HAL open science

A Survey of Sparse Representation: Algorithms and Applications

Zheng Zhang, Yong Xu, Jian Yang, Xuelong Li, David Zhang

To cite this version:

Zheng Zhang, Yong Xu, Jian Yang, Xuelong Li, David Zhang. A Survey of Sparse Representation: Algorithms and Applications . IEEE Access, 2015, 10.1109/ACCESS.2015.2430359 . hal-01311245

HAL Id: hal-01311245

https://hal.science/hal-01311245v1

Submitted on 26 Apr 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

A survey of sparse representation: algorithms and applications

Zheng Zhang, *Student Member, IEEE,* Yong Xu, *Senior Member, IEEE,*
Jian Yang, *Member, IEEE,* Xuelong Li, *Fellow, IEEE,* and David Zhang, *Fellow, IEEE*

***Abstract*—Sparse representation has attracted much attention from researchers in fields of signal processing, image processing, computer vision and pattern recognition. Sparse representation also has a good reputation in both theoretical research and practical applications. Many different algorithms have been proposed for sparse representation. The main purpose of this article is to provide a comprehensive study and an updated review on sparse representation and to supply a guidance for researchers. The taxonomy of sparse representation methods can be studied from various viewpoints. For example, in terms of different norm minimizations used in sparsity constraints, the methods can be roughly categorized into five groups: sparse representation with l_0-norm minimization, sparse representation with l_p-norm (0<p<1) minimization, sparse representation with l_1-norm minimization and sparse representation with $l_{2,1}$-norm minimization. In this paper, a comprehensive overview of sparse representation is provided. The available sparse representation algorithms can also be empirically categorized into four groups: greedy strategy approximation, constrained optimization, proximity algorithm-based optimization, and homotopy algorithm-based sparse representation. The rationales of different algorithms in each category are analyzed and a wide range of sparse representation applications are summarized, which could sufficiently reveal the potential nature of the sparse representation theory. Specifically, an experimentally comparative study of these sparse representation algorithms was presented. The Matlab code used in this paper can be available at: http://www.yongxu.org/lunwen.html.**

***Index Terms*—Sparse representation, compressive sensing, greedy algorithm, constrained optimization, proximal algorithm, homotopy algorithm, dictionary learning**

I. Introduction

WITH advancements in mathematics, linear representation methods (LRBM) have been well studied and have recently received considerable attention [1, 2]. The sparse representation method is the most representative methodology of the LRBM and has also been proven to be an extraordinary powerful solution to a wide range of application fields, especially in signal processing, image processing, machine learning, and computer vision, such as image denoising, debluring, inpainting, image restoration, super-resolution, visual tracking, image classification and image segmentation [3–10]. Sparse representation has shown huge potential capabilities in handling these problems.

Sparse representation, from the viewpoint of its origin, is directly related to compressed sensing (CS) [11–13], which is one of the most popular topics in recent years. Donoho [11] first proposed the original concept of compressed sensing. CS theory suggests that if a signal is sparse or compressive, the original signal can be reconstructed by exploiting a few measured values, which are much less than the ones suggested by previously used theories such as Shannon's sampling theorem (SST). Candes et al. [13], from the mathematical perspective, demonstrated the rationale of CS theory, i.e. the original signal could be precisely reconstructed by utilizing a small portion of Fourier transformation coefficients. Baraniuk [12] provided a concrete analysis of compressed sensing and presented a specific interpretation on some solutions of different signal reconstruction algorithms. All these literature [11–17] laid the foundation of CS theory and provided the theoretical basis for future research. Thus, a large number of algorithms based on CS theory have been proposed to address different problems in various fields. Moreover, CS theory always includes the three basic components: sparse representation, encoding measuring, and reconstructing algorithm. As an indispensable prerequisite of CS theory, the sparse representation theory [4, 7–10, 17] is the most outstanding technique used to conquer difficulties that appear in many fields. For example, the methodology of sparse representation is a novel signal sampling method for the sparse or compressible signal and has been successfully applied to signal processing [4–6].

Sparse representation has attracted much attention in recent years and many examples in different fields can be found where sparse representation is definitely beneficial and favorable [18, 19]. One example is image classification, where the basic goal is to classify the given test image into several predefined categories. It has been demonstrated that natural images can be sparsely represented from the perspective of the properties of visual neurons. The sparse representation based classification (SRC) method [20] first assumes that the test sample can be sufficiently represented by samples from the same subject. Specifically, SRC exploits the linear combination of training samples to represent the test sample and computes sparse representation coefficients of the linear representation system, and then calculates the reconstruction residuals of each class employing the sparse representation

Zheng Zhang and Yong Xu is with the Bio-Computing Research Center, Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen 518055, Guangdong, P.R. China; Key Laboratory of Network Oriented Intelligent Computation, Shenzhen 518055, Guangdong, P.R. China e-mail: (yongxu@ymail.com).

Jian Yang is with the College of Computer Science and Technology, Nanjing University of Science and Technology, Nanjing 210094, P. R. China.

Xuelong Li is with the Center for OPTical IMagery Analysis and Learning (OPTIMAL), State Key Laboratory of Transient Optics and Photonics, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an 710119, Shaanxi, P. R. China.

David Zhang is with the Biometrics Research Center, The Hong Kong Polytechnic University, Hong Kong

Corresponding author: Yong Xu (email: yongxu@ymail.com).

coefficients and training samples. The test sample will be classified as a member of the class, which leads to the minimum reconstruction residual. The literature [20] has also demonstrated that the SRC method has great superiorities when addressing the image classification issue on corrupted or disguised images. In such cases, each natural image can be sparsely represented and the sparse representation theory can be utilized to fulfill the image classification task.

For signal processing, one important task is to extract key components from a large number of clutter signals or groups of complex signals in coordination with different requirements. Before the appearance of sparse representation, SST and Nyquist sampling law (NSL) were the traditional methods for signal acquisition and the general procedures included sampling, coding compression, transmission, and decoding. Under the frameworks of SST and NSL, the greatest difficulty of signal processing lies in efficient sampling from mass data with sufficient memory-saving. In such a case, sparse representation theory can simultaneously break the bottleneck of conventional sampling rules, i.e. SST and NSL, so that it has a very wide application prospect. Sparse representation theory proposes to integrate the processes of signal sampling and coding compression. Especially, sparse representation theory employs a more efficient sampling rate to measure the original sample by abandoning the pristine measurements of SST and NSL, and then adopts an optimal reconstruction algorithm to reconstruct samples. In the context of compressed sensing, it is first assumed that all the signals are sparse or approximately sparse enough [4, 6, 7]. Compared to the primary signal space, the size of the set of possible signals can be largely decreased under the constraint of sparsity. Thus, massive algorithms based on the sparse representation theory have been proposed to effectively tackle signal processing issues such as signal reconstruction and recovery. To this end, the sparse representation technique can save a significant amount of sampling time and sample storage space and it is favorable and advantageous.

A. Categorization of sparse representation techniques

Sparse representation theory can be categorized from different viewpoints. Because different methods have their individual motivations, ideas, and concerns, there are varieties of strategies to separate the existing sparse representation methods into different categories from the perspective of taxonomy. For example, from the viewpoint of "atoms", available sparse representation methods can be categorized into two general groups: naive sample based sparse representation and dictionary learning based sparse representation. However, on the basis of the availability of labels of "atoms", sparse representation and learning methods can be coarsely divided into three groups: supervised learning, semi-supervised learning, and unsupervised learning methods. Because of the sparse constraint, sparse representation methods can be divided into two communities: structure constraint based sparse representation and sparse constraint based sparse representation. Moreover, in the field of image classification, the representation based classification methods consist of two main categories in terms of the way of exploiting the "atoms": the holistic representation based method and local representation based method [21]. More specifically, holistic representation based methods exploit training samples of all classes to represent the test sample, whereas local representation based methods only employ training samples (or atoms) of each class or several classes to represent the test sample. Most of the sparse representation methods are holistic representation based methods. A typical and representative local sparse representation methods is the two-phase test sample sparse representation (TPTSR) method [9]. In consideration of different methodologies, the sparse representation method can be grouped into two aspects: pure sparse representation and hybrid sparse representation, which improves the pre-existing sparse representation methods with the aid of other methods. The literature [22] suggests that sparse representation algorithms roughly fall into three classes: convex relaxation, greedy algorithms, and combinational methods. In the literature [23, 24], from the perspective of sparse problem modeling and problem solving, sparse decomposition algorithms are generally divided into two sections: greedy algorithms and convex relaxation algorithms. On the other hand, if the viewpoint of optimization is taken into consideration, the problems of sparse representation can be divided into four optimization problems: the smooth convex problem, nonsmooth nonconvex problem, smooth nonconvex problem, and nonsmooth convex problem. Furthermore, Schmidt et al. [25] reviewed some optimization techniques for solving l_1-norm regularization problems and roughly divided these approaches into three optimization strategies: sub-gradient methods, unconstrained approximation methods, and constrained optimization methods. The supplementary file attached with the paper also offers more useful information to make fully understandings of the 'taxonomy' of current sparse representation techniques in this paper.

In this paper, the available sparse representation methods are categorized into four groups, i.e. the greedy strategy approximation, constrained optimization strategy, proximity algorithm based optimization strategy, and homotopy algorithm based sparse representation, with respect to the analytical solution and optimization viewpoints.

(1) In the greedy strategy approximation for solving sparse representation problem, the target task is mainly to solve the sparse representation method with l_0-norm minimization. Because of the fact that this problem is an NP-hard problem [26], the greedy strategy provides an approximate solution to alleviate this difficulty. The greedy strategy searches for the best local optimal solution in each iteration with the goal of achieving the optimal holistic solution [27]. For the sparse representation method, the greedy strategy approximation only chooses the most k appropriate samples, which are called k-sparsity, to approximate the measurement vector.

(2) In the constrained optimization strategy, the core idea is to explore a suitable way to transform a non-differentiable optimization problem into a differentiable optimization problem by replacing the l_1-norm minimization term, which is convex but nonsmooth, with a differentiable optimization term, which is convex and smooth. More specifically, the constrained optimization strategy substitutes the l_1-norm minimization term

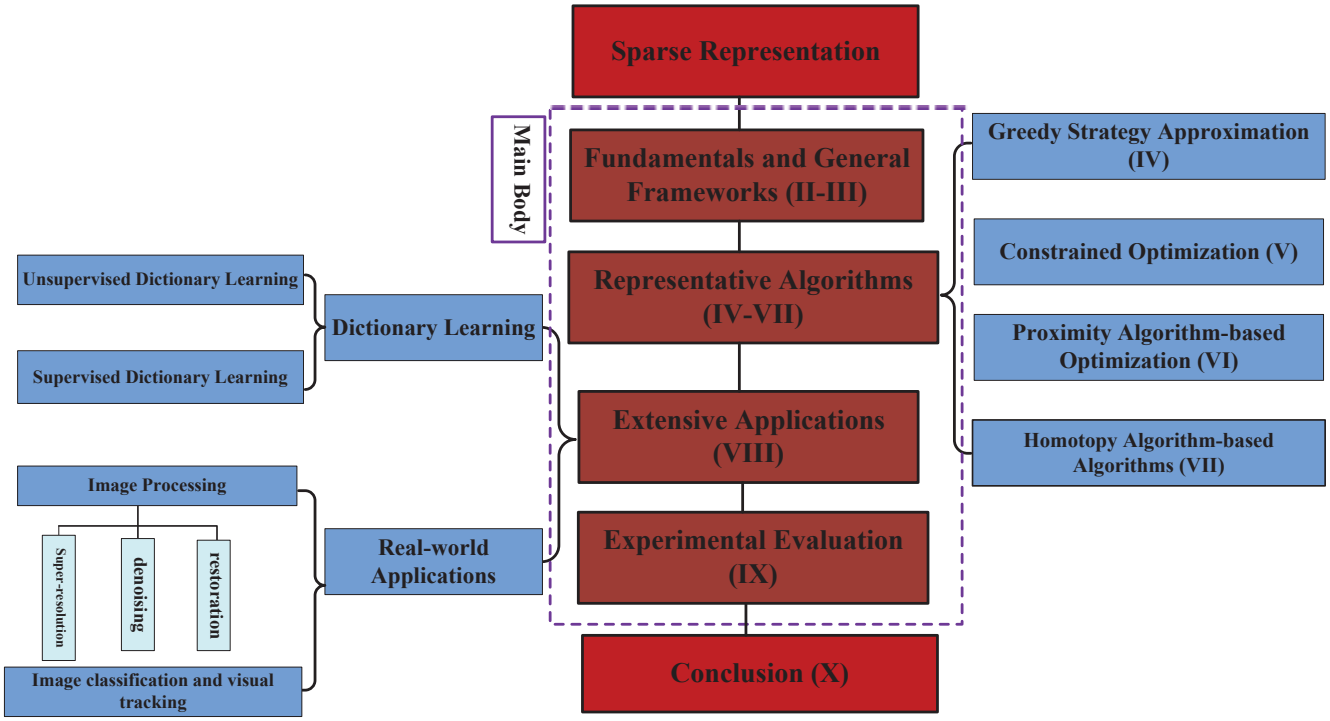


Fig. 1: The structure of this paper. The main body of this paper mainly consists of four parts: basic concepts and frameworks in Section II-III, representative algorithms in Section IV-VII and extensive applications in Section VIII, massive experimental evaluations in Section IX. Conclusion is summarized in Section X.

with an equal constraint condition on the original unconstraint problem. If the original unconstraint problem is reformulated into a differentiable problem with constraint conditions, it will become an uncomplicated problem in the consideration of the fact that l_1-norm minimization is global non-differentiable.

(3) Proximal algorithms can be treated as a powerful tool for solving nonsmooth, constrained, large-scale, or distributed versions of the optimization problem [28]. In the proximity algorithm based optimization strategy for sparse representation, the main task is to reformulate the original problem into the specific model of the corresponding proximal operator such as the soft thresholding operator, hard thresholding operator, and resolvent operator, and then exploits the proximity algorithms to address the original sparse optimization problem.

(4) The general framework of the homotopy algorithm is to iteratively trace the final desired solution starting from the initial point to the optimal point by successively adjusting the homotopy parameter [29]. In homotopy algorithm based sparse representation, the homotopy algorithm is used to solve the l_1-norm minimization problem with k-sparse property.

B. Motivation and objectives

In this paper, a survey on sparse representation and overview available sparse representation algorithms from viewpoints of the mathematical and theoretical optimization is provided. This paper is designed to provide foundations of the study on sparse representation and aims to give a good start to newcomers in computer vision and pattern recognition communities, who are interested in sparse representation methodology and its related fields. Extensive state-of-art sparse representation methods are summarized and the ideas, algorithms, and wide applications of sparse representation are comprehensively presented. Specifically, there is concentration on introducing an up-to-date review of the existing literature and presenting some insights into the studies of the latest sparse representation methods. Moreover, the existing sparse representation methods are divided into different categories. Subsequently, corresponding typical algorithms in different categories are presented and their distinctness is explicitly shown. Finally, the wide applications of these sparse representation methods in different fields are introduced.

The remainder of this paper is mainly composed of four parts: basic concepts and frameworks are shown in Section II and Section III, representative algorithms are presented in Section IV-VII and extensive applications are illustrated in Section VIII, massive experimental evaluations are summarized in Section IX. More specifically, the fundamentals and preliminary mathematic concepts are presented in Section II, and then the general frameworks of the existing sparse representation with different norm regularizations are summarized in Section III. In Section IV, the greedy strategy approximation method is presented for obtaining a sparse representation solution, and in Section V, the constrained optimization strategy is introduced for solving the sparse representation issue. Furthermore, the proximity algorithm based optimization strategy and Homotopy strategy for addressing the sparse representation problem are outlined in Section VI and Section VII, respectively. Section VIII presents extensive applications of sparse represen-

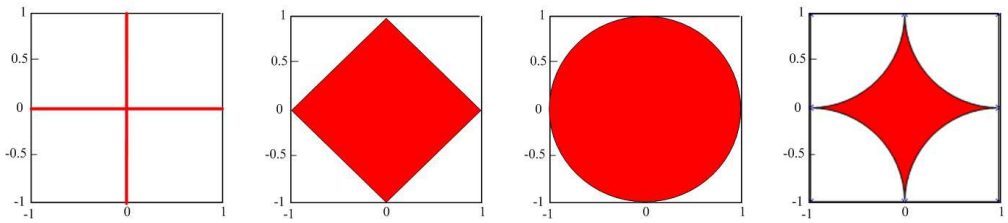


Fig. 2: Geometric interpretations of different norms in 2-D space [7]. (a), (b), (c), (d) are the unit ball of the l_0-norm, l_1-norm, l_2-norm, l_p-norm (0<p<1) in 2-D space, respectively. The two axes of the above coordinate systems are x_1 and x_2.

tation in widespread and prevalent fields including dictionary learning methods and real-world applications. Finally, Section IX offers massive experimental evaluations and conclusions are drawn and summarized in Section X. The structure of the this paper has been summarized in Fig. 1.

II. Fundamentals and preliminary concepts

A. Notations

In this paper, vectors are denoted by lowercase letters with bold face, e.g. $\boldsymbol{x}$. Matrices are denoted by uppercase letter, e.g. X and their elements are denoted with indexes such as X_i. In this paper, all the data are only real-valued.

Suppose that the sample is from space $\mathbb{R}^d$ and thus all the samples are concatenated to form a matrix, denoted as $D \in \mathbb{R}^{d\times n}$. If any sample can be approximately represented by a linear combination of dictionary D and the number of the samples is larger than the dimension of samples in D, i.e. $n > d$, dictionary D is referred to as an over-complete dictionary. A signal is said to be compressible if it is a sparse signal in the original or transformed domain when there is no information or energy loss during the process of transformation.

"**sparse**" or "**sparsity**" of a vector means that some elements of the vector are zero. We use a linear combination of a basis matrix $A \in R^{N\times N}$ to represent a signal $\boldsymbol{x} \in R^{N\times 1}$, i.e. $\boldsymbol{x} = A\boldsymbol{s}$ where $\boldsymbol{s} \in R^{N\times 1}$ is the column vector of weighting coefficients. If only k ($k \ll N$) elements of $\boldsymbol{s}$ are nonzero and the rest elements in $\boldsymbol{s}$ are zero, we call the signal $\boldsymbol{x}$ is k-sparse.

B. Basic background

The standard inner product of two vectors, $\boldsymbol{x}$ and $\boldsymbol{y}$ from the set of real n dimensions, is defined as

$$\langle \boldsymbol{x}, \boldsymbol{y}\rangle = \boldsymbol{x}^T\boldsymbol{y} = \boldsymbol{x}_1\boldsymbol{y}_1 + \boldsymbol{x}_2\boldsymbol{y}_2 + \cdots + \boldsymbol{x}_n\boldsymbol{y}_n \tag{II.1}$$

The standard inner product of two matrixes, $X \in \mathbb{R}^{m\times n}$ and $Y \in \mathbb{R}^{m\times n}$ from the set of real $m \times n$ matrixes, is denoted as the following equation

$$\langle X, Y\rangle = tr(X^TY) = \sum_{i=1}^{m}\sum_{j=1}^{n} X_{ij}Y_{ij} \tag{II.2}$$

where the operator $tr(A)$ denotes the trace of the matrix A, i.e. the sum of its diagonal entries.

Suppose that $\boldsymbol{v} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_n]$ is an n dimensional vector in Euclidean space, thus

$$\|\boldsymbol{v}\|_p = (\sum_{i=1}^{n} |\boldsymbol{v}_i|^p)^{1/p} \tag{II.3}$$

is denoted as the p-norm or the l_p-norm ($1 \le p \le \infty$) of vector v.

When p=1, it is called the l_1-norm. It means the sum of absolute values of the elements in vector v , and its geometric interpretation is shown in Fig. 2b, which is a square with a forty-five degree rotation.

When p=2, it is called the l_2-norm or Euclidean norm. It is defined as $\|\boldsymbol{v}\|_2 = (\boldsymbol{v}_1^2 + \boldsymbol{v}_2^2 + \cdots + \boldsymbol{v}_n^2)^{1/2}$, and its geometric interpretation in 2-D space is shown in Fig. 2c which is a circle.

In the literature, the sparsity of a vector v is always related to the so-called l_0-norm, which means the number of the nonzero elements of vector $\boldsymbol{v}$. Actually, the l_0-norm is the limit as $p \to 0$ of the l_p-norms [8] and the definition of the l_0-norm is formulated as

$$\|\boldsymbol{v}\|_0 = \lim_{p\to 0}\|\boldsymbol{v}\|_p^p = \lim_{p\to 0}\sum_{i=1}^{n}|v_i|^p \tag{II.4}$$

We can see that the notion of the l_0-norm is very convenient and intuitive for defining the sparse representation problem. The property of the l_0-norm can also be presented from the perspective of geometric interpretation in 2-D space, which is shown in Fig. 2a, and it is a crisscross.

Furthermore, the geometric meaning of the l_p-norm (0<p<1) is also presented, which is a form of similar recessed pentacle shown in Fig. 2d.

On the other hand, it is assumed that $f(x)$ is the function of the l_p-norm (p>0) on the parameter vector $\boldsymbol{x}$, and then the following function is obtained:

$$f(x) = \|\boldsymbol{x}\|_p^p = (\sum_{i=1}^{n}|\boldsymbol{x}_i|^p) \tag{II.5}$$

The relationships between different norms are summarized in Fig. 3. From the illustration in Fig. 3, the conclusions are as follows. The l_0-norm function is a nonconvex, nonsmooth, discontinuity, global nondifferentiable function. The l_p-norm (0<p<1) is a nonconvex, nonsmooth, global nondifferentiable function. The l_1-norm function is a convex, nonsmooth, global nondifferentiable function. The l_2-norm function is a convex, smooth, global differentiable function.

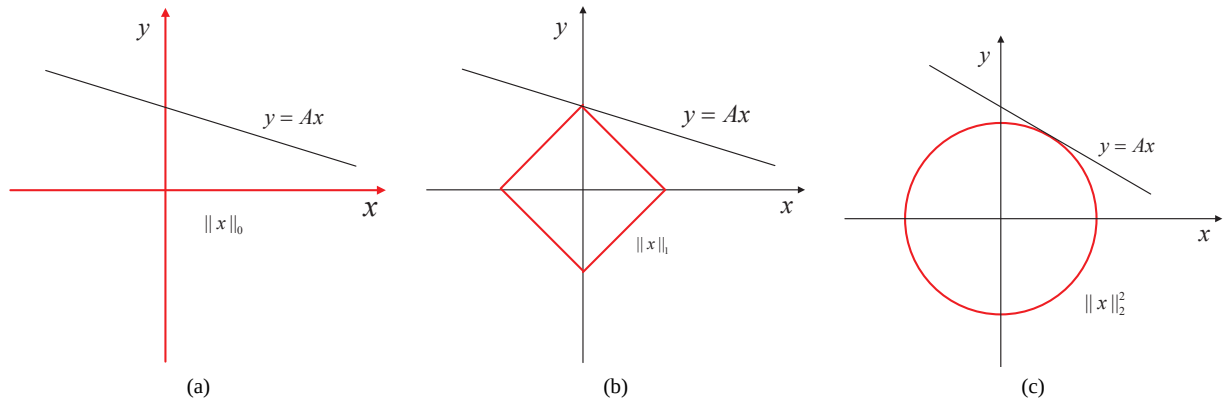


Fig. 4: The geometry of the solutions of different norm regularization in 2-D space [7]. (a), (b) and (c) are the geometry of the solutions of the l_0-norm, l_1-norm, l_2-norm minimization, respectively.

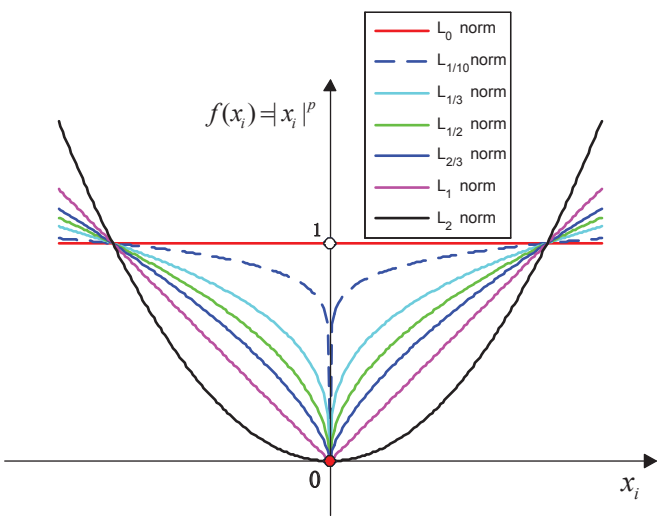


Fig. 3: Geometric interpretations of different norms in 1-D space [7].

In order to more specifically elucidate the meaning and solutions of different norm minimizations, the geometry in 2-D space is used to explicitly illustrate the solutions of the l_0-norm minimization in Fig. 4a, l_1-norm minimization in Fig. 4b, and l_2-norm minimization in Fig. 4c. Let $S = \{x^* : Ax = y\}$ denote the line in 2-D space and a hyperplane will be formulated in higher dimensions. All possible solution x^* must lie on the line of S. In order to visualize how to obtain the solution of different norm-based minimization problems, we take the l_1-norm minimization problem as an example to explicitly interpret. Suppose that we inflate the l_1-ball from an original status until it hits the hyperplane S at some point. Thus, the solution of the l_1-norm minimization problem is the aforementioned touched point. If the sparse solution of the linear system is localized on the coordinate axis, it will be sparse enough. From the perspective of Fig. 4, it can be seen that the solutions of both the l_0-norm and l_1-norm minimization are sparse, whereas for the l_2-norm minimization, it is very difficult to rigidly satisfy the condition of sparsity. However, it has been demonstrated that the representation solution of the l_2-norm minimization is not strictly sparse enough but "limitedly-sparse", which means it possesses the capability of discriminability [30].

The Frobenius norm, L_1-norm of matrix $X \in \mathbb{R}^{m\times n}$, and l_2-norm or spectral norm are respectively defined as

$$\|X\|_F = (\sum_{i=1}^{n}\sum_{j=1}^{m} X_{j,i}^2)^{1/2}, \|X\|_{L_1} = max_{j=1,\dots,n}\sum_{i=1}^{m}|x_{ij}|,$$
$$\|X\|_2 = \delta_{max}(X) = (\lambda_{max}(X^TX))^{1/2} \tag{II.6}$$

where δ is the singular value operator and the l_2-norm of X is its maximum singular value [31].

The $l_{2,1}$-norm or R_1-norm is defined on matrix term, that is

$$\|X\|_{2,1} = \sum_{i=1}^{n}(\sum_{j=1}^{m} X_{j,i}^2)^{1/2} \tag{II.7}$$

As shown above, a norm can be viewed as a measure of the length of a vector $\boldsymbol{v}$. The distance between two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, or matrices X and Y, can be measured by the length of their differences, i.e.

$$dist(\boldsymbol{x},\boldsymbol{y}) = \|\boldsymbol{x}-\boldsymbol{y}\|_2^2, \quad dist(X,Y) = \|X-Y\|_F \tag{II.8}$$

which are denoted as the distance between $\boldsymbol{x}$ and $\boldsymbol{y}$ in the context of the l_2-norm and the distance between X and Y in the context of the Frobenius norm, respectively.

Assume that $X \in \mathbb{R}^{m\times n}$ and the rank of X, i.e. $rank(X) = r$. The SVD of X is computed as

$$X = U\Lambda V^T \tag{II.9}$$

where $U \in \mathbb{R}^{m\times r}$ with $U^TU = I$ and $V \in \mathbb{R}^{n\times r}$ with $V^TV = I$. The columns of U and V are called left and right singular vectors of X, respectively. Additionally, Λ is a diagonal matrix and its elements are composed of the singular values of X, i.e. $\Lambda = diag(\lambda_1, \lambda_2, \cdots, \lambda_r)$ with $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_r > 0$. Furthermore, the singular value decomposition can be rewritten as

$$X = \sum_{i=1}^{r}\lambda_i \boldsymbol{u}_i \boldsymbol{v}_i \tag{II.10}$$

where λ_i, $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$ are the i-th singular value, the i-th column of U, and the i-th column of V, respectively [31].

III. Sparse representation problem with different norm regularizations

In this section, sparse representation is summarized and grouped into different categories in terms of the norm regularizations used. The general framework of sparse representation is to exploit the linear combination of some samples or "atoms" to represent the probe sample, to calculate the representation solution, i.e. the representation coefficients of these samples or "atoms", and then to utilize the representation solution to reconstruct the desired results. The representation results in sparse representation, however, can be greatly dominated by the regularizer (or optimizer) imposed on the representation solution [32–35]. Thus, in terms of the different norms used in optimizers, the sparse representation methods can be roughly grouped into five general categories: sparse representation with the l_0-norm minimization [36, 37], sparse representation with the l_p-norm (0<p<1) minimization [38–40], sparse representation with the l_1-norm minimization [41–44], sparse representation with the $l_{2,1}$-norm minimization [45–49], sparse representation with the l_2-norm minimization [9, 50, 51].

A. Sparse representation with l_0-norm minimization

Let $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n \in \mathbb{R}^d$ be all the n known samples and matrix $X \in \mathbb{R}^{d\times n}$ (d<n), which is constructed by known samples, is the measurement matrix or the basis dictionary and should also be an over-completed dictionary. Each column of X is one sample and the probe sample is $\boldsymbol{y} \in \mathbb{R}^d$, which is a column vector. Thus, if all the known samples are used to approximately represent the probe sample, it should be expressed as:

$$\boldsymbol{y} = \boldsymbol{x}_1\boldsymbol{\alpha}_1 + \boldsymbol{x}_2\boldsymbol{\alpha}_2 + \cdots + \boldsymbol{x}_n\boldsymbol{\alpha}_n \tag{III.1}$$

where $\boldsymbol{\alpha}_i$ (i=1,2,$\cdots$,n) is the coefficient of $\boldsymbol{x}_i$ and Eq. III.1 can be rewritten into the following equation for convenient description:

$$\boldsymbol{y} = X\boldsymbol{\alpha} \tag{III.2}$$

where matrix X=[$\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n$] and $\boldsymbol{\alpha}$=[$\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \cdots, \boldsymbol{\alpha}_n$]^T.

However, problem III.2 is an underdetermined linear system of equations and the main problem is how to solve it. From the viewpoint of linear algebra, if there is not any prior knowledge or any constraint imposed on the representation solution $\boldsymbol{\alpha}$, problem III.2 is an ill-posed problem and will never have a unique solution. That is, it is impossible to utilize equation III.2 to uniquely represent the probe sample y using the measurement matrix X. To alleviate this difficulty, it is feasible to impose an appropriate regularizer constraint or regularizer function on representation solution $\boldsymbol{\alpha}$. The sparse representation method demands that the obtained representation solution should be sparse. Hereafter, the meaning of 'sparse' or 'sparsity' refers to the condition that when the linear combination of measurement matrix is exploited to represent the probe sample, many of the coefficients should be zero or very close to zero and few of the entries in the representation solution are differentially large.

The sparsest representation solution can be acquired by solving the linear representation system III.2 with the l_0-norm minimization constraint [52]. Thus problem III.2 can be converted to the following optimization problem:

$$\hat{\boldsymbol{\alpha}} = \arg\min \|\boldsymbol{\alpha}\|_0 \quad s.t. \quad \boldsymbol{y} = X\boldsymbol{\alpha} \tag{III.3}$$

where $\|\cdot\|_0$ refers to the number of nonzero elements in the vector and is also viewed as the measure of sparsity. Moreover, if just k ($k < n$) atoms from the measurement matrix X are utilized to represent the probe sample, problem III.3 will be equivalent to the following optimization problem:

$$\boldsymbol{y} = X\boldsymbol{\alpha} \quad s.t. \quad \|\boldsymbol{\alpha}\|_0 \le k \tag{III.4}$$

Problem III.4 is called the k-sparse approximation problem. Because real data always contains noise, representation noise is unavoidable in most cases. Thus the original model III.2 can be revised to a modified model with respect to small possible noise by denoting

$$\boldsymbol{y} = X\boldsymbol{\alpha} + \boldsymbol{s} \tag{III.5}$$

where $\boldsymbol{s} \in \mathbb{R}^d$ refers to representation noise and is bounded as $\|\boldsymbol{s}\|_2 \le \varepsilon$. With the presence of noise, the sparse solutions of problems III.3 and III.4 can be approximately obtained by resolving the following optimization problems:

$$\hat{\boldsymbol{\alpha}} = \arg\min \|\boldsymbol{\alpha}\|_0 \quad s.t. \quad \|\boldsymbol{y} - X\boldsymbol{\alpha}\|_2^2 \le \varepsilon \tag{III.6}$$

or

$$\hat{\boldsymbol{\alpha}} = \arg\min \|\boldsymbol{y} - X\boldsymbol{\alpha}\|_2^2 \quad s.t. \quad \|\boldsymbol{\alpha}\|_0 \le \varepsilon \tag{III.7}$$

Furthermore, according to the Lagrange multiplier theorem, a proper constant λ exists such that problems III.6 and III.7 are equivalent to the following unconstrained minimization problem with a proper value of λ.

$$\hat{\boldsymbol{\alpha}} = L(\boldsymbol{\alpha}, \lambda) = \arg\min \|\boldsymbol{y} - X\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_0 \tag{III.8}$$

where λ refers to the Lagrange multiplier associated with $\|\boldsymbol{\alpha}\|_0$.

B. Sparse representation with l_1-norm minimization

The l_1-norm originates from the Lasso problem [41, 42] and it has been extensively used to address issues in machine learning, pattern recognition, and statistics [53–55]. Although the sparse representation method with l_0-norm minimization can obtain the fundamental sparse solution of $\boldsymbol{\alpha}$ over the matrix X, the problem is still a non-deterministic polynomial-time hard (NP-hard) problem and the solution is difficult to approximate [26]. Recent literature [20, 56–58] has demonstrated that when the representation solution obtained by using the l_1-norm minimization constraint is also content with the condition of sparsity and the solution using l_1-norm minimization with sufficient sparsity can be equivalent to the solution obtained by l_0-norm minimization with full probability. Moreover, the l_1-norm optimization problem has an analytical solution and can be solved in polynomial time. Thus, extensive sparse representation methods with the l_1-norm minimization have

been proposed to enrich the sparse representation theory. The applications of sparse representation with the l_1-norm minimization are extraordinarily and remarkably widespread. Correspondingly, the main popular structures of sparse representation with the l_1-norm minimization , similar to sparse representation with l_0-norm minimization, are generally used to solve the following problems:

$$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_1 \quad s.t. \quad \boldsymbol{y} = X\boldsymbol{\alpha} \tag{III.9}$$

$$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_1 \quad s.t. \quad \|\boldsymbol{y} - X\boldsymbol{\alpha}\|_2^2 \leq \varepsilon \tag{III.10}$$

or

$$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} \|\boldsymbol{y} - X\boldsymbol{\alpha}\|_2^2 \quad s.t. \quad \|\boldsymbol{\alpha}\|_1 \leq \tau \tag{III.11}$$

$$\hat{\boldsymbol{\alpha}} = L(\boldsymbol{\alpha}, \lambda) = \arg\min_{\boldsymbol{\alpha}} \frac{1}{2}\|y - X\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1 \tag{III.12}$$

where λ and τ are both small positive constants.

C. Sparse representation with l_p-norm (0<p<1) minimization

The general sparse representation method is to solve a linear representation system with the l_p-norm minimization problem. In addition to the l_0-norm minimization and l_1-norm minimization, some researchers are trying to solve the sparse representation problem with the l_p-norm (0<p<1) minimization, especially $p = 0.1$, $\frac{1}{2}$, $\frac{1}{3}$, or 0.9 [59–61]. That is, the sparse representation problem with the l_p-norm (0<p<1) minimization is to solve the following problem:

$$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_p^p \quad s.t. \quad \|\boldsymbol{y} - X\boldsymbol{\alpha}\|_2^2 \leq \varepsilon \tag{III.13}$$

or

$$\hat{\boldsymbol{\alpha}} = L(\boldsymbol{\alpha}, \lambda) = \arg\min_{\boldsymbol{\alpha}} \|\boldsymbol{y} - X\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_p^p \tag{III.14}$$

In spite of the fact that sparse representation methods with the l_p-norm (0<p<1) minimization are not the mainstream methods to obtain the sparse representation solution, it tremendously influences the improvements of the sparse representation theory.

D. Sparse representation with $l_{2,1}$-norm minimization

The representation solution obtained by the l_2-norm minimization is not rigorously sparse. It can only obtain a 'limitedly-sparse' representation solution, i.e. the solution has the property that it is discriminative and distinguishable but is not really sparse enough [30]. The objective function of the sparse representation method with the l_2-norm minimization is to solve the following problem:

$$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_2^2 \quad s.t. \quad \|\boldsymbol{y} - X\boldsymbol{\alpha}\|_2^2 \leq \varepsilon \tag{III.15}$$

or

$$\hat{\boldsymbol{\alpha}} = L(\boldsymbol{\alpha}, \lambda) = \arg\min_{\boldsymbol{\alpha}} \|\boldsymbol{y} - X\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_2^2 \tag{III.16}$$

On the other hand, the $l_{2,1}$-norm is also called the rotation invariant l_1-norm, which is proposed to overcome the difficulty of robustness to outliers [62]. The objective function of the sparse representation problem with the $l_{2,1}$-norm minimization is to solve the following problem:

$$\arg\min_{A} \|Y - XA\|_{2,1} + \mu\|A\|_{2,1} \tag{III.17}$$

where $Y = [\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_N]$ refers to the matrix composed of samples, $A = [\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_N]$ is the corresponding coefficient matrix of X, and μ is a small positive constant. Sparse representation with the $l_{2,1}$-norm minimization can be implemented by exploiting the proposed algorithms in literature [45–47].

IV. Greedy strategy approximation

Greedy algorithms date back to the 1950s. The core idea of the greedy strategy [7, 23] is to determine the position based on the relationship between the atom and probe sample, and then to use the least square to evaluate the amplitude value. Greedy algorithms can obtain the local optimized solution in each step in order to address the problem. However, the greedy algorithm can always produce the global optimal solution or an approximate overall solution [7, 23]. Addressing sparse representation with l_0-norm regularization, i.e. problem III.3, is an NP hard problem [20, 56]. The greedy strategy provides a special way to obtain an approximate sparse representation solution. The greedy strategy actually can not directly solve the optimization problem and it only seeks an approximate solution for problem III.3.

A. Matching pursuit algorithm

The matching pursuit (MP) algorithm [63] is the earliest and representative method of using the greedy strategy to approximate problem III.3 or III.4. The main idea of the MP is to iteratively choose the best atom from the dictionary based on a certain similarity measurement to approximately obtain the sparse solution. Taking as an example of the sparse decomposition with a vector sample y over the over-complete dictionary D, the detailed algorithm description is presented as follows:

Suppose that the initialized representation residual is $\boldsymbol{R}_0 = \boldsymbol{y}$, $D = [\boldsymbol{d}_1, \boldsymbol{d}_2, \cdots, \boldsymbol{d}_N] \in \mathbb{R}^{d\times N}$ and each sample in dictionary D is an l_2-norm unity vector, i.e. $\|\boldsymbol{d}_i\| = 1$. To approximate $\boldsymbol{y}$, MP first chooses the best matching atom from D and the selected atom should satisfy the following condition:

$$|\langle \boldsymbol{R}_0, \boldsymbol{d}_{l_0}\rangle| = sup|\langle \boldsymbol{R}_0, \boldsymbol{d}_i\rangle| \tag{IV.1}$$

where l_0 is a label index from dictionary D. Thus $\boldsymbol{y}$ can be decomposed into the following equation:

$$\boldsymbol{y} = \langle \boldsymbol{y}, \boldsymbol{d}_{l_0}\rangle \boldsymbol{d}_{l_0} + \boldsymbol{R}_1 \tag{IV.2}$$

So $\boldsymbol{y} = \langle \boldsymbol{R}_0, \boldsymbol{d}_{l_0}\rangle \boldsymbol{d}_{l_0} + \boldsymbol{R}_1$ where $\langle \boldsymbol{R}_0, \boldsymbol{d}_{l_0}\rangle \boldsymbol{d}_{l_0}$ represents the orthogonal projection of $\boldsymbol{y}$ onto $\boldsymbol{d}_{l_0}$, and $\boldsymbol{R}_1$ is the representation residual by using $\boldsymbol{d}_{l_0}$ to represent $\boldsymbol{y}$. Considering the fact that $\boldsymbol{d}_{l_0}$ is orthogonal to $\boldsymbol{R_1}$, Eq. IV.2 can be rewritten as

$$\|\boldsymbol{y}\|^2 = |\langle \boldsymbol{y}, \boldsymbol{d}_{l_0}\rangle|^2 + \|\boldsymbol{R}_1\|^2 \tag{IV.3}$$

To obtain the minimum representation residual, the MP algorithm iteratively figures out the best matching atom from the over-completed dictionary, and then utilizes the representation residual as the next approximation target until the termination condition of iteration is satisfied. For the t-th iteration, the best matching atom is $\boldsymbol{d}_{l_t}$ and the approximation result is found from the following equation:

$$\boldsymbol{R}_t = \langle \boldsymbol{R}_t, \boldsymbol{d}_{l_t} \rangle \boldsymbol{d}_{l_t} + \boldsymbol{R}_{t+1} \tag{IV.4}$$

where the $\boldsymbol{d}_{l_t}$ satisfies the equation:

$$|\langle \boldsymbol{R}_t, \boldsymbol{d}_{l_t} \rangle| = sup|\langle \boldsymbol{R}_t, \boldsymbol{d}_i \rangle| \tag{IV.5}$$

Clearly, $\boldsymbol{d}_{l_t}$ is orthogonal to $\boldsymbol{R_{k+1}}$, and then

$$\|\boldsymbol{R}_k\|^2 = |\langle \boldsymbol{R}_t, \boldsymbol{d}_{l_t} \rangle|^2 + \|\boldsymbol{R}_{t+1}\|^2 \tag{IV.6}$$

For the n-th iteration, the representation residual $\|\boldsymbol{R}_n\|^2 \leq \tau$ where τ is a very small constant and the probe sample $\boldsymbol{y}$ can be formulated as:

$$\boldsymbol{y} = \sum_{j=1}^{n-1} \langle \boldsymbol{R}_j, \boldsymbol{d}_{l_j} \rangle \boldsymbol{d}_{l_j} + \boldsymbol{R}_n \tag{IV.7}$$

If the representation residual is small enough, the probe sample $\boldsymbol{y}$ can approximately satisfy the following equation: $\boldsymbol{y} \approx \sum_{j=1}^{n-1} \langle \boldsymbol{R}_j, \boldsymbol{d}_{l_j} \rangle \boldsymbol{d}_{l_j}$ where $n \ll N$. Thus, the probe sample can be represented by a small number of elements from a large dictionary. In the context of the specific representation error, the termination condition of sparse representation is that the representation residual is smaller than the presupposed value. More detailed analysis on matching pursuit algorithms can be found in the literature [63].

B. Orthogonal matching pursuit algorithm

The orthogonal matching pursuit (OMP) algorithm [36, 64] is an improvement of the MP algorithm. The OMP employs the process of orthogonalization to guarantee the orthogonal direction of projection in each iteration. It has been verified that the OMP algorithm can be converged in limited iterations [36]. The main steps of OMP algorithm have been summarized in Algorithm 1.

Algorithm 1. Orthogonal matching pursuit algorithm
Task: Approximate the constraint problem:
$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_0 \;\; s.t. \;\; \boldsymbol{y} = X\boldsymbol{\alpha}$

Input: Probe sample $\boldsymbol{y}$, measurement matrix X, sparse coefficients vector $\boldsymbol{\alpha}$
Initialization: $t = 1$, $\boldsymbol{r}_0 = y$, $\boldsymbol{\alpha} = 0$, $D_0 = \phi$, index set $\Lambda_0 = \phi$ where ϕ denotes empty set, τ is a small constant.
While $\|\boldsymbol{r}_t\| > \tau$ do
  Step 1: Find the best matching sample, i.e. the biggest inner product between $\boldsymbol{r}_{t-1}$ and $\boldsymbol{x}_j$ ($j \notin \Lambda_{t-1}$) by exploiting $\lambda_t = \arg\max_{j \notin \Lambda_{t-1}} |\langle \boldsymbol{r}_{t-1}, \boldsymbol{x}_j \rangle|$.
  Step 2: Update the index set $\Lambda_t = \Lambda_{t-1} \bigcup \lambda_t$ and reconstruct data set $D_t = [D_{t-1}, \boldsymbol{x}_{\lambda_t}]$.
  Step 3: Compute the sparse coefficient by using the least square algorithm $\tilde{\boldsymbol{\alpha}} = \arg\min \|\boldsymbol{y} - D_t \tilde{\boldsymbol{\alpha}}\|_2^2$.
  Step 4: Update the representation residual using $\boldsymbol{r}_t = \boldsymbol{y} - D_t \tilde{\boldsymbol{\alpha}}$.
  Step 5: $t = t + 1$.
End
Output: D, $\boldsymbol{\alpha}$

C. Series of matching pursuit algorithms

It is an excellent choice to employ the greedy strategy to approximate the solution of sparse representation with the l_0-norm minimization. These algorithms are typical greedy iterative algorithms. The earliest algorithms were the matching pursuit (MP) and orthogonal matching pursuit (OMP). The basic idea of the MP algorithm is to select the best matching atom from the overcomplete dictionary to construct sparse approximation during each iteration, to compute the signal representation residual, and then to choose the best matching atom till the stopping criterion of iteration is satisfied. Many more greedy algorithms based on the MP and OMP algorithm such as the efficient orthogonal matching pursuit algorithm [65] subsequently have been proposed to improve the pursuit algorithm. Needell et al. proposed an regularized version of orthogonal matching pursuit (ROMP) algorithm [37], which recovered all k sparse signals based on the Restricted Isometry Property of random frequency measurements, and then proposed another variant of OMP algorithm called compressive sampling matching pursuit (CoSaMP) algorithm [66], which incorporated several existing ideas such as restricted isometry property (RIP) and pruning technique into a greedy iterative structure of OMP. Some other algorithms also had an impressive influence on future research on CS. For example, Donoho et al. proposed an extension of OMP, called stage-wise orthogonal matching pursuit (StOMP) algorithm [67], which depicted an iterative algorithm with three main steps, i.e. threholding, selecting and projecting. Dai and Milenkovic proposed a new method for sparse signal reconstruction named subspace pursuit (SP) algorithm [68], which sampled signals satisfying the constraints of the RIP with a constant parameter. Do et al. presented a sparsity adaptive matching pursuit (SAMP) algorithm [69], which borrowed the idea of the EM algorithm to alternatively estimate the sparsity and support set. Jost et al. proposed a tree-based matching pursuit (TMP) algorithm [70], which constructed a tree structure and employed a structuring strategy to cluster similar signal atoms from a highly redundant dictionary as a new dictionary. Subsequently, La and Do proposed a new tree-based orthogonal matching pursuit (TBOMP) algorithm [71], which treated the sparse tree representation as an additional prior knowledge for linear inverse systems by using a small number of samples. Recently, Karahanoglu and Erdogan conceived a forward-backward pursuit (FBP) method [72] with two greedy stages, in which the forward stage enlarged the support estimation and the backward stage removed some unsatisfied atoms. More detailed treatments of the greedy pursuit for sparse representation can be found in the literature [23].

V. Constrained optimization strategy

Constrained optimization strategy is always utilized to obtain the solution of sparse representation with the l_1-norm regularization. The methods that address the non-differentiable unconstrained problem will be presented by reformulating it as a smooth differentiable constrained optimization problem. These methods exploit the constrained optimization method with efficient convergence to obtain the sparse solution. What

is more, the constrained optimization strategy emphasizes the equivalent transformation of $\|\boldsymbol{\alpha}\|_1$ in problem III.12 and employs the new reformulated constrained problem to obtain a sparse representation solution. Some typical methods that employ the constrained optimization strategy to solve the original unconstrained non-smooth problem are introduced in this section.

A. Gradient Projection Sparse Reconstruction

The core idea of the gradient projection sparse representation method is to find the sparse representation solution along with the gradient descent direction. The first key procedure of gradient projection sparse reconstruction (GPSR) [73] provides a constrained formulation where each value of $\boldsymbol{\alpha}$ can be split into its positive and negative parts. Vectors $\boldsymbol{\alpha}_+$ and $\boldsymbol{\alpha}_-$ are introduced to denote the positive and negative coefficients of $\boldsymbol{\alpha}$, respectively. The sparse representation solution $\boldsymbol{\alpha}$ can be formulated as:

$$\boldsymbol{\alpha} = \boldsymbol{\alpha}_+ - \boldsymbol{\alpha}_-,\ \boldsymbol{\alpha}_+ \geq 0,\ \boldsymbol{\alpha}_- \geq 0 \tag{V.1}$$

where the operator $(\cdot)_+$ denotes the positive-part operator, which is defined as $(x)_+$=max$\{0, x\}$. Thus, $\|\boldsymbol{\alpha}\|_1 = \mathbf{1}_d^T\boldsymbol{\alpha}_+ + \mathbf{1}_d^T\boldsymbol{\alpha}_-$, where $\mathbf{1}_d = [\underbrace{1, 1, \cdots, 1}_{d}]^T$ is a d–dimensional vector with d ones. Accordingly, problem III.12 can be reformulated as a constrained quadratic problem:

$$\begin{aligned}\arg\min L(\boldsymbol{\alpha}) = \arg\min \frac{1}{2}\|\boldsymbol{y} - X[\boldsymbol{\alpha}_+ - \boldsymbol{\alpha}_-]\|_2^2 + \\ \lambda(\mathbf{1}_d^T\boldsymbol{\alpha}_+ + \mathbf{1}_d^T\boldsymbol{\alpha}_-) \quad s.t. \quad \boldsymbol{\alpha}_+ \geq 0,\ \boldsymbol{\alpha}_- \geq 0\end{aligned} \tag{V.2}$$

or

$$\begin{aligned}\arg\min L(\boldsymbol{\alpha}) = \arg\min \frac{1}{2}\|\boldsymbol{y} - [X_+, X_-][\boldsymbol{\alpha}_+ - \boldsymbol{\alpha}_-]\|_2^2 \\ +\lambda(\mathbf{1}_d^T\boldsymbol{\alpha}_+ + \mathbf{1}_d^T\boldsymbol{\alpha}_-) \quad s.t. \quad \boldsymbol{\alpha}_+ \geq 0,\ \boldsymbol{\alpha}_- \geq 0\end{aligned} \tag{V.3}$$

Furthermore, problem V.3 can be rewritten as:

$$\arg\min\ G(\boldsymbol{z}) = \boldsymbol{c}^T\boldsymbol{z} + \frac{1}{2}\boldsymbol{z}^T A\boldsymbol{z} \quad s.t. \quad \boldsymbol{z} \geq \mathbf{0} \tag{V.4}$$

where $\boldsymbol{z} = [\boldsymbol{\alpha}_+; \boldsymbol{\alpha}_-]$, $\boldsymbol{c} = \lambda\mathbf{1}_{2d} + [-X^T\boldsymbol{y}; X^T\boldsymbol{y}]$, $\mathbf{1}_{2d} = [\underbrace{1, \cdots, 1}_{2d}]^T$, $A = \begin{pmatrix} X^TX & -X^TX \\ -X^TX & X^TX \end{pmatrix}$.

The GPSR algorithm employs the gradient descent and standard line-search method [31] to address problem V.4. The value of z can be iteratively obtained by utilizing

$$\arg\min\ \boldsymbol{z}^{t+1} = \boldsymbol{z}^t - \sigma\nabla G(\boldsymbol{z}^t) \tag{V.5}$$

where the gradient of $\nabla G(\boldsymbol{z}^t) = \boldsymbol{c} + A\boldsymbol{z}^t$ and σ is the step size of the iteration. For step size σ, GPSR updates the step size by using

$$\sigma^t = \arg\min_{\sigma}\ G(\boldsymbol{z}^t - \sigma g^t) \tag{V.6}$$

where the function g^t is pre-defined as

$$g_i^t = \begin{cases} (\nabla G(\boldsymbol{z}^t))_i, & if\ \ \boldsymbol{z}_i^t > 0\ \ or\ \ (\nabla G(\boldsymbol{z}^t))_i < 0 \\ 0, & otherwise. \end{cases} \tag{V.7}$$

Problem V.6 can be addressed with the close-form solution

$$\sigma^t = \frac{(g^t)^T(g^t)}{(g^t)^T A(g^t)} \tag{V.8}$$

Furthermore, the basic GPSR algorithm employs the backtracking linear search method [31] to ensure that the step size of gradient descent, in each iteration, is a more proper value. The stop condition of the backtracking linear search should satisfy

$$\begin{aligned}G((\boldsymbol{z}^t - \sigma^t\nabla G(\boldsymbol{z}^t))_+) > G(\boldsymbol{z}^t) - \beta\nabla G(\boldsymbol{z}^t)^T \\ (\boldsymbol{z}^t - (\boldsymbol{z}^t - \sigma^t\nabla G(\boldsymbol{z}^t))_+)\end{aligned} \tag{V.9}$$

where β is a small constant. The main steps of GPSR are summarized in Algorithm 2. For more detailed information, one can refer to the literature [73].

Algorithm 2. Gradient Projection Sparse Reconstruction (GPSR)
Task: To address the unconstraint problem:
$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} \frac{1}{2}\|\boldsymbol{y} - X\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1$

Input: Probe sample y, the measurement matrix X, small constant λ
Initialization: $t = 0$, $\beta \in (0, 0.5)$, $\gamma \in (0, 1)$, given $\boldsymbol{\alpha}$ so that $\boldsymbol{z} = [\boldsymbol{\alpha}_+, \boldsymbol{\alpha}_-]$.
While not converged do
Step 1: Compute σ^t exploiting Eq. V.8 and $\sigma^t \leftarrow mid(\sigma_{min}, \sigma^t, \sigma_{max})$, where $mid(\cdot,\cdot,\cdot)$ denotes the middle value of the three parameters.
Step 2: While Eq. V.9 not satisfied do $\sigma^t \leftarrow \gamma\sigma^t$ end
Step 3: $\boldsymbol{z}^{t+1} = (\boldsymbol{z}^t - \sigma^t\nabla G(\boldsymbol{z}^t))_+$ and $t = t + 1$.
End
Output: $\boldsymbol{z}^{t+1}$, $\boldsymbol{\alpha}$

B. Interior-point method based sparse representation strategy

The Interior-point method [31] is not an iterative algorithm but a smooth mathematic model and it always incorporates the Newton method to efficiently solve unconstrained smooth problems of modest size [28]. When the Newton method is used to address the optimization issue, a complex Newton equation should be solved iteratively which is very time-consuming. A method named the truncated Newton method can effectively and efficiently obtain the solution of the Newton equation. A prominent algorithm called the truncated Newton based interior-point method (TNIPM) exists, which can be utilized to solve the large-scale l_1-regularized least squares (i.e. $l_1_l_s$) problem [74].

The original problem of $l_1_l_s$ is to solve problem III.12 and the core procedures of $l_1_l_s$ are shown below:
(1) Transform the original unconstrained non-smooth problem to a constrained smooth optimization problem.
(2) Apply the interior-point method to reformulate the constrained smooth optimization problem as a new unconstrained smooth optimization problem.
(3) Employ the truncated Newton method to solve this unconstrained smooth problem.

The main idea of the $l_1_l_s$ will be briefly described. For simplicity of presentation, the following one-dimensional problem is used as an example.

$$|\alpha| = \arg\min_{-\sigma \leq \alpha \leq \sigma} \sigma \tag{V.10}$$

where σ is a proper positive constant.

Thus, problem III.12 can be rewritten as

$$\begin{aligned}\hat{\boldsymbol{\alpha}} &= \arg\min \tfrac{1}{2}\|\boldsymbol{y}-X\boldsymbol{\alpha}\|_2^2+\lambda\|\boldsymbol{\alpha}\|_1 \\ &= \arg\min \tfrac{1}{2}\|\boldsymbol{y}-X\boldsymbol{\alpha}\|_2^2+\lambda\textstyle\sum_{i=1}^N \min_{-\boldsymbol{\sigma}_i\le\boldsymbol{\alpha}_i\le\boldsymbol{\sigma}_i}\boldsymbol{\sigma}_i \\ &= \arg\min \tfrac{1}{2}\|\boldsymbol{y}-X\boldsymbol{\alpha}\|_2^2+\lambda\min_{-\boldsymbol{\sigma}_i\le\boldsymbol{\alpha}_i\le\boldsymbol{\sigma}_i}\textstyle\sum_{i=1}^N\boldsymbol{\sigma}_i \\ &= \arg\min_{-\boldsymbol{\sigma}_i\le\boldsymbol{\alpha}_i\le\boldsymbol{\sigma}_i} \tfrac{1}{2}\|\boldsymbol{y}-X\boldsymbol{\alpha}\|_2^2+\lambda\textstyle\sum_{i=1}^N\boldsymbol{\sigma}_i\end{aligned} \tag{V.11}$$

Thus problem III.12 is also equivalent to solve the following problem:

$$\hat{\boldsymbol{\alpha}}=\arg\min_{\boldsymbol{\alpha},\boldsymbol{\sigma}\in\mathbb{R}^N}\frac{1}{2}\|\boldsymbol{y}-X\boldsymbol{\alpha}\|_2^2+\lambda\sum_{i=1}^N\boldsymbol{\sigma}_i\ s.t.\ -\boldsymbol{\sigma}_i\le\boldsymbol{\alpha}_i\le\boldsymbol{\sigma}_i \tag{V.12}$$

or

$$\begin{aligned}\hat{\boldsymbol{\alpha}}=\arg\min_{\boldsymbol{\alpha},\boldsymbol{\sigma}\in\mathbb{R}^N}\frac{1}{2}\|\boldsymbol{y}-X\boldsymbol{\alpha}\|_2^2+\lambda\sum_{i=1}^N\boldsymbol{\sigma}_i \\ s.t.\quad \boldsymbol{\sigma}_i+\boldsymbol{\alpha}_i\ge 0,\ \boldsymbol{\sigma}_i-\boldsymbol{\alpha}_i\ge 0\end{aligned} \tag{V.13}$$

The interior-point strategy can be used to transform problem V.13 into an unconstrained smooth problem

$$\hat{\boldsymbol{\alpha}}=\arg\min_{\boldsymbol{\alpha},\boldsymbol{\sigma}\in\mathbb{R}^N}G(\boldsymbol{\alpha},\boldsymbol{\sigma})=\frac{v}{2}\|\boldsymbol{y}-X\boldsymbol{\alpha}\|_2^2+\lambda v\sum_{i=1}^N\boldsymbol{\sigma}_i-B(\boldsymbol{\alpha},\boldsymbol{\sigma}) \tag{V.14}$$

where $B(\boldsymbol{\alpha},\boldsymbol{\sigma})=\sum_{i=1}^N log(\boldsymbol{\sigma}_i+\boldsymbol{\alpha}_i)+\sum_{i=1}^N log(\boldsymbol{\sigma}_i-\boldsymbol{\alpha}_i)$ is a barrier function, which forces the algorithm to be performed within the feasible region in the context of unconstrained condition.

Subsequently, $l_1_l_s$ utilizes the truncated Newton method to solve problem V.14. The main procedures of addressing problem V.14 are presented as follows:
First, the Newton system is constructed

$$H\begin{bmatrix}\triangle\boldsymbol{\alpha}\\ \triangle\boldsymbol{\sigma}\end{bmatrix}=-\nabla G(\boldsymbol{\alpha},\boldsymbol{\sigma})\in\mathbb{R}^{2N} \tag{V.15}$$

where $H=-\nabla^2G(\boldsymbol{\alpha},\boldsymbol{\sigma})\in\mathbb{R}^{2N\times 2N}$ is the Hessian matrix, which is computed using the preconditioned conjugate gradient algorithm, and then the direction of linear search $[\triangle\boldsymbol{\alpha},\triangle\boldsymbol{\sigma}]$ is obtained.
Second, the Lagrange dual of problem III.12 is used to construct the dual feasible point and duality gap:
a) The Lagrangian function and Lagrange dual of problem III.12 are constructed. The Lagrangian function is reformulated as

$$L(\boldsymbol{\alpha},\boldsymbol{z},\boldsymbol{u})=\boldsymbol{z}^T\boldsymbol{z}+\lambda\|\boldsymbol{\alpha}\|_1+u(X\boldsymbol{\alpha}-\boldsymbol{y}-\boldsymbol{z}) \tag{V.16}$$

where its corresponding Lagrange dual function is

$$\begin{aligned}\hat{\boldsymbol{\alpha}}=\arg\max F(\boldsymbol{u})=-\frac{1}{4}\boldsymbol{u}^T\boldsymbol{u}-\boldsymbol{u}^T\boldsymbol{y}\quad s.t. \\ |(X^T\boldsymbol{u})_i|\le\lambda_i\ (i=1,2,\cdots,N)\end{aligned} \tag{V.17}$$

b) A dual feasible point is constructed

$$\boldsymbol{u}=2s(\boldsymbol{y}-X\boldsymbol{\alpha}),\ s=\min\{\lambda/|2\boldsymbol{y}_i-2(X^TX\boldsymbol{\alpha})_i|\}\forall i \tag{V.18}$$

where u is a dual feasible point and s is the step size of the linear search.

c) The duality gap is constructed, which is the gap between the primary problem and the dual problem:

$$g=\|\boldsymbol{y}-X\boldsymbol{\alpha}\|+\lambda\|\boldsymbol{\alpha}\|_1-F(\boldsymbol{u}) \tag{V.19}$$

Third, the method of backtracking linear search is used to determine an optimal step size of the Newton linear search. The stopping condition of the backtracking linear search is

$$G(\boldsymbol{\alpha}+\eta^t\triangle\boldsymbol{\alpha},\boldsymbol{\sigma}+\eta^t\triangle\boldsymbol{\sigma})>G(\boldsymbol{\alpha},\boldsymbol{\sigma})+\rho\eta^t\nabla G(\boldsymbol{\alpha},\boldsymbol{\sigma})[\triangle\boldsymbol{\alpha},\triangle\boldsymbol{\sigma}] \tag{V.20}$$

where $\rho\in(0,0.5)$ and $\eta^t\in(0,1)$ is the step size of the Newton linear search.
Finally, the termination condition of the Newton linear search is set to

$$\zeta=\min\{0.1,\beta g/\|h\|_2\} \tag{V.21}$$

where the function $h=\nabla G(\boldsymbol{\alpha},\boldsymbol{\sigma})$, β is a small constant, and g is the duality gap. The main steps of algorithm $l_1_l_s$ are summarized in Algorithm 3. For further description and analyses, please refer to the literature [74].

Algorithm 3. Truncated Newton based interior-point method (TNIPM) for $l_1_l_s$
Task: To address the unconstraint problem:
$\hat{\boldsymbol{\alpha}}=\arg\min_{\boldsymbol{\alpha}}\frac{1}{2}\|\boldsymbol{y}-X\boldsymbol{\alpha}\|_2^2+\lambda\|\boldsymbol{\alpha}\|_1$

Input: Probe sample $\boldsymbol{y}$, the measurement matrix X, small constant λ
Initialization: $t=1$, $v=\frac{1}{\lambda}$, $\rho\in(0,0.5)$, $\sigma=\mathbf{1}_N$
Step 1: Employ preconditioned conjugate gradient algorithm to obtain the approximation of H in Eq. V.15, and then obtain the descent direction of linear search $[\triangle\boldsymbol{\alpha}^t,\triangle\boldsymbol{\sigma}^t]$.
Step 2: Exploit the algorithm of backtracking linear search to find the optimal step size of Newton linear search η^t, which satisfies the Eq. V.20.
Step 3: Update the iteration point utilizing $(\boldsymbol{\alpha}^{t+1},\boldsymbol{\sigma}^{t+1})=(\boldsymbol{\alpha}^t,\boldsymbol{\sigma}^t)+(\triangle\boldsymbol{\alpha}^t+\triangle\boldsymbol{\sigma}^t)$.
Step 4: Construct feasible point using eq. V.18 and duality gap in Eq. V.19, and compute the termination tolerance ζ in Eq. V.21.
Step 5: If the condition $g/F(\boldsymbol{u})>\zeta$ is satisfied, stop; Otherwise, return to step 1, update v in Eq. V.14 and $t=t+1$.
Output: $\boldsymbol{\alpha}$

The truncated Newton based interior-point method (TNIPM) [75] is a very effective method to solve the l_1-norm regularization problems. Koh et al. [76] also utilized the TNIPM to solve large scale logistic regression problems, which employed a preconditioned conjugate gradient method to compute the search step size with warm-start techniques. Mehrotra proposed to exploit the interior-point method to address the primal-dual problem [77] and introduced the second-order derivation of Taylor polynomial to approximate a primal-dual trajectory. More analyses of interior-point method for sparse representation can be found in the literature [78].

C. Alternating direction method (ADM) based sparse representation strategy

This section shows how the ADM [43] is used to solve primal and dual problems in III.12. First, an auxiliary variable is introduced to convert problem in III.12 into a constrained problem with the form of problem V.22. Subsequently, the alternative direction method is used to efficiently address the sub-problems of problem V.22. By introducing the auxiliary

term $\boldsymbol{s} \in \mathbb{R}^d$, problem III.12 is equivalent to a constrained problem

$$\arg\min_{\boldsymbol{\alpha},s} \frac{1}{2\tau}\|\boldsymbol{s}\|_2 + \|\boldsymbol{\alpha}\|_1 \quad s.t. \quad \boldsymbol{s} = \boldsymbol{y} - X\boldsymbol{\alpha} \tag{V.22}$$

The optimization problem of the augmented Lagrangian function of problem V.22 is considered

$$\begin{aligned}\arg\min_{\boldsymbol{\alpha},\boldsymbol{s},\lambda} L(\boldsymbol{\alpha},\boldsymbol{s},\boldsymbol{\lambda}) = \frac{1}{2\tau}\|\boldsymbol{s}\|_2 + \|\boldsymbol{\alpha}\|_1 - \lambda^T(\boldsymbol{s}+ \\ X\boldsymbol{\alpha} - y) + \frac{\mu}{2}\|\boldsymbol{s} + X\boldsymbol{\alpha} - \boldsymbol{y}\|_2^2\end{aligned} \tag{V.23}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^d$ is a Lagrange multiplier vector and μ is a penalty parameter. The general framework of ADM is used to solve problem V.23 as follows:

$$\begin{cases} \boldsymbol{s}^{t+1} &= \arg\min L(\boldsymbol{s},\boldsymbol{\alpha}^t,\boldsymbol{\lambda}^t) & (a) \\ \boldsymbol{\alpha}^{t+1} &= \arg\min L(\boldsymbol{s}^{t+1},\boldsymbol{\alpha},\boldsymbol{\lambda}^t) & (b) \\ \boldsymbol{\lambda}^{t+1} &= \boldsymbol{\lambda}^t - \mu(\boldsymbol{s}^{t+1} + X\boldsymbol{\alpha}^{t+1} - \boldsymbol{y}) & (c) \end{cases} \tag{V.24}$$

First, the first optimization problem V.24(a) is considered

$$\begin{aligned}\arg\min L(\boldsymbol{s},\boldsymbol{\alpha}^t,\boldsymbol{\lambda}^t) =& \frac{1}{2\tau}\|\boldsymbol{s}\|_2 + \|\boldsymbol{\alpha}^t\|_1 - (\boldsymbol{\lambda}^t)^T(\boldsymbol{s} + X\boldsymbol{\alpha}^t \\ &- \boldsymbol{y}) + \frac{\mu}{2}\|\boldsymbol{s} + X\boldsymbol{\alpha}^t - \boldsymbol{y}\|_2^2 \\ =& \frac{1}{2\tau}\|\boldsymbol{s}\|_2 - (\boldsymbol{\lambda}^t)^T\boldsymbol{s} + \frac{\mu}{2}\|\boldsymbol{s} + X\boldsymbol{\alpha}^t - \boldsymbol{y}\|_2^2 + \\ &\|\boldsymbol{\alpha}^t\|_1 - (\boldsymbol{\lambda}^t)^T(X\boldsymbol{\alpha}^t - \boldsymbol{y})\end{aligned} \tag{V.25}$$

Then, it is known that the solution of problem V.25 with respect to s is given by

$$\boldsymbol{s}^{t+1} = \frac{\tau}{1+\mu\tau}(\boldsymbol{\lambda}^t - \mu(\boldsymbol{y} - X\boldsymbol{\alpha}^t)) \tag{V.26}$$

Second, the optimization problem V.24(b) is considered

$$\begin{aligned}\arg\min L(\boldsymbol{s}^{t+1},\boldsymbol{\alpha},\lambda^t) = \frac{1}{2\tau}\|\boldsymbol{s}^{t+1}\|_2 + \|\boldsymbol{\alpha}\|_1 - (\boldsymbol{\lambda})^T(\boldsymbol{s}^{t+1} \\ + X\boldsymbol{\alpha} - \boldsymbol{y}) + \frac{\mu}{2}\|\boldsymbol{s}^{t+1} + X\boldsymbol{\alpha} - \boldsymbol{y}\|_2^2\end{aligned}$$

which is equivalent to

$$\begin{aligned}&\arg\min\{\|\boldsymbol{\alpha}\|_1 - (\boldsymbol{\lambda}^t)^T(\boldsymbol{s}^{t+1} + X\boldsymbol{\alpha} - \boldsymbol{y}) + \frac{\mu}{2}\|\boldsymbol{s}^{t+1} + \\ &\qquad X\boldsymbol{\alpha} - \boldsymbol{y}\|_2^2\} \\ &= \|\boldsymbol{\alpha}\|_1 + \frac{\mu}{2}\|\boldsymbol{s}^{t+1} + X\boldsymbol{\alpha} - \boldsymbol{y} - \boldsymbol{\lambda}^t/\mu\|_2^2 \\ &= \|\boldsymbol{\alpha}\|_1 + f(\boldsymbol{\alpha})\end{aligned} \tag{V.27}$$

where $f(\boldsymbol{\alpha}) = \frac{\mu}{2}\|\boldsymbol{s}^{t+1} + X\boldsymbol{\alpha} - \boldsymbol{y} - \boldsymbol{\lambda}^t/\mu\|_2^2$. If the second order Taylor expansion is used to approximate $f(\boldsymbol{\alpha})$, the problem V.27 can be approximately reformulated as

$$\begin{aligned}\arg\min\{\|\boldsymbol{\alpha}\|_1 + (\boldsymbol{\alpha} - \boldsymbol{\alpha}^t)^T X^T(\boldsymbol{s}^{t+1} + X\boldsymbol{\alpha}^t - \boldsymbol{y} - \boldsymbol{\lambda}^t/\mu) \\ + \frac{1}{2\tau}\|\boldsymbol{\alpha} - \boldsymbol{\alpha}^t\|_2^2\}\end{aligned} \tag{V.28}$$

where τ is a proximal parameter. The solution of problem V.28 can be obtained by the soft thresholding operator

$$\boldsymbol{\alpha}^{t+1} = soft\{\boldsymbol{\alpha}^t - \tau X^T(\boldsymbol{s}^{t+1} + X\boldsymbol{\alpha}^t - \boldsymbol{y} - \boldsymbol{\lambda}^t/\mu), \frac{\tau}{\mu}\} \tag{V.29}$$

where $soft(\sigma,\eta) = sign(\sigma)\max\{|\sigma| - \eta, 0\}$.
Finally, the Lagrange multiplier vector $\boldsymbol{\lambda}$ is updated by using Eq. V.24(c).

The algorithm presented above utilizes the second order Taylor expansion to approximately solve the sub-problem V.27 and thus the algorithm is denoted as an inexact ADM or approximate ADM. The main procedures of the inexact ADM based sparse representation method are summarized in Algorithm 4. More specifically, the inexact ADM described above is to reformulate the unconstrained problem as a constrained problem, and then utilizes the alternative strategy to effectively address the corresponding sub-optimization problem. Moreover, ADM can also efficiently solve the dual problems of the primal problems III.9-III.12. For more information, please refer to the literature [43, 79].

Algorithm 4. Alternating direction method (ADM) based sparse representation strategy
Task: To address the unconstraint problem:
$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} \frac{1}{2}\|\boldsymbol{y} - X\boldsymbol{\alpha}\|_2^2 + \tau\|\boldsymbol{\alpha}\|_1$

Input: Probe sample y, the measurement matrix X, small constant λ
Initialization: $t = 0$, $\boldsymbol{s}^0 = \boldsymbol{0}$, $\boldsymbol{\alpha}^0 = \boldsymbol{0}$, $\boldsymbol{\lambda}^0 = \boldsymbol{0}$, $\tau = 1.01$, μ is a small constant.
Step 1: Construct the constraint optimization problem of problem III.12 by introducing the auxiliary parameter and its augmented Lagrangian function, i.e. problem (V.22) and (V.23).
While not converged do
  Step 2: Update the value of the $\boldsymbol{s}^{t+1}$ by using Eq. (V.25).
  Step 2: Update the value of the $\boldsymbol{\alpha}^{t+1}$ by using Eq. (V.29).
  Step 3: Update the value of the $\boldsymbol{\lambda}^{t+1}$ by using Eq. (V.24(c)).
  Step 4: $\mu^{t+1} = \tau\mu^t$ and $t = t + 1$.
End While
Output: $\boldsymbol{\alpha}^{t+1}$

VI. Proximity algorithm based optimization strategy

In this section, the methods that exploit the proximity algorithm to solve constrained convex optimization problems are discussed. The core idea of the proximity algorithm is to utilize the proximal operator to iteratively solve the sub-problem, which is much more computationally efficient than the original problem. The proximity algorithm is frequently employed to solve nonsmooth, constrained convex optimization problems [28]. Furthermore, the general problem of sparse representation with l_1-norm regularization is a nonsmooth convex optimization problem, which can be effectively addressed by using the proximal algorithm.

Suppose a simple constrained optimization problem is

$$\min\{h(\boldsymbol{x})|\boldsymbol{x} \in \chi\} \tag{VI.1}$$

where $\chi \subset \mathbb{R}^n$. The general framework of addressing the constrained convex optimization problem VI.1 using the proximal algorithm can be reformulated as

$$\tilde{\boldsymbol{x}}^t = \arg\min\{h(\boldsymbol{x}) + \frac{\tau}{2}\|\boldsymbol{x} - \boldsymbol{x}^t\|^2|\boldsymbol{x} \in \chi\} \tag{VI.2}$$

where τ and $\boldsymbol{x}^t$ are given. For definiteness and without loss of generality, it is assumed that there is the following linear constrained convex optimization problem

$$\arg\min\{F(\boldsymbol{x}) + G(\boldsymbol{x})|\boldsymbol{x} \in \chi\} \tag{VI.3}$$

The solution of problem VI.3 obtained by employing the proximity algorithm is:

$$\begin{aligned}\boldsymbol{x}^{t+1} &= \arg\min\{F(\boldsymbol{x}) + \langle \nabla G(\boldsymbol{x}^t), \boldsymbol{x} - \boldsymbol{x}^t\rangle + \frac{1}{2\tau}\|\boldsymbol{x} - \boldsymbol{x}^t\|^2\} \\ &= \arg\min\{F(\boldsymbol{x}) + \frac{1}{2\tau}\|\boldsymbol{x} - \boldsymbol{\theta}^t\|^2\}\end{aligned} \tag{VI.4}$$

where $\boldsymbol{\theta} = \boldsymbol{x}^t - \tau\nabla G(\boldsymbol{x}^t)$. More specifically, for the sparse representation problem with l_1-norm regularization, the main problem can be reformulated as:

$$\begin{aligned} &\min P(\boldsymbol{\alpha}) = \{\lambda\|\boldsymbol{\alpha}\|_1 \mid A\boldsymbol{\alpha} = \boldsymbol{y}\} \\ or \quad &\min P(\boldsymbol{\alpha}) = \{\lambda\|\boldsymbol{\alpha}\|_1 + \|A\boldsymbol{\alpha} - \boldsymbol{y}\|_2^2 \mid \boldsymbol{\alpha} \in \mathbb{R}^n\}\end{aligned} \tag{VI.5}$$

which are considered as the constrained sparse representation of problem III.12.

A. Soft thresholding or shrinkage operator

First, a simple form of problem III.12 is introduced, which has a closed-form solution, and it is formulated as:

$$\begin{aligned}\boldsymbol{\alpha}^* = \min_{\boldsymbol{\alpha}} h(\boldsymbol{\alpha}) =& \lambda\|\boldsymbol{\alpha}\|_1 + \frac{1}{2}\|\boldsymbol{\alpha} - s\|^2 \\ =& \sum_{j=1}^{N}\lambda|\boldsymbol{\alpha}_j| + \sum_{j=1}^{N}\frac{1}{2}(\boldsymbol{\alpha}_j - s_j)^2\end{aligned} \tag{VI.6}$$

where $\boldsymbol{\alpha}^*$ is the optimal solution of problem VI.6, and then there are the following conclusions:
(1) if $\boldsymbol{\alpha}_j > 0$, then $h(\boldsymbol{\alpha}) = \lambda\boldsymbol{\alpha} + \frac{1}{2}\|\boldsymbol{\alpha} - s\|^2$ and its derivative is $h'(\boldsymbol{\alpha}_j) = \lambda + \boldsymbol{\alpha}_j^* - s_j$.
Let $h'(\boldsymbol{\alpha}_j) = 0 \Rightarrow \boldsymbol{\alpha}_j^* = s_j - \lambda$, where it indicates $s_j > \lambda$;
(2) if $\boldsymbol{\alpha}_j < 0$, then $h(\boldsymbol{\alpha}) = -\lambda\boldsymbol{\alpha} + \frac{1}{2}\|\boldsymbol{\alpha} - s\|^2$ and its derivative is $h'(\boldsymbol{\alpha}_j) = -\lambda + \boldsymbol{\alpha}_j^* - s_j$.
Let $h'(\boldsymbol{\alpha}_j) = 0 \Rightarrow \boldsymbol{\alpha}_j^* = s_j + \lambda$, where it indicates $s_j < -\lambda$;
(3) if $-\lambda \le s_j \le \lambda$, and then $\boldsymbol{\alpha}_j^* = 0$.
So the solution of problem VI.6 is summarized as

$$\boldsymbol{\alpha}_j^* = \begin{cases} s_j - \lambda, & if \quad s_j > \lambda \\ s_j + \lambda, & if \quad s_j < -\lambda \\ 0, & otherwise \end{cases} \tag{VI.7}$$

The equivalent expression of the solution is $\boldsymbol{\alpha}^* = shrink(s, \lambda)$, where the j-th component of $shrink(s, \lambda)$ is $shrink(s, \lambda)_j = sign(s_j)\max\{|s_j| - \lambda, 0\}$. The operator $shrink(\bullet)$ can be regarded as a proximal operator.

B. Iterative shrinkage thresholding algorithm (ISTA)

The objective function of ISTA [80] has the form of

$$\arg\min F(\boldsymbol{\alpha}) = \frac{1}{2}\|X\boldsymbol{\alpha} - \boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1 = f(\boldsymbol{\alpha}) + \lambda g(\boldsymbol{\alpha}) \tag{VI.8}$$

and is usually difficult to solve. Problem VI.8 can be converted to the form of an easy problem VI.6 and the explicit procedures are presented as follows.

First, Taylor expansion is used to approximate $f(\boldsymbol{\alpha}) = \frac{1}{2}\|X\boldsymbol{\alpha} - \boldsymbol{y}\|_2^2$ at a point of $\boldsymbol{\alpha}^t$. The second order Taylor expansion is

$$\begin{aligned} f(\boldsymbol{\alpha}) = f(\boldsymbol{\alpha}^t) + (\boldsymbol{\alpha} - \boldsymbol{\alpha}^t)^T\nabla f(\boldsymbol{\alpha}^t) + \frac{1}{2}(\boldsymbol{\alpha} - \boldsymbol{\alpha}^t)^T \\ H_f(\boldsymbol{\alpha}^t)(\boldsymbol{\alpha} - \boldsymbol{\alpha}^t) + \cdots \end{aligned} \tag{VI.9}$$

where $H_f(\boldsymbol{\alpha}^t)$ is the Hessian matrix of $f(\boldsymbol{\alpha})$ at $\boldsymbol{\alpha}^t$. For the function $f(\boldsymbol{\alpha})$, $\nabla f(\boldsymbol{\alpha}) = X^T(X\boldsymbol{\alpha} - \boldsymbol{y})$ and $H_f(\boldsymbol{\alpha}) = X^TX$ can be obtained.

$$\begin{aligned} f(\boldsymbol{\alpha}) = \frac{1}{2}\|X\boldsymbol{\alpha}^t - \boldsymbol{y}\|_2^2 + (\boldsymbol{\alpha} - \boldsymbol{\alpha}^t)^TX^T(X\boldsymbol{\alpha}^t - \boldsymbol{y}) + \\ \frac{1}{2}(\boldsymbol{\alpha} - \boldsymbol{\alpha}^t)^TX^TX(\boldsymbol{\alpha} - \boldsymbol{\alpha}^t) \end{aligned} \tag{VI.10}$$

If the Hessian matrix $H_f(\boldsymbol{\alpha})$ is replaced or approximated in the third term above by using a scalar $\frac{1}{\tau}I$, and then

$$\begin{aligned} f(\boldsymbol{\alpha}) \approx \frac{1}{2}\|X\boldsymbol{\alpha}^t - \boldsymbol{y}\|_2^2 + (\boldsymbol{\alpha} - \boldsymbol{\alpha}^t)^TX^T(X\boldsymbol{\alpha}^t - \boldsymbol{y}) \\ + \frac{1}{2\tau}(\boldsymbol{\alpha} - \boldsymbol{\alpha}^t)^T(\boldsymbol{\alpha} - \boldsymbol{\alpha}^t) = Q_t(\boldsymbol{\alpha}, \boldsymbol{\alpha}^t) \end{aligned} \tag{VI.11}$$

Thus problem VI.8 using the proximal algorithm can be successively addressed by

$$\boldsymbol{\alpha}^{t+1} = \arg\min Q_t(\boldsymbol{\alpha}, \boldsymbol{\alpha}^t) + \lambda\|\boldsymbol{\alpha}\|_1 \tag{VI.12}$$

Problem VI.12 is reformulated to a simple form of problem VI.6 by

$$\begin{aligned} Q_t(\boldsymbol{\alpha}, \boldsymbol{\alpha}^t) &= \frac{1}{2}\|X\boldsymbol{\alpha}^t - \boldsymbol{y}\|_2^2 + (\boldsymbol{\alpha} - \boldsymbol{\alpha}^t)^TX^T(X\boldsymbol{\alpha}^t - \boldsymbol{y}) + \\ &\qquad \frac{1}{2\tau}\|\boldsymbol{\alpha} - \boldsymbol{\alpha}^t\|_2^2 \\ &= \frac{1}{2}\|X\boldsymbol{\alpha}^t - \boldsymbol{y}\|_2^2 + \frac{1}{2\tau}\|\boldsymbol{\alpha} - \boldsymbol{\alpha}^t + \tau X^T(X\boldsymbol{\alpha}^t - \boldsymbol{y})\|_2^2 \\ &\qquad - \frac{\tau}{2}\|X^T(X\boldsymbol{\alpha}^t - \boldsymbol{y})\|_2^2 \\ &= \frac{1}{2\tau}\|\boldsymbol{\alpha} - (\boldsymbol{\alpha}^t - \tau X^T(X\boldsymbol{\alpha}^t - \boldsymbol{y}))\|_2^2 + B(\boldsymbol{\alpha}^t) \end{aligned} \tag{VI.13}$$

where the term $B(\boldsymbol{\alpha}^t) = \frac{1}{2}\|X\boldsymbol{\alpha}^t - \boldsymbol{y}\|_2^2 - \frac{\tau}{2}\|X^T(X\boldsymbol{\alpha}^t - \boldsymbol{y})\|^2$ in problem VI.12 is a constant with respect to variable $\boldsymbol{\alpha}$, and it can be omitted. As a result, problem VI.12 is equivalent to the following problem:

$$\boldsymbol{\alpha}^{t+1} = \arg\min \frac{1}{2\tau}\|\boldsymbol{\alpha} - \theta(\boldsymbol{\alpha}^t)\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1 \tag{VI.14}$$

where $\theta(\boldsymbol{\alpha}^t) = \boldsymbol{\alpha}^t - \tau X^T(X\boldsymbol{\alpha}^t - \boldsymbol{y})$.

The solution of the simple problem VI.6 is applied to solve problem VI.14 where the parameter t is replaced by the equation $\boldsymbol{\theta}(\boldsymbol{\alpha}^t)$, and the solution of problem VI.14 is $\boldsymbol{\alpha}^{t+1} = shrink(\boldsymbol{\theta}(\boldsymbol{\alpha}^t), \lambda\tau)$. Thus, the solution of ISTA is reached. The techniques used here are called linearization or preconditioning and more detailed information can be found in the literature [80, 81].

C. Fast Iterative shrinkage thresholding algorithm (FISTA)

The fast iterative shrinkage thresholding algorithm (FISTA) is an improvement of ISTA. FISTA [82] not only preserves the efficiency of the original ISTA but also promotes the effectiveness of ISTA so that FISTA can obtain global convergence.

Considering that the Hessian matrix $H_f(\boldsymbol{\alpha})$ is approximated by using a scalar $\frac{1}{\tau}I$ for ISTA in Eq. VI.9, FISTA utilizes the minimum Lipschitz constant of the gradient $\nabla f(\boldsymbol{\alpha})$ to approximate the Hessian matrix of $f(\boldsymbol{\alpha})$, i.e. $L(f) = 2\lambda_{max}(X^TX)$. Thus, the problem VI.8 can be converted to the problem below:

$$f(\boldsymbol{\alpha}) \approx \frac{1}{2}\|X\boldsymbol{\alpha}^t - \boldsymbol{y}\|_2^2 + (\boldsymbol{\alpha} - \boldsymbol{\alpha}^t)^T X^T (X\boldsymbol{\alpha}^t - \boldsymbol{y}) + \frac{L}{2}(\boldsymbol{\alpha} - \boldsymbol{\alpha}^t)^T(\boldsymbol{\alpha} - \boldsymbol{\alpha}^t) = P_t(\boldsymbol{\alpha}, \boldsymbol{\alpha}^t) \tag{VI.15}$$

where the solution can be reformulated as

$$\boldsymbol{\alpha}^{t+1} = \arg\min \frac{L}{2}\|\boldsymbol{\alpha} - \theta(\boldsymbol{\alpha}^t)\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1 \tag{VI.16}$$

where $\theta(\boldsymbol{\alpha}^t) = \boldsymbol{\alpha}^t - \frac{1}{L}X^T(X\boldsymbol{\alpha}^t - \boldsymbol{y})$.

Moreover, to accelerate the convergence of the algorithm, FISTA also improves the sequence of iteration points, instead of employing the previous point it utilizes a specific linear combinations of the previous two points $\{\boldsymbol{\alpha}^t, \boldsymbol{\alpha}^{t-1}\}$, i.e.

$$\boldsymbol{\alpha}^t = \boldsymbol{\alpha}^t + \frac{\mu^t - 1}{\mu^{t+1}}(\boldsymbol{\alpha}^t - \boldsymbol{\alpha}^{t-1}) \tag{VI.17}$$

where μ^t is a positive sequence, which satisfies $\mu^t \geq (t+1)/2$, and the main steps of FISTA are summarized in Algorithm 5. The backtracking linear research strategy can also be utilized to explore a more feasible value of L and more detailed analyses on FISTA can be found in the literature [82, 83].

Algorithm 5. Fast Iterative shrinkage thresholding algorithm (FISTA)
Task: To address the problem $\hat{\boldsymbol{\alpha}} = \arg\min F(\boldsymbol{\alpha}) = \frac{1}{2}\|X\boldsymbol{\alpha} - \boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1$

Input: Probe sample $\boldsymbol{y}$, the measurement matrix X, small constant λ
Initialization: $t = 0$, $\mu^0 = 1$, $L = 2\Lambda_{max}(X^TX)$, i.e. Lipschitz constant of ∇f.
While not converged do
  Step 1: Exploit the shrinkage operator in equation VI.7 to solve problem VI.16.
  Step 2: Update the value of μ using $\mu^{t+1} = \frac{1+\sqrt{1+4(\mu^t)^2}}{2}$.
  Step 3: Update iteration sequence $\boldsymbol{\alpha}^t$ using equation VI.17.
End
Output: $\boldsymbol{\alpha}$

D. Sparse reconstruction by separable approximation (SpaRSA)

Sparse reconstruction by separable approximation (SpaRSA) [84] is another typical proximity algorithm based on sparse representation, which can be viewed as an accelerated version of ISTA. SpaRSA provides a general algorithmic framework for solving the sparse representation problem and here a simple specific SpaRSA with adaptive continuation on ISTA is introduced. The main contributions of SpaRSA are trying to optimize the parameter λ in problem VI.8 by using the worm-starting technique, i.e. continuation, and choosing a more reliable approximation of $H_f(\boldsymbol{\alpha})$ in problem VI.9 using the Barzilai-Borwein (BB) spectral method [85]. The worm-starting technique and BB spectral approach are introduced as follows.

(1) Utilizing the worm-starting technique to optimize λ

The values of λ in the sparse representation methods discussed above are always set to be a specific small constant. However, Hale et al. [86] concluded that the technique that exploits a decreasing value of λ from a warm-starting point can more efficiently solve the sub-problem VI.14 than ISTA that is a fixed point iteration scheme. SpaRSA uses an adaptive continuation technique to update the value of λ so that it can lead to the fastest convergence. The procedure regenerates the value of λ using

$$\lambda = \max\{\gamma\|X^T\boldsymbol{y}\|_\infty, \lambda\} \tag{VI.18}$$

where γ is a small constant.

(2) Utilizing the BB spectral method to approximate $H_f(\boldsymbol{\alpha})$

ISTA employs $\frac{1}{\tau}I$ to approximate the matrix $H_f(\boldsymbol{\alpha})$, which is the Hessian matrix of $f(\boldsymbol{\alpha})$ in problem VI.9 and FISTA exploits the Lipschitz constant of $\nabla f(\boldsymbol{\alpha})$ to replace $H_f(\boldsymbol{\alpha})$. However, SpaRSA utilizes the BB spectral method to choose the value of τ to mimic the Hessian matrix. The value of τ is required to satisfy the condition:

$$\frac{1}{\tau^{t+1}}(\boldsymbol{\alpha}^{t+1} - \boldsymbol{\alpha}^t) \approx \nabla f(\boldsymbol{\alpha}^{t+1}) - \nabla f(\boldsymbol{\alpha}^t) \tag{VI.19}$$

which satisfies the minimization problem

$$\frac{1}{\tau^{t+1}} = \arg\min \|\frac{1}{\tau}(\boldsymbol{\alpha}^{t+1} - \boldsymbol{\alpha}^t) - (\nabla f(\boldsymbol{\alpha}^{t+1}) - \nabla f(\boldsymbol{\alpha}^t))\|_2^2 = \frac{(\boldsymbol{\alpha}^{t+1} - \boldsymbol{\alpha}^t)^T(\nabla f(\boldsymbol{\alpha}^{t+1}) - \nabla f(\boldsymbol{\alpha}^t))}{(\boldsymbol{\alpha}^{t+1} - \boldsymbol{\alpha}^t)^T(\boldsymbol{\alpha}^{t+1} - \boldsymbol{\alpha}^t)} \tag{VI.20}$$

For problem VI.14, SpaRSA requires that the value of λ is a decreasing sequence using the Eq. VI.18 and the value of τ should meet the condition of Eq. VI.20. The sparse reconstruction by separable approximation (SpaRSA) is summarized in Algorithm 6 and more information can be found in the literature [84].

Algorithm 6. Sparse reconstruction by separable approximation (SpaRSA)
Task: To address the problem
$$\hat{\boldsymbol{\alpha}} = \arg\min F(\boldsymbol{\alpha}) = \frac{1}{2}\|X\boldsymbol{\alpha} - \boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1$$

Input: Probe sample y, the measurement matrix X, small constant λ
Initialization: $t = 0$, $i = 0$, $\boldsymbol{y}^0 = \boldsymbol{y}$, $\frac{1}{\tau^0}I \approx H_f(\boldsymbol{\alpha}) = X^TX$, tolerance $\varepsilon = 10^{-5}$.
  Step 1: $\lambda_t = \max\{\gamma\|X^T\boldsymbol{y}^t\|_\infty, \lambda\}$.
  Step 2: Exploit shrinkage operator to solve problem VI.14, i.e. $\boldsymbol{\alpha}^{i+1} = shrink(\boldsymbol{\alpha}^i - \tau^i X^T(X^T\boldsymbol{\alpha}^t - \boldsymbol{y}), \lambda_t\tau^i)$.
  Step 3: Update the value of $\frac{1}{\tau^{i+1}}$ using the Eq. VI.20.
  Step 4: If $\frac{\|\boldsymbol{\alpha}^{i+1} - \boldsymbol{\alpha}^i\|}{\boldsymbol{\alpha}^i} \leq \varepsilon$, go to step 5; Otherwise, return to step 2 and $i = i + 1$.
  Step 5: $\boldsymbol{y}^{t+1} = \boldsymbol{y} - X\boldsymbol{\alpha}^{t+1}$.
  Step 6: If $\lambda_t = \lambda$, stop; Otherwise, return to step 1 and $t = t + 1$.
Output: $\boldsymbol{\alpha}^i$

E. $l_{1/2}$-norm regularization based sparse representation

Sparse representation with the l_p-norm (0<p<1) regularization leads to a nonconvex, nonsmooth, and non-Lipschitz optimization problem and its general forms are described as problems III.13 and III.14. The l_p-norm (0<p<1) regularization problem is always difficult to be efficiently addressed and it has also attracted wide interests from large numbers of research groups. However, the research group led by Zongben Xu summarizes the conclusion that the most impressive and representative algorithm of the l_p-norm (0<p<1) regularization is sparse representation with the $l_{1/2}$-norm regularization [87]. Moreover, they have proposed some effective methods to solve the $l_{1/2}$-norm regularization problem [60, 88].

In this section, a half proximal algorithm is introduced to solve the $l_{1/2}$-norm regularization problem [60], which matches the iterative shrinkage thresholding algorithm for the l_1-norm regularization discussed above and the iterative hard thresholding algorithm for the l_0-norm regularization. Sparse representation with the $l_{1/2}$-norm regularization is explicitly to solve the problem as follows:

$$\hat{\boldsymbol{\alpha}} = \arg\min\{F(\boldsymbol{\alpha}) = \|X\boldsymbol{\alpha} - \boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_{1/2}^{1/2}\} \quad \text{(VI.21)}$$

where the first-order optimality condition of $F(\boldsymbol{\alpha})$ on $\boldsymbol{\alpha}$ can be formulated as

$$\nabla F(\boldsymbol{\alpha}) = X^T(X\boldsymbol{\alpha} - \boldsymbol{y}) + \frac{\lambda}{2}\nabla(\|\boldsymbol{\alpha}\|_{1/2}^{1/2}) = 0 \quad \text{(VI.22)}$$

which admits the following equation:

$$X^T(\boldsymbol{y} - X\boldsymbol{\alpha}) = \frac{\lambda}{2}\nabla(\|\boldsymbol{\alpha}\|_{1/2}^{1/2}) \quad \text{(VI.23)}$$

where $\nabla(\|\boldsymbol{\alpha}\|_{1/2}^{1/2})$ denotes the gradient of the regularization term $\|\boldsymbol{\alpha}\|_{1/2}^{1/2}$. Subsequently, an equivalent transformation of Eq. VI.23 is made by multiplying a positive constant τ and adding a parameter $\boldsymbol{\alpha}$ to both sides. That is,

$$\boldsymbol{\alpha} + \tau X^T(\boldsymbol{y} - X\boldsymbol{\alpha}) = \boldsymbol{\alpha} + \tau\frac{\lambda}{2}\nabla(\|\boldsymbol{\alpha}\|_{1/2}^{1/2}) \quad \text{(VI.24)}$$

To this end, the resolvent operator [60] is introduced to compute the resolvent solution of the right part of Eq. VI.24, and the resolvent operator is defined as

$$R_{\lambda,\frac{1}{2}}(\bullet) = \left(I + \frac{\lambda\tau}{2}\nabla(\|\bullet\|_{1/2}^{1/2})\right)^{-1} \quad \text{(VI.25)}$$

which is very similar to the inverse function of the right part of Eq. VI.24. The resolvent operator is always satisfied no matter whether the resolvent solution of $\nabla(\|\bullet\|_{1/2}^{1/2})$ exists or not [60]. Applying the resolvent operator to solve problem VI.24

$$\begin{aligned}\boldsymbol{\alpha} &= (I + \frac{\lambda\tau}{2}\nabla(\|\bullet\|_{1/2}^{1/2}))^{-1}(\boldsymbol{\alpha} + \tau X^t(\boldsymbol{y} - X\boldsymbol{\alpha})) \\ &= R_{\lambda,1/2}(\boldsymbol{\alpha} + \tau X^T(\boldsymbol{y} - X\boldsymbol{\alpha}))\end{aligned} \quad \text{(VI.26)}$$

can be obtained which is well-defined. $\theta(\boldsymbol{\alpha}) = \boldsymbol{\alpha} + \tau X^T(\boldsymbol{y} - X\boldsymbol{\alpha})$ is denoted and the resolvent operator can be explicitly expressed as:

$$R_{\lambda,\frac{1}{2}}(\boldsymbol{x}) = (f_{\lambda,\frac{1}{2}}(\boldsymbol{x}_1), f_{\lambda,\frac{1}{2}}(\boldsymbol{x}_2), \cdots, f_{\lambda,\frac{1}{2}}(\boldsymbol{x}_N))^T \quad \text{(VI.27)}$$

where

$$\begin{aligned}f_{\lambda,\frac{1}{2}}(\boldsymbol{x}_i) &= \frac{2}{3}\boldsymbol{x}_i(1 + \cos(\frac{2\pi}{3} - \frac{2}{3}g_\lambda(\boldsymbol{x}_i)), \\ g_\lambda(\boldsymbol{x}_i) &= \arg\cos(\frac{\lambda}{8}(\frac{|\boldsymbol{x}_i|}{3})^{-\frac{3}{2}})\end{aligned} \quad \text{(VI.28)}$$

which have been demonstrated in the literature [60].

Thus the half proximal thresholding function for the $l_{1/2}$-norm regularization is defined as below:

$$h_{\lambda\tau,\frac{1}{2}}(\boldsymbol{x}_i) = \begin{cases} f_{\lambda\tau,\frac{1}{2}}(\boldsymbol{x}_i), & if \quad |\boldsymbol{x}_i| > \frac{\sqrt[3]{54}}{4}(\lambda\tau)^{\frac{2}{3}} \\ 0, & otherwise \end{cases} \quad \text{(VI.29)}$$

where the threshold $\frac{\sqrt[3]{54}}{4}(\lambda\tau)^{\frac{2}{3}}$ has been conceived and demonstrated in the literature [60].

Therefore, if Eq. VI.29 is applied to Eq. VI.27, the half proximal thresholding function, instead of the resolvent operator, for the $l_{1/2}$-norm regularization problem VI.25 can be explicitly reformulated as:

$$\boldsymbol{\alpha} = H_{\lambda\tau,\frac{1}{2}}(\theta(\boldsymbol{\alpha})) \quad \text{(VI.30)}$$

where the half proximal thresholding operator H [60] is deductively constituted by Eq. VI.29.

Up to now, the half proximal thresholding algorithm has been completely structured by Eq. VI.30. However, the options of the regularization parameter λ in Eq. VI.24 can seriously dominate the quality of the representation solution in problem VI.21, and the values of λ and τ can be specifically fixed by

$$\tau = \frac{1-\varepsilon}{\|X\|^2} \quad and \quad \lambda = \frac{\sqrt{96}}{9\tau}|[\theta(\boldsymbol{\alpha})]_{k+1}|^{\frac{3}{2}} \quad \text{(VI.31)}$$

where ε is a very small constant, which is very close to zero, the k denotes the limit of sparsity (i.e. k-sparsity), and $[\bullet]_k$ refers to the k-th largest component of $[\bullet]$. The half proximal thresholding algorithm for $l_{1/2}$-norm regularization based sparse representation is summarized in Algorithm 7 and more detailed inferences and analyses can be found in the literature [60, 88].

Algorithm 7. The half proximal thresholding algorithm for $l_{1/2}$-norm regularization
Task: To address the problem
$\hat{\boldsymbol{\alpha}} = \arg\min F(\boldsymbol{\alpha}) = \|X\boldsymbol{\alpha} - \boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_{1/2}^{1/2}$

Input: Probe sample y, the measurement matrix X
Initialization: $t = 0$, $\varepsilon = 0.01$, $\tau = \frac{1-\varepsilon}{\|X\|^2}$.
While not converged do
Step 1: Compute $\theta(\boldsymbol{\alpha}^t) = \boldsymbol{\alpha}^t + \tau X^T(\boldsymbol{y} - X\boldsymbol{\alpha}^t)$.
Step 2: Compute $\lambda_t = \frac{\sqrt{96}}{9\tau}|[\theta(\boldsymbol{\alpha}^t)]_{k+1}|^{\frac{3}{2}}$ in Eq. VI.31.
Step 3: Apply the half proximal thresholding operator to obtain the representation solution $\boldsymbol{\alpha}_{t+1} = H_{\lambda_t\tau,\frac{1}{2}}(\theta(\boldsymbol{\alpha}^t))$.
Step 4: $t = t + 1$.
End
Output: $\boldsymbol{\alpha}$

F. Augmented Lagrange Multiplier based optimization strategy

The Lagrange multiplier is a widely used tool to eliminate the equality constrained problem and convert it to address the unconstrained problem with an appropriate penalty function.

Specifically, the sparse representation problem III.9 can be viewed as an equality constrained problem and the equivalent problem III.12 is an unconstrained problem, which augments the objective function of problem III.9 with a weighted constraint function. In this section, the augmented Lagrangian method (ALM) is introduced to solve the sparse representation problem III.9.

First, the augmented Lagrangian function of problem III.9 is conceived by introducing an additional equality constrained function, which is enforced on the Lagrange function in problem III.12. That is,

$$L(\boldsymbol{\alpha},\lambda)=\|\boldsymbol{\alpha}\|_1+\frac{\lambda}{2}\|\boldsymbol{y}-X\boldsymbol{\alpha}\|_2^2 \quad s.t. \quad \boldsymbol{y}-X\boldsymbol{\alpha}=0 \tag{VI.32}$$

Then, a new optimization problem VI.32 with the form of the Lagrangain function is reformulated as

$$\arg\min L_\lambda(\boldsymbol{\alpha},\boldsymbol{z})=\|\boldsymbol{\alpha}\|_1+\frac{\lambda}{2}\|\boldsymbol{y}-X\boldsymbol{\alpha}\|_2^2+\boldsymbol{z}^T(\boldsymbol{y}-X\boldsymbol{\alpha}) \tag{VI.33}$$

where $z\in\mathbb{R}^d$ is called the Lagrange multiplier vector or dual variable and $L_\lambda(\boldsymbol{\alpha},\boldsymbol{z})$ is denoted as the augmented Lagrangian function of problem III.9. The optimization problem VI.33 is a joint optimization problem of the sparse representation coefficient $\boldsymbol{\alpha}$ and the Lagrange multiplier vector $\boldsymbol{z}$. Problem VI.33 is solved by optimizing $\boldsymbol{\alpha}$ and $\boldsymbol{z}$ alternatively as follows:

$$\begin{aligned}\boldsymbol{\alpha}^{t+1}&=\arg\min L_\lambda(\boldsymbol{\alpha},\boldsymbol{z}^t)\\&=\arg\min(\|\boldsymbol{\alpha}\|_1+\frac{\lambda}{2}\|\boldsymbol{y}-X\boldsymbol{\alpha}\|_2^2+(\boldsymbol{z}^t)^TX\boldsymbol{\alpha})\end{aligned} \tag{VI.34}$$

$$\boldsymbol{z}^{t+1}=\boldsymbol{z}^t+\lambda(\boldsymbol{y}-X\boldsymbol{\alpha}^{t+1}) \tag{VI.35}$$

where problem VI.34 can be solved by exploiting the FISTA algorithm. Problem VI.34 is iteratively solved and the parameter z is updated using Eq. VI.35 until the termination condition is satisfied. Furthermore, if the method of employing ALM to solve problem VI.33 is denoted as the primal augmented Lagrangian method (PALM) [89], the dual function of problem III.9 can also be addressed by the ALM algorithm, which is denoted as the dual augmented Lagrangian method (DALM) [89]. Subsequently, the dual optimization problem III.9 is discussed and the ALM algorithm is utilized to solve it.

First, consider the following equation:

$$\|\boldsymbol{\alpha}\|_1=\max_{\|\boldsymbol{\theta}\|_\infty\le 1}\langle\boldsymbol{\theta},\boldsymbol{\alpha}\rangle \tag{VI.36}$$

which can be rewritten as

$$\begin{aligned}&\|\boldsymbol{\alpha}\|_1=\max\{\langle\boldsymbol{\theta},\boldsymbol{\alpha}\rangle-I_{B_\infty^1}\}\\ or\quad &\|\boldsymbol{\alpha}\|_1=\sup\{\langle\boldsymbol{\theta},\boldsymbol{\alpha}\rangle-I_{B_\infty^1}\}\end{aligned} \tag{VI.37}$$

where $B_p^\lambda=\{\boldsymbol{x}\in R^N \mid \|\boldsymbol{x}\|_p\le\lambda\}$ and $I_\Omega(\boldsymbol{x})$ is a indicator function, which is defined as $I_\Omega(\boldsymbol{x})=\begin{cases}0 & ,\boldsymbol{x}\in\Omega\\ \infty & ,\boldsymbol{x}\notin\Omega\end{cases}$.

Hence,

$$\|\boldsymbol{\alpha}\|_1=\max\{\langle\boldsymbol{\theta},\boldsymbol{\alpha}\rangle\ :\boldsymbol{\theta}\in B_\infty^1\} \tag{VI.38}$$

Second, consider the Lagrange dual problem of problem III.9 and its dual function is

$$g(\lambda)=\inf_{\boldsymbol{\alpha}}\{\|\boldsymbol{\alpha}\|_1+\boldsymbol{\lambda}^T(\boldsymbol{y}-X\boldsymbol{\alpha})\}=\boldsymbol{\lambda}^T\boldsymbol{y}-\sup_{\boldsymbol{\alpha}}\{\boldsymbol{\lambda}^TX\boldsymbol{\alpha}-\|\boldsymbol{\alpha}\|_1\} \tag{VI.39}$$

where $\boldsymbol{\lambda}\in\mathbb{R}^d$ is a Lagrangian multiplier. If the definition of conjugate function is applied to Eq. VI.37, it can be verified that the conjugate function of $I_{B_\infty^1}(\theta)$ is $\|\boldsymbol{\alpha}\|_1$. Thus Eq. VI.39 can be equivalently reformulated as

$$g(\boldsymbol{\lambda})=\boldsymbol{\lambda}^T\boldsymbol{y}-I_{B_\infty^1}(X^T\boldsymbol{\lambda}) \tag{VI.40}$$

The Lagrange dual problem, which is associated with the primal problem III.9, is an optimization problem:

$$\max_\lambda \boldsymbol{\lambda}^T\boldsymbol{y} \quad s.t. \quad (X^T\boldsymbol{\lambda})\in B_\infty^1 \tag{VI.41}$$

Accordingly,

$$\min_{\lambda,z}-\boldsymbol{\lambda}^T\boldsymbol{y} \quad s.t. \quad \boldsymbol{z}-X^T\boldsymbol{\lambda}=0,\ \boldsymbol{z}\in B_\infty^1 \tag{VI.42}$$

Then, the optimization problem VI.42 can be reconstructed as

$$\begin{aligned}\arg\min_{\boldsymbol{\lambda},\boldsymbol{z},\boldsymbol{\mu}} L(\boldsymbol{\lambda},\boldsymbol{z},\boldsymbol{\mu})&=-\boldsymbol{\lambda}^T\boldsymbol{y}-\boldsymbol{\mu}^T(\boldsymbol{z}-X^T\boldsymbol{\lambda})\\&+\frac{\tau}{2}\|\boldsymbol{z}-X^T\boldsymbol{\lambda}\|_2^2 \quad s.t. \quad \boldsymbol{z}\in B_\infty^1\end{aligned} \tag{VI.43}$$

where $\boldsymbol{\mu}\in\mathbb{R}^d$ is a Lagrangian multiplier and τ is a penalty parameter.

Finally, the dual optimization problem VI.43 is solved and a similar alternating minimization idea of PALM can also be applied to problem VI.43, that is,

$$\begin{aligned}\boldsymbol{z}^{t+1}&=\arg\min_{z\in B_\infty^1} L_\tau(\boldsymbol{\lambda}^t,\boldsymbol{z},\boldsymbol{\mu}^t)\\&=\arg\min_{\boldsymbol{z}\in B_\infty^1}\{-\boldsymbol{\mu}^T(\boldsymbol{z}-X^T\boldsymbol{\lambda}^t)+\frac{\tau}{2}\|\boldsymbol{z}-X^T\boldsymbol{\lambda}^t\|_2^2\}\\&=\arg\min_{\boldsymbol{z}\in B_\infty^1}\{\frac{\tau}{2}\|\boldsymbol{z}-(X^T\boldsymbol{\lambda}^t+\frac{2}{\tau}\boldsymbol{\mu}^T)\|_2^2\}\\&=P_{B_\infty^1}(X^T\boldsymbol{\lambda}^t+\frac{1}{\tau}\boldsymbol{\mu}^T)\end{aligned} \tag{VI.44}$$

where $P_{B_\infty^1}(\boldsymbol{u})$ is a projection, or called a proximal operator, onto B_∞^1 and it is also called group-wise soft-thresholding. For example, let $x=P_{B_\infty^1}(\boldsymbol{u})$, then the i-th component of solution $\boldsymbol{x}$ satisfies $\boldsymbol{x}_i=sign(\boldsymbol{u}_i)\min\{|\boldsymbol{u}_i|,1\}$

$$\begin{aligned}\boldsymbol{\lambda}^{t+1}&=\arg\min_\lambda L_\tau(\boldsymbol{\lambda},\boldsymbol{z}^{t+1},\boldsymbol{\mu}^t)\\&=\arg\min_\lambda\{-\boldsymbol{\lambda}^T\boldsymbol{y}+(\boldsymbol{\mu}^t)^TX^T\boldsymbol{\lambda}+\frac{\tau}{2}\|\boldsymbol{z}^{t+1}-X^T\boldsymbol{\lambda}\|_2^2\}\\&=Q(\boldsymbol{\lambda})\end{aligned} \tag{VI.45}$$

Take the derivative of $Q(\boldsymbol{\lambda})$ with respect to $\boldsymbol{\lambda}$ and obtain

$$\boldsymbol{\lambda}^{t+1}=(\tau XX^T)^{-1}(\tau X\boldsymbol{z}^{t+1}+\boldsymbol{y}-X\boldsymbol{\mu}^t) \tag{VI.46}$$

$$\boldsymbol{\mu}^{t+1}=\boldsymbol{\mu}^t-\tau(\boldsymbol{z}^{t+1}-X^T\boldsymbol{\lambda}^{t+1}) \tag{VI.47}$$

The DALM for sparse representation with l_1-norm regularization mainly exploits the augmented Lagrange method to address the dual optimization problem of problem III.9

and a proximal operator, the projection operator, is utilized to efficiently solve the subproblem. The algorithm of DALM is summarized in Algorithm 8. For more detailed description, please refer to the literature [89].

Algorithm 8. Dual augmented Lagrangian method for l_1-norm regularization
Task: To address the dual problem of $\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_1 \quad s.t. \quad \boldsymbol{y} = X\boldsymbol{\alpha}$

Input: Probe sample y, the measurement matrix X, a small constant λ^0.
Initialization: $t = 0$, $\varepsilon = 0.01$, $\tau = \frac{1-\varepsilon}{\|X\|^2}$, $\mu^0 = 0$.
While not converged do
Step 1: Apply the projection operator to compute
$\boldsymbol{z}^{t+1} = P_{B^1_\infty}(X^T\boldsymbol{\lambda}^t + \frac{1}{\tau}\boldsymbol{\mu}^T)$.
Step 2: Update the value of $\boldsymbol{\lambda}^{t+1} = (\tau XX^T)^{-1}(\tau X\boldsymbol{z}^{t+1} + \boldsymbol{y} - X\boldsymbol{\mu}^t)$.
Step 3: Update the value of $\boldsymbol{\mu}^{t+1} = \boldsymbol{\mu}^t - \tau(\boldsymbol{z}^{t+1} - X^T\boldsymbol{\lambda}^{t+1})$.
Step 4: $t = t + 1$.
End While
Output: $\boldsymbol{\alpha} = \boldsymbol{\mu}[1:N]$

G. Other proximity algorithm based optimization methods

The theoretical basis of the proximity algorithm is to first construct a proximal operator, and then utilize the proximal operator to solve the convex optimization problem. Massive proximity algorithms have followed up with improved techniques to improve the effectiveness and efficiency of proximity algorithm based optimization methods. For example, Elad et al. proposed an iterative method named parallel coordinate descent algorithm (PCDA) [90] by introducing the element-wise optimization algorithm to solve the regularized linear least squares with non-quadratic regularization problem.

Inspired by belief propagation in graphical models, Donoho et al. developed a modified version of the iterative thresholding method, called approximate message passing (AMP) method [91], to satisfy the requirement that the sparsity undersampling tradeoff of the new algorithm is equivalent to the corresponding convex optimization approach. Based on the development of the first-order method called Nesterov's smoothing framework in convex optimization, Becker et al. proposed a generalized Nesterov's algorithm (NESTA) [92] by employing the continuation-like scheme to accelerate the efficiency and flexibility. Subsequently, Becker et al. [93] further constructed a general framework, i.e. templates for convex cone solvers (TFOCS), for solving massive certain types of compressed sensing reconstruction problems by employing the optimal first-order method to solve the smoothed dual problem of the equivalent conic formulation of the original optimization problem. Further detailed analyses and inference information related to proximity algorithms can be found in the literature [28, 83].

VII. Homotopy algorithm based sparse representation

The concept of homotopy derives from topology and the homotopy technique is mainly applied to address a nonlinear system of equations problem. The homotopy method was originally proposed to solve the least square problem with the l_1-penalty [94]. The main idea of homotopy is to solve the original optimization problems by tracing a continuous parameterized path of solutions along with varying parameters. Having a highly intimate relationship with the conventional sparse representation method such as least angle regression (LAR) [42], OMP [64] and polytope faces pursuit (PFP) [95], the homotopy algorithm has been successfully employed to solve the l_1-norm minimization problems. In contrast to LAR and OMP, the homotopy method is more favorable for sequentially updating the sparse solution by adding or removing elements from the active set. Some representative methods that exploit the homotopy-based strategy to solve the sparse representation problem with the l_1-norm regularization are explicitly presented in the following parts of this section.

A. LASSO homotopy

Because of the significance of parameters in l_1-norm minimization, the well-known LASSO homotopy algorithm is proposed to solve the LASSO problem in III.9 by tracing the whole homotopy solution path in a range of decreasing values of parameter λ. It is demonstrated that problem III.12 with an appropriate parameter value is equivalent to problem III.9 [29]. Moreover, it is apparent that as we change λ from a very large value to zero, the solution of problem III.12 is converging to the solution of problem III.9 [29]. The set of varying value λ conceives the solution path and any point on the solution path is the optimality condition of problem III.12. More specifically, the LASSO homotopy algorithm starts at an large initial value of parameter λ and terminates at a point of λ, which approximates zero, along the homotopy solution path so that the optimal solution converges to the solution of problem III.9. The fundamental of the homotopy algorithm is that the homotopy solution path is a piecewise linear path with a discrete number of operations while the value of the homotopy parameter changes, and the direction of each segment and the step size are absolutely determined by the sign sequence and the support of the solution on the corresponding segment, respectively [96].

Based on the basic ideas in a convex optimization problem, it is a necessary condition that the zero vector should be a solution of the subgradient of the objective function of problem III.12. Thus, we can obtain the subgradiential of the objective function with respect to $\boldsymbol{\alpha}$ for any given value of λ, that is,

$$\frac{\partial L}{\partial \boldsymbol{\alpha}} = -X^T(y - X\boldsymbol{\alpha}) + \lambda\partial\|\boldsymbol{\alpha}\|_1 \tag{VII.1}$$

where the first term $r = X^T(y - X\boldsymbol{\alpha})$ is called the vector of residual correlations, and $\partial\|\boldsymbol{\alpha}\|_1$ is the subgradient obtained by

$$\partial\|\boldsymbol{\alpha}\|_1 = \left\{ \boldsymbol{\theta} \in R^N \middle| \begin{array}{ll} \boldsymbol{\theta}_i = sgn(\boldsymbol{\alpha}_i), & \boldsymbol{\alpha}_i \neq 0 \\ \boldsymbol{\theta}_i \in [-1, 1], & \boldsymbol{\alpha}_i = 0 \end{array} \right\}$$

Let Λ and $\boldsymbol{u}$ denote the support of $\boldsymbol{\alpha}$ and the sign sequence of $\boldsymbol{\alpha}$ on its support Λ, respectively. X_Λ denotes that the indices of all the samples in X_Λ are all included in the support set Λ. If we analyze the KKT optimality condition for problem III.12, we can obtain the following two equivalent conditions of problem VII.1, i.e.

$$X_\Lambda(\boldsymbol{y} - X\boldsymbol{\alpha}) = \lambda\boldsymbol{u}; \quad \|X^T_{\Lambda^c}(\boldsymbol{y} - X\boldsymbol{\alpha})\|_\infty \leq \lambda \tag{VII.2}$$

where Λ^c denotes the complementary set of the set Λ. Thus, the optimality conditions in VII.2 can be divided into N constraints and the homotopy algorithm maintains both of the conditions along the optimal homotopy solution path for any $\lambda \geq 0$. As we decrease the value of λ to $\lambda - \tau$, for a small value of τ, the following conditions should be satisfied

$$\begin{array}{ll} X_\Lambda^T(\boldsymbol{y} - X\boldsymbol{\alpha}) + \tau X_\Lambda^T X\boldsymbol{\delta} = (\lambda - \tau)\boldsymbol{u} & (a) \\ \|p + \tau q\|_\infty \leq \lambda - \tau & (b) \end{array} \tag{VII.3}$$

where $\boldsymbol{p} = X^T(\boldsymbol{y} - X\boldsymbol{\alpha})$, $\boldsymbol{q} = X^T X\boldsymbol{\delta}$ and $\boldsymbol{\delta}$ is the update direction.

Generally, the homotopy algorithm is implemented iteratively and it follows the homotopy solution path by updating the support set by decreasing parameter λ from a large value to the desired value. The support set of the solution will be updated and changed only at a critical point of λ, where either an existing nonzero element shrinks to zero or a new nonzero element will be added into the support set. The two most important parameters are the step size τ and the update direction $\boldsymbol{\delta}$. At the l-th stage (if $(X_\Lambda^T X_\Lambda)^{-1}$ exists), the homotopy algorithm first calculates the update direction, which can be obtained by solving

$$X_\Lambda^T X_\Lambda \boldsymbol{\delta}_l = \boldsymbol{u} \tag{VII.4}$$

Thus, the solution of problem VII.4 can be written as

$$\boldsymbol{\delta}_l = \begin{cases} (X_\Lambda^T X_\Lambda)^{-1}\boldsymbol{u}, & on\ \Lambda \\ 0, & otherwise \end{cases} \tag{VII.5}$$

Subsequently, the homotopy algorithm computes the step size τ to the next critical point by tracing the homotopy solution path. i.e. the homotopy algorithm moves along the update direction until one of constraints in VII.3 is not satisfied. At this critical point, a new nonzero element must enter the support Λ, or one of the nonzero elements in $\boldsymbol{\alpha}$ will be shrink to zero, i.e. this element must be removed from the support set Λ. Two typical cases may lead to a new critical point, where either condition of VII.3 is violated. The minimum step size which leads to a critical point can be easily obtained by computing $\tau_l^* = min(\tau_l^+, \tau_l^-)$, and τ_l^+ and τ_l^- are computed by

$$\tau_l^+ = min_{i \in \Lambda^c}\left(\frac{\lambda - \boldsymbol{p}_i}{1 - \boldsymbol{x}_i^T X_\Lambda \boldsymbol{\delta}_l}, \frac{\lambda + \boldsymbol{p}_i}{1 + \boldsymbol{x}_i^T X_\Lambda \boldsymbol{\delta}_l}\right)_+ \tag{VII.6}$$

$$\tau_l^- = min_{i \in \Lambda}\left(\frac{-\boldsymbol{\alpha}_l^i}{\boldsymbol{\delta}_l^i}\right)_+ \tag{VII.7}$$

where $p_i = \boldsymbol{x}_i^T(\boldsymbol{y} - \boldsymbol{x}_i\boldsymbol{\alpha}_l^i)$ and $min(\cdot)_+$ denotes that the minimum is operated over only positive arguments. τ_l^+ is the minimum step size that turns an inactive element at the index i^+ in to an active element, i.e. the index i^+ should be added into the support set. τ_l^- is the minimum step size that shrinks the value of a nonzero active element to zero at the index i^- and the index i^- should be removed from the support set. The solution is updated by $\boldsymbol{\alpha}_{l+1} = \boldsymbol{\alpha}_l + \tau_l^*\boldsymbol{\delta}$, and its support and sign sequence are renewed correspondingly.

The homotopy algorithm iteratively computes the step size and the update direction, and updates the homotopy solution and its corresponding support and sign sequence till the condition $\|\boldsymbol{p}\|_\infty = 0$ is satisfied so that the solution of problem III.9 is reached. The principal steps of the LASSO homotopy algorithm have been summarized in Algorithm 9. For further description and analyses, please refer to the literature [29, 96].

Algorithm 9. Lasso homotopy algorithm
Task: To addrss the Lasso problem:
$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} \|\boldsymbol{y} - X\boldsymbol{\alpha}\|_2^2 \ \ s.t. \ \ \|\boldsymbol{\alpha}\|_1 \leq \varepsilon$

Input: Probe sample y, measurement matrix X.
Initialization: $l = 1$, initial solution $\boldsymbol{\alpha}_l$ and its support set Λ_l.
Repeat:
Step 1: Compute update direction δ_l by using Eq. (VII.5).
Step 2: Compute τ_l^+ and τ_l^- by using Eq. (VII.6) and Eq. (VII.7).
Step 3: Compute the optimal minimum step size τ_l^* by using $\tau_l^* = min\{\tau_l^+, \tau_l^-\}$.
Step 4: Update the solution $\boldsymbol{\alpha}_{l+1}$ by using $\boldsymbol{\alpha}_{l+1} = \boldsymbol{\alpha}_l + \tau_l^*\boldsymbol{\delta}_l$.
Step 5: Update the support set:
If $\tau_l^+ == \tau_l^-$ then
Remove the i^- from the support set, i.e. $\Lambda_{l+1} = \Lambda_l \backslash i^-$.
else
Add the i^+ into the support set, i.e. $\Lambda_{l+1} = \Lambda_l \bigcup i^+$
End if
Step 6: $l = l + 1$.
Until $\|X^T(\boldsymbol{y} - X\boldsymbol{\alpha})\|_\infty = 0$
Output: $\boldsymbol{\alpha}_{l+1}$

B. BPDN homotopy

Problem III.11, which is called basis pursuit denoising (BPDN) in signal processing, is the unconstrained Lagrangian function of the LASSO problem III.9, which is an unconstrained problem. The BPDN homotopy algorithm is very similar to the LASSO homotopy algorithm. If we consider the KKT optimality condition for problem III.12, the following condition should be satisfied for the solution $\boldsymbol{\alpha}$

$$\|X^T(\boldsymbol{y} - X\boldsymbol{\alpha})\|_\infty \leq \lambda \tag{VII.8}$$

As for any given value of λ and the support set Λ, the following two conditions also need to be satisfied

$$X_\Lambda^T(\boldsymbol{y} - X\boldsymbol{\alpha}) = \lambda\boldsymbol{u}; \quad \|X_{\Lambda^c}^T(\boldsymbol{y} - X\boldsymbol{\alpha})\|_\infty \leq \lambda \tag{VII.9}$$

The BPDN homotopy algorithm directly computes the homotopy solution by

$$\boldsymbol{\alpha} = \begin{cases} (X_\Lambda^T X_\Lambda)^{-1}(X_\Lambda^T\boldsymbol{y} - \lambda\boldsymbol{u}), & on\ \Lambda \\ 0, & otherwise \end{cases} \tag{VII.10}$$

which is somewhat similar to the soft-thresholding operator. The value of the homotopy parameter λ is initialized with a large value, which satisfies $\lambda_0 > \|X^T y\|_\infty$. As the value of the homotopy parameter λ decreases, the BPDN homotopy algorithm traces the solution in the direction of $(X_\Lambda^T X_\Lambda)^{-1}u$ till the critical point is obtained. Each critical point is reached when either an inactive element is transferred into an active element, i.e. its corresponding index should be added into the support set, or an nonzero active element value in $\boldsymbol{\alpha}$ shrinks to zero, i.e. its corresponding index should be removed from the support set. Thus, at each critical point, only one element is updated, i.e. one element being either removed from or added into the active set, and each operation is very

computationally efficient. The algorithm is terminated when the value of the homotopy parameter is lower than its desired value. The BPDN homotopy algorithm has been summarized in Algorithm 10. For further detail description and analyses, please refer to the literature [42].

Algorithm 10. BPDN homotopy algorithm
Task: To address the Lasso problem:
$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} \|\boldsymbol{y} - X\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1$

Input: Probe sample $\boldsymbol{y}$, measurement matrix X.
Initialization: $l = 0$, initial solution $\boldsymbol{\alpha}_0$ and its support set Λ_0, a large value λ_0, step size τ, tolerance ε.
Repeat:
Step 1: Compute update direction $\boldsymbol{\delta}_{l+1}$ by using
$\boldsymbol{\delta}_{l+1} = (X_\Lambda^T X_\Lambda)^{-1}\boldsymbol{u}_l$.
Step 2: Update the solution $\boldsymbol{\alpha}_{l+1}$ by using Eq. (VII.10).
Step 3: Update the support set and the sign sequence set.
Step 6: $\lambda_{l+1} = \lambda_l - \tau$, $l = l + 1$.
Until $\lambda \leq \varepsilon$
Output: $\boldsymbol{\alpha}_{l+1}$

C. Iterative Reweighting l_1-norm minimization via homotopy

Based on the homotopy algorithm, Asif and Romberg [96] presented a enhanced sparse representation objective function, a weighted l_1-norm minimization, and then provided two fast and accurate solutions, i.e. the iterative reweighting algorithm, which updated the weights with a new ones, and the adaptive reweighting algorithm, which adaptively selected the weights in each iteration. Here the iterative reweighting algorithm via homotopy is introduced. The objective function of the weighted l_1-norm minimization is formulated as

$$argmin\frac{1}{2}\|X\boldsymbol{\alpha} - \boldsymbol{y}\|_2^2 + \|W\boldsymbol{\alpha}\|_1 \tag{VII.11}$$

where $W = diag[w_1, w_2, \cdots, w_N]$ is the weight of the l_1-norm and also is a diagonal matrix. For more explicit description, problem VII.11 can be rewritten as

$$argmin\frac{1}{2}\|X\boldsymbol{\alpha} - \boldsymbol{y}\|_2^2 + \sum_{i=1}^{N} w_i|\boldsymbol{\alpha}_i| \tag{VII.12}$$

A common method [42, 73] to update the weight W is achieved by exploiting the solution of problem VII.12, i.e. $\boldsymbol{\alpha}$, at the previous iteration, and for the i-th element of the weight w_i is updated by

$$w_i = \frac{\lambda}{|\boldsymbol{\alpha}_i| + \sigma} \tag{VII.13}$$

where parameters λ and σ are both small constants. In order to efficiently update the solution of problem (7-9), the homotopy algorithm introduces a new weight of the l_1-norm and a new homotopy based reweighting minimization problem is reformulated as

$$argmin\ \frac{1}{2}\|X\boldsymbol{\alpha} - \boldsymbol{y}\|_2^2 + \sum_{i=1}^{N}((1-\sigma)\hat{w}_i + \sigma\hat{w}_i)|\boldsymbol{\alpha}_i| \tag{VII.14}$$

where $\hat{w}_i$ denotes the new obtained weight by the homotopy algorithm, parameter τ is denoted as the homotopy parameter varying from 0 to 1. Apparently, problem VII.14 can be evolved to problem VII.12 with the increasing value of the homotopy parameter by tracing the homotopy solution path. Similar to the LASSO homotopy algorithm, problem VII.14 is also piecewise linear along the homotopy path, and for any value of σ, the following conditions should be satisfied

$$\begin{array}{ll} x_i^T(X\boldsymbol{\alpha} - y) = -((1-\sigma)w_i + \sigma\hat{w}_i)u_i & for\ i \in \Lambda\ (a) \\ |x_i^T(X\boldsymbol{\alpha} - y)| < (1-\sigma)w_i + \sigma\hat{w}_i & for\ i \in \Lambda^c\ (b) \end{array} \tag{VII.15}$$

where x_i is the i-th column of the measurement X, w_i and $\hat{w}_i$ are the given weight and new obtained weight, respectively. Moreover, for the optimal step size σ, when the homotopy parameter changes from σ to $\sigma + \tau$ in the update direction $\boldsymbol{\delta}$, the following optimality conditions also should be satisfied

$$\begin{array}{ll} X_\Lambda^T(X\boldsymbol{\alpha} - \boldsymbol{y}) + \tau X_\Lambda^T X\boldsymbol{\delta} = & \\ -((1-\sigma)W + \sigma\hat{W})\boldsymbol{u} + \tau(W - \hat{W})\boldsymbol{u} & (a) \\ |\boldsymbol{p} - \tau\boldsymbol{q}| \leq r + \tau s & (b) \end{array} \tag{VII.16}$$

where $\boldsymbol{u}$ is the sign sequence of $\boldsymbol{\alpha}$ on its support Λ, $p_i = x_i^T(X\boldsymbol{\alpha} - \boldsymbol{y})$, $q_i = x_i^T X\boldsymbol{\delta}$, $r_i = (1-\sigma)w_i + \sigma\hat{w}_i$ and $s_i = \hat{w}_i - w_i$. Thus, at the l-th stage (if $(X_i^T X_i)^{-1}$ exists), the update direction of the homotopy algorithm can be computed by

$$\boldsymbol{\delta}_l = \begin{cases} (X_\Lambda^T X_\Lambda)^{-1}(W - \hat{W})\boldsymbol{u}, & on\ \Lambda \\ 0, & otherwise \end{cases} \tag{VII.17}$$

The step size which can lead to a critical point can be computed by $\tau_l^* = min(\tau_l^+, \tau_l^-)$, and τ_l^+ and τ_l^- are computed by

$$\tau_l^+ = min_{i\in\Lambda^c}\left(\frac{r_i - p_i}{q_i - s_i}, \frac{-r_i - p_i}{q_i + s_i}\right)_+ \tag{VII.18}$$

$$\tau_l^- = min_{i\in\Lambda}\left(\frac{-\boldsymbol{\alpha}_l^i}{\boldsymbol{\delta}_l^i}\right)_+ \tag{VII.19}$$

where τ_l^+ is the minimum step size so that the index i^+ should be added into the support set and τ_l^- is the minimum step size that shrinks the value of a nonzero active element to zero at the index i^-. The solution and homotopy parameter are updated by $\boldsymbol{\alpha}_{l+1} = \boldsymbol{\alpha}_l + \tau_l^*\boldsymbol{\delta}$, and $\sigma_{l+1} = \sigma_l + \tau_l^*$, respectively. The homotopy algorithm updates its support set and sign sequence accordingly until the new critical point of the homotopy parameter $\sigma_{l+1} = 1$. The main steps of this algorithm are summarized in Algorithm 11 and more information can be found in literature [96].

D. Other homotopy algorithms for sparse representation

The general principle of the homotopy method is to reach the optimal solution along with the homotopy solution path by evolving the homotopy parameter from a known initial value to the final expected value. There are extensive hotomopy algorithms, which are related to the sparse representation with the l_1-norm regularization. Malioutov et al. first exploited the homotopy method to choose a suitable parameter for l_1-norm regularization with a noisy term in an underdetermined system and employed the homotopy continuation-based method to solve BPDN for sparse signal processing [97]. Garrigues and Ghaoui [98] proposed a modified homotopy algorithm to solve the Lasso problem with online observations by optimizing the

Algorithm 11. Iterative reweighting homotopy algorithm for weighted l_1-norm minimization
Task: To addrss the weighted l_1-norm minimization:
$\hat{\boldsymbol{\alpha}} = arg\min \frac{1}{2}\|X\boldsymbol{\alpha} - \boldsymbol{y}\|_2^2 + W\|\boldsymbol{\alpha}\|_1$

Input: Probe sample y, measurement matrix X.
Initialization: $l = 1$, initial solution $\boldsymbol{\alpha}_l$ and its support set Λ_l, $\sigma_1 = 0$.
Repeat:
Step 1: Compute update direction $\boldsymbol{\delta}_l$ by using Eq. (VII.17).
Step 2: Compute p, q, r and s by using Eq. (VII.16).
Step 2: Compute τ_l^+ and τ_l^- by using Eq. (VII.18) and Eq. (VII.19).
Step 3: Compute the step size τ_l^* by using
$\tau_l^* = min\{\tau_l^+, \tau_l^-\}$.
Step 4: Update the solution $\boldsymbol{\alpha}_{l+1}$ by using $\boldsymbol{\alpha}_{l+1} = \boldsymbol{\alpha}_l + \tau_l^* \boldsymbol{\delta}_l$.
Step 5: Update the support set:
If $\tau_l^+ == \tau_l^-$ then
Shrink the value to zero at the index i^- and remove i^-, i.e. $\Lambda_{l+1} = \Lambda_l \backslash i^-$.
else
Add the i^+ into the support set, i.e. $\Lambda_{l+1} = \Lambda_l \bigcup i^+$
End if
Step 6: $\sigma_{l+1} = \sigma_l + \tau_l$ and $l = l + 1$.
Until $\sigma_{l+1} = 1$
Output: $\boldsymbol{\alpha}_{l+1}$

homotopy parameter from the current solution to the solution after obtaining the next new data point. Efron et al. [42] proposed a basic pursuit denoising (BPDN) homotopy algorithm, which shrinked the parameter to a final value with series of efficient optimization steps. Similar to BPDN homotopy, Asif [99] presented a homotopy algorithm for the Dantzing selector (DS) under the consideration of primal and dual solution. Asif and Romberg [100] proposed a framework of dynamic updating solutions for solving l_1-norm minimization programs based on homotopy algorithm and demonstrated its effectiveness in addressing the decoding issue. More recent literature related to homotopy algorithms can be found in the streaming recovery framework [101] and a summary [102].

VIII. The Applications of the Sparse Representation Method

Sparse representation technique has been successfully implemented to numerous applications, especially in the fields of computer vision, image processing, pattern recognition and machine learning. More specifically, sparse representation has also been successfully applied to extensive real-world applications, such as image denoising, deblurring, inpainting, super-resolution, restoration, quality assessment, classification, segmentation, signal processing, object tracking, texture classification, image retrieval, bioinformatics, biometrics and other artificial intelligence systems. Moreover, dictionary learning is one of the most typical representative examples of sparse representation for realizing the sparse representation of a signal. In this paper, we only concentrate on the three applications of sparse representation, i.e. sparse representation in dictionary learning, image processing, image classification and visual tracking.

A. Sparse representation in dictionary learning

The history of modeling dictionary could be traced back to 1960s, such as the fast Fourier transform (FFT) [103]. An over-complete dictionary that can lead to sparse representation is usually achieved by exploiting pre-specified set of transformation functions, i.e. transform domain method [5], or is devised based on learning, i.e. dictionary learning methods [104]. Both of the transform domain and dictionary learning based methods transform image samples into other domains and the similarity of transformation coefficients are exploited [105]. The difference between them is that the transform domain methods usually utilize a group of fixed transformation functions to represent the image samples, whereas the dictionary learning methods apply sparse representations on a over-complete dictionary with redundant information. Moreover, exploiting the pre-specified transform matrix in transform domain methods is attractive because of its fast and simplicity. Specifically, the transform domain methods usually represent the image patches by using the orthonormal basis such as over-complete wavelets transform [106], super-wavelet transform [107], bandelets [108], curvelets transform [109], contourlets transform [110] and steerable wavelet filters [111]. However, the dictionary learning methods exploiting sparse representation have the potential capabilities of outperforming the pre-determined dictionaries based on transformation functions. Thus, in this subsection we only focus on the modern over-complete dictionary learning methods.

An effective dictionary can lead to excellent reconstruction results and satisfactory applications, and the choice of dictionary is also significant to the success of sparse representation technique. Different tasks have different dictionary learning rules. For example, image classification requires that the dictionary contains discriminative information such that the solution of sparse representation possesses the capability of distinctiveness. The purpose of dictionary learning is motivated from sparse representation and aims to learn a faithful and effective dictionary to largely approximate or simulate the specific data. In this section, some parameters are defined as matrix $Y = [\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_N]$, matrix $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_t]^T$, and dictionary $D = [\boldsymbol{d}_1, \boldsymbol{d}_2, \cdots, \boldsymbol{d}_M]$.

From the notations of the literature [22, 112], the framework of dictionary learning can be generally formulated as an optimization problem

$$\arg \min_{D \in \Omega, x_i} \left\{ \frac{1}{N} \sum_{i=1}^{N} (\frac{1}{2}\|\boldsymbol{y}_i - D\boldsymbol{x}_i\|_2^2 + \lambda P(\boldsymbol{x}_i)) \right\} \quad \text{(VIII.1)}$$

where $\Omega = \{D = [\boldsymbol{d}_1, \boldsymbol{d}_2, \cdots, \boldsymbol{d}_M] : \boldsymbol{d}_i^T \boldsymbol{d}_i = 1, i = 1, 2, \cdots, M\}$ (M here may not be equal to N), N denotes the number of the known data set (eg. training samples in image classification), $\boldsymbol{y}_i$ is the i-th sample vector from a known set, D is the learned dictionary and $\boldsymbol{x}_i$ is the sparsity vector. $P(\boldsymbol{x}_i)$ and λ are the penalty or regularization term and a tuning parameter, respectively. The regularization term of problem VIII.1 controls the degree of sparsity. That is, different kinds of the regularization terms can immensely dominate the dictionary learning results.

One spontaneous idea of defining the penalty term $P(\boldsymbol{x}_i)$ is to introduce the l_0-norm regularization, which leads to the sparsest solution of problem VIII.1. As a result, the theory of sparse representation can be applied to dictionary

learning. The most representative dictionary learning based on the l_0-norm penalty is the K-SVD algorithm [8], which is widely used in image denoising. Because the solution of l_0-norm regularization is usually a NP-hard problem, utilizing a convex relaxation strategy to replace l_0-norm regularization is an advisable choice for dictionary learning. As a convex relaxation method of l_0-norm regularization, the l_1-norm regularization based dictionary learning has been proposed in large numbers of dictionary learning schemes. In the stage of convex relaxation methods, there are three optimal forms for updating a dictionary: the one by one atom updating method, group atoms updating method, and all atoms updating method [112]. Furthermore, because of over-penalization in l_1-norm regularization, non-convex relaxation strategies also have been employed to address dictionary learning problems. For example, Fan and Li proposed a smoothly clipped absolution deviation (SCAD) penalty [113], which employed an iterative approximate Newton-Raphson method for penalizing least sequences and exploited the penalized likelihood approaches for variable selection in linear regression models. Zhang introduced and studied the non-convex minimax concave (MC) family [114] of non-convex piecewise quadratic penalties to make unbiased variable selection for the estimation of regression coefficients, which was demonstrated its effectiveness by employing an oracle inequality. Friedman proposed to use the logarithmic penalty for a model selection [115] and used it to solve the minimization problems with non-convex regularization terms. From the viewpoint of updating strategy, most of the dictionary learning methods always iteratively update the sparse approximation or representation solution and the dictionary alternatively, and more dictionary learning theoretical results and analyses can be found in the literature [116, 117].

Recently, varieties of dictionary learning methods have been proposed and researchers have attempted to exploit different strategies for implementing dictionary learning tasks based on sparse representation. There are several means to categorize these dictionary learning algorithms into various groups. For example, dictionary learning methods can be divided into three groups in the context of different norms utilized in the penalty term, that is, l_0-norm regularization based methods, convex relaxation methods and non-convex relaxation methods [118]. Moreover, dictionary learning algorithms can also be divided into three other categories in the presence of different structures. The first category is dictionary learning under the probabilistic framework such as maximum likelihood methods [119], the method of optimal directions (MOD) [120], and the maximum a posteriori probability method [121]. The second category is clustering based dictionary learning approaches such as KSVD [122], which can be viewed as a generalization of K-means. The third category is dictionary learning with certain structures, which are grouped into two significative aspects, i.e. directly modeling the relationship between each atom and structuring the corrections between each atom with purposive sparsity penalty functions. There are two typical models for these kinds of dictionary learning algorithms, sparse and shift-invariant representation of dictionary learning and structure sparse regularization based dictionary learning, such as hierarchical sparse dictionary learning [123] and group or block sparse dictionary learning [124]. Recently, some researchers [22] categorized the latest methods of dictionary learning into four groups, online dictionary learning [125], joint dictionary learning [126], discriminative dictionary learning [127], and supervised dictionary learning [128].

Although there are extensive strategies to divide the available sparse representation based dictionary learning methods into different categories, the strategy used here is to categorize the current prevailing dictionary learning approaches into two main classes: supervised dictionary learning and unsupervised dictionary learning, and then specific representative algorithms are explicitly introduced.

1) Unsupervised dictionary learning: From the viewpoint of theoretical basis, the main difference of unsupervised and supervised dictionary learning relies on whether the class label is exploited in the process of learning for obtaining the dictionary. Unsupervised dictionary learning methods have been widely implemented to solve image processing problems, such as image compression, and feature coding of image representation [129, 130].

(1) KSVD for unsupervised dictionary learning

One of the most representative unsupervised dictionary learning algorithms is the KSVD method [122], which is a modification or an extension of method of directions (MOD) algorithm. The objective function of KSVD is

$$\arg\min_{D,X}\{\|Y-DX\|_F^2\} \quad s.t. \quad \|\boldsymbol{x}_i\|_0 \leq k,\ i=1,2,\cdots,N \tag{VIII.2}$$

where $Y \in \mathbb{R}^{d\times N}$ is the matrix composed of all the known examples, $D \in \mathbb{R}^{d\times N}$ is the learned dictionary, $X \in \mathbb{R}^{N\times N}$ is the matrix of coefficients, k is the limit of sparsity and $\boldsymbol{x}_i$ denotes the i-th row vector of the matrix X. Problem VIII.2 is a joint optimization problem with respect to D and X, and the natural method is to alternatively optimize the D and X iteratively.

Algorithm 12. The K-SVD algorithm for dictionary learning
Task: Learning a dictionary D: $\arg\min_{D,X}\|Y-DX\|_F^2\ s.t.\ \|\boldsymbol{x}_i\|_0 \leq k,\ i=1,2,\cdots,N$

Input: The matrix composed of given samples $Y=[\boldsymbol{y}_1,\boldsymbol{y}_2,\cdots,\boldsymbol{y}_m]$.
Initialization: Set the initial dictionary D to the l_2–norm unit matrix, $i=1$.
While not converged do
  Step 1: For each given example $\boldsymbol{y}_i$, employing the classical sparse representation with l_0-norm regularization to solve problem VIII.3 for further estimating X^i,set $l=1$.
    While l is not equal to k do
      Step 2: Compute the overall representation residual $E_l = Y - \sum_{j\neq l}\boldsymbol{d}_j\boldsymbol{x}_j^T$.
      Step 3: Extract the column items of E_l which corresponds to the nonzero elements of $\boldsymbol{x}_l^T$ and obtain E_l^P.
      Step 4: SVD decomposes E_l^P into $E_l^P = U\Lambda V^T$.
      Step 5: Update $\boldsymbol{d}_l$ to the first column of U and update corresponding coefficients in $\boldsymbol{x}_l^T$ by $\Lambda(1,1)$ times the first column of V.
      Step 6: $l=l+1$.
    End While
  Step 7: $i=i+1$.
End While
Output: dictionary D

More specifically, when fixing dictionary D, problem VIII.2

is converted to

$$\arg\min_{X} \|Y - DX\|_F^2 \quad s.t. \quad \|\boldsymbol{x}_i\|_0 \leq k,\ i = 1, 2, \cdots, N \tag{VIII.3}$$

which is called sparse coding and k is the limit of sparsity. Then, its subproblem is considered as follows:

$$\arg\min_{\boldsymbol{x}_i} \|\boldsymbol{y}_i - D\boldsymbol{x}_i\|_2^2 \quad s.t. \quad \|\boldsymbol{x}_i\|_0 \leq k,\ i = 1, 2, \cdots, N$$

where we can iteratively resort to the classical sparse representation with l_0 -norm regularization such as MP and OMP, for estimating $\boldsymbol{x}_i$.

When fixing X, problem VIII.3 becomes a simple regression model for obtaining D, that is

$$\hat{D} = \arg\min_{D} \|Y - DX\|_F^2 \tag{VIII.4}$$

where $\hat{D} = YX^{\dagger} = YX^T(XX^T)^{-1}$ and the method is called MOD. Considering that the computational complexity of the inverse problem in solving problem VIII.4 is $O(n^3)$, it is favorable, for further improvement, to update dictionary D by fixing the other variables. The strategy of the KSVD algorithm rewrites the problem VIII.4 into

$$\begin{aligned}\hat{D} &= \arg\min_{D} \|Y - DX\|_F^2 = \arg\min_{D} \|Y - \sum_{j=1}^{N} \boldsymbol{d}_j \boldsymbol{x}_j^T\|_F^2 \\ &= \arg\min_{D} \|(Y - \sum_{j \neq l} \boldsymbol{d}_j \boldsymbol{x}_j^T) - \boldsymbol{d}_l \boldsymbol{x}_l^T\|_F^2\end{aligned} \tag{VIII.5}$$

where $\boldsymbol{x}_j$ is the j-th row vector of the matrix X. First the overall representation residual $E_l = Y - \sum_{j \neq l} \boldsymbol{d}_j \boldsymbol{x}_j^T$ is computed, and then $\boldsymbol{d}_l$ and $\boldsymbol{x}_l$ are updated. In order to maintain the sparsity of $\boldsymbol{x}_l^T$ in this step, only the nonzero elements of $\boldsymbol{x}_l^T$ should be preserved and only the nonzero items of E_l should be reserved, i.e. E_l^P, from $\boldsymbol{d}_l \boldsymbol{x}_l^T$. Then, SVD decomposes E_l^P into $E_l^P = U \Lambda V^T$, and then updates dictionary $\boldsymbol{d}_l$. The specific KSVD algorithm for dictionary learning is summarized to Algorithm 12 and more information can be found in the literature [122].

(2) Locality constrained linear coding for unsupervised dictionary learning

The locality constrained linear coding (LLC) algorithm [130] is an efficient local coordinate linear coding method, which projects each descriptor into a local constraint system to obtain an effective codebook or dictionary. It has been demonstrated that the property of locality is more essential than sparsity, because the locality must lead to sparsity but not vice-versa, that is, a necessary condition of sparsity is locality, but not the reverse [130].

Assume that $Y = [\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_N] \in \mathbb{R}^{d \times N}$ is a matrix composed of local descriptors extracted from examples and the objective dictionary $D = [\boldsymbol{d}_1, \boldsymbol{d}_2, \cdots, \boldsymbol{d}_N] \in \mathbb{R}^{d \times N}$. The objective function of LLC is formulated as

$$\begin{aligned}&\arg\min_{x_i, D} \sum_{i=1}^{N} \|\boldsymbol{y}_i - D\boldsymbol{x}_i\|_2^2 + \mu \|\boldsymbol{b} \odot \boldsymbol{x}_i\|_2^2 \\ &s.t. \quad \mathbf{1}^T \boldsymbol{x}_i = 1,\ i = 1, 2, \cdots, N\end{aligned} \tag{VIII.6}$$

where μ is a small constant as a regularization parameter for adjusting the weighting decay speed, $\odot$ is the operator of the element-wise multiplication, $\boldsymbol{x}_i$ is the code for $\boldsymbol{y}_i$, $\mathbf{1} \in \mathbb{R}^{N \times 1}$ is defined as a vector with all elements as 1 and vector $\boldsymbol{b}$ is the locality adaptor, which is, more specifically, set as

$$\boldsymbol{b} = \exp\left(\frac{dist(\boldsymbol{y}_i, D)}{\sigma}\right) \tag{VIII.7}$$

where $dist(\boldsymbol{y}_i, D) = [dist(\boldsymbol{y}_i, \boldsymbol{d}_1), \cdots, dist(\boldsymbol{y}_i, \boldsymbol{d}_N)]$ and $dist(\boldsymbol{y}_i, \boldsymbol{d}_j)$ denotes the distance between $\boldsymbol{y}_i$ and $\boldsymbol{d}_j$ with different distance metrics, such as Euclidean distance and Chebyshev distance. Specifically, the i-th value of vector $\boldsymbol{b}$ is defined as $\boldsymbol{b_i} = \exp\left(\frac{dist(\boldsymbol{y}_i, d_i)}{\sigma}\right)$.

The K-Means clustering algorithm is applied to generate the codebook D, and then the solution of LLC can be deduced as:

$$\hat{\boldsymbol{x}}_i = (C_i + \mu\ diag^2(\boldsymbol{b}))\backslash\mathbf{1} \tag{VIII.8}$$

$$\boldsymbol{x}_i = \hat{\boldsymbol{x}}_i / \ \mathbf{1}^T \hat{\boldsymbol{x}}_i \tag{VIII.9}$$

where the operator $a\backslash b$ denotes $a^{-1}b$, and $C_i = (D^T - \mathbf{1}\boldsymbol{y}_i^T)(D^T - \mathbf{1}\boldsymbol{y}_i^T)^T$ is the covariance matrix with respect to $\boldsymbol{y}_i$. This is called the LLC algorithm. Furthermore, the incremental codebook optimization algorithm has also been proposed to obtain a more effective and optimal codebook, and the objective function is reformulated as

$$\begin{aligned}&\arg\min_{x_i, D} \sum_{i=1}^{N} \|\boldsymbol{y}_i - D\boldsymbol{x}_i\|_2^2 + \mu \|\boldsymbol{b} \odot \boldsymbol{x}_i\|_2^2 \\ &s.t. \quad \mathbf{1}^T \boldsymbol{x}_i = 1,\ \forall i; \|\boldsymbol{d}_j\|_2^2 \leq 1, \forall j\end{aligned} \tag{VIII.10}$$

Actually, the problem VIII.10 is a process of feature extraction and the property of 'locality' is achieved by constructing a local coordinate system by exploiting the local bases for each descriptor, and the local bases in the algorithm are simply obtained by using the K nearest neighbors of $\boldsymbol{y}_i$. The incremental codebook optimization algorithm in problem VIII.10 is a joint optimization problem with respect to D and x_i, and it can be solved by iteratively optimizing one when fixing the other alternatively. The main steps of the incremental codebook optimization algorithm are summarized in Algorithm 13 and more information can be found in the literature [130].

(3) Other unsupervised dictionary learning methods

A large number of different unsupervised dictionary learning methods have been proposed. The KSVD algorithm and LLC algorithm are only two typical unsupervised dictionary learning algorithms based on sparse representation. Additionally, Jenatton et al. [123] proposed a tree-structured dictionary learning problem, which exploited tree-structured sparse regularization to model the relationship between each atom and defined a proximal operator to solve the primal-dual problem. Zhou et al. [131] developed a nonparametric Bayesian dictionary learning algorithm, which utilized hierarchical Bayesian to model parameters and employed the truncated beta-Bernoulli process to learn the dictionary. Ramirez and Shapiro [132] employed minimum description length to

Algorithm 13. The incremental codebook optimization algorithm
Task: Learning a dictionary D: $\arg\min_{x_i,D} \sum_{i=1}^{N} \|\boldsymbol{y}_i - D\boldsymbol{x}_i\|_2^2 + \mu\|\boldsymbol{b} \odot \boldsymbol{x}_i\|_2^2 \;\; s.t. \;\; \mathbf{1}^T\boldsymbol{x}_i = 1, \; \forall i; \|\boldsymbol{d}_j\|_2^2 \leq 1, \forall j$

Input: The matrix composed of given samples $Y = [\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_N]$.
Initialization: $i = 1$, $\varepsilon = 0.01$, D initialized by K-Means clustering algorithm.
While i is not equal to N do
Step 1: Initialize $\boldsymbol{b}$ with $1 \times N$ zero vector.
Step 2: Update locality constraint parameter b with
$$\boldsymbol{b}_j = \exp\left(-\frac{dist(\boldsymbol{y}_i, \boldsymbol{d}_j)}{\sigma}\right) \text{ for } \forall j.$$
Step 3: Normalize $\boldsymbol{b}$ using the equation $\boldsymbol{b} = \frac{\boldsymbol{b} - \boldsymbol{b}_{min}}{\boldsymbol{b}_{max} - \boldsymbol{b}_{min}}$.
Step 4: Exploit the LLC coding algorithm to obtain $\boldsymbol{x}_i$.
Step 5: Keep the set of D^i, whose corresponding entries of the code $\boldsymbol{x}_i$ are greater than ε, and drop out other elements, i.e. $index \leftarrow \{j \mid abs\{\boldsymbol{x}_i(j)\} > \varepsilon\} \; \forall j$ and $D^i \leftarrow D(:, index)$.
Step 6: Update x_i exploiting $\arg\max \|\boldsymbol{y}_i - D^i\boldsymbol{x}_i\|_2^2 \;\; s.t. \;\; \mathbf{1}^T\boldsymbol{x}_i = 1$.
Step 7: Update dictionary D using a classical gradient descent method with respect to problem VIII.6.
Step 8: $i = i + 1$.
End While
Output: dictionary D

model an effective framework of sparse representation and dictionary learning, and this framework could conveniently incorporate prior information into the process of sparse representation and dictionary learning. Some other unsupervised dictionary learning algorithms also have been validated. Mairal et al. proposed an online dictionary learning [133] algorithm based on stochastic approximations, which treated the dictionary learning problem as the optimization of a smooth convex problem over a convex set and employed an iterative online algorithm at each step to solve the subproblems. Yang and Zhang proposed a sparse variation dictionary learning (SVDL) algorithm [134] for face recognition with a single training sample, in which a joint learning framework of adaptive projection and a sparse variation dictionary with sparse bases were simultaneously constructed from the gallery image set to the generic image set. Shi et al. proposed a minimax concave penalty based sparse dictionary learning (MCPSDL) [112] algorithm, which employed a non-convex relaxation online scheme, i.e. a minimax concave penalty, instead of using regular convex relaxation approaches as approximation of l_0-norm penalty in sparse representation problem, and designed a coordinate descend algorithm to optimize it. Bao et al. proposed a dictionary learning by proximal algorithm (DLPM) [118], which provided an efficient alternating proximal algorithm for solving the l_0-norm minimization based dictionary learning problem and demonstrated its global convergence property.

2) Supervised dictionary learning: Unsupervised dictionary learning just considers that the examples can be sparsely represented by the learned dictionary and leaves out the label information of the examples. Thus, unsupervised dictionary learning can perform very well in data reconstruction, such as image denoising and image compressing, but is not beneficial to perform classification. On the contrary, supervised dictionary learning embeds the class label into the process of sparse representation and dictionary learning so that this leads to the learned dictionary with discriminative information for effective classification.

(1) Discriminative KSVD for dictionary learning

Discriminative KSVD (DKSVD) [127] was designed to solve image classification problems. Considering the priorities of supervised learning theory in classification, DKSVD incorporates the dictionary learning with discriminative information and classifier parameters into the objective function and employs the KSVD algorithm to obtain the global optimal solution for all parameters. The objective function of the DKSVD algorithm is formulated as

$$\langle D, C, X\rangle = \arg\min_{D,C,X} \|Y - DX\|_F^2 + \mu\|H - CX\|_F^2 + \eta\|C\|_F^2 \quad s.t. \quad \|\boldsymbol{x}_i\|_0 \leq k \tag{VIII.11}$$

where Y is the given input samples, D is the learned dictionary, X is the coefficient term, H is the matrix composed of label information corresponding to Y, C is the parameter term for classifier, and η and μ are the weights. With a view to the framework of KSVD, problem VIII.11 can be rewritten as

$$\langle D, C, X\rangle = \arg\min_{D,C,X} \left\| \begin{pmatrix} Y \\ \sqrt{\mu}H \end{pmatrix} - \begin{pmatrix} D \\ \sqrt{\mu}C \end{pmatrix} X \right\|_F^2 + \eta\|C\|_F^2 \quad s.t. \quad \|\boldsymbol{x}_i\|_0 \leq k \tag{VIII.12}$$

In consideration of the KSVD algorithm, each column of the dictionary will be normalized to l_2-norm unit vector and $\begin{pmatrix} D \\ \sqrt{\mu}C \end{pmatrix}$ will also be normalized, and then the penalty term $\|C\|_F^2$ will be dropped out and problem VIII.12 will be reformulated as

$$\langle Z, X\rangle = \arg\min_{Z,X} \|W - ZX\|_F^2 \quad s.t. \quad \|\boldsymbol{x}_i\|_0 \leq k \tag{VIII.13}$$

where $W = \begin{pmatrix} Y \\ \sqrt{\mu}H \end{pmatrix}$, $Z = \begin{pmatrix} D \\ \sqrt{\mu}C \end{pmatrix}$ and apparently the formulation VIII.13 is the same as the framework of KSVD [122] in Eq. VIII.2 and it can be efficiently solved by the KSVD algorithm.

More specifically, the DKSVD algorithm contains two main phases: the training phase and classification phase. For the training phase, Y is the matrix composed of the training samples and the objective is to learn a discriminative dictionary D and the classifier parameter C. DKSVD updates Z column by column and for each column vector $\boldsymbol{z}_i$, DKSVD employs the KSVD algorithm to obtain $\boldsymbol{z}_i$ and its corresponding weight. Then, the DKSVD algorithm normalizes the dictionary D and classifier parameter C by

$$\begin{aligned} D' &= [\boldsymbol{d}'_1, \boldsymbol{d}'_2, \cdots, \boldsymbol{d}'_M] = [\tfrac{\boldsymbol{d}_1}{\|\boldsymbol{d}_1\|}, \tfrac{\boldsymbol{d}_2}{\|\boldsymbol{d}_2\|}, \cdots, \tfrac{\boldsymbol{d}_M}{\|\boldsymbol{d}_M\|}] \\ C' &= [\boldsymbol{c}'_1, \boldsymbol{c}'_2, \cdots, \boldsymbol{c}'_M] = [\tfrac{\boldsymbol{c}_1}{\|\boldsymbol{d}_1\|}, \tfrac{\boldsymbol{c}_2}{\|\boldsymbol{d}_2\|}, \cdots, \tfrac{\boldsymbol{c}_M}{\|\boldsymbol{d}_M\|}] \\ \boldsymbol{x}'_i &= \boldsymbol{x}_i \times \|\boldsymbol{d}_i\| \end{aligned} \tag{VIII.14}$$

For the classification phase, Y is the matrix composed of the test samples. Based on the obtained learning results D' and C', the sparse coefficient matrix $\hat{\boldsymbol{x}}_i$ can be obtained for

each test sample $\boldsymbol{y}_i$ by exploiting the OMP algorithm, which is to solve

$$\hat{\boldsymbol{x}}_i = \arg\min \|\boldsymbol{y}_i - D'\boldsymbol{x}'_i\|_2^2 \quad s.t. \quad \|\boldsymbol{x}'_i\|_0 \leq k \quad \text{(VIII.15)}$$

On the basis of the corresponding sparse coefficient $\hat{\boldsymbol{x}}_i$, the final classification, for each test sample $\boldsymbol{y}_i$, can be performed by judging the label result by multiplying $\hat{\boldsymbol{x}}_i$ by classifier C', that is,

$$label = C' \times \hat{\boldsymbol{x}}_i \quad \text{(VIII.16)}$$

where the $label$ is the final predicted label vector. The class label of $\boldsymbol{y}_i$ is the determined class index of $label$.

The main highlight of DKSVD is that it employs the framework of KSVD to simultaneously learn a discriminative dictionary and classifier parameter, and then utilizes the efficient OMP algorithm to obtain a sparse representation solution and finally integrate the sparse solution and learned classifier for ultimate effective classification.

(2) Label consistent KSVD for discriminative dictionary learning

Because of the classification term, a competent dictionary can lead to effectively classification results. The original sparse representation for face recognition [20] regards the raw data as the dictionary, and then reports its promising classification results. In this section, a label consistent KSVD (LC-KSVD) [135, 136] is introduced to learn an effective discriminative dictionary for image classification. As an extension of D-KSVD, LC-KSVD exploits the supervised information to learn the dictionary and integrates the process of constructing the dictionary and optimal linear classifier into a mixed reconstructive and discriminative objective function, and then jointly obtains the learned dictionary and an effective classifier. The objective function of LC-KSVD is formulated as

$$\begin{aligned}\langle D, A, C, X\rangle = \arg \min_{D,A,C,X} \|Y - DX\|_F^2 + \mu\|L - AX\|_F^2 \\ +\eta\|H - CX\|_F^2 \ s.t. \ \|\boldsymbol{x}_i\|_0 \leq k\end{aligned} \quad \text{(VIII.17)}$$

where the first term denotes the reconstruction error, the second term denotes the discriminative sparse-code error, and the final term denotes the classification error. Y is the matrix composed of all the input data, D is the learned dictionary, X is the sparse code term, μ and η are the weights of the corresponding contribution items, A is a linear transformation matrix, H is the matrix composed of label information corresponding to Y, C is the parameter term for classifier and L is a joint label matrix for labels of Y and D. For example, providing that $Y = [y_1 \dots y_4]$ and $D = [d_1 \dots d_4]$ where y_1, y_2, d_1 and d_2 are from the first class, and y_3, y_4, d_3 and d_4 are from the second class, and then the joint label matrix L can be defined as $L = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$. Similar to the DKSVD algorithm, the objective function VIII.17 can also be reformulated as

$$\langle Z, X\rangle = \arg\min_{Z,X} \|T - ZX\|_2^2 \quad s.t. \quad \|\boldsymbol{x}_i\|_0 \leq k \quad \text{(VIII.18)}$$

where $T = \begin{pmatrix} Y \\ \sqrt{\mu}L \\ \sqrt{\eta}H \end{pmatrix}$, $Z = \begin{pmatrix} D \\ \sqrt{\mu}A \\ \sqrt{\eta}C \end{pmatrix}$.

The learning process of the LC-KSVD algorithm, as is DKSVD, can be separated into two sections, the training term and the classification term. In the training section, since problem VIII.18 completely satisfies the framework of KSVD, the KSVD algorithm is applied to update Z atom by atom and compute X. Thus Z and X can be obtained. Then, the LC-KSVD algorithm normalizes dictionary D, transform matrix A, and the classifier parameter C by

$$\begin{aligned} D' &= [\boldsymbol{d}'_1, \boldsymbol{d}'_2, \cdots, \boldsymbol{d}'_M] = [\tfrac{\boldsymbol{d}_1}{\|\boldsymbol{d}_1\|}, \tfrac{\boldsymbol{d}_2}{\|\boldsymbol{d}_2\|}, \cdots, \tfrac{\boldsymbol{d}_M}{\|\boldsymbol{d}_M\|}] \\ A' &= [\boldsymbol{a}'_1, \boldsymbol{a}'_2, \cdots, \boldsymbol{a}'_M] = [\tfrac{\boldsymbol{a}_1}{\|\boldsymbol{d}_1\|}, \tfrac{\boldsymbol{a}_2}{\|\boldsymbol{d}_2\|}, \cdots, \tfrac{\boldsymbol{a}_M}{\|\boldsymbol{d}_M\|}] \\ C' &= [\boldsymbol{c}'_1, \boldsymbol{c}'_2, \cdots, \boldsymbol{c}'_M] = [\tfrac{\boldsymbol{c}_1}{\|\boldsymbol{d}_1\|}, \tfrac{\boldsymbol{c}_2}{\|\boldsymbol{d}_2\|}, \cdots, \tfrac{\boldsymbol{c}_M}{\|\boldsymbol{d}_M\|}] \end{aligned} \quad \text{(VIII.19)}$$

In the classification section, Y is the matrix composed of the test samples. On the basis of the obtained dictionary D', the sparse coefficient $\hat{\boldsymbol{x}}_i$ can be obtained for each test sample $\boldsymbol{y}_i$ by exploiting the OMP algorithm, which is to solve

$$\hat{\boldsymbol{x}}_i = \arg\min \|\boldsymbol{y}_i - D'\boldsymbol{x}'_i\|_2^2 \quad s.t. \quad \|\boldsymbol{x}'_i\|_0 \leq k \quad \text{(VIII.20)}$$

The final classification is based on a simple linear predictive function

$$l = \arg\max_f \{f = C' \times \hat{\boldsymbol{x}}_i\} \quad \text{(VIII.21)}$$

where f is the final predicting label vector and the test sample $\boldsymbol{y}_i$ is classified as a member of the l-th class.

The main contribution of LC-KSVD is to jointly incorporate the discriminative sparse coding term and classifier parameter term into the objective function for learning a discriminative dictionary and classifier parameter. The LC-KSVD demonstrates that the obtained solution, compared to other methods, can prevent learning a suboptimal or local optimal solution in the process of learning a dictionary [135].

(3) Fisher discrimination dictionary learning for sparse representation

Fisher discrimination dictionary learning (FDDL) [137] incorporates the supervised information (class label information) and the Fisher discrimination message into the objective function for learning a structured discriminative dictionary, which is used for pattern classification. The general model of FDDL is formulated as

$$J(D, X) = \arg\min_{D,X} \{f(Y, D, X) + \mu\|X\|_1 + \eta g(X)\} \quad \text{(VIII.22)}$$

where Y is the matrix composed of input data, D is the learned dictionary, X is the sparse solution, and μ and η are two constants for tradeoff contributions. The first component is the discriminative fidelity term, the second component is the sparse regularization term, and the third component is the discriminative coefficient term, such as Fisher discrimination criterion in Eq. (VIII.23).

Considering the importance of the supervised information, i.e. label information, in classification, FDDL respectively updates the dictionary and computes the sparse representation solution class by class. Assume that Y^i denotes the matrix of i-th class of input data, vector X^i denotes the sparse representation coefficient of the learned dictionary D over Y^i and X^i_j denotes the matrix composed of the sparse representation solutions, which correspond to the j-th class coefficients from X^i. D^i is denoted as the learned dictionary corresponding to the i-th class. Thus, the objective function of FDDL is

$$J(D,X) = \arg\min_{D,X}(\sum_{i=1}^{c} f(Y^i, D, X^i) + \mu\|X\|_1 + \eta(tr(S_W(X) - S_B(X)) + \lambda\|X\|_F^2)) \quad \text{(VIII.23)}$$

where $f(Y^i, D, X^i) = \|Y^i - DX^i\|_F^2 + \|Y^i - D^iX^i_i\|_F^2 + \sum_{j\neq i}\|D^jX^i_j\|_F^2$ and $S_W(X)$ and $S_B(X)$ are the within-class scatter of X and between-class scatter of X, respectively. c is the number of the classes. To solve problem VIII.23, a natural idea of optimization is to alternatively optimize D and X class by class, and then the process of optimization is briefly introduced.

When fixing D, problem VIII.23 can be solved by computing X^i class by class, and its sub-problem is formulated as

$$J(X^i) = \arg\min_{X^i}\left(f(Y^i, D, X^i) + \mu\|X^i\|_1 + \eta g(X^i)\right) \quad \text{(VIII.24)}$$

where $g(X^i) = \|X^i - M_i\|_F^2 - \sum_{t=1}^{c}\|M_t - M\|_F^2 + \lambda\|X^i\|_F^2$ and M_j and M denote the mean matrices corresponding to the j-th class of X^i and X^i, respectively. Problem VIII.23 can be solved by the iterative projection method in the literature [138].

When fixing $\boldsymbol{\alpha}$, problem VIII.23 can be rewritten as

$$J(D) = \arg\min_{D}(\|Y^i - D^iX^i - \sum_{j\neq i} D^jX^j\|_F^2 + \|Y^i - D^iX^i_i\|_F^2 + \sum_{j\neq i}\|D^iX^i_j\|_F^2) \quad \text{(VIII.25)}$$

where X^i here denotes the sparse representation of Y over D^i. In this section, each column of the learned dictionary is normalized to a unit vector with l_2-norm. The optimization of problem VIII.25 computes the dictionary class by class and it can be solved by exploiting the algorithm in the literature [139].

The main contribution of the FDDL algorithm lies in combining the Fisher discrimination criterion into the process of dictionary learning. The discriminative power comes from the method of constructing the discriminative dictionary using the function f in problem VIII.22 and simultaneously formulates the discriminative sparse representation coefficients by exploiting the function g in problem VIII.22.

(4) Other supervised dictionary learning for sparse representation

Unlike unsupervised dictionary learning, supervised dictionary learning emphasizes the significance of the class label information and incorporates it into the learning process to enforce the discrimination of the learned dictionary. Recently, massive supervised dictionary learning algorithms have been proposed. For example, Yang et al. [139] presented a metaface dictionary learning method, which is motivated by 'metagenes' in gene expression data analysis. Rodriguez and Sapiro [140] produced a discriminative non-parametric dictionary learning (DNDL) framework based on the OMP algorithm for image classification. Kong et al. [141] introduced a learned dictionary with commonalty and particularity, called DL-COPAR, which integrated an incoherence penalty term into the objective function for obtaining the class-specific sub-dictionary. Gao et al. [142] learned a hybrid dictionary, i.e. category-specific dictionary and shared dictionary, which incorporated a cross-dictionary incoherence penalty and self-dictionary incoherence penalty into the objective function for learning a discriminative dictionary. Jafari and Plumbley [143] presented a greedy adaptive dictionary learning method, which updated the learned dictionary with a minimum sparsity index. Some other supervised dictionary learning methods are also competent in image classification, such as supervised dictionary learning in [144]. Zhou et al. [145] developed a joint dictionary learning algorithm for object categorization, which jointly learned a commonly shared dictionary and multiply category-specific dictionaries for correlated object classes and incorporated the Fisher discriminant fidelity term into the process of dictionary learning. Ramirez et al. proposed a method of dictionary learning with structured incoherence (DLSI) [140], which unified the dictionary learning and sparse decomposition into a sparse dictionary learning framework for image classification and data clustering. Ma et al. presented a discriminative low-rank dictionary learning for sparse representation (DLRD_SR) [146], in which the sparsity and the low-rank properties were integrated into one dictionary learning scheme where sub-dictionary with discriminative power was required to be low-rank. Lu et al. developed a simultaneous feature and dictionary learning [147] method for face recognition, which jointly learned the feature projection matrix for subspace learning and the discriminative structured dictionary. Yang et al. introduced a latent dictionary learning (LDL) [148] method for sparse representation based image classification, which simultaneously learned a discriminative dictionary and a latent representation model based on the correlations between label information and dictionary atoms. Jiang et al. presented a submodular dictionary learning (SDL) [149] method, which integrated the entropy rate of a random walk on a graph and a discriminative term into a unified objective function and devised a greedy-based approach to optimize it. Si et al. developed a support vector guided dictionary learning (SVGDL) [150] method, which constructed a discriminative term by using adaptively weighted summation of the squared distances for all pairwise of the sparse representation solutions.

B. Sparse representation in image processing

Recently, sparse representation methods have been extensively applied to numerous real-world applications [151, 152]. The techniques of sparse representation have been gradually ex-

tended and introduced to image processing, such as super-resolution image processing, image denoising and image restoration.

First, the general framework of image processing using sparse representation especially for image reconstruction should be introduced:

Step 1: Partition the degraded image into overlapped patches or blocks.

Step 2: Construct a dictionary, denoted as D, and assume that the following sparse representation formulation should be satisfied for each patch or block x of the image:

$\hat{\boldsymbol{\alpha}} = arg\min \|\boldsymbol{\alpha}\|_p \quad s.t. \quad \|x - HD\boldsymbol{\alpha}\|_2^2 \leq \varepsilon$

where H is a degradation matrix and $0 \leq p \leq 1$.

Step 3: Reconstruct each patch or block by exploiting $\hat{x} = D\hat{\boldsymbol{\alpha}}$.

Step 4: Put the reconstructed patch to the image at the corresponding location and average each overlapped patches to make the reconstructed image more consistent and natural.

Step 5: Repeat step 1 to 4 several times till a termination condition is satisfied.

The following part of this subsection is to explicitly introduce some image processing techniques using sparse representation.

The main task of super-resolution image processing is to extract the high super-resolution image from its low resolution counterpart and this challenging problem has attracted much attention. The most representative work was proposed to exploit the sparse representation theory to generate a super-resolution (SRSR) image from a single low-resolution image in literature [153].

SRSR is mainly performed on two compact learned dictionaries D_l and D_h, which are denoted as dictionaries of low-resolution image patches and its corresponding high-resolution image patches, respectively. D_l is directly employed to recover high-resolution images from dictionary D_h. Let X and Y denote the high-resolution and its corresponding low-resolution images, respectively. $\boldsymbol{x}$ and $\boldsymbol{y}$ are a high-resolution image patch and its corresponding low-resolution image patch, respectively. Thus, $\boldsymbol{x} = P\boldsymbol{y}$ and P is the projection matrix. Moreover, if the low resolution image Y is produced by downsampling and blurring from the high resolution image X, the following reconstruction constraint should be satisfied

$$Y = SBX \tag{VIII.26}$$

where S and B are a downsampling operator and a blurring filter, respectively. However, the solution of problem VIII.26 is ill-posed because infinite solutions can be achieved for a given low-resolution input image Y. To this end, SRSR [153] provides a prior knowledge assumption, which is formulated as

$$\boldsymbol{x} = D_h\boldsymbol{\alpha} \quad s.t. \quad \|\boldsymbol{\alpha}\|_0 \leq k \tag{VIII.27}$$

where k is a small constant. This assumption gives a prior knowledge condition that any image patch $\boldsymbol{x}$ can be approximately represented by a linear combination of a few training samples from dictionary D_h. As presented in Subsection III-B, problem VIII.27 is an NP-hard problem and sparse representation with l_1-norm regularization is introduced. If the desired representation solution $\boldsymbol{\alpha}$ is sufficiently sparse, problem VIII.27 can be converted into the following problem:

$$arg\min \|\boldsymbol{\alpha}\|_1 \quad s.t. \quad \|\boldsymbol{x} - D_h\boldsymbol{\alpha}\|_2^2 \leq \varepsilon \tag{VIII.28}$$

or

$$arg\min \|\boldsymbol{x} - D_h\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1 \tag{VIII.29}$$

where ε is a small constant and λ is the Lagrange multiplier. The solution of problem VIII.28 can be achieved by two main phases, i.e. local model based sparse representation (LMBSR) and enhanced global reconstruction constraint. The first phase of SRSR, i.e. LMBSR, is operated on each image patch, and for each low-resolution image patch $\boldsymbol{y}$, the following equation is satisfied

$$arg\min \|F\boldsymbol{y} - FD_l\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1 \tag{VIII.30}$$

where F is a feature extraction operator. One-pass algorithm similar to that of [154] is introduced to enhance the compatibility between adjacent patches. Furthermore, a modified optimization problem is proposed to guarantee that the super-resolution reconstruction coincides with the previously obtained adjacent high-resolution patches, and the problem is reformulated as

$$arg\min \|\boldsymbol{\alpha}\|_1 \; s.t. \; \|F\boldsymbol{y} - FD_l\boldsymbol{\alpha}\|_2^2 \leq \varepsilon_1; \; \|\boldsymbol{v} - LD_h\boldsymbol{\alpha}\|_2^2 \leq \varepsilon_2 \tag{VIII.31}$$

where $\boldsymbol{v}$ is the previously obtained high-resolution image on the overlap region, and L refers to the region of overlap between the current patch and previously obtained high-resolution image. Thus problem VIII.31 can be rewritten as

$$arg\min \; \|\hat{\boldsymbol{y}} - D\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1 \tag{VIII.32}$$

where $\hat{\boldsymbol{y}} = \begin{bmatrix} F\boldsymbol{y} \\ \boldsymbol{v} \end{bmatrix}$ and $D = \begin{bmatrix} FD_l \\ LD_h \end{bmatrix}$. Problem VIII.32 can be simply solved by previously introduced solution of the sparse representation with l_1-norm minimization. Assume that the optimal solution of problem VIII.32, i.e. $\boldsymbol{\alpha}^*$, is achieved, the high-resolution patch can be easily reconstructed by $\boldsymbol{x} = D_h\boldsymbol{\alpha}^*$.

The second phase of SRSR enforces the global reconstruction constraint to eliminate possible unconformity or noise from the first phase and make the obtained image more consistent and compatible. Suppose that the high-resolution image obtained by the first phase is denoted as matrix X_0, we project X_0 onto the solution space of the reconstruction constraint VIII.26 and the problem is formulated as follows

$$X^* = arg\min \|X - X_0\|_2^2 \quad s.t. \quad Y = SBX \tag{VIII.33}$$

Problem VIII.33 can be solved by the back-projection method in [155] and the obtained image X^* is regarded as the final optimal high-resolution image. The entire super-resolution via sparse representation is summarized in algorithm 14 and more information can be found in the literature [153].

Furthermore, extensive other methods based on sparse representation have been proposed to solve the super-resolution image processing problem. For example, Yang et al. presented a modified version called joint dictionary learning via sparse representation (JDLSR) [156], which jointly learned

Algorithm 14. Super-resolution via sparse representation

Input: Training image patches dictionaries D_l and D_h, a low-resolution image Y.
For each overlapped 3×3 patches $\boldsymbol{y}$ of Y using one-pass algorithm, from left to right and top to bottom
Step 1: Compute optimal sparse representation coefficients $\boldsymbol{\alpha}^*$ in problem (VIII.32).
Step 2: Compute the high-resolution patch by $\boldsymbol{x} = D_h \boldsymbol{\alpha}^*$.
Step 3: Put the patch $\boldsymbol{x}$ into a high-resolution image X_0 in corresponding location.
End
Step 4: Compute the final super-resolution image X^* in problem (VIII.33).
Output: X^*

two dictionaries that enforced the similarity of sparse representation for low-resolution and high-resolution images. Tang et al. [157] first explicitly analyzed the rationales of the sparse representation theory in performing the super-resolution task, and proposed to exploit the L_2-Boosting strategy to learn coupled dictionaries, which were employed to construct sparse coding space. Zhang et al. [158] presented an image super-resolution reconstruction scheme by employing the dual-dictionary learning and sparse representation method for image super-resolution reconstruction and Gao et al. [159] proposed a sparse neighbor embedding method, which incorporated the sparse neighbor search and HoG clustering method into the process of image super-resolution reconstruction. Fernandez-Granda and Candes [160] designed a transform-invariant group sparse regularizer by implementing a data-driven non-parametric regularizers with learned domain transform on group sparse representation for high image super-resolution. Lu et al. [161] proposed a geometry constrained sparse representation method for single image super-resolution by jointly obtaining an optimal sparse solution and learning a discriminative and reconstructive dictionary. Dong et al. [162] proposed to harness an adaptive sparse optimization with nonlocal regularization based on adaptive principal component analysis enhanced by nonlocal similar patch grouping and nonlocal self-similarity quadratic constraint to solve the image high super-resolution problem. Dong et al. [163] proposed to integrate an adaptive sparse domain selection and an adaptive regularization based on piecewise autoregressive models into the sparse representations framework for single image super-resolution reconstruction. Mallat and Yu [164] proposed a sparse mixing estimator for image super-resolution, which introduced an adaptive estimator models by combining a group of linear inverse estimators based on different prior knowledge for sparse representation.

Noise in an image is unavoidable in the process of image acquisition. The need for sparse representation may arise when noise exists in image data. In such a case, the image with noise may lead to missing information or distortion such that this results in a decrease of the precision and accuracy of image processing. Eliminating such noise is greatly beneficial to many applications. The main goal of image denoising is to distinguish the actual signal and noise signal so that we can remove the noise and reconstruct the genuine image. In the presence of image sparsity and redundancy representation [4, 7], sparse representation for image denoising first extracts the sparse image components, which are regarded as useful information, and then abandons the representation residual, which is treated as the image noise term, and finally reconstructs the image exploiting the pre-obtained sparse components, i.e. noise-free image. Extensive research articles for image denoising based on sparse representation have been published. For example, Donoho [8, 29, 165] first discovered the connection between the compressed sensing and image denoising. Subsequently, the most representative work of using sparse representation to make image denoising was proposed in literature [166], in which a global sparse representation model over learned dictionaries (SRMLD) was used for image denoising. The following prior assumption should be satisfied: every image block of image x, denoted as z, can be sparsely represented over a dictionary D, i.e. the solution of the following problem is sufficiently sparse:

$$arg \min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_0 \quad s.t. \quad D\boldsymbol{\alpha} = \boldsymbol{z} \tag{VIII.34}$$

And an equivalent problem can be reformulated for a proper value of λ, i.e.

$$arg \min_{\boldsymbol{\alpha}} \|D\boldsymbol{\alpha} - \boldsymbol{z}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_0 \tag{VIII.35}$$

If we take the above prior knowledge into full consideration, the objective function of SRMLD based on Bayesian treatment is formulated as

$$arg \min_{D, \boldsymbol{\alpha}_i, \boldsymbol{x}} \delta \|\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \sum_{i=1}^{M} \|D\boldsymbol{\alpha}_i - P_i \boldsymbol{x}\|_2^2 + \sum_{i=1}^{M} \lambda_i \|\boldsymbol{\alpha}_i\|_0 \tag{VIII.36}$$

where $\boldsymbol{x}$ is the finally denoised image, $\boldsymbol{y}$ the measured image with white and additive Gaussian white noise, P_i is a projection operator that extracts the i-th block from image x, M is the number of the overlapping blocks, D is the learned dictionary, $\boldsymbol{\alpha}_i$ is the coefficients vector, δ is the weight of the first term and λ_i is the Lagrange multiplier. The first term in VIII.36 is the log-likelihood global constraint such that the obtained noise-free image x is sufficiently similar to the original image y. The second and third terms are the prior knowledge of the Bayesian treatment, which is presented in problem VIII.35. The optimization of problem VIII.35 is a joint optimization problem with respect to D, $\boldsymbol{\alpha}_i$ and x. It can be solved by alternatively optimizing one variable when fixing the others. The process of optimization is briefly introduced below.

When dictionary D and the solution of sparse representation $\boldsymbol{\alpha}_i$ are fixed, problem VIII.36 can be rewritten as

$$arg \min_{x} \delta \|\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \sum_{i=1}^{M} \|D\boldsymbol{\alpha}_i - \boldsymbol{z}\|_2^2 \tag{VIII.37}$$

where $\boldsymbol{z} = P_i \boldsymbol{x}$. Apparently, problem VIII.37 is a simple convex optimization problem and has a closed-form solution, which is given by

$$\boldsymbol{x} = \left(\sum_{i=1}^{M} P_i^T P_i + \delta I \right)^{-1} \left(\sum_{i=1}^{M} P_i^T D\boldsymbol{\alpha}_i + \delta \boldsymbol{y} \right)^{-1} \tag{VIII.38}$$

When $\boldsymbol{x}$ is given, problem (VIII.36) can be written as

$$arg \min_{D,\boldsymbol{\alpha}_i} \sum_{i=1}^{M} \|D\boldsymbol{\alpha}_i - P_i\boldsymbol{x}\|_2^2 + \sum_{i=1}^{M} \lambda_i \|\boldsymbol{\alpha}_i\|_0 \qquad \text{(VIII.39)}$$

where the problem can be divided into M sub-problems and the i-th sub-problem can be reformulated as the following dictionary learning problem:

$$arg \min_{D,\boldsymbol{\alpha}_i} \|D\boldsymbol{\alpha}_i - \boldsymbol{z}\|_2^2 \quad s.t. \quad \|\boldsymbol{\alpha}_i\|_0 \leq \tau \qquad \text{(VIII.40)}$$

where $\boldsymbol{z} = P_i\boldsymbol{x}$ and τ is small constant. One can see that the sub-problem VIII.39 is the same as problem VIII.2 and it can be solved by the KSVD algorithm previously presented in Subsection VIII-A2. The algorithm of image denoising exploiting sparse and redundant representation over learned dictionary is summarized in Algorithm 15, and more information can be found in literature [166].

Algorithm 15. Image denoising via sparse and redundant representation over learned dictionary
Task: To denoise a measured image y from white and additional Gaussian white noise:
$arg \min_{D,\boldsymbol{\alpha}_i,x} \delta\|\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \sum_{i=1}^{M} \|D\boldsymbol{\alpha}_i - P_i\boldsymbol{x}\|_2^2 + \sum_{i=1}^{M} \lambda_i \|\boldsymbol{\alpha}_i\|_0$

Input: Measured image sample $\boldsymbol{y}$, the number of training iteration T.
Initialization: $t = 1$, set $\boldsymbol{x} = \boldsymbol{y}$, D initialized by an overcomplete DCT dictionary.
While $t \leq T$ do
Step 1: For each image patch $P_i\boldsymbol{x}$, employ the KSVD algorithm to update the values of sparse representation solution $\boldsymbol{\alpha}_i$ and corresponding dictionary D.
Step 2: $t = t + 1$
End While
Step 3: Compute the value of $\boldsymbol{x}$ by using Eq.(VIII.38).
Output: denoised image $\boldsymbol{x}$

Moreover, extensive modified sparse representation based image denoising algorithms have been proposed. For example, Dabov et al. [167] proposed an enhanced sparse representation with a block-matching 3-D (BM3D) transform-domain filter based on self-similarities and an enhanced sparse representation by clustering similar 2-D image patches into 3-D data spaces and an iterative collaborative filtering procedure for image denoising. Mariral et al. [168] proposed the use of extending the KSVD-based grayscale algorithm and a generalized weighted average algorithm for color image denoising. Protter and Elad [169] extended the techniques of sparse and redundant representations for image sequence denoising by exploiting spatio-temporal atoms, dictionary propagation over time and dictionary learning. Dong et al. [170] designed a clustering based sparse representation algorithm, which was formulated by a double-header sparse optimization problem built upon dictionary learning and structural clustering. Recently, Jiang et al. [171] proposed a variational encoding framework with a weighted sparse nonlocal constraint, which was constructed by integrating image sparsity prior and nonlocal self-similarity prior into a unified regularization term to overcome the mixed noise removal problem. Gu et al. [172] studied a weighted nuclear norm minimization (WNNM) method with F-norm fidelity under different weighting rules optimized by non-local self-similarity for image denoising. Ji et al. [173] proposed a patch-based video denoising algorithm by stacking similar patches in both spatial and temporal domain to formulate a low-rank matrix problem with the nuclear norm. Cheng et al. [174] proposed an impressive image denoising method based on an extension of the KSVD algorithm via group sparse representation.

The primary purpose of image restoration is to recover the original image from the degraded or blurred image. The sparse representation theory has been extensively applied to image restoration. For example, Bioucas-Dias and Figueirdo [175] introduced a two-step iterative shrinkage/thresholding (TwIST) algorithm for image restoration, which is more efficient and can be viewed as an extension of the IST method. Mairal et al. [176] presented a multiscale sparse image representation framework based on the KSVD dictionary learning algorithm and shift-invariant sparsity prior knowledge for restoration of color images and video image sequence. Recently, Mairal et al. [177] proposed a learned simultaneous sparse coding (LSSC) model, which integrated sparse dictionary learning and nonlocal self-similarities of natural images into a unified framework for image restoration. Zoran and Weiss [178] proposed an expected patch log likelihood (EPLL) optimization model, which restored the image from patch to the whole image based on the learned prior knowledge of any patch acquired by Maximum A-Posteriori estimation instead of using simple patch averaging. Bao et al. [179] proposed a fast orthogonal dictionary learning algorithm, in which a sparse image representation based orthogonal dictionary was learned in image restoration. Zhang et al. [180] proposed a group-based sparse representation, which combined characteristics from local sparsity and nonlocal self-similarity of natural images to the domain of the group. Dong et al. [181, 182] proposed a centralized sparse representation (CSR) model, which combined the local and nonlocal sparsity and redundancy properties for variational problem optimization by introducing a concept of sparse coding noise term.

Here we mainly introduce a recently proposed simple but effective image restoration algorithm CSR model [181]. For a degraded image $\boldsymbol{y}$, the problem of image restoration can be formulated as

$$\boldsymbol{y} = H\boldsymbol{x} + \boldsymbol{v} \qquad \text{(VIII.41)}$$

where H is a degradation operator, $\boldsymbol{x}$ is the original high-quality image and $\boldsymbol{v}$ is the Gaussian white noise. Suppose that the following two sparse optimization problems are satisfied

$$\boldsymbol{\alpha}_x = arg \min \|\boldsymbol{\alpha}\|_1 \quad s.t. \quad \|\boldsymbol{x} - D\boldsymbol{\alpha}\|_2^2 \leq \varepsilon \qquad \text{(VIII.42)}$$

$$\boldsymbol{\alpha}_y = arg \min \|\boldsymbol{\alpha}\|_1 \quad s.t. \quad \|\boldsymbol{x} - HD\boldsymbol{\alpha}\|_2^2 \leq \varepsilon \qquad \text{(VIII.43)}$$

where y and x respectively denote the degraded image and original high-quality image, and ε is a small constant. A new concept called sparse coding noise (SCN) is defined

$$\boldsymbol{v}_\alpha = \boldsymbol{\alpha}_y - \boldsymbol{\alpha}_x \qquad \text{(VIII.44)}$$

Given a dictionary D, minimizing SCN can make the image better reconstructed and improve the quality of the image restoration because $\boldsymbol{x}^* = \hat{\boldsymbol{x}} - \tilde{\boldsymbol{x}} = D\boldsymbol{\alpha}_y - D\boldsymbol{\alpha}_x = Dv_\alpha$.

Thus, the objective function is reformulated as

$$\boldsymbol{\alpha}_y = arg\min_{\boldsymbol{\alpha}} \|y - HD\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1 + \mu\|\boldsymbol{\alpha} - \boldsymbol{\alpha}_x\|_1 \tag{VIII.45}$$

where λ and μ are both constants. However, the value of $\boldsymbol{\alpha}_x$ is difficult to directly evaluate. Because many nonlocal similar patches are associated with the given image patch i, clustering these patches via block matching is advisable and the sparse code of searching similar patch l to patch i in cluster Ω_i, denoted by $\boldsymbol{\alpha}_{il}$, can be computed. Moreover, the unbiased estimation of $\boldsymbol{\alpha}_x$, denoted by $E[\boldsymbol{\alpha}_x]$, empirically can be approximate to $\boldsymbol{\alpha}_x$ under some prior knowledge [181], and then SCN algorithm employs the nonlocal means estimation method [183] to evaluate the unbiased estimation of $\boldsymbol{\alpha}_x$, that is, using the weighted average of all $\boldsymbol{\alpha}_{il}$ to approach $E[\boldsymbol{\alpha}_x]$, i.e.

$$\boldsymbol{\theta}_i = \sum_{l\in\Omega_i} w_{il}\boldsymbol{\alpha}_{il} \tag{VIII.46}$$

where $w_{il} = \exp\left(-\|\boldsymbol{x}_i - \boldsymbol{x}_{il}\|_2^2/h\right)/N$, $\boldsymbol{x}_i = D\boldsymbol{\alpha}_i$, $\boldsymbol{x}_{il} = D\boldsymbol{\alpha}_{il}$, N is a normalization parameter and h is a constant. Thus, the objective function VIII.45 can be rewritten as

$$\boldsymbol{\alpha}_y = arg\min_{\boldsymbol{\alpha}} \|\boldsymbol{y} - HD\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1 + \mu\sum_{i=1}^{M}\|\boldsymbol{\alpha}_i - \boldsymbol{\theta}_i\|_1 \tag{VIII.47}$$

where M is the number of the separated patches. In the j-th iteration, the solution of problem VIII.47 is iteratively performed by

$$\boldsymbol{\alpha}_y^{j+1} = arg\min_{\boldsymbol{\alpha}} \|\boldsymbol{y} - HD\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1 + \mu\sum_{i=1}^{M}\|\boldsymbol{\alpha}_i - \boldsymbol{\theta}_i^j\|_1 \tag{VIII.48}$$

It is obvious that problem VIII.47 can be optimized by the augmented Lagrange multiplier method [184] or the iterative shrinkage algorithm in [185]. According to the maximum average posterior principle and the distribution of the sparse coefficients, the regularization parameter λ and constant μ can be adaptively determined by $\lambda = \frac{2\sqrt{2}\rho^2}{\sigma_i}$ and $\mu = \frac{2\sqrt{2}\rho^2}{\eta_i}$, where ρ, σ_i and η_i are the standard deviations of the additive Gaussian noise, $\boldsymbol{\alpha}_i$ and the SCN signal, respectively. Moreover, in the process of image patches clustering for each given image patch, a local PCA dictionary is learned and employed to code each patch within its corresponding cluster. The main procedures of the CSR algorithm are summarized in Algorithm 16 and readers may refer to literature [181] for more details.

Algorithm 16. Centralized sparse representation for image restoration

Initialization: Set $\boldsymbol{x} = \boldsymbol{y}$, initialize regularization parameter λ and μ, the number of training iteration T, $t = 0$, $\boldsymbol{\theta}^0 = 0$.
Step 1: Partition the degraded image into M overlapped patches.
While $t \leq T$ do
Step 2: For each image patch, update the corresponding dictionary for each cluster via k-means and PCA.
Step 3: Update the regularization parameters λ and μ by using
$$\lambda = \frac{2\sqrt{2}\rho^2}{\sigma_t} \text{ and } \mu = \frac{2\sqrt{2}\rho^2}{\eta_t}.$$
Step 4: Compute the nonlocal means estimation of the unbiased estimation of $\boldsymbol{\alpha}_x$, i.e. $\boldsymbol{\theta}_i^{t+1}$, by using Eq. (VIII.46) for each image patch.
Step 5: For a given $\boldsymbol{\theta}_i^{t+1}$, compute the sparse representation solution, i.e. $\boldsymbol{\alpha}_y^{t+1}$, in problem (VIII.48) by using the extended iterative shrinkage algorithm in literature [185].
Step 6: $t = t + 1$
End While
Output: Restored image $\boldsymbol{x} = D\boldsymbol{\alpha}_y^{t+1}$

C. Sparse representation in image classification and visual tracking

In addition to these effective applications in image processing, several other fields for sparse representation have been proposed and extensively studied in image classification and visual tracking. Since Wright et al. [20] proposed to employ sparse representation to perform robust face recognition, more and more researchers have been applying the sparse representation theory to the fields of computer vision and pattern recognition, especially in image classification and object tracking. Experimental results have suggested that the sparse representation based classification method can somewhat overcome the challenging issues from illumination changes, random pixel corruption, large block occlusion or disguise.

As face recognition is a representative component of pattern recognition and computer vision applications, the applications of sparse representation in face recognition can sufficiently reveal the potential nature of sparse representation. The most representative sparse representation for face recognition has been presented in literature [18] and the general scheme of sparse representation based classification method is summarized in Algorithm 17. Suppose that there are n training samples, $X = [x_1, x_2, \cdots, x_n]$ from c classes. Let X_i denote the samples from the i-th class and the testing sample is y.

Algorithm 17. The scheme of sparse representation based classification method

Step 1: Normalize all the samples to have unit l_2-norm.
Step 2: Exploit the linear combination of all the training samples to represent the test sample and the following l_1-norm minimization problem is satisfied
$$\boldsymbol{\alpha}^* = arg\min\|\boldsymbol{\alpha}\|_1 \ s.t. \ \|\boldsymbol{y} - X\boldsymbol{\alpha}\|_2^2 \leq \varepsilon.$$
Step 3: Compute the representation residual for each class
$$\boldsymbol{r}_i = \|\boldsymbol{y} - X_i\boldsymbol{\alpha}_i^*\|_2^2$$
where $\boldsymbol{\alpha}_i^*$ here denotes the representation coefficients vector associated with the i-th class.
Step 4: Output the identity of the test sample y by judging
$$label(\boldsymbol{y}) = arg\min_i(\boldsymbol{r}_i).$$

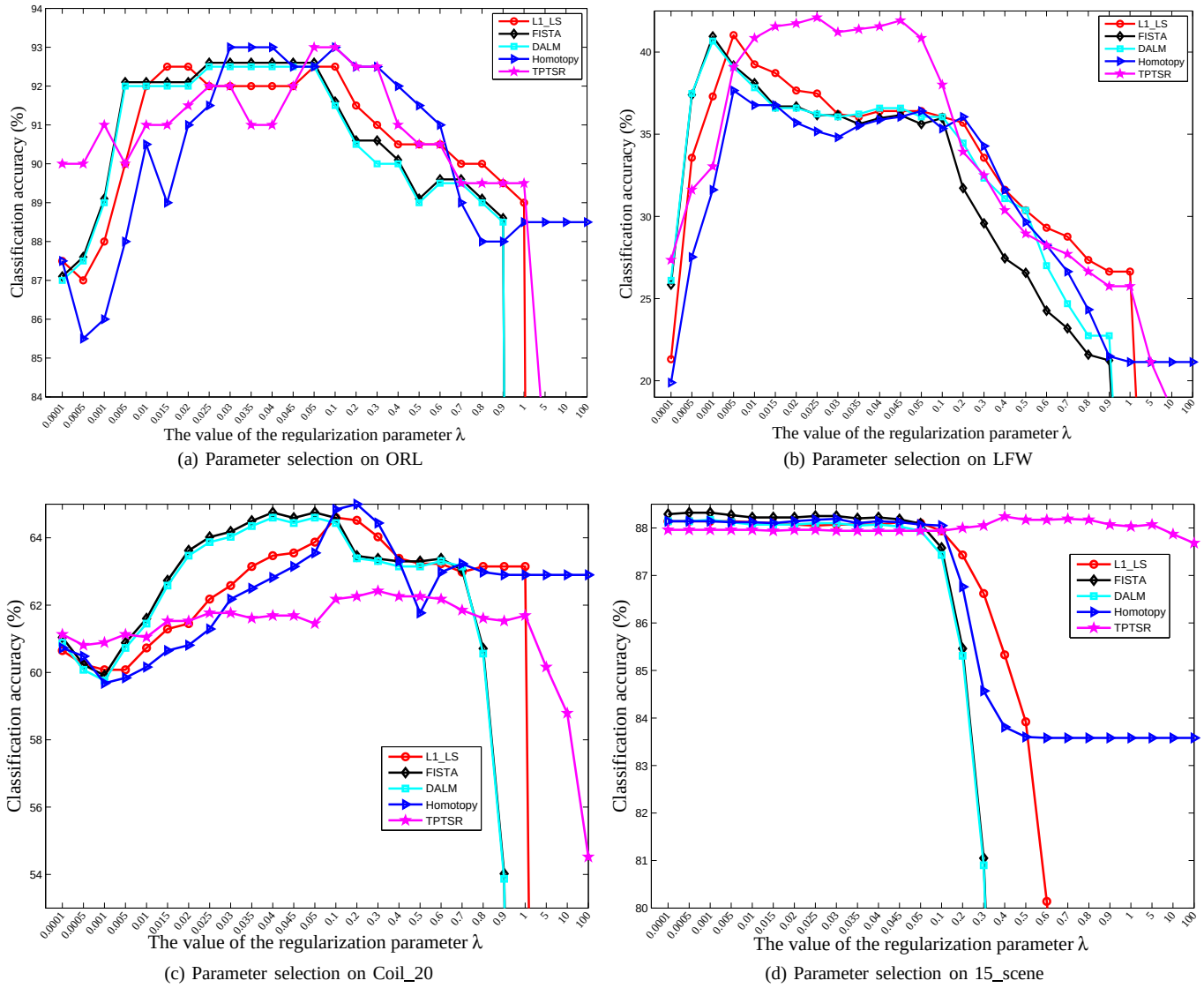
Numerous sparse representation based classification methods have been proposed to improve the robustness, effectiveness and efficiency of face recognition. For example, Xu et al. [9] proposed a two-phase sparse representation based classification method, which exploited the l_2-norm regularization rather than the l_1-norm regularization to perform a coarse to fine sparse representation based classification, which was very efficient in comparison with the conventional l_1-norm regularization based sparse representation. Deng et al. [186] proposed an extended sparse representation method (ESRM) for improving the robustness of SRC by eliminating the variations in face recognition, such as disguise, occlusion, expression and illumination. Deng et al. [187] also proposed a framework of superposed sparse representation based classification, which emphasized the prototype and vari-

ation components from uncontrolled images. He et al. [188] proposed utilizing the maximum correntropy criterion named CESR embedding non-negative constraint and half-quadratic optimization to present a robust face recognition algorithm. Yang et al. [189] developed a new robust sparse coding (RSC) algorithm, which first obtained a sparsity-constrained regression model based on maximum likelihood estimation and exploited an iteratively reweighted regularized robust coding algorithm to solve the pre-proposed model. Some other sparse representation based image classification methods also have been developed. For example, Yang et al. [190] introduced an extension of the spatial pyramid matching (SPM) algorithm called ScSPM, which incorporated SIFT sparse representation into the spatial pyramid matching algorithm. Subsequently, Gao et al. [191] developed a kernel sparse representation with the SPM algorithm called KSRSPM, and then proposed another version of an improvement of the SPM called LSc-SPM [192], which integrated the Laplacian matrix with local features into the objective function of the sparse representation method. Kulkarni and Li [193] proposed a discriminative affine sparse codes method (DASC) on a learned affine-invariant feature dictionary from input images and exploited the AdaBoost-based classifier to perform image classification. Zhang et al. [194] proposed integrating the non-negative sparse coding, low-rank and sparse matrix decomposition (LR-Sc$^+$SPM) method, which exploited non-negative sparse coding and SPM for achieving local features representation and employed low-rank and sparse matrix decomposition for sparse representation, for image classification. Recently, Zhang et al. [195] presented a low-rank sparse representation (LRSR) learning method, which preserved the sparsity and spatial consistency in each procedure of feature representation and jointly exploited local features from the same spatial proximal regions for image classification. Zhang et al. [196] developed a structured low-rank sparse representation (SLRSR) method for image classification, which constructed a discriminative dictionary in training terms and exploited low-rank matrix reconstruction for obtaining discriminative representations. Tao et al. [197] proposed a novel dimension reduction method based on the framework of rank preserving sparse learning, and then exploited the projected samples to make effective Kinect-based scene classification. Zhang et al. [198] proposed a discriminative tensor sparse coding (RTSC) method for robust image classification. Recently, low-rank based sparse representation became a popular topic such as non-negative low-rank and sparse graph [199]. Some sparse representation methods in face recognition can be found in a review [83] and other more image classification methods can be found in a more recent review [200].

Mei et al. employed the idea of sparse representation to visual tracking [201] and vehicle classification [202], which introduced nonnegative sparse constraints and dynamic template updating strategy. It, in the context of the particle filter framework, exploited the sparse technique to guarantee that each target candidate could be sparsely represented using the linear combinations of fewest targets and particle templates. It also demonstrated that sparse representation can be propagated to address object tracking problems. Extensive sparse representation methods have been proposed to address the visual tracking problem. In order to design an accelerated algorithm for l_1 tracker, Li et al. [203] proposed two real-time compressive sensing visual tracking algorithms based on sparse representation, which adopted dimension reduction and the OMP algorithm to improve the efficiency of recovery procedure in tracking, and also developed a modified version of fusing background templates into the tracking procedure for robust object tracking. Zhang et al. [204] directly treated object tracking as a pattern recognition problem by regarding all the targets as training samples, and then employed the sparse representation classification method to do effective object tracking. Zhang et al. [205] employed the concept of sparse representation based on a particle filter framework to construct a multi-task sparse learning method denoted as multi-task tracking for robust visual tracking. Additionally, because of the discriminative sparse representation between the target and the background, Jia et al. [206] conceived a structural local sparse appearance model for robust object tracking by integrating the partial and spatial information from the target based on an alignment-pooling algorithm. Liu et al. [207] proposed constructing a two-stage sparse optimization based online visual tracking method, which jointly minimized the objective reconstruction error and maximized the discriminative capability by choosing distinguishable features. Liu et al. [208] introduced a local sparse appearance model (SPT) with a static sparse dictionary learned from k-selection and dynamic updated basis distribution to eliminate potential drifting problems in the process of visual tracking. Bao et al. [209] developed a fast real time l_1-tracker called the APG-$l_1$tracker, which exploited the accelerated proximal gradient algorithm to improve the l_1-tracker solver in [201]. Zhong et al. [210] addressed the object tracking problem by developing a sparsity-based collaborative model, which combined a sparsity-based classifier learned from holistic templates and a sparsity-based template model generated from local representations. Zhang et al. [211] proposed to formulate a sparse feature measurement matrix based on an appearance model by exploiting non-adaptive random projections, and employed a coarse-to-fine strategy to accelerate the computational efficiency of tracking task. Lu et al. [212] proposed to employ both non-local self-similarity and sparse representation to develop a non-local self-similarity regularized sparse representation method based on geometrical structure information of the target template data set. Wang et al. [213] proposed a sparse representation based online two-stage tracking algorithm, which learned a linear classifier based on local sparse representation on favorable image patches. More detailed visual tracking algorithms can be found in the recent reviews [214, 215].

IX. Experimental evaluation

In this section, we take the object categorization problem as an example to evaluate the performance of different sparse representation based classification methods. We analyze and compare the performance of sparse representation with the most typical algorithms: OMP [36], $l_1_l_s$ [76], PALM [89], FISTA [82], DALM [89], homotopy [99] and TPTSR [9].



(a) Parameter selection on ORL

(b) Parameter selection on LFW

(c) Parameter selection on Coil_20

(d) Parameter selection on 15_scene

Fig. 5: Classification accuracies of using different sparse representation based classification methods versus varying values of the regularization parameter λ on the (a) ORL (b) LFW (c) Coil20 and (d) Fifteen scene datasets.

Plenties of data sets have been collected for object categorization, especially for image classification. Several image data sets are used in our experimental evaluations.

ORL: The ORL database includes 400 face images taken from 40 subjects each providing 10 face images [216]. For some subjects, the images were taken at different times, with varying lighting, facial expressions, and facial details. All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position (with tolerance for some side movement). Each image was resized to a 56×46 image matrix by using the down-sampling algorithm.

LFW face dataset: The Labeled Faces in the Wild (LFW) face database is designed for the study of unconstrained identity verification and face recognition [217]. It contains more than 13,000 images of faces collected from the web under the unconstrained conditions. Each face has been labeled with the name of the people pictured. 1680 of the people pictured have two or more distinct photos in the database. In our experiments, we chose 1251 images from 86 peoples and each subject has 10-20 images [218]. Each image was manually cropped and was resized to 32×32 pixels.

Extended YaleB face dataset: The extended YaleB database contains 2432 front face images of 38 individuals and each subject having around 64 near frontal images under different illuminations [219]. The main challenge of this database is to overcome varying illumination conditions and expressions. The facial portion of each original image was cropped to a 192×168 image. All images in this data set for our experiments simply resized these face images to 32×32 pixels.

COIL20 dataset: Columbia Object Image Library (COIL-20) database consists of 1,440 size normalized gray-scale images of 20 objects [220]. Different object images are captured at every angle in a 360 rotation. Images of the objects were taken from varying angles at pose intervals of five degrees and

each object has 72 images.

Fifteen scene dataset: This dataset contains 4485 images under 15 natural scene categories presented in literature [221] and each category includes 210 to 410 images. The 15 scenes categories are office, kitchen, living room, bedroom, store, industrial, tall building, inside cite, street, highway, coast, open country, mountain, forest and suburb. A wide range of outdoor and indoor scenes are included in this dataset. The average image size is around 250×300 pixels and the spatial pyramid matching features are used in our experiments.

A. Parameter selection

Parameter selection, especially selection of the regularization parameter λ in different minimization problems, plays an important role in sparse representation. In order to make fair comparisons with different sparse representation algorithms, performing the optimal parameter selection for different sparse representation algorithms on different datasets is advisable and indispensable. In this subsection, we perform extensive experiments for selecting the best value of the regularization parameter λ with a wide range of options. Specifically, we implement the $l_1_l_s$, FISTA, DALM, homotopy and TPTSR algorithms on different databases to analyze the importance of the regularization parameter. Fig. 5 summarizes the classification accuracies of exploiting different sparse representation based classification methods with varying values of regularization parameter λ on the two face datasets, i.e. ORL and LFW face datasets, and two object datasets, i.e. COIL20 and Fifteen scene datasets. On the ORL and LFW face datasets, we respectively selected the first five and eight face images of each subject as training samples and the rest of image samples for testing. As for the experiments on the COIL20 and fifteen scene datasets, we respectively treated the first ten images of each subject in both datasets as training samples and used all the remaining images as test samples. Moreover, from Fig. 5, one can see that the value of regularization parameter λ can significantly dominate the classification results, and the values of λ for achieving the best classification results on different datasets are distinctly different. An interesting scenario is that the performance of the TPTSR algorithm is almost not influenced by the variation of regularization parameter λ in the experiments on fifteen scene dataset, as shown in Fig. 5(d). However, the best classification accuracy can be always obtained within the range of 0.0001 to 1. Thus, the value of the regularization parameter is set within the range from 0.0001 to 1.

B. Experimental results

In order to test the performance of different kinds of sparse representation methods, an empirical study of experimental results is conducted in this subsection and seven typical sparse representation based classification methods are selected for performance evaluation followed with extensive experimental results. For all datasets, following most previous published work, we randomly choose several samples of every class as training samples and used the rest as test samples and the experiments are repeated 10 times with the optimal parameter obtained using the cross validation approach. The gray-level features of all images in these data sets are used to perform classification. For the sake of computational efficiency, principle component analysis algorithm is used as a preprocessing step to preserve 98% energy of all the data sets. The classification results and computational time have been summarized in Table I. From the experimental results on different databases, we can conclude that there still does not exist one extraordinary algorithm that can achieve the best classification accuracy on all databases. However, some algorithms are noteworthy to be paid much more attention. For example, the $l_1_l_s$ algorithm in most cases can achieve better classification results than the other algorithms on the ORL database, and when the number of training samples of each class is five, the $l_1_l_s$ algorithm can obtain the highest classification result of 95.90%. The TPTSR algorithm is very computationally efficient in comparison with other sparse representation with l_1-norm minimization algorithms and the classification accuracies obtained by the TPTSR algorithm are very similar and sometimes even better than the other sparse representation based classification algorithms.

The computational time is another indicator for measuring the performance of one specific algorithm. As shown in Table I, the average computational time of each algorithm is shown at the bottom of the table for one specific number of training samples. Note that the computational time of OMP and TPTSR algorithms are drastically lower than that of other sparse representation with l_1-norm minimization algorithms. This is mainly because the sparse representation with l_1-norm minimization algorithms always iteratively solve the l_1-norm minimization problem. However, the OMP and TPTSR algorithms both exploit the fast and efficient least squares technique, which guarantees that the computational time is significantly less than other l_1-norm based sparse representation algorithms.

C. Discussion

Lots of sparse representation methods have been available in past decades and this paper introduces various sparse representation methods from some viewpoints, including their motivations, mathematical representations and the main algorithms. Based on the experimental results summarized in Section IX, we have the following observations.

First, a challenging task of choosing a suitable regularization parameter for sparse representation should make further extensive studies. We can see that the value of the regularization parameter can remarkably influence the performance of the sparse representation algorithms and adjusting the parameters in sparse representation algorithms requires expensive labor. Moreover, adaptive parameter selection based sparse representation methods is preferable and very few methods have been proposed to solve this critical issue.

Second, although sparse representation algorithms have achieved distinctly promising performance on some real-world databases, many efforts should be made in promoting the accuracy of sparse representation based classification, and the

Data set ($\#Tr$)	OMP	l_1_ls	PALM	FISTA	DALM	Homotopy	TPTSR
ORL(1)	64.94±2.374	68.50±2.021	68.36±1.957	70.67±2.429	70.22±2.805	66.53±1.264	**71.56±3.032**
ORL(2)	80.59±2.256	**84.84±2.857**	80.66±2.391	84.72±3.242	84.38±2.210	83.88±2.115	83.38±2.019
ORL(3)	89.00±1.291	89.71±1.313	86.82±1.959	90.00±3.141	90.36±1.829	89.32±1.832	**90.71±1.725**
ORL(4)	91.79±1.713	**94.83±1.024**	88.63±2.430	94.13±1.310	94.71±1.289	94.38±1.115	94.58±1.584
ORL(5)	93.75±2.125	**95.90±1.150**	92.05±1.039	95.60±1.761	95.50±1.269	95.60±1.430	95.75±1.439
ORL(6)	95.69±1.120	97.25±1.222	92.06±1.319	96.69±1.319	96.56±1.724	**97.31±1.143**	95.81±1.642
Average Time(5)	0.0038s	0.1363s	7.4448s	0.9046s	0.8013s	0.0100s	**0.0017s**
LFW(3)	22.22±1.369	**28.24±0.667**	15.16±1.202	26.55±0.767	25.46±0.705	26.12±0.831	27.32±1.095
LFW(5)	27.83±1.011	**35.58±1.489**	12.89±1.286	34.13±0.459	33.90±1.181	33.95±1.680	35.43±1.409
LFW(7)	32.76±2.318	40.17±2.061	11.63±0.937	39.86±1.226	38.40±1.890	38.04±1.251	**40.92±1.201**
LFW(9)	35.14±1.136	**44.93±1.123**	7.84±1.278	43.86±1.492	43.56±1.393	42.29±2.721	44.72±1.793
Average Time(7)	**0.0140s**	0.6825s	33.2695s	3.0832s	3.9906s	0.2372s	0.0424s
Extended YaleB(3)	44.20±2.246	63.10±2.341	63.73±2.073	62.84±2.623	**63.76±2.430**	64.22±2.525	56.23±2.153
Extended YaleB(6)	72.48±2.330	81.97±0.850	81.93±0.930	**82.25±0.734**	81.74±1.082	81.64±1.159	78.53±1.731
Extended YaleB(9)	83.42±0.945	88.90±0.544	88.50±1.096	**89.31±0.829**	89.26±0.781	89.12±0.779	86.49±1.165
Extended YaleB(12)	88.23±0.961	**92.49±0.622**	91.07±0.725	92.03±1.248	91.85±0.710	92.03±0.767	91.30±0.741
Extended YaleB(15)	91.97±0.963	94.22±0.719	93.19±0.642	**94.50±0.824**	93.07±0.538	93.67±0.860	93.38±0.785
Average Time(12)	**0.0116s**	3.2652s	17.4516s	1.5739s	1.9384s	0.5495s	0.0198s
COIL20(3)	75.90±1.656	77.62±2.347	70.26±2.646	75.80±2.056	76.67±2.606	**78.46±2.603**	78.16±2.197
COIL20(5)	83.00±1.892	82.63±1.701	79.55±1.153	84.09±2.003	84.38±1.319	**84.58±1.487**	83.69±1.804
COIL20(7)	87.26±1.289	88.22±1.304	82.88±1.445	88.89±1.598	89.00±1.000	**89.36±1.147**	87.75±1.451
COIL20(9)	89.56±1.763	90.97±1.595	84.94±1.563	90.16±1.366	**91.82±1.555**	91.44±1.198	89.41±2.167
COIL20(11)	91.70±0.739	92.98±1.404	87.16±1.184	93.43±1.543	**93.46±1.327**	93.55±1.205	92.71±1.618
COIL20(13)	92.49±1.146	94.29±0.986	88.36±1.283	94.50±0.850	93.92±1.102	**94.93±0.788**	92.72±1.481
Average Time(13)	**0.0038s**	0.0797s	7.5191s	0.7812s	0.7762s	0.0159s	0.0053s
Fifteen scene(3)	85.40±1.388	**86.83±1.082**	86.15±1.504	86.48±1.542	85.89±1.624	86.15±1.073	86.62±1.405
Fifteen scene(6)	89.14±1.033	90.34±0.685	89.97±0.601	90.82±0.921	90.12±0.998	89.65±0.888	**90.83±0.737**
Fifteen scene(9)	83.42±0.945	88.90±0.544	88.50±1.096	89.31±0.829	89.26±0.781	89.12±0.779	**90.64±0.940**
Fifteen scene(12)	91.67±0.970	92.06±0.536	**92.76±0.905**	92.22±0.720	92.45±0.860	92.35±0.706	92.33±0.563
Fifteen scene(15)	93.32±0.609	93.37±0.506	93.63±0.510	93.63±0.787	93.53±0.829	**93.84±0.586**	93.80±0.461
Fifteen scene(18)	93.61±0.334	94.31±0.551	94.67±0.678	94.28±0.396	94.16±0.344	94.16±0.642	**94.78±0.494**
Average Time(18)	**0.0037s**	0.0759s	0.9124s	0.8119s	0.8500s	0.1811s	0.0122s

TABLE I: Classification accuracies (mean classification error rates ± standard deviation %) of different sparse representation algorithms with different numbers of training samples. The bold numbers are the lowest error rates and the least time cost of different algorithms.

robustness of sparse representation should be further enhanced. In terms of the recognition accuracy, the algorithms of $l_1_l_s$, homotopy and TPTSR achieve the best overall performance. Considering the experimental results of exploiting the seven algorithms on the five databases, the $l_1_l_s$ algorithm has eight highest classification accuracies, followed by homotopy and TPTSR, in comparison with other algorithms. One can see that the sparse representation based classification methods still can not obtain satisfactory results on some challenge databases. For example, all these representative algorithms can achieve relatively inferior experimental results on the LFW dataset shown in Subsection IX-B, because the LFW dataset is designed for studying the problem of unconstrained face recognition [217] and most of the face images are captured under complex environments. One can see that the PALM algorithm has the worst classification accuracy on the LFW dataset and the classification accuracy even decreases mostly with the increase of the number of the training samples. Thus, devising more robust sparse representation algorithm is an urgent issue.

Third, enough attention should be paid on the computational inefficiency of sparse representation with l_1-norm minimization. One can see that high computational complexity is one of the most major drawbacks of the current sparse representation methods and also hampers its applications in real-time processing scenarios. In terms of speed, PALM, FISTA and DALM take much longer time to converge than the other methods. The average computational time of OMP and TPTSR is the two lowest algorithms. Moreover, compared with the l_1-regularized sparse representation based classification methods, the TPTSR has very competitive classification accuracy but significantly low complexity. Efficient and effective sparse representation methods are urgently needed by real-time applications. Thus, developing more efficient and effective methods is essential for future study on sparse representation.

Finally, the extensive experimental results have demonstrated that there is no absolute winner that can achieve the best performance for all datasets in terms of classification accuracy and computational efficiency. However, $l_1_l_s$, TPTSR and homotopy algorithms as a whole outperform the other algorithms. As a compromising approach, the OMP algorithm can achieve distinct efficiency without sacrificing much recognition rate in comparison with other algorithms and it also has been extensively applied to some complex learning algorithms as a function.

X. Conclusion

Sparse representation has been extensively studied in recent years. This paper summarizes and presents various available sparse representation methods and discusses their motivations, mathematical representations and extensive applications. More specifically, we have analyzed their relations in theory and empirically introduced the applications including dictionary

learning based on sparse representation and real-world applications such as image processing, image classification, and visual tracking.

Sparse representation has become a fundamental tool, which has been embedded into various learning systems and also has received dramatic improvements and unprecedented achievements. Furthermore, dictionary learning is an extremely popular topic and is closely connected with sparse representation. Currently, efficient sparse representation, robust sparse representation, and dictionary learning based on sparse representation seem to be the main streams of research on sparse representation methods. The low-rank representation technique has also recently aroused intensive research interests and sparse representation has been integrated into low-rank representation for constructing more reliable representation models. However, the mathematical justification of low-rank representation seems not to be elegant as sparse representation. Because employing the ideas of sparse representation as a prior can lead to state-of-the-art results, incorporating sparse representation with low-rank representation is worth further research. Moreover, subspace learning also has been becoming one of the most prevailing techniques in pattern recognition and computer vision. It is necessary to further study the relationship between sparse representation and subspace learning, and constructing more compact models for sparse subspace learning becomes one of the popular topics in various research fields. The transfer learning technique has emerged as a new learning framework for classification, regression and clustering problems in data mining and machine learning. However, sparse representation research still has been not fully applied to the transfer learning framework and it is significant to unify the sparse representation and low-rank representation techniques into the transfer learning framework to solve domain adaption, multitask learning, sample selection bias and covariate shift problems. Furthermore, researches on deep learning seems to become an overwhelming trend in the computer vision field. However, dramatically expensive training effort is the main limitation of current deep learning technique and how to fully introduce current sparse representation methods into the framework of deep learning is valuable and unsolved.

The application scope of sparse representation has emerged and has been widely extended to machine learning and computer vision fields. Nevertheless, the effectiveness and efficiency of sparse representation methods cannot perfectly meet the need for real-world applications. Especially, the complexities of sparse representation have greatly affected the applicability, especially the applicability to large scale problems. Enhancing the robustness of sparse representation is considered as another indispensable problem when researchers design algorithms. For image classification, the robustness should be seriously considered, such as the robustness to random corruptions, varying illuminations, outliers, occlusion and complex backgrounds. Thus, developing an efficient and robust sparse representation method for sparse representation is still the main challenge and to design a more effective dictionary is being expected and is beneficial to the performance improvement.

Sparse representation still has wide potential for various possible applications, such as event detection, scene reconstruction, video tracking, object recognition, object pose estimation, medical image processing, genetic expression and natural language processing. For example, the study of sparse representation in visual tracking is an important direction and more depth studies are essential to future further improvements of visual tracking research.

In addition, most sparse representation and dictionary learning algorithms focus on employing the l_0-norm or l_1-norm regularization to obtain a sparse solution. However, there are still only a few studies on $l_{2,1}$-norm regularization based sparse representation and dictionary learning algorithms. Moreover, other extended studies of sparse representation may be fruitful. In summary, the recent prevalence of sparse representation has extensively influenced different fields. It is our hope that the review and analysis presented in this paper can help and motivate more researchers to propose perfect sparse representation methods.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grant 61370163, Grant 61233011, and Grant 61332011, and in part by the Shenzhen Municipal Science and Technology Innovation Council under Grant JCYJ20130329151843309, Grant JCYJ20130329151843309, Grant JCYJ20140417172417174, Grant CXZZ20140904154910774, the China Postdoctoral Science Foundation funded project, No. 2014M560264 and the Shaanxi Key Innovation Team of Science and Technology (Grant no.: 2012KCT-04).

We would like to thank Jian Wu for many inspiring discussions and he is ultimately responsible for many of the ideas in the algorithm and analysis. We would also like to thank Dr. Zhihui Lai, Dr. Jinxing Liu and Xiaozhao Fang for constructive suggestions. Moreover, we thank the editor, an associate editor, and referees for helpful comments and suggestions which greatly improved this paper.

References

[1] B. K. Natarajan, "Sparse approximate solutions to linear systems," *SIAM journal on computing*, vol. 24, no. 2, pp. 227–234, 1995.

[2] M. Huang, W. Yang, J. Jiang, Y. Wu, Y. Zhang, W. Chen, and Q. Feng, "Brain extraction based on locally linear representation-based classification," *NeuroImage*, vol. 92, pp. 322–339, 2014.

[3] X. Lu and X. Li, "Group sparse reconstruction for image segmentation," *Neurocomputing*, vol. 136, pp. 41–48, 2014.

[4] M. Elad, M. Figueiredo, and Y. Ma, "On the role of sparse and redundant representations in image processing," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 972–982, 2010.

[5] S. Mallat, *A wavelet tour of signal processing: the sparse way*. Academic press, 2008.

[6] J. Starck, F. Murtagh, and J. M. Fadili, *Sparse image and signal processing: wavelets, curvelets, morphological diversity*. Cambridge University Press, 2010.

[7] M. Elad, *Sparse and redundant representations: from theory to applications in signal and image processing*. Springer, 2010.

[8] A. M. Bruckstein, D. L. Donoho, and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images," *SIAM review*, vol. 51, no. 1, pp. 34–81, 2009.

[9] Y. Xu, D. Zhang, J. Yang, and J. Yang, "A two-phase test sample sparse representation method for use with face recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 9, pp. 1255–1262, 2011.

[10] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. Huang, and S. Yan, "Sparse representation for computer vision and pattern recognition," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1031–1044, 2010.

[11] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[12] R. G. Baraniuk, "Compressive sensing," *IEEE signal processing magazine*, vol. 24, no. 4, pp. 118–121, 2007.

[13] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[14] E. J. Candes and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, 2008.

[15] Y. Tsaig and D. L. Donoho, "Extensions of compressed sensing," *Signal processing*, vol. 86, no. 3, pp. 549–571, 2006.

[16] E. J. Candes, "Compressive sampling," in *Proceedings of the International Congress of Mathematicians: Madrid, August 22-30, 2006: invited lectures*, 2006, pp. 1433–1452.

[17] E. Candes and J. Romberg, "Sparsity and incoherence in compressive sampling," *Inverse problems*, vol. 23, no. 3, p. 969, 2007.

[18] X. Lu, H. Wu, Y. Yuan, P. Yan, and X. Li, "Manifold regularized sparse nmf for hyperspectral unmixing," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 5, pp. 2815–2826, 2013.

[19] Y. Yuan, X. Li, Y. Pang, X. Lu, and D. Tao, "Binary sparse nonnegative matrix factorization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 5, pp. 772–779, 2009.

[20] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2009.

[21] Z. Zhang, Z. Li, B. Xie, L. Wang, and Y. Chen, "Integrating globality and locality for robust representation based classification," *Mathematical Problems in Engineering*, vol. 2014, 2014.

[22] H. Cheng, Z. Liu, L. Yang, and X. Chen, "Sparse representation and learning in visual recognition: Theory and applications," *Signal Processing*, vol. 93, no. 6, pp. 1408–1425, 2013.

[23] J. A. Tropp, A. C. Gilbert, and M. J. Strauss, "Algorithms for simultaneous sparse approximation. part i: Greedy pursuit," *Signal Processing*, vol. 86, no. 3, pp. 572–588, 2006.

[24] J. A. Tropp, "Algorithms for simultaneous sparse approximation. part ii: Convex relaxation," *Signal Processing*, vol. 86, no. 3, pp. 589–602, 2006.

[25] M. Schmidt, G. Fung, and R. Rosales, "Optimization methods for l_1-regularization," University of British Columbia, West Mall Vancouver, B.C. Canada V6T 1Z4, Tech. Rep., 2009.

[26] E. Amaldi and V. Kann, "On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems," *Theoretical Computer Science*, vol. 209, no. 1, pp. 237–260, 1998.

[27] J. A. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Transactions on Information Theory*, vol. 50, no. 10, pp. 2231–2242, 2004.

[28] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 123–231, 2013.

[29] D. Donoho and Y. Tsaig, "Fast solution of l_1-norm minimization problems when the solution may be sparse." *IEEE Transactions on Information Theory*, vol. 54, no. 11, pp. 4789–4812, 2008.

[30] Z. Zhang, L. Wang, Q. Zhu, Z. Liu, and Y. Chen, "Noise modeling and representation based classification methods for face recognition," *Neurocomputing*, vol. 148, pp. 420–429, 2015.

[31] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2009.

[32] J. Chen and X. Huo, "Theoretical results on sparse representations of multiple-measurement vectors," *IEEE Transactions on Signal Processing*, vol. 54, no. 12, pp. 4634–4643, 2006.

[33] J. K. Pant, W. S. Lu, and A. Antoniou, "Unconstrained regularized l_p-norm based algorithm for the reconstruction of sparse signals," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, 2011, pp. 1740–1743.

[34] R. Chartrand and W. Yin, "Iteratively reweighted algorithms for compressive sensing," in *Proceedings of the IEEE international conference on Acoustics, speech and signal processing*, 2008, pp. 3869–3872.

[35] J. Yang, L. Zhang, Y. Xu, and J. Yang, "Beyond sparsity: The role of l_1-optimizer in pattern classification," *Pattern Recognition*, vol. 45, no. 3, pp. 1104–1118, 2012.

[36] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[37] D. Needell and R. Vershynin, "Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit," *Foundations of computational mathematics*, vol. 9, no. 3, pp. 317–334, 2009.

[38] R. Saab, R. Chartrand, and O. Yilmaz, "Stable sparse

approximations via nonconvex optimization," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2008, pp. 3885–3888.

[39] R. Chartrand, "Exact reconstruction of sparse signals via nonconvex minimization," *IEEE Signal Processing Letters*, vol. 14, no. 10, pp. 707–710, 2007.

[40] Z. B. Xu, "Data modeling: Visual psychology approach and $l_{1/2}$ regularization theory," in *Proceeding of International Congress of Mathmaticians*, 2010, pp. 3151–3184.

[41] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society Series B*, pp. 267–288, 1996.

[42] B. Efron, T. Hastie, I. Johnstone, I. Johnstone, and R. Tibshirani, "Least angle regression," *The Annals of statistics*, vol. 32, no. 2, pp. 407–499, 2004.

[43] J. Yang and Y. Zhang, "Alternating direction algorithms for l_1-problems in compressive sensing," *SIAM journal on scientific computing*, vol. 33, no. 1, pp. 250–278, 2011.

[44] M. Schmidt, G. Fung, and R. Rosales, "Fast optimization methods for l_1 regularization: A comparative study and two new approaches," in *Machine Learning: ECML 2007*. Springer, 2007, pp. 286–297.

[45] F. Nie, H. Huang, X. Cai, and C. Ding, "Efficient and robust feature selection via joint $l_{2,1}$-norms minimization," in *Proceedings of the Advances in Neural Information Processing Systems*, 2010, pp. 1813–1821.

[46] Y. Yang, H. T. Shen, Z. Ma, Z. Huang, and X. Zhou, "$l_{2,1}$-norm regularized discriminative feature selection for unsupervised learning," in *Proceedings of the International Joint Conference on Artificial Intelligence*, vol. 22, no. 1, 2011, pp. 1589–1594.

[47] X. S. Shi, Y. J. Yang, Z. H. Guo, and Z. H. Lai, "Face recognition by sparse discriminant analysis via joint $l_{2,1}$-norm minimization," *Pattern Recognition*, vol. 47, no. 7, pp. 2447–2453, 2014.

[48] J. Liu, S. Ji, and J. Ye, "Multi-task feature learning via efficient $l_{2,1}$-norm minimization," in *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, 2009, pp. 339–348.

[49] C. Hou, F. Nie, X. Li, D. Yi, and Y. Wu, "Joint embedding learning and sparse regression: A framework for unsupervised feature selection," *IEEE Transactions on Cybernetics*, vol. 44, no. 6, pp. 793–804, 2014.

[50] I. Naseem, R. Togneri, and M. Bennamoun, "Linear regression for face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 11, pp. 2106–2112, 2010.

[51] D. Zhang, M. Yang, and X. Feng, "Sparse representation or collaborative representation: Which helps face recognition?" in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011, pp. 471–478.

[52] D. Donoho and M. Elad, "Optimally sparse representation in general (nonorthogonal) dictionaries via l_1 minimization," *Proceedings of the National Academy of Sciences*, vol. 100, no. 5, pp. 2197–2202, 2003.

[53] L. Liu, L. Shao, F. Zheng, and X. Li, "Realistic action recognition via sparsely-constructed gaussian processes," *Pattern Recognition*, vol. 47, no. 12, pp. 3819–3827, 2014.

[54] V. M. Patel and R. Chellappa, "Sparse representations, compressive sensing and dictionaries for pattern recognition," in *Proceedings of the IEEE First Asian Conference on Pattern Recognition*, 2011, pp. 325–329.

[55] Y. Yuan, X. Lu, and X. Li, "Learning hash functions using sparse reconstruction," in *Proceedings of International Conference on Internet Multimedia Computing and Service*, 2014, pp. 14–18.

[56] D. Donoho, "For most large underdetermined systems of linear equations the minimal l_1-norm solution is also the sparsest solution," *Communications on pure and applied mathematics*, vol. 59, no. 6, pp. 797–829, 2006.

[57] E. Candes, J. Romberg, , and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on pure and applied mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.

[58] E. Candes and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?" *IEEE Transaction on Information Theory*, vol. 52, no. 12, pp. 5406–5425, 2006.

[59] L. Qin, Z. C. Lin, Y. Y. She, and C. Zhang, "A comparison of typical l_p minimization algorithms," *Neurocomputing*, vol. 119, pp. 413–424, 2013.

[60] Z. Xu, X. Chang, F. Xu, and H. Zhang, "$l_{1/2}$ regularization: A thresholding representation theory and a fast solver," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 7, pp. 1013–1027, 2012.

[61] S. Guo, Z. Wang, and Q. Ruan, "Enhancing sparsity via l_p (0<p<1) minimization for robust face recognition," *Neurocomputing*, vol. 99, pp. 592–602, 2013.

[62] C. Ding, D. Zhou, X. He, and H. Zha, "R1-pca: rotational invariant l_1-norm principal component analysis for robust subspace factorization," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 281–288.

[63] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[64] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proceedings of the Twenty-Seventh Asilomar Conference on Signals, Systems and Computers*, 1993, pp. 40–44.

[65] S. N. Vitaladevuni, P. Natarajan, and R. Prasad, "Efficient orthogonal matching pursuit using sparse random projections for scene and video classification," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011, pp. 2312–2319.

[66] D. Needell and J. A. Tropp, "Cosamp: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, 2009.

[67] D. L. Donoho, Y. Tsaig, I. Drori, and J. Starck, "Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit," *IEEE Transactions on Information Theory*, vol. 58, no. 2, pp. 1094–1121, 2012.

[68] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *IEEE Transactions on Information Theory*, vol. 55, no. 5, pp. 2230–2249, 2009.

[69] T. T. Do, L. Gan, N. Nguyen, and T. Tran, "Sparsity adaptive matching pursuit algorithm for practical compressed sensing," in *Proceedings of the 42nd Asilomar Conference on Signals, Systems and Computers*, 2008, pp. 581–587.

[70] P. Jost, P. Vandergheynst, and P. Frossard, "Tree-based pursuit: Algorithm and properties," *IEEE Transactions on Signal Processing*, vol. 54, no. 12, pp. 4685–4697, 2006.

[71] C. La and M. N. Do, "Tree-based orthogonal matching pursuit algorithm for signal reconstruction," in *IEEE International Conference on Image Processing*, 2006, pp. 1277–1280.

[72] N. B. Karahanoglu and H. Erdogan, "Compressed sensing signal recovery via forward–backward pursuit," *Digital Signal Processing*, vol. 23, no. 5, pp. 1539–1548, 2013.

[73] M. Figueiredo, R. D. Nowak, and S. J. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 586–597, 2007.

[74] S. J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "An interior-point method for large-scale l_1-regularized least squares," *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 606–617, 2007.

[75] L. F. Portugal, M. Resende, G. Veiga, and J. Judice, "A truncated primal-infeasible dual-feasible network interior point method," *Networks*, vol. 35, no. 2, pp. 91–108, 2000.

[76] K. Koh, S. J. Kim, and S. P. Boyd, "An interior-point method for large-scale l_1-regularized logistic regression." *Journal of Machine learning research*, vol. 8, no. 8, pp. 1519–1555, 2007.

[77] S. Mehrotra, "On the implementation of a primal-dual interior point method," *SIAM Journal on optimization*, vol. 2, no. 4, pp. 575–601, 1992.

[78] S. J. Wright, *Primal-dual interior-point methods*. SIAM, 1997, vol. 54.

[79] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[80] M. Figueiredo and R. Nowak, "A bound optimization approach to wavelet-based image deconvolution," in *Proceedings of the IEEE International Conference on Image Processing*, vol. 2, 2005, pp. II–782–5.

[81] P. L. Combettes and J. C. Pesquet, "Proximal splitting methods in signal processing," in *Fixed-point algorithms for inverse problems in science and engineering*, 2011, pp. 185–212.

[82] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[83] A. Y. Yang, S. Sastryand, A. Ganesh, and 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.

[84] S. J. Wright, R. D. Nowak, and M. Figueiredo, "Sparse reconstruction by separable approximation," *IEEE Transactions on Signal Processing*, vol. 57, no. 7, pp. 2479–2493, 2009.

[85] Y. H. Dai, W. Hager, K. Schittkowski, and H. Zhang, "The cyclic barzilai—borwein method for unconstrained optimization," *IMA Journal of Numerical Analysis*, vol. 26, no. 3, pp. 604–627, 2006.

[86] E. T. Hale, W. Yin, and Y. Zhang, "A fixed-point continuation method for l_1-regularized minimization with applications to compressed sensing," Rice University, St, Houston, USA, Tech. Rep., 2007.

[87] J. Zeng, Z. Xu, B. Zhang, W. Hong, and Y. Wu, "Accelerated $l_{1/2}$ regularization based sar imaging via bcr and reduced newton skills," *Signal Processing*, vol. 93, no. 7, pp. 1831–1844, 2013.

[88] H. J. Zeng, S. Lin, Y. Wang, and Z. Xu, "$l_{1/2}$ regularization: convergence of iterative half thresholding algorithm," *IEEE Transactions on Signal Processing*, vol. 62, no. 9, pp. 2317–2329, 2014.

[89] A. Yang, Z. Zhou, A. Balasubramanian, S. Sastry, and Y. Ma, "Fast l_1-minimization algorithms for robust face recognition," *IEEE Transactions on Image Processing*, vol. 22, no. 8, pp. 3234–3246, 2013.

[90] M. Elad, B. Matalon, and M. Zibulevsky, "Coordinate and subspace optimization methods for linear least squares with non-quadratic regularization," *Applied and Computational Harmonic Analysis*, vol. 23, no. 3, pp. 346–367, 2007.

[91] D. L. Donoho, A. Maleki, and A. Montanari, "Message-passing algorithms for compressed sensing," *Proceedings of the National Academy of Sciences*, vol. 106, no. 45, pp. 18 914–18 919, 2009.

[92] S. Becker, J. Bobin, and E. J. Candes, "Nesta: a fast and accurate first-order method for sparse recovery," *SIAM Journal on Imaging Sciences*, vol. 4, no. 1, pp. 1–39, 2011.

[93] S. R. Becker, E. J. Candes, and M. C. Grant, "Templates for convex cone problems with applications to sparse signal recovery," *Mathematical Programming Computation*, vol. 3, no. 3, pp. 165–218, 2011.

[94] M. R. Osborne, B. Presnell, and B. A. Turlach, "A new approach to variable selection in least squares problems," *IMA journal of numerical analysis*, vol. 20, no. 3, pp. 389–403, 2000.

[95] M. D. Plumbley, "Recovery of sparse representations by polytope faces pursuit," in *Independent Component Analysis and Blind Signal Separation*, 2006, pp. 206–213.

[96] M. S. Asif and J. Romberg, "Fast and accurate algorithms for re-weighted l_1-norm minimization," *IEEE Transactions on Signal Processing*, vol. 61, no. 23, pp. 5905 – 5916, 2012.

[97] D. M. Malioutov, M. Cetin, and A. S. Willsky, "Homotopy continuation for sparse signal representation," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'05)*, vol. 5, 2005, pp. v733–v736.

[98] P. Garrigues and L. E. Ghaoui, "An homotopy algorithm for the lasso with online observations," in *Advances in neural information processing systems*, 2009, pp. 489–496.

[99] M. S. Asif, "Primal dual pursuit: A homotopy based algorithm for the dantzig selector," Master's thesis, Georgia Institute of Technology, Atlanta, Georgia, USA, 2008.

[100] M. S. Asif and J. Romberg, "Dynamic updating for l_1 minimization," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 421–434, 2010.

[101] ——, "Sparse recovery of streaming signals using l_1-homotopy," *arXiv preprint arXiv:1306.3331*, 2013.

[102] M. S. Asif, "Dynamic compressive sensing: Sparse recovery algorithms for streaming signals and video," Ph.D. dissertation, Georgia Institute of Technology, Atlanta, Georgia, Unit State of America, 2013.

[103] J. Cooley and J. Tukey, "An algorithm for the machine calculation of complex fourier series," *Mathematics of computation*, vol. 19, no. 90, pp. 297–301, 1965.

[104] R. Rubinstein, A. Bruckstein, and M. Elad, "Dictionaries for sparse representation modeling," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1045–1057, 2010.

[105] L. Shao, R. Yan, X. Li, and Y. Liu, "From heuristic optimization to dictionary learning: a review and comprehensive comparison of image denoising algorithms," *IEEE Transactions on Cybernetics*, vol. 44, no. 7, pp. 1001–1013, 2014.

[106] E. Simoncelli and E. Adelson, "Noise removal via bayesian wavelet coring," in *International Conference on Image Processing*, vol. 1, 1996, pp. 379–382.

[107] W. He, Y. Zi, B. Chen, F. Wu, and Z. He, "Automatic fault feature extraction of mechanical anomaly on induction motor bearing using ensemble super-wavelet transform," *Mechanical Systems and Signal Processing*, vol. 54, pp. 457–480, 2015.

[108] E. L. Pennec and S. Mallat, "Sparse geometric image representations with bandelets," *IEEE Transactions on Image Processing*, vol. 14, no. 4, pp. 423–438, 2005.

[109] J. Starck, E. Candes, and D. Donoho, "The curvelet transform for image denoising," *Image Processing, IEEE Transactions on*, vol. 11, no. 6, pp. 670–684, 2002.

[110] M. Do and M. Vetterli, "The contourlet transform: an efficient directional multiresolution image representation," *IEEE Transactions on Image Processing*, vol. 14, no. 12, pp. 2091–2106, 2005.

[111] E. Simoncelli, W. Freeman, E. Adelson, and D. Heeger, "Shiftable multiscale transforms," *IEEE Transactions on Information Theory*, vol. 38, no. 2, pp. 587–607, 1992.

[112] J. Shi, X. Ren, G. Dai, J. Wang, and Z. Zhang, "A non-convex relaxation approach to sparse dictionary learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 1809–1816.

[113] J. Fan and R. Li, "Variable selection via nonconcave penalized likelihood and its oracle properties," *Journal of the American Statistical Association*, vol. 96, no. 456, pp. 1348–1360, 2001.

[114] C. H. Zhang, "Nearly unbiased variable selection under minimax concave penalty," *The Annals of Statistics*, pp. 894–942, 2010.

[115] J. H. Friedman, "Fast sparse regression and classification," *International Journal of Forecasting*, vol. 28, no. 3, pp. 722–738, 2012.

[116] R. Rubinstein, A. M. Bruckstein, and M. Elad, "Dictionaries for sparse representation modeling," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1045–1057, 2010.

[117] I. Tosic and P. Frossard, "Dictionary learning," *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 27–38, 2011.

[118] C. Bao, H. Ji, Y. Quan, and Z. Shen, "l_0 norm based dictionary learning by proximal methods with global convergence," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3858–3865.

[119] A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by v1?" *Vision research*, vol. 37, no. 23, pp. 3311–3325, 1997.

[120] K. Engan, S. O. Aase, and J. H.Husy, "Multi-frame compression: Theory and design," *Signal Processing*, vol. 80, no. 10, pp. 2121–2140, 2000.

[121] K. Kreutz-Delgado, J. F. Murray, B. D. Rao, K. Engan, T. W. Lee, and T. J. Sejnowski, "Dictionary learning algorithms for sparse representation," *Neural computation*, vol. 15, no. 2, pp. 349–396, 2003.

[122] M. E. M. Aharon and A. Bruckstein, "k-svd: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[123] R. Jenatton, J. Mairal, G. Obozinski, and F. Bach, "Proximal methods for sparse hierarchical dictionary learning," in *Proceedings of the 27th International Conference on Machine Learning*, 2010, pp. 487–494.

[124] S. Bengio, F. Pereira, Y. Singer, and D. Strelow, "Group sparse coding," in *Proceedings of the Advances in Neural Information Processing Systems*, 2009, pp. 82–89.

[125] B. Zhao, F. L, and E. P. Xing, "Online detection of unusual events in videos via dynamic sparse coding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 3313–3320.

[126] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Robust visual tracking via multi-task sparse learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2042–2049.

[127] Q. Zhang and B. Li, "Discriminative k-svd for dictionary learning in face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2691–2698.

[128] Y. Yang, Y. Yang, Z. Huang, H. T. Shen, and F. Nie, "Tag localization with spatial correlations and joint group sparsity," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 881–888.

[129] O. Bryt and M. Elad, "Compression of facial images using the k-svd algorithm," *Journal of Visual Communication and Image Representation*, vol. 19, no. 4, pp. 270–282, 2008.

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

[131] M. Zhou, H. Chen, J. Paisley, L. Ren, L. Li, Z. Xing, D. Dunson, G. Sapiro, and L. Carin, "Nonparametric bayesian dictionary learning for analysis of noisy and incomplete images," *IEEE Transactions on Image Processing*, vol. 21, no. 1, pp. 130–144, 2012.

[132] I. Ramirez and G. Sapiro, "An mdl framework for sparse coding and dictionary learning," *IEEE Transactions on Signal Processing*, vol. 60, no. 6, pp. 2913–2927, 2012.

[133] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *The Journal of Machine Learning Research*, vol. 11, pp. 19–60, 2010.

[134] M. Yang, L. Van, and L. Zhang, "Sparse variation dictionary learning for face recognition with a single sample per person," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013, pp. 689–696.

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

[136] Z. Jiang, Z. Lin, and L. Davis, "Label consistent k-svd: learning a discriminative dictionary for recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2651–2664, 2013.

[137] M. Yang, D. Zhang, and X. Feng, "Fisher discrimination dictionary learning for sparse representation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2011, pp. 543–550.

[138] L. Rosasco, A. Verri, M. Santoro, S. Mosci, and S. Villa, "Iterative projection methods for structured sparsity regularization," Massachusetts institute of technology, cambridge, MA, USA, Tech. Rep., 2009.

[139] M. Yang, L. Zhang, J. Yang, and 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.

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

[141] S. Kong and D. Wang, "A dictionary learning approach for classification: separating the particularity and the commonality," in *European Conference on Computer Vision (ECCV)*, 2012, pp. 186–199.

[142] S. Gao, I. W. Tsang, and Y. Ma, "Learning category-specific dictionary and shared dictionary for fine-grained image categorization," *IEEE Transactions on Image Processing*, vol. 23, no. 2, pp. 623–634, 2013.

[143] M. G. Jafari and M. D. Plumbley, "Fast dictionary learning for sparse representations of speech signals," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 5, pp. 1025–1031, 2011.

[144] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Supervised dictionary learning," in *Proceedings of the Advances in neural information processing systems*, 2009, pp. 1033–1040.

[145] N. Zhou, Y. Shen, J. Peng, and J. Fan, "Learning inter-related visual dictionary for object recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3490–3497.

[146] L. Ma, C. Wang, B. Xiao, and W. Zhou, "Sparse representation for face recognition based on discriminative low-rank dictionary learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2586–2593.

[147] J. Lu, G. Wang, W. Deng, and P. Moulin, "Simultaneous feature and dictionary learning for image set based face recognition," in *European Conference on Computer Vision (ECCV)*, 2014, pp. 265–280.

[148] M. Yang, D. Dai, L. Shen, and L. V. Gool, "Latent dictionary learning for sparse representation based classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 4138–4145.

[149] Z. Jiang, G. Zhang, and L. S. Davis, "Submodular dictionary learning for sparse coding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3418–3425.

[150] S. Cai, W. Zuo, L. Zhang, X. Feng, and P. Wang, "Support vector guided dictionary learning," in *European Conference on Computer Vision (ECCV)*, 2014, pp. 624–639.

[151] X. Qin, J. Shen, X. Li, and Y. Jia, "A new sparse feature-based patch for dense correspondence," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, 2014, pp. 1–6.

[152] L. He, D. Tao, X. Li, and X. Gao, "Sparse representation for blind image quality assessment," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 1146–1153.

[153] J. Yang, J. Wright, Y. Ma, and T. Huang, "Image

super-resolution as sparse representation of raw image patches," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[154] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-based super-resolution," *IEEE Computer Graphics and Applications*, vol. 22, no. 2, pp. 56–65, 2002.

[155] M. Irani and S. Peleg, "Motion analysis for image enhancement: Resolution, occlusion, and transparency," *Journal of Visual Communication and Image Representation*, vol. 4, no. 4, pp. 324–335, 1993.

[156] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Transactions on Image Processing*, vol. 19, no. 11, pp. 2861–2873, 2010.

[157] Y. Tang, Y. Yuan, P. Yan, and X. Li, "Greedy regression in sparse coding space for single-image super-resolution," *Journal of visual communication and image representation*, vol. 24, no. 2, pp. 148–159, 2013.

[158] J. Zhang, C. Zhao, R. Xiong, S. Ma, and D. Zhao, "Image super-resolution via dual-dictionary learning and sparse representation," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, 2012, pp. 1688–1691.

[159] X. Gao, K. Zhang, D. Tao, and X. Li, "Image super-resolution with sparse neighbor embedding," *IEEE Transactions on Image Processing*, vol. 21, no. 7, pp. 3194–3205, 2012.

[160] C. Fernandez-Granda and E. J. Candes, "Super-resolution via transform-invariant group-sparse regularization," in *IEEE International Conference on Computer Vision (ICCV)*, 2013, pp. 3336–3343.

[161] X. Lu, H. Yuan, P. Yan, Y. Yuan, and X. Li, "Geometry constrained sparse coding for single image super-resolution," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 1648–1655.

[162] W. Dong, G. Shi, L. Zhang, and X. Wu, "Super-resolution with nonlocal regularized sparse representation," in *Proceedings of the Visual Communications and Image Processing*, 2010, pp. 77 440H–77 440H.

[163] W. Dong, D. Zhang, and G. Shi, "Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization," *IEEE Transactions on Image Processing*, vol. 20, no. 7, pp. 1838–1857, 2011.

[164] S. Mallat and G. Yu, "Super-resolution with sparse mixing estimators," *Transactions on Image Processing*, vol. 19, no. 11, pp. 2889–2900, 2010.

[165] D. L. Donoho, "De-noising by soft-thresholding," *IEEE Transactions on Information Theory*, vol. 41, no. 3, pp. 613–627, 1995.

[166] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3736–3745, 2006.

[167] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, 2007.

[168] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Transactions on Image Processing*, vol. 17, no. 1, pp. 53–69, 2008.

[169] M. Protter and M. Elad, "Image sequence denoising via sparse and redundant representations," *IEEE Transactions on Image Processing*, vol. 18, no. 1, pp. 27–35, 2009.

[170] W. Dong, X. Li, D. Zhang, and G. Shi, "Sparsity-based image denoising via dictionary learning and structural clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 457–464.

[171] J. Jiang, L. Zhang, and J. Yang, "Mixed noise removal by weighted encoding with sparse nonlocal regularization," *IEEE Transactions on Image Processing*, vol. 23, no. 6, pp. 2651–2662, 2014.

[172] S. Gu, L. Zhang, W. Zuo, and X. Feng, "Weighted nuclear norm minimization with application to image denoising," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[173] H. Ji, C. Liu, Z. Shen, and Y. Xu, "Robust video denoising using low rank matrix completion," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 1791–1798.

[174] P. Cheng, C. Deng, S. Wang, and C. Zhang, "Image denoising via group sparse representation over learned dictionary," in *Proceedings of the Eighth International Symposium on Multispectral Image Processing and Pattern Recognition*, 2013, pp. 891 916–891 916.

[175] J. M. Bioucas-Dias and M. Figueiredo, "A new twist: two-step iterative shrinkage/thresholding algorithms for image restoration," *IEEE Transactions on Image Processing*, vol. 16, no. 12, pp. 2992–3004, 2007.

[176] J. Mairal, G. Sapiro, and M. Elad, "Learning multiscale sparse representations for image and video restoration," *Multiscale Modeling and Simulation*, vol. 7, no. 1, pp. 214–241, 2008.

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

[178] D. Zoran and Y. Weiss, "From learning models of natural image patches to whole image restoration," in *Proceedings of the IEEE International Conference on Computer Vision*, 2011, pp. 479–486.

[179] C. Bao, J. F. Cai, and H. Ji, "Fast sparsity-based orthogonal dictionary learning for image restoration," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 3384–3391.

[180] J. Zhang, D. Zhao, and W. Gao, "Group-based sparse representation for image restoration," *IEEE Transactions on Image Processing*, vol. 34, no. 9, pp. 1864–1870, 2014.

[181] W. Dong, D. Zhang, and G. Shi, "Centralized sparse representation for image restoration," in *Proceedings of the IEEE International Conference on Computer Vision*,

2011, pp. 1259–1266.

[182] W. Dong, L. Zhang, G. Shi, and X. Li, "Nonlocally centralized sparse representation for image restoration," *IEEE Transactions on Image Processing*, vol. 22, no. 4, pp. 1620–1630, 2013.

[183] A. Buades, B. Coll, and J. M. Morel, "A non-local algorithm for image denoising," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2, 2005, pp. 60–65.

[184] A. Nedic, D. Bertsekas, and A. Ozdaglar, "Convex analysis and optimization," *Athena Scientific*, 2003.

[185] I. Daubechies, M. Defrise, and C. D. Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on pure and applied mathematics*, vol. 57, no. 11, pp. 1413–1457, 2004.

[186] W. Deng, J. Hu, and J. Guo, "Extended src: Undersampled face recognition via intraclass variant dictionary," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1864–1870, 2012.

[187] ——, "In defense of sparsity based face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 399–406.

[188] R. He, W. S. Zheng, and B. G. Hu, "Maximum correntropy criterion for robust face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1561–1576, 2011.

[189] M. Yang, D. Zhang, and J. Yang, "Robust sparse coding for face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 625–632.

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

[191] S. Gao, I. W. H. Tsang, and L. T. Chia, "Kernel sparse representation for image classification and face recognition," in *European Conference on Computer Vision (ECCV)*, 2010, pp. 1–14.

[192] S. Gao, I. W. Tsang, L. T. Chia, and P. Zhao, "Local features are not lonely-laplacian sparse coding for image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 3555–3561.

[193] N. Kulkarni and B. Li, "Discriminative affine sparse codes for image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 1609–1616.

[194] C. Zhang, J. Liu, Q. Tian, C. Xu, H. Lu, and S. Ma, "Image classification by non-negative sparse coding, low-rank and sparse decomposition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 1673–1680.

[195] T. Zhang, B. Ghanem, S. Liu, C. Xu, and N. Ahuja, "Low-rank sparse coding for image classification," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 281–288.

[196] Y. Zhang, Z. Jiang, and L. S. Davis, "Learning structured low-rank representations for image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 676–683.

[197] D. Tao, L. Jin, Y. Zhao, and X. Li, "Rank preserving sparse learning for kinect based scene classification," *IEEE Transactions on Cybernetics*, vol. 43, no. 5, pp. 1406–1417, 2013.

[198] Y. Zhang, Z. Jiang, and L. S. Davis, "Discriminative tensor sparse coding for image classification," in *Proceedings of the British Machine Vision Conference*, 2013.

[199] L. Zhuang, S. Gao, J. Tang, J. Wang, Z. Lin, and Y. Ma, "Constructing a non-negative low rank and sparse graph with data-adaptive features," *arXiv preprint arXiv:1409.0964*, 2014.

[200] R. Rigamonti, V. Lepetit, G. Gonzlez, E. Turetken, F. Benmansour, M. Brown, and P. Fua, "On the relevance of sparsity for image classification," *Computer Vision and Image Understanding*, vol. 125, pp. 115–127, 2014.

[201] X. Mei and H. Ling, "Robust visual tracking using l_1 minimization," in *Proceedings of the IEEE 12th International Conference on Computer Vision*, 2009, pp. 1436–1443.

[202] ——, "Robust visual tracking and vehicle classification via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 11, pp. 2259–2272, 2011.

[203] H. Li, C. Shen, and Q. Shi, "Real-time visual tracking using compressive sensing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 1305–1312.

[204] S. Zhang, H. Yao, X. Sun, and S. Liu, "Robust object tracking based on sparse representation," in *Proceedings of the Visual Communications and Image Processing*, 2010, pp. 77 441N–77 441N–8.

[205] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Robust visual tracking via multi-task sparse learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2042–2049.

[206] X. Jia, H. Lu, and M. H. Yang, "Visual tracking via adaptive structural local sparse appearance model," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 1822–1829.

[207] B. Liu, L. Yang, J. Huang, P. Meer, L. Gong, and C. Kulikowski, "Robust and fast collaborative tracking with two stage sparse optimization," in *European Conference on Computer Vision (ECCV)*, 2010, pp. 624–637.

[208] B. Liu, J. Huang, C. Kulikowski, and L. Yang, "Robust tracking using local sparse appearance model and k-selection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 1313–1320.

[209] C. Bao, Y. Wu, H. Ling, and H. Ji, "Real time robust l_1 tracker using accelerated proximal gradient approach," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 1830–1837.

[210] W. Zhong, H. Lu, and M. H. Yang, "Robust object tracking via sparsity-based collaborative model," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 1838–1845.

[211] K. Zhang, L. Zhang, and M. Yang, "Fast compressive tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 10, pp. 2002–2015, 2014.

[212] X. Lu, Y. Yuan, and P. Yan, "Robust visual tracking with discriminative sparse learning," *Pattern Recognition*, vol. 46, no. 7, pp. 1762–1771, 2013.

[213] N. Wang, J. Wang, and D. Y. Yeung, "Online robust non-negative dictionary learning for visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 657–664.

[214] S. Zhang, H. Yao, X. Sun, and X. Lu, "Sparse coding based visual tracking: Review and experimental comparison," *Pattern Recognition*, vol. 46, no. 7, pp. 1772–1788, 2013.

[215] A. Smeulders, D. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1442–1468, 2013.

[216] F. Samaria and A. Harter, "Parameterisation of a stochastic model for human face identification," in *Proceedings of the Second IEEE Workshop on Applications of Computer Vision*, 1994, pp. 138–142.

[217] G. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," Technical Report 07-49, University of Massachusetts, Amherst, Tech. Rep., 2007.

[218] S. Wang, J. Yang, M. Sun, X. Peng, M. Sun, and C. Zhou, "Sparse tensor discriminant color space for face verification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 6, pp. 876–888, 2012.

[219] A. Georghiades, P. Belhumeur, and D. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 643–660, 2001.

[220] S. Nene, S. Nayar, and H. Murase, "Columbia object image library (coil-20)," Technical Report CUCS-005-96, Tech. Rep., 1996.

[221] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006, pp. 2169–2178.

Zheng Zhang received the B.S degree from Henan University of Science and Technology and M.S degree from Shenzhen Graduate School, Harbin Institute of Technology (HIT) in 2012 and 2014, respectively. Currently, he is pursuing the Ph.D. degree in computer science and technology at Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen, China. His current research interests include pattern recognition, machine learning and computer vision.

Yong Xu was born in Sichuan, China, in 1972. He received his B.S. degree and M.S. degree at Air Force Institute of Meteorology (China) in 1994 and 1997, respectively. He received the Ph.D. degree in Pattern recognition and Intelligence System at the Nanjing University of Science and Technology (NUST) in 2005. Now, he works at Shenzhen Graduate School, Harbin Institute of Technology. His current interests include pattern recognition, biometrics, machine learning and video analysis.

Jian Yang received the B.S. degree in mathematics from the Xuzhou Normal University in 1995. He received the M.S. degree in applied mathematics from the Changsha Railway University in 1998 and the Ph.D. degree from the Nanjing University of Science and Technology (NUST), on the subject of pattern recognition and intelligence systems in 2002. In 2003, he was a postdoctoral researcher at the University of Zaragoza. From 2004 to 2006, he was a Postdoctoral Fellow at Biometrics Centre of Hong Kong Polytechnic University. From 2006 to 2007, he was a Postdoctoral Fellow at Department of Computer Science of New Jersey Institute of Technology. Now, he is a professor in the School of Computer Science and Technology of NUST. He is the author of more than 80 scientific papers in pattern recognition and computer vision. His journal papers have been cited more than 1600 times in the ISI Web of Science, and 2800 times in the Web of Scholar Google. His research interests include pattern recognition, computer vision and machine learning. Currently, he is an associate editor of Pattern Recognition Letters and IEEE TRANSACTION ON NEURAL NETWORKS AND LEARNING SYSTEMS, respectively.

Xuelong Li is a Full Professor with the Center for OPTical IMagery Analysis and Learning (OPTIMAL), State Key Laboratory of Transient Optics and Photonics, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an 710119, Shaanxi, P.R. China.

David Zhang graduated in Computer Science from Peking University. He received his M.Sc. in Computer Science in 1982 and his Ph.D. in 1985 from the Harbin Institute of Technology (HIT). From 1986 to 1988 he was a Postdoctoral Fellow at Tsinghua University and then an Associate Professor at the Academia Sinica, Beijing. In 1994 he received his second Ph.D. in Electrical and Computer Engineering from the University of Waterloo, Ontario, Canada. Currently, he is a Chair Professor at the Hong Kong Polytechnic University where he is the Founding Director of the Biometrics Technology Centre (UGC/CRC) supported by the Hong Kong SAR Government in 1998. He also serves as Visiting Chair Professor in Tsinghua University, and Adjunct Professor in Shanghai Jiao Tong University, Peking University, Harbin Institute of Technology, and the University of Waterloo. He is the Founder and Editor-in-Chief, International Journal of Image and Graphics (IJIG); Book Editor, Springer International Series on Biometrics (KISB); Organizer, the first International Conference on Biometrics Authentication (ICBA); Associate Editor of more than ten international journals including IEEE Transactions and Pattern Recognition; Technical Committee Chair of IEEE CIS and the author of more than 10 books and 200 journal papers. Professor Zhang is a Croucher Senior Research Fellow, Distinguished Speaker of the IEEE Computer Society, and a Fellow of both IEEE and IAPR.