathbf{x}_0$'s margin by approximating $f(\mathbf{x})$ with its tangent plane at $\mathbf{x}_0$. Such approximation is appropriate only when gradient $\nabla f(\mathbf{x})$ does not change too much as $f(\mathbf{x})$ descents from $f(\mathbf{x}_0)$ to 0, which is generally true based on the following observations. First, within each polytope for a stable active set Λ, $\nabla f_\Lambda(\mathbf{x})$ is a linear function of $\mathbf{x}$ and will not change a lot if $\mathbf{x}_0$ lies close to the boundary. Second, as implied by the empirical findings in Fig. 1, if we have two contiguous polytopes corresponding respectively to two stable active sets, Λ_1 and Λ_2, which are the same except for one entry, then with a high probability the gradient of decision function in the two polytopes will be approximately the same near their border: $\nabla f_{\Lambda_1} \approx \nabla f_{\Lambda_2}$. This observation allows us to approximate the decision function over a number of polytopes with a common tangent plane. Third, as will be discussed in Sec. 3, we are more interested in the data samples near the decision boundary when optimizing a large margin classifier. Thus, those faraway samples whose margins cannot be accurately approximated can be safely ignored. Therefore, our approximation is also suitable for the use with margin maximization.



(a) (b)

Figure 1. The histograms of the (a) correlation and (b) magnitude ratio between the decision function gradients ∇f_{Λ_1} and ∇f_{Λ_2} on the MNIST data set. ∇f_{Λ_1} is the gradient at original data $\mathbf{x}$, and ∇f_{Λ_2} is the gradient at data with a small perturbation $\Delta\mathbf{x}$ from $\mathbf{x}$, such that only one of the conditions in Eq. (5) is violated. Both (a) and (b) are highly peaked around 1.

Once the decision function $f(\mathbf{x})$ is linearly approximated, the margin γ of $\mathbf{x}_0$ is simply its distance (with sign) to the hyperplane $f(\mathbf{x}) = 0$:

$$\begin{aligned}\gamma(\mathbf{x}_0) =& \frac{f(\mathbf{x}_0)}{\|\nabla f(\mathbf{x}_0)\|_2} = \frac{f(\mathbf{x}_0)}{\|\nabla f_\Lambda(\mathbf{x}_0)\|_2} \\ =& \frac{r_{c_2} - r_{c_1}}{2\|\boldsymbol{\Phi}_{c_2}^T \mathbf{e}_{c_2} - \boldsymbol{\Phi}_{c_1}^T \mathbf{e}_{c_1}\|_2},\end{aligned} \tag{10}$$

where we use the relationship $\mathbf{e}_c = \boldsymbol{\Phi}_c\mathbf{x} + \boldsymbol{\nu}_c$ to simplify the expression in (10); all the $\boldsymbol{\Phi}_c$'s and $\boldsymbol{\nu}_c$'s are calculated according to (7) and (8) with the active set Λ of $\mathbf{x}_0$'s sparse code. It should be noted that the decision function gradient ∇f is not defined on the borders of convex polytopes with different active sets. In such a case, we just replace $\|\nabla f\|_2$ with the largest directional derivative evaluated in all the pertinent polytopes.

In SRC, all data samples are usually normalized onto the unit ball such that $\|\mathbf{x}\|_2 = 1$. In this way, the change of $f(\mathbf{x})$ in the direction of $\mathbf{x}_0$ itself should not be taken into account when we calculate the margin of $\mathbf{x}_0$. The margin

[2] Each atom can be assigned with a positive, negative, or zero coefficient.

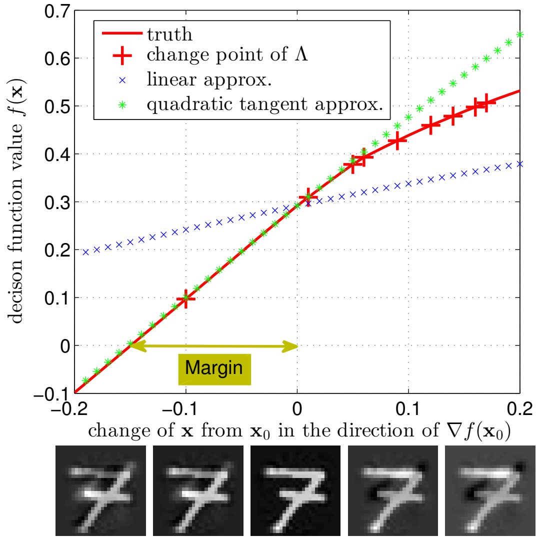


Figure 2. Top: decision function $f(\mathbf{x})$ for class "7" against class "4" in the MNIST data set and its approximations, where $\mathbf{x}$ changes in the 1D neighborhood of a sample $\mathbf{x}_0$ in the direction of gradient $\nabla f(\mathbf{x}_0)$. Bottom: the images of $\mathbf{x}$ as it moves in the direction of $\nabla f(\mathbf{x}_0)$ (from left to right). The central image corresponds to the original sample $\mathbf{x}_0$.

expression can be further amended as

$$\gamma(\mathbf{x}_0) = \frac{f(\mathbf{x}_0)}{\|\mathbf{M}\nabla f(\mathbf{x}_0)\|_2} = \frac{r_{c_2} - r_{c_1}}{2\|\mathbf{M}(\mathbf{\Phi}_{c_2}^T \mathbf{e}_{c_2} - \mathbf{\Phi}_{c_1}^T \mathbf{e}_{c_1})\|_2}, \tag{11}$$

where $\mathbf{M} = (\mathbf{I} - \mathbf{x}_0\mathbf{x}_0^T)$.

Fig. 2 graphically illustrates our margin approximation approach for one image sample $\mathbf{x}_0$ from class "7" in the MNIST digits data set. We evaluate the ground truth value of decision function $f(\mathbf{x})$ at a series of data points $\mathbf{x}$ in a 1D interval generated by shifting $\mathbf{x}_0$ along the direction of $\nabla f(\mathbf{x}_0)$, and record all the points where the active set of sparse code changes. We can see that the piecewise smooth $f(\mathbf{x})$ (plotted as a red curve) can be well approximated by the tangent of local quadratic decision function (green asterisk) in a neighborhood where the active set (whose stable region is delimitated by red plus) does not change too much. However, the linear approximation (blue cross) suggested by Eq. (3) is much less accurate, though they all intersect at point $\mathbf{x}_0$. The margin (indicated by golden arrow) we find for this example is very close to its true value. Fig. 2 also shows how the appearance of the image signal is distorted to the imposter class "4" from its true class "7" as it moves along the gradient of decision function.

3. Maximum-Margin Dictionary Learning

The concept of maximum margin has been widely employed in training classifiers, and it serves as the cornerstone of many popular models including SVM. The classical analysis on SVM [22] established the relationship between the margin of the training set and the classifier's generalization error bound. Recently, a similar effort has been made for sparsity-based linear predictive classifier [18], which motivates us to design the dictionary for SRC from a perspective based on the margin analysis given in Sec. 2.

Suppose we have a set of N labeled training data samples: $\{\mathbf{x}_i, y_i\}_{i=1\ldots N}$. Learning a discriminative dictionary $\mathbf{D}^*$ for SRC can be generally formulated as the following optimization problem:

$$\mathbf{D}^* = \arg\min_{\mathbf{D}\in\mathcal{D}} \frac{1}{N}\sum_i \mathcal{L}(\mathbf{x}_i, y_i; \mathbf{D}). \tag{12}$$

where $\mathcal{D}$ denotes $\mathbb{R}^{m\times n}$ dictionary space with unit-norm atoms. To maximize the margin of a training sample close to the decision boundary of SRC, we follow the similar idea in SVM and define the loss function $\mathcal{L}(\mathbf{x}, y; \mathbf{D})$ using a multiclass hinge function:

$$\mathcal{L}(\mathbf{x}, y; \mathbf{D}) = \sum_{c\neq y} \max\{0, -\gamma(\mathbf{x}, y, c) + b\}, \tag{13}$$

where b is a non-negative parameter controlling the minimum required margin between classes, and

$$\gamma(\mathbf{x}, y, c) = \frac{r_c - r_y}{2\|\mathbf{M}(\mathbf{\Phi}_c^T \mathbf{e}_c - \mathbf{\Phi}_y^T \mathbf{e}_y)\|_2}, \tag{14}$$

is the margin of sample $\mathbf{x}$ with label y calculated against a competing class $c \neq y$, which is adopted from Eq. (11). The loss function in (13) is zero if the sample margin is equal or greater than b; otherwise, it gives penalty linearly proportional to negative margin. Different from what is defined in SVM, the margin we use here is unnormalized since the unit dictionary atom constraint ensures the objective function is bounded. Moreover, (13) promotes multi-class margin by summing over all possible imposter classes c and optimizing the single parameter $\mathbf{D}$ that is shared by all classes. This offers an advantage over a set of one-versus-rest binary classifiers whose margins can only be optimized separately.

According to the numerator in (14), the residual difference between the correct and incorrect classes, $r_c - r_y$, should be maximized to achieve a large margin. Such requirement is consistent with the classification scheme in (2), and it has also been enforced in other dictionary learning algorithms such as [16]. In addition, we further introduce a novel term in the denominator of (14), which normalizes the nonuniform gradient of SRC decision function in different local regions and leads to a better estimation to the true sample margin.

3.1. Online Dictionary Learning

We solve the optimization problem in Eq. (12) using an online algorithm based on stochastic gradient descent method, which is usually favored when the objective function is an expectation over a large number of training samples [15]. In our algorithm, the dictionary is first initialized with a reasonable guess $\mathbf{D}^0$ (which can be the concatenation of sub-dictionaries obtained by applying K-means or random selection to each class). Then we go through the whole data set multiple times and iteratively update the dictionary with decreasing step size until convergence. In the t-th iteration, a single sample $(\mathbf{x}, y)$ is drawn from the data set randomly and the dictionary is updated in the direction of the gradient of its loss function:

$$\mathbf{D}^t = \mathbf{D}^{t-1} - \rho^t [\nabla_{\mathbf{D}} \mathcal{L}(\mathbf{x}, y; \mathbf{D}^{t-1})]^T, \tag{15}$$

where ρ^t is the step size at iteration t. It is selected as $\rho^t = \frac{\rho^0}{\sqrt{(t-1)/N+1}}$ with initial step size ρ^0. The gradient of our loss function is

$$\nabla_{\mathbf{D}} \mathcal{L}(\mathbf{x}, y; \mathbf{D}) = - \sum_{c \in \mathcal{C}(\mathbf{x}, y)} \nabla_{\mathbf{D}} \gamma(\mathbf{x}, y, c) \tag{16}$$

where $\mathcal{C}(\mathbf{x}, y) = \{c | c \neq y, \gamma(\mathbf{x}, y, c) < b\}$. We ignore those competing classes with zero margin gradient ($\gamma(\mathbf{x}, y, c) > b$) or zero sub-gradient ($\gamma(\mathbf{x}, y, c) = b$). The latter case occurs with very low probability in practice and thus will not affect the convergence of stochastic gradient descent as long as a suitable step size is chosen [3].

All that remains to be evaluated is $\nabla_{\mathbf{D}} \gamma(\mathbf{x}, y, c)$, which can be obtained by taking derivative of Eq. (14) with respect to $\mathbf{D}$. We realize from the results in [18] that the active set Λ for any particular sample $\mathbf{x}$ is stable when there is a small perturbation applied to dictionary $\mathbf{D}$. Since the approximation of margin is also based on a locally stable Λ, we can safely deduce that $\gamma(\mathbf{x}, y, c)$ is a differentiable function of $\mathbf{D}$. In this way, we circumvent the trouble of indifferentiability when directly taking derivative of sparse code with respect to $\mathbf{D}$ as has been done in [14, 26]. In addition, since (14) depends only on $\mathbf{D}_\Lambda$, we just need to update those dictionary atoms corresponding to the active set Λ of each sample $\mathbf{x}$. The dictionary updating rule in (15) can be rewritten as:

$$\mathbf{D}_\Lambda^t = \mathbf{D}_\Lambda^{t-1} + \rho^t \cdot [\nabla_{\mathbf{D}_\Lambda} \gamma(\mathbf{x}, y, c)]^T, \tag{17}$$

which is repeated for all $c \in \mathcal{C}(\mathbf{x}, y)$. The specific form of $\nabla_{\mathbf{D}_\Lambda} \gamma(\mathbf{x}, y, c)$ is given in supplementary material. Once the dictionary is updated in the current iteration, all its atoms are projected to the unit ball to comply with the constraint that $\mathbf{D} \in \mathcal{D}$. The overall Maximum-Margin Dictionary Learning (MMDL) approach is summarized in Algorithm 1.

Algorithm 1 Maximum-Margin Dictionary Learning (MMDL) for SRC

```
Input: labeled data set $\mathcal{S} = \{\mathbf{x}_i, y_i\}$, dictionary size $n$,
    sparse regularization $\lambda$, required margin $b$
Output: dictionary $\mathbf{D}$
initialize $\mathbf{D}$ with all class-wise sub-dictionaries $\mathbf{D}_c$ (obtained using K-means)
set $t = 1$
while not converge do
    randomly permute data set $\mathcal{S}$
    for each $(\mathbf{x}, y) \in \mathcal{S}$ do
        for each $c$ in $\mathcal{C}(\mathbf{x}, y)$ do
            update $\mathbf{D}_\Lambda$ according to Eq. (17)
        end for
        $\mathbf{d}_j \leftarrow \mathbf{d}_j / \|\mathbf{d}_j\|$ for each $j \in \Lambda$
        $t \leftarrow t + 1$
    end for
end while
return $\mathbf{D}$
```

3.2. Interpreting the Learning Algorithm

The gradient term in (17) takes a very complicated form as given in supplementary material. Nevertheless, some intuition can be obtained from its expression about how our algorithm works. We first notice that Eq. (17) will add $\mathbf{e}_c$ to all the active atoms associated with class c and subtract $\mathbf{e}_y$ from all the active atoms associated with class y, both with a scaling factor proportional to each atom's sparse coefficient. Such operation effectively "pulls" those active atoms of correct class towards the signal, and "pushes" those active atoms of imposter class away from the signal, which is similar to the strategies used to optimize codebook in Learning Vector Quantization (LVQ) [11] and Large Margin Nearest Neighbor (LMNN) [23]. In addition, (17) also uses the overall reconstruction error $\mathbf{e}$ and the projections of $\mathbf{e}_c$ and $\mathbf{e}_y$ as the ingredients to update the active atoms from *all* the classes, which is reasonable because the sparse code is jointly determined by all the active atoms.

On the other hand, we observe from Eq. (16) that only those difficult samples that have small margins against the imposter classes are selected to participate in dictionary training. Similar sample selection schemes are also found in LVQ and LMNN. Therefore, our choice of hinge loss function is supported from the perspective of other previously developed large-margin classifiers.

4. Experimental Results

4.1. Algorithm Analysis

To get a better understanding of the proposed method, we first conduct some analysis on its behavior in this section using a subset of 20,000 training samples from the MNIST [12] digits data set.

The accuracy of SRC margin approximation, which is key to the effectiveness of our method, is first investigated. Because it is impossible to find the exact margin of a sample

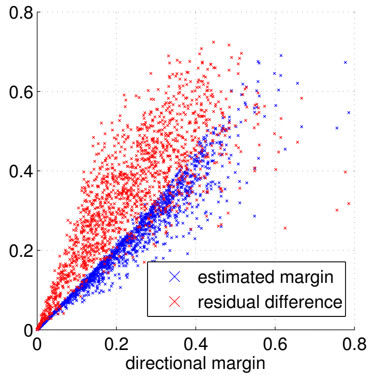


Figure 3. Distributions of estimated margin $\gamma(\mathbf{x})$ and residual different $r_{c_2} - r_{c_1}$ plotted against directional margin measured in the gradient direction $\nabla f(\mathbf{x})$.

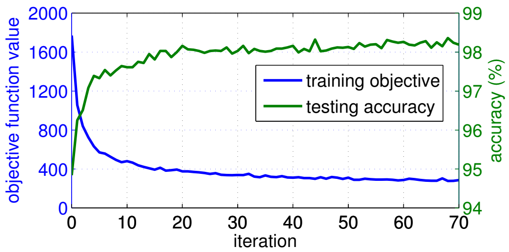


Figure 4. The objective function on training set and recognition accuracy on test set during the iterations of MMDL algorithm.

$\mathbf{x}_0$, we use the shortest distance between $\mathbf{x}_0$ and the decision boundary *in the gradient direction* $\nabla f(\mathbf{x}_0)$ as a surrogate to the ground truth margin. Such "directional margin" is found by a line search and plotted in Fig. 3 against the estimated margin $\gamma(\mathbf{x}_0)$ using Eq. (11) for all the samples. A strong linear relationship is observed between the directional and estimated margin, especially for those samples with small margins which are indeed to the interest of our algorithm. We also plot the distribution of residual difference $r_{c_2} - r_{c_1}$, which shows a weaker correlation with the directional margin. This justifies that maximizing $\gamma(\mathbf{x})$ as defined in (14) is a better choice than simply maximizing $r_{c_2} - r_{c_1}$ for large-margin optimization.

The behavior of our MMDL algorithm is examined in Fig. 4. The objective function value over the training samples decreases steadily and converges in about 70 iterations. At the same time, the recognition accuracy on a separate test set is remarkably improved during the iterations, indicating a good correspondence between our margin-based objective function and SRC's generalized performance [3].

The minimum required margin b in Eq. (13) is an important parameter in MMDL, whose effect on recognition performance is shown in Table 1. A too small value of b leads to over-fitting to training set, while a too large value leads to bias of the classification objective. We find $b = 0.1$ is generally a good choice on different data sets, and gradually reducing b during the iterations can help the algorithm focus more on those hard samples near decision boundary.

Table 1. The effect of parameter b on classification accuracy.

b	0	0.05	0.1	0.15	0.2
train acc.	100.00	100.00	99.44	98.45	97.39
test acc.	96.78	98.01	**98.13**	97.36	96.77

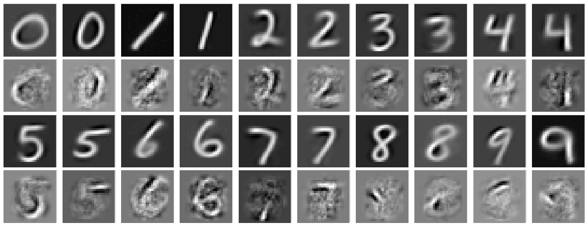


Figure 5. Dictionary atoms for MNIST digits data, learned using unsupervised sparse coding (row 1, 3) and MMDL (row 2, 4).

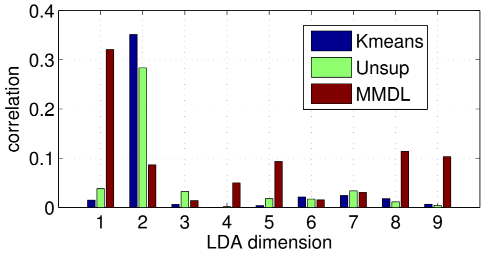


Figure 6. Correlation between the first principal component of atoms from different dictionaries and the LDA directions of the MNIST training data.

The image patterns of some dictionary atoms obtained using MMDL are shown in Fig. 5, together with those obtained using unsupervised sparse coding [13], which were used to initialize the dictionary in MMDL. The discriminative atoms trained with MMDL look quite different from their initial reconstructive appearances, and place more emphasis on local edge features that are unique to each class. The discriminative power of our learned dictionary can be further demonstrated in Fig. 6, which shows that, compared with K-means and unsupervised sparse coding, the MMDL algorithm learns dictionary atoms whose first principle component has a much higher correlation with most of the LDA directions (especially the first one) of the training data. Although LDA directions may not be optimal for SRC, our dictionary atoms appear to be more devoted to discriminative features instead of reconstructive ones.

4.2. Recognition Performance Evaluation

Now we report the recognition performance of the proposed method on several benchmark data sets. SRC is most well known for face recognition task, therefore we first test

[3] We do observe some small fluctuations on the testing accuracy, which is caused by the stochastic gradient descent.

Table 2. Recognition accuracies (%) on face databases.

Method	Extended YaleB	AR Face
Full	**97.34**	96.50
Subsample	91.20	73.17
KSVD [1]	88.63	90.00
Kmeans	95.44	90.00
Unsup [13]	96.35	90.33
LC-KSVD [10]	95.00	93.70
MMDL	**97.34**	**97.33**
Error reduction (%)	27.12	72.39

Table 3. Performance of SRC on the MNIST digits database.

Training method / Size of $\mathbf{D}$	Accuracy (%)
Subsample / 30000	98.05
Subsample / 150	82.19
Kmeans / 150	94.19
Unsup [13] / 150	94.84
Ramirez *et al.* [20] / 800	98.74
MMDL / 150	**98.76**
Error reduction (%)	75.97

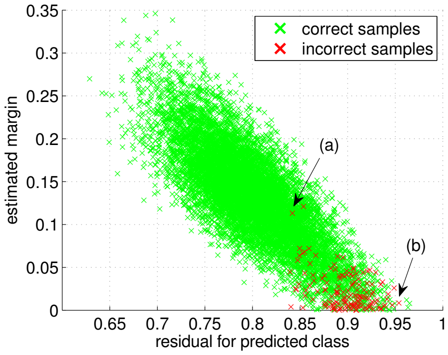


Figure 7. Distributions of correctly and incorrectly classified test samples plotted against estimated margin and reconstruction residual using the atoms from predicted class.

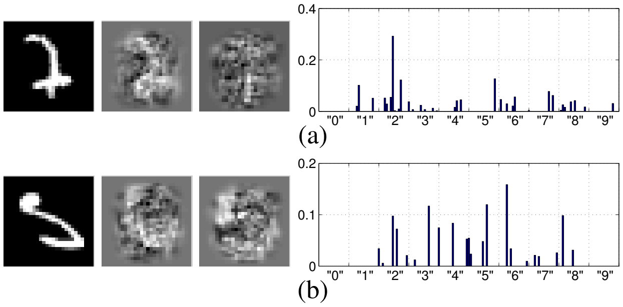


Figure 8. Two misclassified samples corresponding to the red crosses marked by (a) and (b) in Fig. 7. From left to right: original sample; reconstruction with atoms of predicted class; reconstruction with atoms of truth class; sparse coefficients.

on two face data sets: extended YaleB [9] and AR face [17]. We use 2,414 face images of 38 subjects from the extended YaleB data set, and a subset containing 2,600 images of 50 female and 50 male subjects from the AR face data set. We follow the procedure in [10] to split the training and test data, and obtain random projected face features of 504(540)-dimension for extended YaleB(AR face). For both data sets, we compare the performance of SRC with dictionaries obtained from the full training set ("Full"), random subsampling of training set ("Subsample"), KSVD [1], K-means ("Kmeans"), unsupervised sparse coding ("Unsup") [13], and our MMDL algorithm. Comparison with the state-of-the-art results of LC-KSVD [10] is also given, which employs a linear classification model on space codes. For extended YaleB(AR face), $15(5)$ atoms per subject are used for all the dictionaries expect for "Full", and λ is set as $0.01(0.005)$. As shown in Table 2, MMDL achieves the highest accuracies on both data sets, and outperforms the "Full" SRC on AR face using a much smaller dictionary. The huge reduction in the error rate of MMDL with respect to its initialization value given by "Unsup" further confirms the effectiveness of our learning algorithm.

Our method is also evaluated on the full MNIST data set, which contains 60,000 images for training and 10,000 for testing. We use PCA to reduce the dimension of each image such that 99% energy is preserved, and set $\lambda = 0.1$. Table 3 lists the classification accuracies of SRC with dictionaries trained using various methods and with different sizes. MMDL is shown to be advantageous over other methods in terms of both accuracy and dictionary compactness, the latter of which implies higher efficiency in computation as well as storage. Note that we are unable to evaluate SRC with the "Full" setting because the memory required for the operation on such a huge dictionary exceeds our system capacity (32GB).

Fig. 7 reveals the distinct distributions of correctly and incorrectly classified samples in terms of estimated margin and reconstruction residual with predicted class. The incorrect samples are observed to have higher residuals and smaller margins, which is expected since hard samples typically can not be well represented by the corresponding classes and lie close to the boundary of imposter classes. This provides another evidence to show the accuracy of our margin estimation. Therefore, the estimated margin can also serve as a metric of classification confidence, based on which the classification results could be further refined. Two cases of failed test samples are illustrated in Fig. 8. The digit "7" in (a) is misclassified as "2" with a large margin due to the strong inter-class similarity and high intra-class variation insufficiently captured by the training set. The digit "5" in (b) cannot be faithfully represented by any class; such an outlier has a very small margin and thus can be potentially detected for special treatment.

5. Conclusion and Future Directions

An in-depth analysis of the classification margin for SRC is presented in this paper. We show that the decision boundary of SRC is a continuous piecewise quadratic hypersurface, and it can be approximated by its tangent plane in a local region to facilitate the margin estimation. A learning algorithm based on stochastic gradient descent is derived to maximize the margins of training samples, which generates compact dictionaries with substantially improved discriminative power observed on several data sets.

In the future work, we will explore better ways to approximate the margin of samples far away from the decision boundary in the hope to further improve dictionary quality. It would also be of great interest to establish a strict relationship between the margin and generalization performance of SRC, so that a better knowledge can be gained about under what circumstances is SRC expected to perform best.

Acknowledgement

The work is supported by the U.S. Army Research Laboratory and U.S. Army Research Office under grant number W911NF-09-1-0383.

References

[1] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. Sig. Proc.*, 54(11):4311–4322, 2006.

[2] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *the 5th Annual Workshop on Computational Learning Theory*, pages 144–152, 1992.

[3] L. Bottou. Stochastic learning. In O. Bousquet and U. von Luxburg, editors, *Advanced Lectures on Machine Learning*, Lecture Notes in Artificial Intelligence, LNAI 3176, pages 146–168. Springer Verlag, 2004.

[4] D. M. Bradley and J. A. Bagnell. Differential sparse coding. In *Adv. NIPS*, pages 113–120, 2008.

[5] C.-K. Chiang, C.-H. Duan, S.-H. Lai, and S.-F. Chang. Learning component-level sparse representation using histogram information for image classification. In *Proc. ICCV*, pages 1519–1526, 2011.

[6] S. F. Cotter. Sparse representation for accurate classification of corrupted and occluded facial expressions. In *Proc. ICASSP*, pages 838–841, 2010.

[7] C. Domeniconi, J. Peng, and D. Gunopulos. Locally adaptive metric nearest-neighbor classification. *IEEE Trans. PAMI*, 24(9):1281–1285, 2002.

[8] K. Engan, S. O. Aase, and J. Hakon Husoy. Method of optimal directions for frame design. In *Proc. ICASSP*, pages 2443–2446, 1999.

[9] A. Georghiades, P. Belhumeur, and D. Kriegman. From few to many: illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. PAMI*, 23(6):643–660, 2001.

[10] Z. Jiang, Z. Lin, and L. S. Davis. Learning a discriminative dictionary for sparse coding via label consistent K-SVD. In *Proc. CVPR*, pages 1697–1704, 2011.

[11] T. Kohonen. Improved versions of learning vector quantization. In *IJCNN International Joint Conference on Neural Networks*, volume 1, pages 545–550, 1990.

[12] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[13] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In *Adv. NIPS*, pages 801–808, 2007.

[14] J. Mairal, F. Bach, and J. Ponce. Task-driven dictionary learning. *IEEE Trans. PAMI*, 32(4), 2012.

[15] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In *Proc. ICML*, pages 689–696, 2009.

[16] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Discriminative learned dictionaries for local image analysis. In *Proc. CVPR*, pages 1–8, 2008.

[17] A. Martinez and R. Benavente. The AR face database. CVC Technical Report 24, 1998.

[18] N. Mehta and A. Gray. Sparsity-based generalization bounds for predictive sparse coding. In *Proc. ICML*, 2013. accepted.

[19] Q. Qiu, Z. Jiang, and R. Chellappa. Sparse dictionary-based representation and recognition of action attributes. In *Proc. ICCV*, pages 707–714, 2011.

[20] I. Ramirez, P. Sprechmann, and G. Sapiro. Classification and clustering via dictionary learning with structured incoherence and shared features. In *Proc. CVPR*, pages 3501–3508, 2010.

[21] M. Soltanolkotabi and E. J. Candes. A geometric analysis of subspace clustering with outliers. *arXiv preprint arXiv:1112.4258*, 2011.

[22] V. Vapnik. *The nature of statistical learning theory*. springer, 1999.

[23] K. Q. Weinberger, J. Blitzer, and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. In *Adv. NIPS*, pages 1473–1480, 2006.

[24] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Trans. PAMI*, 31(2):210–227, 2009.

[25] J. Yang, J. Wang, and T. S. Huang. Learning the sparse representation for classification. In *Proc. ICME*, pages 1–6, 2011.

[26] J. Yang, Z. Wang, Z. Lin, X. Shu, and T. Huang. Bilevel sparse coding for coupled feature spaces. In *Proc. CVPR*, 2012.

[27] M. Yang, L. Zhang, X. Feng, and D. Zhang. Fisher discrimination dictionary learning for sparse representation. In *Proc. ICCV*, pages 543–550, 2011.

[28] L. Zhang, M. Yang, and X. Feng. Sparse representation or collaborative representation: Which helps face recognition? In *Proc. ICCV*, pages 471–478, 2011.

[29] H. Zou, T. Hastie, and R. Tibshirani. On the “degrees of freedom” of the LASSO. *The Annals of Statistics*, 35(5):2173–2192, 2007.

A. Derivation of the Active Set Stability Conditions in Eq. (5)

Let $\boldsymbol{\alpha}$ be the sparse code for signal $\mathbf{x}$ as given by Eq. (1). When $\mathbf{x}$ is perturbed by $\Delta\mathbf{x}$, the sparse code changes by $\Delta\boldsymbol{\alpha}$. If the new sparse code $\boldsymbol{\alpha} + \Delta\boldsymbol{\alpha}$ shares the same support Λ and sign vector $\mathbf{s}_\Lambda$ with $\boldsymbol{\alpha}$, then the following conditions for LASSO solution must be satisfied:

$$|\mathbf{d}_j^T[\mathbf{x} + \Delta\mathbf{x} - \mathbf{D}_\Lambda(\boldsymbol{\alpha}_\Lambda + \Delta\boldsymbol{\alpha}_\Lambda)]| \leq \lambda/2,\ \forall j \notin \Lambda, \tag{A.1}$$

$$\mathbf{d}_j^T[\mathbf{x} + \Delta\mathbf{x} - \mathbf{D}_\Lambda(\boldsymbol{\alpha}_\Lambda + \Delta\boldsymbol{\alpha}_\Lambda)] = \lambda/2,\ \forall j \in \Lambda, \tag{A.2}$$

$$\mathbf{s}_\Lambda \odot (\boldsymbol{\alpha}_\Lambda + \Delta\boldsymbol{\alpha}_\Lambda) > \mathbf{0}. \tag{A.3}$$

From the equality condition (A.2), it is easy to see that

$$\Delta\boldsymbol{\alpha}_\Lambda = (\mathbf{D}_\Lambda^T\mathbf{D}_\Lambda)^{-1}\mathbf{D}_\Lambda^T\Delta\mathbf{x}. \tag{A.4}$$

Plug (A.4) into (A.1) and (A.3), and we can obtain the conditions in (5).

B. Calculation of the Margin Gradient in Eq. (17)

Denote $\mathbf{q}_c = \boldsymbol{\Phi}_c^T\mathbf{e}_c - \boldsymbol{\Phi}_y^T\mathbf{e}_y$ and $Q_c = \|\mathbf{M}\mathbf{q}_c\|$. The gradient of margin defined in Eq. (14) can be expressed as

$$\nabla_{\mathbf{D}}\gamma(\mathbf{x}, y, c) = \frac{1}{2Q_c^2}[\nabla_{\mathbf{D}}(r_c - r_y)Q_c - (r_c - r_y)Q_c^{-1}\mathbf{q}_c^T\mathbf{M}^T\mathbf{M}\nabla_{\mathbf{D}}(\boldsymbol{\Phi}_c^T\mathbf{e}_c - \boldsymbol{\Phi}_y^T\mathbf{e}_y)]. \tag{B.1}$$

Eq. (B.1) entails the evaluation of the following derivatives for any particular class c:

$$\nabla_{\mathbf{D}}r_c = \frac{\partial\|\mathbf{e}_c\|^2}{\partial\mathbf{e}_c}\nabla_{\mathbf{D}}\mathbf{e}_c = 2{\mathbf{e}_c}^T(\nabla_{\mathbf{D}}\boldsymbol{\Phi}_c\mathbf{x} + \nabla_{\mathbf{D}}\boldsymbol{\nu}_c), \tag{B.2}$$

and

$$\nabla_{\mathbf{D}}(\boldsymbol{\Phi}_c^T\mathbf{e}_c) = \nabla_{\mathbf{D}}(\boldsymbol{\Phi}_c)^T\mathbf{e}_c + \boldsymbol{\Phi}_c^T\nabla_{\mathbf{D}}\mathbf{e}_c = \nabla_{\mathbf{D}}(\boldsymbol{\Phi}_c)^T\mathbf{e}_c + \boldsymbol{\Phi}_c^T(\nabla_{\mathbf{D}}\boldsymbol{\Phi}_c\mathbf{x} + \nabla_{\mathbf{D}}\boldsymbol{\nu}_c). \tag{B.3}$$

We still need to find $\nabla_{\mathbf{D}}\boldsymbol{\Phi}_c$ and $\nabla_{\mathbf{D}}\boldsymbol{\nu}_c$. Their derivatives with respect to the (i, j)-th element of $\mathbf{D}$, d_{ij}, are

$$\begin{aligned}
\frac{\partial\boldsymbol{\Phi}_c}{\partial d_{ij}} &= \frac{\partial\mathbf{D}_\Lambda}{\partial d_{ij}}\mathbf{P}_c(\mathbf{D}_\Lambda^T\mathbf{D}_\Lambda)^{-1}\mathbf{D}_\Lambda^T + \mathbf{D}_\Lambda\mathbf{P}_c\left[\frac{\partial(\mathbf{D}_\Lambda^T\mathbf{D}_\Lambda)^{-1}}{\partial d_{ij}}\mathbf{D}_\Lambda^T + (\mathbf{D}_\Lambda^T\mathbf{D}_\Lambda)^{-1}\frac{\partial\mathbf{D}_\Lambda^T}{\partial d_{ij}}\right] \\
&= I(j \in \Lambda)\cdot\left[I(cls(j) = c)\cdot\mathbf{u}_i\mathbf{A}^{-1}(\Lambda^{-1}(j), :)\mathbf{D}_\Lambda^T - \mathbf{D}_\Lambda\mathbf{P}_c\mathbf{A}^{-1}\frac{\partial\mathbf{D}_\Lambda^T\mathbf{D}_\Lambda}{\partial d_{ij}}\mathbf{A}^{-1}\mathbf{D}_\Lambda^T + \mathbf{D}_\Lambda\mathbf{P}_c\mathbf{A}^{-1}(:, \Lambda^{-1}(j))\mathbf{u}_i^T\right] \\
&= I(j \in \Lambda)\cdot\Big[I(cls(j) = c)\cdot\mathbf{u}_i\mathbf{A}^{-1}(\Lambda^{-1}(j), :)\mathbf{D}_\Lambda^T - \mathbf{D}_\Lambda\mathbf{P}_c\mathbf{A}^{-1}\mathbf{u}_{\Lambda^{-1}(j)}\mathbf{D}_\Lambda(i, :)\mathbf{A}^{-1}\mathbf{D}_\Lambda^T \\
&\quad -\mathbf{D}_\Lambda\mathbf{P}_c\mathbf{A}^{-1}\mathbf{D}_\Lambda(i, :)^T\mathbf{u}_{\Lambda^{-1}(j)}^T\mathbf{A}^{-1}\mathbf{D}_\Lambda^T + \mathbf{D}_\Lambda\mathbf{P}_c\mathbf{A}^{-1}(:, \Lambda^{-1}(j))\mathbf{u}_i^T\Big] \\
&= I(j \in \Lambda)\cdot\Big[I(cls(j) = c)\cdot\mathbf{u}_i\mathbf{A}^{-1}(\Lambda^{-1}(j), :)\mathbf{D}_\Lambda^T - \mathbf{D}_\Lambda\mathbf{P}_c\mathbf{A}^{-1}(:, \Lambda^{-1}(j))\mathbf{D}_\Lambda(i, :)\mathbf{A}^{-1}\mathbf{D}_\Lambda^T \\
&\quad -\mathbf{D}_\Lambda\mathbf{P}_c\mathbf{A}^{-1}\mathbf{D}_\Lambda(i, :)^T\mathbf{A}^{-1}(\Lambda^{-1}(j), :)\mathbf{D}_\Lambda^T + \mathbf{D}_\Lambda\mathbf{P}_c\mathbf{A}^{-1}(:, \Lambda^{-1}(j))\mathbf{u}_i^T\Big],
\end{aligned} \tag{B.4}$$

and

$$\begin{aligned}
\frac{\partial\boldsymbol{\nu}_c}{\partial d_{ij}} &= -\frac{\lambda}{2}\frac{\partial\mathbf{D}_\Lambda}{\partial d_{ij}}\mathbf{P}_c(\mathbf{D}_\Lambda^T\mathbf{D}_\Lambda)^{-1}\mathbf{s}_\Lambda + \frac{\lambda}{2}\mathbf{D}_\Lambda\mathbf{P}_c(\mathbf{D}_\Lambda^T\mathbf{D}_\Lambda)^{-1}\frac{\partial\mathbf{D}_\Lambda^T\mathbf{D}_\Lambda}{\partial d_{ij}}(\mathbf{D}_\Lambda^T\mathbf{D}_\Lambda)^{-1}\mathbf{s}_\Lambda \\
&= -\frac{\lambda}{2}I(j \in \Lambda)\cdot\big[I(cls(j) = c)\cdot\mathbf{u}_i\mathbf{A}^{-1}(\Lambda^{-1}(j), :)\mathbf{s}_\Lambda - \mathbf{D}_\Lambda\mathbf{P}_c\mathbf{A}^{-1}(:, \Lambda^{-1}(j))\mathbf{D}_\Lambda(i, :)\mathbf{A}^{-1}\mathbf{s}_\Lambda \\
&\quad -\mathbf{D}_\Lambda\mathbf{P}_c\mathbf{A}^{-1}\mathbf{D}_\Lambda(i, :)^T\mathbf{A}^{-1}(\Lambda^{-1}(j), :)\mathbf{s}_\Lambda\big],
\end{aligned} \tag{B.5}$$

where $I(\cdot)$ is the indicator function, $cls(j)$ is the class label for j-th dictionary atom, $\mathbf{A} = \mathbf{D}_\Lambda^T\mathbf{D}_\Lambda$ [4], $\mathbf{u}_i$ is $m \times 1$ unit column vector with i-th element equal to 1, $\mathbf{u}_{\Lambda^{-1}(j)}$ is $s \times 1$ unit column vector with $\Lambda^{-1}(j)$-th element equal to 1, $\Lambda(k)$[5] denotes

[4] In practice, we set $\mathbf{A} = \mathbf{D}_\Lambda^T\mathbf{D}_\Lambda + \epsilon \cdot \mathbf{I}$ to ensure the stability of the inverse of $\mathbf{A}$, where ϵ is a small positive constant.

[5] Here Λ is used to denote set or function interchangeably, depending on its context.

the k-th element of Λ in ascending order, and $\Lambda^{-1}(\cdot)$ is the inverse function of $\Lambda(\cdot)$. Plug Eq. (B.4) and (B.5) into Eq. (B.2) and (B.3), and we can obtain

$$\begin{aligned}
\frac{\partial r_c}{\partial d_{ij}} &= 2I(j\in\Lambda)\cdot{\mathbf{e}_c}^T\Big\{\big[I(cls(j)=c)\cdot\mathbf{u}_i\mathbf{A}^{-1}(\Lambda^{-1}(j),:)\mathbf{D}_\Lambda^T-\mathbf{D}_\Lambda\mathbf{P}_c\mathbf{A}^{-1}(:,\Lambda^{-1}(j))\mathbf{D}_\Lambda(i,:)\mathbf{A}^{-1}\mathbf{D}_\Lambda^T\\
&\quad -\mathbf{D}_\Lambda\mathbf{P}_c\mathbf{A}^{-1}\mathbf{D}_\Lambda(i,:)^T\mathbf{A}^{-1}(\Lambda^{-1}(j),:)\mathbf{D}_\Lambda^T+\mathbf{D}_\Lambda\mathbf{P}_c\mathbf{A}^{-1}(:,\Lambda^{-1}(j))\mathbf{u}_i^T\big]\,\mathbf{x}\\
&\quad -\frac{\lambda}{2}\big[I(cls(j)=c)\cdot\mathbf{u}_i\mathbf{A}^{-1}(\Lambda^{-1}(j),:)\mathbf{s}_\Lambda-\mathbf{D}_\Lambda\mathbf{P}_c\mathbf{A}^{-1}(:,\Lambda^{-1}(j))\mathbf{D}_\Lambda(i,:)\mathbf{A}^{-1}\mathbf{s}_\Lambda\\
&\quad -\mathbf{D}_\Lambda\mathbf{P}_c\mathbf{A}^{-1}\mathbf{D}_\Lambda(i,:)^T\mathbf{A}^{-1}(\Lambda^{-1}(j),:)\mathbf{s}_\Lambda\big]\Big\}\\
&= 2I(j\in\Lambda)\cdot{\mathbf{e}_c}^T\big\{I(cls(j)=c)\cdot\mathbf{u}_i\mathbf{A}^{-1}(\Lambda^{-1}(j),:)\mathbf{D}_\Lambda^T\mathbf{x}-\mathbf{D}_\Lambda\mathbf{P}_c\mathbf{A}^{-1}(:,\Lambda^{-1}(j))\mathbf{D}_\Lambda(i,:)\mathbf{A}^{-1}\mathbf{D}_\Lambda^T\mathbf{x}\\
&\quad -\mathbf{D}_\Lambda\mathbf{P}_c\mathbf{A}^{-1}\mathbf{D}_\Lambda(i,:)^T\mathbf{A}^{-1}(\Lambda^{-1}(j),:)\mathbf{D}_\Lambda^T\mathbf{x}+\mathbf{D}_\Lambda\mathbf{P}_c\mathbf{A}^{-1}(:,\Lambda^{-1}(j))\cdot x_i\\
&\quad -I(cls(j)=c)\frac{\lambda}{2}\cdot\mathbf{u}_i\mathbf{A}^{-1}(\Lambda^{-1}(j),:)\mathbf{s}_\Lambda+\frac{\lambda}{2}\mathbf{D}_\Lambda\mathbf{P}_c\mathbf{A}^{-1}(:,\Lambda^{-1}(j))\mathbf{D}_\Lambda(i,:)\mathbf{A}^{-1}\mathbf{s}_\Lambda\\
&\quad +\frac{\lambda}{2}\mathbf{D}_\Lambda\mathbf{P}_c\mathbf{A}^{-1}\mathbf{D}_\Lambda(i,:)^T\mathbf{A}^{-1}(\Lambda^{-1}(j),:)\mathbf{s}_\Lambda\big\}\\
&= 2I(j\in\Lambda)\cdot{\mathbf{e}_c}^T\big\{I(cls(j)=c)\alpha_j\cdot\mathbf{u}_i-\mathbf{D}_\Lambda\mathbf{P}_c\left[e_i\cdot\mathbf{A}^{-1}(:,\Lambda^{-1}(j))+\alpha_j\cdot\mathbf{A}^{-1}\mathbf{D}_\Lambda(i,:)^T\right]\big\}, \qquad \text{(B.6)}
\end{aligned}$$

and

$$\begin{aligned}
&\frac{\partial(\mathbf{\Phi}_c^T\mathbf{e}_c)}{\partial d_{ij}}\\
&= I(j\in\Lambda)\cdot\big[I(cls(j)=c)\cdot\mathbf{D}_\Lambda\mathbf{A}^{-1}(:,\Lambda^{-1}(j))\mathbf{u}_i^T-\mathbf{D}_\Lambda\mathbf{A}^{-1}\mathbf{D}_\Lambda(i,:)^T\mathbf{A}^{-1}(\Lambda^{-1}(j),:)\mathbf{P}_c\mathbf{D}_\Lambda^T\\
&\quad -\mathbf{D}_\Lambda\mathbf{A}^{-1}(:,\Lambda^{-1}(j))\mathbf{D}_\Lambda(i,:)\mathbf{A}^{-1}\mathbf{P}_c\mathbf{D}_\Lambda^T+\mathbf{u}_i\mathbf{A}^{-1}(\Lambda^{-1}(j),:)\mathbf{P}_c\mathbf{D}_\Lambda^T\big]\,\mathbf{e}_c\\
&\quad +I(j\in\Lambda)\cdot\mathbf{\Phi}_c^T\big\{I(cls(j)=c)\alpha_j\cdot\mathbf{u}_i-\mathbf{D}_\Lambda\mathbf{P}_c\left[e_i\cdot\mathbf{A}^{-1}(:,\Lambda^{-1}(j))+\alpha_j\cdot\mathbf{A}^{-1}\mathbf{D}_\Lambda(i,:)^T\right]\big\}\\
&= I(j\in\Lambda)\cdot\big\{I(cls(j)=c)\cdot\mathbf{D}_\Lambda\mathbf{A}^{-1}(:,\Lambda^{-1}(j))e_{ci}-[\mathbf{A}^{-1}(\Lambda^{-1}(j),:)\mathbf{P}_c\mathbf{D}_\Lambda^T\mathbf{e}_c]\cdot\mathbf{D}_\Lambda\mathbf{A}^{-1}\mathbf{D}_\Lambda(i,:)^T\\
&\quad -[\mathbf{D}_\Lambda(i,:)\mathbf{A}^{-1}\mathbf{P}_c\mathbf{D}_\Lambda^T\mathbf{e}_c]\cdot\mathbf{D}_\Lambda\mathbf{A}^{-1}(:,\Lambda^{-1}(j))+[\mathbf{A}^{-1}(\Lambda^{-1}(j),:)\mathbf{P}_c\mathbf{D}_\Lambda^T\mathbf{e}_c]\cdot\mathbf{u}_i\\
&\quad +\mathbf{\Phi}_c^T\left[I(cls(j)=c)\alpha_j\cdot\mathbf{u}_i-e_i\cdot\mathbf{D}_\Lambda\mathbf{P}_c\mathbf{A}^{-1}(:,\Lambda^{-1}(j))-\alpha_j\cdot\mathbf{D}_\Lambda\mathbf{P}_c\mathbf{A}^{-1}\mathbf{D}_\Lambda(i,:)^T\right]\big\}, \qquad \text{(B.7)}
\end{aligned}$$

where $\mathbf{e}=\mathbf{D}_\Lambda\boldsymbol{\alpha}_\Lambda-\mathbf{x}$. It is clear from Eq. (B.6) and Eq. (B.7) that the margin gradient in Eq. (B.1) with respect to all the inactive dictionary atoms $\{\mathbf{d}_j|j\notin\Lambda\}$ is zero.

The non-zero derivatives with respect to the j-th atom $\mathbf{d}_j$ with $j\in\Lambda$ can be rewritten as

$$\begin{aligned}
\frac{\partial r_c}{\partial \mathbf{d}_j} &= 2\alpha_j I(cls(j)=c)\cdot\mathbf{e}_c^T-2\mathbf{e}_c^T\mathbf{D}_\Lambda\mathbf{P}_c\mathbf{A}^{-1}(:,\Lambda^{-1}(j))\cdot\mathbf{e}^T-2\alpha_j\cdot\mathbf{e}_c^T\mathbf{D}_\Lambda\mathbf{P}_c\mathbf{A}^{-1}\mathbf{D}_\Lambda^T\\
&= 2\alpha_j[I(cls(j)=c)-1]\cdot\mathbf{e}_c^T-2\mathbf{e}_c^T\boldsymbol{\beta}_{c,\Lambda^{-1}(j)}\cdot\mathbf{e}^T-2\alpha_j\cdot\mathbf{e}_c^T\mathbf{\Phi}_c, \qquad \text{(B.8)}
\end{aligned}$$

and

$$\begin{aligned}
\frac{\partial(\mathbf{\Phi}_c^T\mathbf{e}_c)}{\partial \mathbf{d}_j} &= I(cls(j)=c)\cdot\mathbf{D}_\Lambda\mathbf{A}^{-1}(:,\Lambda^{-1}(j))\mathbf{e}_c^T-[\mathbf{A}^{-1}(\Lambda^{-1}(j),:)\mathbf{P}_c\mathbf{D}_\Lambda^T\mathbf{e}_c]\cdot\mathbf{D}_\Lambda\mathbf{A}^{-1}\mathbf{D}_\Lambda^T\\
&\quad -\mathbf{D}_\Lambda\mathbf{A}^{-1}(:,\Lambda^{-1}(j))\mathbf{e}_c^T(\mathbf{\Phi}_c+\mathbf{I})+[\mathbf{A}^{-1}(\Lambda^{-1}(j),:)\mathbf{P}_c\mathbf{D}_\Lambda^T\mathbf{e}_c]\cdot\mathbf{I}\\
&\quad +\mathbf{\Phi}_c^T\left[I(cls(j)=c)\alpha_j\cdot\mathbf{I}-\mathbf{D}_\Lambda\mathbf{P}_c\mathbf{A}^{-1}(:,\Lambda^{-1}(j))\mathbf{e}^T-\alpha_j\cdot(\mathbf{\Phi}_c+\mathbf{I})\right]\\
&= \boldsymbol{\beta}_{\Lambda^{-1}(j)}\mathbf{e}_c^T\{[I(cls(j)=c)-1]\cdot\mathbf{I}-\mathbf{\Phi}_c\}+(\mathbf{e}_c^T\boldsymbol{\beta}_{c,\Lambda^{-1}(j)})\cdot(\mathbf{I}-\mathbf{D}_\Lambda\mathbf{A}^{-1}\mathbf{D}_\Lambda^T)\\
&\quad +\mathbf{\Phi}_c^T\left\{[I(cls(j)=c)-1]\alpha_j\cdot\mathbf{I}-\boldsymbol{\beta}_{c,\Lambda^{-1}(j)}\mathbf{e}^T-\alpha_j\cdot\mathbf{\Phi}_c\right\}, \qquad \text{(B.9)}
\end{aligned}$$

where $\boldsymbol{\beta}_{c,\Lambda^{-1}(j)}=\mathbf{D}_\Lambda\mathbf{P}_c\mathbf{A}^{-1}(:,\Lambda^{-1}(j))$, and $\boldsymbol{\beta}_{\Lambda^{-1}(j)}=\mathbf{D}_\Lambda\mathbf{A}^{-1}(:,\Lambda^{-1}(j))$. Combining the derivatives of all the active atoms, we can obtain the derivatives with respect to $\mathbf{D}_\Lambda$:

$$\frac{\partial r_c}{\partial \mathbf{D}_\Lambda}=2(\mathbf{P}_c-\mathbf{I})\boldsymbol{\alpha}_\Lambda\mathbf{e}_c^T-2\boldsymbol{\beta}_c^T\mathbf{e}_c\mathbf{e}^T-2\boldsymbol{\alpha}_\Lambda\mathbf{e}_c^T\mathbf{\Phi}_c, \qquad \text{(B.10)}$$

and

$$\frac{\mathbf{q}_c^T \mathbf{M}^T \mathbf{M} \partial(\mathbf{\Phi}_c^T \mathbf{e}_c)}{\partial \mathbf{D}_\Lambda} = (\mathbf{P}_c - \mathbf{I})\boldsymbol{\beta}^T \tilde{\mathbf{q}}_c \mathbf{e}_c^T - \boldsymbol{\beta}^T \tilde{\mathbf{q}}_c \mathbf{e}_c^T \mathbf{\Phi}_c + \boldsymbol{\beta}_c^T \mathbf{e}_c \tilde{\mathbf{q}}_c^T (\mathbf{I} - \boldsymbol{\beta}\mathbf{D}_\Lambda^T) + (\mathbf{P}_c - \mathbf{I})\boldsymbol{\alpha}_\Lambda \tilde{\mathbf{q}}_c^T \mathbf{\Phi}_c^T - \boldsymbol{\beta}_c^T \mathbf{\Phi}_c \tilde{\mathbf{q}}_c \mathbf{e}^T - \boldsymbol{\alpha}_\Lambda \tilde{\mathbf{q}}_c^T \mathbf{\Phi}_c^T \mathbf{\Phi}_c, \tag{B.11}$$

where $\tilde{\mathbf{q}}_c = \mathbf{M}^T \mathbf{M} \mathbf{q}_c$. Now plug Eq. (B.10) and (B.11) into Eq. (B.1), and we arrive at

$$\nabla_{\mathbf{D}_\Lambda} \gamma(\mathbf{x}, y, c) = \sum_{i=1}^{6} \nabla_{\mathbf{D}_\Lambda} \gamma^i(\mathbf{x}, y, c), \tag{B.12}$$

where

$$\begin{aligned}
\nabla_{\mathbf{D}_\Lambda} \gamma^1(\mathbf{x}, y, c) =& \mathbf{P}_c \left\{ w_1 \cdot \boldsymbol{\alpha}_\Lambda + w_2 \cdot \boldsymbol{\beta}^T \tilde{\mathbf{q}}_c \right\} \mathbf{e}_c^T - \mathbf{P}_y \left\{ w_1 \cdot \boldsymbol{\alpha}_\Lambda + w_2 \cdot \boldsymbol{\beta}^T \tilde{\mathbf{q}}_c \right\} \mathbf{e}_y^T, \\
\nabla_{\mathbf{D}_\Lambda} \gamma^2(\mathbf{x}, y, c) =& -\{ w_1 \cdot (\boldsymbol{\beta}_c^T \mathbf{e}_c - \boldsymbol{\beta}_y^T \mathbf{e}_y) + w_2 \cdot (\boldsymbol{\beta}_c^T \mathbf{\Phi}_c - \boldsymbol{\beta}_y^T \mathbf{\Phi}_y) \tilde{\mathbf{q}}_c \} \mathbf{e}^T, \\
\nabla_{\mathbf{D}_\Lambda} \gamma^3(\mathbf{x}, y, c) =& -\left\{ w_1 \cdot \boldsymbol{\alpha}_\Lambda + w_2 \cdot \boldsymbol{\beta}^T \tilde{\mathbf{q}}_c \right\} [\mathbf{e}_c^T (\mathbf{\Phi}_c + \mathbf{I}) - \mathbf{e}_y^T (\mathbf{\Phi}_y + \mathbf{I})], \\
\nabla_{\mathbf{D}_\Lambda} \gamma^4(\mathbf{x}, y, c) =& w_2 \cdot \left\{ \boldsymbol{\beta}_c^T \mathbf{e}_c - \boldsymbol{\beta}_y^T \mathbf{e}_y \right\} \tilde{\mathbf{q}}_c^T (\mathbf{I} - \boldsymbol{\beta} \mathbf{D}_\Lambda^T), \\
\nabla_{\mathbf{D}_\Lambda} \gamma^5(\mathbf{x}, y, c) =& \mathbf{P}_c \{ w_2 \cdot \boldsymbol{\alpha}_\Lambda \} \tilde{\mathbf{q}}_c^T \mathbf{\Phi}_c^T - \mathbf{P}_y \{ w_2 \cdot \boldsymbol{\alpha}_\Lambda \} \tilde{\mathbf{q}}_c^T \mathbf{\Phi}_y^T, \\
\nabla_{\mathbf{D}_\Lambda} \gamma^6(\mathbf{x}, y, c) =& -\{ w_2 \cdot \boldsymbol{\alpha}_\Lambda \} \tilde{\mathbf{q}}_c^T [\mathbf{\Phi}_c^T (\mathbf{\Phi}_c + \mathbf{I}) - \mathbf{\Phi}_y^T (\mathbf{\Phi}_y + \mathbf{I})],
\end{aligned}$$

and $w_1 = \frac{1}{Q_c}$, $w_2 = -\frac{1}{2Q_c^3}(r_c - r_y)$, $\boldsymbol{\beta} = \mathbf{D}_\Lambda \mathbf{A}^{-1}$, $\boldsymbol{\beta}_c = \mathbf{D}_\Lambda \mathbf{P}_c \mathbf{A}^{-1}$. We notice that the first term in Eq. (B.12) adjusts the atoms associated with classes y and c with their respective reconstruction error vectors $\mathbf{e}_y$ and $\mathbf{e}_c$. The second term adjusts all the active atoms with global error vector $\mathbf{e}$. All the other terms adjust the dictionary atoms with various projections of $\mathbf{e}_y$ and $\mathbf{e}_c$.