Online Low-Rank Representation Learning for Joint Multi-subspace Recovery and Clustering

Bo Li, Risheng Liu, Junjie Cao, Jie Zhang, Yu-Kun Lai, Xiuping Liu

***Abstract*—Benefiting from global rank constraints, the low-rank representation (LRR) method has been shown to be an effective solution to subspace learning. However, the global mechanism also means that the LRR model is not suitable for handling large-scale data or dynamic data. For large-scale data, the LRR method suffers from high time complexity, and for dynamic data, it has to recompute a complex rank minimization for the entire data set whenever new samples are dynamically added, making it prohibitively expensive. Existing attempts to online LRR either take a stochastic approach or build the representation purely based on a small sample set and treat new input as out-of-sample data. The former often requires multiple runs for good performance and thus takes longer time to run, and the latter formulates online LRR as an out-of-sample classification problem and is less robust to noise. In this paper, a novel online low-rank representation subspace learning method is proposed for both large-scale and dynamic data. The proposed algorithm is composed of two stages: static learning and dynamic updating. In the first stage, the subspace structure is learned from a small number of data samples. In the second stage, the intrinsic principal components of the entire data set are computed incrementally by utilizing the learned subspace structure, and the low-rank representation matrix can also be incrementally solved by an efficient online singular value decomposition (SVD) algorithm. The time complexity is reduced dramatically for large-scale data, and repeated computation is avoided for dynamic problems. We further perform theoretical analysis comparing the proposed online algorithm with the batch LRR method. Finally, experimental results on typical tasks of subspace recovery and subspace clustering show that the proposed algorithm performs comparably or better than batch methods including the batch LRR, and significantly outperforms state-of-the-art online methods.**

***Index Terms*—Low-rank representation, subspace learning, large-scale data, dynamic data, online learning.**

I. Introduction

Multi-subspace recovery and clustering are two basic tasks in machine learning. Generally, it is assumed that the data points are drawn from multiple low-dimensional manifold subspaces. The basic task of subspace recovery is to extract the underlying low-dimensional subspaces, and subspace clustering is to segment the data into the corresponding subspaces correctly. Benefiting from the global mechanism, representation-based subspace learning has attracted considerable attention in recent years. Low-rank representation (LRR) [1] is one of the popular self-expressive subspace learning methods, which aims at jointly finding the lowest rank of the whole data space. LRR has shown good performance in numerous research problems in computer vision, such as salient object detection [2], segmentation and grouping [3], [4], background subtraction [5], tracking [6] and 3D visual recovery [7], etc.

Benefiting from the global self-expressiveness framework, LRR can effectively extract the intrinsic manifold structure of the global space and is robust to noise and outliers. However, the self-expressiveness framework aims at finding the representation relationships of the whole data space jointly, which leads to the limitation that most of the existing LRR subspace learning algorithms are batch methods processing whole data simultaneously and designed for static data, i.e., the dataset is fixed during processing. However, batch methods have significant drawbacks: 1) The computational complexity can be high with a large number of sample points. 2) Learning methods designed for static data cannot handle dynamic problems effectively where new sample points are incrementally generated, and the learned subspaces need to be updated accordingly. Dynamic data is becoming increasingly popular with sensor data such as surveillance videos, traffic control sensor data, as well as Internet data dynamically uploaded by users. In such scenarios, learning subspace structures from dynamic data is essential. Static learning methods attempt to extract subspace structures by utilizing the full data, which is not applicable for dynamic data. Furthermore, whenever the data is updated, the static learning process has to be repeatedly reapplied to the entire data set, which is prohibitively expensive.

The most relevant work to this paper is [8]. In [8], the large-scale LRR problem is formulated as an out-of-sample classification problem under the assumption that the subspace structure of the whole data space can be learned from a small portion of it. Firstly, a small number of data points are chosen as the in-sample data to learn the structure of the whole space, and then each out-of-sample data is assigned to the nearest subspace spanned by in-sample data according to the minimal residual of original data. While being efficient, the method does *not* really compute the original low-rank representation of the out-of-sample data, and is less robust to noise for subspace clustering.

Bo Li is with the School of Mathematics and Information Sciences, Nanchang Hangkong University, Nanchang, 330063, School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin, 541004, China. e-mail: libo@nchu.edu.cn

Risheng Liu is with DUT-RU International School of Information Science & Engineering, Dalian University of Technology, Dalian, 116620, the State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an, 710071, and Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, Dalian University of Technology, Dalian, 116620.

Junjie Cao (corresponding author, jjcao1231@gmail.com) and Xiuping Liu are with the School of Mathematical Sciences, Dalian University of Technology, Dalian, 116024, China.

Jie Zhang is with the School of Mathematics, Liaoning Normal University, Dalian, 116029, China.

Yu-Kun Lai is with the School of Computer Sciences and Informatics, Cardiff University, Cardiff, CF24 3AA, UK.

Another related online LRR method is [9], in which LRR is solved based on stochastic optimization of an equivalent reformulation of the batch LRR. Although it is designed for unsupervised clustering, it can be easily generalized for semi-supervised subspace learning. The algorithm processes one sample per time instance and hence its memory cost is independent of the total number of samples, significantly enhancing the computation and storage efficiency. However, due to the mechanism of stochastic optimization, the learning performance is poor at the beginning, and improves after a sufficient number of stochastic iterations. In order to improve the performance, multiple runs over the dataset are applied [9], thus it can be time-consuming for large-scale data and may not always be suitable for dynamic data. Moreover, the stochastic optimization process also means that the method can be misled in early iterations by data samples corrupted with noise, resulting in a performance drop even with multiple runs, as we later show in the experimental results.

In this paper, we focus on the study of a novel online LRR learning method for joint multi-subspace recovery and clustering. We assume that the *static training* data covers all the subspaces, so the *initial* subspace structure of the whole data space can be learned from the partial data, similar to [8]. However, *fundamental differences* exist: for our method, the intrinsic low-rank representation of the entire data set is incrementally learned, while for [8], only the in-sample data is used to learn the clustering structure, and the out-of-sample data is not used for learning but simply *classified* to the nearest subspace spanned by in-sample data. Although the Frobenius-norm-based learning method used in [8] has been proved to be a good approximation to the nuclear-norm learning [10], it still suffers from noise as the out-of-sample data is learned in the original data space rather than the intrinsic space used in the proposed method.

The flowchart of the proposed method is shown in Fig. 1. The algorithm consists of two stages: static learning and dynamic online updating. Firstly, the intrinsic subspace structure is learned from a subset of the whole data. In the second stage, the principal components of the remaining data will be incrementally pursued, and the global representation matrix on the whole data will be updated incrementally and efficiently.

The main contributions of this method include:

- Compared to batch LRR, our online LRR learning method reduces the computational complexity for large-scale data dramatically while producing subspace learning results of comparable quality..
- Our online LRR avoids repeatedly recomputing the complex low-rank optimization when new data points are introduced, and thus can handle online data efficiently; such data is prohibitively expensive to process with batch LRR methods.
- Our method does not suffer from limitations of existing online LRR methods. In particular, our method achieves significantly improved learning accuracy over existing online LRR methods while being much faster. In addition, our method is much more robust to noise.

The rest of this paper is organized as follows. Section II gives a brief review on related work. The preliminaries about low-rank based subspace learning is introduced in Section III. In Section IV, we propose the framework of our online LRR subspace learning method, and further present theoretical studies which show that under certain conditions the subspaces learned using our online method are identical to those learned by the batch method. Experimental results are shown in Section V. In order to evaluate the performance of the proposed method, we compare it with related state-of-the-art methods (both batch and online). Finally, we draw conclusions and discuss future work in Section VI.

II. Related work

A key component in subspace learning is to construct a good affinity graph of the data space. In general, based on the ways affinity graphs are constructed, such methods can be classified into local distance based and global linear representation-based.

Traditional local methods adopt Euclidean distances between pairwise data points to build similarity graphs. These methods include Laplacian Eigenmaps [11], K-nearest neighbors (K-NN) [12], Locally Linear Embedding (LLE) [13] etc. Local methods can capture the local structure of the data space, and the produced affinity graph is sparse and discriminative. However, they ignore the global characteristics of the entire data set, so are sensitive to noise and outliers. Compared with local distance based methods, global representation-based methods assume that each data point can be linearly represented by the basis formed by an over-complete dictionary. Regularizations are needed on the representation space to ensure unique solutions, and various methods are developed based on different regularizations, including sparse subspace clustering (SSC) [14], low-rank representation (LRR) [1], etc.

Sparse subspace clustering assumes that a data point lying in the union of multiple subspaces admits a sparse representation with respect to the dictionary formed by all the other data points. It has also shown that under the assumption that the subspaces are independent, the data points will be segmented into the underlying subspaces according to the sparse representation coefficients. SSC has achieved state-of-the-art performance in several applications, such as face recognition [15], image stylization [16], image enhancement [17], etc. Compared with sparse representation models, low-rank representation methods based on the rank constraints on the whole data are more suitable to pursue intrinsic structure of the data space. For instance, Robust Principal Component Analysis (RPCA) [18] proposed by Candés et al. shows that under some mild conditions the data points sampled from a single subspace can be exactly recovered by the rank minimization model. The work by Liu et al. [1] extends the recovery of corrupted data from a single subspace to multiple subspaces, and finds that the structure of multiple subspaces can be robustly revealed by the lowest rank representation coefficients of a given dictionary. In [1] rigorous theoretical studies are also provided to show that the representation matrix has block diagonal structure under some mild conditions, which is crucial to the subspace clustering problem.

However, both SSC and LRR are under the self-expressiveness framework, i.e., each sample is represented by

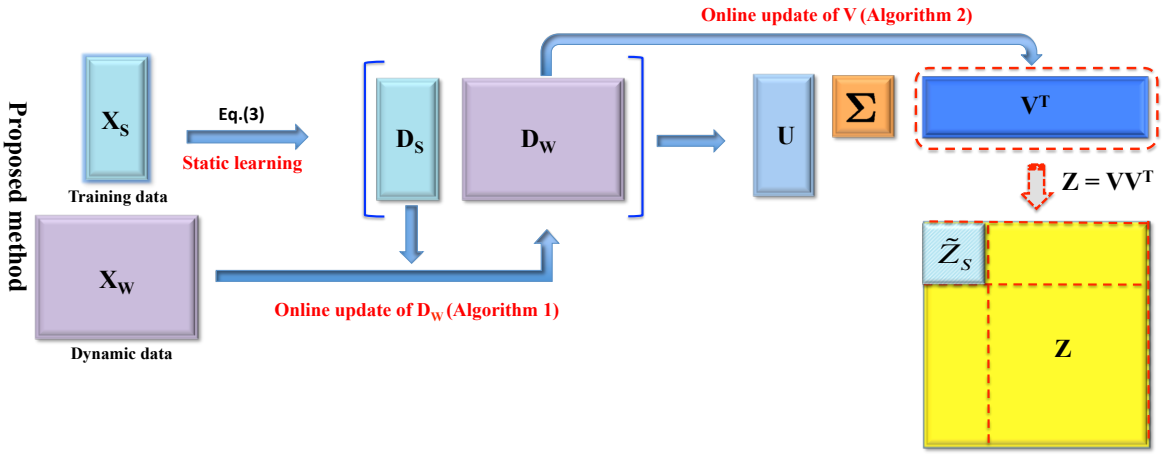


Fig. 1. Illustration of the proposed online subspace learning algorithm.

a linear combination of the remaining samples. They are not suitable for large-scale or dynamic data clustering. Firstly, the time complexities of SSC and LRR are proportional to the number of samples in the whole dataset to the third power, which are expensive for large-scale clustering problems. Even with fast implementation, the time complexity is still high for large-scale data. Secondly, they cannot practically handle dynamic data because they have to recompute the complex learning process for the whole data repeatedly when new samples are added, which is prohibitively time consuming. In this paper, we focus on the study of an LRR subspace learning method for large-scale and dynamic data.

Online methods that process data incrementally provide a feasible way to solve large-scale and dynamic problems. However, since rank minimization tightly depends on the whole data matrix and the constraints are coupled, it is challenging to extend existing LRR algorithms to provide an online solution of the low-rank based clustering problem. From this perspective, two recent works for online LRR are most relevant. Shen et al. [9] extend the online algorithm of RPCA to LRR by using stochastic optimization. Although the time complexity is dramatically reduced, it suffers from the following limitations: Due to the stochastic nature, the learning performance is poor at the beginning with few samples, and gradually improves after a sufficient number of stochastic iterations. In order to improve the performance of the initial samples, the paper uses a strategy where samples are fed into the algorithm in multiple iterations. While effective in improving the learning performance, the computational complexity can be high for large-scale dynamic data. Different from [9], Peng et al. [8] do not focus on solving the original LRR problem. Their method is designed for subspace clustering via a classification process. The algorithm is composed of four steps, namely sampling, clustering, coding and classification. Firstly, the large-scale data set is split into two parts: in-sample data and out-of-sample data. In the first two steps, a small number of data points are chosen as the in-sample data and the cluster membership between them is computed. Then in the third and fourth steps each out-of-sample data point is assigned to the nearest subspace spanned by in-sample data according to the minimal residual criterion. The method is efficient compared with batch LRR. However, as the classification is based on the representation learned from the original data rather than the learned intrinsic features for the entire data set, this method is sensitive to noise. In this paper, we propose a novel online LRR method which does not require multiple iterations of processing sample data, and efficiently and effectively learn subspace structure suitable for both online subspace classification and recovery. As we will show later, the learning accuracy of our approach is comparable to that of the batch methods, and significantly better than existing online methods, especially for noisy data sets. Our method is also at least several times faster than existing online methods [8], [9] for larger data sets.

Due to the powerful representation learning ability of deep learning, subspace learning frameworks based on deep learning [19] have recently been proposed. Deep learning methods are good at learning high-level features, and benefit much from the powerful computing capability of GPU for massively parallel computation. For the work [19], a sparse constraint on the *whole* dataset has to be computed in advance which is prohibitively expensive for large-scale problems and unsuitable for dynamic problems. However, under the similar assumption as proposed in our paper, i.e., the true subspace structure can be recovered by partial training data, the method [19] can potentially be extended to handle large-scale and dynamic problems.

III. Preliminaries: Low-Rank Based Subspace Learning

Given sufficient samples from c independent subspaces, the task of subspace learning is to extract the underlying low-dimensional subspaces where high-dimensional data samples lie in. Let d be the dimension of the data samples. By arranging the n_i samples from the i-th class as columns of a matrix $\mathbf{X}_i = [x_{i,1}, x_{i,2}, ..., x_{i,n_i}] \in \mathbb{R}^{d \times n_i}$, we obtain the data matrix $\mathbf{X} = [\mathbf{X}_1\ \mathbf{X}_2\ ...\ \mathbf{X}_c] \in \mathbb{R}^{d \times n}$, where $n = \sum_{i=1}^{c} n_i$ is the total number of samples.

A. Robust Principal Component Analysis (RPCA)

RPCA aims at recovering a low-rank data matrix $\mathbf{D}$ from corrupted observations $\mathbf{X} = \mathbf{D} + \mathbf{E}$, where $\mathbf{E}$ is the error matrix. The corrupted entries in $\mathbf{E}$ are unknown and the errors can be arbitrarily large, but they are assumed to be sparse. Under the above assumption, RPCA can be solved by solving the following regularized rank minimization problem:

$$\min_{\mathbf{D},\mathbf{E}} rank(\mathbf{D}) + \lambda\|\mathbf{E}\|_0,\ s.t.\ \mathbf{X} = \mathbf{D} + \mathbf{E}$$

where λ is the balance parameter. However, the rank function is not convex and difficult to optimize. Under some mild conditions, the optimization is equivalent to the following convex problem:

$$\min_{\mathbf{D},\mathbf{E}} \|\mathbf{D}\|_* + \lambda\|\mathbf{E}\|_1,\ s.t.\ \mathbf{X} = \mathbf{D} + \mathbf{E}$$

where $\|\cdot\|_*$ means the nuclear norm, which is the best convex envelope of the rank. The nuclear norm of a matrix is equal to the sum of the singular values of the matrix. The work [20] shows that under fairly general conditions, the problem can be solved even if the rank of $\mathbf{D}$ grows almost linearly w.r.t. the dimension of the matrix, and the errors in $\mathbf{E}$ are up to a constant fraction of all entries.

RPCA has been successfully applied to many machine learning and computer vision problems, such as automatic image alignment [21], [22], face modeling [23] and visual tracking [24].

B. Low-Rank Representation (LRR)

LRR is a typical representation-based subspace learning method, assuming a data vector can be represented as a linear combination of the remaining vectors. Given a set of data vectors drawn from a union of multiple subspaces, LRR aims at simultaneously finding the lowest rank representation of the whole data. Compared with sparse subspace clustering (SSC), LRR better captures the global subspace structure due to the use of global rank constraints.

For the noise-free case, LRR takes the data $\mathbf{X}$ itself as a dictionary and seeks the representation matrix $\mathbf{Z}$ with the lowest rank:

$$\min_{\mathbf{Z}} rank(\mathbf{Z}),\ s.t.\ \mathbf{X} = \mathbf{X}\mathbf{Z}.$$

Similar to RPCA, the above problem is NP-hard, and can be relaxed to the following convex optimization:

$$\min_{\mathbf{Z}} \|\mathbf{Z}\|_*,\ s.t.\ \mathbf{X} = \mathbf{X}\mathbf{Z}.$$

When the data is noisy, an additional error matrix $\mathbf{E}$ is introduced, which is assumed to be sparse, leading to an $\ell_{2,1}$-norm term added to the objective function:

$$\min_{\mathbf{Z}} \|\mathbf{Z}\|_* + \lambda\|\mathbf{E}\|_{2,1},\ s.t.\ \mathbf{X} = \mathbf{X}\mathbf{Z} + \mathbf{E}. \tag{1}$$

Although LRR has achieved state-of-the-art performance for certain applications, the computation complexity of the LRR model is as high as $O(n^3)$, where n is the number of data samples. Therefore LRR cannot efficiently handle large-scale data. In [25], Lin et al. proposed a linearized alternating direction method to solve the LRR model. Although it is accelerated by linearizing the quadratic term in the subproblem, the complexity is still $O(n^2)$. Liu et al. [1] proposed an accelerated solver for the LRR model with a pre-calculated orthogonal matrix, which has the complexity of $O(d^2n + d^3)$ for each iteration, where $d \ll n$. However, this algorithm usually suffers from low convergence rate, and many iterations are often needed. In [10], the connections between nuclear-norm and Frobenius-norm-based representations were studied. It is theoretically proved that both nuclear-norm and Frobenius-norm-based learning methods can be unified into a common framework, i.e., they are in the form of $\mathbf{V}\mathcal{P}(\mathbf{\Sigma})\mathbf{V}^T$, where $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ is the singular value decomposition (SVD) of a given data matrix and $\mathcal{P}(\cdot)$ denotes the shrinkage-thresholding operator. However, the computational complexity of Frobenius-norm-based methods is still high, especially for large scale data, as they have to compute the complex SVD operation.

C. Robust Shape Interaction (RSI)

Since corrupted data is used as the dictionary, the LRR model (1) can only work when the noise is sample-specific, i.e., some data points are corrupted and the remainder are clean. When the noise level is high or the proportion of outliers is relatively large, it cannot extract the intrinsic subspace structure correctly.

In [26], an improved version of LRR called Robust Shape Interaction (RSI) is proposed:

$$\min_{\mathbf{D},\mathbf{E}} rank(\mathbf{D}) + \lambda\|\mathbf{E}\|_{2,1},\ s.t.\ \mathbf{X} = \mathbf{D} + \mathbf{E} \tag{2}$$
$$\min_{\mathbf{Z}} rank(\mathbf{Z}),\ s.t.\ \mathbf{D} = \mathbf{D}\mathbf{Z}.$$

Intuitively, this model removes most of the noise, and adopts cleaner data as the dictionary, so it is more robust than the standard LRR, in particular when the data is heavily corrupted.

IV. Proposed method

For large-scale data, the computation complexity is a major challenge for existing LRR methods. For dynamic data where new samples are incrementally added, it is impossible to load the whole dataset for learning, and repeated computation when each time a new sample is added is prohibitively expensive, even for mid-scale problems. In this section, an efficient online LRR subspace learning algorithm is proposed for both large-scale data and dynamic data, addressing these fundamental limitations.

The proposed online LRR method is based on the following general assumptions: (1) data points are drawn from independent subspaces; (2) the static training data covers all the subspaces, which implies that one can use a small portion of data to learn the subspace structure of the whole data space. Similar assumptions are also made in [8]. However, unlike [8], we do *not* make the assumption that the subspace structure learned from the subset of data is sufficiently accurate, and use the remaining data to incrementally refine the learned subspace structure.

The proposed algorithm consists of the following two stages: static learning and dynamic updating. In the first stage,

a subset of data points are chosen as training data to learn the intrinsic subspace structure based on the batch LRR model. Then in the second dynamic updating stage, the low-rank components for the remaining data or dynamically added data will be updated using sparse reconstruction based on the subspace basis previously learned in the static training stage, and finally the global low-rank affinity matrix will be efficiently solved by utilizing the training data and dynamic data with the Sequential Karhunen-Loeve (SKL) [27] algorithm. As only a small portion of data points are involved in the complex low-rank optimization, the computational complexity is reduced dramatically for large-scale data. For dynamic data, there is no need to solve the rank minimization problem repeatedly when new samples are dynamically added, and the global affinity matrix can be updated incrementally based on the existing results efficiently.

A. Static Learning

The goal of the static learning stage is to learn the intrinsic low-dimensional structure of high-dimensional data samples using partial data from the whole data space. As initialization, a small number of samples which can cover all of the c subspaces are randomly chosen as the training data (see Fig. 2 (b) for an example). Let m_i be the number of samples from the i-th subspace. $\mathbf{X}_S^i = [x_{i,1}, x_{i,2}, ..., x_{i,m_i}] \in \mathbb{R}^{d\times m_i}$ is the matrix containing all the samples from the i-th subspace. They form the sampled training data matrix $\mathbf{X}_S = [\mathbf{X}_S^1\ \mathbf{X}_S^2\ ...\ \mathbf{X}_S^c] \in \mathbb{R}^{d\times m}$, where $m = \sum_{i=1}^{c} m_i$ is the total number of the training samples. With the partial data $\mathbf{X}_S$, the low-rank component matrix $\mathbf{D}_S$ and the intrinsic representation matrix $\mathbf{Z}_S$ can be recovered by the following minimization problem :

$$\min_{\mathbf{D}_S,\mathbf{E}_S} rank(\mathbf{D}_S) + \lambda\|\mathbf{E}_S\|_{2,1},\ s.t.\ \mathbf{X}_S = \mathbf{D}_S + \mathbf{E}_S, \quad (3)$$

$$\min_{\mathbf{Z}_S} rank(\mathbf{Z}_S),\ s.t.\ \mathbf{D}_S = \mathbf{D}_S\mathbf{Z}_S. \quad (4)$$

Instead of using the original LRR model (Eqn. 1), we choose the improved LRR method (Eqn. 2). The formulae (3) and (4) provide many insights for the improved LRR method. Firstly, we can see that the subproblem (Eqn. 3) is actually an RPCA with columnwise-sparse noise [28]. It implies that the proposed model firstly reduces the noise and outliers and adopts a cleaner dictionary $\mathbf{D}_S$, so it is more robust than traditional low-rank based methods. Secondly, this also helps improve robustness: In the early work [18], [29], Candés et al. showed that the performance of low rank pursuit degrades with increasing coherence of dictionary entries. In order to avoid this problem, recent work by Liu et al. [30] has shown that when the dictionary itself is low-rank, the LRR will be immune to dictionary coherence. So, the subproblem (Eqn. 3) not only reduces noise, but also eliminates coherence influence.

In the following, we will analyze the learning power of the LRR method with only partial data observed. This is consistent with the static learning stage of our method, since the dynamic data is yet to be seen.

In order to study the influence of the unobserved data, we split the data into two parts: $\mathbf{X} = [\mathbf{X}_S\ \mathbf{X}_W]$, where $\mathbf{X}_S$ represents the (static) training data, i.e., the partial observed data, while $\mathbf{X}_W$ is the unobserved, hidden data. In the following, we will prove that under some mild conditions, the true subspace membership can be revealed by using LRR with only the partial training data $\mathbf{X}_S$.

Theorem 1: Given data $\mathbf{X} = \mathbf{X}_0 + \mathbf{E} = [\mathbf{X}_S\ \mathbf{X}_W] + \mathbf{E}$, where $\mathbf{X}_0 \in \mathbb{R}^{d\times n}$ is of rank r and has incoherence parameter μ, where intuitively, incoherence indicates that each data point contains sufficient information of the subspace, $\mathbf{E}$ contains outliers, and $\mathbf{X}$ has RWD (Relatively Well-Definedness) parameter η [31][1]. When the size of training data $m \geq \frac{49(11+4\eta)^2\mu r}{324\eta^2+49(11+4\eta)^2\mu r}n$, the true subspace membership can be revealed by using LRR with only the partial training data $\mathbf{X}_S$.

Proof: As only partial training data is observed, we consider the following LRR problem for noise-free data

$$\min_{\mathbf{Z},\mathbf{E}} \|\mathbf{Z}\|_*,\ s.t.\ \mathbf{X}_S = [\mathbf{X}_S\ \mathbf{X}_W]\mathbf{Z} \quad (5)$$

where $\mathbf{Z} = [\mathbf{Z}_{S|W};\ \mathbf{Z}_{W|S}]$ with $\mathbf{Z}_{S|W}$ and $\mathbf{Z}_{W|S}$ corresponding to $\mathbf{X}_S$ and $\mathbf{X}_W$ respectively. We can see that the representation dictionary $\mathbf{A} = [\mathbf{X}_S\ \mathbf{X}_W]$ is always sufficient. According to Theorem 3.1 in [32], the minimization of this problem has a unique solution: $\mathbf{Z}^*_{S|W} = \mathbf{V}_S\mathbf{V}_S^T$ and $\mathbf{Z}^*_{W|S} = \mathbf{V}_W\mathbf{V}_S^T$, where $[\mathbf{X}_S\ \mathbf{X}_W] = \mathbf{U}\Sigma\mathbf{V}^T$ and $\mathbf{V} = [\mathbf{V}_S\ \mathbf{V}_W]$.

Furthermore, the relationship between the observed training data $\mathbf{X}_S$ and the unobserved data $\mathbf{X}_W$ can be further investigated. From [32] we obtain

$$\begin{aligned}\mathbf{X}_S &= [\mathbf{X}_S\ \mathbf{X}_W]\mathbf{Z}^* = \mathbf{X}_S\mathbf{Z}^*_{S|W} + \mathbf{X}_W\mathbf{Z}^*_{W|S}\\ &= \mathbf{X}_S\mathbf{Z}^*_{S|W} + \mathbf{X}_W\mathbf{V}_W\mathbf{V}_S^T\\ &= \mathbf{X}_S\mathbf{Z}^*_{S|W} + \mathbf{U}\Sigma\mathbf{V}_W^T\mathbf{V}_W\Sigma^{-1}\mathbf{U}^T\mathbf{X}_S\\ &= \mathbf{X}_S\mathbf{Z}^*_{S|W} + \mathbf{L}^*_{W|S}\mathbf{X}_S\end{aligned}$$

where $\mathbf{L}^*_{W|S} = \mathbf{U}\Sigma\mathbf{V}_W^T\mathbf{V}_W\Sigma^{-1}\mathbf{U}^T$. As we assume that both the training data $\mathbf{X}_S$ and the unobserved data $\mathbf{X}_W$ are sampled from the same subspaces with rank r, we can get $rank(\mathbf{Z}^*_{S|W}) \leq r$ and $rank(\mathbf{L}^*_{W|S}) \leq r$, which implies that both $\mathbf{Z}^*_{S|W}$ and $\mathbf{L}^*_{W|S}$ should be of low-rank. Therefore, the subspace membership can be revealed by minimizing

$$\begin{aligned}&\min_{\mathbf{Z}_{S|W},\mathbf{L}_{W|S}} \|\mathbf{Z}_{S|W}\|_* + \|\mathbf{L}_{W|S}\|_*,\\ &\quad s.t.\ \mathbf{X}_S = \mathbf{X}_S\mathbf{Z}_{S|W} + \mathbf{L}_{W|S}\mathbf{X}_S\end{aligned} \quad (6)$$

Following [32], suppose $(\mathbf{Z}^*_{S|W},\ \mathbf{L}^*_{W|S})$ are the minimizer of (6), then $\mathbf{Z}^*_{S|W}$ is an approximate recovery to $\mathbf{Z}_{S|W}$ in (5). Therefore, the true subspace membership $\mathbf{Z}_{S|W}$ can be revealed by using only the partial training data $\mathbf{X}_S$.

For corrupted data with outliers, the subspace membership $\mathbf{Z}_{S|W}$ can be solved by minimizing the following convex optimization problem

$$\begin{aligned}\min_{\mathbf{Z}_{S|W},\mathbf{L}_{W|S},E} &= \|\mathbf{Z}_{S|W}\|_* + \|\mathbf{L}_{W|S}\|_* + \lambda\|\mathbf{E}\|_{2,1},\\ &s.t.\ \mathbf{X}_S = \mathbf{X}_S\mathbf{Z}_{S|W} + \mathbf{L}_{W|S}\mathbf{X}_S + \mathbf{E}\end{aligned}$$

[1]The RWD parameter η should not be extremely small so as to guarantee the success of LRR, as detailed in [31].

From the theoretical analysis in [31] and [28], for the corrupted data with outliers $\mathbf{X} = [\mathbf{X}_S\ \mathbf{X}_W] + \mathbf{E}$, under some mild conditions, when the size of the training data m satisfies

$$m \geq \frac{49(11+4\eta)^2\mu r}{324\eta^2 + 49(11+4\eta)^2\mu r} n$$

the LRR model can reveal the subspace structure exactly, where η is the RWD parameter of $\mathbf{X}$, μ is the incoherence parameter and r is the rank of data $[\mathbf{X}_S\ \mathbf{X}_W]$.

In short, if the sample complexity m satisfies the above condition, the true subspace membership can be revealed by using LRR with only the partial training data $\mathbf{X}_S$. ■

The theorem above states that the subspace structure can be learned with partial data using LRR under certain conditions. In practice, static learning is only used as initialization in our online LRR framework, and our method provides an online solution that approximates the traditional LRR. Even for challenging real-world problems where the assumptions are not fully satisfied, our method can still work well. As we will demonstrate with extensive experiments in Section V, our online LRR achieves performance similar to and sometimes better than the batch LRR method, and significantly better than SLRR [8] where the subspace structure is extracted purely based on the partial training data. The influence of varying the size of the training data and the proportion of corrupted data points will also be discussed in Section V.

The subproblem (3) is a small-scale RPCA problem and can be solved efficiently by the augmented Lagrange multiplier method (ALM) proposed by Lin [33]. The subproblem (4) has a closed form solution, known as the shape interaction matrix [26]. Once the optimal $\mathbf{D}_S$ in Eqn. (3) is obtained, the optimal solution of problem (4) can be solved by $\mathbf{Z}_S = \mathbf{V}_S\mathbf{V}_S^T$, where $[\mathbf{U}_S, \mathbf{\Sigma}_S, \mathbf{V}_S]$ is the skinny singular value decomposition (SVD) of $\mathbf{D}_S$, which is readily available when solving Eqn. (3). Therefore, the computation complexity of subproblem (4) is only a matrix multiplication.

Based on basic assumptions at the beginning of this section, the low-rank component matrix $\mathbf{D}_S$ recovered by the training data $\mathbf{X}_S$ should cover the entire subspace, i.e., the intrinsic low-rank component of each data sample in the entire data space can be approximately linearly represented by the columns of $\mathbf{D}_S$.

B. Dynamic Updating

For most of the existing LRR methods, when l new data samples $\mathbf{X}_W \in \mathbb{R}^{d\times l}$ are added, they have to recompute the problem (3) for the entire data set $[\mathbf{X}_S\ \mathbf{X}_W] \in \mathbb{R}^{d\times(m+l)}$. This is computationally very expensive and conceptually unnecessary: The LRR result of previous data is thrown away, and for each dynamically added sample, the model (3) has to be computed repeatedly, which includes a time-consuming SVD computation. In this paper, we develop an online updating algorithm for dynamic data, which also works effectively for large-scale data, where a small subset of data is used in the static training stage, and the remaining samples can be seen as dynamically added data. The proposed dynamic updating method can extract the low-rank component matrix $\mathbf{D}_W$ incrementally for dynamically added data $\mathbf{X}_W$ based on the learning results on the training data $\mathbf{X}_S$. Furthermore, the low-rank representation matrix $\mathbf{Z}$ on the whole data $[\mathbf{X}_S\ \mathbf{X}_W]$ can also be updated incrementally without the need of repeatedly solving the complex SVD problem. The proposed method successfully avoids repeatedly solving complex rank minimization for incrementally added samples; the time-consuming rank optimization (Eqn. 3) only needs to be solved once in the static training step.

The online updating algorithm can be divided into the following two steps: updating of $\mathbf{D}_W$ and updating of $\mathbf{Z}$.

1) Updating of $\mathbf{D}_W$: Based on the assumptions in this paper and analysis from the static learning, the intrinsic low-rank component of each sample in the entire data space can be linearly represented by the column vectors of $\mathbf{D}_S$, apart from sparse noise. Therefore, for the dynamically added data samples $\mathbf{X}_W$, the low-rank component matrix $\mathbf{D}_W$ corresponding to subproblem (3) should be linearly represented using the basis from the column space of $\mathbf{D}_S$. Furthermore, based on the assumption that the data samples are drawn from independent subspaces, each data sample should only be represented by the basis vectors from the same subspace, which implies that each sample should be sparsely represented. Based on the analysis above, the low-rank component matrix $\mathbf{D}_W$ can be solved by sparse reconstruction as shown in the following Theorem 2.

Theorem 2: Let $[\mathbf{U}_S, \mathbf{\Sigma}_S, \mathbf{V}_S]$ be the skinny SVD of $\mathbf{D}_S$. For the dynamically added data $\mathbf{X}_W$, its low rank components $\mathbf{D}_W$ corresponding to subproblem (3) can be solved by

$$\begin{aligned} \min_{\mathbf{P},\mathbf{E}_W} \|\mathbf{E}_W\|_1,\ s.t.\ \mathbf{X}_W &= \mathbf{D}_W + \mathbf{E}_W \\ and\quad \mathbf{D}_W &= \mathbf{U}_S\mathbf{P}, \end{aligned} \tag{7}$$

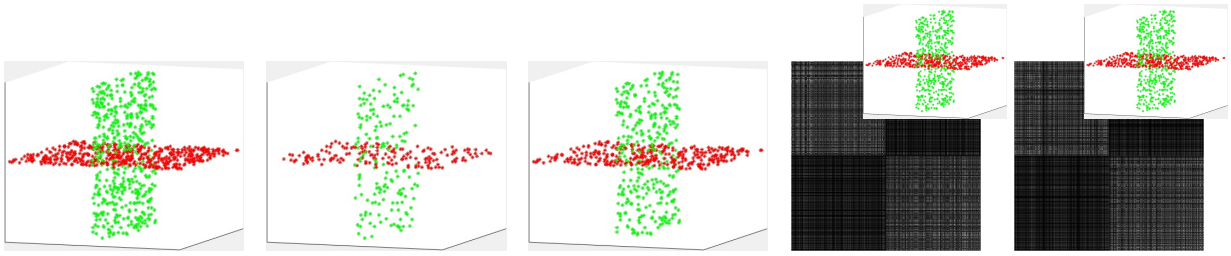
where $\mathbf{P}$ contains combination weights to recover $\mathbf{D}$ using the basis $\mathbf{U}_S$. This means $[\mathbf{D}_S\ \mathbf{D}_W]$ contains the low-rank components of the whole dataset $[\mathbf{X}_S\ \mathbf{X}_W]$.

Proof: Based on the assumption above that the low-rank components $\mathbf{D}_W$ of new samples $\mathbf{X}_W$ can be represented as linear combinations of column vectors of $\mathbf{D}_S$ as the basis, it is obvious that the rank of $[\mathbf{D}_S\ \mathbf{D}_W] = [\mathbf{U}_S\mathbf{\Sigma}_S\mathbf{V}_S\ \mathbf{U}_S\mathbf{P}]$ should not be larger than the rank of $\mathbf{D}_S$. On the other hand, since $\mathbf{D}_S$ is the optimal low-rank solution to the problem (3) with the training samples $\mathbf{X}_S$ (a subproblem of $[\mathbf{D}_S\ \mathbf{D}_W]$), it is not possible to find solutions with lower rank than $[\mathbf{D}_S\ \mathbf{D}_W]$ for problem (3) with data $[\mathbf{X}_S\ \mathbf{X}_W]$. Combining formulae (3) and (7), we can reach the conclusion that $\mathbf{D} = [\mathbf{D}_S\ \mathbf{D}_W]$ and $\mathbf{E} = [\mathbf{E}_S\ \mathbf{E}_W]$ form an optimal solution to the following problem:

$$\min_{\mathbf{D},\mathbf{E}} \|\mathbf{D}\|_* + \lambda\|\mathbf{E}\|_{2,1},\ s.t.\ [\mathbf{X}_S\ \mathbf{X}_W] = \mathbf{D} + \mathbf{E}. \tag{8}$$

As the optimization problem (8) is convex, $[\mathbf{D}_S\ \mathbf{D}_W]$ should also be the unique solution. Therefore, under the hypothesis that the training data is sufficient to cover the subspace, $[\mathbf{D}_S\ \mathbf{D}_W]$ corresponds to the low-rank components of the whole data $[\mathbf{X}_S\ \mathbf{X}_W]$ with new samples added. ■

The problem (7) can be efficiently solved by Alternating Direction Method (ADM) [33], by minimizing the following



Original data (a) | Partial training data (b) | Updated data online (c) | LRR(Acc = 0.9520) (d) | OLRR(Acc = 0.9528) (e)

Fig. 2. Illustration of steps and results of the proposed online LRR learning algorithm, compared with the traditional batch LRR.

augmented Lagrangian function w.r.t. $\mathbf{E}_W$ and $\mathbf{P}$ with other variables fixed:

$$\|\mathbf{E}_W\|_1 + \langle \mathbf{v}, \mathbf{X}_W - \mathbf{U}_S\mathbf{P} - \mathbf{E}_W \rangle + \frac{\beta}{2}\|\mathbf{X}_W - \mathbf{U}_S\mathbf{P} - \mathbf{E}_W\|^2, \tag{9}$$

where β is the penalty parameter. The implementation of the ADM algorithm is shown in Algorithm 1, which is similar to [34].

Algorithm 1 Solving the optimization model (9) using Alternating Direction Method (ADM).

Input:
Dynamically added data $\mathbf{X}_W$, subspace bases $\mathbf{U}_S$.
Initialize: $\mathbf{E}_W^0 = \mathbf{0}, \mathbf{P}^0 = \mathbf{0}, \mathbf{v}_0 = \mathbf{0}, \bar{\beta} \gg \beta_0 > 0, \rho > 1, i = 0$. $\bar{\beta}, \beta_0$, and ρ are chosen constants[2].

1: **repeat**
2: $\mathbf{E}_W^{i+1} = S_{\beta_i^{-1}}(\mathbf{X}_W - \mathbf{U}_S\mathbf{P} + v_i/\beta_i)$, where $S_{\beta_i^{-1}}$ is the soft-thresholding operator defined as $S_{\beta_i^{-1}}(a) = sign(a)\max(0, |a| - \beta_i^{-1})$;
3: $\mathbf{P}^{i+1} = \mathbf{U}_S^T(\mathbf{X}_W - \mathbf{E}_W^{i+1} + \mathbf{v}_i/\beta_i)$;
4: $\mathbf{v}_{i+1} = \mathbf{v}_i + \beta_i(\mathbf{X}_W - \mathbf{U}_S\mathbf{P}^{i+1} - \mathbf{E}_W^{i+1})$;
5: $\beta_{i+1} = \min(\rho\beta_i, \bar{\beta})$;
6: $i \leftarrow i + 1$.
7: **until** convergence

Output: Return the optimal solution $\{\mathbf{P}^*, \mathbf{E}^*\}$

For dynamic clustering, or when large-scale data is processed, it is prohibitively slow to recompute the model (3) each time when new samples are given. Note that in real-world scenarios when more and more new data samples are incrementally added, traditional LRR methods will need to solve increasingly large problems, whereas for our approach the time complexity is proportional to the *newly added samples*, not any samples previously added.

2) Updating of $\mathbf{Z}$*:* According to the formula (4), the low-rank representation matrix $\mathbf{Z}$ can be obtained explicitly by firstly solving the SVD of $[\mathbf{D}_S\ \mathbf{D}_W] = \mathbf{U}\mathbf{\Sigma}\mathbf{V}$, and then working out $\mathbf{Z} = \mathbf{V}\mathbf{V}^T$, which is also known as the Shape Interaction Matrix (SIM) of $[\mathbf{D}_S\ \mathbf{D}_W]$. However, the computational complexity of the SVD of $[\mathbf{D}_S\ \mathbf{D}_W]$ is extremely high for large-scale data. Instead of recomputing the SVD of $[\mathbf{D}_S\ \mathbf{D}_W]$, we adopt the online Sequential Karhunen-Loeve (SKL) [27] algorithm, which incrementally updates the eigenbasis with dynamically added data. Given $\mathbf{U}_S$ and $\mathbf{\Sigma}_S$ from SVD of $\mathbf{D}_S$, which is already available when solving (Eqn. (3) in the previous step, SVD of $[\mathbf{D}_S\ \mathbf{D}_W]$ can be computed efficiently using the SKL algorithm (see Algorithm 2).

[2]In most of the experiments of this paper, the parameters are chosen as $\beta_0 = 2/mean(|X_W|)$, $\rho = 1.05$.

Algorithm 2 SKL Algorithm for Online SVD of $[\mathbf{D}_S\ \mathbf{D}_W]$.

Input: SVD of matrix $\mathbf{D}_S$: $[\mathbf{U}_S, \mathbf{\Sigma}_S, \mathbf{V}_S] = SVD(\mathbf{D}_S)$, and the learned low-rank matrix $\mathbf{D}_W$

1: Obtain $\mathbf{Q}$ and $\mathbf{R}$ by taking the QR decomposition of $[\mathbf{U}_S\mathbf{\Sigma}_S\ \mathbf{D}_W] : \mathbf{Q}\mathbf{R} = QR([\mathbf{U}_S\mathbf{\Sigma}_S\ \mathbf{D}_W])$. Note that the matrix $\mathbf{U}_S\mathbf{\Sigma}_S$ is already column orthogonal, so the QR decomposition can be performed on the columns of $\mathbf{D}_W$ only.
2: Compute the SVD of $\mathbf{R}$: $\widetilde{\mathbf{U}}\widetilde{\mathbf{\Sigma}}\widetilde{\mathbf{V}}^T = SVD(\mathbf{R})$. Only the singular values bigger than 0 are kept.
3: The SVD of $[\mathbf{D}_S\ \mathbf{D}_W]$ can be obtained as $\mathbf{U} = \mathbf{Q}\widetilde{\mathbf{U}}$, $\mathbf{\Sigma} = \widetilde{\mathbf{\Sigma}}$, $\mathbf{V}^T = \mathbf{\Sigma}^{-1}\mathbf{U}^T[\mathbf{D}_S\ \mathbf{D}_W]$.

Output: Output the SVD of $[\mathbf{D}_S\ \mathbf{D}_W]$: $[\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}]$.

Finally, the low-rank representation matrix $\mathbf{Z}$ for the whole data $[\mathbf{X}_S\ \mathbf{X}_W]$ can be obtained explicitly using the shape interaction matrix

$$\mathbf{Z} = \mathbf{V}\mathbf{V}^T. \tag{10}$$

The LRR matrix $\mathbf{Z}$ obtained using our algorithm has nice properties, as described in Theorem 3 below.

Theorem 3: The global low-rank representation matrix $\mathbf{Z}$ obtained by the proposed online LRR subspace learning algorithm is guaranteed to be symmetric and have block-diagonal structure.

This can be proved in a similar way as [26].

3) Computational complexity analysis: Following Algorithm 1, the computational complexity of each iteration of problem (9) is $O(drl)$, where d is the dimension of data, l is the number of incrementally updated samples in $\mathbf{X}_W$ and r ($r \ll l$) is the rank of the column space. In contrast, traditional LRR methods need to recompute the model (3) for the entire data $[\mathbf{X}_S\ \mathbf{X}_W]$ and the computational complexity for each iteration is $O(dr(m+l)^2)$. From Algorithm 2, we can see that by using the SKL algorithm, the complexity is reduced dramatically from $O((m+l)^2)$ to $O(m+l)$, i.e. proportional to the number of sample points.

The online LRR subspace learning algorithm proposed in this paper can be summarized in Algorithm 3.

Algorithm 3 Online Low-Rank Representation Classification Algorithm for Dynamic Clustering.

Input: Initial static learning data set $\mathbf{X}_S = [\mathbf{X}_S^1\ \mathbf{X}_S^2\ ...\ \mathbf{X}_S^c] \in \mathbb{R}^{d\times m}$, dynamically added samples $\mathbf{X}_W \in \mathbb{R}^{d\times l}$

1: **Stage 1: Static learning.** Solve the subproblem (3) to obtain the low rank component $\mathbf{D}_S$ of training set $\mathbf{X}_S$.

2: **Stage 2: Dynamic updating.**

- Updating of $\mathbf{D}_W$: Given the newly added sample $\mathbf{X}_W$, find its approximation $\mathbf{D}_W$ in the column space by solving problem (9) (Algorithm 1).
- Updating of $\mathbf{Z}$: The final low rank representation matrix $\mathbf{Z}$ for $[\mathbf{X}_S\ \mathbf{X}_W]$ can be solved using shape interaction matrix with the SKL algorithm for online SVD (Algorithm 2).

Output: The global low-rank representation matrix $\mathbf{Z}$.

V. Experiments

In this section, we evaluate the performance of the online low-rank representation subspace learning algorithm proposed in this paper using extensive experiments on both synthetic data and public databases. We compare our method with state-of-the-art methods on several evaluation metrics.

Algorithms. For the experiment on synthetic data, we compare the proposed algorithm only with the original LRR model [1] because the purpose of this experiment is to demonstrate the correctness of the online learning method. For the task of subspace recovery, we compare our algorithm with three typical representation-based subspace learning methods, LRR [1], RSI [26] and OLRSC (Online Low-Rank Subspace Clustering) [9]. The first two are batch methods, whereas OLRSC is a state-of-the-art online LRR method. For the task of subspace clustering, in addition to LRR, RSI, OLRSC, we also compare with other online learning frameworks for representation-based subspace clustering, including SLRR (Scalable Low-rank Representation) [8] and SSSC (Scalable Sparse Subspace Clustering) [8].

Evaluation Metrics. For the task of subspace recovery, we evaluate the fitness of the recovered subspaces (with each column being normalized) and the ground truth by the Expressed Variance (EV) [35] which is widely used in the literature:

$$EV(\mathbf{D},\mathbf{L}) = Tr(\mathbf{D}\mathbf{D}^T\mathbf{L}\mathbf{L}^T)/Tr(\mathbf{L}\mathbf{L}^T)$$

where $\mathbf{D}$ and $\mathbf{L}$ are the recovered subspace and the ground truth subspace respectively, and $Tr(\cdot)$ is the trace of the matrix. The value of EV ranges between 0 and 1, and a higher value means better recovery. For the task of subspace clustering, standard normalized mutual information (NMI) [36] and clustering accuracy [37] are used as metrics for evaluation.

The following describes the experiments and results.

A. Synthetic data

The experiment on synthetic data is designed for two purposes. Firstly, it is useful to evaluate the correctness of the proposed method, i.e., the low-rank components and the global affinity graph learned incrementally should be as accurate as those obtained by batch LRR methods while the computational complexity is reduced dramatically. Secondly, the experiment on synthetic data will give valuable insights for choosing suitable parameters. Synthetic data is noise free and the data generation process can be fully controlled.

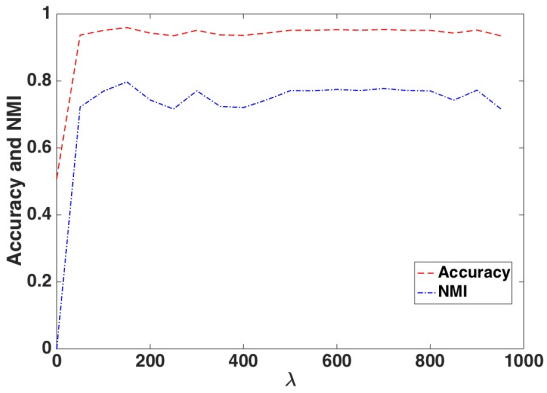


Fig. 3. The accuracy and NMI according to different values of parameter λ.

TABLE I
Accuracy and running time on synthetic data

	Accuracy	NMI	running time(s)
LRR [1]	0.9520	0.6775	1.7154
Ours	0.9528	0.7324	0.1706 (0.0745 + 0.0961)

In this experiment, we generate two 3-dimensional independent subspaces, i.e., two planes perpendicular to each other, and 500 points are sampled from each subspace (plane) to form the synthetic data $\mathbf{X} = [\mathbf{X}_1\ \mathbf{X}_2] \in \mathbb{R}^{3\times 1000}$, as shown in Fig. 2 (a). As the main purpose of the experiment on synthetic data is to evaluate the correctness of the proposed online method, we just compare our proposed algorithm with the original batch LRR method [1], which is solved by an accelerated augmented Lagrange multiplier (ALM) method. We compare these two methods using the following three metrics: running time, clustering accuracy and normalized mutual information (NMI).

LRR [1] is performed on the whole dataset $\mathbf{X}$, while for the proposed online LRR method, half of data points from each subspace are randomly chosen as the static training data (Fig. 2(b)), and the rest are treated as dynamic samples added later on. The parameter λ in both LRR model and our model (3) is set to 100. The experimental results are shown in Fig. 2. The reconstructed data points obtained using our online algorithm are shown in (c). (d) and (e) show corresponding low-rank data and the learned affinity matrices obtained using LRR [1] and our proposed method (OLRR), respectively. The running times, clustering accuracy and NMI are shown in Table I. For our method, we also show the breakdown of the running time into static learning and online update stages.

The experimental results are in line with our expectation. From the visualization of the recovered subspace, it can be seen that our proposed method can learn the subspace as well as the batch method [1], with even better clustering performance, while reducing the running time dramatically

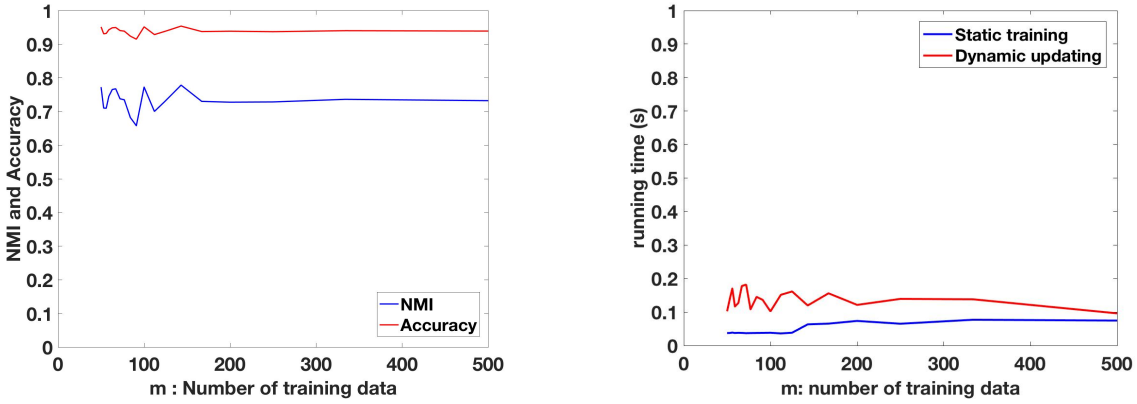


Fig. 4. The accuracy and running times with different numbers of training data.

(over 10 times faster than [1]). Note that our method has further benefit of being able to process online data. Our method can efficiently handle cases where data samples are incrementally added, in which case batch methods would be prohibitively expensive.

Influence of Parameter Settings. Another purpose of the experiment on synthetic data is to guide the parameter settings. There are 2 major parameters for our proposed algorithm, λ in formula (3) and the number of training data m in Section IV-A.

The choice of parameter λ depends on the data distribution. We experimented with λ from 0.0001 to 1000, and computed the corresponding clustering accuracy and NMI, which are shown in Fig. 3. We can see that for the synthetic data, when $\lambda > 20$ the performance is consistently good. Intuitively, as the synthetic data is generated based on an accurate distribution, the representation error should be as small as zero, and therefore a bigger λ is needed to penalize sparse errors.

Another significant parameter is the number of training data m. The choice of parameter m determines whether the intrinsic subspace structure can be learned accurately, which is key to the performance of the proposed method. In order to understand the influence of parameter m, we choose m from 50 to 500, and m samples are randomly chosen with equal chance from each subspace to compose the training data.

Each experiment is conducted 10 folds and the average accuracy and run times are reported in Fig. 4. We can see that our method performs consistently well, and when $m > 200$, the performance of the proposed method is more stable. As discussed in Section IV-A, according to the general learning theory of RPCA, when the sampling rate is sufficiently high, the low-rank component can be exactly extracted. From Fig. 4 we can see that 200 data points are sufficient for this problem. The running times of static learning step and dynamic updating process are shown in Fig. 4 (right). With increasing m, the running time of static learning increases whereas that of the dynamic update reduces.

B. Subspace recovery

In this section, we evaluate the performance of the proposed online LRR learning method for subspace recovery, which aims at recovering original data from the learned subspace structure. For intuitive visualization and more convincing evaluation of real-world performance, we adopt a standard handwritten digit benchmark USPS. The USPS handwritten digit database[3] is shown to roughly reside in a low-dimensional subspace. The USPS database contains 9298 digit images of "0" through "9", each of which is of size 16×16 pixels, with 256 gray levels per pixel. In the experiment, each image is represented by a 256-dimensional vector. Fig. 6 (top row) shows some original sample images from the database.

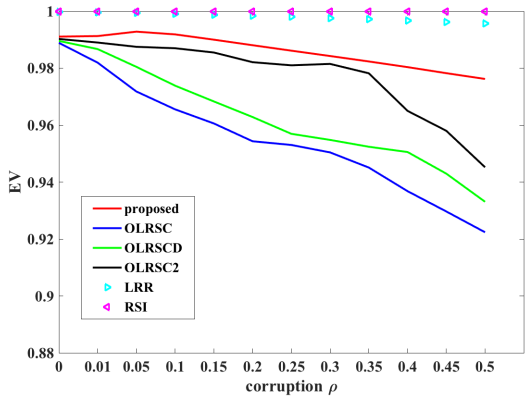


Fig. 5. The EV curves of different algorithms with varying levels of corruption.

In this experiment, we compare the performance of the proposed online LRR learning method against LRR, RSI and OLRSC from the aspects of recovery performance and running time. In order to evaluate the robustness of the proposed algorithm, different levels of sample-specific corruption are added,

$$\tilde{\mathbf{X}} = \mathbf{X} + \mathbf{E}$$

where $\mathbf{X}$ is the ground truth USPS data, and $\mathbf{E}$ is the sample-specified errors whose ρ fraction of entries are non-zero and follow an i.i.d. uniform distribution over $[-1, 1]$. In this paper,

[3]http://www.gaussianprocess.org/gpml/data/

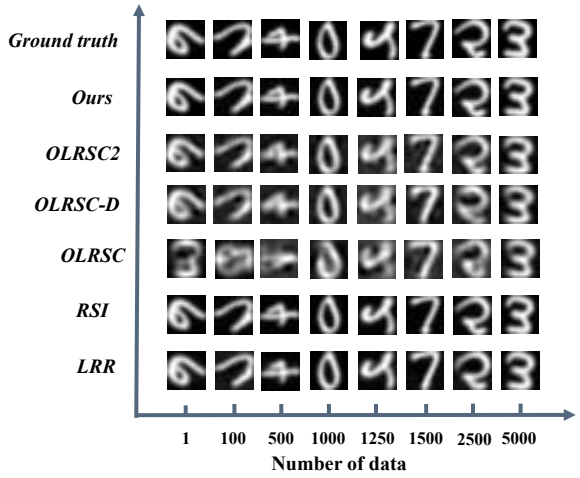


Fig. 6. Ground truth images from the USPS database and reconstructed images by different methods.

TABLE II
THE AVERAGE RUNNING TIME OF DIFFERENT METHODS ON USPS DATA.

Method	LRR [1]	RSI [26]	OLRSC [9]	OLRSC-D [9]	OLRSC2 [9]	OURS
Time(s)	258.89	45.36	15.02	21.06	33.55	4.76

we set ρ as 0, 0.01 and from 0.05 to 0.5 with step size 0.05. In this experiment, unless specified otherwise, 1/3 samples of each subspace are chosen for static training.

OLRSC [9] is based on a stochastic optimization process which can work without a static learning stage. However, due to the stochastic nature, the performance at the beginning when only a small number of samples are processed is poor. In order to improve the performance, following [9] the stochastic optimization has to be performed more than once on the whole data, resulting in high computational complexity, which is not suitable for the dynamic clustering problem. In addition, as OLRSC [9] is designed for unsupervised learning, for fairness, we propose an improved strategy of applying OLRSC [9] for dynamic problems (hereafter referred to as OLRSC-D) by which the proper basis is first learned by the stochastic optimization on a small set of the whole dataset, of the same size as our static training set, and then the remaining data can be learned online based on the learned basis. We compare original one-pass OLRSC, OLRSC-D and the 2-fold OLRSC (referred to as OLRSC2) in the experiments.

Each algorithm is conducted 10 folds, the average EV values are shown in Fig. 5 and the average running times are reported in Table II. From Fig. 5, we can see that the basic LRR model can always obtain the exact recovery. RSI can also achieve robust performance with the average EV values larger than 0.999. For clean data, the proposed method achieves similar performance as OLRSC and its variants. However, with an increasing level of corruption, the EV values of OLRSC methods drop rapidly while our method can maintain robust performance.

In order to intuitively illustrate the recovery results, we present the reconstructed images for the noise-free case (i.e., $\rho = 0$) and the ground truth images in Fig. 6. We can see

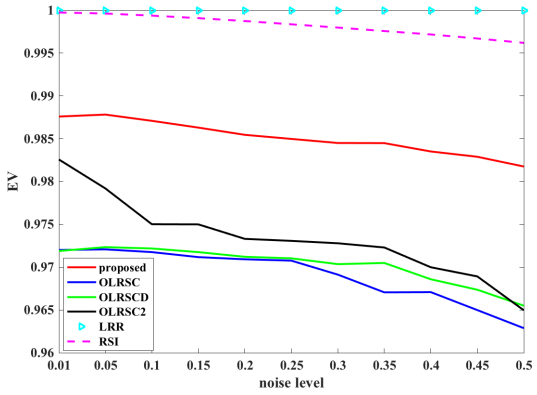


Fig. 7. The EV curves of different algorithms against varying levels of Laplacian noise.

that the results by OLRSC are generally poor for the first 1000 samples. Along with the increasing number of training samples, the dictionary learned by the stochastic optimization adopted in OLRSC is becoming more and more accurate, leading to better reconstructed results. For OLRSC-D, the basis dictionary is firstly learned on the training dataset, which improves the performance at the early stage of stochastic optimization. By using a repeated learning process, the recovery results of OLRSC2 are robust. However, from Table II, it is clear that the running time of the repeated OLRSC on the whole data is 2 times more than that of OLRSC, and over 6 times more than our method. Note that our method is an online method that produces recovery results with incrementally added samples, which is a significant advantage compared with batch methods (SSC and LRR). Since the average performance of OLRSC-D is similar to OLRSC and between OLRSC and OLRSC2, only the 1-fold OLRSC and the 2-fold OLRSC2 are performed in the rest of this paper.

In Fig. 7, Laplacian noise of different levels is added and the average EV curves are reported. It is noted that the ℓ_1 norm is adopted in all algorithms for the Laplacian noise regularization. It can be seen that the batch methods LRR and RSI achieve best performance. The performance of OLRSC and its variants drops rapidly while our method can also maintain robust performance against Laplacian noise.

C. Subspace Clustering

In this section, we evaluate the performance of the proposed method on the task of subspace clustering, which aims to split the data samples into disjoint clusters based on subspace structure. 5 real-world databases varying from small scale to large scale are chosen as shown in Table III. For computational efficiency, the data dimensions of extended YaleB and MNIST are first reduced by Principal Component Analysis (PCA) to retain 98% energy. For MNIST, 20,000 samples are randomly selected to form the MNIST20K dataset since the spectral clustering is time and memory consuming for the entire database.

In addition to SSC, LRR, RSI, OLRSC and OLRSC2, we also compare our method with other related online frameworks

TABLE III
DATABASES FOR SUBSPACE CLUSTERING

Database	♯ samples	Dim. of features	♯ classes
extended YaleB	2414	114	38
Satimg	6435	36	6
USPS	9298	256	10
Pendigits	10992	16	10
MNIST20K	20000	200	10

TABLE IV
SETTINGS OF λ FOR DIFFERENT ALGORITHMS

Database	SSC [14]	LRR [1]	RSI [26]	OLRSC [9]	SSSC [8]	SLRR [8]	Ours
YaleB	0.5	0.5	0.5	λ_2 = 2.75e-4	0.5	3.1	0.45
Satimg	–	1e-6	1e-6	λ_2 = 5e-7	5e-6	1e-6	5e-6
USPS	0.5	0.5	0.5	λ_2 = 0.0104	0.5	3.1	0.2
Pendigits	–	–	–	λ_2 = 0.02	0.5	3.1	1000
MNIST20K	–	–	–	λ_2 = 5e-5	0.05	0.01	0.001

designed for subspace clustering, including scalable sparse subspace clustering (SSSC) and scalable low-rank representation (SLRR). Note that SSSC and SLRR do not solve the original subspace clustering problem exactly, but instead only cluster in-sample data and "clustering" of out-of-sample data is solved by a classification process. For the remaining SSC, LRR, OLRSC and the proposed method, the spectral clustering method [38] is used based on the global representation matrix learned by each algorithm. The global representation matrix $\mathbf{Z}$ computed by RSI and the proposed method is guaranteed to be symmetric, so it can be directly used for spectral clustering. For SSC, LRR and OLRSC, $\hat{\mathbf{Z}} = |\mathbf{Z}| + |\mathbf{Z}'|$ is used to symmetrize the matrix.

Parameter Settings. There is a common parameter λ in all of the compared algorithms, which is used to balance the data fidelity term and the regularization term. Different databases may require different choices of λ to work most effectively. For fair comparison, λ is tuned for all the methods such that the best performance is obtained. The settings used are reported in Table IV[4]. For online methods including SLRR, SSSC and the proposed algorithm, there is another important parameter m which refers to the number of static training samples. In the following experiment, m is set as 1/3 of the whole data points. Extra experiments were performed on Pendigits database to evaluate the influence of parameter m.

Performance Comparison. We report the clustering accuracy, NMI and the running times of these methods in Table V. Due to the high computational complexity of batch methods on large-scale data, we are unable to obtain results of the SSC and LRR methods for Pendigits and MNIST databases within reasonable amount of time. It can be seen that in most cases our method outperforms the other methods in terms of clustering accuracy and/or running times. For USPS and Pendigits, our method achieves the best performance among online methods, and obtains the highest NMI score on the extended YaleB database. Although SSSC achieves higher accuracy on MNIST20K, its running time is 87 times more than our method, thus not practical for large-scale data. Our method reduces the running times dramatically compared with batch methods. For example, on the USPS database, the traditional LRR takes 258.89s while our method just needs 4.76s. The running time of our method also outperforms the other online methods in majority of cases. Although OLRSC is faster than our method on YaleB and Satimg, its clustering accuracy is poor. In order to improve the performance, a repeated process has to be conducted (referred to OLRSC2 in the table), which is significantly slower.

The influence of parameter m. The proposed method shares the similar assumption as SLRR [8], i.e., the subspace structure of the whole data space can be learned from partial training data. Therefore, the parameter m which refers to the number of training data plays an important role for the learning performance. In this experiment, the same m is set for the proposed method and SLRR [8], and the clustering accuracy and NMI score of both methods on the Pendigits database are reported in Fig. 8. The parameter m is set as $(1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000)$, which implies that for each object $m/10$ samples are chosen as the training data. For each value the experiments are conducted 10 folds and the average accuracy and NMI scores are reported. We can see that the accuracies of both methods are in the range of (0.7, 0.85), which is in line with the basic assumption that under some mild conditions the subspace structure of the whole data space can be learned from partial training data. However, since SLRR [8] only uses in-sample data to learn the subspace, whereas our method updates the subspace structure incrementally with online data, the proposed method is much more robust and achieves significantly better performance when the size of the static training data is smaller.

Robustness to Noise. Finally, we evaluate the robustness of different methods to noise. We randomly add different levels of noise to the original data. With half of the data contaminated by 5%, 10% and 15% Gaussian noise, the clustering results are shown in Table VI. The performances of OLRSC and SLRR suffer a sharp decline when the noise is heavy (e.g. 15%). For OLRSC, as the basis of the subspace is learned by stochastic optimization, when the data is contaminated, the misleading dictionary basis will be pursued, resulting in poor performance. Furthermore, if the learning process is conducted repeatedly, the error will be propagated and accumulated, leading to even worse performance (OLRSC2 vs. OLRSC). For SLRR, since the clustering results are obtained by classification of out-of-sample based on the learned subspace structure from the in-sample data, the clustering performance is sensitive to noise. In contrast, our proposed method can learn reliable subspace structure, which is robust to noise. Furthermore, the proposed method incorporates an RPCA-type preprocessing (3), which leads to a better performance.

VI. CONCLUSION

In this paper, an online LRR subspace learning method for large-scale and dynamic data is proposed. Compared with the traditional LRR model, the proposed algorithm only needs to compute the complex rank minimization once, and for each dynamically added sample, the global low-rank representation

[4]For OLRSC, λ_2 is tuned as shown in the table, and other parameters are fixed, $\lambda_1 = 1$, $\lambda_3 = 1/\sqrt{n}$, where n is the number of data points.

TABLE V
RUNNING TIME AND CLUSTERING ACCURACY OF DIFFERENT ALGORITHMS.

Database	SSC [14]			SSSC [8]			LRR [1]			SLRR [8]		
	Accuracy	NMI	time	Accuracy	NMI	time	Accuracy	NMI	time	Accuracy	NMI	time
YaleB	0.5898	0.6625	64.68	0.5676	0.6059	128.23	**0.7365**	0.7756	51.2212	0.6920	0.7460	32.21
Satimg	0.6977	0.7015	136.51	0.6524	0.5786	5.03	**0.7906**	0.7364	26.5171	0.6476	0.3936	28.19
USPS	0.7084	0.7266	2231.6	0.6900	0.6536	70.06	0.6342	0.5314	258.89	0.6890	**0.7406**	26.80
Pendigits	–	–	–	0.8131	0.7141	17.23	–	–	–	0.8021	0.7131	19.2541
MNIST20K	–	–	–	**0.6259**	**0.6205**	2074.11	–	–	–	0.5694	0.5546	3718.04

OLRSC [9]			OLRSC2 [9]			RSI [26]			Ours		
Accuracy	NMI	time	Accuracy	NMI	time	Accuracy	NMI	time	Accuracy	NMI	time
0.6332	0.5848	**5.05**	0.7052	0.7043	9.0924	0.7411	0.7831	9.7870	0.7125	**0.7845**	7.8341
0.6016	0.5121	**8.3626**	0.6542	0.5301	13.5313	0.7863	**0.7412**	10.3971	0.6737	0.5710	8.9194
0.6395	0.5989	15.02	0.7327	0.7041	33.55	0.6985	0.6772	45.36	**0.7497**	0.7105	**4.76**
0.5612	0.5435	12.89	0.7136	0.6264	19.83	–	–	–	**0.8195**	**0.7307**	**2.6197**
0.5114	0.4871	81.93	0.5774	0.5543	128.33	–	–	–	0.6225	0.5851	**23.75**

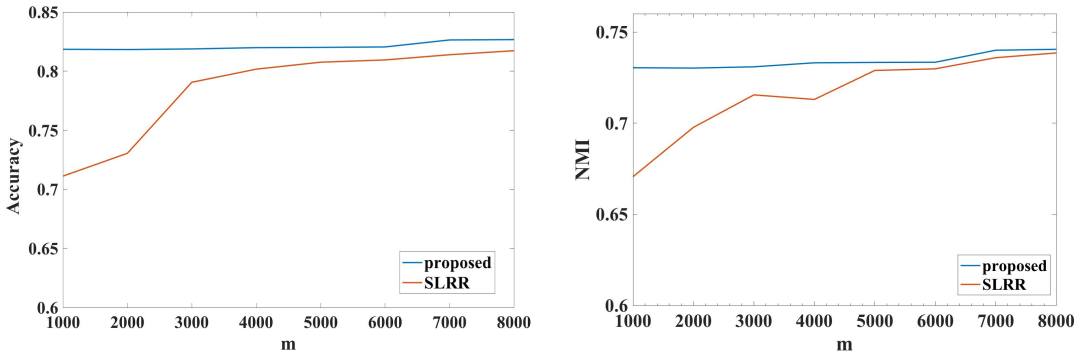


Fig. 8. The accuracy and NMI curves for the proposed method and SLRR [8] with varying parameter m.

TABLE VI
THE CLUSTERING ACCURACY OF DIFFERENT ALGORITHMS ON PENDIGITS WITH DIFFERENT NOISE LEVELS.

Noise	OLRSC [9]		OLRSC2 [9]		SSSC [8]		SLRR [8]		Ours	
	Accuracy	NMI	Accuracy	NMI	Accuracy	NMI	Accuracy	NMI	Accuracy	NMI
5%	0.5609	0.5193	0.6108	0.5634	0.6994	0.6949	0.6904	0.6720	0.7927	0.7025
10%	0.5293	0.5022	0.5744	0.5265	0.6207	0.6585	0.6689	0.6038	0.7748	0.6866
15%	0.4760	0.3744	0.3515	0.2783	0.4940	0.5083	0.5238	0.4234	0.7545	0.6797

matrix can be computed incrementally based on the existing learned results efficiently. Extensive experiments on both synthetic and real-world benchmark data for both subspace recovery and clustering tasks verify that the proposed online LRR algorithm can exploit the intrinsic subspace structure as accurately as traditional LRR while reducing the computational complexity dramatically. Our method is naturally a two-stage algorithm. In the future, we would like to exploit an end-to-end approach to further improve the solution. Handling large-scale, dynamic data is particularly demanded when processing temporal data streams and Internet data, and we would like to investigate further applications of the proposed technique.

ACKNOWLEDGEMENTS

We would like to thank the authors of [1], [8], [9] for providing their source code. Bo Li is partially funded by natural science foundation of China (NSFC) (61562062, 61762064, 61262050), Risheng Liu is partially funded by NSFC (61672125, 61300086, and 61632019), the Fundamental Research Funds for the Central Universities (DUT15QY15) and the Hong Kong Scholar Program (No. XJ2015008). Juejie Cao is funded by NSFC 61363048 and the Fundamental Research Funds for the Central Universities DUT16QY02. Xiuping Liu is funded by NSFC (6137014361432003) and Jie Zhang is funded by NSFC 61702245.

REFERENCES

[1] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 171–184, 2013.

[2] X. Shen and Y. Wu, "A unified approach to salient object detection via low rank matrix recovery," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 853–860.

[3] B. Cheng, G. Liu, J. Wang, Z. Huang, and S. Yan, "Multi-task low-rank affinity pursuit for image segmentation," in *International Conference on Computer Vision*, 2011, pp. 2439–2446.

[4] B. Li, C. Lu, Z. Wen, C. Leng, and X. Liu, "Locality-constrained nonnegative robust shape interaction subspace clustering and its applications," *Digital Signal Processing*, vol. 60, pp. 113–121, 2017.

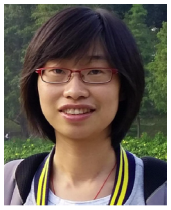
[5] X. Guo, X. Wang, L. Yang, X. Cao, and Y. Ma, "Robust foreground detection using smoothness and arbitrariness constraints," in *European Conference on Computer Vision*. Springer, 2014, pp. 535–550.

[6] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Low-rank sparse learning for robust visual tracking," in *European Conference on Computer Vision*. Springer, 2012, pp. 470–484.
[7] H. Mobahi, Z. Zhou, A. Y. Yang, and Y. Ma, "Holistic 3d reconstruction of urban structures from low-rank textures," in *IEEE International Conference on Computer Vision Workshops*. IEEE, 2011, pp. 593–600.
[8] X. Peng, H. Tang, L. Zhang, Z. Yi, and S. Xiao, "A unified framework for representation-based subspace clustering of out-of-sample and large-scale data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 12, pp. 2499–2512, 2016.
[9] J. Shen, P. Li, and H. Xu, "Online low-rank subspace clustering by basis dictionary pursuit," in *International Conference on Machine Learning*, 2016, pp. 622–631.
[10] X. Peng, C. Lu, Z. Yi, and H. Tang, "Connections between nuclear-norm and Frobenius-norm-based representations," *IEEE Transactions on Neural Networks and Learning Systems*, 2016.
[11] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, 2003.
[12] W. Dong, C. Moses, and K. Li, "Efficient k-nearest neighbor graph construction for generic similarity measures," in *International Conference on World Wide Web*. ACM, 2011, pp. 577–586.
[13] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.
[14] 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.
[15] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2009.
[16] B. Li, F. Zhao, Z. Su, X. Liang, Y.-K. Lai, and P. L. Rosin, "Example-based image colorization using locality consistent sparse representation," *IEEE Transactions on Image Processing*, vol. 26, no. 11, 2017.
[17] 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.
[18] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of the ACM (JACM)*, vol. 58, no. 3, p. 11, 2011.
[19] X. Peng, S. Xiao, J. Feng, W.-Y. Yau, and Z. Yi, "Deep subspace clustering with sparsity prior," in *International Joint Conference on Artificial Intelligence*, 2016.
[20] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma, "Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization," in *Advances in Neural Information Processing Systems*, 2009, pp. 2080–2088.
[21] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma, "RASL: Robust alignment by sparse and low-rank decomposition for linearly correlated images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2233–2246, 2012.
[22] Z. Zhang, A. Ganesh, X. Liang, and Y. Ma, "Tilt: Transform invariant low-rank textures," *International Journal of Computer Vision*, vol. 99, no. 1, pp. 1–24, 2012.
[23] W. t. Tan, G. Cheung, and Y. Ma, "Face recovery in conference video streaming using robust principal component analysis," in *IEEE International Conference on Image Processing*, 2011, pp. 3225–3228.
[24] T. Zhang, S. Liu, N. Ahuja, M.-H. Yang, and B. Ghanem, "Robust visual tracking via consistent low-rank sparse learning," *International Journal of Computer Vision*, vol. 111, no. 2, pp. 171–190, 2015.
[25] X. Ren and Z. Lin, "Linearized alternating direction method with adaptive penalty and warm starts for fast solving transform invariant low-rank textures," *International Journal of Computer Vision*, vol. 104, no. 1, pp. 1–14, 2013.
[26] W. Siming and L. Zhouchen, "Analysis and improvement of low rank representation for subspace segmentation," *arXiv preprint arXiv:1107.1561*, 2011.
[27] A. Levy and M. Lindenbaum, "Sequential Karhunen-Loeve basis extraction and its application to images," *IEEE Transactions on Image Processing*, vol. 9, no. 8, pp. 1371–1374, 2000.
[28] H. Xu, C. Caramanis, and S. Sanghavi, "Robust pca via outlier pursuit," in *Advances in Neural Information Processing Systems*, 2010, pp. 2496–2504.
[29] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Foundations of Computational Mathematics*, vol. 9, no. 6, pp. 717–772, 2009.
[30] G. Liu and P. Li, "Recovery of coherent data via low-rank dictionary pursuit," in *Advances in Neural Information Processing Systems*, 2014, pp. 1206–1214.
[31] G. Liu, H. Xu, and S. Yan, "Exact subspace segmentation and outlier detection by low-rank representation," in *Artificial Intelligence and Statistics*, 2012, pp. 703–711.
[32] G. Liu and S. Yan, "Latent low-rank representation for subspace segmentation and feature extraction," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1615–1622.
[33] Z. Lin, R. Liu, and Z. Su, "Linearized alternating direction method with adaptive penalty for low-rank representation," in *Advances in Neural Information Processing Systems*, 2011, pp. 612–620.
[34] C. Zhang, R. Liu, T. Qiu, and Z. Su, "Robust visual tracking via incremental low-rank features learning," *Neurocomputing*, vol. 131, pp. 237–247, 2014.
[35] H. Xu, C. Caramanis, and S. Mannor, "Principal component analysis with contaminated data: The high dimensional case," *arXiv preprint arXiv:1002.4658*, 2010.
[36] D. Cai, X. He, and J. Han, "Document clustering using locality preserving indexing," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 12, pp. 1624–1637, 2005.
[37] Y. Zhao and G. Karypis, "Empirical and theoretical comparisons of selected criterion functions for document clustering," *Machine Learning*, vol. 55, no. 3, pp. 311–331, 2004.
[38] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

Bo Li received the Ph.D. degree in 2008 in computational mathematics, Dalian University of Technology (DUT), Dalian, China. Now he is the associate professor in the School of Mathematics and Informatics Science of Nanchang Hangkong University. His current research interests include the areas of image processing and computer graphics.

Risheng Liu received the BSc and PhD degrees both in mathematics from the Dalian University of Technology in 2007 and 2012, respectively. He was a visiting scholar in the Robotic Institute of Carnegie Mellon University from 2010 to 2012. He served as Hong Kong Scholar Research Fellow at the Hong Kong Polytechnic University from 2016 to 2017. He is currently an associate professor with the Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, Internal School of Information and Software Technology, Dalian University of Technology. His research interests include machine learning, optimization, computer vision and multimedia. He was a co-recipient of the IEEE ICME Best Student Paper Award in both 2014 and 2015. Two papers ware also selected as Finalist of the Best Paper Award in ICME 2017. He is a member of the IEEE and ACM.

Junjie Cao is a lecturer in School of Mathematical Sciences at Dalian University of Technology, P.R. China. He received Ph.D. degree in computational mathematics from Dalian University of Technology. His research interests include shape modelling, image processing and machine learning.



Jie Zhang received the PhD degree in 2015 from the Dalian University of Technology, china. She is currently a lecturer with the School of Mathematics, Liaoning Normal University, China. Her current research Interests include geometric processing and machine learning.

Yu-Kun Lai received the bachelors and Ph.D. degrees in computer science from Tsinghua University, China, in 2003 and 2008, respectively. He is currently a Senior Lecturer of visual computing with the School of Computer Science and Informatics, Cardiff University. His research interests include computer graphics, geometry processing, image processing, and computer vision. He is on the Editorial Board of The Visual Computer.

Xiuping Liu is a Professor in School of Mathematical Sciences at Dalian University of Technology, P.R. China. She received Ph.D. degree in computational mathematics from Dalian University of Technology. Her research interests include shape modelling and analysing.