Pattern Recognition 46 (2013) 1277–1287

Contents lists available at SciVerse ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Locality-sensitive dictionary learning for sparse representation based classification

Chia-Po Wei [a], Yu-Wei Chao [b], Yi-Ren Yeh [a], Yu-Chiang Frank Wang [a],*

[a] Research Center for Information Technology Innovation (CITI), Academia Sinica, Taipei 115, Taiwan
[b] Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109-2122, USA

ARTICLE INFO

Article history:
Received 31 March 2012
Received in revised form
1 November 2012
Accepted 13 November 2012
Available online 23 November 2012

Keywords:
Sparse representation
Dictionary learning
Data locality

ABSTRACT

Motivated by image reconstruction, sparse representation based classification (SRC) has been shown to be an effective method for applications like face recognition. In this paper, we propose a locality-sensitive dictionary learning algorithm for SRC, in which the designed dictionary is able to preserve local data structure, resulting in improved image classification. During the dictionary update and sparse coding stages in the proposed algorithm, we provide closed-form solutions and enforce the data locality constraint throughout the learning process. In contrast to previous dictionary learning approaches utilizing sparse representation techniques, which did not (or only partially) take data locality into consideration, our algorithm is able to produce a more representative dictionary and thus achieves better performance. We conduct experiments on databases designed for face and handwritten digit recognition. For such reconstruction-based classification problems, we will confirm that our proposed method results in better or comparable performance as state-of-the-art SRC methods do, while less training time for dictionary learning can be achieved.

1. Introduction

Sparse representation has recently been applied to a variety of applications in computer vision and image processing [1–3]. Its success is attributed to the fact that the dimensionality of signals such as natural images is often much lower than that which is observed, and thus it offers a more compact yet better description of natural signals for the above applications. To extend sparse representation to the problems of classification, Wright et al. [4] proposed sparse representation based classification (SRC). Based on image reconstruction, SRC assigns a test input to the class with the smallest reconstruction error using the corresponding sparse representation. SRC has been shown to produce impressive performance on face recognition and attracts more attention of the researchers, since both feature extraction (i.e., sparse representation) and classification can be achieved simultaneously without the need to train additional classifiers. Besides face recognition [4–6], SRC has also been applied to handwritten digit recognition [7–9].

In order to derive sparse representation for signals like images, one needs to utilize an over-complete dictionary for reconstruction purposes and deploys a sparsity constraint on the resulting weight coefficients. Depending on the applications of interest (e.g., image reconstruction or classification), one should carefully design the associated over-complete dictionary. For example, while a common approach for SRC is to directly apply the training images as the dictionary (e.g., [4,5]), recent research progress has shown that the learning of data and application-driven dictionary typically outperforms those using a predefined one [10,3]. Many efforts have been devoted to the learning of a proper dictionary for particular applications like image denoising [11,12], image inpainting [13,14], and image classification [15–17,7,9].

Data locality has been observed to be a key issue in the problems of clustering, dimension reduction [18,19], density estimation [20], anomaly detection [21], and image classification [22–25]. In pattern recognition, the kNN classifier can be considered as a recognition algorithm using data locality, since it considers the locality information of training data for performing classification. To be more precise, kNN assigns the class label for a test input based on the majority of the nearest training data of the same class. Motivated by the importance of data locality, we propose a novel dictionary learning approach for sparse representation based classification. Our method utilizes data locality for classification purposes and provides closed-form solutions for both dictionary update and sparse coding stages. Compared to standard SRC or recently proposed locality-constrained linear coding (LLC), our algorithm not only achieves improved classification performance (as supported by our experiments), but also offers fast convergence.

* Corresponding author. Tel.: 886-2-27872300x2368.
E-mail addresses: cpwei@citi.sinica.edu.tw (C.-P. Wei), ywchao@umich.edu (Y.-W. Chao), yryeh@citi.sinica.edu.tw (Y.-R. Yeh), ycwang@citi.sinica.edu.tw (Y.-C. Frank Wang).

0031-3203/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.patcog.2012.11.014

The remaining of this paper is organized as follows. Section 2 reviews related works on sparse representation, including its application for dictionary learning. In Section 3, we present our proposed algorithm for locality-sensitive dictionary learning for sparse representation based classification. Experimental results on real image data are presented in Section 4. Finally, Section 5 concludes the paper.

2. Related works

We first review the mathematical models for sparse representation using a predetermined dictionary. The formulation of sparse representation based classification (SRC) will be discussed, followed by the comparisons of several dictionary learning methods for SRC related classification algorithms.

2.1. Sparse representation and its variants

Given a signal $\mathbf{x} \in \mathbb{R}^{d\times 1}$ and an over-complete dictionary $\mathbf{D} \in \mathbb{R}^{d\times K}$ (in which $d \ll K$), sparse representation aims to express $\mathbf{x}$ as a compact linear combination of columns of $\mathbf{D}$

$$\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_0 \quad \text{s.t.} \quad \mathbf{x} = \mathbf{D}\boldsymbol{\alpha}, \tag{1}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^{K\times 1}$ is considered as the sparse representation of $\mathbf{x}$ in terms of $\mathbf{D}$, and the notation $\|\boldsymbol{\alpha}\|_0$ counts the number of non-zero entries in $\boldsymbol{\alpha}$. Solving (1) is known to be NP-hard and numerically unstable [26]. Some greedy algorithms [27,28] have been proposed to approximate the desired solution. Although these algorithms are simple and easy to implement, the approximated solutions are suboptimal [29]. Recent developments have led to the convex relaxation of (1) by replacing the nonconvex ℓ_0-norm with the convex ℓ_1-norm

$$\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_1 \quad \text{s.t.} \quad \mathbf{x} = \mathbf{D}\boldsymbol{\alpha}.$$

It is shown that if the solution $\boldsymbol{\alpha}$ of (1) is sufficiently sparse, one can solve the sparse representation via an ℓ_1-norm minimization problem [30,31]. To deal with the case that $\mathbf{x}$ may have small dense noise, the following related optimization problem is considered in the literature (known as Lasso [32]):

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

where the regularization parameter λ controls the sparsity of $\boldsymbol{\alpha}$. There is a massive list of algorithms using different techniques to solve (2) in recent literature. A review of these algorithms can be found in [33].

Recently, Wang et al. [23] proposed a coding scheme named locality-constrained linear coding (LLC), which replaces the sparsity regularization in (2) by a locality adaptor, i.e.,

$$\min_{\boldsymbol{\alpha}} \|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + \lambda \|\mathbf{p} \odot \boldsymbol{\alpha}\|_2^2$$
$$\text{s.t.} \quad \mathbf{1}^\top \boldsymbol{\alpha} = 1. \tag{3}$$

The symbol $\odot$ in (3) denotes element-wise multiplication, and $\mathbf{p}$ is the *locality adaptor* in which the kth entry calculates the distance between $\mathbf{x}$ and the kth column of $\mathbf{D}$. The purpose of LLC is to produce similar coding results $\boldsymbol{\alpha}$ when the input instances $\mathbf{x}$ are close to each other. It is worth noting that, the locality regularization term in (3) implies that the coefficient $\boldsymbol{\alpha}$ would be *sparse*, since this introduced regularization term penalizes the non-zero entries whose corresponding atoms are far away from the input signal $\mathbf{x}$. It has been shown in [23] that LLC is able to produce promising classification results if the resulting sparse coefficients are used as features for training and testing.

In [25], a graph regularized sparse coding (GraphSC) was presented, which imposed a graph Laplacian constraint on the standard Lasso formulation as follows:

$$\min_{\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_N} \sum_{i=1}^{N} \|\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i\|_2^2 + \lambda_1 \sum_{i=1}^{N}\sum_{j=1}^{N} W_{ij}\boldsymbol{\alpha}_j^\top \boldsymbol{\alpha}_i + \lambda_2 \sum_{i=1}^{N} \|\boldsymbol{\alpha}_i\|_1.$$

In the above equation, $\mathbf{x}_1, \ldots, \mathbf{x}_N$ are input data instances and W_{ij} is the weight of the edge formed by the instance pair $\mathbf{x}_i$ and $\mathbf{x}_j$. If $\mathbf{x}_i$ is among the k-nearest neighbors of $\mathbf{x}_j$, then $W_{ij} = 1$; otherwise, $W_{ij} = 0$. The Laplacian regularizer $\sum_{i=1}\sum_{j=1} W_{ij}\boldsymbol{\alpha}_j^\top \boldsymbol{\alpha}_i$ ensures that similar inputs would obtain similar coding results $\boldsymbol{\alpha}_i$. Unlike LLC, however, GraphSC *does not* take the locality between the input data and the dictionary into consideration. Therefore, it is possible for GraphSC to select dictionary atoms (training data) farther away from the input instance, which is not preferable for classification purposes. Moreover, GraphSC involves an ℓ_1-regularization term in its objective function, and thus it is computationally more expensive to solve the resulting optimization problems.

2.2. Sparse representation for classification

Sparse representation based classification (SRC) has been recently proposed by Wright et. al. [4] for robust face recognition. Since our proposed method is based on the framework of SRC, we now discuss this method for the sake of clarity. Suppose that there exist N training images from J object classes, and each class j has N_j images. Let $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_J] \in \mathbb{R}^{d\times N}$ be the training set, where $\mathbf{X}_j \in \mathbb{R}^{d\times N_j}$ contains training images of the jth class as its columns. All columns of $\mathbf{X}$ are normalized to have a unit ℓ_2-norm. Given a test image $\mathbf{y} \in \mathbb{R}^{d\times 1}$, the SRC algorithm classifies $\mathbf{y}$ using its sparse representation $\boldsymbol{\alpha}$, which is computed via the above ℓ_1-norm minimization process over the entire training image set. More precisely, SRC solves

$$\min_{\boldsymbol{\alpha}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\alpha}\|_2^2 + \lambda_{\mathrm{SRC}} \|\boldsymbol{\alpha}\|_1, \tag{4}$$

which is identical to (2) with dictionary $\mathbf{D}$ replaced by training set $\mathbf{X}$. Once (4) is solved, classification will be done based on the minimum class-wise reconstruction error. In other words, let $\boldsymbol{\alpha} = [\boldsymbol{\alpha}^1; \boldsymbol{\alpha}^2; \ldots; \boldsymbol{\alpha}^J]$ and $\boldsymbol{\alpha}^j \in \mathbb{R}^{N_j\times 1}$ be the entries of $\boldsymbol{\alpha}$ associated with class j. We recognize the test input $\mathbf{y}$ as class j^* according to the following rule:

$$j^* = \arg\min_{j} \|\mathbf{y} - \mathbf{X}_j\boldsymbol{\alpha}^j\|_2^2. \tag{5}$$

The underlying assumption behind SRC is that, if the test image $\mathbf{y}$ belongs to class j, it should be presented in the column space of $\mathbf{X}_j$. Therefore, the non-zero elements of $\boldsymbol{\alpha}$ will mainly be observed in $\boldsymbol{\alpha}^j$ and thus satisfy (5).

2.3. Dictionary learning with sparse representation

2.3.1. Dictionary learning for data reconstruction

In order to achieve an improved reconstruction or representation performance, prior work has applied (2) to learn an over-complete dictionary for sparse data representation. More precisely, let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N] \in \mathbb{R}^{d\times N}$ be the training set, where $\mathbf{x}_i$ is a data instance with dimension d. One can learn a dictionary $\mathbf{D} \in \mathbb{R}^{d\times K}$ by solving the following optimization problem:

$$\min_{\mathbf{D},\mathbf{A}} \|\mathbf{X} - \mathbf{D}\mathbf{A}\|_F^2 + \lambda_{\mathrm{DL}} \sum_{i=1}^{N} \|\boldsymbol{\alpha}_i\|_1$$
$$\text{s.t.} \|\mathbf{d}_k\|_2 \le 1 \quad \forall k = 1, \ldots, K, \tag{6}$$

where $\mathbf{d}_k \in \mathbb{R}^{d\times 1}$ is the kth column of $\mathbf{D}$, and $\boldsymbol{\alpha}_i \in \mathbb{R}^{K\times 1}$ is the ith column of $\mathbf{A}$, which denotes the sparse representation of $\mathbf{x}_i$. λ_{DL} is the regularization parameter balancing the reconstruction error and sparsity. Note that all atoms $\mathbf{d}_k$'s are required to have lengths less than or equal to 1. If removing this constraint, the encoded coefficients $\boldsymbol{\alpha}_i$ will approach to zero to favor the ℓ_1-norm sparse penalty, while the columns of $\mathbf{D}$ will tend to be large to keep the product $\mathbf{D}\boldsymbol{\alpha}_i$ unchanged [34].

The optimization problem in (6) is not convex in $\mathbf{D}$ and $\mathbf{A}$. Typically, one would iterate between two steps, sparse coding and dictionary update, to obtain the desired dictionary $\mathbf{D}$ and sparse coefficient matrix $\mathbf{A}$. In the stage of sparse coding, the matrix $\mathbf{A}$ can be determined by minimizing the objective function of (6) with a fixed $\mathbf{D}$. This procedure can be efficiently solved by numerous algorithms in literature [33]. For dictionary updating, the coefficient matrix $\mathbf{A}$ is fixed and one calculates $\mathbf{D}$ that solves (6), which is a least squares problem with quadratic constraints and can be solved by existing convex optimization algorithms. Another popular dictionary learning approach is K-SVD [10]. K-SVD utilizes the ℓ_0-norm constraint in (1) rather than the ℓ_1-norm as discussed earlier. Inspired by K-means clustering, the dictionary update step in K-SVD is performed by sequentially updating each dictionary atom to minimize the reconstruction error. The above dictionary learning methods simply aim at finding the best dictionary which is able to sparsely represent each input data instance. When one apply SRC or use the coefficient $\boldsymbol{\alpha}$ as features to train classifiers, there is no guarantee that a good separation between classes exists.

2.3.2. Dictionary learning for data classification

Generally, one can divide existing *supervised* dictionary learning approaches into three different categories. The first group of such algorithms simply applies the standard SRC framework, in which the dictionary in (4) is learned from each class of the training data, and the test input is encoded and classified accordingly. In [35], this dictionary learning approach was applied to face recognition, and the authors considered this as *metaface* learning. Mairal et al. [36] added an *softmax* cost function term into the sparse representation formulation for improved SRC. Ramirez et al. [7] introduced an incoherent term on dictionaries from different classes into the objective function, which encourages dictionaries associated with different classes to be as independent/uncorrelated as possible. In [9], the Fisher discrimination criterion was imposed on $\boldsymbol{\alpha}$ so that the learned dictionary favored data classification. To determine the class label of a test input, all the above methods apply the classification rule of SRC (i.e., (5)).

The second category of dictionary algorithms integrates the classification model (e.g., SVM or logistic function) into the sparse representation framework, and thus the classification model and the dictionary will be jointly learned during the training stage (e.g., [37,16,38,39]). To classify a test input, the methods of [37,16] are required to solve the joint optimization problem on both sparse representation and the introduced classification model.

Different from the above two types of approaches, the third category of dictionary learning algorithms imposes different discrimination constraints on the sparse coefficients in (4), and utilizes the sparse coefficients as features to train a classifier such as SVM for classification purposes. For example, Huang and Aviyente [40] added a discriminative term on $\boldsymbol{\alpha}$ based on Fisher's linear discriminant analysis as the additional constraint, and they used the sparse coefficients $\boldsymbol{\alpha}$ to train the SVM as the classifier. We note that, while the methods in the latter two categories of approaches report improved classification performance, one typically needs to solve complicated optimization problems, and learning of additional classifiers on the resulting sparse coefficients $\boldsymbol{\alpha}$ is required.

In this paper, we propose a novel dictionary learning approach using the framework of SRC. Our approach belongs to the first category and does not require the introduction of classification model or additional discrimination constraints on the sparse coefficients. We advocate the use of data locality for both reconstruction and classification purposes in our proposed framework. Since closed form solutions can be obtained in both dictionary update and sparse coding stages, our approach is easy to implement and exhibits excellent classification ability due to the exploitation of data locality.

3. Locality-sensitive dictionary learning for sparse representation based classification

We now detail our proposed algorithm in this section. In Section 3.1, we first discuss the difference between data sparsity and locality, and why advancing data locality is preferable for classification. Section 3.2 introduces our proposed locality-sensitive dictionary learning algorithm, including the discussions on our improvements over existing works in terms of representation/classification capabilities and convergence rates. The derivation of the closed-form solutions for our algorithm is presented in Section 3.3, and two classification rules for recognizing test inputs are discussed in Section 3.4.

3.1. Locality vs. sparsity

As highlighted in the introduction, data locality has been widely utilized in many pattern recognition problems such as dimension reduction [18,19], density estimation [20], anomaly detection [21], and classification [22–25]. While SRC has recently been used for addressing image classification problems (see Section 2.2), whether sparsity is sufficient or necessary for solving such tasks has recently be investigated [23,24,41].

It has been pointed out in [23] that enforcing the locality constraint in (3) would imply the sparsity for the resulting encoding coefficients, since only the dictionary atoms close to the test input would be selected for data reconstruction. On the other hand, the standard sparse representation formulation in (2) does not favor this choice. Since SRC is based on class-wise minimum reconstruction error, it is not preferable if one selects dictionary atoms far away from the test input for reconstruction. This is due to the fact that atoms farther away from the input data are less likely to be in the class which the test input belongs to (which is the underlying assumption of the kNN classifier). In other words, standard sparse representation does not preserve the information of data locality during its encoding process, while the idea of LLC is able to preserve such information and thus favors both data representation and classification. This motivates us to propose a locality-sensitive dictionary learning algorithm with performance and convergence guarantees.

Below we give an example to visualize the differences between dictionaries based on data sparsity and locality. We take face images from the Extended Yale B database (discussed later in experiments) for example. For a subject with 32 training images (each with 192×168 pixels), we perform PCA and project the data onto the first 300 eigenvectors. As a result, the data matrix $\mathbf{X} \in \mathbb{R}^{300\times32}$ can be obtained. Let $\mathbf{D}_{\mathrm{SRC}} \in \mathbb{R}^{300\times8}$ be the dictionary for SRC [4] in which the columns are randomly chosen from $\mathbf{X}$ (i.e., methods based on data sparsity without dictionary learning). In addition, we have $\mathbf{D}_{\mathrm{metaface}}$ and $\mathbf{D}_{\mathrm{LSRC}}$ as the learned dictionaries for metaface [35] (based on sparsity) and our proposed LSRC (as detailed in the next subsection), in which the size of the dictionary is also chosen to be eight. We plot the training data $\mathbf{X}$, $\mathbf{D}_{\mathrm{SRC}}$, $\mathbf{D}_{\mathrm{metaface}}$, $\mathbf{D}_{\mathrm{LSRC}}$ onto the 2D space via multidimensional scaling in Fig. 1. From Fig. 1, we see that $\mathbf{D}_{\mathrm{SRC}}$ is formed by randomly choosing eight instances from the data matrix $\mathbf{X}$, and thus the atoms of $\mathbf{D}_{\mathrm{SRC}}$ coincide with some atoms of $\mathbf{X}$. $\mathbf{D}_{\mathrm{metaface}}$ is

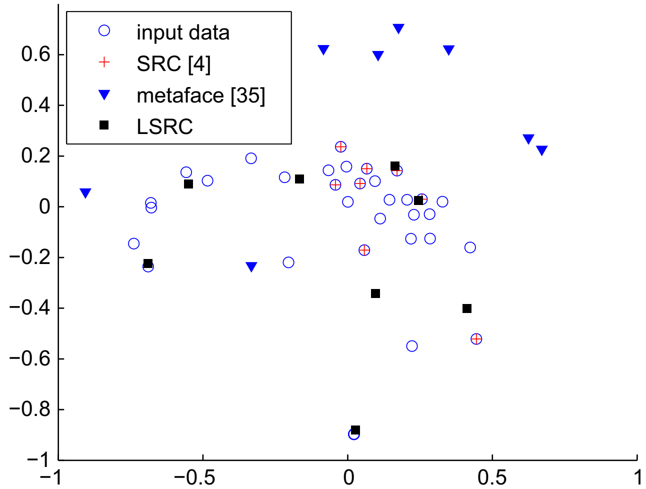

Fig. 1. Example face image data (from the Extended Yale B Database) and the associated dictionaries with SRC [4], metaface [35], and our LSRC in the 2D space via multidimensional scaling.

learned using the standard sparse representation formulation, and thus the eight learned dictionary atoms do not sufficiently encode data locality information (some of the learned atoms are even far away from the training instances). When properly integrating both data sparsity and locality (as our proposed method LSRC does), $\mathbf{D}_{\text{LSRC}}$ better describes the training data and thus improved classification performance can be expected.

3.2. Locality-sensitive dictionary learning

Our proposed locality-sensitive dictionary learning is formulated as follows:

$$\min_{\mathbf{D},\mathbf{A}} \|\mathbf{X}-\mathbf{DA}\|_F^2+\lambda_{\text{DL}}\sum_{i=1}^{N}\|\mathbf{p}_i\odot\boldsymbol{\alpha}_i\|_2^2$$
$$\text{s.t.}\mathbf{1}^\top\boldsymbol{\alpha}_i=1\forall i=1,\ldots,N, \quad (7)$$

where $\boldsymbol{\alpha}_i\in\mathbb{R}^{K\times 1}$ is the ith column of $\mathbf{A}$, the symbol $\odot$ denotes the element-wise multiplication, and $\mathbf{p}_i\in\mathbf{R}^{K\times 1}$ is the *locality adaptor* whose kth element is given by $p_{ik}=\text{dist}(\mathbf{x}_i,\mathbf{d}_k)$, where $\text{dist}(\cdot)$ is a distance function defining a distance between the two inputs. We will consider two types of distance functions in Sections 3.3.1 and 3.3.2, respectively. The shift-invariant constraint $\mathbf{1}^\top\boldsymbol{\alpha}_i=1$ enforces the coding results $\boldsymbol{\alpha}$ to remain the same even if the origin of the data coordinate system is shifted as proved in [22].

It is worth noting that a locality-constrained linear coding (LLC) algorithm was recently proposed in [23]. An online dictionary learning technique was presented in [23], which approximately solves (7) with the *norm-bounded* constraint on $\mathbf{D}$ (i.e., the dictionary atoms $\mathbf{d}_k$ are required to have a unit length in LLC). Our proposed dictionary algorithm is very different from LLC in two aspects, as we now discuss.

3.2.1. Removing the norm-bounded constraint on **D** for improved representation and classification

Followed by the standard sparse representation formulation in (6), the original LLC imposes the norm-bounded constraint $\|\mathbf{d}_k\|_2\le 1$ in its formulation. However, such a constraint is not required for locality-sensitive dictionary learning, as we now discuss. For locality-sensitive dictionary learning, the length of columns of $\mathbf{D}$ cannot be arbitrarily large due to the enforcement of the existing locality constraint $\|\mathbf{p}_i\odot\boldsymbol{\alpha}_i\|_2^2$ (recall that each entry of $\mathbf{p}_i$ measures the distance between $\mathbf{x}_i$ and the corresponding column of $\mathbf{D}$). In contrast to LLC, removing this additional norm-bounded constraint in our proposed formulation in (7) would produce a better optimal value for the objective function, since now fewer constraints are imposed on the proposed minimization problem. As a result, the benefits of dropping the norm-bounded constraint in our proposed formulation are twofold. First, we are able to obtain a dictionary $\mathbf{D}$ which better fits the local data structure and favors classification (as supported by our experiments later). Second, as we detail later, closed-form solutions can be derived for both dictionary update and sparse coding stages when solving (7), and thus faster convergence can be expected.

3.2.2. Discussions on the convergence rates

It is worth noting that, the original LLC algorithm (i.e., Algorithm 4.1 in [23]) does not directly minimize $\sum_{i=1}^{N}f(\mathbf{D},\mathbf{x}_i)$. Instead, it minimizes $f(\mathbf{D},\mathbf{x}_i)$ whenever a data instance $\mathbf{x}_i$ is drawn from $\mathbf{X}$, and solves the approximated version of the original minimization problem in an incremental setting (see [23] for details). Since the objective function of (7) can be written as $\min_{\mathbf{D},\mathbf{A}}\sum_{i=1}^{N}f(\mathbf{D},\mathbf{x}_i)$, where

$$f(\mathbf{D},\mathbf{x}_i)=\|\mathbf{x}_i-\mathbf{D}\boldsymbol{\alpha}_i\|_2^2+\lambda_{\text{DL}}\|\mathbf{p}_i\odot\boldsymbol{\alpha}_i\|_2^2. \quad (8)$$

We proposed to directly minimize $\sum_{i=1}^{N}f(\mathbf{D},\mathbf{x}_i)$ in a batch mode, in which closed-form solutions for updating the dictionary and deriving the sparse coefficient can be derived as we discuss in Section 3.3.

We now provide an example to illustrate the difference between our proposed method and LLC in dictionary learning in terms of solving the optimization problem and the associated convergence rates. We consider the data $\mathbf{X}$ as 1005 images from the first category of the USPS handwritten digit dataset [42] (as we use later in our experiments). Each image is of size 16 × 16 pixels. Thus, $\mathbf{X}$ is of size 256 × 1005). For both our proposed method and LLC, the goal is to learn a dictionary $\mathbf{D}$ of size 256 × 200 for $\mathbf{X}$. To make the comparisons more complete, we consider the standard LLC with the norm-bounded constraint, and the relaxed version of LLC without the norm-bounded constraint (denoted in LLC_{uc}). Fig. 2 shows the values of the objective function of (7) versus time for the three approaches. From this figure, it can be seen that our algorithm converges faster and obtains a smaller optimization value than the two LLC methods do. The large gap between the two LLC approaches indicates the effect of dropping the norm-bounded constraint. As expected, without the norm-bounded constraint, a smaller value for the objective function will be obtained, and thus

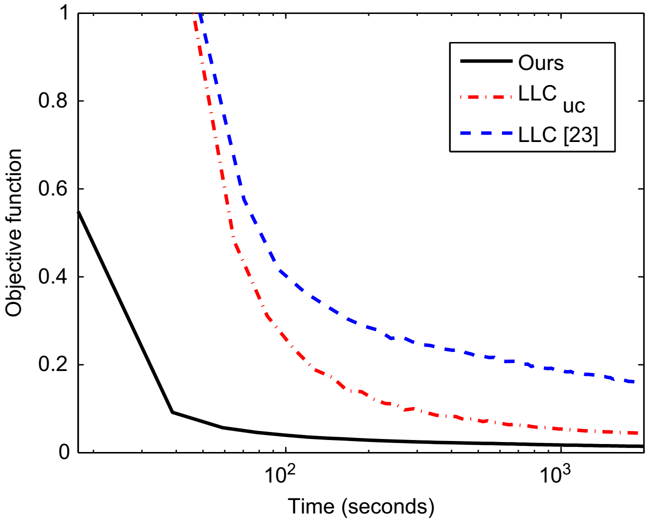

Fig. 2. Comparisons of our method and LLC [23] (with and without the norm-bounded constraint) in optimizing (7) using the USPS handwritten digit dataset. Note that the LLC without the norm-bounded constraint is denoted as LLC_{uc}.

the learned dictionary $\mathbf{D}$ better describes the data $\mathbf{X}$. Nevertheless, among these three methods, ours achieves the minimum optimization value and converges much faster than the other two. This example supports our claim that the norm-bounded constraints on $\mathbf{D}$ are not necessary for dictionary learning, if the data locality regularization is imposed on dictionary learning formulation. Later in our experiments, we will verify that our method achieves improved classification performance than the standard LLC does.

3.3. Algorithms

In our locality-sensitive dictionary learning algorithm, we consider two different locality adaptors to preserve the data structure for improved representation and classification. To define distance metric, the two locality adaptors use the ℓ_2-norm and the exponential function, respectively. It is worth repeating that, using our proposed formulation with either of these two locality adaptors, we are able to derive closed form solutions for both dictionary $\mathbf{D}$ and sparse coefficient vector $\boldsymbol{\alpha}_i$. A related work [43] also considered ℓ_2-norm locality adaptors for dictionary learning, but their locality adaptors are multiplied with the absolute value of $\boldsymbol{\alpha}$, and hence their results still require to solve ℓ_1-minimization problems. We now provide detailed derivations and discussions using the two locality adaptors.

3.3.1. ℓ_2-norm locality adaptor

Let the entries of the locality adaptor $\mathbf{p}_i$ in (7) be

$$p_{ik} = \|\mathbf{x}_i - \mathbf{d}_k\|_2,$$

which measures the Euclidean distance between the input instance $\mathbf{x}_i$ and the kth dictionary atom. We also apply an iterative procedure to update the dictionary and the encoded sparse vector $\boldsymbol{\alpha}_i$. In the sparse coding step, $\mathbf{D}$ is fixed in (7) and thus $\boldsymbol{\alpha}_i$ will be the solution of the following optimization problem

$$\min_{\boldsymbol{\alpha}_i} \|\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i\|_2^2 + \lambda_{\mathrm{DL}} \|\mathbf{p}_i \odot \boldsymbol{\alpha}_i\|_2^2$$
$$\text{s.t.} \mathbf{1}^\top \boldsymbol{\alpha}_i = 1, \tag{9}$$

for $i = 1,2,\ldots,N$. To determine the solution $\boldsymbol{\alpha}_i$ for (9), we consider the Lagrange function $L(\boldsymbol{\alpha}_i,\eta)$, which is defined as

$$\|\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i\|_2^2 + \lambda_{\mathrm{DL}} \|\mathbf{p}_i \odot \boldsymbol{\alpha}_i\|_2^2 + \eta(\mathbf{1}^\top \boldsymbol{\alpha}_i - 1),$$

which can be re-formulated as

$$\boldsymbol{\alpha}_i^\top \mathbf{C} \boldsymbol{\alpha}_i + \lambda_{\mathrm{DL}} \boldsymbol{\alpha}_i^\top \mathrm{diag}(\mathbf{p}_i)^2 \boldsymbol{\alpha}_i + \eta(\mathbf{1}^\top \boldsymbol{\alpha}_i - 1),$$

where $\mathbf{C} = (\mathbf{x}_i \mathbf{1}^\top - \mathbf{D})^\top (\mathbf{x}_i \mathbf{1}^\top - \mathbf{D})$, and $\mathrm{diag}(\mathbf{p}_i)$ is a diagonal matrix whose nonzero elements are the entries of $\mathbf{p}_i$. Let $\partial L(\boldsymbol{\alpha}_i,\eta) / \partial \boldsymbol{\alpha}_i = 0$, we have

$$\boldsymbol{\Phi} \boldsymbol{\alpha}_i + \eta \mathbf{1} = 0, \tag{10}$$

where $\boldsymbol{\Phi} \coloneqq 2(\mathbf{C} + \lambda_{\mathrm{DL}}\, \mathrm{diag}(\mathbf{p}_i)^2)$. Once we pre-multiply (10) by $1^\top \boldsymbol{\Phi}^{-1}$, we obtain $\eta = -(\mathbf{1}^\top \boldsymbol{\Phi}^{-1} \mathbf{1})^{-1}$. Substituting η into (10) gives the analytical solution of (9) as

$$\tilde{\boldsymbol{\alpha}}_i = (\mathbf{C} + \lambda_{\mathrm{DL}}\, \mathrm{diag}(\mathbf{p}_i)^2)^{-1} \mathbf{1}$$
$$\boldsymbol{\alpha}_i = \tilde{\boldsymbol{\alpha}}_i / (\mathbf{1}^\top \tilde{\boldsymbol{\alpha}}_i). \tag{11}$$

Hence, it is not necessary to solve an ℓ_1-norm minimization problem like (2).

On the other hand, the dictionary update stage needs to solve

$$\min_{\mathbf{D}} \|\mathbf{X} - \mathbf{D}\mathbf{A}\|_F^2 + \lambda_{\mathrm{DL}} \sum_{i=1}^{N} \|\mathbf{p}_i \odot \boldsymbol{\alpha}_i\|_2^2. \tag{12}$$

Let the objective function of (12) be denoted by $F(\mathbf{D})$. To derive the analytical solution of (12), we take the partial derivatives of $F(\mathbf{D})$ with respect to $\mathbf{d}_k$ for $k \in \{1,2,\ldots,K\}$, which gives

$$\frac{\partial F}{\partial \mathbf{d}_k} = \sum_{i=1}^{N} -2\alpha_{ik}(\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i) - 2\lambda \alpha_{ik}^2 (\mathbf{x}_i - \mathbf{d}_k),$$

or equivalently,

$$\left(\frac{\partial F}{\partial \mathbf{d}_k}\right)^\top = \sum_{i=1}^{N} (-2\alpha_{ik}(1+\lambda\alpha_{ik})(\mathbf{x}_i)^\top + 2\left(\lambda \alpha_{ik}^2 \mathbf{d}_k^\top + \alpha_{ik} \sum_{j=1}^{K} \alpha_{ij} \mathbf{d}_j^\top\right)), \tag{13}$$

where we omit the subscript DL from λ for simplicity. Since (12) is now an unconstrained convex optimization problem (as will be shown later), its global minimum can be easily calculated at the point whose partial derivatives of $F(\mathbf{D})$ are zeroes. By setting the partial derivatives of (13) equal to zero for $k = 1,2,\ldots,K$, we have

$$\mathbf{U}\mathbf{D}^\top = \mathbf{V}, \tag{14}$$

where the matrices $\mathbf{U} \in \mathbb{R}^{K \times K}$ and $\mathbf{V} \in \mathbb{R}^{K \times d}$ are

$$\mathbf{U} = \sum_{i=1}^{N} \begin{pmatrix} (1+\lambda)\alpha_{i1}^2 & \alpha_{i1}\alpha_{i2} & \cdots & \alpha_{i1}\alpha_{iK} \\ \alpha_{i1}\alpha_{i2} & (1+\lambda)\alpha_{i2}^2 & \cdots & \alpha_{i2}\alpha_{iK} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{i1}\alpha_{iK} & \alpha_{i2}\alpha_{iK} & \cdots & (1+\lambda)\alpha_{iK}^2 \end{pmatrix},$$
$$\mathbf{V} = \sum_{i=1}^{N} \begin{pmatrix} \alpha_{i1}(1+\lambda\alpha_{i1})(\mathbf{x}_i)^\top \\ \alpha_{i2}(1+\lambda\alpha_{i2})(\mathbf{x}_i)^\top \\ \vdots \\ \alpha_{iK}(1+\lambda\alpha_{iK})(\mathbf{x}_i)^\top \end{pmatrix}. \tag{15}$$

We then obtain the solution $\mathbf{D}$ by solving the linear system (14).

In our work, we alternate between the two steps, sparse coding and dictionary update, for obtaining the optimal solutions $\mathbf{A}$ and $\mathbf{D}$. To address the convergence issue, we note that such an iterative procedure is known as *block coordinate descent* or *non-linear Gauss-Seidel methods* [44]. It has been proved by Proposition 2.7.1 in [44] that if solutions of $\mathbf{A}$ and $\mathbf{D}$ are *unique*, the sequences of $\mathbf{A}$ and $\mathbf{D}$ calculated by such an iterative procedure will converge to stationary points. As a result, the convergence of our algorithm would be guaranteed.

To verify that the solutions of $\mathbf{A}$ and $\mathbf{D}$ are unique for the use of ℓ_2-norm locality adaptors, and we refer to Eqs. (9)–(15) for discussions. In the sparse coding stage (when $\mathbf{D}$ is fixed), the analytical solution of $\mathbf{A}$ exists and is unique as derived in (11). As for the dictionary update stage of (12), we have the closed-form solution for $\mathbf{D}$ using (14), in which matrix $\mathbf{U}$ can be shown to be *positive definite* (see Appendix A). As a result, the uniqueness of the solution $\mathbf{D}$ can be verified. To confirm that the solution $\mathbf{D}$ is also unique for the later use of exponential locality adaptors, we use Appendix A to show that (12) is a strictly convex optimization problem (i.e., the Hessian matrix $\mathbf{U}$ of the objective function $F(\mathbf{D})$ is also positive definite even with the use of exponential locality adaptors). Therefore, the strict convexity of (12) can be assured, and thus the existence of the unique solution $\mathbf{D}$ can be guaranteed.

3.3.2. Exponential locality adaptor

We now consider the exponential locality adaptor $\mathbf{p}_i$, in which each entry is

$$p_{ik} = \sqrt{\exp\left(\frac{\|\mathbf{x}_i - \mathbf{d}_k\|_2^2}{\sigma}\right)}, \tag{16}$$

and σ is a positive number. Since p_{ik}^2 grows exponentially with $\|\mathbf{x}_i - \mathbf{d}_k\|^2/\sigma$, the exponential locality adaptor gives very large p_{ik} when $\mathbf{x}_i$ and $\mathbf{d}_k$ are far apart. This property is useful when we want to stress the importance of data locality. (Since p_{ik} is the weight of the sparse coefficient α_{ik} in (7), a large value of p_{ik} causes α_{ik} to be small.) We note that our exponential locality adaptor is slightly different from that in the original LLC ($p_{ik} = \exp(\|\mathbf{x}_i - \mathbf{d}_k\|_2/\sigma)$), since our exponential locality adaptor allows us to derive analytical solutions in our dictionary update stage.

Similar to our derivations in the case of ℓ_2-norm locality adaptor, we calculate the partial derivatives of $F(\mathbf{D})$ with respect to the columns of $\mathbf{D}$:

$$\frac{\partial F}{\partial \mathbf{d}_k} = \sum_{i=1}^{N} -2\alpha_{ik}(\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i) - 2\frac{\lambda\, p_{ik}^2}{\sigma}\alpha_{ik}^2(\mathbf{x}_i - \mathbf{d}_k)$$

or equivalently,

$$\left(\frac{\partial F}{\partial \mathbf{d}_k}\right)^{\top} = \sum_{i=1}^{N}\left(-2\alpha_{ik}\left(1 + \frac{\lambda p_{ik}^2}{\sigma}\alpha_{ik}\right)(\mathbf{x}_i)^{\top} + 2\left(\frac{\lambda p_{ik}^2}{\sigma}\alpha_{ik}^2\mathbf{d}_k^{\top} + \alpha_{ik}\sum_{j=1}^{K}\alpha_{ij}\mathbf{d}_j^{\top}\right)\right),$$

where $k \in \{1,2,\ldots,K\}$. Setting the partial derivatives of $F(\mathbf{D})$ equal to zero gives $g(\mathbf{D}) \coloneqq \mathbf{DU} - \mathbf{V}^{\top} = 0$, where

$$\mathbf{U} = \sum_{i=1}^{N}\begin{pmatrix} (1+\lambda_{i1})\alpha_{i1}^2 & \alpha_{i1}\alpha_{i2} & \cdots & \alpha_{i1}\alpha_{iK} \\ \alpha_{i1}\alpha_{i2} & (1+\lambda_{i2})\alpha_{i2}^2 & \cdots & \alpha_{i2}\alpha_{iK} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{i1}\alpha_{iK} & \alpha_{i2}\alpha_{iK} & \cdots & (1+\lambda_{iK})\alpha_{iK}^2 \end{pmatrix},$$

$$\mathbf{V} = \sum_{i=1}^{N}\begin{pmatrix} \alpha_{i1}(1+\lambda_{i1}\alpha_{i1})(\mathbf{x}_i)^{\top} \\ \alpha_{i2}(1+\lambda_{i2}\alpha_{i2})(\mathbf{x}_i)^{\top} \\ \vdots \\ \alpha_{iK}(1+\lambda_{iK}\alpha_{iK})(\mathbf{x}_i)^{\top} \end{pmatrix} \qquad (17)$$

with $\lambda_{ik} = \lambda\, p_{ik}^2/\sigma$ for $k = 1,2,\ldots,K$. Note that $\mathbf{U}$ and $\mathbf{V}$ defined in (18) include p_{ik}, and thus they are functions of $\mathbf{d}_k$ (or $\mathbf{D}$). (Below, $\mathbf{U}$ and $\mathbf{V}$ are denoted as $\mathbf{U_D}$ and $\mathbf{V_D}$, respectively). Unlike the case of ℓ_2-norm locality adaptor, $g(\mathbf{D}) = 0$ is now a nonlinear equation. We use Newton's method to search for the solution to $g(\mathbf{D}) = 0$, i.e.,

$$\mathbf{D}_{p+1} = \mathbf{D}_p - g(\mathbf{D}_p)(g'(\mathbf{D}_p))^{-1},$$

where $\mathbf{D}_{p+1}$ is the dictionary at $p+1$ iteration. It can be verified that $g'(\mathbf{D}_p) = \mathbf{U}_{\mathbf{D}_p}$, and thus

$$\mathbf{D}_{p+1} = \mathbf{D}_p - \left(\mathbf{D}_p\mathbf{U}_{\mathbf{D}_p} - \mathbf{V}_{\mathbf{D}_p}^{\top}\right)\mathbf{U}_{\mathbf{D}_p}^{-1} = \mathbf{V}_{\mathbf{D}_p}^{\top}\mathbf{U}_{\mathbf{D}_p}^{-1}.$$

Similarly, to calculate $\mathbf{A}$ and $\mathbf{D}$, we alternate between sparse coding and dictionary update stages and use (11) and the above equation for deriving the optimal solutions.

Since our dictionary learning algorithm will be applied to address classification problems in terms of SRC, we apply the proposed formulation and learn distinct dictionaries for each class (rather than one dictionary for all classes). As a result, we will be solving the following class-wise optimization problem

$$\min_{\mathbf{D}_j,\mathbf{A}_j}\|\mathbf{X}_j - \mathbf{D}_j\mathbf{A}_j\|_F^2 + \lambda_{\mathrm{DL}}\sum_{i=1}^{N_j}\|\mathbf{p}_{ji}\odot\boldsymbol{\alpha}_{ji}\|_2^2$$
$$\text{s.t.}\,\mathbf{1}^{\top}\boldsymbol{\alpha}_{ji} = 1 \quad \forall i = 1,\ldots,N_j \text{ and } j = 1,\ldots,J. \qquad (18)$$

In the above equation, J is the number of classes, and $\mathbf{x}_{ji}$ is the ith column of $\mathbf{X}_j$. The vector $\boldsymbol{\alpha}_{ji}$ denotes the ith column of $\mathbf{A}_j$, and $\mathbf{p}_{ji}$ is the locality adaptor for distance between $\mathbf{x}_{ji}$ and $\mathbf{D}_j$. Note that (18) is identical to (7) except that $\mathbf{X}$, $\mathbf{D}$, and $\mathbf{A}$ are replaced by $\mathbf{X}_j$, $\mathbf{D}_j$, and $\mathbf{A}_j$, respectively. Therefore, we can solve (18) in the same iterative manner from (9) to (17) and learn the dictionary and sparse coefficients for each class for classification purposes.

Algorithm 1. Locality-Sensitive Dictionary Learning

```
1: Input: Arrange $N_j$ training samples from the $j$th class as columns of a matrix $\mathbf{X}_j \in \mathbb{R}^{d\times N_j}$ for $j = 1,2,\ldots,J$, and denote $\mathbf{X}_j = [\mathbf{x}_{j1},\mathbf{x}_{j2},\ldots,\mathbf{x}_{jN_j}]$.
2: for $j = 1$ to $J$ do
3:   Initialize $\mathbf{D}_j \in \mathbb{R}^{d\times K_j}$.
4:   while the stopping criterion is violated
5:     Sparse Coding: Use (11) to solve (18) with $\mathbf{D}_j$, $\mathbf{x}_{ji}$, $\boldsymbol{\alpha}_{ji}$, respectively, for $i = 1,2,\ldots,N_j$, where $\boldsymbol{\alpha}_{ji}$ is the $i$th column of $\mathbf{A}_j$.
6:     Dictionary Update: Solve (18) with $\mathbf{A}_j$ fixed. The analytical solution is given by $\mathbf{D}_j = (\mathbf{U}_j^{-1}\mathbf{V}_j)^{\top}$, where $\mathbf{U}_j$ and $\mathbf{V}_j$ are defined as in (15) for the $\ell_2$-norm adaptor or in (17) for the exponential adaptor, in which $\mathbf{x}_i$ and $\alpha_{ik}$'s are replaced by $\mathbf{x}_{ji}$ and the entries of $\boldsymbol{\alpha}_{ji}$, respectively.
7:   end while
8: end for
9: Output: $\mathbf{D}_1,\mathbf{D}_2,\ldots,\mathbf{D}_J$.
```

We illustrate our proposed locality-sensitive dictionary learning method with SRC in Algorithm 1. From this algorithm, we note that the while loop would terminate if the value of the objective function in (18) is no longer decreasing (or decreasing too slowly), or the maximum number of iterations is achieved. Note that in Step 6 of Algorithm 1, the entry λ_{ik} in (17) has the term p_{ik} defined as in (16), which requires to utilize the dictionary atom $\mathbf{d}_k$ determined in the previous iteration. To initialize $\mathbf{D}_j$ in Step 3, one can randomly sample the columns of $\mathbf{X}_j$, or can use kmeans clustering to partition the columns of $\mathbf{X}_j$ into K_j sets, and $\mathbf{D}_j$ is formed by the mean of each set.

3.4. Groupwise SRC with locality-sensitive dictionaries

Using Algorithm 1, we are able to learn locality-sensitive dictionaries for each class using training data. Now let $\mathbf{D} = [\mathbf{D}_1,\mathbf{D}_2,\ldots,\mathbf{D}_J]$, where $\mathbf{D}_j$ is the dictionary learned for class j, the SRC algorithm introduced in Section 2.2 can be applied to address classification problems. However, since our sparse representation $\boldsymbol{\alpha}$ is calculated via (18) which does not include the ℓ_1-norm constraint, we modify the original SRC algorithm and replace the ℓ_1-norm constraint in (4) by our locality constraint. As a result, the modified SRC algorithm satisfies

$$\min_{\boldsymbol{\alpha}}\|\mathbf{y} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + \lambda_{\mathrm{SRC}}\|\mathbf{p}\odot\boldsymbol{\alpha}\|_2^2, \qquad (19)$$

where $\mathbf{p}$ is either the ℓ_2-norm or the exponential locality adaptor depending on which one is used in training, and the coefficient vector $\boldsymbol{\alpha}$ for the test input $\mathbf{y}$ can be easily calculated by (11). Once the vector $\boldsymbol{\alpha}$ is computed, we classify the test input by the minimum class-wise reconstruction error (i.e., (5)). We call this classification rule as *locality-sensitive SRC* (LSRC).

We now summarize our LSRC classification rule as follows. First, we compute the sparse representation $\boldsymbol{\alpha}$ of the test input $\mathbf{y}$ using the collection of all dictionaries $\mathbf{D}$ according to (19). Next, $\boldsymbol{\alpha}$ is partitioned into $[\boldsymbol{\alpha}^1;\boldsymbol{\alpha}^2;\ldots;\boldsymbol{\alpha}^J]$, where $\boldsymbol{\alpha}^j$ is the sparse representation vector for class j. We then calculate the reconstruction error for each class, i.e., $\|\mathbf{y} - \mathbf{D}_j\boldsymbol{\alpha}^j\|_2$, and we assign the test input to the class with the minimum reconstruction error.

In addition to LSRC, we propose an alternative and more efficient way to classify the test input $\mathbf{y}$, suitable for the case where the dictionary size is large. Different from the above LSRC classification rule, we now compute the sparse representation of the test input $\mathbf{y}$ for class j by solving

$$\min_{\boldsymbol{\alpha}^j}\|\mathbf{y}-\mathbf{D}_j\boldsymbol{\alpha}^j\|_2^2+\lambda_{\mathrm{SRC}}\|\mathbf{p}_j\odot\boldsymbol{\alpha}^j\|_2^2, \qquad (20)$$

where $\mathbf{D}_j$ is the locality-sensitive dictionary learned for class j (as discussed in Section 3.3). Once we obtain $\boldsymbol{\alpha}^j$, the decision rule is the same as that of the LSRC. We referred to this classification rule as groupwise LSRC (or G-LSRC). Motivated by (18), G-LSRC aims at encoding the test input using only one dictionary $\mathbf{D}_j$, and the term *group* implies the class of interest. The algorithm of our G-LSRC is summarized in Algorithm 2.

It is worth noting that, while the only difference between LSRC and G-LSRC is the way to calculate $\boldsymbol{\alpha}^j$, the encoding of $\boldsymbol{\alpha}^j$ using G-LSRC is significantly computationally less expensive than that using LSRC due to the smaller size of the dictionary. This is because that the determination of the analytical solution of (19) requires to solve a linear equation whose system matrix is $K\times K$ (K is the number of columns in $\mathbf{D}$). When K is large, solving (19) becomes very difficult. Since the size of $\mathbf{D}_j$ is only a fraction of K, it is much easier to obtain $\boldsymbol{\alpha}^j$ in (20) for $j=1,2\ldots,J$ than to solve (19) for the entire $\boldsymbol{\alpha}$.

Algorithm 2. Groupwise Locality-Sensitive SRC

```
Input: Let $\mathbf{D}_j \in \mathbb{R}^{d\times K_j}$ be the dictionary for object
    $j\in\{1,2,\ldots,J\}$ and $\mathbf{y}\in\mathbb{R}^d$ be a test sample.
for $j=1$ to $J$ do
    Compute the locality adaptor $\mathbf{p}_j$ with entries:
    $p_{jk}=\begin{cases}\|\mathbf{y}-\mathbf{d}_{jk}\|_2 & (\ell_2\text{−norm})\\ \sqrt{\exp\left(\|\mathbf{y}-\mathbf{d}_{jk}\|_2^2/\sigma\right)} & (\text{exponent})\end{cases}$
    for $k=1,2,\ldots,K_j$, where $\mathbf{d}_{jk}$ is the $k$th column of $\mathbf{D}_j$, and $\sigma$ is
    a positive number.
    Solve the optimization problem (20) by the analytical
    solution:
    $\boldsymbol{\beta}^j=(\mathbf{C}_j+\lambda_{\mathrm{SRC}}\,\mathrm{diag}(\mathbf{p}_j)^2)^{-1}\mathbf{1}$
    $\boldsymbol{\alpha}^j=\boldsymbol{\beta}^j/(\mathbf{1}^\top\boldsymbol{\beta}^j)$,
    where $\mathbf{C}_j=(\mathbf{y}\mathbf{1}^\top-\mathbf{D}_j)^\top(\mathbf{y}\mathbf{1}^\top-\mathbf{D}_j)$, and $\lambda_{\mathrm{SRC}}$ is a parameter
    that controls the sparsity of $\boldsymbol{\alpha}^j$.
    Compute the residuals $r_j(\mathbf{y})=\|\mathbf{y}-\mathbf{D}_j\boldsymbol{\alpha}^j\|_2$.
end for
Output: identity($\mathbf{y}$)=arg min$_j r_j(\mathbf{y})$.
```

4. Experiments

Since SRC performs classification (e.g., face recognition) based on the data reconstruction error, we conduct experiments on face or digit image datasets to verify the effectiveness of our proposed method. Related classification works based SRC also apply the same strategy and address similar classification problems [8,9]. We note that we do not consider image datasets like Caltech-101 [45] or PASCAL VOC 2007 [46] that require the selection of higher-level features and the design of the associated classifiers in this paper, since it is clear that classification rules based on image reconstruction cannot be applied to such datasets.

4.1. Face recognition

For face recognition, we conduct experiments on the ORL, AR, and Extended Yale B databases. Our proposed approach considers two different locality adaptors discussed in Section 3.3, which are denoted as LSRC-L2 and LSRC-EXP, where L2 and EXP indicate the ℓ_2-norm and the exponential locality adaptors, respectively. We compare our approaches with the standard SRC [4], metaface learning [35], and the standard LLC algorithm [23]. For SRC and metaface learning, we apply the Homotopy method developed by [47] to solve the ℓ_1 minimization problems, since it is known to be accurate and efficient among various ℓ_1 minimization techniques as reported in [48]. The classification rules of LSRC-L2, LSRC-EXP, and LLC are the same,[1] i.e., the LSRC algorithm presented in Section 3.4. Metaface learning uses the same classification rule as the SRC does, while the SRC does not learn the dictionary and can be considered as a baseline approach for comparisons.

For face recognition, we choose Eigenfaces as the features for training and testing, and its dimension will depend on the size of the training set for each database. We consider the learning of different numbers q of dictionary atoms in our experiments, and randomly sample q instances as the dictionary for training when performing the experiments. We note that when q equals the size of the training set, the optimal solution for (7) can be derived explicitly, i.e., $\mathbf{D}=\mathbf{X}$ and $\mathbf{A}=\mathbf{I}$, which gives $\mathbf{p}_i=0$. In other words, both the reconstruction error $\|\mathbf{X}-\mathbf{D}\mathbf{A}\|_F$ and the locality penalty $\|\mathbf{p}_i\odot\boldsymbol{\alpha}_i\|_2$ will be zeroes, meaning that the learned dictionary will simply be the entire training data set itself. Thus, for the above scenario, metaface learning, LLC, and both of our LSRC-L2 and LSRC-EXP will use the entire training set as the learned dictionary. As for testing, both LLC and our LSRC-EXP will result in the same coding (and the same recognition performance) due to the use of the same exponential locality adaptor.

4.1.1. Classification performance

ORL Database The ORL database [49] contains 400 face images of 40 people, each of size 112 × 92 pixels (see Fig. 3 for example). The images were taken under different lighting conditions, facial expressions (open or closed eyes, smiling or not smiling) and facial details (glasses or not). We randomly and equally split the data into training and test sets (i.e., five images for each set), and consider the size of the Eigenface as 100. We perform 10 random trials, and report the average recognition rates of various methods in Fig. 4. It is worth noting that the recognition results of different dictionary learning methods converge if the number of dictionary atoms to be learned per class equals the number of training images available for each. This is because the direct use of the entire training set will be a trivial and optimal solution for dictionary learning. The same remarks can also be applied to the following two datasets.

From Fig. 4, it is clear that the performance of SRC degraded dramatically as the dictionary size decreases, since SRC does not have the ability to learn dictionaries from the training data. Our LSRC-EXP achieved the best result for all dictionary sizes. We also observe that the recognition performance of our LSRC-EXP and LSRC-L2 did not significantly decrease when a smaller dictionary size was of use, while those for other approaches were much more sensitive to the size of the dictionary.

We note that some recent works on the ORL database have reported a 100% recognition rate [50,51]. These works typically focus on the extraction/selection of local features (e.g., Gabor wavelet, Curvelet, or Local Binary Pattern) and the design of the corresponding algorithms for classification, which are beyond the scope of this paper. The experiments presented in this section are meant to validate the effectiveness of our algorithm for both dictionary learning and classification, while there is no need to design additional classifiers. Therefore, we do not explicitly compare our approach to those discussed in [50,51].

AR Database The AR database [52] contains over 4000 frontal images for 126 individuals. There are 26 face images available for

[1] Note that LSRC-EXP and LLC employ the exponential locality adaptor, while LSRC-L2 uses the ℓ_2-norm locality adaptor.

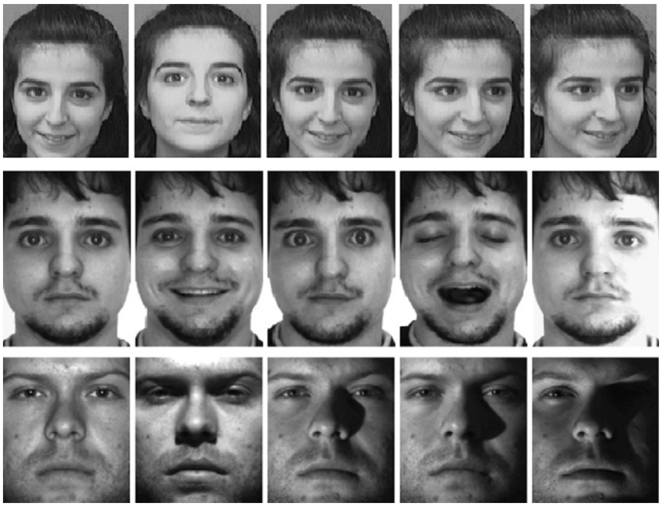

Fig. 3. Example images from the ORL (first row), AR (second row), extended Yale B (third row) databases.

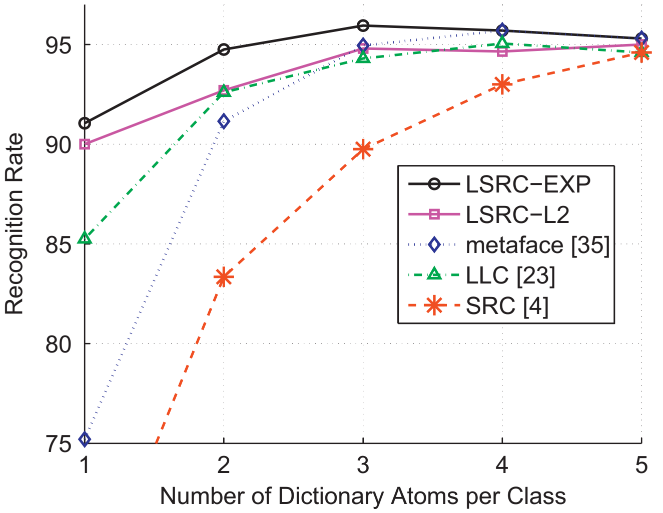

Fig. 4. Recognition performance of different SRC based methods with different numbers of atoms selected per class for the ORL database. Note that recognition results of different dictionary learning methods converge if the number of dictionary atoms to be learned per class equals the number of training images available for each.

each person, and the images are taken under different variations, including illumination, expression, and facial occlusion/disguise in two separate sessions. The images are cropped to 165 × 120 pixels and converted to gray scale (see Fig. 3 for example). We choose a subset of the dataset consisting of 50 male and 50 female subjects. For each subject, 14 images with only illumination and expression variations were selected (seven images from Session 1 and seven images from Session 2). The AR database is more challenging than the ORL database, since AR has more subjects, and the lighting conditions of AR vary more sharply.

In our experiments, we randomly split the data into a training set with 10 images and a test set with 4 images. Similar to the setting of the ORL database, we perform 10 random trials, and report the average recognition rates of various methods in Fig. 5, in which the dimension of the Eigenface is 300. From Fig. 5, it can be observed that our LSRC-EXP outperformed other sparse representation based approaches, while the performance of our method was less sensitive to the variations of the dictionary size.

Extended Yale B Database The Extended Yale B database [53] consists of 2414 frontal-face images of 38 subjects (about 64 images for each person). The cropped and normalized 192 × 168 face images were captured under various laboratory-controlled lighting conditions [54] (see Fig. 3 for example). Compared to the previous two databases, each subject in the Extended Yale B database has more images available, while only illumination variations are presented in the face images (no expression or pose variations). For each subject, we randomly select 32 images for training and the rest for testing. We again perform 10 random trials, and report the average recognition rates of various methods in Fig. 6 (the dimension of the Eigenfaces is set as 300). From previous two experiments, we already observed that our proposed LSRC algorithm with an exponential locality adaptor is more preferable, we only conduct experiments with LSRC-EXP on this database. From Fig. 6, we can see that LSRC-EXP, LLC, and metaface learning achieved similar performance, while our LSRC-EXP still performed slightly better (and the standard SRC without dictionary learning achieved the lowest recognition rates).

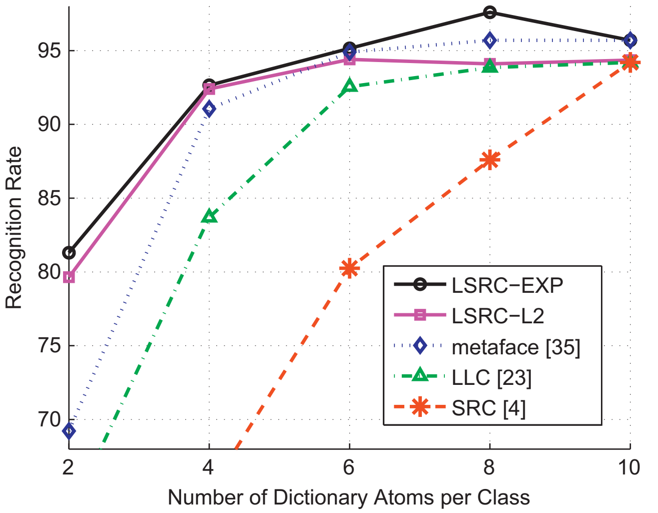

Fig. 5. Recognition performance of different SRC based methods with different numbers of atoms selected per class for the AR database.

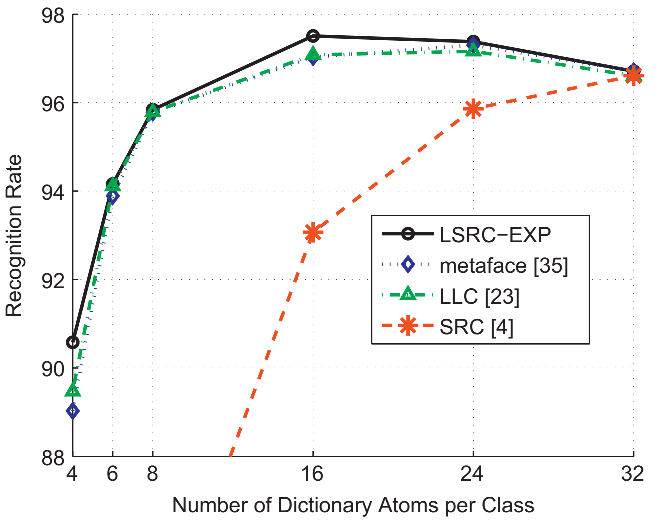

Fig. 6. Recognition performance of different SRC based methods with different numbers of atoms selected per class for the Extended Yale B database.

4.1.2. Computational time

We now present the averaged training time for dictionary learning and the testing time for classifying a image, and compare

Table 1
Computation time in seconds for training a dictionary with ORL, AR, and Extended Yale B.

Method/Database	ORL	AR	Extended Yale B
LSRC-L2	0.1096	0.2356	0.6585
LSRC-EXP	0.2508	0.5065	1.0764
LLC [23]	0.2402	0.4852	1.4980
Metaface [35]	0.9321	3.1018	39.9794

Table 2
Computation time in seconds for classifying a test input with ORL, AR, and Extended Yale B.

Method/Database	ORL	AR	Extended Yale B
LSRC-L2	0.0069	0.1637	0.0384
LSRC-EXP	0.0066	0.1631	0.0375
LLC [23]	0.0069	0.1635	0.0375
SRC [4], Metaface [35]	0.0870	0.5076	0.2109

the computation time of different approaches. In Table 1, we report the training time for learning 3, 6, and 8 dictionary atoms for each class of ORL, AR, and Extended Yale B, respectively. As discussed earlier, the feature dimension for each database is 100, 300, and 300, respectively. The corresponding testing time is reported in Table 2.

We note that the approaches of LLC, metaface, and SRC are implemented by ourselves with Matlab, except that we use the same exponential locality adaptor in (18) for both LLC and our proposed LGSR-EXP for fair comparisons. For best performance, the parameter λ in (2) is set to 0.001 for all databases for SRC and metaface learning. For our LSRC-L2, the parameter λ_{DL} in (18) is set to 0.1 for ORL and AR databases, and is set to 0.3 for the Extended Yale B database. For LSRC-EXP and LLC, we use the same parameter setting; the parameter λ_{DL} in (18) is set to 0.001 for all databases, and the parameter σ in (18) is set to 0.3 for ORL and AR, and is set to 0.5 for Extended Yale B. The parameter λ_{SRC} in (19) for classification has the same value as λ_{DL}. The runtime estimates reported in this paper were obtained on an Intel Quad Core PC with 2.33 GHz processors and 4 GB RAM.

From Table 1, we see that our approach is computationally more efficient than the standard LLC (and much more efficient than metaface) in training, since we produce closed form solutions in both dictionary update and sparse coding steps in our algorithm. As for classifying a test input, the computation time of ours is comparable to that of LLC, since both utilize (11) when calculating the sparse coefficient. The SRC and metaface learning required longer classification time, and this is because that both need to solve the ℓ_1 minimization problems to obtain the sparse coefficient.

4.2. Handwritten digit recognition

In the second part of our experiment, we address the problem of handwritten digit recognition for the USPS dataset [42], which is composed of 7291 training images and 2007 test images of size 16×16 pixels (see Fig. 7 for examples). PCA is applied to reduce the image dimension from 256 to 64, and the number of dictionary atoms per class is 180. Unlike the case of face recognition, we have more training samples in handwritten digit recognition. Therefore, we choose G-LSRC (see Section 3.4) as the classification rule rather than LSRC, because G-LSRC is more time efficient than LSRC under this scenario. We compare our approach with several recently proposed dictionary leaning methods (with sparse representation) [23,7,9], SRC [4], and k-nearest neighbor (kNN) classifier. For our method and LLC, we use the same classification rule and the same parameters $\sigma = 0.6$, $\lambda_{\text{DL}} = 1$, $\lambda_{\text{SRC}} = 0.1$.

Table 3
Error rates for different dictionary leaning methods on the USPS handwritten digit dataset.

kNN	SRC [4]	FDDL [9]	Ramirez [7]	LLC [23]	Ours
5.2	4.24	3.69	3.98	4.48	3.79

Fig. 7. Example images from the USPS handwritten digit dataset.

Table 3[2] lists the recognition performance of different methods. The best result is produced by FDDL [9], while our approach achieves a comparable error rate. We note that FDDL is more complex than our method, since FDDL aims at minimizing a objective function including the reconstruction error, within-class scatter, between-class scatter, and a non-differentiable term. Therefore, our algorithm is computationally more efficient than FDDL.

5. Conclusion

In this paper, we presented a novel dictionary learning algorithm for SRC with the exploitation of data locality. Employing the locality regularization term for data reconstruction results in improved data representation and classification due to the ability to preserve data locality. In addition, the introduction of the data locality constraint also implies sparse representation, which has been shown to produce promising results on many image classification tasks. By comparing with LLC, our proposed dictionary learning algorithm keeps the locality adaptor during the dictionary update stage. This carries our method to compute the dictionary **D** more accurately while LLC only provides an approximated solution. Furthermore, with the locality regularization term, closed forms solution can be easily derived for the dictionary update and sparse coding stages. Therefore, our dictionary learning algorithm is computationally more efficient to solve than those which require to solve ℓ_0 or ℓ_1-norm minimization problems. Since our classification rule is based on the minimum class-wise reconstruction error, we do not need to train additional classifiers like some of prior SRC based methods did. Finally, experiments on both face and handwritten digit recognition support the effectiveness and efficiency of our proposed dictionary learning algorithm.

[2] We implement the methods of SRC and LLC and report the error rates, while the remaining are quoted from [7,9] directly.

Appendix A. Positive definiteness of U

Note that both **U** in (15) and (17) can be expressed as $\mathbf{U}=\sum_{i=1}^{N}(\boldsymbol{\alpha}_i\boldsymbol{\alpha}_i^\top+\boldsymbol{\Omega}_i)$, where

$$\boldsymbol{\Omega}_i=\begin{cases}\mathrm{diag}(\lambda\alpha_{i1}^2,\ldots,\lambda\alpha_{iK}^2) \text{ for } \mathbf{U} \text{ defined in (15)},\\ \mathrm{diag}(\lambda_{i1}\alpha_{i1}^2,\ldots,\lambda_{iK}\alpha_{iK}^2) \text{ for } \mathbf{U} \text{ defined in (17)}.\end{cases}$$

Therefore, $\mathbf{z}^\top\mathbf{U}\mathbf{z}\geq 0$ for any $\mathbf{z}\in\mathbb{R}^{K\times 1}$, which assures that **U** is positive semi-definite. If we can show **U** is nonsingular, then the positive definiteness of **U** is proved. If **z** belongs to the null space of **U**, then $\mathbf{z}^\top\mathbf{U}\mathbf{z}=0$, which implies $\sum_{i=1}^{N}\mathbf{z}^\top\boldsymbol{\Omega}_i\mathbf{z}=0$ and $\sum_{i=1}^{N}\mathbf{z}^\top\boldsymbol{\alpha}_i\boldsymbol{\alpha}_i^\top\mathbf{z}=0$. The last equality further implies $\mathbf{z}^\top\boldsymbol{\alpha}_i=0$ for $i=1,\ldots,N$, or equivalently,

$$\mathbf{z}^\top[\boldsymbol{\alpha}_1\ \cdots\ \boldsymbol{\alpha}_N]=\mathbf{z}^\top\mathbf{A}=\mathbf{0}.$$

If **A** has full row rank, i.e., $\mathrm{rank}(\mathbf{A})=K$, then the only vector **z** in $\mathbb{R}^{K\times 1}$ satisfying $\mathbf{z}^\top\mathbf{A}=\mathbf{0}$ is the zero vector. This means if $\mathbf{z}\neq\mathbf{0}$, then $\mathbf{z}^\top\mathbf{U}\mathbf{z}\neq\mathbf{0}$. Hence, the null space of **U** is $\{\mathbf{0}\}$, and the nonsingularity of **U** is proved. If **A** does not have full row rank, i.e., $\mathrm{rank}(\mathbf{A})=r<K$, we can use the full rank decomposition to factorize **A** as $\mathbf{A}_1\mathbf{A}_2$, where $\mathbf{A}_1\in\mathbb{R}^{K\times r}$ and $\mathbf{A}_2\in\mathbb{R}^{r\times N}$. Then restart the dictionary learning algorithm with the new initial dictionary $\mathbf{D}\mathbf{A}_1\in\mathbb{R}^{d\times r}$, and the sizes of **D** and **A** are changed from $d\times K$ and $K\times N$ to $d\times r$ and $r\times N$, respectively. One can repeat this process until **A** has full row rank so that the null space of **U** is $\{\mathbf{0}\}$ (i.e., **U** is nonsingular), and the positive definiteness of **U** is proved.

References

[1] M. Elad, M. Figueiredo, Y. Ma, On the role of sparse and redundant representations in image processing, Proceedings of the IEEE 98 (6) (2010) 972–982.

[2] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. Huang, S. Yan, Sparse representation for computer vision and pattern recognition, Proceedings of the IEEE 98 (6) (2010) 1031–1044.

[3] I. Toić, P. Frossard, Dictionary learning: what is the right representation for my signal? IEEE Signal Processing Magazine 28 (2) (2011) 27–38.

[4] J. Wright, A. Yang, A. Ganesh, S. Sastry, Y. Ma, Robust face recognition via sparse representation, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (2) (2009) 210–227.

[5] X.-T. Yuan, S. Yan, Visual classification with multi-task joint sparse representation, in: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2010, pp. 3493–3500.

[6] A. Wagner, J. Wright, A. Ganesh, Z. Zhou, H. Mobahi, Y. Ma, Towards a practical face recognition system: robust alignment and illumination by sparse representation, IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (2) (2012) 372–386.

[7] I. Ramirez, P. Sprechmann, G. Sapiro, Classification and clustering via dictionary learning with structured incoherence and shared features, in: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2010, pp. 3501–3508.

[8] J. Yang, J. Wang, T. Huang, Learning the sparse representation for image classification, in: Proceedings of the IEEE International Conference on Multimedia & Expo (ICME), 2011.

[9] M. Yang, L. Zhang, X. Feng, D. Zhang, Fisher discrimination dictionary learning for sparse representation, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2011, pp. 543–550.

[10] M. Aharon, M. Elad, A. Bruckstein, K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation, IEEE Transactions on Signal Processing 54 (11) (2006) 4311–4322.

[11] M. Elad, M. Aharon, Image denoising via sparse and redundant representations over learned dictionaries, IEEE Transactions on Image Processing 15 (12) (2006) 3736–3745.

[12] J. Mairal, M. Elad, G. Sapiro, Sparse representation for color image restoration, IEEE Transactions on Image Processing 17 (1) (2008) 53–69.

[13] J. Mairal, F. Bach, J. Ponce, G. Sapiro, A. Zisserman, Non-local sparse models for image restoration, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2009, pp. 2272–2279.

[14] J. Mairal, F. Bach, J. Ponce, G. Sapiro, Online dictionary learning for sparse coding, in: Proceedings of the International Conference on Machine Learning (ICML), 2009, pp. 689–696.

[15] K. Engan, K. Skretting, J. Herredsvela, T. Gulsrud, Frame texture classification method applied on mammograms for detection of abnormalities, International Journal of Signal Processing 4 (2) (2008) 122–132.

[16] J. Mairal, F. Bach, J. Ponce, G. Sapiro, A. Zisserman, Supervised dictionary learning, in: Advances in Neural Information Processing Systems, vol. 21, 2009, pp. 1033–1040.

[17] J. Yang, K. Yu, Y. Gong, T. Huang, Linear spatial pyramid matching using sparse coding for image classification, in: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2009, pp. 1794–1801.

[18] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (5500) (2000) 2323–2326.

[19] J. Tenenbaum, V. De Silva, J. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (5500) (2000) 2319–2323.

[20] A. Elgammal, R. Duraiswami, L. Davis, Efficient kernel density estimation using the fast gauss transform with applications to color modeling and tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (11) (2003) 1499–1504.

[21] M. Breunig, H.-P. Kriegel, R.T. Ng, J. Sander, LOF: identifying density-based local outliers, in: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, 2000.

[22] K. Yu, T. Zhang, Y. Gong, Nonlinear learning using local coordinate coding, in: Advances in Neural Information Processing Systems, vol. 22, 2009, pp. 2223–2231.

[23] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, Y. Gong, Locality-constrained linear coding for image classification, in: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2010, pp. 3360–3367.

[24] Y.-W. Chao, Y.-R. Yeh, Y.-W. Chen, Y.-J. Lee, Y.-C.F. Wang, Locality-constrained group sparse representation for robust face recognition, in: Proceedings of the IEEE International Conference on Image Processing (ICIP), 2011, pp. 761–764.

[25] M. Zheng, J. Bu, C. Chen, C. Wang, L. Zhang, G. Qiu, D. Cai, Graph regularized sparse coding for image representation, IEEE Transactions on Image Processing 20 (5) (2011) 1327–1336.

[26] B. Natarajan, Sparse approximate solutions to linear systems, SIAM Journal on Computing 24 (2) (1995) 227–234.

[27] S.G. Mallat, Z. Zhang, Matching pursuits with time-frequency dictionaries, IEEE Transactions on Signal Processing 41 (12) (1993) 3397–3415.

[28] J.A. Tropp, Greed is good: algorithmic results for sparse approximation, IEEE Transactions on Information Theory 50 (10) (2004) 2231–2242.

[29] D. Donoho, Y. Tsaig, Fast solution of ℓ_1-norm minimization problems when the solution may be sparse, IEEE Transactions on Information Theory 54 (11) (2008) 4789–4812.

[30] S.S. Chen, D.L. Donoho, M.A. Saunders, Atomic decomposition by basis pursuit, SIAM Review 43 (1) (2001) 129–159.

[31] D.L. Donoho, For most large underdetermined systems of linear equations the minimal l1-norm solution is also the sparsest solution, Communications on Pure and Applied Mathematics 59 (6) (2006) 797–829.

[32] R. Tibshirani, Regression shrinkage and selection via the lasso, Journal of the Royal Statistical Society, Series B 58 (1996) 267–288.

[33] J.A. Tropp, S.J. Wright, Computational methods for sparse solution of linear inverse problems, Proceedings of the IEEE 98 (6) (2010) 948–958.

[34] H. Lee, A. Battle, R. Raina, A.Y. Ng, Efficient sparse coding algorithms, in: Advances in Neural Information Processing Systems, vol. 19, 2007, pp. 801–808.

[35] M. Yang, L. Zhang, J. Yang, D. Zhang, Metaface learning for sparse representation based face recognition, in: Proceedings of the IEEE International Conference on Image Processing (ICIP), 2010, pp. 1601–1604.

[36] J. Mairal, F. Bach, J. Ponce, G. Sapiro, A. Zisserman, Discriminative learned dictionaries for local image analysis, in: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2008.

[37] D.-S. Pham, S. Venkatesh, Joint learning and dictionary construction for pattern recognition, in: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2008.

[38] Q. Zhang, B. Li, Discriminative K-SVD for dictionary learning in face recognition, in: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2010, pp. 2691–2698.

[39] Z. Jiang, Z. Lin, L.S. Davis, Learning a discriminative dictionary for sparse coding via label consistent K-SVD, in: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2011, pp. 1697–1704.

[40] K. Huang, S. Aviyente, Sparse representation for signal classification, in: Advances in Neural Information Processing Systems, vol. 19, 2007, pp. 609–616.

[41] L. Zhang, M. Yang, X. Feng, Sparse representation or collaborative representation: which helps face recognition?, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2011.

[42] J.J. Hull, A database for handwritten text recognition research, IEEE Transactions on Pattern Analysis and Machine Intelligence 16 (5) (1994) 550–554.

[43] B. Xie, M. Song, D. Tao, Large-scale dictionary learning for local coordinate coding, in: Proceedings of the British Machine Vision Conference (BMVC), 2010, pp. 36.1–36.9.

[44] D. Bertsekas, Nonlinear Programming, Athena Scientific, 1999.

[45] L. Fei-Fei, R. Fergus, P. Perona, Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories, in: IEEE CVPR Workshop on Generative-Model Based Vision, 2004.

[46] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, A. Zisserman, The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results, 〈http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html〉.

[47] M. Asif, J. Romberg, Dynamic updating for ℓ_1 minimization, IEEE Journal of Selected Topics in Signal Processing 4 (2) (2010) 421–434.

[48] A. Yang, S. Sastry, A. Ganesh, Y. Ma, Fast ℓ_1-minimization algorithms and an application in robust face recognition: a review, in: Proceedings of the IEEE International Conference on Image Processing (ICIP), 2010, pp. 1849–1852.

[49] F.S. Samaria, A.C. Harter, Parameterisation of a stochastic model for human face identification, in: Proceedings of 2nd IEEE Workshop on Applications of Computer Vision, 1994, pp. 138–142.

[50] C. Liu, H. Wechsler, Independent component analysis of gabor features for face recognition, IEEE Transactions on Neural Networks 14 (4) (2003) 919–928.

[51] A. Mohammed, R. Minhas, Q.J. Wu, M. Sid-Ahmed, Human face recognition based on multidimensional PCA and extreme learning machine, Pattern Recognition 44 (10–11) (2011) 2588–2597.

[52] A. Martinez, R. Benavente, The AR Face Database, CVC Technical Report 24.

[53] A.S. Georghiades, P.N. Belhumeur, D.J. Kriegman, From few to many: illumination cone models for face recognition under variable lighting and pose, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (6) (2001) 643–660.

[54] K.-C. Lee, J. Ho, D.J. Kriegman, Acquiring linear subspaces for face recognition under variable lighting, IEEE Transactions on Pattern Analysis on Machine Intelligence 27 (5) (2005) 684–698.