

Trace Quotient with Sparsity Priors for Learning Low Dimensional Image Representations

Xian Wei, *Member, IEEE*, Hao Shen, *Member, IEEE*, and Martin Kleinsteuber

Abstract—This work studies the problem of learning appropriate low dimensional image representations. We propose a generic algorithmic framework, which leverages two classic representation learning paradigms, i.e., sparse representation and the trace quotient criterion, to disentangle underlying factors of variation in high dimensional images. Specifically, we aim to learn simple representations of low dimensional, discriminant factors by applying the trace quotient criterion to well-engineered sparse representations. We construct a unified cost function, coined as the SPARse LOW dimensional representation (*SparLow*) function, for jointly learning both a sparsifying dictionary and a dimensionality reduction transformation. The *SparLow* function is widely applicable for developing various algorithms in three classic machine learning scenarios, namely, unsupervised, supervised, and semi-supervised learning. In order to develop efficient joint learning algorithms for maximizing the *SparLow* function, we deploy a framework of sparse coding with appropriate convex priors to ensure the sparse representations to be locally differentiable. Moreover, we develop an efficient geometric conjugate gradient algorithm to maximize the *SparLow* function on its underlying Riemannian manifold. Performance of the proposed *SparLow* algorithmic framework is investigated on several image processing tasks, such as 3D data visualization, face/digit recognition, and object/scene categorization.

Index Terms—Representation learning, sparse representation, trace quotient, dictionary learning, geometric conjugate gradient algorithm, supervised learning, unsupervised learning, semi-supervised learning.

1 INTRODUCTION

FINDING appropriate low dimensional representations of data is a long-standing challenging problem in data processing and machine learning. Particularly, suitable low dimensional image representations have demonstrated prominent capabilities and conveniences in various image processing applications, such as image visualization [1], [2], segmentation [3], clustering [3], [4], and classification [5], [6]. Recent developments in representation learning confirm that proper data representations are the key to success of modern machine learning algorithms. One of its major challenges is how to automatically extract suitable representations of data by employing certain general-purpose learning mechanisms to promote solutions to machine learning problems [7], [8]. In this work, we aim to develop an effective two-layer representation learning paradigm for constructing low dimensional image representations, which is capable of revealing task-specific information in image processing.

1.1 Related Work

Sparse representation is a well-known powerful tool to explore structure of images for specific learning tasks [2]. Images of interest are assumed to admit sparse representations with respect to a collection of atoms, known as a *dictionary*. Namely, each image can be constructed as a linear combination of only a few atoms. With such a model, atoms are explanatory factors that are capable of describing intrinsic structures of the images [7]. Prominent applications in image processing include image

reconstruction, super-resolution, denoising, and inpainting, e.g. [10], [11], [12]. The corresponding methods are often referred to as *data-driven sparse representation*. Moreover, for a given dictionary, sparse representations of images can be interpreted as extracting features from the images, which capture sufficient information to reconstruct the original images. By feeding the sparse representations directly to classifiers or other state of the art methods, improved performance has been observed in various image processing applications, such as face recognition [13], motion segmentation [3] and object categorization [14], [15], [16]. These observations suggest that appropriate sparse representations of images are capable of facilitating specific learning tasks. We refer to these approaches as *task-driven sparse representation*.

Performance of task-driven sparse representation methods is known to depend significantly on the construction of dictionaries. For example, for solving the problem of image classification, class-specific dictionaries can be constructed by directly selecting images from each class, either randomly [13] or according to certain structured priors [15], [16], to achieve better results than the state of the art. Dictionaries can also be learned with respect to specific criteria, such as locality of images [14], [17], class-specific discrimination [18], optimal Fisher discrimination criterion [19], and maximal mutual information [20], to further improve performance in image classification. Alternatively, sparse representations can also be combined with the classical expected risk minimization formulation, such as *least squares loss* [21], [22], [23], *logistic loss* [24], and *square hinge loss* [25]. In the literature, these methods are referred to as *task-driven dictionary learning (TDDL)* [24]. Note, that these methods are mainly devoted to supervised learning.

All aforementioned approaches often solve a problem of learning both a dictionary and a task-specific parameter, either sequentially or simultaneously. Sparse representations of images are treated as inputs to task-specific learning algorithms, such

- X. Wei is with Fujian Institute of Research on the Structure of Matter, Chinese Academy of Sciences (CAS), China and Technical University of Munich, Germany. E-mail: xian.wei@tum.de
- H. Shen (corresponding author) is with fortiss GmbH, Munich, and the Technical University of Munich, Germany. E-mail: shen@fortiss.de
- M. Kleinsteuber is with the Technical University of Munich, Germany and Mercateo AG, Munich, Germany. E-mail: kleinsteuber@tum.de

as classifiers or predictors. Early work in [26] applies a linear projection to extract low dimensional features that preserve inner products of pair-wise sparse coefficients for facilitating image classification. A similar approach in [27] shows that low dimensional features of sparse representations of images, obtained by applying Principal Component Analysis (PCA) to sparse representations, are capable of enhancing performance in image visualization and clustering. More recently, applying the spectral clustering framework to sparse representations of images yields the so-called *Sparse Subspace Clustering* method [3], leading to promising results in motion segmentation and face clustering. All these observations indicate that low dimensional features of sparse representations can be more task-specific in achieving higher performance. Such a phenomenon can be studied in a more general framework of *disentangled representation learning* [55], which aims to disentangle underlying factors of variation in data [7], [8], [28]. Specifically, sparse representations of images can be regarded as a tool for modelling such factors in a first instance. In a second instance, a task-dependent combination of the sparse coefficients then has the potential to further improve the images' representations. So far, such representation learning paradigms are often constructed as a separated two-stage unsupervised encoding scheme, i.e., the further disentangling instrument is independent from the layer of sparse representation. Although sparse coding can also be utilized to construct a deep learning architecture for extracting more abstract representations [7], such an approach often requires special computing hardware and a very large amount of training images, resulting in prohibitive training efforts compared to two-layer learning approaches.

1.2 Motivation and Main Contributions

Generally speaking, in machine learning, we aim to enable a computer to assign meaningful semantics to a given event. Let us denote by Ω the space of events, and by \mathcal{Z} the space of semantics. For example, the space of events can include positional cues in a 3D scene, while the semantics can be subject classes. The event space Ω is observed through some sensors (e.g. a camera), so we model observations as a random variable $X: \Omega \rightarrow \mathcal{X}$, where we can consider \mathcal{X} to be some subset of \mathbb{R}^m . Specifically, the goal of machine learning is to find a map F from \mathcal{X} to \mathcal{Z} . It is important in practice that such a map is conceptually simple, meaning that realizations that belong to the same semantic are close to each other, or realizations that correspond to different semantics are easily separable, etc. In other words, the distribution of \mathcal{X} should allow for a conceptually simple map to extract desired information. We call such a distribution *disentangled with respect to the semantics of interest*. In this setting, the problem of representation learning can be interpreted as learning a map F that transforms the raw data $\mathbf{X} \subset \mathcal{X}$ in order to obtain a distribution that is more disentangled with respect to the semantics of interest. For simplicity, we identify the space of semantics \mathcal{Z} directly with this representation space.

In this paper, we propose a framework to learn a map F for images that outputs aforementioned disentangled representations. The central idea is that F consists of a concatenation of a sparse coding step, which is inherently non-linear, followed by a linear projection. The parameters of F are learned jointly by optimizing a trace quotient (TQ) criterion, which also allows to include information about the semantics of interest.

This generic criterion is shared by various classic dimensionality reduction (DR) methods, including PCA, Linear Discriminant

Analysis (LDA) [5], Linear Local Embedding (LLE) [2], Marginal Fisher Analysis (MFA) [6], Orthogonal Neighborhood Preserving Projection (ONPP) [31], Locality Preserving Projections (LPP) [4], Orthogonal LPP (OLPP) [32], Spectral Clustering (SC) [3], semi-supervised LDA (SDA) [33], etc.

Recently, the authors of this paper proposed to employ the TQ criterion to further disentangle sparse representations of image data to facilitate unsupervised learning tasks [34], and developed a joint disentangling framework, coined as SPARse LOW dimensional representation learning (*SparLow*). It has been further tested on supervised learning tasks [35]. Although numerical evidences of the proposed framework have been provided in these two works, a complete investigation of the *SparLow* learning framework has not been systematically conducted. Hence, the goal of this work is to fully investigate the potential of the *SparLow* framework as a fundamental representation learning instrument in a broad spectrum of machine learning, i.e., *supervised*, *unsupervised*, and *semi-supervised learning*, and to further explore its capacity in image processing applications. The main contributions are as follows:

- 1) The difficulty of constructing an efficient algorithm to optimize the *SparLow* cost function lies in the differentiability of sparse representations with respect to a given dictionary. To address this issue, we consider the sparse coding problem by minimizing a quadratic reconstruction error with appropriate convex sparsity priors. We show that when sparse representations of all data samples are unique, then the sparse representations can be interpreted as a locally differentiable function with respect to the dictionary, and the first derivative of such a sparse representation has a closed-form expression.
- 2) The joint disentanglement problem results in solving an optimization problem that is defined on an underlying Riemannian manifold. Although optimization on Riemannian manifolds is nowadays well established, derivation of a Riemannian Conjugate Gradient (CG) *SparLow* algorithm remains a sophisticated task. In this work, we present a generic framework of Riemannian CG *SparLow* algorithms with necessary technical details, so that both theorists and practitioners can benefit from it.
- 3) Finally, sensitivity of the *SparLow* framework with respect to its parameters is investigated by numerical experiments.

Compared to state of the art *TDDL* methods, the proposed *SparLow* framework shares the following three merits: i) It introduces a generic formulation for learning both a dictionary and an orthogonal DR transformation in unsupervised, supervised and semi-supervised learning settings, in contrast to the existing supervised *TDDL* approaches [21], [22], [24], [25] and two-layer unsupervised learning framework [3], [27]; ii) Compared to the popular class-wise sparse coding approaches, which often learn one dictionary for each class [18], and employ a set of binary classifiers in either “one-versus-all” or “one-versus-one” scheme for multiclass classification [24], [25], *SparLow* only learns a compact dictionary and an orthogonal projection for all classes. It significantly reduces the computational complexity and memory burden for classifications problems with many classes; iii) Different to the *TDDL* methods that compute the sparse representation by using the sparsity prior of Lasso or Elastic Net [36], *SparLow* allows for more general convex sparsity priors.

The paper is organized as follows. Section 2 provides a brief review on both sparse representations and the trace quotient

optimization. In Section 3, we construct a generic cost function for learning both a sparsifying dictionary and an orthogonal DR transformation, and then discuss its several exemplifications in Section 4. A geometric CG algorithm is developed in Section 5, together with their experimental evaluations presented in Section 6. Finally, conclusions and outlook are given in Section 7.

2 SPARSE CODING AND TRACE QUOTIENT

In this section, we first briefly review some state of the art results of sparse representations with convex priors, which formulate the first layer of the *SparLow*. Then, the TQ optimization based dimensionality reduction is presented, which is used to construct the second layer of the *SparLow*. The presented hypothesis, restrictions and reformulations of both sparse representations and dimensionality reduction enable us to develop the joint learning paradigm of the *SparLow*, which is fully investigated in Sections 3, 4 and 5.

We start with an introduction to notations and definitions used in the paper. In this paper, we denote sets and manifolds with fraktur letters, such as \mathfrak{P} , \mathfrak{S} , and \mathfrak{G} , and $|\mathfrak{P}|$ the cardinality of the set \mathfrak{P} . Matrices are written as boldface capital letters like \mathbf{X} , Φ , column vectors are denoted by boldfaced small letters, e.g. \mathbf{x} , \mathbf{d} , whereas scalars are either capital or small letters, such as n and N . We denote by \mathbf{I}_n the $n \times n$ -identity matrix, $(\cdot)^\top$ the matrix transpose, $\text{tr}(\cdot)$ the trace of a square matrix. Furthermore, $\|\cdot\|_1$ and $\|\cdot\|_2$ denote the ℓ_1 -, ℓ_2 -norm of a vector, and $\|\cdot\|_F$ the Frobenius norm of a matrix.

2.1 Sparse Coding with Convex Priors

Let $\mathbf{X} := [\mathbf{x}_1, \dots, \mathbf{x}_n] \in \mathbb{R}^{m \times n}$ be a collection of n data points in \mathbb{R}^m . Sparse coding aims to find a collection of *atoms* $\mathbf{d}_i \in \mathbb{R}^m$ for $i = 1, \dots, r$, such that each data point can be approximated by a linear combination of a small subset of atoms. Specifically, by denoting $\mathbf{D} := [\mathbf{d}_1, \dots, \mathbf{d}_r] \in \mathbb{R}^{m \times r}$, referred to as a *dictionary*, all data samples \mathbf{x}_i for all $i = 1, \dots, n$ are assumed to be modeled as

$$\mathbf{x}_i = \mathbf{D}\phi_i + \epsilon_i, \quad (1)$$

where $\phi_i \in \mathbb{R}^r$ is the corresponding sparse representation of \mathbf{x}_i , and $\epsilon_i \in \mathbb{R}^m$ is a small additive residual, such as noise.

One popular solution of the sparse coding problem above is given by solving the following minimization problem

$$\min_{\mathbf{D} \in \mathbb{R}^{m \times r}, \Phi \in \mathbb{R}^{r \times n}} \sum_{i=1}^n \frac{1}{2} \|\mathbf{x}_i - \mathbf{D}\phi_i\|_2^2 + g(\phi_i), \quad (2)$$

with $\Phi := [\phi_1, \dots, \phi_n] \in \mathbb{R}^{r \times n}$. Here, the first term penalizes the reconstruction error of sparse representations, and the second term is a sparsity promoting regularizer. Often, the function g is chosen to be separable, i.e., its evaluation is computed as the sum of functions of the individual components of its argument.

Definition 1 (Separable sparsity regularizer). *Let $\phi := [\phi_1, \dots, \phi_r]^\top \in \mathbb{R}^r$. A function $g: \mathbb{R}^r \rightarrow [0, +\infty)$ is a separable sparsity regularizer if*

$$g(\phi) := \sum_{i=1}^r g_i(\phi_i), \quad (3)$$

with $g_i(\phi_i) \geq 0$ and $g_i(0) = 0$.

There are many choices for g in the literature, such as the ℓ_0 -(quasi)-norm and its variations. It is important to notice that an

optimization procedure to solve the problem as in Eq. (2) will force the norm of columns of \mathbf{D} to infinity, and consequently drive the value $g(\phi_i)$ to zero. To avoid such trivial solutions, it is common to restrict all atoms $\mathbf{d}_i \in \mathbb{R}^m$ to have unit norm, i.e., the set of dictionaries is a product manifold of r times the $(m-1)$ -dimensional unit sphere, i.e.,

$$\mathfrak{S}(m, r) := \{\mathbf{D} \in \mathbb{R}^{m \times r} \mid \|\mathbf{d}_i\|_2 = 1\}. \quad (4)$$

If the dictionary \mathbf{D} is fixed, Problem (2) degenerates into a collection of decoupled sample-wise *sparse regression* problems. Specifically, for each sample \mathbf{x} , we have

$$\min_{\phi \in \mathbb{R}^r} f_{\mathbf{x}}(\phi) := \frac{1}{2} \|\mathbf{x} - \mathbf{D}\phi\|_2^2 + g(\phi). \quad (5)$$

If the sparsity regularizer g is *strictly convex*, then the *sparse regression* problem (5) has a unique solution. Thus, the solution of the sample-wise sparse regression problem can be treated as a function in \mathbf{x} , i.e.,

$$\phi_{\mathbf{D}}(\mathbf{x}) := \underset{\phi \in \mathbb{R}^r}{\text{argmin}} f_{\mathbf{x}}(\phi). \quad (6)$$

With sparse representations of all samples being calculated, specific learning algorithms can be directly applied to these sparse coefficients to extract further representations.

By choosing the function g to be the *elastic net regularizer* [36], i.e.,

$$g_e(\phi) := \lambda_1 \|\phi\|_1 + \lambda_2 \|\phi\|_2^2, \quad (7)$$

where parameters $\lambda_1, \lambda_2 > 0$ are chosen to ensure stability and uniqueness of the sparse solution, for a given dictionary \mathbf{D} , the solution $\phi_{\mathbf{D}}(\mathbf{x}_i)$ as in Eq. (6) can be considered as a function in \mathbf{D} , i.e., $\phi_{\mathbf{D}}: \mathbb{R}^m \rightarrow \mathbb{R}^r$, with a closed-form expression [24]. Moreover, the sparse representation $\phi_{\mathbf{D}}$ is locally differentiable with respect to the dictionary \mathbf{D} . Such a convenient result has lead to a joint learning approach to optimize the cost function associated with the *TDDL* methods, e.g. [24], [25].

Unfortunately, for a general choice of g , there is no guarantee to have a closed-form expression of the sparse representation. Recent work in [37] proposes the choice of g to be a full unnormalized Kullback-Leibler (KL) divergence. Although the corresponding sparse representation has no closed-form expression, its first derivative does have an explicit formula, which enables the development of gradient-based optimization algorithms. It is worth noticing that both the elastic net regularizer and the full unnormalized KL divergence regularizer belong to the category of convex sparsity regularizer. Hence, we hypothesize that the choice of an appropriate convex sparsity prior can facilitate the development of efficient joint learning algorithms for discovering abstract representations of sparse coefficients of images. Such a hypothesis is proven to be true in Section 5.1.

2.2 Optimization of the Trace Quotient Criterion

Classic DR methods aim to find a lower-dimensional representation $\mathbf{y}_i \in \mathbb{R}^l$ of given data samples $\mathbf{x}_i \in \mathbb{R}^m$ with $l < m$, via a mapping $\mu: \mathbb{R}^m \rightarrow \mathbb{R}^l$, which captures certain application dependent properties of the data. Many classic DR methods restrict the mapping μ to be an orthogonal projection. Let us denote the set of $m \times l$ orthonormal matrices by

$$\mathfrak{S}(l, m) := \{\mathbf{U} \in \mathbb{R}^{m \times l} \mid \mathbf{U}^\top \mathbf{U} = \mathbf{I}_l\}. \quad (8)$$

Specifically, in this work, we confine ourselves to the form of orthogonal projections as $\mu(\mathbf{x}) := \mathbf{U}^\top \mathbf{x}$. This model covers

a wide range of classic supervised and unsupervised learning methods, such as LDA, MFA, PCA, OLPP, and ONPP. Further details are given in Section 4.

One generic algorithmic framework to find optimal $\mathbf{U} \in \mathfrak{St}(l, m)$ is formulated as a maximization problem of the so-called *trace quotient* or *trace ratio*, i.e.,

$$\operatorname{argmax}_{\mathbf{U} \in \mathfrak{St}(l, m)} \frac{\operatorname{tr}(\mathbf{U}^\top \mathbf{A} \mathbf{U})}{\operatorname{tr}(\mathbf{U}^\top \mathbf{B} \mathbf{U}) + \sigma}, \quad (9)$$

where matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m \times m}$ are often symmetric positive semidefinite, and constant $\sigma > 0$ is chosen to prevent the denominator from being zero. Both matrices \mathbf{A} and \mathbf{B} are constructed to measure the “similarity” between data points according to the specific problems, e.g. [31], [38]. Specifically, in following sections, they are represented as smooth functions to measure the discrepancy between sparse coefficient pairs $(\phi_{\mathbf{D}}(\mathbf{x}_i), \phi_{\mathbf{D}}(\mathbf{x}_j))$, $\forall i, j$. The methodological details and examples will be given and discussed in Sections 3 and 4.

It is obvious that solutions of the problem in Eq. (9) are rotation invariant, i.e., let $\mathbf{U}^* \in \mathfrak{St}(l, m)$ be a solution of the problem, then so is $\mathbf{U}^* \mathbf{\Theta}$ for any $\mathbf{\Theta} \in \mathbb{R}^{l \times l}$ being orthogonal. In other words, the solution set of the problem in Eq. (9) is the set of all l -dimensional linear subspaces in \mathbb{R}^m . In order to cope with this structure, we employ the Grassmann manifold, which can be alternatively identified as the set of all m -dimensional rank- l orthogonal projectors, i.e.,

$$\mathfrak{Gr}(l, m) := \{\mathbf{U} \mathbf{U}^\top \mid \mathbf{U} \in \mathfrak{St}(l, m)\}. \quad (10)$$

Thus, the trace quotient problem can be formulated as

$$\operatorname{argmax}_{\mathbf{P} \in \mathfrak{Gr}(l, m)} \frac{\operatorname{tr}(\mathbf{P} \mathbf{A})}{\operatorname{tr}(\mathbf{P} \mathbf{B}) + \sigma}. \quad (11)$$

Although various efficient optimization algorithms have been developed to solve the trace quotient problem, see [31], [38], [39], the construction described in the next section requires further nontrivial, constructive development.

3 A JOINT DISENTANGLING FRAMEWORK

In this section, we construct the *SparLow* cost function, which adopts the construction of sparse coding with convex priors in the framework of TQ maximization to extract low dimensional representations of sparse codings of images.

The *SparLow* function allows to jointly learn both a sparsifying dictionary and an orthogonal projection in the framework of TQ maximization. Let us denote by $\Phi(\mathbf{D}, \mathbf{X}) := [\phi_{\mathbf{D}}(\mathbf{x}_1), \dots, \phi_{\mathbf{D}}(\mathbf{x}_n)] \in \mathbb{R}^{r \times n}$ the sparse representation of the data $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n]$ for a given dictionary \mathbf{D} computed by solving the sparse regression problems as in Eq. (6). Let $\mathcal{A}: \mathbb{R}^{r \times n} \rightarrow \mathbb{R}^{r \times r}$ and $\mathcal{B}: \mathbb{R}^{r \times n} \rightarrow \mathbb{R}^{r \times r}$ be two smooth functions that serve as generating functions for the matrices \mathbf{A} and \mathbf{B} in the trace quotient in Eq. (11). Constructions of the two structure matrix-valued function \mathcal{A} and \mathcal{B} are according to the specific learning tasks and exemplified in Section 4. We can define a generic trace quotient function on sparse representations as

$$f: \mathfrak{S}(m, r) \times \mathfrak{Gr}(l, r) \rightarrow \mathbb{R} \\ f(\mathbf{D}, \mathbf{P}) := \frac{\operatorname{tr}(\mathbf{P} \mathcal{A}(\Phi(\mathbf{D}, \mathbf{X})))}{\operatorname{tr}(\mathbf{P} \mathcal{B}(\Phi(\mathbf{D}, \mathbf{X}))) + \sigma}. \quad (12)$$

From the perspective of learning representations, the projection \mathbf{P} aims to capture low-dimensional discriminant features in sparse

representations of images. It is important to notice that the function f is not necessarily differentiable, unless the structure functions \mathcal{A} and \mathcal{B} are differentiable in the dictionary \mathbf{D} , i.e., the sparse representations $\Phi(\mathbf{D}, \mathbf{X})$ are differentiable in \mathbf{D} . This issue is further discussed in Section 5.1.

In order to prevent solution dictionaries from being highly coherent, which is necessary for guaranteeing the local smoothness of sparse solutions [12], we employ a log-barrier function on the scalar product of all dictionary columns to control the mutual coherence of the learned dictionary \mathbf{D} , i.e., for dictionary $\mathbf{D} = [\mathbf{d}_1, \dots, \mathbf{d}_r] \in \mathbb{R}^{m \times r}$, we define

$$g_c(\mathbf{D}) := - \sum_{1 \leq i < j \leq r} \frac{1}{2} \log(1 - (\mathbf{d}_i^\top \mathbf{d}_j)^2). \quad (13)$$

It is worth noticing that the interplay between the exactness of sparse representations of images $\Phi(\mathbf{D}, \mathbf{X})$ and the measure of discrimination of TQ is indirect. As observed in preliminary experiments of this work, it is very difficult to ensure maximization of the function f regularized by g_c to extract good low-dimensional representations of images for the learning tasks. Namely, sparse representations $\Phi(\mathbf{D}, \mathbf{X})$ are data-driven disentangling factors of variation in the first layer, which carry data information independently from the (potentially) task-specific second layer. Such an observation might also be interpreted as overfitting of sparse representations to the second layer’s target. In order to deal with this problem, we propose to adopt the *warm start* strategy from an optimal data-driven dictionary, and restrict the new dictionary to lie in a neighborhood of the warm start to explicitly balance both data-driven and task-driven information. Specifically, we propose the following regularizer on the dictionary as

$$g_d(\mathbf{D}) := \frac{1}{2} \|\mathbf{D} - \mathbf{D}^*\|_F^2, \quad (14)$$

where \mathbf{D}^* is the optimal data-driven dictionary learned from the data \mathbf{X} . In the rest of the paper, we refer to it as the *data regularizer*. It measures the distance between an estimated dictionary \mathbf{D} and the dictionary \mathbf{D}^* in terms of the Frobenius norm. Practically, we set \mathbf{D}^* to be a dictionary produced by state of the art methods, such as K-SVD [10]. Our experiments have verified that g_d guarantees stable performance of *SparLow* algorithms, see Section 6.2.1.

To summarize, we construct the following cost function to jointly learn both a sparsifying dictionary and an orthogonal transformation, i.e.,

$$J: \mathfrak{S}(m, r) \times \mathfrak{Gr}(l, r) \rightarrow \mathbb{R} \\ J(\mathbf{D}, \mathbf{P}) := f(\mathbf{D}, \mathbf{P}) - \mu_1 g_c(\mathbf{D}) - \mu_2 g_d(\mathbf{D}), \quad (15)$$

where the two weighting factors $\mu_1 > 0$ and $\mu_2 > 0$ control the influence of the two regularizers on the final solution. In this work, we refer to it as the *SparLow* function.

4 EXEMPLIFICATIONS OF THE *SparLow* MODEL

In the previous section, we construct the *SparLow* function for extracting low dimensional representations of sparse codings of images. In what follows, we exemplify counterparts of several classic unsupervised, supervised and semi-supervised learning methods by constructing various structure smooth functions \mathcal{A} and \mathcal{B} in Eq. (12).

4.1 Unsupervised *SparLow*

We firstly introduce three unsupervised learning methods within the *SparLow* framework.

4.1.1 PCA-like SparLow

The standard PCA method computes an orthogonal transformation $\mathbf{U} \in \mathfrak{St}(l, m)$, so that the variance of the low dimensional representations of the data \mathbf{X} is maximized, i.e., \mathbf{U} is the solution of the following maximization problem

$$\max_{\mathbf{U} \in \mathfrak{St}(l, m)} \text{tr}(\mathbf{U}^\top \mathbf{X} \mathbf{\Pi}_n \mathbf{X}^\top \mathbf{U}), \quad (16)$$

where $\mathbf{\Pi}_r := \mathbf{I}_r - \frac{1}{r} \mathbf{1}_r \mathbf{1}_r^\top$ is the centering matrix in \mathbb{R}^r with $\mathbf{1}_r = [1, \dots, 1]^\top \in \mathbb{R}^r$. In the framework of trace quotient, the denominator is trivially a constant, i.e.,

$$\text{tr}(\mathbf{U}^\top \mathbf{B}_{\text{pca}} \mathbf{U}) = \text{tr}(\mathbf{X} \mathbf{\Pi}_n \mathbf{X}^\top), \quad (17)$$

with $\mathbf{B}_{\text{pca}} = \text{tr}(\mathbf{X} \mathbf{\Pi}_n \mathbf{X}^\top) \mathbf{I}_n$. Thus, the standard PCA can be rewritten in the form of trace quotient as

$$\begin{cases} \mathcal{A}_{\text{pca}}(\mathbf{X}) := \mathbf{X} \mathbf{\Pi}_n \mathbf{X}^\top \\ \mathcal{B}_{\text{pca}}(\mathbf{X}) := \text{tr}(\mathbf{X} \mathbf{\Pi}_n \mathbf{X}^\top) \mathbf{I}_n. \end{cases} \quad (18)$$

By adopting the sparse representations $\Phi(\mathbf{D}, \mathbf{X})$ (by Φ for short) to substitute \mathbf{X} in Eq. (18), the two structure functions are defined as

$$\begin{cases} \mathcal{A}_{\text{pca}}(\Phi) := \Phi \mathbf{\Pi}_n \Phi^\top \\ \mathcal{B}_{\text{pca}}(\Phi) := \text{tr}(\Phi \mathbf{\Pi}_n \Phi^\top) \mathbf{I}_r. \end{cases} \quad (19)$$

We refer to the corresponding *SparLow* algorithm as the *PCA-SparLow* algorithm.

It is worth noticing that both the *SparLow* formulations and their original ones have the similar trace quotient structures, such as Eq. (19) and Eq. (18), the original *SparLow* formulations (e.g., Eq. (18)) will not be redundantly presented in the rest of the section.

4.1.2 LLE-like SparLow

The original LLE algorithm aims to find low dimensional representations of the data via fitting directly the barycentric coordinates of a point based on its neighbors constructed in the original data space [2]. It is well known that the low dimensional representations in the LLE method can only be computed implicitly. Therefore, the so-called ONPP method introduces an explicit orthogonal transformation between the original data and its low dimensional representation [31]. Specifically, the ONPP method solves the following problem

$$\min_{\mathbf{U} \in \mathfrak{St}(l, m)} \text{tr}(\mathbf{U}^\top \mathbf{X} \mathbf{Z}_{\text{lle}} \mathbf{X}^\top \mathbf{U}), \quad (20)$$

where $\mathbf{Z}_{\text{lle}} = (\mathbf{I}_n - \mathbf{W})^\top (\mathbf{I}_n - \mathbf{W})$ with $\mathbf{W} \in \mathbb{R}^{n \times n}$ being the matrix of barycentric coordinates of the data. Similar to the construction for *PCA-SparLow*, we construct the following functions for an LLE-like *SparLow* approach

$$\begin{cases} \mathcal{A}_{\text{lle}}(\Phi) := \Phi \mathbf{Z}_{\text{lle}} \Phi^\top \\ \mathcal{B}_{\text{lle}}(\Phi) := \text{tr}(\Phi \mathbf{Z}_{\text{lle}} \Phi^\top) \mathbf{I}_r. \end{cases} \quad (21)$$

4.1.3 Laplacian SparLow

Another popular category of unsupervised learning methods are the ones involving a Laplacian matrix of data, e.g. Locality Preserving Projection (LPP) [4], Orthogonal LPP (OLPP) [32], Linear Graph Embedding (LGE) [6], and Spectral Clustering [3]. Let us denote by $d_{ij} := \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2/t)$ the Laplacian similarity between two data points \mathbf{x}_i and \mathbf{x}_j with constant $t > 0$. Similar

to the approaches applied above, we adopt a simple formulation by setting

$$\begin{cases} \mathcal{A}_{\text{lap}}(\Phi) := \Phi \mathbf{Z}_{\text{lap}} \Phi^\top \\ \mathcal{B}_{\text{lap}}(\Phi) := \Phi \mathbf{Y}_{\text{lap}} \Phi^\top, \end{cases} \quad (22)$$

with $\mathbf{Z}_{\text{lap}} := \{z_{ij}\} \in \mathbb{R}^{n \times n}$ being a real symmetric matrix measuring the similarity between data pairs $(\mathbf{x}_i, \mathbf{x}_j)$, and $\mathbf{Y}_{\text{lap}} := \{y_{ii}\} \in \mathbb{R}^{n \times n}$ being a diagonal matrix having $y_{ii} := \sum_{j \neq i} z_{ij}$ for all i, j . Specifically, the similarity matrix \mathbf{Z}_{lap} can be computed by applying a Gaussian kernel function on the distance between two data samples, i.e., $z_{ij} = d_{ij}$ if ϕ_i and ϕ_j are adjacent, $z_{ij} = 0$ otherwise.

4.2 Supervised SparLow

In this subsection, we focus on the supervised *SparLow* learning for solving classification problems. Assume that there are c classes of images. Let $\mathbf{X}_i = [\mathbf{x}_{i1}, \dots, \mathbf{x}_{in_i}] \in \mathbb{R}^{m \times n_i}$ for $i = 1, \dots, c$ with n_i being the number of samples in the i^{th} class. The corresponding sparse coefficients are denoted by $\Phi_i := [\phi_{i1}, \dots, \phi_{in_i}] \in \mathbb{R}^{r \times n_i}$, and $\Phi := [\Phi_1, \dots, \Phi_c] \in \mathbb{R}^{r \times n}$ with $n = \sum_{i=1}^c n_i$.

4.2.1 LDA SparLow

The classic LDA algorithm [5] aims to find low-dimensional representations of the high dimensional data, so that the between-class scatter is maximized, while the within-class scatter is minimized. Let us define by $\bar{\phi}_i \in \mathbb{R}^r$ the center of the i -th class. The within-class scatter matrix is computed as

$$\begin{aligned} \mathcal{B}_{\text{lda}}(\Phi(\mathbf{D}, \mathbf{X})) &= \sum_{i=1}^c \sum_{j=1}^{n_i} (\phi_{ij} - \bar{\phi}_i)(\phi_{ij} - \bar{\phi}_i)^\top \\ &= \sum_{i=1}^c \Phi_i \mathbf{\Pi}_{n_i} \Phi_i^\top = \Phi \mathbf{L}^w \Phi^\top, \end{aligned} \quad (23)$$

with $\mathbf{L}^w := \text{diag}(\mathbf{\Pi}_{n_1}, \dots, \mathbf{\Pi}_{n_c}) \in \mathbb{R}^{n \times n}$ being a block diagonal matrix, whose diagonal blocks are the centering matrices in \mathbb{R}^{n_i} , associated with the corresponding classes. Let $\bar{\phi} \in \mathbb{R}^r$ be the centre of all classes. Then, we can define the between-class scatter matrix as

$$\begin{aligned} \mathcal{A}_{\text{lda}}(\Phi) &= \sum_{i=1}^c n_i (\bar{\phi}_i - \bar{\phi})(\bar{\phi}_i - \bar{\phi})^\top \\ &= \left[\phi_1 \frac{1_{n_1}}{\sqrt{n_1}}, \dots, \phi_c \frac{1_{n_c}}{\sqrt{n_c}} \right] \mathbf{\Pi}_c \left[\phi_1 \frac{1_{n_1}}{\sqrt{n_1}}, \dots, \phi_c \frac{1_{n_c}}{\sqrt{n_c}} \right]^\top \\ &= \Phi \mathbf{L}^b \Phi^\top, \end{aligned} \quad (24)$$

where $\mathbf{L}^b := \mathbf{C}^b \mathbf{\Pi}_c (\mathbf{C}^b)^\top$ with $\mathbf{C}^b = \text{Bdiag}(\frac{1_{n_1}}{\sqrt{n_1}}, \dots, \frac{1_{n_c}}{\sqrt{n_c}}) \in \mathbb{R}^{n \times c}$.

4.2.2 MFA SparLow

Marginal Fisher Analysis (MFA) [6], also known as Linear Discriminant Embedding (LDE), is the supervised version of the Laplacian Eigenmaps [31]. The main idea is to maintain the original neighbor relations of points from the same class while pushing apart the neighboring points of different classes.

Let $\mathfrak{N}_{k_1}^+(\phi_i)$ denote the set of k_1 nearest neighbors which share the same label with ϕ_i , and $\mathfrak{N}_{k_2}^-(\phi_i)$ denote the set of k_2 nearest neighbors among the data points whose labels are different

to that of ϕ_i . We construct two matrices $\mathbf{Z}_{\text{mfa}}^+ := \{z_{ij}^+\} \in \mathbb{R}^{n \times n}$ and $\mathbf{Z}_{\text{mfa}}^- := \{z_{ij}^-\} \in \mathbb{R}^{n \times n}$ with

$$z_{ij}^+ = \begin{cases} 1, & \phi_j \in \mathcal{N}_{k_1}^+(\phi_i) \text{ or } \phi_i \in \mathcal{N}_{k_1}^+(\phi_j), \\ 0, & \text{otherwise,} \end{cases} \quad (25)$$

and

$$z_{ij}^- = \begin{cases} 1, & \phi_j \in \mathcal{N}_{k_1}^-(\phi_i) \text{ or } \phi_i \in \mathcal{N}_{k_1}^-(\phi_j), \\ 0, & \text{otherwise.} \end{cases} \quad (26)$$

Then, the Laplacian matrices for characterizing the inter-class and intra-class locality are defined as

$$\mathbf{L}_{\text{mfa}}^- = \mathbf{Y}_{\text{mfa}}^- - \mathbf{Z}_{\text{mfa}}^-, \quad \text{and} \quad \mathbf{L}_{\text{mfa}}^+ = \mathbf{Y}_{\text{mfa}}^+ - \mathbf{Z}_{\text{mfa}}^+, \quad (27)$$

where \mathbf{Y}^+ and \mathbf{Y}^- are two diagonal matrices defined as

$$y_{ii}^+ = \sum_{j \neq i} z_{ij}^+, \quad \text{and} \quad y_{ii}^- = \sum_{j \neq i} z_{ij}^-. \quad (28)$$

Then, we construct the following functions for MFA-like *SparLow* approach, i.e.,

$$\begin{cases} \mathcal{A}_{\text{mfa}}(\Phi) := \Phi \mathbf{L}_{\text{mfa}}^- \Phi^\top \\ \mathcal{B}_{\text{mfa}}(\Phi) := \Phi \mathbf{L}_{\text{mfa}}^+ \Phi^\top. \end{cases} \quad (29)$$

4.2.3 MVR SparLow

Many challenging problems, e.g. multi-label classification, can be modeled as multivariate ridge regression (MVR) by solving the following minimization problem

$$\min_{\mathbf{Z}, \mathbf{U}, \mathbf{W}} \|\mathbf{Z} - \mathbf{W}^\top \mathbf{U}^\top \Phi(\mathbf{D}, \mathbf{X})\|_F^2 + \mu \|\mathbf{W}\|_F^2, \quad (30)$$

where $\mathbf{Z} \in \mathbb{R}^{d \times n}$ is the target matrix, $\mathbf{U} \in \mathcal{G}(l, r)$, $\mathbf{W} \in \mathbb{R}^{l \times d}$ and $\mu \in \mathbb{R}^+$. By freezing both \mathbf{Z} and \mathbf{U} , a solution to the problem as in Eq. (30) with respect to \mathbf{W} has a closed form expression as

$$\mathbf{W} = (\mathbf{U}^\top (\Phi \Phi^\top + \mu \mathbf{I}_r) \mathbf{U})^{-1} \mathbf{U}^\top \Phi \mathbf{Z}^\top. \quad (31)$$

Using this closed expression to substitute \mathbf{W} in Eq. (30), we can rewrite Eq. (30) in the form of the *SparLow* as

$$\begin{cases} \mathcal{A}_{\text{mvr}}(\Phi) := -\Phi \mathbf{Z}_{\text{mvr}}^\top \mathbf{Z}_{\text{mvr}} \Phi^\top \\ \mathcal{B}_{\text{mvr}}(\Phi) := (\Phi (\Phi^\top + \mu \mathbf{I}_r)). \end{cases} \quad (32)$$

Therein, \mathbf{Z}_{mvr} can be constructed as the binary class labels of input signals, which is usually coded as $\mathbf{Z}_{\text{mvr}} \in \mathbb{R}^{c \times n}$ with $\mathbf{z}_i = [z_{i1}, \dots, z_{ic}]^\top$, $z_{ij} = 1$ if \mathbf{z}_i is in class j , $z_{ij} = 0$ otherwise. Alternatively, \mathbf{Z}_{mvr} can also be a handcrafted indicator matrix according to labels, e.g. the “discriminative” sparse codes in [22].

4.3 Semi-supervised SparLow

In this subsection, we demonstrate that the *SparLow* model is also well suited to exploit unlabeled data in a semi-supervised setting. Assume that there are n_l labeled samples $\mathbf{X}_l \in \mathbb{R}^{m \times n_l}$ and n_u unlabeled samples $\mathbf{X}_u \in \mathbb{R}^{m \times n_u}$, with $n = n_l + n_u$. For a given dictionary \mathbf{D} , we denote by $\Phi_l \in \mathbb{R}^{r \times n_l}$ and $\Phi_u \in \mathbb{R}^{r \times n_u}$ the corresponding sparse coefficients. The first assumption to support *semi-supervised SparLow model* is that the learned dictionary for a specific class is also effective for learning good sparse features from unlabeled data [24]. Secondly, we follow the way that learning semi-supervised DR settings associated with preserving the global data manifold structure, namely, nearby points will have

similar lower-dimensional representations [33], [40] or labels [41], [42].

Given the whole dataset \mathbf{X} , let us define the graph Laplacian matrix $\mathbf{L} = \mathbf{Y} - \mathbf{Z} \in \mathbb{R}^{n \times n}$, where $\mathbf{Z} := \{z_{ij}\}$ with z_{ij} weighting the edge between adjacency data pairs (ϕ_i, ϕ_j) , $z_{ij} = 0$ otherwise. \mathbf{Y} is diagonal with $y_{ii} := \sum_{j \neq i} z_{ij}$ for all i, j .

4.3.1 Semi-supervised LDA SparLow

For labeled dataset \mathbf{X}_l , we adopt the criterion of LDA and compute matrices \mathbf{L}^w and \mathbf{L}^b as same to Section 4.2.1. Hence the total scatter matrix can be written as

$$S_t(\Phi(\mathbf{D}, \mathbf{X}_l)) := \Phi \mathbf{L}^t \Phi^\top, \quad (33)$$

with $\mathbf{L}^t := \mathbf{L}^w + \mathbf{L}^b \in \mathbb{R}^{n_l \times n_l}$. Similar to the constructions of Section 4.2.1, we construct the formulations of semi-supervised LDA *SparLow* as

$$\begin{cases} \mathcal{A}_{\text{sda}}(\Phi) := \Phi \tilde{\mathbf{L}}^b \Phi^\top, \\ \mathcal{B}_{\text{sda}}(\Phi) := \Phi (\tilde{\mathbf{L}}^t + \alpha \mathbf{L}) \Phi^\top, \end{cases} \quad (34)$$

where $\alpha \in \mathbb{R}$ controls the influence of labeled Laplacian matrix $\tilde{\mathbf{L}}^t \in \mathbb{R}^{n_l \times n_l}$ and global Laplacian matrix $\mathbf{L} \in \mathbb{R}^{n \times n}$. Therein, $\tilde{\mathbf{L}}^b$ and $\tilde{\mathbf{L}}^t$ are the augmented matrices of \mathbf{L}^b and \mathbf{L}^t , namely,

$$\tilde{\mathbf{L}}^b = \begin{bmatrix} \mathbf{L}^b & \mathbf{0}_{n_u \times n_u} \\ \mathbf{0}_{n_u \times n_u} & \mathbf{0}_{n_u \times n_u} \end{bmatrix}, \quad \text{and} \quad \tilde{\mathbf{L}}^t = \begin{bmatrix} \mathbf{L}^t & \mathbf{0}_{n_u \times n_u} \\ \mathbf{0}_{n_u \times n_u} & \mathbf{0}_{n_u \times n_u} \end{bmatrix}.$$

4.3.2 Semi-supervised Laplacian SparLow

We now consider the semi-supervised version of Laplacian *SparLow*. For the dataset \mathbf{X} , let us define the nonlocal graph Laplacian matrix $\mathbf{L}^N = \mathbf{Y}^N - \mathbf{Z}^N$ in $\mathbb{R}^{n \times n}$ with z_{ij}^N weighting the edge between non-adjacency data pairs (ϕ_i, ϕ_j) , $z_{ij}^N = 0$ otherwise. \mathbf{Y}^N is diagonal with $y_{ii}^N := \sum_{j \neq i} z_{ij}^N$ for all i, j . For the labeled dataset \mathbf{X}_l , we adopt the settings of Section 4.2.2 and define interclass local Laplacian matrix $\mathbf{L}^- \in \mathbb{R}^{n_l \times n_l}$ and intraclass local Laplacian matrix $\mathbf{L}^+ \in \mathbb{R}^{n_l \times n_l}$. Hence, we construct the formulations of Semi-supervised Laplacian *SparLow* as

$$\begin{cases} \mathcal{A}_{\text{slap}}(\Phi) := \Phi (\tilde{\mathbf{L}}^- + \alpha_1 \mathbf{L}^N) \Phi^\top \\ \mathcal{B}_{\text{slap}}(\Phi) := \Phi (\tilde{\mathbf{L}}^+ + \alpha_2 \mathbf{L}^N) \Phi^\top, \end{cases} \quad (35)$$

with $\alpha_1 \in \mathbb{R}^+$, $\alpha_2 \in \mathbb{R}^+$ control the influence of labeled Laplacian matrix and unlabeled Laplacian matrix. Similar to the setting of Section 4.3.1, let $\tilde{\mathbf{L}}^- \in \mathbb{R}^{n_l \times n_l}$ and $\tilde{\mathbf{L}}^+ \in \mathbb{R}^{n_l \times n_l}$ denote the augmented matrices of \mathbf{L}^- and \mathbf{L}^+ .

4.3.3 Semi-supervised MVR SparLow

The supervised linear (label-based) regression (e.g. SVM) associated with a manifold regularisation [42], is another popular framework for resolving semi-supervised learning problems. In this section, we adopt an MVR model associated with a manifold regularization as

$$\begin{aligned} \min_{\mathbf{U}, \mathbf{W}} & \|\mathbf{Z}_l - \mathbf{W}^\top \mathbf{U}^\top \Phi_l\|_F^2 + \rho_1 \|\mathbf{W}\|_F^2 \\ & + \rho_2 \text{tr}(\mathbf{W}^\top \mathbf{U}^\top \Phi \mathbf{L} \Phi^\top \mathbf{U} \mathbf{W}), \end{aligned} \quad (36)$$

where $\mathbf{Z}_l \in \mathbb{R}^{d \times n_l}$ is the target matrix for \mathbf{X}_l , $\mathbf{U} \in \mathcal{G}(l, r)$, $\mathbf{W} \in \mathbb{R}^{l \times d}$ and $\rho_1, \rho_2 \in \mathbb{R}^+$. By fixing \mathbf{U} , minimization of the problem as in Eq. (36) with respect to \mathbf{W} leads to a closed-form expression

$$\mathbf{W} = (\mathbf{U}^\top (\Phi_l \Phi_l^\top + \rho_1 \mathbf{I}_r + \rho_2 \Phi \mathbf{L} \Phi^\top) \mathbf{U})^{-1} \mathbf{U}^\top \Phi_l \mathbf{Z}_l^\top. \quad (37)$$

Using this closed-form expression to substitute the \mathbf{W} in Eq. (36), we can rewrite Eq. (36) in the form of *SparLow* as

$$\begin{cases} \mathcal{A}_{\text{smvr}}(\Phi) := -\Phi_l \mathbf{Z}_l^\top \mathbf{Z}_l \Phi_l^\top \\ \mathcal{B}_{\text{smvr}}(\Phi) := \Phi_l \Phi_l^\top + \rho_1 \mathbf{I}_r + \rho_2 \Phi L \Phi^\top. \end{cases} \quad (38)$$

5 OPTIMIZATION ALGORITHM FOR *SparLow*

In this section, we firstly investigate the differentiability of the *SparLow* function, and then present a geometric conjugate gradient algorithm that maximizes the *SparLow* cost function on the underlying Riemannian manifold.

5.1 Differentiability of the *SparLow* Function

In this subsection, we investigate the differentiability of the *SparLow* function J , and derive its Euclidean gradient in the embedding space of the product manifold $\mathfrak{S}(m, r) \times \mathfrak{G}(l, r) \subset \mathbb{R}^{m \times r} \times \mathbb{R}^{r \times r}$, which is the building block for computing the Riemannian gradient of J in Section 5.2.

By the construction of the *SparLow* function being differentiable in the orthogonal projection \mathbf{P} , we can compute the Euclidean gradient of J with respect to \mathbf{P} as

$$\nabla_J(\mathbf{P}) = \frac{\mathcal{A}(\Phi(\mathbf{D}, \mathbf{X})) - f(\mathbf{D}, \mathbf{P}) \cdot \mathcal{B}(\Phi(\mathbf{D}, \mathbf{X}))}{\text{tr}(\mathbf{P}\mathcal{B}(\Phi(\mathbf{D}, \mathbf{X}))) + \sigma}. \quad (39)$$

The Euclidean gradient of J with respect to \mathbf{D} consists of three components, i.e.,

$$\nabla_J(\mathbf{D}) = \nabla_f(\mathbf{D}) - \mu_1 \nabla_{g_c}(\mathbf{D}) - \mu_2 \nabla_{g_d}(\mathbf{D}), \quad (40)$$

two of which are the Euclidean gradients of the two regularizers, which can be simply computed as

$$\nabla_{g_d}(\mathbf{D}) = \mathbf{D} - \mathbf{D}^*, \quad (41)$$

and

$$\nabla_{g_c}(\mathbf{D}) = \mathbf{D} \sum_{1 \leq i < j \leq r} \frac{2\mathbf{d}_i^\top \mathbf{d}_j}{1 - (\mathbf{d}_i^\top \mathbf{d}_j)^2} (\mathbf{e}_i \mathbf{e}_j^\top + \mathbf{e}_j \mathbf{e}_i^\top), \quad (42)$$

with $\mathbf{e}_i \in \mathbb{R}^r$ being the i -th basis vector of \mathbb{R}^r . The computation of the first component $\nabla_f(\mathbf{D})$ requires the differentiability of the sparse representation $\Phi(\mathbf{D}, \mathbf{X})$.

Given images \mathbf{x}_i and dictionary \mathbf{D} , let $\phi_i^* := [\varphi_1^*, \dots, \varphi_r^*]^\top \in \mathbb{R}^r$ be the sparse representation given by solving the sparse regression problems as in Eq. (6). We denote the set of indexes of non-zero entries of ϕ_i^* , known as the *support* of ϕ_i^* , by

$$\mathfrak{P}(\mathbf{x}_i, \mathbf{D}) := \{j \in \{1, \dots, r\} | \varphi_j^* \neq 0\}. \quad (43)$$

The differentiability of the sparse representation $\Phi(\mathbf{D}, \mathbf{X})$ requires the following assumptions.

Assumption 1. A separable regularizer g is strictly convex, and each component-wise function g_i for $i = 1, \dots, r$ is differentiable everywhere except at the origin, i.e., $g'_i(x) \neq 0$ for $x \neq 0$.

Remark 1. The strict convexity of g ensures the uniqueness of solutions of the sample-wise sparse regression, and also enables explicit characterizations of the unique global minimum. It is easy to show that most popular convex sparsifying regularizers, e.g. the elastic net regularizer, and full unnormalized Kullback-Leibler (KL) divergence, fulfill this assumption.

Assumption 2. For an arbitrary $\mathbf{D} \in \mathfrak{S}(m, r)$, the union of the supports of all sparse representations ϕ_i^* for all $i = 1, \dots, n$ is the complete set of indices of atoms, i.e.,

$$\bigcup_{i=1}^n \mathfrak{P}(\mathbf{x}_i, \mathbf{D}) = \{1, \dots, r\}. \quad (44)$$

Assumption 2 ensures all atoms in the dictionary are updated at any dictionary \mathbf{D} . Then, we can derive the following proposition about the differentiability of the sparse representation $\Phi(\mathbf{D}, \mathbf{X})$.

Proposition 1. If both Assumption 1 and 2 hold true, then the sparse representation $\Phi(\mathbf{D}, \mathbf{X})$ is differentiable on $\mathfrak{S}(m, r)$.

The proof of the proposition is given in Appendix. The proposition leads straightforwardly the following corollary about the differentiability of the *SparLow* function.

Corollary 1. If both Assumption 1 and 2 hold true, then the *SparLow* function J defined in Eq. (15) is differentiable on $\mathfrak{S}(m, r) \times \mathfrak{G}(l, r)$.

Finally, in order to develop gradient-based algorithms, we need the first derivative of $\Phi(\mathbf{D}, \mathbf{X})$ to admit closed-form expression. We refer to Appendix for more details and the proof of the following result.

Proposition 2. Let a separable regularizer g satisfy Assumption 1. If each component-wise function g_i has non-degenerate Hessian except at the origin, i.e., $g''_i(x) > 0$ for $x \neq 0$, where $g''_i(x)$ denotes the second derivative of function g_i , then the first derivative of $\Phi(\mathbf{D}, \mathbf{X})$ has a closed-form expression.

In order to compute the Euclidean gradient of f with respect to \mathbf{D} , we need to compute the first derivative of f at $\mathbf{D} \in \mathbb{R}^{m \times r}$ in direction $\Xi \in \mathbb{R}^{m \times r}$, as

$$\begin{aligned} D_1 f(\mathbf{D})\Xi = & \text{tr} \left(\frac{\mathbf{P}(\mathbf{D}\mathcal{A}(\Phi) \circ \mathbf{D}\Phi(\mathbf{D}, \mathbf{X})\Xi)}{\text{tr}(\mathbf{P}\mathcal{B}(\Phi(\mathbf{D}, \mathbf{X}))) + \sigma} \right. \\ & \left. - \frac{f(\mathbf{D}, \mathbf{P}) \cdot \mathbf{D}\mathcal{B}(\Phi) \circ \mathbf{D}\Phi(\mathbf{D}, \mathbf{X})\Xi}{\text{tr}(\mathbf{P}\mathcal{B}(\Phi(\mathbf{D}, \mathbf{X}))) + \sigma} \right), \end{aligned} \quad (45)$$

where $\mathbf{D}\mathcal{A}(\Phi): \mathbb{R}^{m \times r} \rightarrow \mathbb{R}^{r \times r}$ and $\mathbf{D}\mathcal{B}(\Phi): \mathbb{R}^{m \times r} \rightarrow \mathbb{R}^{r \times r}$ are the directional derivatives of $\mathcal{A}(\cdot)$ and $\mathcal{B}(\cdot)$, respectively. Its calculation is dependent on the concrete construction of the two matrix-valued functions \mathcal{A} and \mathcal{B} . Although there are various constructions of $\mathcal{A}(\Phi)$ and $\mathcal{B}(\Phi)$ for different learning paradigms, see Section 4, there are two basic forms, namely, $\Phi\mathbf{Z}\Phi^\top$ and $\text{tr}(\Phi\mathbf{Z}\Phi^\top)\mathbf{I}_r$ with $\mathbf{Z} \in \mathbb{R}^{n \times n}$ being some structure matrix specified in Section 4. The tedious computation of Eq. (45) can be generalized in the following form

$$D_1 f(\mathbf{D})\Xi = \text{tr} \left(\underbrace{\frac{\tilde{\mathcal{A}}(\Phi) - f(\mathbf{D}, \mathbf{P})\tilde{\mathcal{B}}(\Phi)}{\text{tr}(\mathbf{P}\mathcal{B}(\Phi(\mathbf{D}, \mathbf{X}))) + \sigma}}_{=: \mathbf{Z} \in \mathbb{R}^{n \times n}} \mathbf{D}\Phi(\mathbf{D}, \mathbf{X})\Xi \right), \quad (46)$$

where $\mathbf{D}\Phi(\mathbf{D}, \mathbf{X})\Xi \in \mathbb{R}^{r \times n}$ is the first derivative of Φ with respect to the dictionary \mathbf{D} . Here, $\tilde{\mathcal{A}}$ and $\tilde{\mathcal{B}}$ are two matrix-valued functions dependent on the specific choice of $\mathcal{A}(\cdot)$ and $\mathcal{B}(\cdot)$. We summarize the general formula for $\tilde{\mathcal{A}}$ and $\tilde{\mathcal{B}}$ for Eq. (46). When $\mathcal{A}(\Phi) = \Phi\mathbf{Z}\Phi^\top$, we have

$$\tilde{\mathcal{A}}(\Phi) = (\mathbf{Z}\Phi\mathbf{P})^\top + \mathbf{Z}\Phi^\top\mathbf{P}. \quad (47)$$

When $\mathcal{B}(\Phi) = \text{tr}(\Phi\mathbf{Z}\Phi^\top)\mathbf{I}_r$, we have

$$\tilde{\mathcal{B}}(\Phi) = \mathbf{Z}^\top\Phi^\top + \mathbf{Z}\Phi^\top. \quad (48)$$

Algorithm 1: A CG-SparLow Framework.

Input : $\mathbf{X} \in \mathbb{R}^{m \times n}$ and functions $\mathcal{A}: \mathbb{R}^{r \times n} \rightarrow \mathbb{R}^{r \times r}$ and $\mathcal{B}: \mathbb{R}^{r \times n} \rightarrow \mathbb{R}^{r \times r}$ as specified in Section 4 ;

Output: Accumulation point $(\mathbf{D}^*, \mathbf{P}^*) \in \mathfrak{S}(m, r) \times \mathfrak{G}(l, r)$;

Step 1: Given an initial guess $\mathbf{D}^{(0)} \in \mathfrak{S}(m, r)$ and $\mathbf{P}^{(0)} \in \mathfrak{G}(l, r)$;

Step 2: Set $j = j + 1$, let $(\mathbf{D}^{(j)}, \mathbf{P}^{(j)}) = (\mathbf{D}^{(j-1)}, \mathbf{P}^{(j-1)})$, and compute the Riemannian gradient

$$\mathbf{G}^{(j)} = \mathbf{H}^{(j)} = \text{grad}_J(\mathbf{D}^{(j)}, \mathbf{P}^{(j)}) ;$$

Step 3: Set $\mathbf{M}^{(j)} = (\mathbf{D}^{(j)}, \mathbf{P}^{(j)})$;

(i) Update $\mathbf{M}^{(j)} \leftarrow \Gamma_{\mathbf{M}^{(j)}, \mathbf{H}^{(j)}}(t^*)$, where

$$t^* = \underset{t \in \mathbb{R}}{\text{argmax}} J \circ \Gamma_{\mathbf{M}^{(j)}, \mathbf{H}^{(j)}}(t) ;$$

(ii) Compute $\mathbf{G}^{(j+1)} = \text{grad}_J(\mathbf{M}^{(j)})$;

(iii) Update $\mathbf{H}^{(j+1)} \leftarrow \mathbf{G}^{(j+1)} + \beta \mathcal{T}_{\mathbf{M}^{(j)}, t^* \mathbf{H}^{(j)}}(\mathbf{H}^{(j)})$, where β is chosen such that $\mathcal{T}_{\mathbf{M}^{(j)}, t^* \mathbf{H}^{(j)}}(\mathbf{G}^{(j)})$ and $\mathbf{H}^{(j+1)}$ conjugate with respect to the Hessian of J at $\mathbf{M}^{(j)}$.

Step 4: If $\|\mathbf{M}^{(j+1)} - \mathbf{M}^{(j)}\|$ is small enough, stop. Otherwise, go to Step 2 ;

Let \mathfrak{P}_i be a shorthand notation for $\mathfrak{P}(\mathbf{x}_i, \mathbf{D})$ and $k_i := |\mathfrak{P}_i|$ denote the cardinality of \mathfrak{P}_i . We denote further $\phi_{\mathfrak{P}_i} := \{\varphi_{i,j}\} \in \mathbb{R}^{k_i}$ with $j \in \mathfrak{P}_i$ and $\mathbf{D}_{\mathfrak{P}_i} \in \mathbb{R}^{m \times k_i}$ being the subset of \mathbf{D} , in which the index of atoms (columns) fall into the support \mathfrak{P}_i . Finally, by recalling the first derivative of the generic sparse coding ϕ as computed in Eq. (12) in Appendix, we compute the Euclidean gradient as

$$\nabla_f(\mathbf{D}) = \sum_{i=1}^n \mathcal{V}(\mathbf{x}_i \mathbf{Z}_{\mathfrak{P}_i} K_i^{-1} - \mathbf{D}_{\mathfrak{P}_i} \phi_{\mathfrak{P}_i} \mathbf{Z}_{\mathfrak{P}_i} K_i^{-1} - \mathbf{D}_{\mathfrak{P}_i} K_i^{-1} \mathbf{Z}_{\mathfrak{P}_i}^\top \phi_{\mathfrak{P}_i}), \quad (49)$$

where $K_i := \mathbf{D}_{\mathfrak{P}_i}^\top \mathbf{D}_{\mathfrak{P}_i} + \text{Hg}(\phi_{\mathfrak{P}_i})$ with $\text{Hg}(\phi_{\mathfrak{P}_i})$ being the Hessian matrix of the regularizer g defined on $\phi_{\mathfrak{P}_i}$. Here, $\mathcal{V}: \mathbb{R}^{m \times k_i} \rightarrow \mathbb{R}^{m \times r}$ produces a matrix by replacing columns of the zero matrix in $\mathbb{R}^{m \times r}$ with columns of matrix $\mathbf{Y} \in \mathbb{R}^{m \times k_i}$ according to the support \mathfrak{P}_i .

5.2 A Geometric CG SparLow Algorithm

In this subsection, we present a geometric CG algorithm on the product manifold $\mathfrak{M} := \mathfrak{S}(m, r) \times \mathfrak{G}(l, r)$ to maximize the SparLow function J as defined in Eq. (15). It is well known that CG algorithms offer prominent properties, such as a superlinear rate of convergence and the applicability to large scale optimization problems with low computational complexity, e.g. in sparse recovery [12]. We refer to [12], [43] for further technical details for these computations.

Classic geometric CG algorithms require the concepts of geodesic and parallel transport, which are often more computationally demanding. In this work, we adopt an alternative approach based on the concept of retraction and its corresponding vector transport. A generic framework of our CG-SparLow algorithm is

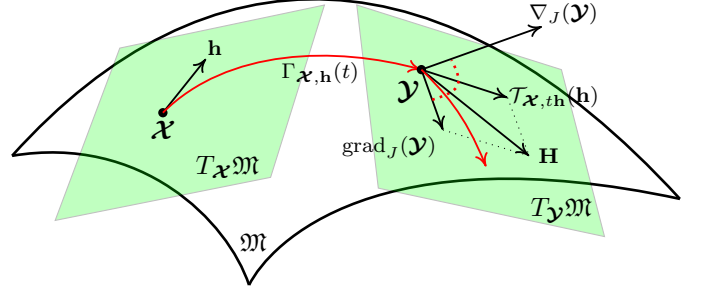


Fig. 1. Some concepts of the Geometric CG algorithm can be visualized in this figure. At first, it depicts two nearby points \mathbf{X} and \mathbf{Y} on a manifold \mathfrak{M} together with their tangent spaces $T_{\mathbf{X}}\mathfrak{M}$ and $T_{\mathbf{Y}}\mathfrak{M}$ (green areas). Secondly, it shows a CG update from the point \mathbf{X} to the point \mathbf{Y} in a search direction \mathbf{h} along the curve $\Gamma_{\mathbf{X},\mathbf{h}}(t)$. The geodesic $\Gamma_{\mathbf{X},\mathbf{h}}(t)$ in the direction of \mathbf{h} connects the two points \mathbf{X} and \mathbf{Y} . Thirdly, it introduces the gradient at \mathbf{Y} , i.e., the Euclidean gradient $\nabla_J(\mathbf{Y})$ and its projection onto the tangent space $\text{grad}_J(\mathbf{Y}) \in T_{\mathbf{Y}}\mathfrak{M}$, namely, Riemannian gradient. Finally, as the new CG search direction at \mathbf{Y} , \mathbf{H} is computed via vector transport $\mathcal{T}_{\mathbf{X},\mathbf{h}}(\mathbf{h})$ and $\text{grad}_J(\mathbf{Y})$.

summarized in Algorithm 1, and further illustrated in Fig. 1. In the rest of this section, we explain the key technical details of the CG-SparLow algorithm.

Firstly, we recall some basic geometry of the product manifold \mathfrak{M} . We denote the tangent space of \mathfrak{M} at $\mathbf{M} := (\mathbf{D}, \mathbf{P})$ by

$$T_{\mathbf{M}}\mathfrak{M} := T_{\mathbf{D}}\mathfrak{S}(m, r) \times T_{\mathbf{P}}\mathfrak{G}(l, r), \quad (50)$$

where $T_{\mathbf{D}}\mathfrak{S}(m, r)$ and $T_{\mathbf{P}}\mathfrak{G}(l, r)$ denote the tangent space of the product of spheres $\mathfrak{S}(m, r)$ and the Grassmann manifold $\mathfrak{G}(l, r)$, respectively. We endow the manifold \mathfrak{M} with the Riemannian metric inherited from the surrounding Euclidean space, i.e.,

$$\langle (\mathbf{D}_1, \mathbf{P}_1), (\mathbf{D}_2, \mathbf{P}_2) \rangle_R := \text{tr}(\mathbf{D}_1 \mathbf{D}_2^\top) + \text{tr}(\mathbf{P}_1 \mathbf{P}_2^\top), \quad (51)$$

with $\mathbf{D}_1 \in \mathbb{R}^{m \times r}$ and $\mathbf{P}_1 \in \mathbb{R}^{r \times r}$.

In each sweep of the CG algorithm (from Step 2 to Step 4 in Algorithm 1), after computing the Euclidean gradient of J as computed in Eq. (39) and (40), we compute the Riemannian gradient of J with respect to the Riemannian metric as defined in Eq. (51) via the associated projection on the tangent space $T_{\mathbf{M}}\mathfrak{M}$. Concretely, we have

$$\text{grad}_J(\mathbf{D}, \mathbf{P}) := (\text{grad}_1 J(\mathbf{D}, \mathbf{P}), \text{grad}_2 J(\mathbf{D}, \mathbf{P})), \quad (52)$$

with $\text{grad}_1 J(\mathbf{D}, \mathbf{P})$ and $\text{grad}_2 J(\mathbf{D}, \mathbf{P})$ being the Riemannian gradients of J with respect to the first and the second parameter, respectively. Specifically, we have

$$\text{grad}_1 J(\mathbf{D}, \mathbf{P}) := \nabla_J(\mathbf{D}) - \mathbf{D} \text{ddiag}(\mathbf{D}^\top \nabla_J(\mathbf{D})) \quad (53)$$

with $\text{ddiag}(\cdot)$ putting the diagonal entries of a square matrix into a diagonal matrix form, and

$$\text{grad}_2 J(\mathbf{D}, \mathbf{P}) := \mathbf{P} \nabla_J(\mathbf{P}) + \nabla_J(\mathbf{P}) \mathbf{P} - 2\mathbf{P}(\nabla_J(\mathbf{P}))\mathbf{P}. \quad (54)$$

A retraction $\Gamma_{\mathbf{M}}: T_{\mathbf{M}}\mathfrak{M} \rightarrow \mathfrak{M}$ is a smooth mapping from the tangent space $T_{\mathbf{M}}\mathfrak{M}$ to the manifold, such that the evaluation $\Gamma_{\mathbf{M}}(0) = \mathbf{M}$ and the derivative $D\Gamma_{\mathbf{M}}(0): T_{\mathbf{M}}\mathfrak{M} \rightarrow T_{\mathbf{M}}\mathfrak{M}$ is the identity mapping. For the unit spheres, we restrict ourselves to the following retraction, for $\mathbf{d} \in \mathfrak{S}(m, 1)$ and $\xi \in T_{\mathbf{d}}\mathfrak{S}(m, 1)$

$$\gamma_{\mathbf{d},\xi}(t) := \frac{\mathbf{d} + t\xi}{\|\mathbf{d} + t\xi\|_2} \in \mathfrak{S}(m, 1). \quad (55)$$

For constructing a retraction on the Grassmann manifold $\mathfrak{Gr}(l, m)$, we need a map with $\mathbf{P} \in \mathfrak{Gr}(l, m)$ and $\Psi \in T_{\mathbf{P}}\mathfrak{Gr}(l, m)$

$$\zeta_{\mathbf{P}, \Psi}(t) := (\mathbf{I}_r + t(\Psi\mathbf{P} - \mathbf{P}\Psi))_Q, \quad (56)$$

where $t > 0$ is the step size, and $(\cdot)_Q$ is the unique QR decomposition of an invertible matrix, i.e., all diagonal entries of the upper triangular part are positive. Then we define the following retraction on $\mathfrak{Gr}(l, m)$ as

$$\gamma_{\mathbf{P}, \Psi}(t) := \zeta_{\mathbf{P}, \Psi}(t)\mathbf{P}(\zeta_{\mathbf{P}, \Psi}(t))^\top \in \mathfrak{Gr}(l, m). \quad (57)$$

In concatenation, we construct a retraction on $\mathfrak{M} := \mathfrak{S}(m, r) \times \mathfrak{Gr}(l, r)$ with $\mathbf{M} := (\mathbf{D}, \mathbf{P}) \in \mathfrak{M}$ and $\mathbf{H} := (\Xi, \Psi) \in T_{\mathbf{M}}\mathfrak{M}$ as

$$\Gamma_{\mathbf{M}, \mathbf{H}}(t) := ([\gamma_{\mathbf{d}_i, \xi_i}(t)]_{i=1, \dots, k}, \gamma_{\mathbf{P}, \Psi}(t)), \quad (58)$$

which is used for implementing a line search algorithm on \mathfrak{M} in Step 3-(i), see [12]. Finally, by recalling the vector transport on $\mathfrak{S}(m, 1)$ with respect to the retraction $\gamma_{\mathbf{d}, \xi}(t)$ in Eq. (55) as

$$\tau_{\mathbf{d}, t\xi}(\tilde{\xi}) := \frac{1}{\|\mathbf{d} + t\xi\|_2} \left(\mathbf{I}_n + \frac{(\mathbf{d} + t\xi)(\mathbf{d} + t\xi)^\top}{\|\mathbf{d} + t\xi\|_2^2} \right) \tilde{\xi}, \quad (59)$$

and the vector transport on $\mathfrak{Gr}(l, m)$ with respect to the retraction $\gamma_{\mathbf{P}, \Psi}(t)$ in Eq. (57) as

$$\tau_{\mathbf{P}, t\Psi}(\tilde{\Psi}) := \zeta_{\mathbf{P}, \Psi}(t)\tilde{\Psi}(\zeta_{\mathbf{P}, \Psi}(t))^\top, \quad (60)$$

we define the vector transport of $\tilde{\mathbf{H}} := (\tilde{\Xi}, \tilde{\Psi}) \in T_{(\mathbf{D}, \mathbf{P})}\mathfrak{M}$ with respect to the retraction $\Gamma_{\mathbf{M}, \mathbf{H}}(t)$ in the direction $t\mathbf{H} = (\Xi, \Psi) \in T_{(\mathbf{D}, \mathbf{P})}\mathfrak{M}$, denoted by $\mathcal{T}_{\mathbf{M}, \mathbf{H}}: T_{\mathbf{M}}\mathfrak{M} \rightarrow T_{\Gamma_{\mathbf{M}, \mathbf{H}}(t)}\mathfrak{M}$, as

$$\mathcal{T}_{\mathbf{M}, t\mathbf{H}}(\tilde{\mathbf{H}}) := ([\tau_{\mathbf{d}_i, t\xi_i}(\tilde{\xi}_i)]_{i=1, \dots, k}, \tau_{\mathbf{P}, t\Psi}(\tilde{\Psi})). \quad (61)$$

For updating the direction parameter β in Step 3-(iii), we employ a formula proposed in [44]

$$\beta^{KH} = \frac{\langle \mathbf{G}^{(j+1)}, \mathbf{G}^{(j+1)} - \mathcal{T}_{\mathbf{M}^{(j)}, t\mathbf{H}^{(j)}}\mathbf{G}^{(j)} \rangle_R}{\langle \mathbf{H}^{(j)}, \mathbf{G}^{(j)} \rangle_R}. \quad (62)$$

6 EXPERIMENTAL EVALUATIONS

In this section, we investigate performance of our proposed *SparLow* framework in several image processing applications. For the convenience of referencing, we adopt the following fashion to name the algorithms in comparison: for example, the PCA-like *SparLow* algorithm described in Section 4.1.1 is referred to as the *PCA-SparLow* and its sequential learning counterpart, which directly applies a PCA on the corresponding sparse representations, as *SparPCA*.

6.1 Experimental Settings

In unsupervised learning experiments, we employ the K-SVD algorithm [10] to compute an empirically optimal data-driven dictionary, then initialize *SparLow* algorithms with its column-wise normalized copy, as required by the regularizer g_d . For both supervised and semi-supervised learning, we adopt the same approach to generate a sub-dictionary for each class, and then concatenate all sub-dictionaries to form a common dictionary. In order to compare the performance of the *SparLow* system with different initial dictionaries, Fig. 3 depicts the optimization process of *LDA-SparLow* performed on the 15-Scenes dataset, with the data-driven dictionary being learned by K-SVD, ODL

[45] and RLS-DLR [46], respectively. Furthermore, Fig. 6 depicts the optimization process of supervised *SparLow* methods with the data-driven dictionary being learned by K-SVD.

With the initial dictionary $\mathbf{D}^{(0)} \in \mathfrak{S}(m, r)$ being given, the initial orthogonal projection $\mathbf{P}^{(0)} \in \mathfrak{Gr}(l, r)$ can be directly obtained by applying classic TQ maximization algorithms on the sparse representations of the samples with respect to $\mathbf{D}^{(0)}$. Certainly, when the number of training samples is huge, it is unnecessary to perform a TQ maximization in order to generate an initialization. Instead, we employ only a selection of random samples to compute the initial orthogonal projection $\mathbf{P}^{(0)}$.

In all experiments, we choose $\sigma = 10^{-3}$ by hand in Eq. (12). The parameters for μ_1, μ_2, r, l in Eq. (15) and λ_1, λ_2 in Eq. (7) or λ in *KL-divergence* [37] could be well tuned via performing cross validation. Images are presented as m -dimensional vectors, and normalized to have unit norm. For datasets without a pre-construction of training set and testing set, all experiments are repeated ten times with different randomly constructed training set and test set, and the average of per-class recognition rates is recorded for each run. For most of our experiments, we employ the *elastic net* method [36] to solve the sparse coding problem (6). An alternative solution based on *KL-divergence* is also evaluated in the application of large scale image processing in Section 6.4.

Fig. 4 plots the recognition rates of *LDA-SparLow*, *MFA-SparLow*, SRC, FDDL [19], and LC-KSVD with varying dictionary sizes (number of atoms). In all cases, the proposed methods perform better than SRC and FDDL, and give significant improvement to LC-KSVD and TDDL. This also confirms that an increasing dimension of sparse representation can enhance linear separability for image classification, as observed in [29], [30].

6.2 Tuning of Parameters

Training a *SparLow* model can be computationally expensive. We firstly investigate the impact of various factors of the learning model. All experiments in this subsection were conducted on the USPS dataset [47], which contains 7291 training images and 2007 testing images. After applying *SparLow* models on the images to produce the corresponding low dimensional representations, we employ the one-nearest neighbor (1NN) method to test the performance of the *SparLow* in terms of classification.

6.2.1 Weighing the Regularisers

Here, we investigate the impact of the two regularizers g_c and g_d on the performance of the *SparLow*, i.e., the inference of weighing parameters μ_1 and μ_2 in Eq. (15). Firstly, we test a special case that $\mu_2 = 0$, i.e., without the *data regularizer*. Fig. 2(a) shows the box plot of results of applying the 1NN classification ten times on the USPS database with random initializations. As usual, the recognition accuracy is chosen as the lowest recognition result of the algorithm after 20 iterations, i.e., the converged value of recognition in this work. The results suggest that the regularizer g_d has the capability of ensuring good reconstruction, and achieving stable discriminations. Fig. 6 depicts the trace of performance over optimization process of supervised *SparLow* on CMU PIE faces with $n_{\text{train}} = 120$ and USPS, respectively. All sub-figures in Fig. 6 and Fig. 2(a) show that the regularizers g_c and g_d can highly improve the stability of recognition accuracy after convergence.

We further investigate the influence of different weighing factors μ_1 and μ_2 for both the *PCA-SparLow* and *LDA-SparLow* on the 1NN classification problem. The experiments are performed

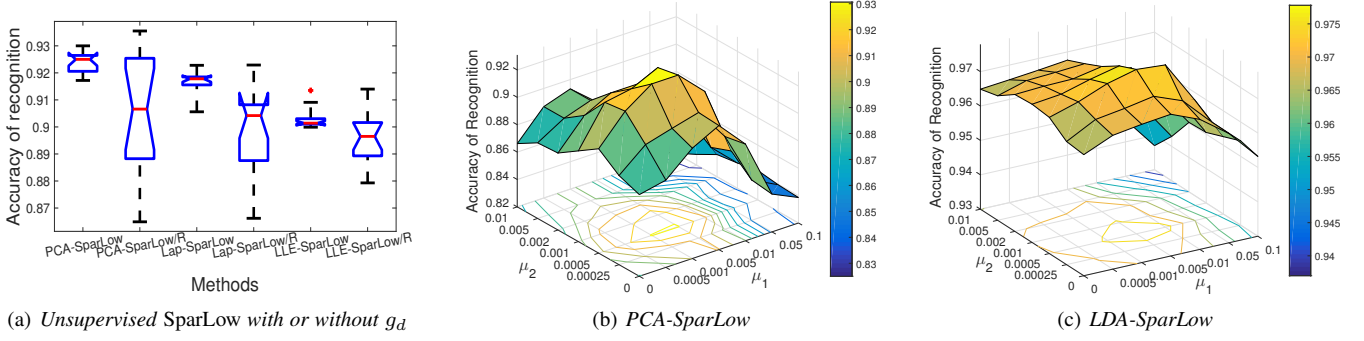


Fig. 2. Impact of the regularizers to the recognition rate on the USPS digits (*SparLow/R* refers to *PCA-SparLow* methods without g_d).

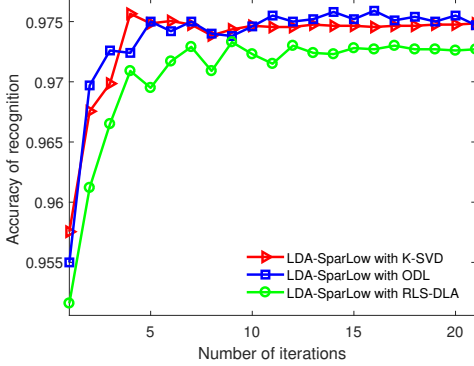


Fig. 3. Trace of performance over optimization process initialized with different sparse coding methods on 15-Scenes dataset.

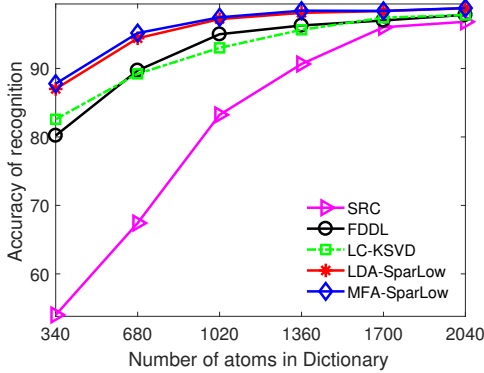


Fig. 4. Comparison on recognition results with different dictionary sizes for PIE faces. The classifier is 1NN.

with $r = 1000$, and $l = 50$ for *PCA-SparLow*, $l = 9$ for *LDA-SparLow*. Fig. 2(b) and Fig. 2(c) depict the 1NN classification results of *PCA-SparLow* and *LDA-SparLow* with respect to different weighing factors μ_1 and μ_2 . It is clear from Fig. 2(b) and Fig. 2(c) that for both *PCA-SparLow* and *LDA-SparLow*, suitable choices of μ_1 and μ_2 can improve the performance of the *SparLow* system.

6.2.2 Targeted Low Dimensionality

The main goal of this work is to learn appropriate low dimensional representations of images. In this experiment, we investigate the impact of the choice of the targeted low dimensionality l to the performance of *SparLow*. As depicted in Fig. 5(a), *SparLow* models of three classic unsupervised learning algorithms are examined in terms of recognition accuracy with respect to different targeted low dimensionality l . It is clear that, when $l \geq 32$, all three tested unsupervised *SparLow* methods perform almost equally

well for this specific task. Similar trends are also observed in the supervised *SparLow*'s as in Fig. 5(b). Hence, we conclude that, after bypassing a threshold of the targeted low dimensionality, the performance of the corresponding *SparLow* method is stable and reliable.

6.2.3 Number of Labelled Samples

For semi-supervised learning, one important aspect is surely the number of labelled samples. We compare the *SparLow* counterparts of three state of the art semi-supervised learning algorithms, i.e., *SDA-SparLow* versus SDA [33], *SLap-SparLow* versus SDE [40], *SMVR-SparLow* versus label propagation methods, e.g. LapRLS [42] and LGC [41].

More specifically, the dictionary $\hat{\mathbf{D}}$ is initialized by Laplacian *SparLow*. For label propagation methods, e.g. LapRLS and LGC, we use the same settings as in [41], [42] for classification. We choose $\mu_1 = 2.5 \times 10^{-4}$, $\mu_2 = 5 \times 10^{-3}$, $\lambda_1 = 0.2$, $\lambda_2 = 10^{-3}$ for all tests. Moreover, we set $\alpha = 0.1$ for *SDA-SparLow* and *SMVR-SparLow*, $\alpha_1 = 0.1$, $\alpha_2 = 0.01$ for *SMFA-SparLow*. The neighborhood size is set to 20. Our results in Fig. 5(c) show that the three semi-supervised *SparLow* methods consistently outperform all other state of the art methods. It is also worth noticing that with an increasing number of labeled samples, semi-supervised *SparLow* methods demonstrate greater advantages over their conventional counterparts.

6.3 Evaluation of Disentanglability

In this subsection, we investigate the ability of unsupervised *SparLow* to disentangle the semantics from observations, which we refer to as *disentanglability* in this section. Arguably, unsupervised disentangling, i.e., automatically disentangling underlying discriminant factors within the data, is one most challenging scenario to test the performance of representation learning. Hereby, the disentangled factors of variation could be discerned consistently across a set of images, such as the class information, various levels of illuminations, resolutions, sharpness, camera orientations, the expressions and the poses of faces, etc. In the following, we take the disentangling factors of the class information, the illuminations, the expressions and the poses of faces as examples to evaluate the disentanglability of the proposed *SparLow* framework.

Our first experiments are performed on the CMU-PIE face dataset [33], which consists of 68 human subjects with 41,368 face images in total. In our experiments, we set $\lambda_1 = 10^{-2}$, $\lambda_2 = 10^{-5}$, $\mu_1 = 2.5 \times 10^{-4}$, $\mu_2 = 5 \times 10^{-3}$. It is often considered to possess strong class information, compared to other classic recognition benchmark datasets. We follow the same experimental

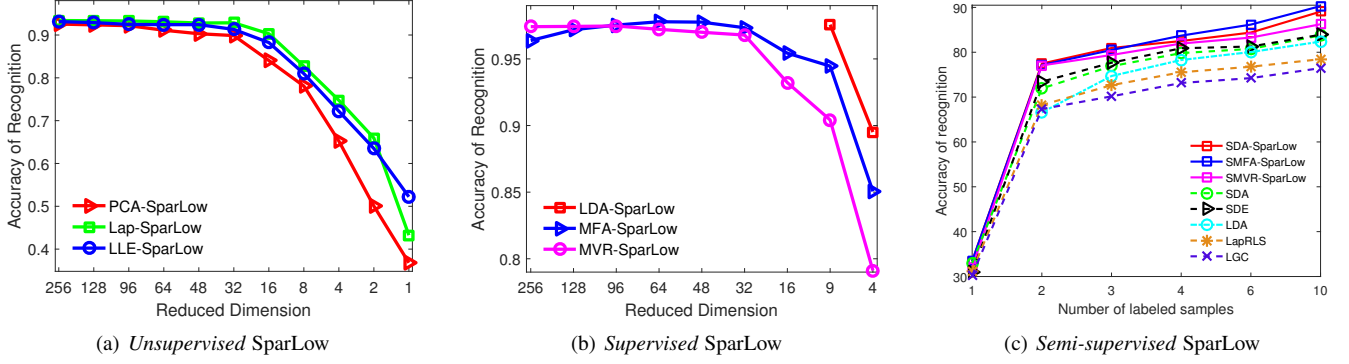


Fig. 5. Impact of targeted low dimensionality and number of labelled samples to the recognition rate of 1NN classification on the USPS digits.

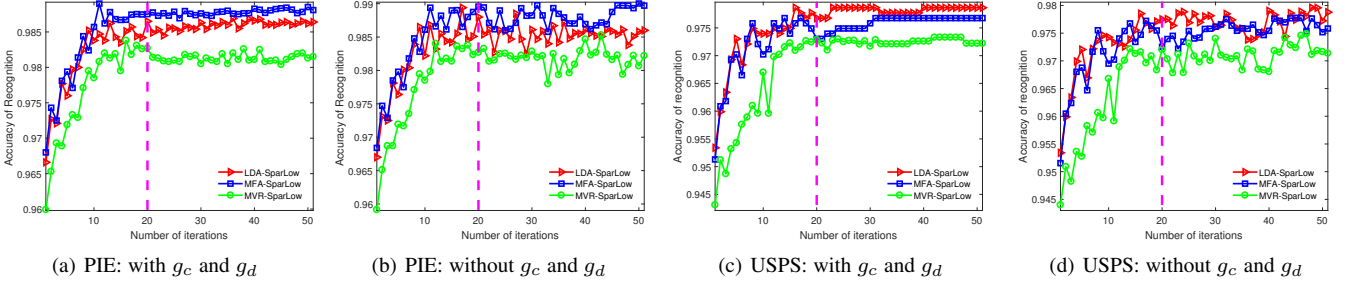


Fig. 6. Trace of performance over optimization process of supervised *SparLow* with or without regularizers.

setting as described in [33] and choose the frontal pose and use all the images under different illuminations, so that we get 64 images for each subject with the resized scale 32×32 . We compare three classic unsupervised learning methods, namely, PCA, ONPP, and OLPP, with its sequential sparsity based methods, and our proposed *SparLow* methods. Fig. 7 reports the performance of the three families of methods, applied to 1NN classification problems on the CMU-PIE dataset. It is clear that the *SparLow* methods outperform by far the state of the art algorithms consistently.

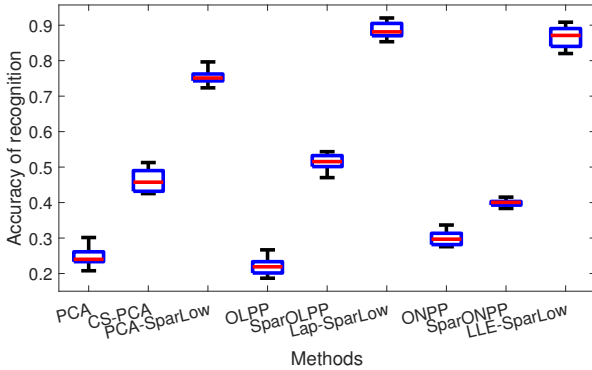


Fig. 7. Face recognition on 68 class PIE faces. The classifier is 1NN. Randomly choose 8160 training samples and 3394 testing samples.

A popular intuitive approach to evaluate disentanglability of representation learning methods is via a visualization of the extracted representations. Similar to the concepts of *Fisherfaces* in [5], *eigenfaces* in [5], *laplacianfaces* in [4], *orthogonal laplacianfaces* in [32], and *orthogonal LLEfaces* in [2], we construct the j^{th} *SparLow* facial disentanglement \mathbf{v}_j as

$$\mathbf{v}_j = \mathbf{D}\mathbf{u}_j \in \mathbb{R}^m, \quad (63)$$

with \mathbf{u}_j being the j^{th} column vector of projection matrix \mathbf{U} . Fig. 8(a) shows the first ten *eigenfaces*, *laplacianfaces*, and *LLE-*

faces from top to bottom, while Fig. 8(b) depicts the first ten basis vectors of learned disentangling factors of variation for *PCA-SparLow*, *Lap-SparLow* and *LLE-SparLow*, accordingly. Clearly, our learned facial features in Fig. 8(b) capture more factors of variation in faces, such as varying poses and expressions (e.g. smile), than the state of the arts in Fig. 8(a).

Furthermore, we aim to intuitively illustrate the disentangled factor by visualizing the low dimensional image representations, i.e., 3D. As suggested in [33], a subset containing 11,554 PIE faces with five near frontal poses (C05, C07, C09, C27, C29) and different illuminations are chosen, thus we nearly get 170 images for each individual. Our experiments of 3D visualization were conducted on PIE faces (class 5, 35, 65), compared to their classic counterparts. As depicted in Fig. 9, the 3D representations captured in the original data space, shown in the first row in Fig. 9, are hardly possible to cluster or group. In particular, the boundary between each pair of faces are completely entangled. It is evidential that visualization powered by the *SparLow*, i.e., the third row in Fig. 9, leads to direct clustering of the faces. In short, the class information is clearly disentangled, while the other approaches fail.

Then we perform 3D visualization on PIE faces without class information. We choose 70 faces from the class 5 with 2 factors of variations, i.e., poses and illuminations. As can be seen from the Fig. 10, the information referred to poses and illuminations could be clearly disentangled. In Fig. 10(a), from left to right, the illumination become stronger. From top to bottom, the poses of faces change from left to right. The similar results are also shown in Fig. 10(b) and Fig. 10(c).

Our last experiments in this subsection are performed on handwritten digits, i.e., the MNIST dataset¹ and the USPS dataset. The MNIST dataset consists of 60000 handwritten digits images

1. <http://yann.lecun.com/exdb/mnist/>

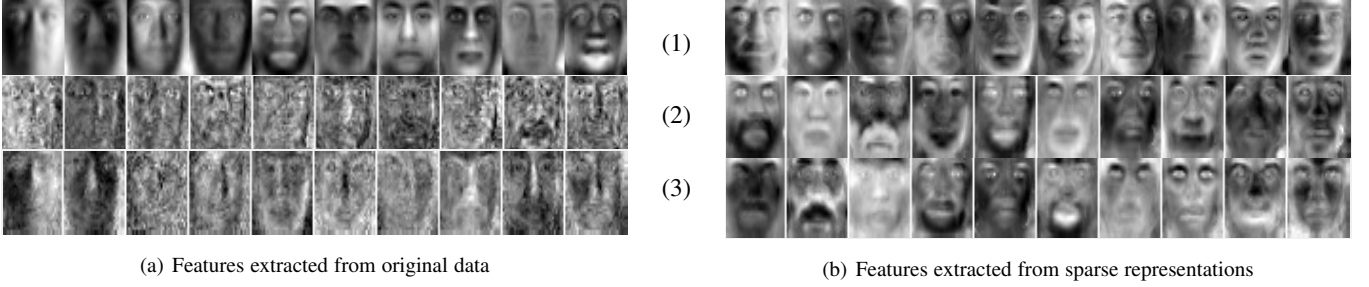


Fig. 8. Visualisation of facial features on PIE faces [33]. The presented features are generated via Eq. (63). From top to bottom: (1) PCA eigenfaces; (2) Laplacianfaces; (3) LLEfaces.

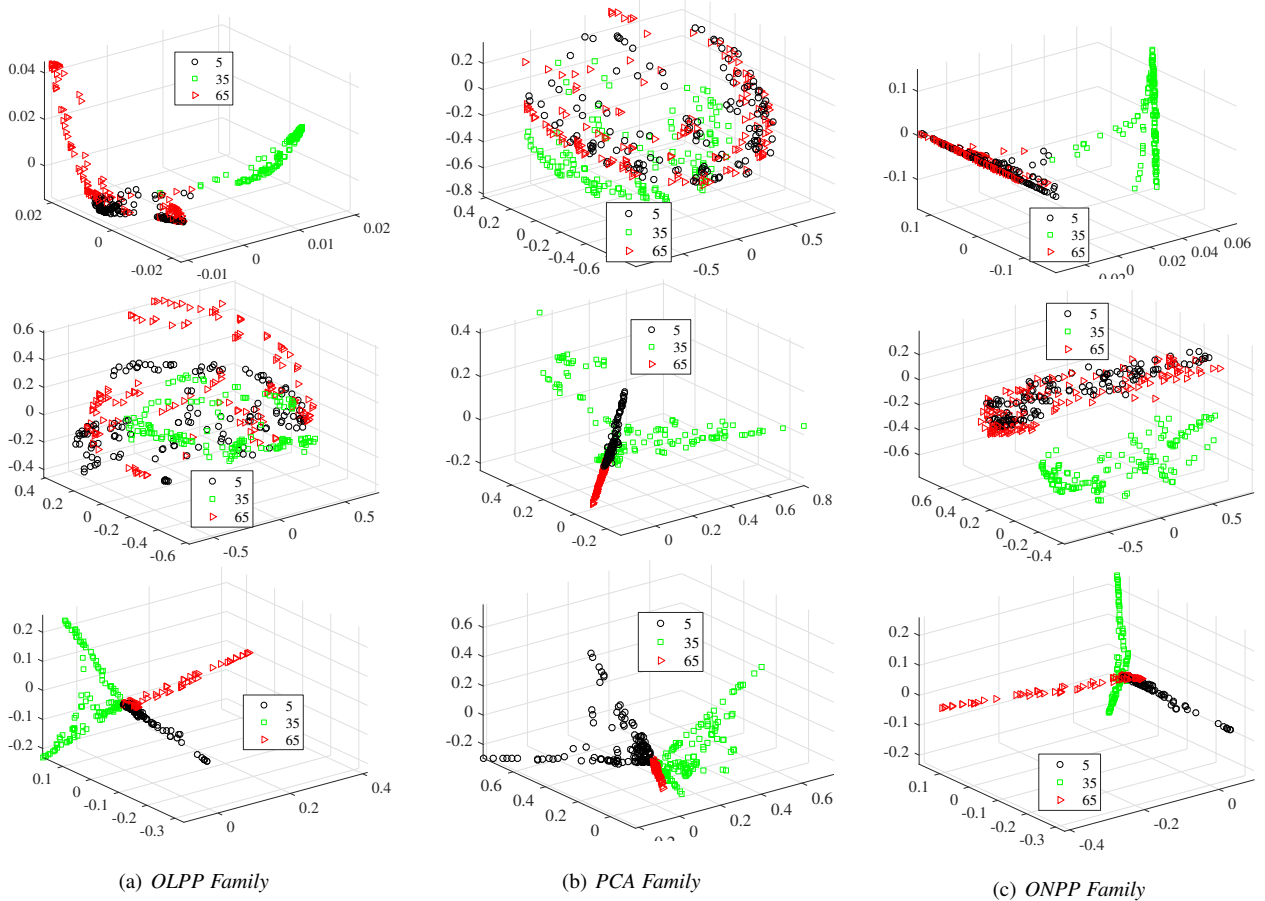


Fig. 9. 3D visualisation of PIE faces (class 5, 35, 65). From top to bottom: Applying OLPP/PCA/ONPP in original space, in sparse space with respect to initial dictionary \hat{D} , and in sparse space with respect to learned dictionary via *SparLow*, respectively.

for training and 10000 digits images for testing. The parameters for elastic net are set to be $\lambda_1 = 0.2$, $\lambda_2 = 2 \times 10^{-5}$, and $\mu_1 = 5 \times 10^{-3}$, $\mu_2 = 4 \times 10^{-4}$, for both experiments on the MNIST and USPS datasets. We compare the *SparLow* methods to several state of the art methods, on the task of 1NN and Gaussian SVM (GSVM) classification. For PCA, KPCA and *PCA-SparLow*, we set $l = 50$, for other methods, we set $l = 20$. For USPS, we use the full training and testing dataset. For MNIST, we randomly choose 30000 images for training, and use standard 10000 testing dataset. Our results in Table 1 suggest that the *SparLow* methods consistently outperform the state of the arts.

6.4 Performance in Large Scale Image Processing

Finally, we investigate performance of the *SparLow* model in large scale image processing applications, specifically, the problem of object categorization on large scale image dataset with complex

backgrounds, such as the images from Caltech-101 [49], Caltech-256 [50] and 15-Scenes [51].

We adopt a popular approach of object categorization to firstly detect certain local image features, such as dense SIFT or dense DHOG (a fast SIFT implementation) [52], then to quantize them into discrete “visual words” over a codebook, and finally to compute a fixed-length Spatial Pyramid Pooling (SPP) vector of acquired “visual words” [16], [17], [51]. We refer to such an approach as the *SIFT/DHOG-SPP* representation. In our experiments, the local descriptor is extracted from $s \times s$ pixel patches densely sampled from each image, specifically, we choose $s = 16$ for SIFT and $s = 16, 25, 31$ for DHOG. The dimension of each SIFT/DHOG descriptor is 128. A codebook with the size of $k = 1024$ or $k = 2048$, is learned for coding SIFT/DHOG descriptors. We then divide the image into 4×4 , 3×3 and 1×1 subregions, i.e., 21 bins. The spatial pooling procedure for each spatial sub-region is applied via the max pooling func-

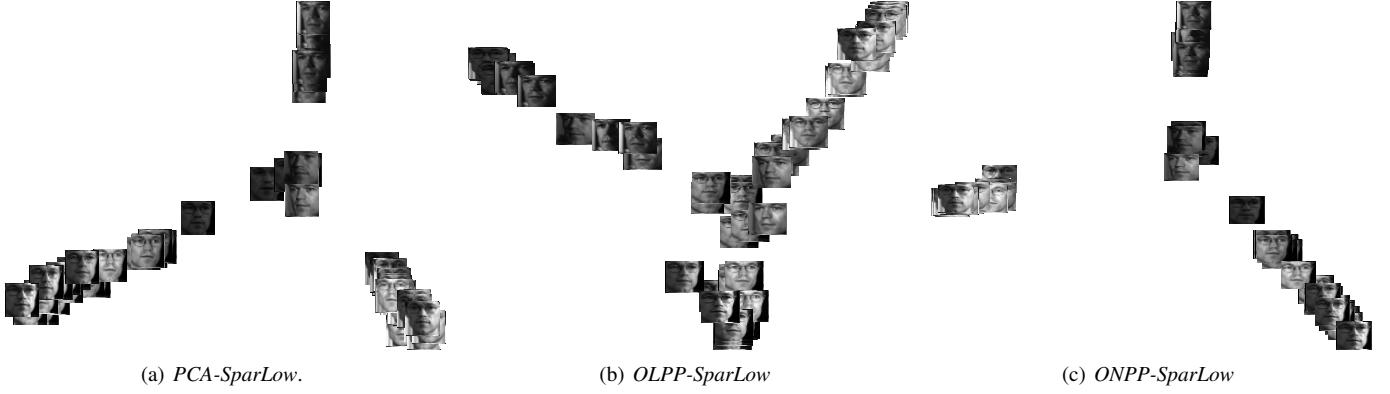


Fig. 10. 2D visualization of PIE faces (class 5).

Methods	USPS (1NN)	USPS (GSVM)	MNIST (1NN)	MNIST (GSVM)
PCA [27]	86.40, $l = 50$	92.43, $l = 50$	84.62, $l = 50$	94.63, $l = 50$
Sparse PCA [48]	87.66	88.91	86.10	90.88
OLPP [32]	84.11	91.48	83.12	94.76
ONPP [31]	87.39	92.73	85.01	95.21
KPCA [31]	89.19, $l = 50$	93.27, $l = 50$	—	—
LLE [2]	68.81	90.43	66.09	93.11
LE [31]	71.85	91.93	68.16	93.90
ISOMAP [31]	64.80	90.13	60.51	91.67
CS-PCA [27]	87.84	94.22	87.65	96.04
<i>PCA-SparLow</i>	92.18, $l = 50$	96.82, $l = 50$	91.23, $l = 50$	97.12
<i>Lap-SparLow</i>	91.83	96.26	89.32	96.91
<i>LLE-SparLow</i>	90.78	96.16	89.10	96.93

TABLE 1

Classification Performance (Accuracy (%)) for the MNIST & USPS datasets of the Proposed *SparLow* methods, with comparisons to some classical unsupervised approaches.

tion associated with an “ ℓ_2 normalisation”, e.g. [16], [17], [22], [25]. The final SPP representations are computed with the size $m = 21504$ or $m = 43008$, and hence are reduced into a low-dimensional PCA-projected subspace. In what follows, we denote by m, m_{PCA}, r, l the dimension of SPP representation, PCA projected subspace, sparse codes, and learned low dimensional representation, respectively.

6.4.1 Number of Reduced Features

It is known that the computational complexity of sparse coding mainly depends on the choice of dictionary size [35]. It is hence necessary to investigate the impact of the number of features to the performance of *SparLow*. We employ a popular approach to firstly apply a classic PCA transformation on the SPP features, and then learn a dictionary on reduced features [13]. In this experiment, we deploy the Caltech-101 dataset [49], which contains 9144 images from 102 classes. Most images are in medium resolution (about 300×300 pixels). Fig. 11 show the recognition results of several supervised *SparLow* and semi-supervised *SparLow* methods with PCA projected SPP features. It is obvious that after reaching a certain number of reduced features, i.e., $l > 1024$, all methods in test show no significant improvement. Moreover, it is worth knowing that the performance of two semi-supervised *SparLow* methods consistently outperform other methods.

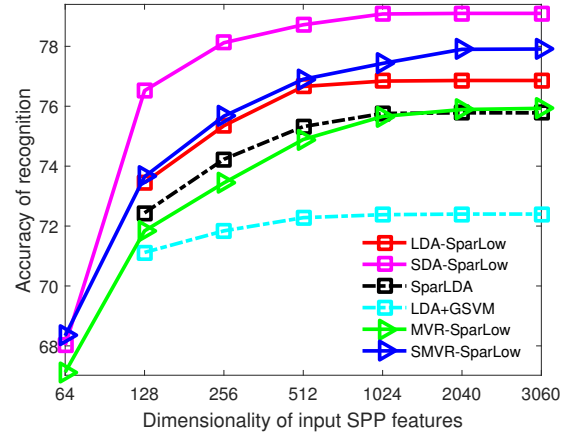


Fig. 11. Recognition results using proposed *MFA-SparLow* in PCA projected subspace on Caltech-101 dataset. $n_{\text{train}} = 30$, $r = 3060$.

6.4.2 Caltech-101 dataset

Learning on Caltech-101 dataset is often considered to be hard, since the number of images per category varies significantly from 31 to 800. In our experiments, we set $\lambda_1 = 5 \times 10^{-2}$, $\lambda_2 = 10^{-5}$, $k = 1024$, $r = 1020$, and choose $l \in \{101, 287, 512\}$ are set for LDA, MFA, MVR related methods, respectively. We confine ourselves to the same experimental settings as used in [16], [17], [51]. We randomly select 1, 5, 10, 15, 20 and 30 labeled images per category for training and the rest images for testing. For semi-supervised *SparLow*, the training set includes all labeled and unlabelled images. Table 2 (a) gives a comparison of *LDA-SparLow*, *SLDA-SparLow* with approaches from the literatures. Note that, for the number of labeled samples $n_l = 1$, LDA and *LDA-SparLow* are not applicable. It shows that our proposed approaches consistently outperform all the competing approaches. Especially, the semi-supervised *SparLow* could significantly improve the recognition accuracy when the labeled training samples are limit. The possible reason is that some categories have large samples, e.g. the category *airplanes* has 800 samples, the $n_l \leq 30$ is too much limit for training such a category. Semi-supervised *SparLow*'s take advantage of all the data for training sparsifying dictionary, which is the key factor to promote the discrimination of sparse representations.

6.4.3 Caltech-256 dataset

The Caltech-256 dataset consists of 30607 images from 256 categories with various resolutions from 113×150 to 960×1280 .

TABLE 2
Classification Performance (Average accuracy (%)) on Caltech-101 & Caltech-256 datasets.

(a) Caltech-101	1	5	10	15	20	25	30	(b) Caltech-256	15	30	45	60
KSPM [51]	—	—	—	56.40	—	—	64.40	KSPM [51]	—	23.34	—	—
ScSPM+SVM [16]	—	—	—	67.0	—	—	73.2	ScSPM+SVM [16]	27.73	34.02	37.46	40.14
LLC+SVM [17]	—	51.15	59.77	65.43	67.74	70.16	73.44	LLC+SVM [17]	34.36	41.19	45.31	47.68
Griffin [50]	—	44.2	54.5	59.0	63.3	65.8	67.60	Griffin [50]	28.30	34.10	—	—
SRC [13], [22]	—	48.8	60.1	64.9	67.5	69.2	70.7	SRC [13], [22]	27.86	33.33	—	—
D-K-SVD [21], [22]	—	49.6	59.5	65.1	68.6	71.1	73.0	D-K-SVD [21], [22]	—	33.72	—	48.88
BMDDL [23]	31.88	56.10	66.01	69.56	71.32	72.28	75.54	BMDDL [23]	35.41	41.56	46.90	52.32
LC-K-SVD [22]	28.9	54.0	63.1	67.7	70.5	72.3	73.6	LC-K-SVD [22]	28.9	34.32	—	—
FDDL [19]	—	51.1	60.6	65.6	68.5	70.4	71.0	FDDL [19]	—	—	—	—
SSPIC [15]	—	55.1	62.1	65.0	67.5	68.9	71.5	LSc [14]	29.99	35.74	38.47	40.32
LDA+GSVM	—	50.49	60.39	64.28	67.10	71.24	72.40	TDDL [24]	33.14	39.05	44.16	49.05
<i>SparLDA</i>	—	54.60	65.26	70.05	72.12	73.2	75.82	<i>SparLDA</i>	35.62	39.53	47.08	51.90
<i>LDA-SparLow</i>	31.23	56.44	67.12	73.82	74.70	76.20	76.86	<i>LDA-SparLow</i>	38.05	43.26	50.32	55.74
<i>SDA-SparLow</i>	46.12	67.42	72.01	76.12	76.64	77.40	78.25	<i>SparMFA</i>	36.42	40.89	47.62	51.76
<i>MFA-SparLow</i>	32.43	57.52	68.44	73.95	75.56	76.63	77.32	<i>MFA-SparLow</i>	38.82	44.05	51.34	56.82
<i>SMFA-SparLow</i>	46.02	68.66	72.92	76.02	77.24	77.80	78.42	<i>SparMVR</i>	32.90	37.91	45.03	50.64
<i>MVR-SparLow</i>	29.30	54.79	65.63	70.56	73.33	75.41	76.15	<i>MVR-SparLow</i>	36.29	40.66	47.92	52.56
<i>SMVR-SparLow</i>	44.82	66.43	70.80	75.48	76.32	77.14	77.76	—	—	—	—	—

Each category has at least 80 images. Unlike the Caltech-101 dataset, this dataset contains multiple objects in various poses at different locations within the images. Existence of background clutter and occlusion result in higher intraclass diversity, which makes the categorization task even harder.

We apply our *DHOG-SPP SparLow* on randomly selected 15, 30, 45, 60 training images per category, respectively. We set $\lambda = 0.1$, $m_{PCA} = 2560$, $k = 2048$, $r = 3128$. For LDA-like *SparLow*, MFA-like *SparLow* and MVR-like *SparLow*, we set $l = 255$, 361 and 387, respectively. Finally, we use GSVM for classifying the low dimensional representations. In this experiment, we use the sparse coding formulation associated with KL-divergence. Table 2 (b) shows that our results outperform the state of the art methods under all the cases. Moreover, we also implemented *TDDL* for comparison, and the each sub-dictionary size of *TDDL* is fixed as $r = 200$. It shows that *TDDL*'s perform worse than the *SparLow*'s. The possible reason is that *TDDL*'s associated with a binary classifier may suffer the huge number of classes.

6.4.4 15-Scenes dataset

We finally evaluate the *SparLow* framework on the 15-Scenes dataset [51]. This dataset contains totally 4485 images falling into 15 categories, with the number of images in each category ranging from 200 to 400 and image size around 300×250 pixels. The image content is diverse, containing not only indoor scenes, such as bedroom, kitchen, but also outdoor scenes, such as building and country views, etc.

Following the common experimental settings, we use *SIFT-SPP* as input with $k = 1024$ and $m_{PCA} = 2000$. For MFA, we set $k_1 = 70$, $k_2 = 100$, and $l = 50$. For *SparMFA* and *MFA-SparLow*, we set $k_1 = 30$, $k_2 = 100$ and $l = 60$. For all supervised and semi-supervised *SparLow* methods, the dictionary size $r = 750$. Table 3 compares our results with several sparse coding methods in [13], [14], [16], [17], [21], [22], [23], [53], GSVM, and the method in [51], which are all using SPP features as input data. As shown in Table 3, our approaches significantly outperform all state of the art approaches. Note that, the bottom three lines are all semi-supervised methods.

Methods	Accuracy	Methods	Accuracy
BMDDL [23]	96.9	Lobel [53]	86.3 ± 0.5
GSVM	72.2	LDA	91.69
KSPM [51]	83.50	<i>SparLDA</i>	95.89
ScSPM+GSVM [16]	80.28	<i>LDA-SparLow</i>	97.47
LLC+GSVM [17]	89.2	MFA	92.82
SRC [13], [22]	91.8	<i>SparMFA</i>	96.65
LSc [14]	89.7	<i>MFA-SparLow</i>	98.46
K-SVD [10] + LDA	92.6	MVR ($l = 512$)	93.10
D-K-SVD [21]	89.01	<i>SparMVR</i>	96.32
LC-K-SVD [22]	92.9	<i>MVR-SparLow</i>	97.55
FDDL [19]	90.2	—	—
SDA [33]	97.28	<i>SDA-SparLow</i>	99.18
SDE [40]	97.66	<i>SLap-SparLow</i>	99.25
LapRLS [42]	94.86	<i>SMVR-SparLow</i>	99.12

TABLE 3
Averaged classification Rate (%) comparison on 15-Scenes dataset.
The classifier is 1NN for the third column if not specified.

7 CONCLUSION

In this work, we present a low dimensional representation learning approach, coined as *SparLow*, which leverages both sparse representation and the trace quotient criterion. It can be considered as a two-layer disentangling mechanism, which applies the trace quotient criterion on the sparse representations. Our proposed generic cost function is defined on a sparsifying dictionary and an orthogonal transformation, which form a product Riemannian manifold. A geometric CG algorithm is developed for optimizing the *SparLow* function. Our experimental results depict that in comparison with the state of the art unsupervised, supervised and semi-supervised representation learnings methods, our proposed *SparLow* framework delivers promising performance in data visualization and classification. Moreover, the proposed *SparLow* is flexible and can be extended to more general cases of low dimensional representation learning models with orthogonal constraints.

ACKNOWLEDGMENTS

This work has been supported by the German Research Foundation (DFG) under Grant No. KL 2189/9-1, the CAS Pioneer Hundred Talents Program (Type C) under Grant No.2017-122, and the National Science Found for Young Scholars under Grant No. 61806186.

The authors would like to acknowledge Prof. Junbin Gao (The University of Sydney Business School, Australia) for his valuable opinions and suggestions.

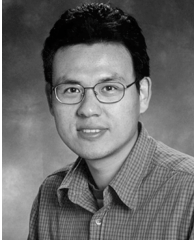
REFERENCES

- [1] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
- [2] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.
- [3] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2765–2781, 2013.
- [4] X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang, "Face recognition using laplacianfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 328–340, 2005.
- [5] P. N. Belhumeur, J. P. Hespanha, and D. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997.
- [6] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, , and S. Lin, "Graph embedding and extensions: A general framework for dimensionality reduction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 1, pp. 40–51, 2007.
- [7] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1789–1828, 2013.
- [8] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.
- [9] M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. Springer, 2010.
- [10] M. Aharon, M. Elad, 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.
- [11] 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.
- [12] S. Hawe, M. Kleinstenuber, and K. Diepold, "Analysis operator learning and its application to image reconstruction," *IEEE Transactions on Image Processing*, vol. 22, no. 6, pp. 2138–2150, 2013.
- [13] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2009.
- [14] S. Gao, I. W.-H. Tsang, and L.-T. Chia, "Laplacian sparse coding, hypergraph laplacian sparse coding, and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 92–104, 2013.
- [15] U. Srinivas, Y. Suo, M. Dao, V. Monga, and T. D. Tran, "Structured sparse priors for image classification," *IEEE Transactions on Image Processing*, vol. 24, no. 6, pp. 1763–1776, 2015.
- [16] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2009, pp. 1794–1801.
- [17] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2010, pp. 3360–3367.
- [18] F. Perronnin, "Universal and adapted vocabularies for generic visual categorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 7, pp. 1243–1256, 2008.
- [19] M. Yang, L. Zhang, X. Feng, and D. Zhang, "Sparse representation based Fisher discrimination dictionary learning for image classification," *International Journal of Computer Vision*, vol. 109, no. 3, pp. 209–232, 2014.
- [20] Q. Qiu, V. M. Patel, and R. Chellappa, "Information-theoretic dictionary learning for image classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 11, pp. 2173–2184, 2014.
- [21] Q. Zhang and B. Li, "Discriminative K-SVD for dictionary learning in face recognition," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2010, pp. 2691–2698.
- [22] Z. Jiang, Z. Lin, and L. S. 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.
- [23] P. Zhou, C. Zhang, and Z. Lin, "Bilevel model-based discriminative dictionary learning for recognition," *IEEE Transactions on Image Processing*, vol. 26, no. 3, pp. 1173–1187, 2017.
- [24] J. Mairal, F. Bach, and J. Ponce, "Task-driven dictionary learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 791–804, 2012.
- [25] J. Yang, K. Yu, and T. Huang, "Supervised translation-invariant sparse coding," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2010, pp. 3517–3524.
- [26] I. A. Gkioulekas and T. Zickler, "Dimensionality reduction using the sparse linear model," in *Advances in Neural Information Processing Systems 24*, J. Shawe-taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, Eds. The MIT Press, 2011, pp. 271–279.
- [27] J. Gao, Q. Shi, and T. S. Caetano, "Dimensionality reduction via compressive sensing," *Pattern Recognition Letters*, vol. 33, no. 9, pp. 1163–1170, 2012.
- [28] Y. Bengio, *Learning Deep Architectures for AI*, ser. Foundations and Trends in Machine Learning. Now Publishers Inc., 2009, vol. 2, no. 1.
- [29] Y. W. Teh, M. Welling, S. Osindero, and G. E. Hinton, "Energy-based models for sparse overcomplete representations," *Journal of Machine Learning Research*, vol. 4, pp. 1235–1260, 2003.
- [30] M. Ranzato, C. Poultney, S. Chopra, and Y. LeCun, "Efficient learning of sparse representations with an energy-based model," in *Advances in Neural Information Processing Systems 19*, 2006, pp. 1137–1144.
- [31] E. Kokiopoulou, J. Chen, and Y. Saad, "Trace optimization and eigenproblems in dimension reduction methods," *Numerical Linear Algebra with Applications*, vol. 18, no. 3, pp. 565–602, 2010.
- [32] D. Cai, X. He, J. Han, and H.-J. Zhang, "Orthogonal laplacianfaces for face recognition," *IEEE Transactions on Image Processing*, vol. 15, no. 11, pp. 3608–3614, 2006.
- [33] D. Cai, X. He, and J. Han, "Semi-supervised discriminant analysis," in *IEEE 11th International Conference on Computer Vision (ICCV)*, 2007, pp. 1–7.
- [34] X. Wei, H. Shen, and M. Kleinstenuber, "Trace quotient meets sparsity: A method for learning low dimensional image representations," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 5268–5277.
- [35] X. Wei, Y. Li, H. Shen, and Y. L. Murphey, "Joint learning dictionary and discriminative features for high dimensional data," in *Proceedings of the 25th International Conference on Pattern Recognition (ICPR)*, 2016.
- [36] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society, Series B*, vol. 67, no. 2, pp. 301–320, 2005.
- [37] J. A. Bagnell and D. M. Bradley, "Differentiable sparse coding," in *Advances in Neural Information Processing Systems 21*, 2009, pp. 113–120.
- [38] J. P. Cunningham and Z. Ghahramani, "Linear dimensionality reduction: Survey, insights, and generalizations," *Journal of Machine Learning Research*, vol. 16, pp. 2859–2900, 2015.
- [39] T. Ngo, M. Bellalij, and Y. Saad, "The trace ratio optimization problem for dimensionality reduction," *SIAM Journal on Matrix Analysis and Applications*, vol. 31, no. 5, pp. 2950–2971, 2010.
- [40] G. Yu, G. Zhang, C. Domeniconi, Z. Yu, and J. You, "Semi-supervised classification based on random subspace dimensionality reduction," *Pattern Recognition*, vol. 45, no. 3, pp. 1119–1135, 2012.
- [41] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," *Advances in neural information processing systems*, vol. 16, no. 16, pp. 321–328, 2004.
- [42] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *The Journal of Machine Learning Research*, vol. 7, pp. 2399–2434, 2006.
- [43] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*. Princeton, NJ: Princeton University Press, book, 2008.
- [44] M. Kleinstenuber and K. Hüper, "An intrinsic CG algorithm for computing dominant subspaces," in *Proceedings of the 32nd IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2007, pp. IV1405–IV1408.
- [45] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *Journal of Machine Learning Research*, vol. 11, pp. 19–60, 2010.
- [46] K. Skretting and K. Engan, "Recursive least squares dictionary learning algorithm," *IEEE Transactions on Signal Processing*, vol. 58, no. 4, pp. 2121–2130, 2010.
- [47] J. J. Hull, "A database for handwritten text recognition research," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 5, pp. 550–554, 1994.
- [48] H. Zou, T. Hastie, and R. Tibshirani, "Sparse principal component analysis," *Journal of Computational and Graphical Statistics*, vol. 15, no. 2, pp. 265–286, 2006.
- [49] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on

- 101 object categories,” *Computer Vision and Image Understanding*, vol. 106, no. 1, pp. 59–70, 2007.
- [50] G. Griffin, A. Holub, and P. Perona, “Caltech-256 object category dataset,” California Institute of Technology, Tech. Rep. 7694, 2007.
- [51] 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 (CVPR)*, vol. 2, 2006, pp. 2169–2178.
- [52] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [53] H. Lobel, R. Vidal, and A. Soto, “Learning shared, discriminative, and compact representations for visual recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 11, pp. 2218–2231, 2015.
- [54] P. Billingsley, *Probability and measure*. John Wiley & Sons, book, 2009.
- [55] H. Kim, and A. Mnih, “Disentangling by factorising,” *Proceedings of the 35th International Conference on Machine Learning (ICML)*, Stockholm, Sweden, 2018.
- [56] H. Amann and J. Escher, *Analysis II*. Birkhäuser Verlag, 2008.



Xian Wei (S'14-M'18) received the Ph.D. degree in Engineering from the Technical University of Munich, Munich, Germany, in 2017. Recently, he joined Fujian Institute of Research on the Structure of Matter, Chinese Academy of Sciences, China, as a leading Researcher of Machine Vision and Pattern Recognition Lab. His research interests focus on sparse coding, deep learning and geometric optimization. The applications include robotic vision, videos or images modeling, synthesis, recognition and semantics.



signal separation, sparse representation, deep representation learning, and reinforcement learning.

Hao Shen (S'04-M'08) received his PhD in Engineering from the Australian National University, Australia, in 2008. From December 2008 to September 2017, he was a post-doctoral researcher at the Institute for Data Processing, Technische Universität München, Germany. In October 2017, he joined fortiss, Research Institute of the Free State Bavaria, Germany, as the leader of Machine Learning Group. His research interests focus on machine learning for signal processing, e.g. geometric optimisation, blind



Martin Kleinstaubert received his Ph.D. in Mathematics from the University of Würzburg, Germany, in 2006. After post-doc positions at National ICT Australia Ltd., the Australian National University, Canberra, Australia, and the University of Würzburg, he has been appointed assistant professor for geometric optimization and machine learning at the Department of Electrical and Computer Engineering, TU München, Germany, in 2009. Since 2016, he is leading the Data Science Group at Mercateo AG, Munich.

APPENDIX

Note, that the numbering of equations in Appendix continues from the numeration in the manuscript. The proof of Proposition 1 requires the following two lemmas.

Lemma 1. *Let $g: \mathbb{R}^r \rightarrow \mathbb{R}$ satisfy Assumption 1. Then, for a given $\mathbf{x} \in \mathbb{R}^m$ and a dictionary $\mathbf{D} \in \mathbb{R}^{m \times r}$, a vector $\phi^* := [\varphi_1^*, \dots, \varphi_r^*]^\top \in \mathbb{R}^r$ is the unique solution to the sparse regression problem as in Eq. (5), if and only if, the following conditions hold*

$$\begin{cases} \mathbf{d}_i^\top (\mathbf{x} - \mathbf{D}\phi^*) = \nabla g_i(\varphi_i^*), & \text{for } \varphi_i \neq 0, \\ \mathbf{d}_i^\top (\mathbf{x} - \mathbf{D}\phi^*) \in \partial g_i(0), & \text{for } \varphi_i = 0. \end{cases} \quad (63)$$

Here $\partial g_i(0) = [-b_1, b_2]$ with $b_1, b_2 > 0$ is the subgradient of element-wise regulariser g_i at 0.

Proof. Since the cost function Eq. (5) is strictly convex as a result of g being strictly convex, the unique solution ϕ^* satisfies the subdifferential optimality condition, i.e.,

$$0 \in \partial f_s(\phi^*), \quad (64)$$

where $\partial f_s(\phi^*)$ is the subdifferential of f_s at ϕ^* . By the construction that the cost function is separable, the subdifferential of the cost function is computed as

$$\mathbf{d}_i^\top (\mathbf{x} - \mathbf{D}\phi^*) \in \partial g_i(\varphi_i^*). \quad (65)$$

By the fact that $\partial g_i(\varphi_i) = \nabla g_i(\varphi_i)$ for all $\varphi_i^* \neq 0$, the result follows. \square

Lemma 2. *Let $g: \mathbb{R}^r \rightarrow \mathbb{R}$ satisfy Assumption 1, and Assumption 2 hold true. Then, for a given $\mathbf{x} \in \mathbb{R}^m$ and a dictionary $\mathbf{D} \in \mathbb{R}^{m \times r}$, the unique solution $\phi^* := [\varphi_1^*, \dots, \varphi_r^*]^\top \in \mathbb{R}^r$ to the sparse regression problem as in Eq. (5) is smooth in an open neighborhood around \mathbf{D}^* .*

Proof. Recall the definition of support $\mathfrak{P}(\mathbf{x}, \mathbf{D}^*)$ as in Eq. (42), we define its complement set as

$$\Omega(\mathbf{x}, \mathbf{D}^*) := \{i \in \{1, \dots, r\} | \varphi_i^* = 0\}. \quad (66)$$

Let $k = |\mathfrak{P}(\mathbf{x}, \mathbf{D})|$, $\phi_{\mathfrak{P}} \in \mathbb{R}^k$ and $\mathbf{D}_{\mathfrak{P}} \in \mathbb{R}^{m \times k}$ being the subset of $\mathbf{D} \in \mathbb{R}^{m \times r}$, in which the indices of columns fall into the support \mathfrak{P} . We construct a function as

$$\begin{aligned} \eta: \mathbb{R}^{m \times r} \times \mathbb{R}^r &\rightarrow \mathbb{R}^r, \\ \eta(\mathbf{D}, \phi) &:= \mathbf{D}_{[\mathfrak{P}, \Omega]}^\top (\mathbf{x} - \mathbf{D}_{[\mathfrak{P}, \Omega]} \phi) - \begin{bmatrix} \nabla g_{\mathfrak{P}}(\phi_{\mathfrak{P}}) \\ \phi_{\Omega} \end{bmatrix}, \end{aligned} \quad (67)$$

with $\phi = [\phi_{\mathfrak{P}}^\top, \phi_{\Omega}^\top]^\top$, $\phi_{\mathfrak{P}} \in \mathbb{R}^k$, and $\phi_{\Omega} \in \mathbb{R}^{r-k}$. The function η is smooth in both \mathbf{D} and ϕ , and $\eta(\mathbf{D}^*, \phi^*) = 0$. Taking the directional derivative of η with respect to \mathbf{D} and ϕ leads to

$$\begin{aligned} &\mathbf{D} \eta(\mathbf{D}, \phi)(\mathbf{H}_{\mathbf{D}}, \mathbf{h}_{\phi}) \\ &= \underbrace{[\chi_1(\mathbf{D}_{[\mathfrak{P}, \Omega]}, \phi) \quad \chi_2(\mathbf{D}_{[\mathfrak{P}, \Omega]}, \phi)]}_{=: J_{\eta}(\mathbf{D}, \phi) \in \mathbb{R}^{r \times (r+m \cdot r)}} \cdot \begin{bmatrix} \mathbf{h}_{\phi} \\ \text{vec}(\mathbf{H}_{\mathbf{D}}) \end{bmatrix}, \end{aligned} \quad (68)$$

where

$$\chi_1(\mathbf{D}_{[\mathfrak{P}, \Omega]}, \phi) := \begin{bmatrix} \mathbf{H}g_{\mathfrak{P}}(\phi_{\mathfrak{P}}) & \mathbf{0}_{k, r-k} \\ \mathbf{0}_{r-k, k} & \mathbf{I}_{r-k} \end{bmatrix} + \mathbf{D}_{[\mathfrak{P}, \Omega]}^\top \mathbf{D}_{[\mathfrak{P}, \Omega]}, \quad (69)$$

and $\chi_2(\mathbf{D}_{[\mathfrak{P}, \Omega]}, \phi) \in \mathbb{R}^{r \times (m \cdot r)}$ is a tedious term without clear knowledge on its rank. Here, $\mathbf{H}g_{\mathfrak{P}}(\phi_{\mathfrak{P}}) \in \mathbb{R}^{k \times k}$ is the Hessian matrix of the separable sparsifying function restricted on the support \mathfrak{P} , i.e., $g_{\mathfrak{P}} := \{g_i\} \in \mathbb{R}^k$ with $i \in \mathfrak{P}$. The matrix $J_{\eta}(\mathbf{D}, \phi)$ is known as the *Jacobian matrix* of η . Our aim is

to ensure the full rankness of the Jacobian matrix to apply the implicit function theorem [1]. Often, it is assumed that $m \leq r$. Let $\kappa(\mathbf{D})$ be the *spark* of \mathbf{D} , i.e., the smallest number of columns from \mathbf{D} that are linearly dependent [2]. If $k < \kappa(\mathbf{D})$, then the Jacobian matrix $J_\eta(\mathbf{D}, \phi)$ is of full rank. Unfortunately, the spark is often very difficult to control during an optimization procedure. Thus, a simple but general fix to make $J_\eta(\mathbf{D}, \phi)$ have full rank is to have Hessian $Hg_{\mathfrak{P}}(\phi_{\mathfrak{P}})$ non-degenerate. In other words, by Assumption 2 and the implicit function theorem, there exist two open neighborhood containing \mathbf{D}^* and ϕ^* , i.e., $\mathbf{D}^* \in \mathcal{U}$ and $\phi^* \in \mathcal{W}$, and a unique continuously differentiable function $\hat{\phi}: \mathcal{U} \rightarrow \mathcal{W}$, so that $\hat{\phi}(\mathbf{D}^*) = \phi^*$ and $\eta(\mathbf{D}, \hat{\phi}(\mathbf{D})) = 0$ for all $(\mathbf{D}, \phi) \in \mathcal{U} \times \mathcal{W}$.

Let us denote $\hat{\phi}(\mathbf{D}) := [\hat{\phi}_1(\mathbf{D}), \dots, \hat{\phi}_r(\mathbf{D})]^\top \in \mathbb{R}^r$. Since $\hat{\phi}$ is continuously differentiable in \mathbf{D} and $\hat{\phi}(\mathbf{D}^*) = \phi^*$, there exists an open subset $\mathcal{U}_\epsilon \subset \mathcal{U}$, so that the following holds true with a gap $\epsilon > 0$ and $\epsilon < \min\{b_1, b_2\}$

$$\begin{cases} |\hat{\phi}_i(\mathbf{D}) - \phi_i^*| < \epsilon, & \text{for } i \in \mathfrak{P}; \\ |\hat{\phi}_i(\mathbf{D})| < \epsilon, & \text{for } i \notin \mathfrak{P}. \end{cases} \quad (70)$$

We then construct the following projection

$$\Pi_{\mathfrak{P}}: \mathbb{R}^r \rightarrow \mathbb{R}^r, \quad \phi \mapsto \tilde{\phi} := [\tilde{\phi}_1, \dots, \tilde{\phi}_r]^\top, \quad (71)$$

where

$$\tilde{\phi}_i := \begin{cases} \phi_i, & \text{for } \phi_i \notin \partial g_i(0); \\ 0, & \text{for } \phi_i \in \partial g_i(0). \end{cases} \quad (72)$$

It is clear that the projection $\Pi_{\mathfrak{P}}$ does not change support, and is hence smooth in ϕ . Consequently, the composition $\Pi_{\mathfrak{P}}(\hat{\phi}(\mathbf{D}))$ is a smooth function in \mathbf{D} , and is the unique solution of the sparse regression problem for given (\mathbf{D}, \mathbf{x}) with $\mathbf{D} \in \mathcal{U}_\epsilon$ by Lemma 1, i.e., $\Pi_{\mathfrak{P}}(\hat{\phi}(\mathbf{D})) = \phi^*(\mathbf{D})$. Thus, the result follows. \square

Finally, the proof of Proposition 1 is straightforward.

Proof. For each sample \mathbf{x}_i with $i = 1, \dots, n$, let us denote by ϕ_i^* the sparse representation with respect to a common dictionary \mathbf{D}^* . We further denote by \mathcal{U}_{ϵ_i} an open neighborhood containing \mathbf{D}^* , so that the unique solution $\phi_i^*(\mathbf{D})$ is smooth. Then a finite number of unions of sets \mathcal{U}_{ϵ_i} is a non-empty set as

$$\mathcal{U}_{\epsilon^*} := \bigcap_{i=1}^n \mathcal{U}_{\epsilon_i} \quad (73)$$

with $\epsilon^* := \min_i \epsilon_i$. It is straightforward to conclude that the complete collection of $\Phi^* = [\phi_1^*, \dots, \phi_n^*]$ is smooth in \mathcal{U}_{ϵ^*} . \square

In the rest of this appendix, we show a lemma, which enables development of gradient based algorithms.

Proposition 2. *Let $g: \mathbb{R}^r \rightarrow \mathbb{R}$ satisfy Assumption 1, and for a given $\mathbf{x} \in \mathbb{R}^m$ and a dictionary $\mathbf{D}^* \in \mathbb{R}^{m \times r}$, let $\phi^* \in \mathbb{R}^r$ be the unique solution to the sparse regression problem as in Eq. (5). Then, the first derivative of $\phi_{\mathfrak{P}}^*$ has a close form expression as*

$$D\phi_{\mathfrak{P}}^*(\mathbf{D}_{\mathfrak{P}})\mathbf{H} = (K(\mathbf{D}_{\mathfrak{P}}))^{-1}(\mathbf{H}^\top \mathbf{x} - (\mathbf{H}^\top \mathbf{D}_{\mathfrak{P}} + \mathbf{D}_{\mathfrak{P}}^\top \mathbf{H})\phi_{\mathfrak{P}}^*). \quad (74)$$

Proof. Recall the results from Lemma 2, the unique solution $\phi^* = [\phi_1^*, \dots, \phi_r^*]^\top \in \mathbb{R}^r$ is smooth in an open neighborhood around \mathbf{D}^* . Since the support $\mathfrak{P}(\mathbf{x}_i, \mathbf{D}^*)$ stays unchanged, the vector $\phi_{\mathfrak{P}}$ is smooth in an appropriate open neighborhood containing $\mathbf{D}_{\mathfrak{P}}$, i.e., we can compute

$$\nabla_{g_{\mathfrak{P}}}(\phi_{\mathfrak{P}}) = \mathbf{D}_{\mathfrak{P}}^\top(\mathbf{x} - \mathbf{D}_{\mathfrak{P}}\phi_{\mathfrak{P}}). \quad (75)$$

We then take the derivative on the both sides of Eq. (75) with respect to $\mathbf{D}_{\mathfrak{P}}$ in direction $\mathbf{H} \in T_{\mathbf{D}_{\mathfrak{P}}} \mathfrak{S}(m, k)$ as

$$D(\nabla_{g_{\mathfrak{P}}}(\phi_{\mathfrak{P}}(\mathbf{D}_{\mathfrak{P}})))\mathbf{H} = \mathbf{H}^\top(\mathbf{x} - \mathbf{D}_{\mathfrak{P}}\phi_{\mathfrak{P}}) - \mathbf{D}_{\mathfrak{P}}^\top \mathbf{H} \phi_{\mathfrak{P}} - \mathbf{D}_{\mathfrak{P}}^\top \mathbf{D}_{\mathfrak{P}} D\phi_{\mathfrak{P}}(\mathbf{D}_{\mathfrak{P}})\mathbf{H}. \quad (76)$$

The left hand side in Eq. (76) can be computed by

$$D(\nabla_{g_{\mathfrak{P}}}(\phi(\mathbf{D})))\mathbf{H} = Hg_{\mathfrak{P}}(\phi_{\mathfrak{P}}) \cdot D\phi_{\mathfrak{P}}(\mathbf{D}_{\mathfrak{P}})\mathbf{H}, \quad (77)$$

where $Hg_{\mathfrak{P}}(\phi_{\mathfrak{P}}) \in \mathbb{R}^{k \times k}$ is the Hessian matrix of function $g_{\mathfrak{P}}$. It is positive definite by Assumption 1. Substituting Eq. (77) into Eq. (76) leads to a linear equation in $D\phi_{\mathfrak{P}}(\mathbf{D}_{\mathfrak{P}})\mathbf{H}$ as

$$K(\mathbf{D}_{\mathfrak{P}}) \cdot D\phi_{\mathfrak{P}}(\mathbf{D}_{\mathfrak{P}})\mathbf{H} = \mathbf{H}^\top \mathbf{x} - (\mathbf{H}^\top \mathbf{D}_{\mathfrak{P}} + \mathbf{D}_{\mathfrak{P}}^\top \mathbf{H})\phi_{\mathfrak{P}}. \quad (78)$$

where $K(\mathbf{D}_{\mathfrak{P}}) := Hg_{\mathfrak{P}}(\phi_{\mathfrak{P}}) + \mathbf{D}_{\mathfrak{P}}^\top \mathbf{D}_{\mathfrak{P}}$ is positive definite. Thus, the closed form expression as in Eq. (74) follows. \square

REFERENCES

- [1] H. Amann and J. Escher, *Analysis II*. Birkhäuser Verlag, 2008.
- [2] M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. Springer, New York, 2010.