Robust low-dimensional structure learning for big data and its applications

JIASHI FENG

(B.Eng., USTC)

A THESIS SUBMITTED FOR

THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

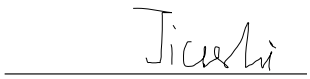
2014

Declaration

I hereby declare that this thesis is my original work and it has been written by me in its entirety.

I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.



Jiashi Feng

May 23, 2014

Acknowledgements

First and foremost, I am deeply indebted to my two advisors, Professor Shuicheng Yan and Professor Huan Xu. It has been an honor to be Ph.D. student co-advised by them. Their support and advice have been invaluable for me, in terms of both personal interaction and professionalism. I have benefited from their broad range of knowledge, deep insight and thorough technical guidance in each and every step of my research during the last four years. I thoroughly enjoyed working with them. Without their inspiration and supervision, this thesis would never have happened.

I am very grateful to Professor Trevor Darrell of the University of California at Berkeley for providing me with the opportunity of visiting his group at Berkeley. I was impressed by his enthusiasm and curiosity, and there I met many great researchers. I am fortunate to have had the chance to collaborate with Professor Shie Mannor at Technion, an experience that helped produce a significant portion of this thesis.

I would thank my friends at LV group, Qiang Chen, Zheng Song, Mengdi Xu, Jian Dong, Wei Xia, Tam Nguyen, Luoqi Liu, Junshi Huang, Min Lin, Canyi Lu, and others. They have created a very pleasant atmosphere in which to conduct research and live my life. I am very grateful to my senior Bingbing Ni for helping me at the beginning of my PhD career. Special thanks goes to Si Liu, Hairong Liu, Professor Congyan Lang and Professor Zilei Wang. The time we work together is my most precious moment in Singapore.

Finally, thanks to my parents for their love and support.

Contents

Summary

The explosive growth of data in the era of big data has presented great challenges to traditional machine learning techniques, since most of them are difficult to apply for handling large-scale, high-dimensional and dynamically changing data. Moreover, most of the current low-dimensional structure learning methods are fragile to the noise explosion in high-dimensional regime, data contamination and outliers, which however are ubiquitous in realistic data. In this thesis, we propose deterministic and online learning methods for robustly recovering the low-dimensional structure of data to solve the above key challenges. These methods possess high efficiency, strong robustness, good scalability and theoretically guaranteed performance in handling big data, even in the presence of noises, contaminations and adversarial outliers. In addition, we also develop practical algorithms for recovering the low-dimensional and informative structure of realistic visual data in several computer vision applications.

Specifically, we first develop a deterministic robust PCA method for recovering low-dimensional subspace of high-dimensional data, where the dimensionality of each datum is comparable or even larger than the number of data. The DHRPCA method is tractable, possesses maximal robustness, and asymptotic consistent in the high-dimensional space. More importantly, by smartly suppressing the affect of outliers in a batch manner, the method exhibits significantly high efficiency for handling large-scale data. Second, we propose two online learning methods, OR-PCA and online RPCA, to further enhance the scalability for robustly learning the low-dimensional structure of big data, under limited memory and computational cost budget. These two methods handle two different types of contaminations within the

data: (1) OR-PCA is for the data with sparse corruption and (2) online RPCA is for the case where a few of the data are completely corrupted. In particular, OR-PCA introduces a matrix factorization reformulation of nuclear norm which enables alternative stochastic optimization to be applicable and converge to the global optimum. Online RPCA devises a randomized sample selection mechanism which possesses provable recovering performance and robustness guarantee under mild condition. Both of these two methods process the data in a streaming manner and thus are memory and computationally efficient for analyzing big data.

Third, we devise two low-dimensional learning algorithms for visual data and solve several important problems in computer vision: (1) geometric pooling which generates discriminative image representation based on the low-dimensional structure of the object class space, and (2) auto-grouped sparse representation for discovering low-dimensional sub-group structure within visual features to generate better feature representations. These two methods achieve state-of-the-art performance on several benchmark datasets for the image classification, image annotation and motion segmentation tasks.

In summary, we develop robust and efficient low-dimensional structure learning algorithms which solve several key challenges imposed by big data for current machine learning techniques and realistic applications in computer vision field.

List of Tables

List of Figures

Chapter 1

Introduction

Both research and industry areas (such as engineering, computer science and economics) are currently generating terabytes (10^{12} bytes) or even petabytes (10^{15} bytes) of data in the observations, numerical simulations and experiments. Moreover, the emergence of e-commerce and web search engines has led us to confront the challenges of even larger scale of data. To be concrete, Google, Microsoft, and other social media companies (*e.g.*, Facebook, YouTube, Twitter) have data on the order of exabytes (10^{18} bytes) or beyond. Exploring the succinct and relational structure of the data removes the redundant and noisy information, and thus provides us with deeper insights into the information contained in the data which benefits our decision making, users behavior analyzing and prediction.

Actually, analysis of the information contained in these data sets have already led to major breakthroughs in fields ranging from economics to computer science and to the development of new information-based industries. However, traditional methods of analysis have been based largely on the assumption that analysts (*e.g.*, the learning and inference algorithms) can work with data within the their limited computing resources, but the growth of "big data" is imposing great challenges to them.

More specifically, the challenges raised by "big data" for the machine learning methods mainly lie on the following two aspects. First, the large scale of the data causes great storage and computational burdens on the modern sophisticated ma-

chine learning, inference and optimization algorithms. Many of existing standard learning algorithms, though they are statistically performing well, are hindered by their high computational complexity and do not scale well to the big data. Secondly, the real data usually contain contamination, which may come from the inherent noises, corruptions in the measuring or sampling process or even malicious contamination. Such noises and corruptions require the learning methods to possess strong robustness in order for yielding accurate inference results.

This thesis focuses on the problem of low-dimensional structure learning for big data analysis. In particular, we investigate and contribute to handling the noise explosion in the high-dimensional regime and the outliers within the data. Second, we apply the online learning algorithms to efficiently process the large-scale data under the limited budget of computational resources. Finally, we demonstrate two applications of the low-dimensional structure learning methods in object recognition and image classification.

1.1 Background and Related works

1.1.1 Low-dimensional Structure Learning

Low-dimensional structure represents a more succinct representation of the observed massive data than their original representation. Finding the low-dimensional structure of the massive observed data is able to remove the noisy or irrelevant information, identify the essential structure of the data and provide us with deeper insight into the information contained within the data. Moreover, with the help of the low-dimensional structure mining, we can more conveniently visualize, process and analyze the data.

Among the traditional low-dimensional structure learning methods, Principal Component Analysis (PCA) [57] is arguably the most popular one. PCA finds a low-dimensional subspace which is able to closely fit the observed data, in the sense of minimizing the square residual error. Following PCA, many other low-dimensional structure learning methods have been developed based on different criterion in ex-

plaining the data. For instance, Locality Preserving Projections (LPP) [122] is proposed to preserve the local relationships among the data after dimension reduction. Besides linear methods, some non-linear low-dimensional manifold learning methods are proposed to discover the underlying manifold structure of the data. Typical examples of those methods include ISOMAP [123], LLE [124], and Laplacian Eigenmap [125]. Some methods also explore the discriminative low-dimensional structure. For example, Linear Discriminative Analysis (LDA) [126], or called Fisher Discriminative Analysis (FDA), pursues a linear projection of the data belonging to different classes in order to maximize the class separability after the linear projection.

Besides pursuing an explicit linear or nonlinear transformation of the data into low-dimensional structure, some matrix decomposition based method has been proposed to implicitly find the underlying low-dimensional structure. A typical method is factorizing the data matrix as a low-rank matrix plus a noisy explaining matrix, where the low-rank factor matrix corresponds to the low-dimensional subspace of the data [44].

Generally, the methods are batch based and need to load all the data into memory to perform the inference. This incurs huge storage cost for processing big data. Moreover, though PCA and other linear methods admit streaming processing scheme, it is well known that they are quite fragile to outliers and have weak robustness.

1.1.2 Robustness in Structure Learning

As discussed above, noises are ubiquitous in realistic data. Traditional low-dimensional structure learning methods are able to handle the noise with small magnitude in relatively low-dimensional regime. However, along with the development of modern data generation and acquisition technologies, the dimensionality of realistic data keeps increasing. For example, images of much higher resolutions than before can be acquired rather conveniently. DNA microarray data, financial data, consumer data also possess quite high dimensionality. In dealing with such high-dimensional data, the dimensionality explosion is inevitable. However, traditional structure learning

methods may fail in this high-dimensional regime [36, 20, 52, 30, 19, 20, 29], due to their breakdown point being inversely proportional to the dimensionality, or the unaffordable computational complexity.

Besides the existence of noise in realistic data, some samples or certain dimension of the data may be corrupted, due to the sensor error or malicious contamination. The outliers will contaminate the data and manipulate the learning results. In fact, many of existing low-dimensional structure learning methods, *e.g.*, standard PCA, are shown to be quite fragile to the outliers. Even one outlier can make the results arbitrarily bad.

Robustifying the traditional machine learning algorithms becomes a hot and quite valuable research topic, especially for processing the realistic data with contamination. In particular, many robust learning methods have been proposed for learning the low-dimensional structure of data [36, 20, 52, 30, 19, 20, 29]. Traditional machine learning algorithms are generally robustified by employing certain robust statistics which have high breakdown point. For instance, some of the existing RPCA methods adopt M-estimator, S-estimator Minimum Covariance Determinant (MCD) estimator to obtain the robust estimation of the sample covariance matrix. Robust regression based on the robust counterpart of vector inner product to enhance the robustness, even though there is contamination on the both design matrix and response variables [127]. Another line of the robust learning is to explicitly model the added noise on the samples, with certain structural prior, such as gross though sparse error used in the PCP robust PCA algorithm [44]. In this thesis, we focus on proposing robust structural learning methods, which can well handle both the noise in high-dimensional regime and the outliers. In this thesis, we propose several robust learning methods which are proved to achieve the maximal robustness.

1.1.3 Online Learning

Online learning is developed for solving the problems where the data are revealed incrementally over time, and the learner needs to make prediction only based on the

data revealed to now, without any knowledge about the coming data in the future. Online learning originates from game theory, but has been studied in many other research fields, including information theory and machine learning. Online learning also becomes of great interest to practitioners due to the recent emergence of large scale applications such as online advertisement placement and online web ranking.

More formally, online learning is performed in a sequence of consecutive rounds, where at round t the learner is given a question, $\mathbf{x}_t$, taken from an instance domain $\mathcal{X}$, and is required to provide an answer to this question, which we denote by p_t. After predicting an answer, the correct answer, y_t, taken from a target domain $\mathcal{Y}$, is revealed and the learner suffers a loss, $l(p_t, y_t)$, which measures the discrepancy between its answer and the correct one. The target of the learner is thus to minimize the cumulative loss $\sum_t l(p_t, y_t)$ or expected loss $\mathbb{E}_{\mathcal{X}} l(p_t, y_t)$.

Online learning obviously has the advantages of cheap memory cost in learning from big data. The online learner only loads one datum or a small batch of the data into the memory at each time instance, and does not need to re-explore the previous data in the learning process. In contrast, batch based machine learning algorithms require to load all the observed data into the memory to perform the parameter learning and inference. This imposes huge computational burden, especially storage burden, on the learners and prevents the learners from scaling to big data.

Though they have appealing efficiency advantages, online learning methods often have quite weak robustness. This is because that the usage of robust statistics for robustifying the learning methods generally requires statistics over all the data. It is difficult for the online learning methods which only have a partial observation of the data to obtain such robust statistics. In this thesis, we investigate and propose robust online learning algorithms for processing big realistic data.

1.2 Thesis Focus and Main Contributions

In this thesis, we focus on robust and efficient low-dimensional structure learning for big data analysis. The main motivations are as follows:

1. For more efficient batch high-dimensional RPCA algorithm. Big data often have high dimensionality. In the high-dimensional regime, noise explosion will destroy the signal and fail many existing low-dimensional subspace learning method. A strategy to handle the noise and outliers is to introduce randomness on the sample selection. However, such method is quite inefficient as only at most one sample is removed in each optimization iteration. A deterministic method is desired for providing high efficiency.

2. With limited budget of memory, how to handle the large-scale dataset. For common users, the computational budget is usually limited. However, traditional machine learning methods are generally batch based, which require to load all the data into memory. This is the bottleneck for processing big data. Therefore, an online learning algorithm which processes the data in a streaming manner and meanwhile preserves the desired property of the batch methods is required.

3. We are also interested in the application of the low-dimensional structure learning method in real applications. In particular, we focus on solving the problem of object recognition in computer vision research field. The discovered low-dimensional structure is able to convey more essential and discriminative information for classification. Thus, based on such structure, more discriminative image representations can be obtained which are more beneficial for image classification and/or object recognition.

In this thesis, the robust low-dimensional structure learning method, especially for the low-dimensional subspace learning, is proposed. Furthermore, we successfully scale the method to big data regime via proposing the online learning method. We also apply the low-dimensional learning method on computer vision applications. More specifically, we conduct research on the following aspects:

1. Deterministic high-dimensional robust PCA method. We first develop a deterministic robust PCA method for recovering low-dimensional subspace of high-

dimensional data, where the dimensionality of each datum is comparable or even larger than the number of data. The DHRPCA method is tractable, possesses maximal robustness, and asymptotic consistent in the high-dimensional space. More importantly, by smartly suppressing the affect of outliers in a batch manner, the method exhibits significantly high efficiency for handling large-scale data.

2. Online robust PCA methods.

   Second, we propose two online learning methods, OR-PCA and online RPCA, to further enhance the scalability for robustly learning the low-dimensional structure of big data, under limited memory and computational cost budget. These two methods handle two different types of contaminations within the data: (1) OR-PCA is for the data with sparse corruption and (2) online RPCA is for the case where a few of the data are completely corrupted. In particular, OR-PCA introduces a matrix factorization reformulation of nuclear norm which enables alternative stochastic optimization to be applicable and converge to the global optimum. Online RPCA devises a randomized sample selection mechanism which possesses provable recovering performance and robustness guarantee under mild condition. Both of these two methods process the data in a streaming manner and thus are memory and computationally efficient for analyzing big data.

3. The applications in computer vision tasks. Furthermore, we devise two low-dimensional learning algorithms for visual data and solve several important problems in computer vision: (1) geometric pooling which generates discriminative image representation based on the low-dimensional structure of the object class space, and (2) auto-grouped sparse representation for discovering low-dimensional sub-group structure within visual features to generate better feature representations. These two methods achieve state-of-the-art performance on several benchmark datasets for the image classification, image annotation and motion segmentation tasks.

1.3 Structure of The Thesis

In Chapter 2, we propose a deterministic robust PCA method for learning the low-dimensional structure of data in high-dimensional regime. Then in Chapter 3 and Chapter 4, we propose two different online robust PCA methods to handle data with different corruption models. Finally, we demonstrate two applications of the low-dimensional structure learning in object recognition and image annotation tasks in Chapter 5 and Chapter 6.

Chapter 2

Robust PCA in High-dimension: A Deterministic Approach

In this chapter, we propose our robust PCA method for handing the data with quite high dimensionality and meanwhile a subset of the data is corrupted to be outliers. We propose a deterministic algorithm which is much more efficient than its randomized counterpart yet possesses the maximal robustness.

2.1 Introduction

This chapter is about robust **p**rincipal **c**omponent **a**nalysis (PCA) for high-dimensional data, a topic that has drawn surging attention in recent years. PCA is one of the most widely used data analysis methods [57]. It constructs a low-dimensional subspace based on a set of principal components (PCs) to approximate the observations in the least-square sense. Standard PCA computes PCs as eigenvectors of the sample covariance matrix. Due to the quadratic error criterion, PCA is notoriously sensitive and fragile, and the quality of its output can suffer severely in the face of even few corrupted samples. Therefore, it is not surprising that many works have been dedicated to robustifying PCA [52, 20, 44].

Analyzing high dimensional data – data sets where the dimensionality of each observation is comparable to or even larger than the number of observations – has

become a critical task in modern statistics and machine learning [6]. Practical high dimensional data, such as DNA microarray data, financial data, consumer data, and climate data, easily have dimensionality ranging from thousand to billions. Partly due to the fact that extending traditional statistical tools (designed for the low dimensional case) into this high-dimensional regime are often unsuccessful, tremendous research efforts have been made to design fresh statistical tools to cope with such "dimensionality explosion".

The work in [61] is among the first to analyze robust PCA algorithms in the high-dimensional setup. They identified three pitfalls, namely diminishing breakdown point, noise explosion and algorithmic intractability, where previous robust PCA algorithms stumble. They then proposed the high-dimensional robust PCA (HR-PCA) algorithm that can effectively overcome these problems, and showed that HR-PCA is tractable, *provably* robust and easily kernelizable. In particular, in contrast to standard PCA and existing robust PCA algorithms, HR-PCA is able to robustly estimate the PCs in the *high-dimensional regime* even in the face of a constant fraction of outliers and extremely low Signal Noise Ratio (SNR) – the breakdown point of HR-PCA is 50%, [1] which is the highest breakdown point can ever be achieved, whereas other existing methods all have breakdown points diminishing to zero. Indeed, to the best of our knowledge, HR-PCA appears to be the only algorithm having these properties in the high-dimensional regime.

Briefly speaking, HR-PCA is an iterative method which in each iteration performs standard PCA, and then *randomly* remove *one* point in a way that outliers are more likely to be removed, so that the algorithm converges to a good output. Because in each iteration, only one point is removed, the number of iterations required to find a good solution is at least as much as the number of outliers. This, combined with the fact that PCA is computationally expensive itself, prevents HR-PCA from effectively handling large-scale data-sets with many outliers. In addition, the performance of HR-PCA depends on the ability of the built-in random removal

[1] Breakdown point is a robustness measure defined as the percentage of corrupted points that can make the output of the algorithm arbitrarily bad.

to eliminate outliers correctly, which is only guaranteed in a *probabilistic* manner.

To address these two issues, we propose a *deterministic* high dimensional robust PCA algorithm (DHR-PCA). Specifically, instead of removing *one* point, the proposed algorithm decreases the weights of *all* observations in each iteration, in a way that the total weight of the outliers will decrease faster than that of the true samples. We show that DHR-PCA inherits all desirable theoretical properties of HR-PCA, including tractability, kernelizability, the maximal breakdown point, provable performance guarantee and asymptotical optimality. Moreover, DHR-PCA can be much more computationally efficient than (randomized) HR-PCA. As we show below, the number of iterations for DHR-PCA to converge is nearly constant, in sharp contrast to HR-PCA whose number of iterations required increases linearly with the number of outliers. Simulations in Section 2.4 show that for any fixed number of iterations, the solution to DHR-PCA is at least as good as HR-PCA, and is significantly better when the number of iterations is small. This is very appealing in practice, as both algorithms are "any-time" algorithms, i.e., one can terminate the algorithms at any time and obtain the best solution so-far.

2.2 Related Work

Besides HR-PCA, there have been abundant works on robust PCA, which we briefly discuss in this section. Robust PCA algorithms focusing on the low-dimensional setup [e.g., 36, 20, 52] can be roughly categorized into two groups. The first group of algorithms pursue robust estimation of the covariance matrix, *e.g.*, M-estimator [32], S-estimator [37], and Minimum Covariance Determinant (MCD) estimator [36]. These algorithms generally provide more robust results, but their applicability is severely limited to small or moderate dimensions, as there are not enough observations to robustly estimate a high-dimensional covariance matrix. The second group of algorithms directly maximize certain robust estimation of univariate variance for the projected observations and then obtain maximizers as the candidate principal components [30, 19, 20, 29]. These algorithms inherit the robustness characteristics

of the adopted estimators and are qualitatively robust. However, all of these algorithms run into unsolvable issues in the high dimensional regime incurred by the curse of dimensionality as stated in the followings.

The targeted high-dimensional regime poses three main challenges to existing robust PCA methods. First, some robust PCA algorithms have breakdown point inversely proportional to the dimensionality, *e.g.*, M-estimator [32], in the high-dimensional regime their breakdown points will diminish and the results will be arbitrarily bad in presence of even few outliers. Second, widely used outlyingness indicators, including Mahalanobis distance and Stahel-Donoho outlyingness [5] are no longer valid, due to a phenomenon termed – "noise explosion" [61]. This causes the algorithms relying on such outlyingness measures [52] to collapse. The third problem is that the dimensionality may be larger than the number of data points and thus some robust estimators including Minimum Volume Ellipsoid (MVE) and Minimum Covariance Determinant (MCD) [36] become degenerated. Furthermore, the extremely high computational complexity of these estimators and projection pursuit methods for high dimensional data prevents them from being tractable.

Finally, we discuss recent works addressing robust PCA using low-rank technique. [44] developed a framework to perform robust PCA using low-rank matrix decomposition. Yet, their method focuses on the scenario that *random* entries of the observation matrix are arbitrarily corrupted, which differs from our setup where one corrupted data point may change the whole column of the observation matrix. The later setup is then investigated in Xu et al. [16]. While their proposed method performs well under a small fraction of outliers, it breaks down for larger fraction of outliers – in particular, the breakdown point is far from 50%. Moreover, the performance scales unfavorably with the magnitude of noise, which makes it not suitable for the high-dimensional setup, due to "noise-explosion".

2.3 The Algorithm

In this section, we first formally state the problem setup of the high dimensional robust PCA. Then we provide the details of the proposed DHR-PCA algorithm and finally present the main theoretic results on the performance guarantees of the algorithm.

2.3.1 Problem Setup

In this subsection, we present the formal problem description of PCA for the high dimensional data with contamination. Our setup, detailed below for completeness, largely follows the pervious work in [61].

Given n observations, there are t observations not corrupted, called authentic samples. The authentic samples $\mathbf{z}_i \in \mathbb{R}^m$ are generated through a linear mapping: $\mathbf{z}_i = A\mathbf{x}_i + \mathbf{n}_i$. Here, noise $\mathbf{n}_i$ is sampled from normal distribution $\mathcal{N}(\mathbf{0}, I_m)$; and the signal $\mathbf{x}_i \in \mathbb{R}^d$ are i.i.d. samples of a random variable $\mathbf{x}$ with mean zero and variance I_d. The matrix $A \in \mathbb{R}^{m\times d}$ and the distribution μ of $\mathbf{x}$ are unknown. We assume μ is absolutely continuous w.r.t. the Borel measure and spherically symmetric. And μ has light tails, i.e., there exist constants $K, C > 0$ such that $\Pr(\|\mathbf{x}\| \geq x) \leq K\exp(-Cx)$ for all $x \geq 0$. We are interested in the case where $n \approx m \gg d$, i.e., the dimensionality of observations is much larger than that of signals and of the same order as the number of observations.

The outliers (the corrupted data) are denoted as $\mathbf{o}_1, \ldots, \mathbf{o}_{n-t} \in \mathbb{R}^m$ and they are with arbitrary values. We only require that $n - t \leq t$, i.e., the number of outliers are not more than that of authentic samples. Let $\lambda \triangleq (n-t)/n$ be the fraction of corrupted points. We observe the contaminated dataset

$$\mathcal{Y} \triangleq \{\mathbf{y}_1, \ldots, \mathbf{y}_n\} = \{\mathbf{z}_1, \ldots, \mathbf{z}_t\} \bigcup \{\mathbf{o}_1, \ldots, \mathbf{o}_{n-t}\},$$

and aim to recover the principal components of A, *i.e.*, the top eigenvectors $\bar{\mathbf{w}}_1, \ldots, \bar{\mathbf{w}}_d$ of AA^T. That is, we seek a collection of orthogonal vectors $\mathbf{w}_1, \ldots, \mathbf{w}_d$, that maxi-

mize the following performance metric called the *Expressed Variance* (E.V.):

$$\text{E.V.}(\mathbf{w}_1, \ldots, \mathbf{w}_d) \triangleq \frac{\sum_{j=1}^{d} \mathbf{w}_j^T A A^T \mathbf{w}_j}{\sum_{j=1}^{d} \bar{\mathbf{w}}_j^T A A^T \bar{\mathbf{w}}_j}.$$

The E.V. represents the portion of signal $A\mathbf{x}$ being expressed by $\mathbf{w}_1, \ldots, \mathbf{w}_d$. Thus, $1 - \text{E.V.}$ is the reconstruction error of the signal. The E.V. is a commonly used evaluation metric for the PCA algorithms [61, 21]. It is always less than one, with equality achieved by a perfect recovery, i.e., the vectors $\mathbf{w}_1, \ldots, \mathbf{w}_d$ have the same span as the true principal components $\{\bar{\mathbf{w}}_1, \ldots, \bar{\mathbf{w}}_d\}$.

The distribution μ affects the performance of the algorithms through its tail. We hence adapt the following tail weight function $\mathcal{V} : [0, 1] \to [0, 1]$ from [61], which essentially represents how the tail of $\bar{\mu}$ contributes to its variance,

$$\mathcal{V}(\alpha) \triangleq \int_{-c_\alpha}^{c_\alpha} x^2 \bar{\mu}(dx),$$

where $\bar{\mu}$ is the one-dimensional margin of μ and c_α is such that $\bar{\mu}([-c_\alpha, c_\alpha]) = \alpha$. Notice that $\mathcal{V}(0) = 0, \mathcal{V}(1) = 1$, and $\mathcal{V}(\cdot)$ is continuous.

2.3.2 Deterministic HR-PCA Algorithm

Our main algorithm is given in Algorithm 1. Here, a Robust Variance Estimator (RVE) $\bar{V}_{\hat{t}}(\cdot)$ is adopted to identify the candidate principal components. For $\mathbf{w} \in \mathcal{S}_m$, the RVE is defined as $\bar{V}_{\hat{t}}(\mathbf{w}) \triangleq \frac{1}{n} \sum_{i=1}^{\hat{t}} |\mathbf{w}^T \mathbf{y}|^2_{(i)}$, where the subscript $(\cdot)$ denotes a non-decreasing order of the variables. And it can be seen that the RVE stands for the following statistics: project $\mathbf{y}_i$ onto the direction $\mathbf{w}$, replace the furthest $n - \hat{t}$ samples by 0, and then compute the variance. If the variance is large, it is likely that a correct principal component direction is found. Otherwise, a number of points with largest variance may be corrupted. Notice that the RVE is always performed on the original observed set $\mathcal{Y}$. We find that RVE coincides with the robust L-estimator, which is defined as a linear combination of order statistics: $T_n = \sum_{i=1}^{n} a_{ni} h(x_{(i)})$ for some function h.

Algorithm 1 DHR-PCA.

Input: Contaminated sample set $\mathcal{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_n\} \subset \mathbb{R}^m$, parameters $d, \hat{t}$.
Output: Recovered PCs: $\mathbf{w}_1^*, \ldots, \mathbf{w}_d^*$.
Initialize $\hat{\mathbf{y}}_i := \mathbf{y}_i, \alpha_i = 1, \forall i = 1, \ldots, n; \text{Opt} := 0$.
repeat

1. Compute the empirical variance matrix

$$\hat{\Sigma} := \frac{1}{n} \sum_{i=1}^{n} \alpha_i \hat{\mathbf{y}}_i \hat{\mathbf{y}}_i^T;$$

2. Perform PCA on $\hat{\Sigma}$. Let $\mathbf{w}_1, \ldots, \mathbf{w}_d$ be the d principle components of $\hat{\Sigma}$;
3. If $\sum_{j=1}^{d} \bar{V}_{\hat{t}}(\mathbf{w}_j) > \text{Opt}$, then let $\text{Opt} := \sum_{j=1}^{d} \bar{V}_{\hat{t}}(\mathbf{w}_j)$ and let $\mathbf{w}_j^* := \mathbf{w}_j$ for $j = 1, \ldots, d$;
4. Calculate

$$\eta = \min_i \frac{1}{\sum_{j=1}^{d} \left(\mathbf{w}_j^T \hat{\mathbf{y}}_i \right)^2}, \forall i : \alpha_i \neq 0.$$

5. Update the sample weight $\alpha_i := \alpha_i - \Delta\alpha_i$, $\forall i : \alpha_i \neq 0$, where $\Delta\alpha_i = \eta\alpha_i \sum_{j=1}^{d} \left(\mathbf{w}_j^T \hat{\mathbf{y}}_i \right)^2$;

until Convergence

We now explain our innovation compared to HR-PCA, and its intuition. In HR-PCA, steps 4 and 5 are replaced by a random removal – the probability $\hat{\mathbf{y}}_i$ being removed is proportional to $\sum_{j=1}^{d} \left(\mathbf{w}_j^T \hat{\mathbf{y}}_i \right)^2$. It has been shown in [61] that *in expectation (and in probability)*, either the number of outliers will decrease faster, or the algorithm will find a good solution. Since in each iteration, only one point is removed, the number of iterations required to find a satisfactory output depends linearly on the number of outliers.

Instead of resorting to a random mechanism, DHR-PCA deterministically reduce the effect of corrupted data points. In particular, Moreover, DHR-PCA operates on all the data points in each iteration, which decouples the dependence of the computational cost on the number of outliers and enhances the efficiency significantly compared with HR-PCA. We consider an artificial example to illustrate this: assume both HR-PCA and DHR-PCA requires M iterations for a data-set $\mathcal{Y}_0$. Now suppose a new data-set $\mathcal{Y}$ contains J identical copies of data-set $\mathcal{Y}_0$. Then the number of iterations for DHR-PCA remains unchanged, while HR-PCA requires

JM iterations. Simulation results for more realistic setups, reported in Section 2.4, also demonstrate that the deterministic algorithm provides higher efficiency than HR-PCA.

Theorem 1 and Theorem 2 below show that the proposed algorithm achieves the same performance guarantees as HR-PCA. The proofs are shown in Section 3.5.

Theorem 1. (Finite Sample Performance) *Let the Algorithm 1 output* $\{\mathbf{w}_1, \ldots, \mathbf{w}_d\}$. *Fix a* $\kappa > 0$, *and let* $\tau = \max(m/n, 1)$. *There exists a universal constant* c_0 *and a constant* C *which can possibly depend on* $\hat{t}/t, \lambda, d, \mu$ *and* κ, *such that for any* $\gamma < 1$, *if* $n/\log^4 n \geq \log^6(1/\gamma)$, *then with probability* $1-\gamma$ *the following holds*

$$\begin{aligned}
&\text{E.V.}\{\mathbf{w}_1, \ldots, \mathbf{w}_d\} \\
\geq\ & \left[\frac{\mathcal{V}\left(1 - \frac{\lambda(1+\kappa)}{(1-\lambda)\kappa}\right)}{(1+\kappa)}\right] \times \left[\frac{\mathcal{V}\left(\frac{\hat{t}}{t} - \frac{\lambda}{1-\lambda}\right)}{\mathcal{V}\left(\frac{\hat{t}}{t}\right)}\right] \\
-\ & \left[\frac{8\sqrt{c_0 \tau d}}{\mathcal{V}\left(\frac{\hat{t}}{t}\right)}\right] \left(\text{trace}(AA^T)\right)^{-1/2} \\
-\ & \left[\frac{2c_0\tau}{\mathcal{V}\left(\frac{\hat{t}}{t}\right)}\right] \left(\text{trace}(AA^T)\right)^{-1} - C\frac{\log^2 n \log^3(1/\gamma)}{\sqrt{n}}.
\end{aligned}$$

We also consider the asymptotic performance of the proposed algorithm when the dimension and the number of data points grow together to infinity. Our asymptotic setting is similar to [61]. Suppose there exists a sequence of sample sets $\{\mathcal{Y}(j)\} = \{\mathcal{Y}(1), \mathcal{Y}(2), \ldots\}$, where $\mathcal{Y}(j), n(j), m(j), A(j), d(j)$, etc., denote the corresponding values of the quantities defined above, the following must hold for some positive constants c_1, c_2:

$$\begin{aligned}
&\lim_{j\to\infty} \frac{n(j)}{m(j)} = c_1; d(j) \leq c_2; m(j) \uparrow +\infty; \\
&\text{trace}\left(A(j)^T A(j)\right) \uparrow \infty.
\end{aligned} \tag{2.1}$$

While $\text{trace}\left(A(j)^T A(j)\right) \uparrow \infty$, if it scales slowly than $\sqrt{m(j)}$, the SNR will asymptotically decrease to zero.

The last three terms in Theorem 1 go to zero as the dimension and number of points scale to infinity, i.e., as n and $m \to \infty$. Therefore, we immediately obtain:

Theorem 2. (Asymptotic Performance) *Given a sequence of* $\{\mathcal{Y}(j)\}$*, if the asymptotic scaling in Expression* (2.1) *holds, and* $\limsup \lambda(j) \leq \lambda^*$*, then the following holds in probability when* $j \uparrow \infty$ *(i.e., when* n *and* $m \uparrow \infty$*),*

$$\begin{aligned} & \liminf_j \text{E.V.}\{\mathbf{w}_1(j), \ldots, \mathbf{w}_d(j)\} \\ \geq & \max_{\kappa} \left[\frac{\mathcal{V}\left(1 - \frac{\lambda^*(1+\kappa)}{(1-\lambda^*)\kappa}\right)}{(1+\kappa)} \right] \times \left[\frac{\mathcal{V}\left(\frac{\hat{t}}{t} - \frac{\lambda^*}{1-\lambda^*}\right)}{\mathcal{V}\left(\frac{\hat{t}}{t}\right)} \right]. \end{aligned} \tag{2.2}$$

Observe that when $\lambda^* = 0$, i.e., the number of outliers scales sublinearly, the right-hand-side converges to 1 by taking $\kappa(j) = \sqrt{\lambda(j)}$, implying that the algorithm is asymptotically optimal. On the other hand, for any $\lambda < 0.5$, the right hand side is strictly positive (picking κ large enough), implying that the breakdown point converges to 50%.

For small λ, we can make use of the light tail condition on $\bar{\mu}$, to establish the following bound that simplifies (2.2). The proof is deferred to the supplementary material.

Corollary 1. *Under the settings of the above theorem, the following holds in probability when* $j \uparrow \infty$ *(i.e., when* $n, p \uparrow \infty$*),*

$$\liminf_j \text{E.V.}\{\mathbf{w}_1(j), \ldots, \mathbf{w}_d(j)\} \geq 1 - \frac{C'\sqrt{\alpha\lambda^* \log(1/\lambda^*)}}{\mathcal{V}(0.5)}.$$

Before concluding this section, we remark that DHR-PCA is easily kernelizable. Specifically, given a mapping function $\phi(\cdot) : \mathbb{R}^m \to \mathcal{H}$ and kernel function $k(\cdot, \cdot)$ satisfying $k(\mathbf{x}, \mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^m$, we can perform dimension reduction without requiring the explicit form of $\phi(\cdot)$ in the kernel PCA [14]. In particular, for the centered mapped features $\{\phi(\mathbf{y}_1), \cdots, \phi(\mathbf{y}_n)\}$, the output PCs can be represented as

$$\mathbf{w}_q = \sum_{j=1}^{n} a_j(k) \phi(\hat{\mathbf{y}}_j).$$

And the feature projection can be calculated by

$$\langle \mathbf{w}_q, \phi(\mathbf{v}) \rangle = \sum_{j=1}^{n} a_j(q) k(\hat{\mathbf{y}}_j, \mathbf{v}),$$

where $a(q)$ is the q^{th} eigenvector of the kernel matrix. Note that Algorithm 1 only involves calculating $\langle \mathbf{w}_q, \phi(\mathbf{y}_i) \rangle$ (in RVE evaluation) and $\langle \mathbf{w}_q, \phi(\sqrt{\alpha_i}\mathbf{y}_i) \rangle$ (in decreasing values of α_i's). Since the kernelization of both these two steps are obtained, the DHR-PCA algorithm can be kernelized easily.

2.4 Simulations

We devote this section to experimentally comparing the proposed DHR-PCA with HR-PCA. Since HR-PCA has shown superior robustness (against the dimensionality and number of outliers) over several robust PCA algorithms and standard PCA [61], we skip simulations for them here.

The numerical study is aimed to illustrate that DHR-PCA is much more efficient than HR-PCA, and meanwhile it achieves competitive performance. Here, we report the results for $d = 1$. We follow the data generation method in [61] to randomly generate an $m \times 1$ matrix and then scale its leading singular value to σ. A λ fraction of outliers are generated on a line with a uniform distribution over $[-\sigma \cdot \text{mag}, \sigma \cdot \text{mag}]$. Thus, "mag" represents the ratio between the magnitude of the outliers and that of the signal $A\mathbf{x}_i$ and is fixed as 10. The value of $\hat{t}$ is set as $(1 - \lambda)n$, if λ is known exactly. Otherwise, $\hat{t}$ can be simply set as $0.5n$. For each parameter setup, we report the average result of 20 tests and standard deviation.

Figure 2.1 shows the results for $m = 100$, 1000 and 10000 cases respectively with $\sigma = 5$. From the figure, we can make following observations. Firstly, DHR-PCA converges much faster than HR-PCA, especially for a large number of outliers. For example, when $m = 10000$ and $\lambda = 0.4$, the proposed algorithm converges using less than 2 iterations in average while HR-PCA needs more than 4000 iterations to converge. Secondly, the computational time for DHR-PCA in *each* iteration is

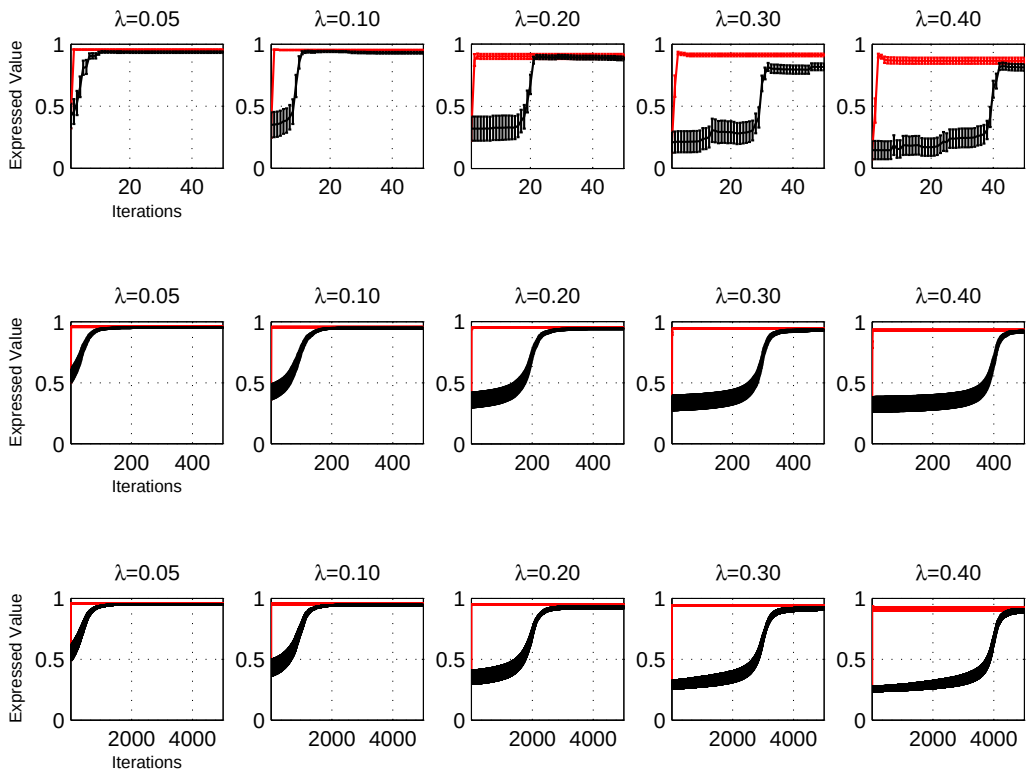


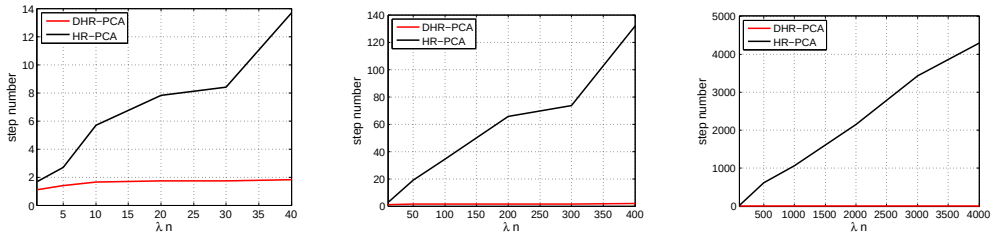
Figure 2.1: DHR-PCA (red line) vs. HR-PCA (black line) with $\sigma = 5$. Upper panel: $m = n = 100$, middle panel: $m = n = 1000$ and bottom panel: $m = n = 10000$. The horizontal axis is the iteration and the vertical axis is the expressive variance value. Please refer to the color version.

always in the same order as HR-PCA. These results well demonstrate that DHR-PCA is much more efficient than HR-PCA.

As for the performance, i.e., the E.V. of the recovered PCs, Figure 2.1 shows that DHR-PCA performs competitively to HR-PCA. For all the cases, the E.V. of final solution of DHR-PCA is always larger than that of HR-PCA. Moreover, if we terminate both algorithms at any early iteration, DHR-PCA always perform better than HR-PCA. This is appealing in practice, as we can terminate DHR-PCA at any time and obtain a satisfactory result in practical implementation. In addition, both DHR-PCA and HR-PCA perform quite well even in presence of varying number of outliers ($\lambda = 0.05$ to 0.4) and small signal magnitude ($\sigma = 5$), which coincides with the results in [61].

We then investigate the relationship between the number of iterations before

convergence and the number of outliers for the two methods. As shown in Figure 2.2, the number of iterations taken by HR-PCA is approximately proportional to the number of corrupted points. This is not surprising, since in each iteration HR-PCA removes at most one outlier. In a stark contrast, the number of required iterations of DHR-PCA remains nearly constant, shown by the flat curve in the figures. This demonstrates that DHR-PCA has good scalability and can potentially be applied to large real applications. We provide more simulations from Figure 2.3 to Figure 2.14. In the following figures, we provide more simulation results for comparison between



(a) $m = n = 100$ (b) $m = n = 1000$ (c) $m = n = 10000$

Figure 2.2: DHR-PCA (red line) vs. HR-PCA (black line) on the iterative steps taken by them before convergence with $\sigma = 5$ and different dimensionality. The horizontal axis λn is number of corrupted data points and the vertical axis is the number of steps. Please refer to the color version.

DHR-PCA and HR-PCA.

2.5 Proof of Theorem 1

In this section, we sketch the proof of Theorem 1. In what follows, we let $d, m/n, \lambda, \hat{t}/t$, and μ be fixed. We can fix a $\lambda \in (0, 0.5)$ w.l.o.g. due to the fact that if a result is shown to hold for λ, then it holds for $\lambda' < \lambda$. The letter c is used to represent a constant, and ϵ is a constant that decreases to zero as n and m increase to infinity. Let $\mathbf{w}_1(s), \ldots, \mathbf{w}_d(s)$ be the candidate solution at stage s. Let $\mathcal{Z}$ and $\mathcal{O}$ be the sets of indices of authentic samples and corrupted samples respectively. We let $\mathcal{B}_d \triangleq \{\mathbf{w} \in \mathbb{R}^d | \|\mathbf{w}\| \leq 1\}$, and $\mathcal{S}_d$ be its boundary. Here Theorems 3 and 4 are directly adapted from [61].

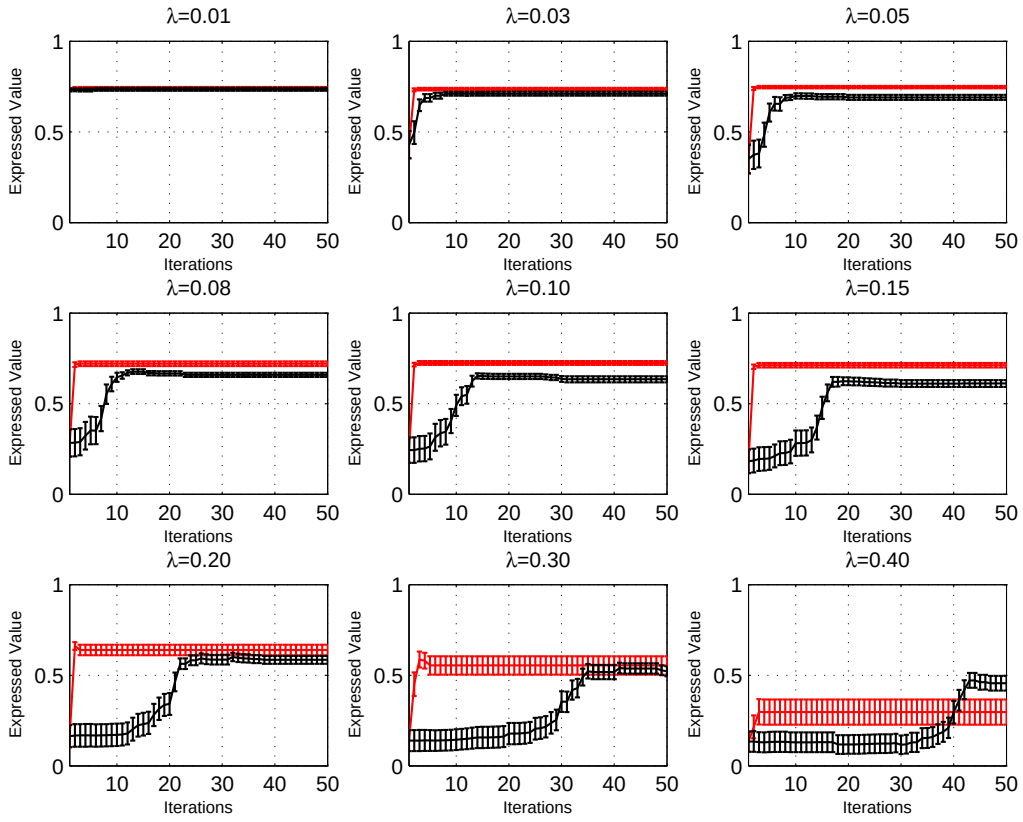


Figure 2.3: DHR-PCA (red line) vs. HR-PCA (black line). $m = n = 100, \sigma = 2$. The horizontal axis is the iteration and the vertical axis is the expressive variance value. Please refer to the color version.

2.5.1 Validity of the Robust Variance Estimator

We first show that the following condition holds with high probability. The detailed proof can be found in [61].

Condition 1. *There exists* $\epsilon_1, \epsilon_2, \bar{c}$ *such that (I)*$\sup_{\mathbf{w}\in\mathcal{S}_d} \left| \frac{1}{t}\sum_{i=1}^{t'} \left|\mathbf{w}^T\mathbf{x}\right|^2_{(i)} - \mathcal{V}\left(\frac{t'}{t}\right)\right| \leq \epsilon_1$*; (II)* $\sup_{\mathbf{w}\in\mathcal{S}_d} \left| \frac{1}{t}\sum_{i=1}^{t} \left|\mathbf{w}^T\mathbf{x}_i\right|^2 - 1\right| \leq \epsilon_2$*; (III)* $\sup_{\mathbf{w}\in\mathcal{S}_m} \frac{1}{t}\sum_{i=1}^{t} \left|\mathbf{w}^T\mathbf{n}_i\right|^2 \leq \bar{c}$.

Theorem 3. *Fix any* $\eta < 1$. *With probability at least* $1 - 3\gamma$, *Condition 1 holds uniformly for all* $t' \leq \eta t$, *with* $\bar{c} = c\tau(1 + \frac{\log(1/\gamma)}{n})$, $\epsilon_2 = c\log^2 n \log^3(1/\gamma)/\sqrt{n}$, *and* $\epsilon_1 = c\sqrt{\frac{\log n + \log(1/\gamma)}{n}} + \frac{c\log^{2.5} n \log^{3.5}(1/\gamma)}{n}$, *for a constant* c *possibly depends on* d, μ *and* η.

Under Condition 1, RVE is a good estimator.

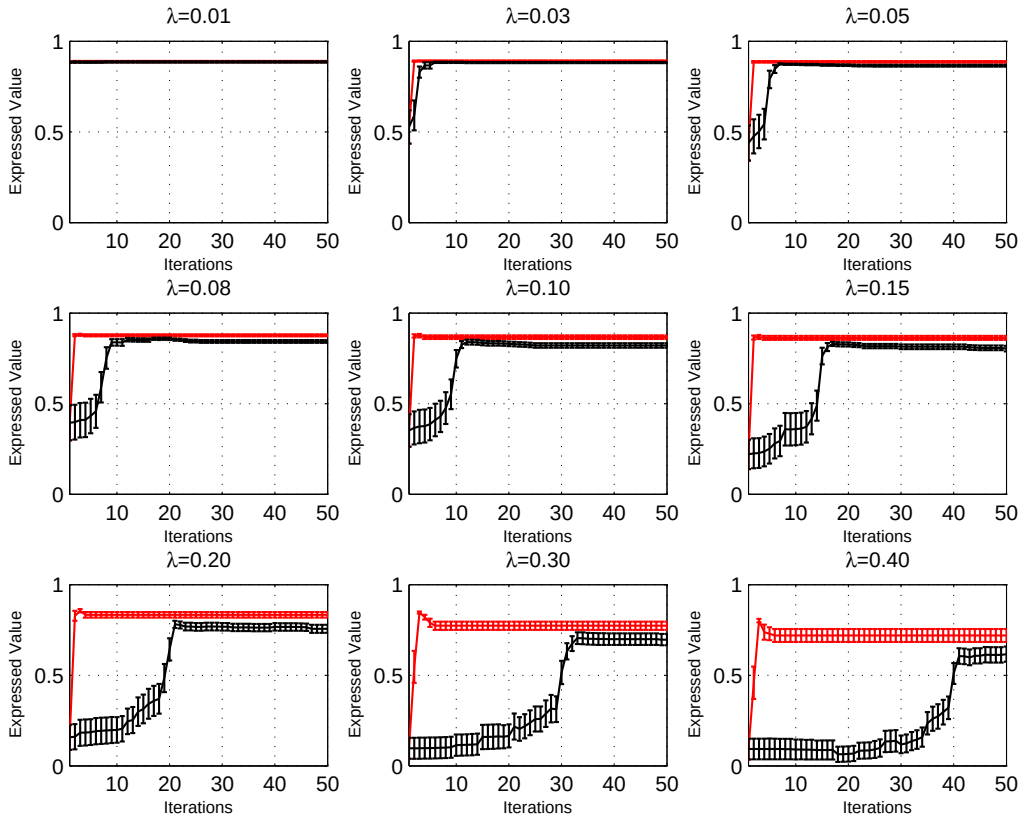


Figure 2.4: DHR-PCA (red line) vs. HR-PCA (black line). $m = n = 100, \sigma = 3$. The horizontal axis is the iteration and the vertical axis is the expressive variance value. Please refer to the color version.

Theorem 4. *Let $t' \leq t$. Suppose Condition 1 holds. Then for all $\mathbf{w} \in \mathcal{S}_m$ the following holds:*

$$\begin{aligned} & (1-\epsilon_1)\|\mathbf{w}^T A\|^2 \mathcal{V}\left(\frac{t'}{t}\right) - 2\|\mathbf{w}^T A\|\sqrt{(1+\epsilon_2)\bar{c}} \\ \leq & \frac{1}{t}\sum_{i=1}^{t'} |\mathbf{w}^T \mathbf{z}|^2_{(i)} \\ \leq & (1+\epsilon_1)\|\mathbf{w}^T A\|^2 \mathcal{V}\left(\frac{t'}{t}\right) + 2\|\mathbf{w}^T A\|\sqrt{(1+\epsilon_2)\bar{c}} + \bar{c}. \end{aligned}$$

From the above theorem, we can immediately obtain the following corollary.

Corollary 2. *Let $t' \leq t$. Suppose Condition 1 holds. Then for all any $\mathbf{w}_1, \cdots, \mathbf{w}_d \in$*

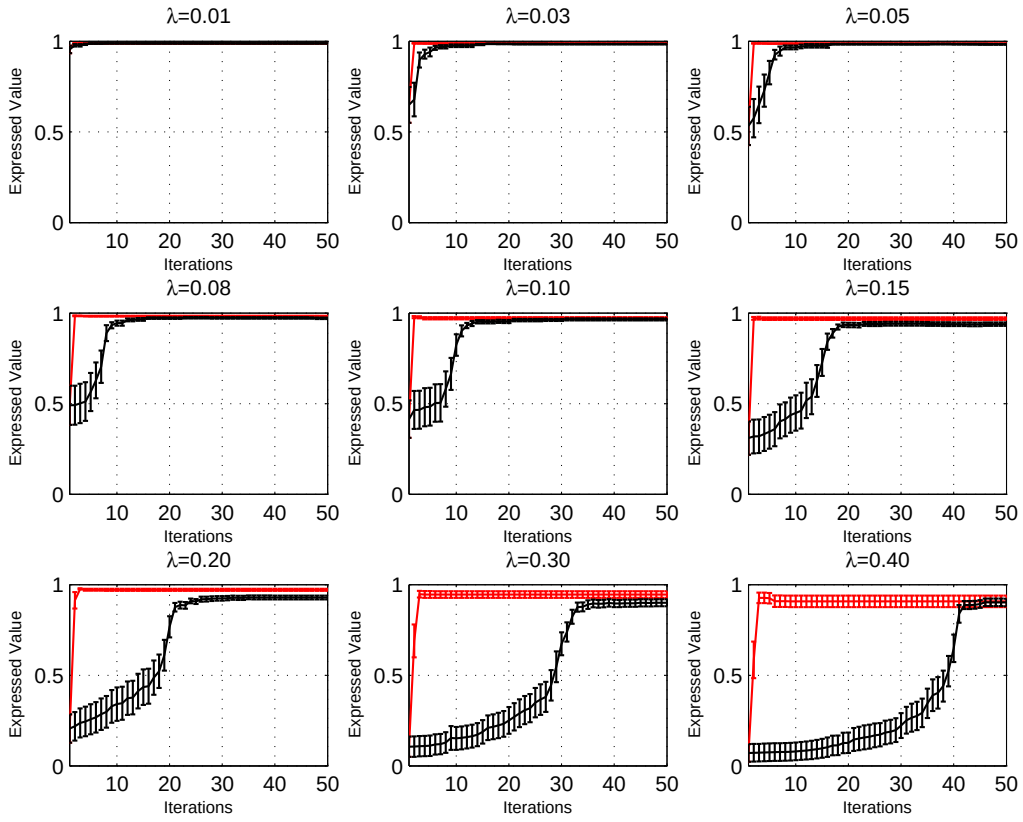


Figure 2.5: DHR-PCA (red line) vs. HR-PCA (black line). $m = n = 100, \sigma = 10$. The horizontal axis is the iteration and the vertical axis is the expressive variance value. Please refer to the color version.

$\mathcal{S}_m$ the following holds:

$$\begin{aligned} & (1-\epsilon_1)\mathcal{V}\left(\frac{t'}{t}\right) H(\mathbf{w}) - 2\sqrt{(1+\epsilon_2)\bar{c}dH(\mathbf{w})} \\ \leq & \sum_{j=1}^{d} \frac{1}{t} \sum_{i=1}^{t'} |\mathbf{w}_j^T \mathbf{z}|_{(i)}^2 \\ \leq & (1+\epsilon_1)\mathcal{V}\left(\frac{t'}{t}\right) H(\mathbf{w}) + 2\sqrt{(1+\epsilon_2)\bar{c}dH(\mathbf{w})} + \bar{c}, \end{aligned}$$

and

$$\begin{aligned} & (1-\epsilon)H(\mathbf{w}) - 2\sqrt{(1+\epsilon)\bar{c}dH(\mathbf{w})} \\ \leq & \sum_{j=1}^{d} \frac{1}{t} \sum_{i=1}^{t} |\mathbf{w}_j^T \mathbf{z}_i|^2 \\ \leq & (1+\epsilon)H(\mathbf{w}) + 2\sqrt{(1+\epsilon)\bar{c}dH(\mathbf{w})} + \bar{c}, \end{aligned}$$

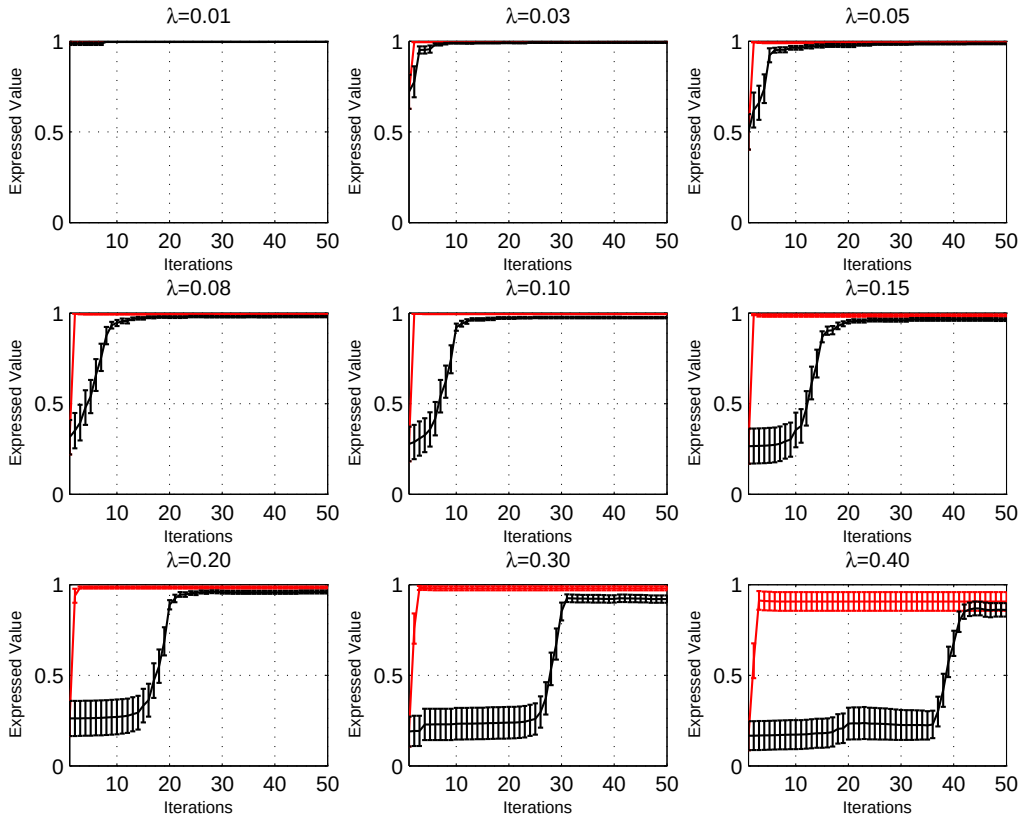


Figure 2.6: DHR-PCA (red line) vs. HR-PCA (black line). $m = n = 100, \sigma = 20$. The horizontal axis is the iteration and the vertical axis is the expressive variance value. Please refer to the color version.

where $H(\mathbf{w}) \triangleq \sum_{j=1}^{d} \|\mathbf{w}_j^T A\|^2$.

2.5.2 Finite Steps for a Good Solution

In this step, we show that the algorithm finds a good solution in a small number of steps. Proving this involves showing that at any given step, either the algorithm finds a good solution, or the weight adjusting step decreases weights of corrupted points more than the authentic points. Let $\alpha_i^{(s)}$ denote the weight of the i^{th} data point in the s^{th} stage. These points are a good solution if the variance of the points projected onto their span is mainly due to the authentic samples rather than the corrupted points. We denote this "good output event at step s" by $\mathcal{E}(s)$, defined as:

$$\mathcal{E}(s) = \quad \left\{ \sum_{i \in \mathcal{Z}} \alpha_i^{(s)} v_i(s) \geq \frac{1}{\kappa} \sum_{i \in \mathcal{O}} \alpha_i^{(s)} v_i(s) \right\},$$

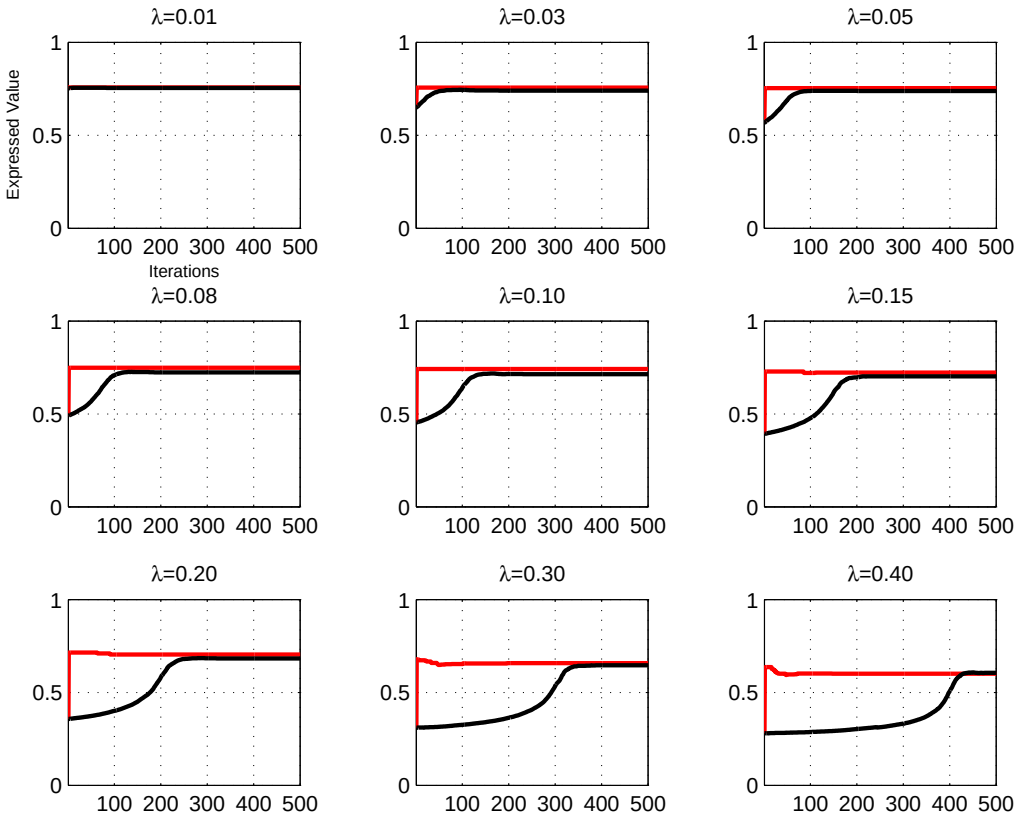


Figure 2.7: DHR-PCA (red line) vs. HR-PCA (black line). $m = n = 1000, \sigma = 2$. The horizontal axis is the iteration and the vertical axis is the expressive variance value. Please refer to the color version.

where the variance $v_i(s) = \sum_{j=1}^{d} \left(\mathbf{w}_j(s)^T \mathbf{y}_i\right)^2$. The intuition is that there cannot be too many steps without finding a good solution, since too many weights of the corrupted points will have been decreased to zero.

Theorem 5. *The event $\mathcal{E}(s)$ is true for some $1 \leq s \leq s_0$, where $s_0 \leq \frac{\lambda n(1+\kappa)}{\kappa}$.*

The proof of the above theorem is provided in the supplementary material. We compare Theorem 5 with its randomized counterpart, Theorem 9 of [61]. The latter states that for HR-PCA, $\mathcal{E}(s)$ succeeds with high probability for some $s \leq (1+\epsilon)(1+\kappa)\lambda n/\kappa$, where ϵ depends on κ and λ, and decreases to 0 when $n \uparrow \infty$ (for fixed κ and λ). Thus, the advantage of Theorem 5 is two-fold: it is deterministic as opposed to probabilistic, and it does not require the decreasing ϵ.

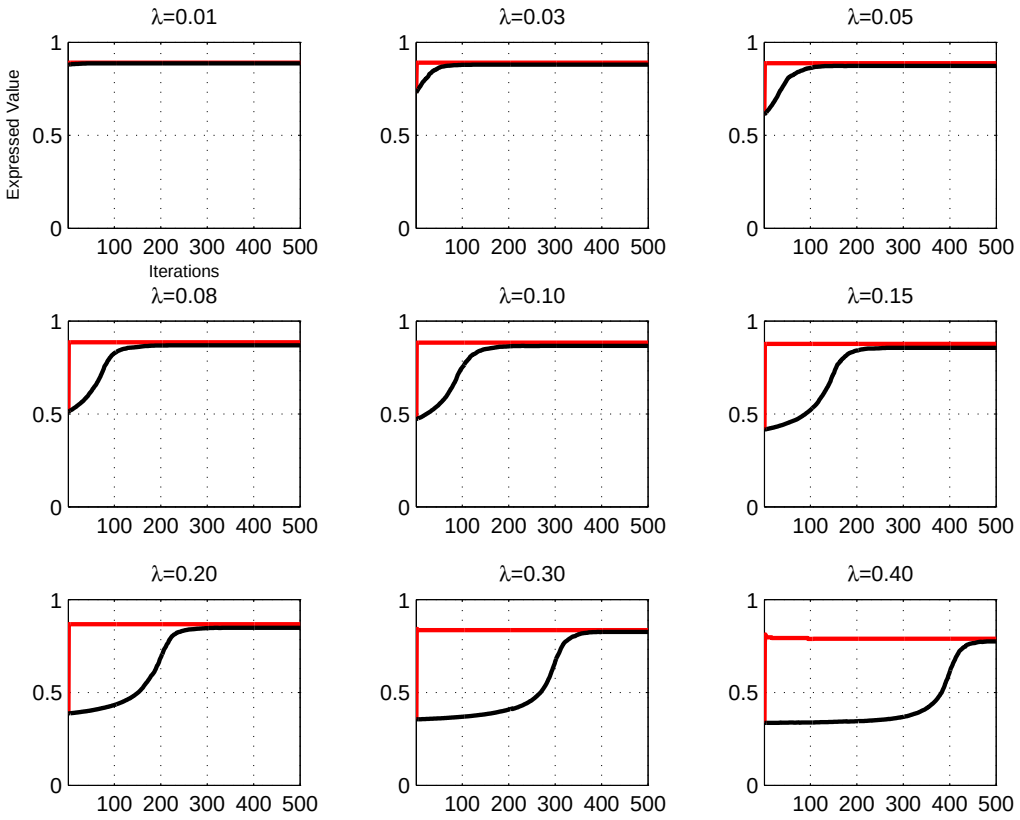


Figure 2.8: DHR-PCA (red line) vs. HR-PCA (black line). $m = n = 1000, \sigma = 3$. The horizontal axis is the iteration and the vertical axis is the expressive variance value. Please refer to the color version.

2.5.3 Bounds on the Solution Performance

Let $\bar{\mathbf{w}}_1, \ldots, \bar{\mathbf{w}}_d$ be the eigenvectors corresponding to the d largest eigenvalues of AA^T, namely the optimal solution, $\mathbf{w}_1^*, \ldots, \mathbf{w}_d^*$ be the output of the Algorithm 1 and $\mathbf{w}_1(s), \ldots, \mathbf{w}_d(s)$ be the candidate solution at stage s. We define $H(\mathbf{w}_1, \ldots, \mathbf{w}_d) \triangleq \sum_{j=1}^{d} \|\mathbf{w}_j^T A\|^2$, and for notational simplification, let $\bar{H} \triangleq H(\bar{\mathbf{w}}_1, \ldots, \bar{\mathbf{w}}_d)$, $H_s \triangleq H(\mathbf{w}_1(s), \ldots, \mathbf{w}_d(s))$, and $H^* \triangleq H(\mathbf{w}_1^*, \ldots, \mathbf{w}_d^*)$.

The statement of the finite-sample and asymptotic theorems (Theorem 1 and Theorem 2, respectively) lower bound the expressed variance, E.V., which is the ratio $H^*/\bar{H}$. The final part of the proof accomplishes this in two main steps. First, we lower bound H_s in terms of $\bar{H}$ where s is some step for which $\mathcal{E}(s)$ is true, i.e., the principal components found by the s^{th} step of the algorithm are "good". By Theorem 5, we know that there is a "small" such s. Based on the true $\mathcal{E}(s)$ and the

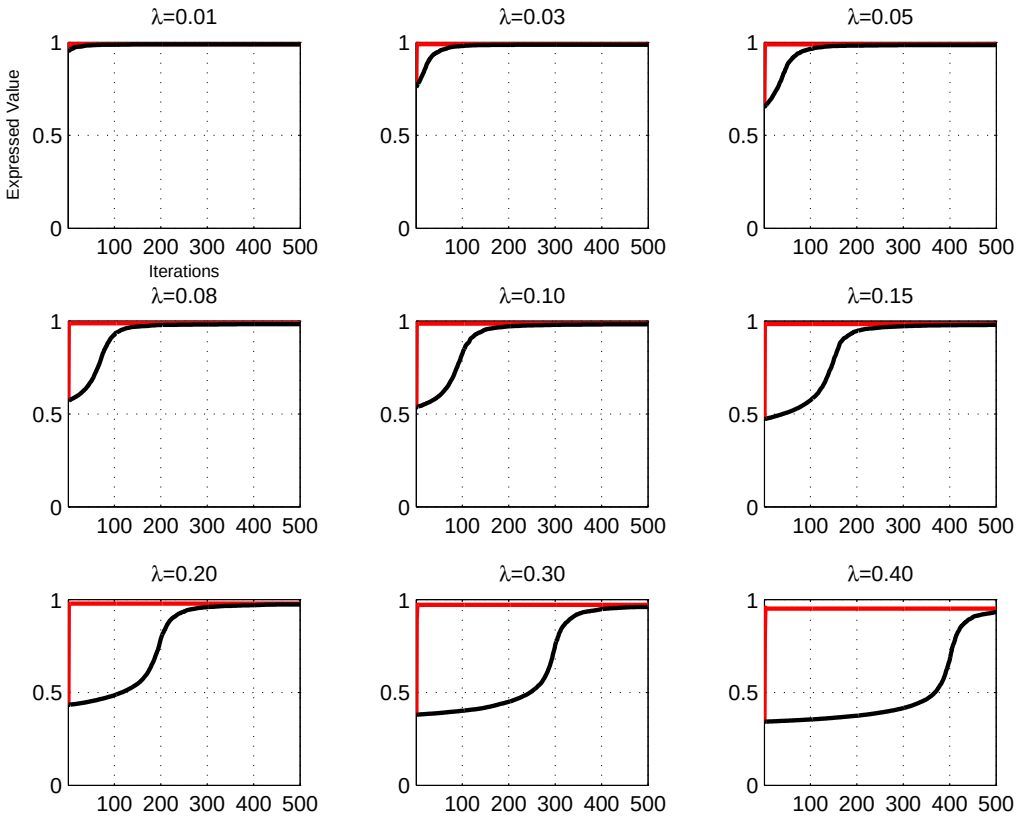


Figure 2.9: DHR-PCA (red line) vs. HR-PCA (black line). $m = n = 1000, \sigma = 10$. The horizontal axis is the iteration and the vertical axis is the expressive variance value. Please refer to the color version.

algorithm definition, we can conclude the bound via some algebraic manipulations. The final output of the algorithm, however, is only guaranteed to have a high value of the robust variance estimator, $\bar{V}$ - that is, even if there is a "good" solution at some intermediate step s, we do not necessarily have a way of identifying it. Thus, the next step lower bounds the value of H^* in terms of the value H of any output $\mathbf{w}'_1, \ldots, \mathbf{w}'_d$ that has a smaller value of the robust variance estimator. The details of these two steps are deferred to the supplementary material. Combining the results of above two steps, we can obtain the following theorem providing a lower bound of the ratio $H^*/\bar{H}$, i.e., the expressed variance.

Theorem 6. *If* $\bigcup_{s=1}^{s_0} \mathcal{E}(s)$ *is true, and there exist* $\epsilon_1 < 1$, ϵ_2, $\bar{c}$ *such that*

$$\sup_{\mathbf{w} \in \mathcal{S}_d} \left| \frac{1}{t} \sum_{i=1}^{t-s_0} |\mathbf{w}^T \mathbf{x}|^2_{(i)} - \mathcal{V}\left(\frac{t-s_0}{t}\right) \right| \leq \epsilon_1$$

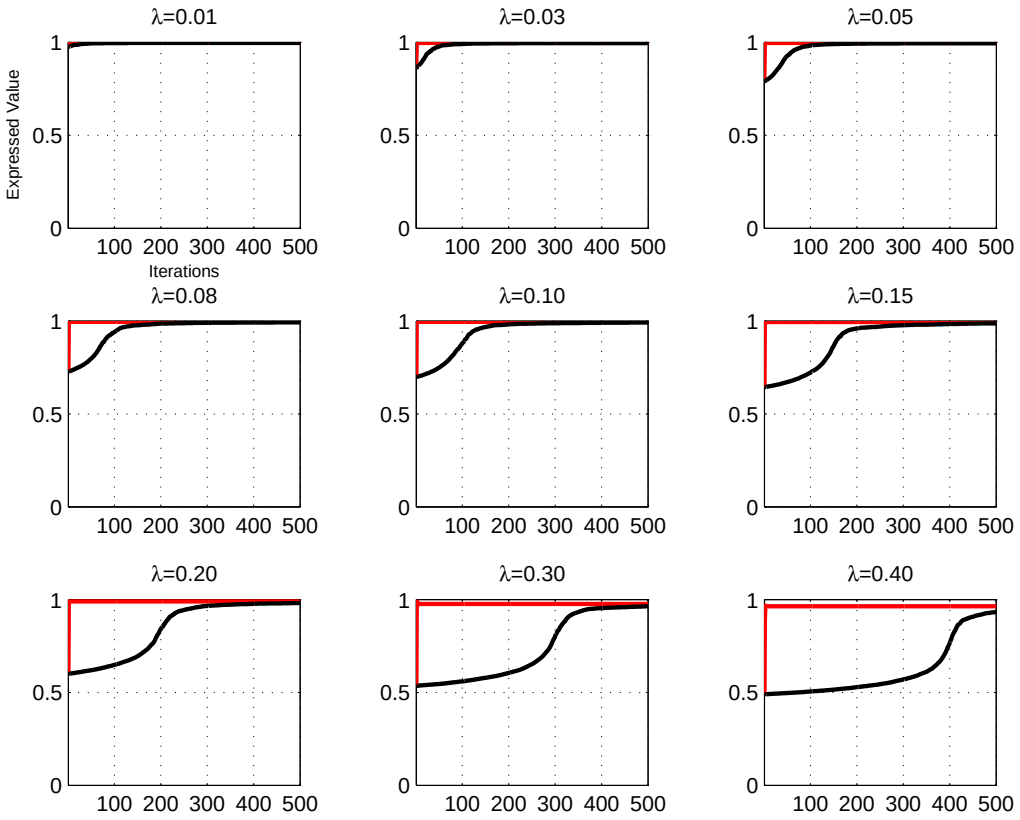


Figure 2.10: DHR-PCA (red line) vs. HR-PCA (black line). $m = n = 1000, \sigma = 20$. The horizontal axis is the iteration and the vertical axis is the expressive variance value. Please refer to the color version.

and Condition 1 holds, then

$$\begin{aligned} \frac{H^*}{\bar{H}} \quad \geq \quad & \frac{(1-\epsilon_1)^2 \mathcal{V}\left(\frac{\hat{t}}{t} - \frac{\lambda}{1-\lambda}\right) \mathcal{V}\left(\frac{t-s_0}{t}\right)}{(1+\epsilon_1)(1+\epsilon_2)(1+\kappa)\mathcal{V}\left(\frac{\hat{t}}{t}\right)} \\ - \quad & \left[\frac{(D_1+D_2)\sqrt{(1+\epsilon_2)\bar{c}d}}{(1+\epsilon_1)(1+\epsilon_2)(1+\kappa)}\right](\bar{H})^{-1/2} \\ - \quad & \left[\frac{(1-\epsilon_1)\mathcal{V}\left(\frac{\hat{t}}{t} - \frac{\lambda}{1-\lambda}\right)\bar{c} + (1+\epsilon_2)\bar{c}}{(1+\epsilon_1)(1+\epsilon_2)\mathcal{V}\left(\frac{\hat{t}}{t}\right)}\right](\bar{H})^{-1}, \end{aligned} \tag{2.3}$$

where $D_1 = (2\kappa+4)(1-\epsilon_1)\mathcal{V}\left(\frac{\hat{t}}{t} - \frac{\lambda}{1-\lambda}\right)$ *and* $D_2 = 4(1+\epsilon_2)(1+\kappa)$.

By bounding all diminishing terms in the right hand side of (2.5), it reduces to Theorem 1. And Theorem 2 follows immediately. The proofs of Theorem 7 and Theorem 1 are similar to those in [61] and we omit it here.

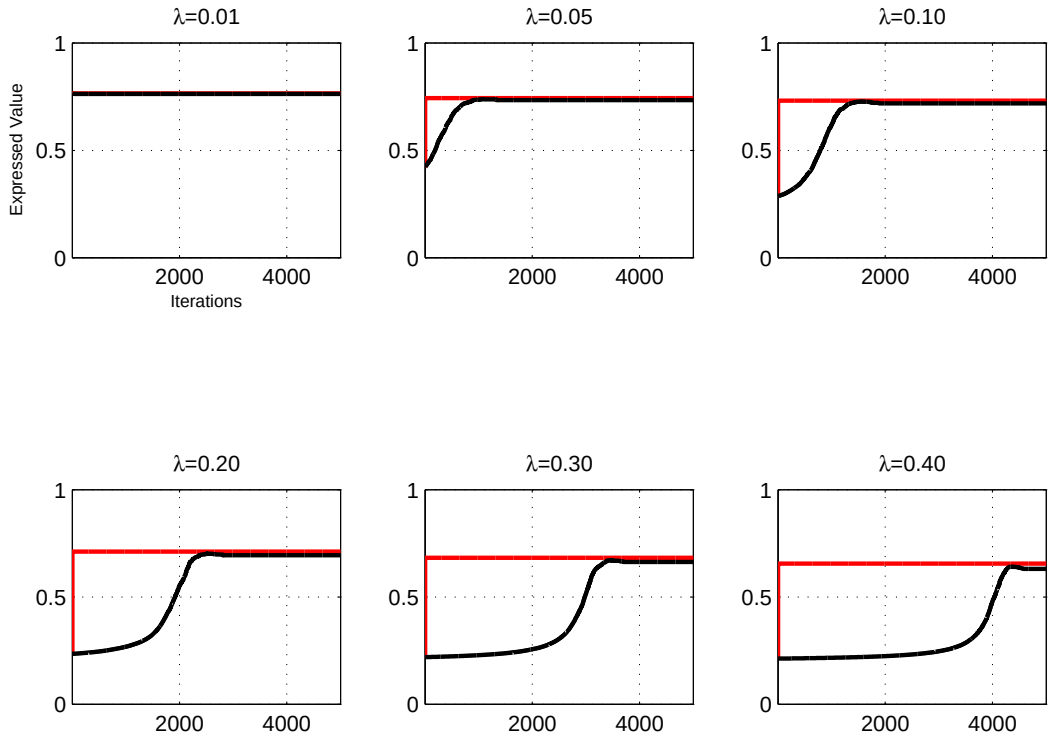


Figure 2.11: DHR-PCA (red line) vs. HR-PCA (black line). $m = n = 10000, \sigma = 2$. The horizontal axis is the iteration and the vertical axis is the expressive variance value. Please refer to the color version.

2.6 Proof of Corollary 1

Lemma 1. *For any $\epsilon > 0$ and $\kappa \in [\epsilon, 1]$, we have $\mathcal{V}(\kappa) - \mathcal{V}(\kappa - \epsilon) \leq C\alpha\epsilon \log^2(1/\epsilon)$.*

Proof. By monotonicity, it suffices to prove that result for $\kappa = 1$. Notice that for $K \geq 2\alpha$,

$$\begin{aligned}
& \mathcal{V}(1) - \mathcal{V}(1-\epsilon) \\
\leq\ & \epsilon K^2 + \mathbb{E}_{x\sim\bar{\mu}}\left(x^2 \cdot \mathbf{1}(x > K)\right) \\
=\ & \epsilon K^2 + \int_{K^2}^{\infty} \Pr_{x\sim\bar{\mu}}(x^2 > z)dz \\
\leq\ & \epsilon K^2 + \int_{K^2}^{\infty} \exp(1 - \sqrt{z}/\alpha)dz \\
=\ & \epsilon K^2 + e_0 \int_{K^2/4\alpha^2}^{\infty} \exp(-2\sqrt{z})dz \\
\overset{(a)}{\leq}\ & \epsilon K^2 + 2e_0 \exp(-\sqrt{z})|_{\infty}^{K^2/4\alpha^2} \\
=\ & \epsilon K^2 + \exp(1 + \ln 2 - K/2\alpha),
\end{aligned}$$

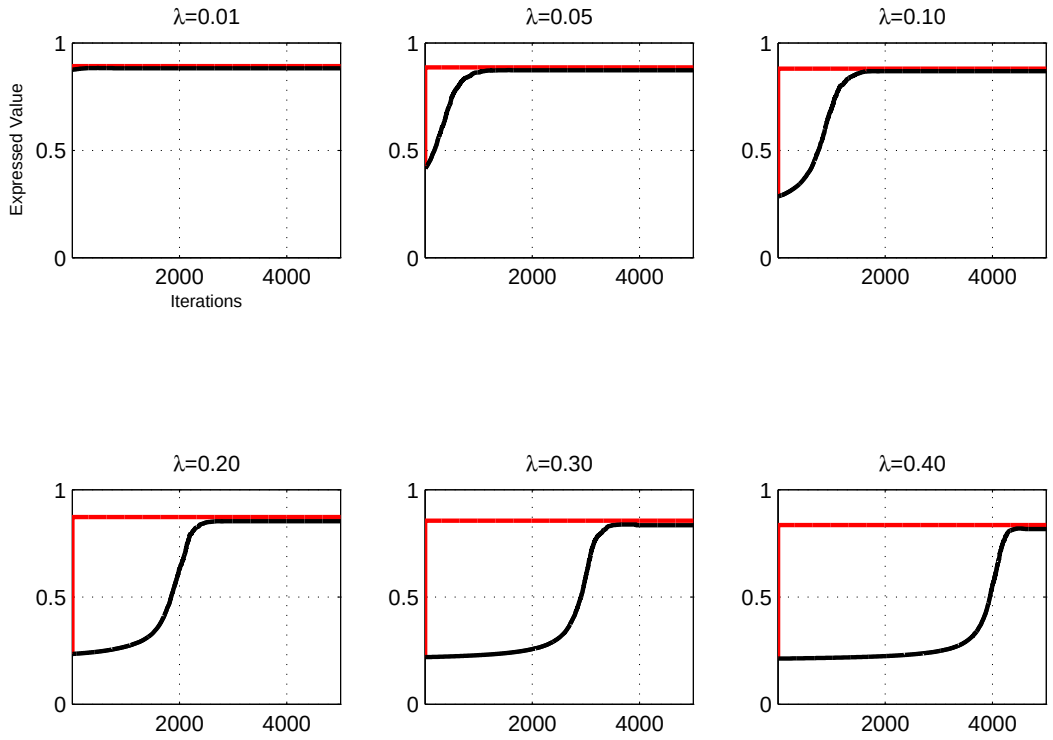


Figure 2.12: DHR-PCA (red line) vs. HR-PCA (black line). $m = n = 10000, \sigma = 3$. The horizontal axis is the iteration and the vertical axis is the expressive variance value. Please refer to the color version.

where (a) holds because when $z \geq 1$, we have $\exp(-\sqrt{z}) \leq 1/\sqrt{z}$, which implies $\exp(-2\sqrt{z}) \leq \frac{d(2\exp(-\sqrt{z}))}{dz}$. Pick $K = 2\alpha \log(1/\epsilon)$, we have that

$$\mathcal{V}(1) - \mathcal{V}(1-c) \leq C\alpha\epsilon \log^2(1/\epsilon).$$

□

Corollary 3. *1 Under the settings of the above theorem, the following holds in probability when $j \uparrow \infty$ (i.e., when $n, p \uparrow \infty$),*

$$\liminf_j \text{E.V.}\{\mathbf{w}_1(j), \ldots, \mathbf{w}_d(j)\} \geq 1 - \frac{C'\sqrt{\alpha\lambda^* \log(1/\lambda^*)}}{\mathcal{V}(0.5)}.$$

Proof. We bound the right-hand-side of Equation (2) to establish the corollary.

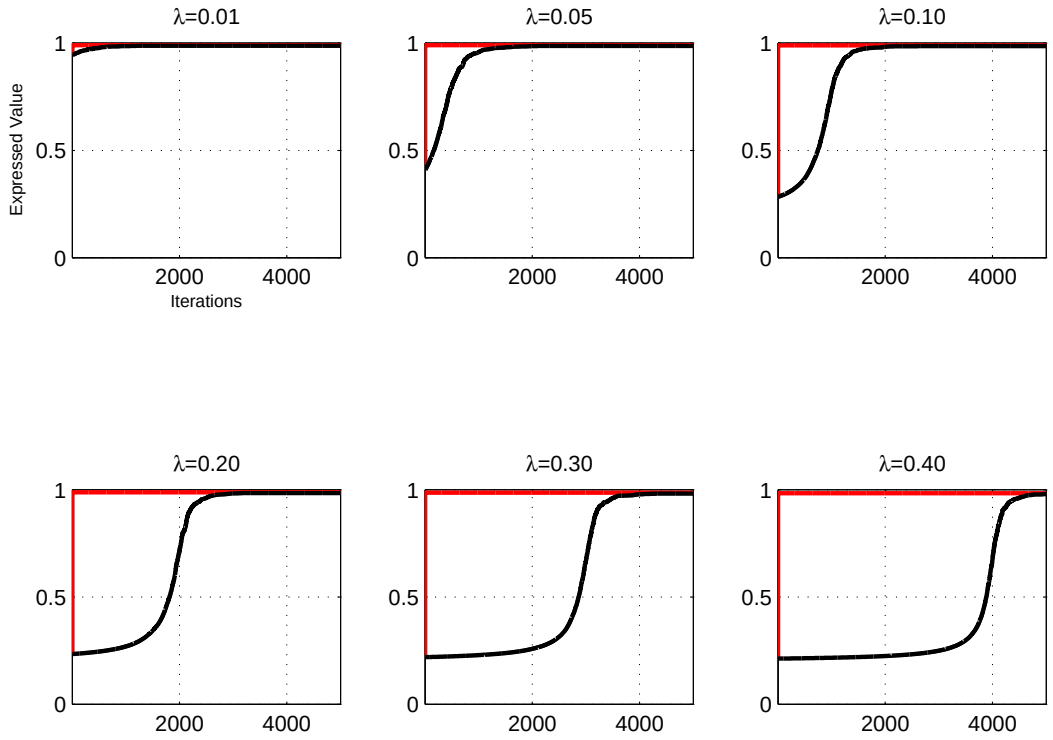


Figure 2.13: DHR-PCA (red line) vs. HR-PCA (black line). $m = n = 10000, \sigma = 10$. The horizontal axis is the iteration and the vertical axis is the expressive variance value. Please refer to the color version.

Notice that

$$
\begin{aligned}
& \left[\frac{\mathcal{V}\left(1-\frac{\lambda^*(1+\kappa)}{(1-\lambda^*)\kappa}\right)}{(1+\kappa)}\right] \times \left[\frac{\mathcal{V}\left(\frac{\hat{t}}{t}-\frac{\lambda^*}{1-\lambda^*}\right)}{\mathcal{V}\left(\frac{\hat{t}}{t}\right)}\right] \\
\overset{(a)}{\geq} & \left[\frac{\mathcal{V}(1)-C\alpha\frac{\lambda^*(1+\kappa)}{(1-\lambda^*)\kappa}\log^2\left(\frac{(1-\lambda^*)\kappa}{\lambda^*(1+\kappa)}\right)}{(1+\kappa)}\right] \times \left[\frac{\mathcal{V}\left(\frac{\hat{t}}{t}\right)-C\alpha\frac{\lambda^*}{1-\lambda^*}\log^2\left(\frac{1-\lambda^*}{\lambda^*}\right)}{\mathcal{V}\left(\frac{\hat{t}}{t}\right)}\right] \\
\overset{(b)}{\geq} & \left[\frac{1}{1+\kappa}-\frac{C\alpha\lambda^*}{(1-\lambda^*)\kappa}\log^2\left(\frac{(1-\lambda^*)\kappa}{\lambda^*(1+\kappa)}\right)\right] \times \left[1-\frac{C\alpha\frac{\lambda^*}{1-\lambda^*}\log^2\left(\frac{1-\lambda^*}{\lambda^*}\right)}{\mathcal{V}\left(\frac{\hat{t}}{t}\right)}\right] \\
\overset{(c)}{\geq} & \left[1-\kappa-\frac{2C\alpha\lambda^*}{\kappa}\log^2\left(\frac{1}{\lambda^*}\right)\right] \times \left[1-\frac{2C\alpha\lambda^*\log^2\left(\frac{1}{\lambda^*}\right)}{\mathcal{V}(0.5)}\right] \\
\geq & 1-\kappa-\frac{C'\alpha\lambda^*}{\kappa}\log^2\left(\frac{1}{\lambda^*}\right)-\frac{C'\alpha\lambda^*\log^2\left(\frac{1}{\lambda^*}\right)}{\mathcal{V}(0.5)} \\
\overset{(d)}{\geq} & 1-\kappa-\frac{2C'\alpha\lambda^*}{\kappa\mathcal{V}(0.5)}\log^2\left(\frac{1}{\lambda^*}\right).
\end{aligned}
$$

Here, (a) is due to Lemma 1; (b) is due to $\mathcal{V}(1) = 1$; (c) holds because $\frac{1}{1+\kappa} \geq 1-\kappa$,

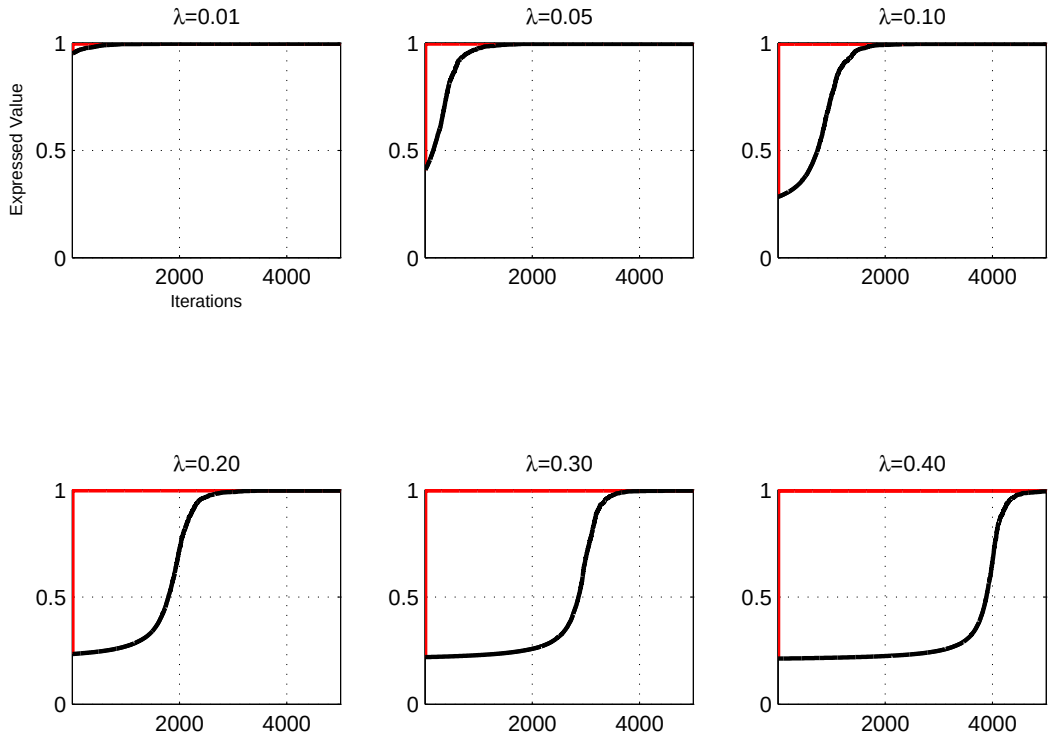


Figure 2.14: DHR-PCA (red line) vs. HR-PCA (black line). $m = n = 10000, \sigma = 20$. The horizontal axis is the iteration and the vertical axis is the expressive variance value. Please refer to the color version.

$1 - \lambda^* \geq 1/2$, and $\mathcal{V}(\hat{t}/t) \geq \mathcal{V}(0.5)$; (d) holds because κ and $\mathcal{V}(0.5)$ are both smaller than or equal to 1. □

2.7 Proof of Theorem 5

Proof. If $\mathcal{E}^c(s)$ is true, then

$$\sum_{j=1}^{d} \sum_{i \in \mathcal{Z}} \alpha_i^{(s)} \left(\mathbf{w}_j(s)^T \mathbf{y}_i \right)^2 < \frac{1}{\kappa} \sum_{j=1}^{d} \sum_{i \in \mathcal{O}} \alpha_i^{(s)} \left(\mathbf{w}_j(s)^T \mathbf{y}_i \right)^2 .$$

Since $\Delta\alpha_i^{(s)} = \eta \alpha_i^{(s)} \sum_{j=1}^{d} \left(\mathbf{w}_j(s)^T \hat{\mathbf{y}}_i \right)^2$, we have

$$\sum_{i \in \mathcal{Z}} \Delta\alpha_i^{(s)} < \frac{1}{\kappa} \sum_{i \in \mathcal{O}} \Delta\alpha_i^{(s)}.$$

If $\bigcap_{s=1}^{s_0} \mathcal{E}^c(s)$ is true,

$$\sum_{s=1}^{s_0} \sum_{i \in \mathcal{Z}} \Delta\alpha_i^{(s)} < \frac{1}{\kappa} \sum_{s=1}^{s_0} \sum_{i \in \mathcal{O}} \Delta\alpha_i^{(s)}.$$

In the Algorithm 1, we eliminate at least one weight coefficient in each iteration. Therefore, to step s_0, we have $\sum_{s=1}^{s_0} \sum_i \Delta\alpha_i^{(s)} \geq s_0$. Namely,

$$\sum_{s=1}^{s_0} \sum_{i \in \mathcal{Z}} \Delta\alpha_i^{(s)} + \sum_{s=1}^{s_0} \sum_{i \in \mathcal{O}} \Delta\alpha_i^{(s)} \geq s_0.$$

Thus,

$$\frac{1}{\kappa} \sum_{s=1}^{s_0} \sum_{i \in \mathcal{O}} \Delta\alpha_i^{(s)} + \sum_{s=1}^{s_0} \sum_{i \in \mathcal{O}} \Delta\alpha_i^{(s)} \geq s_0.$$

From the above inequality, we can obtain

$$\lambda n \geq \sum_{s=1}^{s_0} \sum_{i \in \mathcal{O}} \Delta\alpha_i^{(s)} \geq \frac{s_0 \kappa}{1+\kappa}.$$

Therefore, we can conclude bound $s_0 \leq \frac{\lambda n(1+\kappa)}{\kappa}$. □

2.8 Proof of Theorem 7

As stated in the main body, our proof comprises following two steps.

Lemma 2. *If $\mathcal{E}(s)$ is true for some $s \leq s_0$, and there exist ϵ_1 such that*

$$\sup_{\mathbf{w} \in \mathcal{S}_d} \left| \frac{1}{t} \sum_{i=1}^{t-s_0} |\mathbf{w}^T \mathbf{x}|_{(i)}^2 - \mathcal{V}\left(\frac{t-s_0}{t}\right) \right| \leq \epsilon_1$$

and $\epsilon_2, \bar{c}$ satisfying conditions (II) and (III) in Theorem 4, then

$$\frac{1}{1+\kappa} \left[(1-\epsilon_1) \mathcal{V}\left(\frac{t-s_0}{t}\right) \bar{H} - 2\sqrt{(1+\epsilon_2)\bar{c}d\bar{H}} \right] \leq (1+\epsilon_2) H_s + 2\sqrt{(1+\epsilon_2)\bar{c}dH_s} + \bar{c}.$$

Proof. If $\mathcal{E}(s)$ is true, then we have

$$\sum_{j=1}^{d}\sum_{i=1}^{t}\alpha_i^{(s)}\left(\mathbf{w}_j(s)^T\mathbf{z}_i\right)^2 \geq \frac{1}{\kappa}\sum_{j=1}^{d}\sum_{i=1}^{n-t}\alpha_i^{(s)}\left(\mathbf{w}_j(s)^T\mathbf{o}_i\right)^2.$$

Thus we have

$$\frac{1}{1+\kappa}\sum_{j=1}^{d}\sum_{i=1}^{n}\alpha_i\left(\mathbf{w}_j(s)^T\mathbf{y}_i\right)^2 \leq \sum_{j=1}^{d}\sum_{i=1}^{t}\alpha_i\left(\mathbf{w}_j(s)^T\mathbf{z}_i\right)^2.$$

Since $\mathbf{w}_1(s),\ldots,\mathbf{w}_d(s)$ is the solution of the s^{th} stage, the following holds by definition of the algorithm

$$\sum_{j=1}^{d}\sum_{i=1}^{n}\alpha_i\left(\bar{\mathbf{w}}_j^T\mathbf{y}_i\right)^2 \leq \sum_{j=1}^{d}\sum_{i=1}^{n}\alpha_i\left(\mathbf{w}_j(s)^T\mathbf{y}_i\right)^2.$$

Since $0\leq\alpha_i\leq 1, \forall i=1,\ldots,n$, we have

$$\sum_{j=1}^{d}\sum_{i=1}^{n}\alpha_i\left(\mathbf{w}_j(s)^T\mathbf{y}_i\right)^2 \leq \sum_{j=1}^{d}\sum_{i=1}^{n}\left(\mathbf{w}_j(s)^T\mathbf{y}_i\right)^2.$$

Since $1\leq s\leq s_0$, from the definition of the algorithm, we have $\sum_{i\in\mathcal{Z}}\alpha_i\geq t-s_0$. Thus

$$\begin{aligned}
&\sum_{i=1}^{t}\alpha_i\left(\bar{\mathbf{w}}_j{}^T\mathbf{z}_i\right)^2 - \sum_{i=1}^{t-s_0}\left|\bar{\mathbf{w}}_j{}^T\mathbf{z}\right|_{(i)}^2\\
=&\sum_{i=1}^{t-s_0}\left[\alpha_{(i)}-1\right]\left|\bar{\mathbf{w}}_j{}^T\mathbf{z}\right|_{(i)}^2 + \sum_{i=t-s_0+1}^{t}\alpha_{(i)}\left|\bar{\mathbf{w}}_j{}^T\mathbf{z}\right|_{(i)}^2\\
\geq&\sum_{i=1}^{t-s_0}\left[\alpha_{(i)}-1\right]\left|\bar{\mathbf{w}}_j{}^T\mathbf{z}\right|_{(t-s_0)}^2 + \sum_{i=t-s_0+1}^{t}\alpha_{(i)}\left|\bar{\mathbf{w}}_j{}^T\mathbf{z}\right|_{(t-s_0)}^2\\
=&\left[\sum_{i=1}^{t}\alpha_{(i)}-(t-s_0)\right]\left|\bar{\mathbf{w}}_j{}^T\mathbf{z}\right|_{(t-s_0)}^2\\
\geq&\ 0.
\end{aligned}$$

Thus we have

$$\sum_{j=1}^{d}\sum_{i=1}^{t-s_0}\left|\bar{\mathbf{w}}_j{}^T\mathbf{z}\right|_{(i)}^2 \le \sum_{j=1}^{d}\sum_{i=1}^{t}\alpha_i\left(\bar{\mathbf{w}}_j{}^T\mathbf{z}_i\right)^2$$
$$\le \sum_{j=1}^{d}\sum_{i=1}^{n}\alpha_i\left(\bar{\mathbf{w}}_j{}^T\mathbf{y}_i\right)^2.$$

Combining the above inequalities, we get

$$\frac{1}{1+\kappa}\sum_{j=1}^{d}\sum_{i=1}^{t-s_0}\left|\bar{\mathbf{w}}_j{}^T\mathbf{z}\right|_{(i)}^2 \le \sum_{j=1}^{d}\sum_{i=1}^{t}\left(\mathbf{w}_j(s)^T\mathbf{z}_i\right)^2.$$

By Corollary 1 we complete the proof. □

The following lemma guarantees that the value H^* of the algorithm's output is lower bounded in term of the value H of any output that has a smaller value of the robust variance estimator.

Lemma 3. *Fix a $\hat{t} \le t$. If $\sum_{j=1}^{d}\bar{V}_{\hat{t}}(\mathbf{w}'_j) \ge \sum_{j=1}^{d}\bar{V}_{\hat{t}}(\mathbf{w}_j)$, and there exists ϵ_1, ϵ_2 and $\bar{c}$ such that $\sup_{\mathbf{w}\in\mathcal{S}_d}\left|\frac{1}{t}\sum_{i=1}^{\hat{t}-\frac{\lambda t}{1-\lambda}}\left|\mathbf{w}^T\mathbf{x}\right|_{(i)}^2 - \mathcal{V}\left(\frac{\hat{t}}{t}-\frac{\lambda}{1-\lambda}\right)\right| \le \epsilon_1$ and conditions in Theorem 4 are satisfied, then*

$$(1-\epsilon_1)\mathcal{V}\left(\frac{\hat{t}}{t}-\frac{\lambda}{1-\lambda}\right)H(\mathbf{w}') - 2\sqrt{(1+\epsilon_2)\bar{c}dH(\mathbf{w}')}$$
$$\le (1+\epsilon_1)H(\mathbf{w})\mathcal{V}\left(\frac{\hat{t}}{t}\right) + 2\sqrt{(1+\epsilon_2)\bar{c}dH(\mathbf{w})} + \bar{c}.$$

Theorem 7. *If $\bigcup_{s=1}^{s_0}\mathcal{E}(s)$ is true, and there exist $\epsilon_1 < 1$, ϵ_2, $\bar{c}$ such that*

$$\sup_{\mathbf{w}\in\mathcal{S}_d}\left|\frac{1}{t}\sum_{i=1}^{t-s_0}\left|\mathbf{w}^T\mathbf{x}\right|_{(i)}^2 - \mathcal{V}\left(\frac{t-s_0}{t}\right)\right| \le \epsilon_1,$$

and Condition 1 holds, then

$$\begin{aligned}\frac{H^*}{\bar{H}} \quad &\geq \quad \frac{(1-\epsilon_1)^2\mathcal{V}\left(\frac{\hat{t}}{t}-\frac{\lambda}{1-\lambda}\right)\mathcal{V}\left(\frac{t-s_0}{t}\right)}{(1+\epsilon_1)(1+\epsilon_2)(1+\kappa)\mathcal{V}\left(\frac{\hat{t}}{t}\right)} \\ &- \quad \left[\frac{\left((2\kappa+4)(1-\epsilon_1)\mathcal{V}\left(\frac{\hat{t}}{t}-\frac{\lambda}{1-\lambda}\right)+4(1+\epsilon_2)(1+\kappa)\right)\sqrt{(1+\epsilon_2)\bar{c}\bar{d}}}{(1+\epsilon_1)(1+\epsilon_2)(1+\kappa)}\right](\bar{H})^{-1/2} \\ &- \quad \left[\frac{(1-\epsilon_1)\mathcal{V}\left(\frac{\hat{t}}{t}-\frac{\lambda}{1-\lambda}\right)\bar{c}+(1+\epsilon_2)\bar{c}}{(1+\epsilon_1)(1+\epsilon_2)\mathcal{V}\left(\frac{\hat{t}}{t}\right)}\right](\bar{H})^{-1}, \qquad (2.4)\end{aligned}$$

Proof. Since $\bigcup_{s=1}^{s_0}\mathcal{E}(s)$ is true, there exists a $s' \leq s_0$ such that $\mathcal{E}(s')$ is true. By Lemma 2 we have

$$\frac{1}{1+\kappa}\left[(1-\epsilon_1)\mathcal{V}\left(\frac{t-s_0}{t}\right)\bar{H}-2\sqrt{(1+\epsilon_2)\bar{c}\bar{d}\bar{H}}\right] \leq (1+\epsilon_2)H_{s'}+2\sqrt{(1+\epsilon_2)\bar{c}\bar{d}H_{s'}}+\bar{c}.$$

By the definition of the algorithm, we have $\sum_{j=1}^{d}\bar{V}_{\hat{t}}(\mathbf{w}_j^*) \geq \sum_{j=1}^{d}\bar{V}_{\hat{t}}(\mathbf{w}_j(s'))$, which by Lemma 3 implies

$$(1-\epsilon_1)\mathcal{V}\left(\frac{\hat{t}}{t}-\frac{\lambda}{1-\lambda}\right)H_{s'}-2\sqrt{(1+\epsilon_2)\bar{c}\bar{d}H_{s'}} \leq (1+\epsilon_1)H^*\mathcal{V}\left(\frac{\hat{t}}{t}\right)+2\sqrt{(1+\epsilon_2)\bar{c}\bar{d}H^*}+\bar{c}.$$

By definition, $H_{s'}, H^* \leq \bar{H}$. Thus we have

$$\begin{aligned}(I) \qquad &\frac{1}{1+\kappa}\left[(1-\epsilon_1)\mathcal{V}\left(\frac{t-s_0}{t}\right)\bar{H}-2\sqrt{(1+\epsilon_2)\bar{c}\bar{d}\bar{H}}\right] \\ &\leq (1+\epsilon_2)H_{s'}+2\sqrt{(1+\epsilon_2)\bar{c}\bar{d}\bar{H}}+\bar{c}; \\ (II) \qquad &(1-\epsilon_1)\mathcal{V}\left(\frac{\hat{t}}{t}-\frac{\lambda}{1-\lambda}\right)H_{s'}-2\sqrt{(1+\epsilon_2)\bar{c}\bar{d}\bar{H}} \\ &\leq (1+\epsilon_1)H^*\mathcal{V}\left(\frac{\hat{t}}{t}\right)+2\sqrt{(1+\epsilon_2)\bar{c}\bar{d}\bar{H}}+\bar{c}.\end{aligned}$$

Rearrange the inequalities, we have

$$\begin{aligned}(I) \qquad &(1-\epsilon_1)\mathcal{V}\left(\frac{t-s_0}{t}\right)\bar{H}-(2\kappa+4)\sqrt{(1+\epsilon_2)\bar{c}\bar{d}\bar{H}}-(1+\kappa)\bar{c} \leq (1+\kappa)(1+\epsilon_2)H_{s'}; \\ (II) \qquad &(1-\epsilon_1)\mathcal{V}\left(\frac{\hat{t}}{t}-\frac{\lambda}{1-\lambda}\right)H_{s'} \leq (1+\epsilon_1)\mathcal{V}\left(\frac{\hat{t}}{t}\right)H^*+4\sqrt{(1+\epsilon_2)\bar{c}\bar{d}\bar{H}}+\bar{c}.\end{aligned}$$

Simplify the inequality, we get

$$\begin{aligned}
\frac{H^*}{\bar{H}} &\geq \frac{(1-\epsilon_1)^2 \mathcal{V}\left(\frac{\hat{t}}{t}-\frac{\lambda}{1-\lambda}\right)\mathcal{V}\left(\frac{t-s_0}{t}\right)}{(1+\epsilon_1)(1+\epsilon_2)(1+\kappa)\mathcal{V}\left(\frac{\hat{t}}{t}\right)} \\
&- \left[\frac{\left((2\kappa+4)(1-\epsilon_1)\mathcal{V}\left(\frac{\hat{t}}{t}-\frac{\lambda}{1-\lambda}\right)+4(1+\epsilon_2)(1+\kappa)\right)\sqrt{(1+\epsilon_2)\bar{c}d}}{(1+\epsilon_1)(1+\epsilon_2)(1+\kappa)}\right](\bar{H})^{-1/2} \\
&- \left[\frac{(1-\epsilon_1)\mathcal{V}\left(\frac{\hat{t}}{t}-\frac{\lambda}{1-\lambda}\right)\bar{c}+(1+\epsilon_2)\bar{c}}{(1+\epsilon_1)(1+\epsilon_2)\mathcal{V}\left(\frac{\hat{t}}{t}\right)}\right](\bar{H})^{-1}, \qquad (2.5)
\end{aligned}$$

□

2.9 Chapter Summary

In this chapter, we proposed a deterministic robust PCA algorithm for high-dimensional data corrupted by arbitrary outliers. The algorithm alternates between a classical PCA and decrease of weight coefficients on all the data points. Theoretical analysis showed that the proposed algorithm is tractable, robust to corrupt points, easily kernelizable, asymptotic consistent and achieving maximal breakdown point of 50% – to the best of our knowledge, the first *deterministic* algorithm that achieves these properties in the high-dimensional setup. More importantly, simulation results demonstrated that the proposed algorithm improves computational efficiency over its randomized counterpart HR-PCA – indeed, the number of iterations required to find a satisfactory solution appears to approximate constant, in sharp contrast to HR-PCA whose number of iterations required increases linearly with the number of outliers.

Chapter 3

Online PCA for Contaminated Data

In the above chapter, we introduce a batch robust PCA method performing well in high-dimensional regime. In this chapter, we propose an online robust PCA method for handing big data with limited computer memory budget.

3.1 Introduction

We investigate the problem of robust **P**rincipal **C**omponent **A**nalysis (PCA) in an online fashion. PCA aims to construct a low-dimensional subspace based on a set of principal components (PCs) to approximate all the observed samples in the least-square sense [57]. Conventionally, it computes PCs as eigenvectors of the sample covariance matrix in *batch mode*, which is both computationally expensive and in particular memory exhausting, when dealing with large scale data. To address this problem, several online PCA algorithms have been developed in literature [53, 38, 26]. For online PCA, at each time instance, a new sample is revealed, and the PCs estimation is updated accordingly without having to re-explore all previous samples. The significant advantages of online PCA algorithms include independence of their storage space requirement of the number of samples, and handling newly revealed samples quite efficiently.

Due to the quadratic error criterion, PCA is notoriously sensitive to corrupted observations (outliers), and the quality of its output can suffer severely in the face of even a few outliers. Therefore, many works have been dedicated to robustifying PCA [52, 44, 61, 47]. However, all of these methods work in batch mode and cannot handle the sequentially revealed samples in the online learning framework. For instance, [61] proposed a high-dimensional robust PCA (HR-PCA) algorithm that is based on iterative performing PCA and randomized removal. Notice that the random removal process involves calculating the order statistics over all the samples to obtain the removal probability. Therefore, all samples must be stored in memory throughout the process. This hinders its application to large scale data, for which storing all data is impractical.

In this chapter, we propose a novel online Robust PCA algorithm to handle contaminated sample set, i.e., sample set that comprises both authentic samples (non-corrupted samples) and outliers (corrupted samples), which are revealed sequentially to the algorithm. Previous online PCA algorithms generally fail in this case, since they update the PCs estimation through minimizing the quadratic error w.r.t. every new sample and are thus sensitive to outliers. The outliers may manipulate the PCs estimation severely and the result can be arbitrarily bad. In contrast, the proposed online RPCA is shown to be robust to the outliers. This is achieved by a probabilistic admittion/rejection procedure when a new sample comes. This is different from previous online PCA methods, where each and every new sample is admitted. The probabilistic admittion/rejection procedure endows online RPCA with the ability to reject more outliers than authentic samples and thus alleviates the affect of outliers and robustifies the PCs estimation. Indeed, we show that given a proper initial estimation, online RPCA is able to steadily improve its output until convergence. We further bound the deviation of the final output from the optimal solution. In fact, under mild conditions, online RPCA can be resistent to 50% outliers, namely having a 50% breakdown point. This is the maximal robustness that can be achieved by any method.

Compared with previous robust PCA methods (typically works in batch mode),

online RPCA only needs to maintain a covariance matrix whose size is independent of the number of data points. Upon accepting a newly revealed sample, online RPCA updates the PCs estimation accordingly without re-exploring the previous samples. Thus, online RPCA can deal with large amounts of data with low *storage expense*. This is in stark contrast with previous robust PCA methods which typically requires to remember all samples. To the best of our knowledge, this is the first attempt to make online PCA work for outlier-corrupted data, with theoretical performance guarantees.

3.2 Related Work

Standard PCA is performed in batch mode, and its high computational complexity may become cumbersome for the large datasets. To address this issue, different online learning techniques have been proposed, for example [17, 24], and many others.

Most of current online PCA methods perform the PCs estimation in an incremental manner [24, 33, 40]. They maintain a covariance matrix or current PCs estimation, and update it according to the new sample incrementally. Those methods provide similar PCs estimation accuracy. Recently, a randomized online PCA algorithm was proposed by [38], whose objective is to minimize the total expected quadratic error minus the total error of the batch algorithm (*i.e.*, the regret). However, all of these online PCA algorithms are not robust to the outliers.

To overcome the sensitiveness of PCA to outliers, many robust PCA algorithms have been proposed [36, 20, 52], which can be roughly categorized into two groups. They either pursue robust estimation of the covariance matrix, *e.g.*, M-estimator [32], S-estimator [37], and Minimum Covariance Determinant (MCD) estimator [36], or directly maximize certain robust estimation of univariate variance for the projected observations [30, 19, 20, 29]. These algorithms inherit the robustness characteristics of the adopted estimators and are qualitatively robust. However, none of them can be directly applied in online learning setting. Recently, [61] and

the following work [47] propose high-dimensional robust PCA, which can achieve maximum 50% breakdown point. However, these methods iteratively remove the observations or tunes the observations weights based on statistics obtained from the whole data set. Thus, when a new data point is revealed, these methods need to re-explore all of the data and become quite computationally intensive.

The most related works to ours are the following three works. In [53], an incremental and robust subspace learning method is proposed. The method proposes to integrate the M-estimation into the standard incremental PCA calculation. Specifically, each newly coming data point is re-weighted by a pre-defined influence function [51] of its residual to the current estimated subspace. However, no performance guarantee is provided in this work. Moreover, the performance of the proposed algorithm relies on the accuracy of PCs obtained previously. And the error will be cumulated inevitably. Recently, a compressive sensing based recursive robust PCA algorithm was proposed in [58]. In this work, the authors focused on the case where the *outliers* can be modeled as *sparse vectors*. In contrast, we do not impose any structural assumption on the outliers. Moreover, the proposed method in [58] essentially solves compressive sensing optimization over a small batch of data to update the PCs estimation instead of using a single sample, and it is not clear how to extend the method to the latter case. Recently, He *et al.* propose an incremental gradient descent method on Grassmannian manifold for solving the robust PCA problem, named GRASTA [50]. However, they also focus on a different case from ours where the outliers are sparse vectors.

3.3 The Algorithm

3.3.1 Problem Setup

Given a set of observations $\{\mathbf{y}_1, \cdots, \mathbf{y}_T\}$ (here T can be finite or infinite) which are revealed sequentially, the goal of online PCA is to estimate and update the principal components (PCs) based on the newly revealed sample $\mathbf{y}_t$ at time instance t. Here, the observations are the mixture of authentic samples (non-corrupted samples) and

outliers (corrupted samples). The authentic samples $\mathbf{z}_i \in \mathbb{R}^p$ are generated through a linear mapping: $\mathbf{z}_i = A\mathbf{x}_i + \mathbf{n}_i$. Here, noise $\mathbf{n}_i$ is sampled from normal distribution $\mathcal{N}(\mathbf{0}, I_p)$; and the signal $\mathbf{x}_i \in \mathbb{R}^d$ are i.i.d. samples of a random variable $\mathbf{x}$ with mean zero and variance I_d. Let μ denote the distribution of $\mathbf{x}$. The matrix $A \in \mathbb{R}^{p\times d}$ and the distribution μ are unknown. We assume μ is absolutely continuous w.r.t. the Borel measure and spherically symmetric. And μ has light tails, *i.e.*, there exist constants $C > 0$ such that $\Pr(\|\mathbf{x}\| \geq x) \leq d\exp(1 - Cx/\alpha\sqrt{d})$ for all $x \geq 0$. The outliers are denoted as $\mathbf{o}_i \in \mathbb{R}^p$ and in particular they are defined as follows.

Definition 1 (Outlier)**.** *A sample $\mathbf{o}_i \in \mathbb{R}^p$ is an outlier w.r.t. the subspace spanned by $\{\overline{\mathbf{w}}_j\}_{j=1}^d$ if it deviates from the subspace,* i.e., $\sum_{j=1}^d |\overline{\mathbf{w}}_j^T \mathbf{o}_i|^2 \leq \Gamma_o$.

In the above definition, we assume that the basis $\overline{\mathbf{w}}_j$ and outliers $\mathbf{o}$ are both *normalized* (see Algorithm 2 step 1)-a) where all the samples are ℓ_2-normalized). Thus, we directly use inner product to define Γ_0. From the definition, a sample is called outlier if it is distant from the underlying subspace of the signal. Note that the outliers can follow arbitrary distribution. In this work, we are interested in the case where the outliers are mixed with authentic samples uniformly in the data stream, *i.e.*, taken any subset of the dataset, the outlier fraction is identical when the size of the subset is large enough.

The input to the proposed online RPCA algorithm is the sequence of observations $\mathcal{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_T\}$, which is union of authentic samples $\mathcal{Z} = \{\mathbf{z}_i\}$ generated by the aforementioned linear model and outliers $\mathcal{O} = \{\mathbf{o}_i\}$. And the outlier fraction in the observations is denoted as λ. Online RPCA aims at learning the PCs robustly and the learning process proceeds in time instances. At the time instance t, online RPCA chooses a set of principal components $\{\mathbf{w}_j^{(t)}\}_{j=1}^d$. And the performance of the estimation is measured by the Expressed Variance (E.V.) [61]:

$$\text{E.V.} \triangleq \frac{\sum_{j=1}^d {\mathbf{w}_j^{(t)}}^T AA^T \mathbf{w}_j^{(t)}}{\sum_{j=1}^d \overline{\mathbf{w}}_j^T AA^T \overline{\mathbf{w}}_j}.$$

Here, $\{\overline{\mathbf{w}}_j\}_{j=1}^d$ denote the true principal components of matrix A. The E.V. repre-

sents the portion of signal $A\mathbf{x}$ being expressed by $\{\mathbf{w}_j^{(t)}\}_{j=1}^d$. Thus, $1 - \text{E.V.}$ is the reconstruction error of the signal. The E.V. is a commonly used evaluation metric for the PCA algorithms [61, 21]. It is always less than one, with equality achieved by a perfect recovery.

3.3.2 Online Robust PCA Algorithm

The details of the proposed online RPCA algorithm are shown in Algorithm 2. In the algorithm, the observation sequence $\mathcal{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_T\}$ is sequentially partitioned into $(T' + 1)$ batches $\{B_0, B_1, B_2, \ldots, B_{T'}\}$. And each batch consists of b observations. Since the authentic samples and outliers are mixed uniformly, the outlier fraction in each batch is also λ. Namely, in each batch B_i, there are $(1 - \lambda)b$ authentic samples and λb outliers.

Note that such small batch partition is only for the ease of illustration and analysis. Since the algorithm only involves standard PCA computation, we can employ any incremental or online PCA method [24, 53] to update the PCs estimation upon accepting a new sample. And the maintained sample covariance matrix, can be set to zero every b time instances. Thus the batch partition is by no means necessary in practical implementation. In the algorithm, the initial PC estimation can be obtained through standard PCA or robust PCA [61] on a mini batch of the samples.

We now explain the intuition of the proposed online RPCA algorithm. Given an initial solution $\mathbf{w}^{(0)}$ which is "closer" to the true PC directions than to the outlier direction [1], the authentic samples will have larger variance along the current PC direction than outliers. Thus in the probabilistic data selection process (as shown in Algorithm 2 step b) to step d)), more authentic samples will be accepted than outliers. Therefore, in the following PC updating based on standard PCA on the accepted data, authentic samples will contribute more than the outliers. And the estimated PCs will be "moved" towards to the true PCs gradually. Such process is repeated until convergence.

[1] In the following section, we will provide a precise description of the required closeness.

Algorithm 2 Online Robust PCA Algorithm

Input: Data sequence $\{\mathbf{y}_1, \ldots, \mathbf{y}_T\}$, buffer size b.
Initialization: Partition the data sequence into small batches $\{B_0, B_1, \ldots, B_{T'}\}$. Each patch contains b data points. Perform PCA on the first batch B_0 and obtain the initial principal component $\{\mathbf{w}_j^{(0)}\}_{j=1}^d$.
$t = 1$. $\mathbf{w}_j^* = \mathbf{w}_j^{(0)}, \forall j = 1, \ldots, d$.
while $t \leq T'$ **do**
1) Initialize the sample covariance matrix: $C^{(t)} = 0$.
for $i = 1$ to b **do**
a) Normalize the data point by its ℓ_2-norm: $\mathbf{y}_i^{(t)} := \mathbf{y}_i^{(t)} / \|\mathbf{y}_i^{(t)}\|_{\ell_2}$.
b) Calculate the variance of $\mathbf{y}_i^{(t)}$ along the direction $\mathbf{w}^{(t-1)}$: $\delta_i = \sum_{j=1}^d \left| \mathbf{w}_j^{(t-1)^T} \mathbf{y}_i^{(t)} \right|^2$.
c) Accept $\mathbf{y}_i^{(t)}$ with probability δ_i.
d) Scale $\mathbf{y}_i^{(t)}$ as $\mathbf{y}_i^{(t)} \leftarrow \mathbf{y}_i^{(t)} / b\sqrt{\delta_i}$.
e) If $\mathbf{y}_i^{(t)}$ is accepted, update

$$C^{(t)} = C^{(t)} + \mathbf{y}_i^{(t)} \mathbf{y}_i^{(t)^T}.$$

end for
2) Perform eigen-decomposition on C_t and obtain the leading d eigenvector $\{\mathbf{w}_j^{(t)}\}_{j=1}^d$.
3) Update the PC as $\mathbf{w}_j^* = \mathbf{w}_j^{(t)}, \forall j = 1, \ldots, d$.
4) $t := t + 1$.
end while
Return $\mathbf{w}^*$.

3.4 Main Results

In this section we present the theoretical performance guarantee of the proposed online RPCA algorithm (Algorithm 2). In the sequel, $\mathbf{w}_j^{(t)}$ is the solution at the t-th time instance. Here w.l.o.g. we assume the the matrix A is normalized such that the E.V. of the true principal component $\overline{\mathbf{w}}_j$ is $\sum_{j=1}^d \overline{\mathbf{w}}_j^T A^T A \overline{\mathbf{w}}_j = 1$. The following theorem provides the performance guarantee of Algorithm 2 under the noisy case. And the performance of $\mathbf{w}^{(t)}$ can be measured by $H(\mathbf{w}^{(t)}) \triangleq \sum_{j=1}^d \|\mathbf{w}^{(t)}{}_j^T A\|^2$. Let $s = \|\mathbf{x}\|_2 / \|\mathbf{n}\|_2$ be the *signal noise ratio*.

Theorem 8 (Noisy Case Performance)**.** *There exist universal constants c_1', c_2' which depend on the signal noise ratio s and $\epsilon_1, \epsilon_2 > 0$ which approximate zero when $s \to \infty$*

or $b \to \infty$, such that if the initial solution $\mathbf{w}_j^{(0)}$ in Algorithm 2 satisfies:

$$\sum_{i=1}^{\lambda b}\sum_{j=1}^{d}\left|\mathbf{w}_j^{(0)^T}\mathbf{o}_i\right|^2 \leq \frac{(1-\lambda)b(1-\epsilon^2)}{c_2'(1-\Gamma_o)}\left(\frac{1}{4}(c_1'(1-\epsilon)-\epsilon_1)^2-\epsilon_2\right),$$

and

$$H(\mathbf{w}^{(0)}) \geq \frac{1}{2}(c_1'(1-2\epsilon)-\epsilon_1)-\sqrt{\frac{(c_1'(1-\epsilon)+\epsilon_1)^2-4\epsilon_2}{4}-\frac{c_2'\sum_{i=1}^{\lambda b}\sum_{j=1}^{d}(\mathbf{w}_j^{(0)^T}\mathbf{o}_i)^2(1-\Gamma_o)}{(1-\lambda)b(1-\epsilon^2)}},$$

then the performance of the solution from Algorithm 2 will be improved in each iteration, and eventually converges to:

$$\lim_{t\to\infty} H(\mathbf{w}^{(t)}) \geq \frac{1}{2}(c_1'(1-2\epsilon)-\epsilon_1)+\sqrt{\frac{(c_1'(1-2\epsilon)-\epsilon_1)^2-4\epsilon_2}{4}-\frac{c_2'\sum_{i=1}^{\lambda b}\sum_{j=1}^{d}(\mathbf{w}_j^{(0)^T}\mathbf{o}_i)^2(1-\Gamma_o)}{(1-\lambda)b(1-\epsilon^2)}}.$$

Here ϵ_1 and ϵ_2 decay as $\tilde{O}(d^{\frac{1}{2}}b^{-\frac{1}{2}}s^{-1})$, and ϵ decays as $\tilde{O}(d^{\frac{1}{2}}b^{-\frac{1}{2}})$. And $c_1' = (s-1)^2/(s+1)^2, c_2' = (1+1/s)^4$.

Remark 1. From Theorem 8, we can observe followings:

1. When the outliers vanish, the second term in the square root of performance $H(\mathbf{w}^{(t)})$ is zero. $H(\mathbf{w}^{(t)})$ will converge to $(c_1'(1-2\epsilon)-\epsilon_1)/2+\sqrt{(c_1'(1-2\epsilon)-\epsilon_1)^2-4\epsilon_2}/2 < c_1'(1-2\epsilon)-\epsilon_1 < c_1' < 1$. Namely, the final performance is smaller than but approximates 1. Here $c_1', \epsilon_1, \epsilon_2$ explain the affect of noise.

2. When $s \to \infty$, the affect of noise is eliminated, $\epsilon_1, \epsilon_2 \to 0, c_1' \to 1$. $H(\mathbf{w}^{(t)})$ converges to $1-2\epsilon$. Here ϵ depends on the ratio of intrinsic dimension over the sample size. And ϵ accounts for the statistical bias due to performing PCA on a small portion of the data.

3. When the batch size increases to infinity, $\epsilon \to 0$, $H(\mathbf{w}^{(t)})$ converges to 1, meaning perfect recovery.

To further investigate the behavior of the proposed online RPCA in presence of outliers, we consider the following noiseless case. For the noiseless case, the signal

noise ratio $s \to \infty$, and thus $c_1', c_2' \to 1$ and $\epsilon_1, \epsilon_2 \to 0$. Then we can immediately obtain the performance bound of Algorithm 2 for the noiseless case from Theorem 8.

Theorem 9 (Noiseless Case Performance)**.** *Suppose there is no noise. If the initial solution* $\mathbf{w}^{(0)}$ *in Algorithm 2 satisfies:*

$$\sum_{i=1}^{\lambda b}\sum_{j=1}^{d}({\mathbf{w}^{(0)}}_j^T\mathbf{o}_i)^2 \leq \frac{(1-\lambda)b}{4(1-\Gamma_o)},$$

and

$$H(\mathbf{w}^{(0)}) \geq \frac{1}{2} - \sqrt{\frac{1}{4} - \frac{\sum_{i=1}^{\lambda b}\sum_{j=1}^{d}({\mathbf{w}_j^{(0)}}^T\mathbf{o}_i)^2(1-\Gamma_o)}{(1-\lambda)b}},$$

the performance of the solution from Algorithm 2 will be improved in each updating and eventually converges to:

$$\lim_{t\to\infty} H(\mathbf{w}^{(t)}) \geq \frac{1}{2} + \sqrt{\frac{1}{4} - \frac{\sum_{i=1}^{\lambda b}\sum_{j=1}^{d}({\mathbf{w}_j^{(0)}}^T\mathbf{o}_i)^2(1-\Gamma_o)}{(1-\lambda)b}}.$$

Remark 2. Observe from Theorem 9 the followings:

1. When the true direction and outlier direction are identical, *i.e.,* $\sum_{j=1}^{d}|\overline{\mathbf{w}}_j^T\mathbf{o}_i|^2 = 1$, the conditions become $\sum_{i=1}^{\lambda b}\sum_{j=1}^{d}({\mathbf{w}^{(0)}}^T\mathbf{o}_i)^2 < \infty$ and $H(\mathbf{w}^{(0)}) \geq 0$. Namely, for whatever initial solution, the final performance will converge to 1.

2. When the true direction and outlier direction are orthogonal, *i.e.,* $\sum_{j=1}^{d}|\overline{\mathbf{w}}_j^T\mathbf{o}_i|^2 = 0$, the conditions for the initial solution becomes $\sum_{i=1}^{\lambda b}\sum_{j=1}^{d}|{\mathbf{w}^{(0)}}_j^T\mathbf{o}_i|^2 \leq \frac{b(1-\lambda)}{4}$, and $H_0 \geq \frac{1}{2} - \sqrt{\frac{1}{4} - \frac{\sum_{i=1}^{\lambda b}\sum_{j=1}^{d}({\mathbf{w}_j^{(0)}}^T\mathbf{o}_i)^2}{(1-\lambda)b}}$. Hence, when the outlier fraction λ increases, the initial solution should be further away from outlier direction.

3. When $0 < \sum_{j=1}^{d}|\overline{\mathbf{w}}_j^T\mathbf{o}_i|^2 < 1$, the performance of online RPCA is improved by at least $2\sqrt{\frac{1}{4} - \frac{\sum_{i=1}^{\lambda b}\sum_{j=1}^{d}({\mathbf{w}_j^{(0)}}^T\mathbf{o}_i)^2(1-\Gamma_o)}{(1-\lambda)b}}$ from its initial solution. Hence, when the initial solution is further away from the outlier direction, the outlier fraction is smaller, or the outlier direction is closer to true direction, the improvement is more significant. Moreover, observe that given a proper initial

solution, even if $\lambda = 0.5$, the performance of online RPCA still has a positive lower bound. Therefore, the breakdown point of online RPCA is 50%, the highest that any algorithm can achieve.

Discussion on the initial condition In Theorem 8 and Theorem 9, a mild condition is imposed on the initial solution. In practice, the initial estimate can be obtained by applying batch RPCA [47] or HRPCA [61] on a small subset of the data. These batch methods are able to provide initial estimate with performance guarantee, which may satisfy the initial condition.

3.5 Proof of The Results

In the proof of Theorem 8, we first show that when the PCs estimation is being improved, the variance of outliers along the PCs will keep decreasing. Then we demonstrate that each PCs updating conducted by Algorithm 2 produces a better PCs estimation and decreases the impact of outliers. Such improvement will continue until convergence, and the final performance has bounded deviation from the optimum.

We provide here some concentration lemmas which are used in the proof of Theorem 8. The proof of these lemmas is provided in the supplementary material. We first show that with high probability, both the largest and smallest eigenvalues of the signals $\mathbf{x}_i$ in the original space converge to 1. This result is adopted from [61].

Lemma 4. *There exists a constant c that only depends on μ and d, such that for all $\gamma > 0$ and b signals $\{\mathbf{x}_i\}_{i=1}^{b}$, the following holds with high probability:*

$$\sup_{\mathbf{w}\in\mathcal{S}_d} \left| \frac{1}{b}\sum_{i=1}^{b} (\mathbf{w}^T\mathbf{x}_i)^2 - 1 \right| \leq \epsilon,$$

where $\epsilon = c\alpha\sqrt{d\log^3 b/b}$.

Next lemma is about the sampling process in the Algorithm 1 from step b) to step d). Though the sampling process is without replacement and the sampled

observations are not i.i.d., the following lemma provides the concentration of the sampled observations.

Lemma 5 (Operator-Bernstein inequality [23]). *Let* $\{\mathbf{z}_i'\}_{i=1}^m$ *be a subset of* $\mathcal{Z} = \{\mathbf{z}_i\}_{i=1}^t$*, which is formed by randomly sampling* without *replacement from* $\mathcal{Z}$*, as in Algorithm 1. Then the following statement holds*

$$\left|\sum_{i=1}^m \mathbf{w}^T\mathbf{z}_i' - \mathbb{E}\left(\sum_{i=1}^m \mathbf{w}^T\mathbf{z}_i'\right)\right| \leq \delta$$

with probability larger than $1 - 2\exp(-\delta^2/4m)$.

We then show that the authentic samples concentrate along the true principal component direction $\overline{\mathbf{w}}$, as stated in the following lemma.

Lemma 6. *If there exists* ϵ *such that*

$$\sup_{\mathbf{w}\in\mathcal{S}_d}\left|\frac{1}{t}\sum_{i=1}^t |\mathbf{w}^T\mathbf{x}_i|^2 - 1\right| \leq \epsilon,$$

and the observations $\mathbf{z}_i$ *are normalized by* ℓ_2*-norm, then for any* $\mathbf{w}_1, \cdots, \mathbf{w}_d \in \mathcal{S}_p$*, the following holds:*

$$\begin{aligned} & \frac{(1-\epsilon)H(\mathbf{w}) - 2\sqrt{(1+\epsilon)H(\mathbf{w})}/s}{(1/s+1)^2} \\ \leq \quad & \frac{1}{t}\sum_{i=1}^t\sum_{j=1}^d (\mathbf{w}_j^T\mathbf{z}_i)^2 \leq \frac{(1+\epsilon)H(\mathbf{w}) + 2\sqrt{(1+\epsilon)H(\mathbf{w})}/s + 1/s^2}{(1/s-1)^2}, \end{aligned}$$

where $H(\mathbf{w}) = \sum_{j=1}^d \|\mathbf{w}_j^T A\|^2$ *and* s *is the signal noise ratio.*

Based on Lemma 11 and Lemma 6, we can provide the following concentration results for the selected observations in the Algorithm 2.

Lemma 7. *If there exists* ϵ *such that*

$$\sup_{\mathbf{w}\in\mathcal{S}_d}\left|\frac{1}{t}\sum_{i=1}^t |\mathbf{w}^T\mathbf{x}_i|^2 - 1\right| \leq \epsilon,$$

and the observations $\{\mathbf{z}'_i\}_{i=1}^m$ *are sampled from* $\{\mathbf{z}_i\}_{i=1}^d$ *as in Algorithm 1, then for any* $\mathbf{w}_1, \ldots, \mathbf{w}_d \in \mathcal{S}_p$*, with large probability, the following holds:*

$$\begin{aligned} & \frac{(1-\epsilon)H(\mathbf{w}) - 2\sqrt{(1+\epsilon)H(\mathbf{w})}/s}{(1/s+1)^2 b/m} - \delta \\ \leq \quad & \frac{1}{t}\sum_{i=1}^{t}\sum_{j=1}^{d}(\mathbf{w}_j^T \mathbf{z}'_i)^2 \leq \frac{(1+\epsilon)H(\mathbf{w}) + 2\sqrt{(1+\epsilon)H(\mathbf{w})}/s + 1/s^2}{(1/s-1)^2 b/m} + \delta, \end{aligned}$$

where $H(\mathbf{w}) \triangleq \sum_{j=1}^d \|\mathbf{w}_j^T A\|^2$*, s is the signal noise ratio and m is the number of sampled observations in each batch and $\delta > 0$ is a small constant.*

We denote the set of accepted authentic samples as $\mathcal{Z}_t$ and the set of accepted outliers as $\mathcal{O}_t$ from the t-th small batch. In the following lemma, we provide the estimation of number of accepted authentic samples $|\mathcal{Z}_t|$ and outliers $|\mathcal{O}_t|$.

Lemma 8. *For the current obtained principal components* $\{\mathbf{w}_j^{(t-1)}\}_{j=1}^d$*, the number of the accepted authentic samples* $|\mathcal{Z}_t|$ *and outliers* $|\mathcal{O}_t|$ *satisfy*

$$\left| \frac{|\mathcal{Z}_t|}{b} - \frac{1}{b}\sum_{i=1}^{(1-\lambda)b}\sum_{j=1}^{d}(\mathbf{w}_j^{(t-1)^T}\mathbf{z}_i)^2 \right| \leq \delta \text{ and } \left| \frac{|\mathcal{O}_t|}{b} - \frac{1}{b}\sum_{i=1}^{\lambda b}\sum_{j=1}^{d}(\mathbf{w}_j^{(t-1)^T}\mathbf{o}_i)^2 \right| \leq \delta$$

with probability at least $1 - e^{-2\delta^2 b}$*. Here $\delta > 0$ is a small constant, λ is the outlier fraction and b is the size of the small batch.*

From the above lemma, we can see that when the batch size b is sufficiently large, the above estimation for $|\mathcal{Z}_t|$ and $|\mathcal{O}_t|$ holds with large probability.

In the following lemma, we show that when the algorithm improves the PCs estimation, the impact of outliers will be decreased accordingly.

Lemma 9. *For an outlier* $\mathbf{o}_i$*, an arbitrary orthogonal basis* $\{\mathbf{w}_j\}_{j=1}^d$ *and the groundtruth basis* $\{\overline{\mathbf{w}}_j\}_{j=1}^d$ *which satisfy that* $\sum_{j=1}^d \mathbf{w}_j^T\mathbf{o}_i \geq \sum_{j=1}^d \overline{\mathbf{w}}_j^T\mathbf{o}_i$ *and* $\sum_{j=1}^d \overline{\mathbf{w}}_j^T\mathbf{w}_j \geq \sum_{j=1}^d \overline{\mathbf{w}}_j^T\mathbf{o}_i$*, the value of* $\sum_{j=1}^d \mathbf{w}_j^T\mathbf{o}_i$ *is a monotonically decreasing function of* $\sum_{j=1}^d \overline{\mathbf{w}}_j^T\mathbf{w}_j$*.*

Being equipped by the above lemmas, we can proceed to prove Theorem 8. The details of the proof is deferred to the supplementary material due to the space limit.

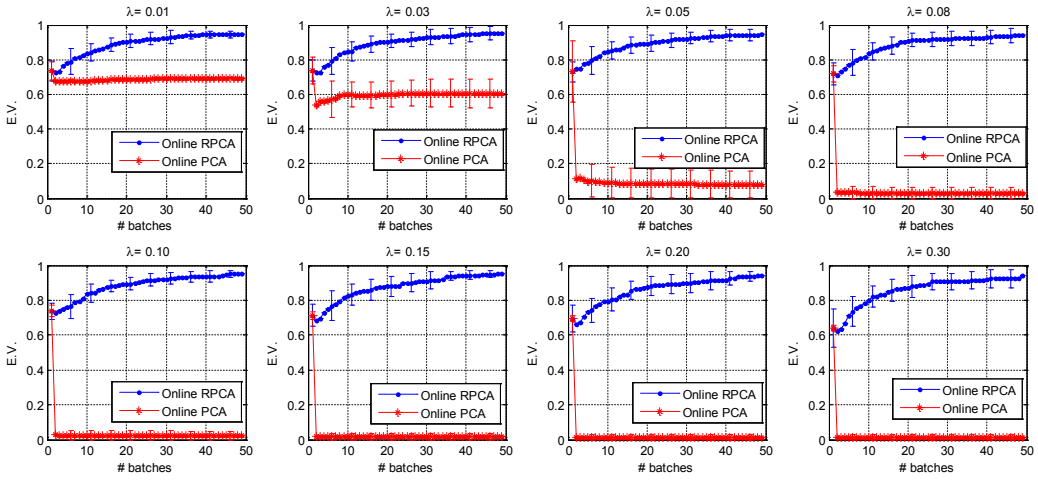


Figure 3.1: Performance comparison of online RPCA (blue line) with online PCA (red line). Here $s = 2, p = 100, T = 10,000, d = 1$.

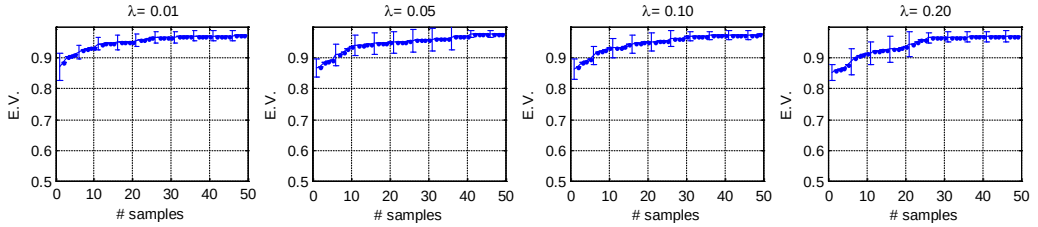


Figure 3.2: Performance of online RPCA. Here $s = 3, p = 100, T = 10,000, d = 1$.

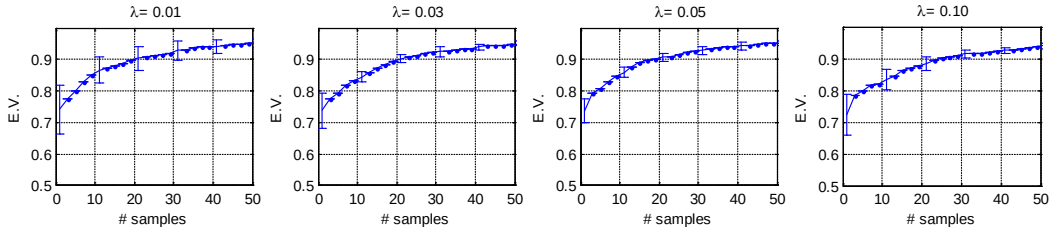


Figure 3.3: Performance of online RPCA. The outliers distribute along 5 different directions. Here $s = 2, p = 100, T = 10,000, d = 1$.

3.6 Simulations

The numerical study is aimed to illustrate the performance of online robust PCA algorithm. We follow the data generation method in [61] to randomly generate a $p \times d$ matrix A and then scale its leading singular value to s, which is the signal noise ratio. A λ fraction of outliers are generated on a line with randomly selected direction. Since it is hard to determine the most adversarial outlier distribution, in

simulations, we generate the outliers concentrate on several directions deviating from the groundtruth subspace. This makes a rather adversarial case and is suitable for investigating the robustness of the proposed RPCA algorithm. In the simulations, in total $T = 10,000$ samples are generated to form the sample sequence. For each parameter setup, we report the average result of 20 tests and standard deviation. The initial solution is obtained by performing standard PCA on the first batch after removing the outliers. Simulation results for $p = 100, d = 1, s = 2, 3$ are shown in Figure 3.1 and Figure 3.2 respectively. More simulation results for the $d > 1$ case are provided in the supplementary material due to the space limit.

From the simulation results, we can make following observations. Firstly, online RPCA can improve the PC estimation steadily. With more samples being revealed, the E.V. of the online RPCA outputs keep increasing. Secondly, the performance of online RPCA is rather robust to outliers. For example, the final result converges to E.V. ≈ 0.95 even with $\lambda = 0.3$ for relatively low signal noise ratio $s = 2$ as shown in Figure 3.1. We further compare the results shown in Figure 3.1 and Figure 3.2 and can observe that when the signal noise ratio s increases, the final performance becomes better. Note that here the initial solution is obtained by simply performing standard PCA on the first 40 clean samples to mimic a relatively good initial solution.

To more clearly demonstrate the robustness of online RPCA to outliers, we implement the online PCA proposed in [38] as baseline for the $\sigma = 2$ case. The results are presented in Figure 3.1, from which we can observe that the performance of online PCA drops due to the sensitiveness to newly coming outliers. When the outlier fraction $\lambda \geq 0.1$, the online PCA cannot recover the true PC directions and the performance is as low as 0.

We also simulate the case where the outliers are distributed on multiple lines. In particular, we investigate the case for outliers distributing on 5 different lines. And the simulation results are presented in Figure 3.3. The results are quite similar to the one with outliers distributing on a single line. We can see that online RPCA also performs quite well for this case.

3.7 Technical Lemmas

Before proving the theoretical results in this chapter, we first present following lemmas used in the proof.

Lemma 10. *There exists a constant c that only depends on μ and d, such that for all $\gamma > 0$ and b signals $\{\mathbf{x}_i\}_{i=1}^b$, the following holds with high probability:*

$$\sup_{\mathbf{w}\in\mathcal{S}_d} \left| \frac{1}{b}\sum_{i=1}^{b} (\mathbf{w}^T\mathbf{x}_i)^2 - 1 \right| \leq \epsilon,$$

where $\epsilon = c\alpha\sqrt{d\log^3 b/b}$.

Lemma 11 (Operator-Bernstein inequality)**.** *Let $\{\mathbf{z}_i'\}_{i=1}^m$ be a subset of $\mathcal{Z} = \{\mathbf{z}_i\}_{i=1}^t$, which is formed by randomly sampling* without *replacement from $\mathcal{Z}$, as in Algorithm 1. Then the following statement holds*

$$\left| \sum_{i=1}^{m} \mathbf{w}^T\mathbf{z}_i' - \mathbb{E}\left(\sum_{i=1}^{m} \mathbf{w}^T\mathbf{z}_i' \right) \right| \leq \delta$$

with probability larger than $1 - 2\exp(-\delta^2/4m)$.

3.8 Proof of Lemma 6

Proof. Suppose the noise magnitude is $\|\mathbf{n}_i\|_2 = 1$ with out loss of generality. And thus the signal magnitude is $\|\mathbf{x}_i\|_2 = s$. Then we have:

$$
\begin{aligned}
& \frac{1}{t}\sum_{i=1}^{t}\sum_{j=1}^{d}\frac{|\mathbf{w}_j^T\mathbf{z}_i|^2}{\|\mathbf{z}_i\|_2^2} \\
= \; & \frac{1}{t}\sum_{i=1}^{t}\sum_{j=1}^{d}\frac{|\mathbf{w}_j^TA\mathbf{x}_i+\mathbf{w}_j^T\mathbf{n}_i|^2}{\|A\mathbf{x}_i+\mathbf{n}_i\|_2^2} \\
\overset{(a)}{\leq} \; & \frac{1}{t}\sum_{i=1}^{t}\sum_{j=1}^{d}\frac{|\mathbf{w}_j^TA\mathbf{x}_i+\mathbf{w}_j^T\mathbf{n}_i|^2}{(\|A\mathbf{x}_i\|_2-\|\mathbf{n}_i\|_2)^2} \\
= \; & \frac{1}{t}\sum_{i=1}^{t}\sum_{j=1}^{d}\frac{|\mathbf{w}_j^TA\mathbf{x}_i+\mathbf{w}_j^T\mathbf{n}_i|^2}{(s-1)^2} \\
= \; & \frac{1}{(s-1)^2}\frac{1}{t}\left\{\sum_{i=1}^{t}\sum_{j=1}^{d}(\mathbf{w}_j^TA\mathbf{x}_i)^2+2\sum_{i=1}^{t}\sum_{j=1}^{d}(\mathbf{w}_j^TA\mathbf{x}_i)(\mathbf{w}_j^T\mathbf{n}_i)+\sum_{i=1}^{t}\sum_{j=1}^{d}(\mathbf{w}_j^T\mathbf{n}_i)^2\right\} \\
\overset{(b)}{\leq} \; & \frac{1}{(s-1)^2}\sum_{j=1}^{d}\left\{\|\mathbf{w}_j^TA\|_2^2\sup_{\mathbf{v}\in\mathcal{S}_d}\frac{1}{t}\sum_{i=1}^{t}|\mathbf{v}^T\mathbf{x}_i|^2+2\sqrt{\frac{1}{t}\sum_{i=1}^{t}(\mathbf{w}_j^TA\mathbf{x}_i)^2}\sqrt{\frac{1}{t}\sum_{i=1}^{t}(\mathbf{w}_j^T\mathbf{n}_i)^2}+\frac{1}{t}\sum_{i=1}^{t}(\mathbf{w}_j^T\mathbf{n}_i)^2\right\} \\
\overset{(c)}{\leq} \; & \frac{(1+\epsilon)\|\mathbf{w}^TA\|^2s^2+2\|\mathbf{w}^TA\|s\sqrt{1+\epsilon}+1}{(\mathrm{s}-1)^2}.
\end{aligned}
$$

Here the inequality (a) is from the triangle inequality. The inequality (b) is from that

$$
\begin{aligned}
& \sum_{i=1}^{t}|\mathbf{w}_j^TA\mathbf{x}_i|^2 = \mathbf{w}_j^TA\left(\sum_{i=1}^{t}\mathbf{x}_i\mathbf{x}_i^T\right)A^T\mathbf{w}_j \\
= \; & \|\mathbf{w}_j^TA\|_2^2\left\{\frac{\mathbf{w}_j^TA}{\|\mathbf{w}_j^TA\|_2}\left(\sum_{i=1}^{t}\mathbf{x}_i\mathbf{x}_i^T\right)\frac{A^T\mathbf{w}_j}{\|\mathbf{w}_j^TA\|_2}\right\} \\
\leq \; & \|\mathbf{w}_j^TA\|_2^2\sup_{\mathbf{v}\in\mathcal{S}_d}\frac{1}{t}\sum_{i=1}^{t}|\mathbf{v}^T\mathbf{x}_i|^2,
\end{aligned}
$$

for the first term and the inequality $(\sum_i a_ib_i)^2 \leq (\sum_i a_i^2)(\sum_i b_i^2)$ for the second term. And the inequality (c) is from the definition of $H(\mathbf{w})$ and applying Lemma 10.

Similarly, we have

$$\begin{aligned}
& \frac{1}{t}\sum_{i=1}^{t}\sum_{j=1}^{d}\frac{|\mathbf{w}_j^T\mathbf{z}_i|^2}{\|\mathbf{z}_i\|_2^2} \\
= & \frac{1}{t}\sum_{i=1}^{t}\sum_{j=1}^{d}\frac{|\mathbf{w}_j^TA\mathbf{x}_i+\mathbf{w}_j^T\mathbf{n}_i|^2}{\|A\mathbf{x}_i+\mathbf{n}_i\|_2^2} \\
\geq & \frac{1}{t}\sum_{i=1}^{t}\sum_{j=1}^{d}\frac{|\mathbf{w}_j^TA\mathbf{x}_i|^2-2|\mathbf{w}_j^TA\mathbf{x}_i||\mathbf{w}_j^T\mathbf{n}_i|}{\|A\mathbf{x}_i+\mathbf{n}_i\|_2^2} \\
\geq & \frac{1}{t}\sum_{i=1}^{t}\sum_{j=1}^{d}\frac{|\mathbf{w}_j^TA\mathbf{x}_i|^2-2|\mathbf{w}_j^TA\mathbf{x}_i||\mathbf{w}_j^T\mathbf{n}_i|}{(\|A\mathbf{x}_i\|_2+\|\mathbf{n}_i\|_2)^2} \\
= & \frac{1}{t}\sum_{i=1}^{t}\sum_{j=1}^{d}\frac{(\mathbf{w}_j^TA\mathbf{x}_i)^2-2|\mathbf{w}_j^TA\mathbf{x}_i||\mathbf{w}_j^T\mathbf{n}_i|}{(s+1)^2} \\
\geq & \frac{(1-\epsilon)H(\mathbf{w})s^2-2s\sqrt{(1+\epsilon)H(\mathbf{w})}}{(s+1)^2}.
\end{aligned}$$

Combining the above two results, we complete the proof. □

3.9 Proof of Lemma 7

Proof. According to Lemma 2, we have

$$\begin{aligned}
& \frac{(1-\epsilon)H(\mathbf{w})-2\sqrt{(1+\epsilon)H(\mathbf{w})}/s}{(1/s+1)^2} \\
\leq & \frac{1}{t}\sum_{i=1}^{t}\sum_{j=1}^{d}(\mathbf{w}_j^T\mathbf{z}_i)^2 \leq \frac{(1+\epsilon)H(\mathbf{w})+2\sqrt{(1+\epsilon)H(\mathbf{w})}/s+1/s^2}{(1/s-1)^2}.
\end{aligned}$$

Next we will show that the method of sampling without replacement given in Algorithm 1 provides an unbiased estimation of $\frac{1}{t}\sum_{i=1}^{t}\sum_{j=1}^{d}(\mathbf{w}_j^T\mathbf{z}_i)^2$. To see this, we define the random variables $X_i = |\mathbf{w}^T\mathbf{z}_i|^2$ and $Y_i = X_i/bX_i$ which is sampled from X_i with probability $p_i = X_i$ and re-scaled by bX_i as in Algorithm 1. Then

$$\mathbb{E}[Y_i] = \sum_{i=1}^{t} p_iY_i = \sum_{i=1}^{t} X_i\frac{X_i}{bX_i} = \sum_{i=1}^{t}\frac{X_i}{b}.$$

Thus,

$$\mathbb{E}\left[\sum_{i=1}^{m} Y_i\right] = \sum_{i=1}^{m} \mathbb{E}[Y_i] = \sum_{i=1}^{m}\sum_{i=1}^{t} \frac{X_i}{b} = \frac{m}{b}\sum_{i=1}^{t} X_i.$$

Namely,

$$\mathbb{E}\left[\sum_{i=1}^{m} \left|\mathbf{w}^T \mathbf{z}_i'\right|^2\right] = \frac{m}{b}\sum_{i=1}^{t} \left|\mathbf{w}^T \mathbf{z}_i\right|^2.$$

Thus, according to Lemma 3, we have

$$\left|\sum_{i=1}^{m} \left|\mathbf{w}^T \mathbf{z}_i'\right|^2 - \frac{m}{b}\sum_{i=1}^{t} \left|\mathbf{w}^T \mathbf{z}_i\right|^2\right| = \left|\sum_{i=1}^{m} \left|\mathbf{w}^T \mathbf{z}_i'\right|^2 - \mathbb{E}\left[\sum_{i=1}^{m} \left|\mathbf{w}^T \mathbf{z}_i'\right|^2\right]\right| < \delta,$$

with probability larger than $1 - 2\exp(-\delta^2/4m)$. Therefore,

$$\frac{m}{b}\sum_{i=1}^{t} \left|\mathbf{w}^T \mathbf{z}_i\right|^2 - \delta \leq \sum_{i=1}^{m} \left|\mathbf{w}^T \mathbf{z}_i'\right|^2 \leq \frac{m}{b}\sum_{i=1}^{t} \left|\mathbf{w}^T \mathbf{z}_i\right|^2 + \delta.$$

Then applying Lemma 11 completes the proof. □

3.10 Proof of Lemma 8

Proof. According to the Algorithm 1, the probability of accepting an authentic sample is

$$\Pr\left(\mathbf{z}_i \text{ is accepted}\right) = \sum_{j=1}^{d} \left(\mathbf{w}_j^{(t-1)^T} \mathbf{z}_i\right)^2.$$

Since there are in total $(1-\lambda)b$ authentic samples, we have

$$\mathbb{E}|\mathcal{Z}_t| = \sum_{i=1}^{(1-\lambda)b}\sum_{j=1}^{d} \left(\mathbf{w}_j^{(t-1)^T} \mathbf{z}_i\right)^2.$$

By applying the Chernoff bound, we have

$$\Pr\left\{||\mathcal{Z}_t| - \mathbb{E}|\mathcal{Z}_t|| < \delta b\right\} \geq 1 - e^{-2\delta^2 b}.$$

Thus,

$$\Pr\left\{\left|\frac{|\mathcal{Z}_t|}{b}-\frac{1}{b}\sum_{i=1}^{(1-\lambda)b}\sum_{j=1}^{d}\left(\mathbf{w}_j^{(t-1)^T}\mathbf{z}_i\right)^2\right|\leq\delta\right\}\geq 1-e^{-2\delta^2 b}.$$

Similarly, the expectation of the number of accepted outliers is

$$\mathbb{E}|\mathcal{O}_t|=\sum_{i=1}^{\lambda b}\sum_{j=1}^{d}\left(\mathbf{w}_j^{(t-1)^T}\mathbf{o}_i\right)^2.$$

And applying Chernoff bound again, we obtain

$$\Pr\left\{\left|\frac{|\mathcal{O}_t|}{b}-\frac{1}{b}\sum_{i=1}^{\lambda b}\sum_{j=1}^{d}\left(\mathbf{w}_j^{(t-1)^T}\mathbf{o}_i\right)^2\right|<\delta\right\}\geq 1-e^{-2\delta^2 b}.$$

□

3.11 Proof of Lemma 9

Proof. For the basis $\{\overline{\mathbf{w}}_j\}_{j=1}^d$ spanning the groundtruth subspace, we can always rotate these basis and align them to the estimated basis $\{\mathbf{w}_j\}_{j=1}^d$ to make sure that $\mathbf{o}_i, \mathbf{w}_j$ and $\overline{\mathbf{w}}_j$ lie within the same plane. We also denote the aligned basis as $\{\overline{\mathbf{w}}_j\}_{j=1}^d$ without causing confusion. For the single basis pair, $\mathbf{w}_j$ and $\overline{\mathbf{w}}_j$, it can be verified that

$$\mathbf{w}_j^T\mathbf{o}_i=(\overline{\mathbf{w}}_j^T\mathbf{o}_i)(\overline{\mathbf{w}}_j^T\mathbf{w}_j)+\sqrt{1-(\overline{\mathbf{w}}_j^T\mathbf{o}_i)^2}\sqrt{1-(\overline{\mathbf{w}}_j^T\mathbf{w}_j)^2},$$

when the basis $\mathbf{w}_j$ satisfies the stated conditions. Thus we have

$$\sum_{j=1}^{d}\mathbf{w}_j^T\mathbf{o}_i=\sum_{j=1}^{d}(\overline{\mathbf{w}}_j^T\mathbf{o}_i)(\overline{\mathbf{w}}_j^T\mathbf{w}_j)+\sum_{j=1}^{d}\sqrt{1-(\overline{\mathbf{w}}_j^T\mathbf{o}_i)^2}\sqrt{1-(\overline{\mathbf{w}}_j^T\mathbf{w}_j)^2}.$$

It is easy to verify that when $\sum_{j=1}^d\overline{\mathbf{w}}_j^T\mathbf{w}_j\geq\sum_{j=1}^d\overline{\mathbf{w}}_j^T\mathbf{o}_i$, the function

$$f(\mathbf{w}_j^T\overline{\mathbf{w}}_j)=\sum_{j=1}^{d}(\overline{\mathbf{w}}_j^T\mathbf{o}_i)(\overline{\mathbf{w}}_j^T\mathbf{w}_j)+\sum_{j=1}^{d}\sqrt{1-(\overline{\mathbf{w}}_j^T\mathbf{o}_i)^2}\sqrt{1-(\overline{\mathbf{w}}_j^T\mathbf{w}_j)^2}$$

is a monotonically decreasing function w.r.t. $\sum_{j=1}^{d} \mathbf{w}_j^T \overline{\mathbf{w}}_j$ be seeing that the increase of any $|\mathbf{w}_j^T \mathbf{o}_i|$ will decrease value of the function. □

3.12 Proof of Theorem 10

Theorem 10 (Noisy Case Performance)**.** *There exist universal constants c_1', c_2' which depend on the signal noise ratio s and $\epsilon_1, \epsilon_2 > 0$ which approximate zero when $s \to \infty$ or $b \to \infty$, such that if the outliers satisfies that $\sum_{j=1}^{d} |\overline{\mathbf{w}}_j^T \mathbf{o}_i|^2 \leq \Gamma_o$, the initial solution $\{\mathbf{w}^{(0)}\}_{j=1}^{d}$ in Algorithm 1 satisfies:*

$$\sum_{i=1}^{\lambda b} \sum_{j=1}^{d} \left| {\mathbf{w}_j^{(0)}}^T \mathbf{o}_i \right|^2 \leq \frac{(1-\lambda) b (1-\epsilon^2)}{c_2'(1-\Gamma_o)} \left(\frac{1}{4}(c_1'(1-\epsilon)+\epsilon_1)^2 - \epsilon_2 \right),$$

and

$$H(\mathbf{w}^{(0)}) \geq \frac{1}{2}(c_1'(1-2\epsilon)+\epsilon_1) - \sqrt{\frac{(c_1'(1-\epsilon)+\epsilon_1)^2 - 4\epsilon_2}{4} - \frac{c_2' \sum_{i=1}^{\lambda b} \sum_{j=1}^{d} ({\mathbf{w}_j^{(0)}}^T \mathbf{o}_i)^2 (1-\Gamma_o)}{(1-\lambda) b (1-\epsilon^2)}},$$

then the performance of the solution from Algorithm 1 will be improved in each iteration, and eventually converges to:

$$\lim_{t \to \infty} H_t \geq \frac{1}{2}(c_1'(1-2\epsilon)+\epsilon_1) + \sqrt{\frac{(c_1'(1-2\epsilon)+\epsilon_1)^2 - 4\epsilon_2}{4} - \frac{c_2' \sum_{i=1}^{\lambda b} \sum_{j=1}^{d} ({\mathbf{w}_j^{(0)}}^T \mathbf{o}_i)^2 (1-\Gamma_o)}{(1-\lambda) b (1-\epsilon^2)}}.$$

Here ϵ_1 and ϵ_2 decay as $\tilde{O}(d^{\frac{1}{2}} b^{-\frac{1}{2}} s^{-1})$, and ϵ decays as $\tilde{O}(d^{\frac{1}{2}} b^{-\frac{1}{2}})$. And $c_1' = (s+1)^2/(s-1)^2, c_2' = (1+1/s)^4$.

Proof of Theorem 10. The sample covariance matrix at trial t is calculated as:

$$C_t = \sum_{\mathbf{z}_i \in \mathcal{Z}_t} \mathbf{z}_i \mathbf{z}_i^T + \sum_{\mathbf{o}_i \in \mathcal{O}_t} \mathbf{o}_i \mathbf{o}_i^T.$$

And we have,

$$\sum_{j=1}^{d} \overline{\mathbf{w}}_j^T C_t \overline{\mathbf{w}}_j = \sum_{j=1}^{d} \sum_{\mathbf{z}_i \in \mathcal{Z}_t} (\overline{\mathbf{w}}_j^T \mathbf{z}_i)^2 + \sum_{j=1}^{d} \sum_{\mathbf{o}_i \in \mathcal{O}_t} (\overline{\mathbf{w}}_j^T \mathbf{o}_i)^2.$$

Thus for the PCA solution $\{\mathbf{w}_j^{(t)}\}_{j=1}^d$ on the current accepted data set $\mathcal{Y}_t = \mathcal{Z}_t \cup \mathcal{O}_t$, we have

$$\sum_{j=1}^{d}\sum_{\mathbf{z}_i\in\mathcal{Z}_t}\left(\mathbf{w}_j^{(t)^T}\mathbf{z}_i\right)^2+\sum_{j=1}^{d}\sum_{\mathbf{o}_i\in\mathcal{O}_t}\left(\mathbf{w}_j^{(t)^T}\mathbf{o}_i\right)^2 \geq \sum_{j=1}^{d}\sum_{\mathbf{z}_i\in\mathcal{Z}_t}\left(\overline{\mathbf{w}}_j{}^T\mathbf{z}_i\right)^2+\sum_{j=1}^{d}\sum_{\mathbf{o}_i\in\mathcal{O}_t}\left(\overline{\mathbf{w}}_j{}^T\mathbf{o}_i\right)^2, \tag{3.1}$$

where the inequality is from the fact that $\{\mathbf{w}_j^{(t)}\}_{j=1}^d$ are the leading eigenvectors of the covariance matrix C_t.

Note that all the data points are normalized by their ℓ_2-norm, therefore

$$\sum_{j=1}^{d}\left(\mathbf{w}_j^{(t)^T}\mathbf{o}_i\right)^2 \leq 1.$$

Thus we have $\sum_{j=1}^d\sum_{\mathbf{o}_i\in\mathcal{O}_t}\mathbf{w}_j^{(t)^T}\mathbf{o}_i^T\mathbf{o}_i\mathbf{w}_j^{(t)} \leq |\mathcal{O}_t|$. Substituting it to (3.1), we can obtain

$$\sum_{j=1}^{d}\sum_{\mathbf{z}_i\in\mathcal{Z}_t}\left(\mathbf{w}_j^{(t)^T}\mathbf{z}_i\right)^2+|\mathcal{O}_t| \geq \sum_{j=1}^{d}\sum_{\mathbf{z}_i\in\mathcal{Z}_t}\left(\overline{\mathbf{w}}_j{}^T\mathbf{z}_i\right)^2+\sum_{j=1}^{d}\sum_{\mathbf{o}_i\in\mathcal{O}_t}\left(\overline{\mathbf{w}}_j{}^T\mathbf{o}_i\right)^2.$$

According to the definition of outliers, the outliers variance along the true PC directions is upper bounded, *i.e.*, $\sum_{j=1}^d\left(\overline{\mathbf{w}}_j{}^T\mathbf{o}_i\right)^2 \leq \Gamma_o$. Thus, we have

$$\frac{1}{|\mathcal{Z}_t|}\sum_{j=1}^{d}\sum_{\mathbf{z}_i\in\mathcal{Z}_t}\left(\mathbf{w}_j^{(t)^T}\mathbf{z}_i\right)^2 \geq \frac{1}{|\mathcal{Z}_t|}\sum_{j=1}^{d}\sum_{\mathbf{z}_i\in\mathcal{Z}_t}(\overline{\mathbf{w}}_j^T\mathbf{z}_i)^2-\frac{|\mathcal{O}_t|}{|\mathcal{Z}_t|}(1-\Gamma_o). \tag{3.2}$$

According to Lemma 7, we have followings hold with large probability $1-2\exp(-\delta^2/4m)$,

$$\frac{1}{|\mathcal{Z}_t|}\sum_{j=1}^{d}\sum_{\mathbf{z}_i\in\mathcal{Z}_t}\left(\mathbf{w}_j^{(t)^T}\mathbf{z}_i\right)^2 \leq \frac{(1+\epsilon)s^2H^{(t)}+2s\sqrt{(1+\epsilon)H^{(t)}}+1}{(s-1)^2b/m}+\delta, \tag{3.3}$$

and

$$\frac{1}{|\mathcal{Z}_t|}\sum_{j=1}^{d}\sum_{\mathbf{z}_i\in\mathcal{Z}_t}(\overline{\mathbf{w}}_j^T\mathbf{z}_i)^2 \geq \frac{(1-\epsilon)s^2-2s\sqrt{(1+\epsilon)}}{(s+1)^2b/m}-\delta. \tag{3.4}$$

Here $m=|\mathcal{Z}^{(t)}|, H^{(t)}=H\left(\mathbf{w}_1^{(t)},\ldots,\mathbf{w}_d^{(t)}\right)$ and we utilize the fact that $H(\overline{\mathbf{w}}_1,\ldots,\overline{\mathbf{w}}_d)=$

1. Substitute (3.3) and (3.4) to (3.2), we can obtain that

$$\frac{(1+\epsilon)H^{(t)}+2\sqrt{(1+\epsilon)H^{(t)}}/s+1/s^2}{(1/s+1)^2}+2\delta\frac{b}{|\mathcal{Z}^{(t)}|}\geq\frac{(1-\epsilon)-2\sqrt{(1+\epsilon)}/s}{(1/s-1)^2}-\frac{b|\mathcal{O}_t|}{|\mathcal{Z}_t|^2}(1-\Gamma_o).$$

According to Lemma 8, we have, with a large probability,

$$\frac{|\mathcal{O}_t|}{|\mathcal{Z}_t|}=\frac{\sum_{j=1}^{d}\sum_{i=1}^{\lambda b}\left(\mathbf{w}_j^{(t-1)^T}\mathbf{w}_o\right)^2}{\sum_{j=1}^{d}\sum_{i=1}^{(1-\lambda)b}\left(\mathbf{w}_j^{(t-1)^T}\mathbf{z}_i\right)^2}\leq\frac{(1/s+1)^2\sum_{j=1}^{d}\sum_{i=1}^{\lambda b}\left(\mathbf{w}_j^{(t-1)^T}\mathbf{w}_o\right)^2}{(1-\lambda)b\left((1-\epsilon)H^{(t)}-2\sqrt{(1+\epsilon)H^{(t)}}/s\right)},$$

and

$$\frac{|\mathcal{Z}_t|}{b}\geq(1-\lambda)\left((1-\epsilon)H^{(t)}-2\sqrt{(1+\epsilon)H^{(t)}}/s\right).$$

Here the inequality is from Lemma 6.

Thus,

$$H^{(t)}\geq c_1-\frac{c_2\sum_{j=1}^{d}\sum_{i=1}^{\lambda b}\left(\mathbf{w}_j^{(t-1)^T}\mathbf{w}_o\right)^2(1-\Gamma_o)}{(1-\lambda)b((1-\epsilon)H^{(t-1)}-\epsilon')}-\bar{\epsilon}\tag{3.5}$$

where

$$\begin{aligned}
&c_1'(1-2\epsilon)\leq c_1=\frac{(s+1)^2(1-\epsilon)}{(s-1)^2(1+\epsilon)}\leq c_1'(1-\epsilon),\\
&c_2=\frac{(1+1/s)^4}{1+\epsilon}=\frac{c_2'}{1+\epsilon},\\
&\bar{\epsilon}=\frac{4(s^2+1)}{(s-1)^2s\sqrt{1+\epsilon}}+\frac{1}{(1+\epsilon)s^2},\\
&\epsilon'=2\sqrt{(1+\epsilon)\bar{c}}/s.
\end{aligned}$$

Here, $c_1'=(s+1)^2/(s-1)^2, c_2'=(1+1/s)^4$.

In obtaining the above inequality (3.5), we utilize the fact that $H^{(t-1)}\leq 1$.

Based on the bound provided in (3.5), the result of Theorem 1 can be proved by

induction. For the PC obtained from the first batch, $\{\mathbf{w}_j^{(1)}\}_{j=1}^d$, we have that

$$H^{(1)} \geq c_1 - \frac{c_2 \sum_{j=1}^{d} \sum_{i=1}^{\lambda b} \left(\mathbf{w}^{(0)}{}_j^T \mathbf{o}_i \right)^2 (1 - \Gamma_o)}{(1-\lambda) b ((1-\epsilon) H^{(0)} - \epsilon')} - \bar{\epsilon}.$$

When the initial solution $\{\mathbf{w}_j^{(0)}\}_{j=1}^d$ satisfies the following conditions:

$$\sum_{i=1}^{\lambda b} \sum_{j=1}^{d} \left(\mathbf{w}_j^{(0)T} \mathbf{o}_i \right)^2 \leq \frac{(1-\lambda) b (1-\epsilon)}{c_2 (1-\Gamma_o)} \left(\frac{1}{4} (c_1 + \epsilon_1)^2 - \epsilon_2 \right) \tag{3.6}$$

and

$$H^{(0)} \geq \frac{1}{2}(c_1 + \epsilon_1) - \sqrt{\frac{1}{4}(c_1 + \epsilon_1)^2 - \epsilon_2 - \frac{c_2 \sum_{i=1}^{\lambda b} \sum_{j=1}^{d} \left(\mathbf{w}_j^{(0)T} \mathbf{w}_o \right)^2 (1 - \Gamma_o)}{(1-\lambda) b (1-\epsilon)}}, \tag{3.7}$$

where

$$\epsilon_1 = \frac{\epsilon'}{1-\epsilon} - \bar{\epsilon}, \text{ and } \epsilon_2 = \frac{\epsilon'(c_1 - \bar{\epsilon})}{1-\epsilon},$$

we can verify that

$$H^{(1)} \geq c_1 - \frac{c_2 \sum_{j=1}^{d} \sum_{i=1}^{\lambda b} \left(\mathbf{w}_j^{(0)T} \mathbf{w}_o \right)^2 (1 - \Gamma_o)}{(1-\lambda) b ((1-\epsilon) H_0 - \epsilon')} - \bar{\epsilon} \geq H^{(0)}.$$

Thus according to Lemma 9, we have

$$\sum_{i=1}^{\lambda b} \sum_{j=1}^{d} \left(\mathbf{w}_j^{(1)T} \mathbf{o}_i \right)^2 \leq \sum_{i=1}^{\lambda b} \sum_{j=1}^{d} \left(\mathbf{w}_j^{(0)T} \mathbf{o}_i \right)^2.$$

Similarly, suppose for the solution in $(t-1)$-th trial, we have $\sum_{i=1}^{\lambda b} \sum_{j=1}^{d} (\mathbf{w}_j^{(t-1)T} \mathbf{o}_i)^2 \leq \sum_{i=1}^{\lambda b} \sum_{j=1}^{d} (\mathbf{w}_j^{(0)T} \mathbf{o}_i)^2$. Thus,

$$
\begin{aligned}
H^{(t)} &\geq c_1 - \frac{c_2 \sum_{i=1}^{\lambda b} \sum_{j=1}^{d} \left(\mathbf{w}_j^{(t-1)^T} \mathbf{o}_i \right)^2 (1 - \Gamma_o)}{(1-\lambda) b ((1-\epsilon) H^{(t-1)} - \epsilon')} - \bar{\epsilon} \\
&\geq c_1'(1-2\epsilon) - \frac{c_2' \sum_{i=1}^{\lambda b} \sum_{j=1}^{d} \left(\mathbf{w}_j^{(0)^T} \mathbf{o}_i \right)^2 (1 - \Gamma_o)}{(1+\epsilon)(1-\lambda) b ((1-\epsilon) H^{(t-1)} - \epsilon')} - \bar{\epsilon}
\end{aligned}
$$

And we can verify that when the initial solution satisfies the conditions (3.6) and (3.7), the performance of the new solution will be improved, namely

$$
H^{(t)} \geq H^{(t-1)}.
$$

And thus $\sum_{i=1}^{\lambda b} \sum_{j=1}^{d} \left(\mathbf{w}_j^{(t-1)^T} \mathbf{o}_i \right)^2$ keeps decreasing according to Lemma 9.

Finally, by letting

$$
c_1'(1-2\epsilon) - \frac{c_2' \sum_{i=1}^{\lambda b} \sum_{j=1}^{d} \left(\mathbf{w}_j^{(0)^T} \mathbf{o}_i \right)^2 (1 - \Gamma_o)}{(1+\epsilon)(1-\lambda) b ((1-\epsilon) H_{t-1} - \epsilon')} - \bar{\epsilon} = H^{(t)},
$$

we can solve out that

$$
H^{(t)} = \frac{1}{2}(c_1'(1-2\epsilon)+\epsilon_1) + \sqrt{\frac{(c_1'(1-2\epsilon)+\epsilon_1)^2}{4} - \epsilon_2 - \frac{c_2' \sum_{i=1}^{\lambda b} \sum_{j=1}^{d} \left(\mathbf{w}_j^{(0)^T} \mathbf{o}_i \right)^2 (1 - \Gamma_o)}{(1-\lambda) b (1-\epsilon^2)}}.
$$

Namely, the final performance will converge as above. □

3.13 Chapter Summary

In this chapter, we proposed an online robust PCA (online RPCA) algorithm for samples corrupted by outliers. The online RPCA alternates between standard PCA for updating PCs and probabilistic selection of the new samples which alleviates the impact of outliers. Theoretical analysis showed that the online RPCA could improve the PC estimation steadily and provided results with bounded deviation from the optimum. To the best of our knowledge, this is the first work to investigate such

online robust PCA problem with theoretical performance guarantee. The proposed online robust PCA algorithm can be applied to handle challenges imposed by the modern big data analysis.

Chapter 4

Online Optimization for Robust PCA

In this chapter, we introduce an online robust PCA method which handles a different setting from the above chapter. In the above chapter, some of the samples are completely corrupted and are outliers. In this chapter, we handle the data each of which is corrupted by sparse gross noises.

4.1 Introduction

Principal Component Analysis (PCA) [57] is arguably the most widely used method for dimensionality reduction in data analysis. However, standard PCA is brittle in the presence of outliers and corruptions [51]. Thus many techniques have been developed towards robustifying it [52, 44, 61, 62, 47]. One prominent example is the Principal Component Pursuit (PCP) method proposed in [44] that robustly finds the low-dimensional subspace through decomposing the sample matrix into a low-rank component and an overall sparse component. It is proved that both components can be recovered exactly through minimizing a weighted combination of the nuclear norm of the first term and ℓ_1 norm of the second one. Thus the subspace estimation is robust to sparse corruptions.

However, PCP and other robust PCA methods are all implemented in a batch

manner. They need to access every sample in each iteration of the optimization. Thus, robust PCA methods require memorizing all samples, in sharp contrast to standard PCA where only the covariance matrix is needed. This pitfall severely limits their scalability to *big data*, which are becoming ubiquitous now. Moreover, for an incremental samples set, when a new sample is added, the optimization procedure has to be re-implemented on all available samples. This is quite inefficient in dealing with incremental sample sets such as network detection, video analysis and abnormal events tracking.

Another pitfall of batch robust PCA methods is that they cannot handle the case where the underlying subspaces are changing gradually. For example, in the video background modeling, the background is assumed to be static across different frames for applying robust PCA [44]. Such assumption is too restrictive in practice. A more realistic situation is that the background is changed gradually along with the camera moving, corresponding to a gradually changing subspace. Unfortunately, traditional batch RPCA methods may fail in this case.

In order to efficiently and robustly estimate the subspace of a large-scale or dynamic samples set, we propose an Online Robust PCA (OR-PCA) method. OR-PCA processes only one sample per time instance and thus is able to efficiently handle big data and dynamic sample sets, saving the memory cost and dynamically estimating the subspace of evolutional samples. We briefly explain our intuition here. The major difficulty of implementing the previous RPCA methods, such as PCP, in an online fashion is that the adopted nuclear norm tightly couples the samples and thus the samples have to be processed simultaneously. To tackle this, OR-PCA pursues the low-rank component in a different manner: using an equivalent form of the nuclear norm, OR-PCA explicitly decomposes the sample matrix into the *multiplication of the subspace basis and coefficients* plus a sparse noise component. Through such decomposition, the samples are decoupled in the optimization and can be processed separately. In particular, the optimization consists of two iterative updating components. The first one is to project the sample onto the current basis and isolate the sparse noise (explaining the outlier contamination), and the second

one is to update the basis given the new sample.

Our main technical contribution is to show the above mentioned iterative optimization sheme converges to the *global optimal* solution of the original PCP formulation, thus we establish the validity of our online method. Our proof is inspired by recent results from [56], who proposed an online dictionary learning method and provided the convergence guarantee of the proposed online dictionary learning method. However, [56] can only guarantee that the solution converges to a stationary point of the optimization problem.

Besides the nice behavior on single subspace recovering, OR-PCA can also be applied for tracking time-variant subspace naturally, since it updates the subspace estimation timely after revealing one new sample. We conduct comprehensive simulations to demonstrate the advantages of OR-PCA for both subspace recovering and tracking in this work.

4.2 Related Work

The robust PCA algorithms based on nuclear norm minimization to recover low-rank matrices are now standard, since the seminal works [59, 46]. Recent works [44, 45] have taken the nuclear norm minimization approach to the decomposition of a low-rank matrix and an overall sparse matrix. Different from the setting of samples being corrupted by sparse noise, Xu *et al.* [62, 61] solve robust PCA in the case that a few samples are completely corrupted. Following the work of [61], Feng *et al.* [47] propose deterministic high-dimensional RPCA which achieves high efficiency. However, all of these RPCA methods are implemented in batch manner and cannot be directly used in online manner.

The most related works to ours are the following two works. In [53], an incremental and robust subspace learning method is proposed. The method proposes to integrate the M-estimation into the standard incremental PCA calculation. Specifically, each newly coming data point is re-weighted by a pre-defined influence function [51] of its residual to the current estimated subspace. However, no performance guaran-

tee is provided in this work. Moreover, the performance of the proposed algorithm relies on the accuracy of PCs obtained previously. And the error will be cumulated inevitably. Recently, a compressive sensing based recursive robust PCA algorithm is proposed in [58]. The proposed method in [58] essentially solves compressive sensing optimization over a small batch of data to update the PCs estimation instead of using a single sample, and it is not clear how to extend the method to the latter case.

As aforementioned, Mairal *et al.* [56] propose an online learning method for dictionary learning and sparse coding. Based on that work, Guan *et al.* [49] propose an online nonnegative matrix factorization method. Both of these two works can be seen as online matrix factorization problem with specific constraints (sparse or nonnegative). Though it can also be seen as a kind of matrix factorization, the method proposed in this work is essentially different from those two works. In this work, an additive sparse noise matrix is considered along with the matrix factorization. Thus the optimization and analysis are different from the ones in those works. Benefitting from explicitly considering the noise, the proposed method possesses certain robustness, which is absent in either the dictionary learning or nonnegative matrix factorization works.

4.3 Problem Formulation

4.3.1 Notation

We use bold letters to denote vectors. In particular, $\mathbf{x} \in \mathbb{R}^p$ denotes an authentic sample without corruption, $\mathbf{e} \in \mathbb{R}^p$ is for the noise, and $\mathbf{z} \in \mathbb{R}^p$ is for the corrupted observation $\mathbf{z} = \mathbf{x} + \mathbf{e}$. Here p denotes the ambient dimension of the observed samples. Let r denote the intrinsic dimension of the subspace underlying $\{\mathbf{x}_i\}_{i=1}^n$. Let n denote the number of observed samples, t denote the index of the sample/time instance. We use capital letters to denote matrices, *e.g.*, $Z \in \mathbb{R}^{p\times n}$ is the matrix of observed samples. Each column $\mathbf{z}_i$ of Z corresponds to one sample. For an arbitrary real matrix E, Let $\|E\|_F$ denote its Frobenius norm, $\|E\|_{\ell_1} = \sum_{i,j} |E_{ij}|$ denote the

ℓ_1-norm of E seen as a long vector in $\mathbb{R}^{p\times n}$, and $\|E\|_* = \sum_i \sigma_i(E)$ denote its nuclear norm, *i.e.*, the sum of its singular values.

4.3.2 Objective Function Formulation

Robust PCA (RPCA) aims to accurately estimate the subspace underlying the observed samples, even though the samples are corrupted by gross but sparse noise. As one of the most popular RPCA methods, the Principal Component Pursuit (PCP) method [44] proposes to solve RPCA by decomposing the observed sample matrix Z into a low-rank component X accounting for the low-dimensional subspace plus an overall sparse component E incorporating the sparse corruption. Under mild conditions, PCP guarantees that the two components X and E can be exactly recovered through solving:

$$\min_{X,E} \frac{1}{2}\|Z - X - E\|_F^2 + \lambda_1\|X\|_* + \lambda_2\|E\|_1. \tag{4.1}$$

To solve the problem in (4.1), iterative optimization methods such as Accelerated Proximal Gradient (APG) [55] or Augmented Lagrangian Multiplier (ALM) [54] methods are often used. However, these optimization methods are implemented in a batch manner. In each iteration of the optimization, they need to access all samples to perform SVD. Hence a huge storage cost is incurred when solving RPCA for big data (*e.g.*, web data, large image set).

In this paper, we consider online implementation of PCP. The main difficulty is that the nuclear norm couples all the samples tightly and thus the samples cannot be considered separately as in typical online optimization problems. To overcome this difficulty, we use an equivalent form of the nuclear norm for the matrix X whose rank is upper bounded by r, as follows [59],

$$\|X\|_* = \inf_{L\in\mathbb{R}^{p\times r}, R\in\mathbb{R}^{n\times r}} \left\{ \frac{1}{2}\|L\|_F^2 + \frac{1}{2}\|R\|_F^2 : X = LR^T \right\}.$$

Namely, the nuclear norm is re-formulated as an explicit low-rank factorization of

X. Such nuclear norm factorization is developed in [43] and well established in recent works [60, 59]. In this decomposition, $L \in \mathbb{R}^{p\times r}$ can be seen as the basis of the low-dimensional subspace and $R \in \mathbb{R}^{n\times r}$ denotes the coefficients of the samples w.r.t. the basis. Thus, the RPCA problem (4.1) can be re-formulated as

$$\min_{X,L\in\mathbb{R}^{p\times r},R\in\mathbb{R}^{n\times r},E} \frac{1}{2}\|Z - X - E\|_F^2 + \frac{\lambda_1}{2}(\|L\|_F^2 + \|R\|_F^2) + \lambda_2\|E\|_1, \text{ s.t. } X = LR^T.$$

Substituting X by LR^T and removing the constraint, the above problem is equivalent to:

$$\min_{L\in\mathbb{R}^{p\times r},R\in\mathbb{R}^{n\times r},E} \frac{1}{2}\|Z - LR^T - E\|_F^2 + \frac{\lambda_1}{2}(\|L\|_F^2 + \|R\|_F^2) + \lambda_2\|E\|_1. \tag{4.2}$$

Though the reformulated objective function is not jointly convex w.r.t. the variables L and R, we prove below that the local minima of (4.2) are global optimal solutions to original problem in (4.1). The details are given in the next section.

Given a finite set of samples $Z = [\mathbf{z}_1, \ldots, \mathbf{z}_n] \in \mathbb{R}^{p\times n}$, solving problem (4.2) indeed minimizes the following *empirical cost function*,

$$f_n(L) \triangleq \frac{1}{n}\sum_{i=1}^{n} \ell(\mathbf{z}_i, L) + \frac{\lambda_1}{2n}\|L\|_F^2, \tag{4.3}$$

where the *loss function* for each sample is defined as

$$\ell(\mathbf{z}_i, L) \triangleq \min_{\mathbf{r},\mathbf{e}} \frac{1}{2}\|\mathbf{z}_i - L\mathbf{r} - \mathbf{e}\|_2^2 + \frac{\lambda_1}{2}\|\mathbf{r}\|_2^2 + \lambda_2\|\mathbf{e}\|_1. \tag{4.4}$$

The loss function measures the representation error for the sample $\mathbf{z}$ on a fixed basis L, where the coefficients on the basis $\mathbf{r}$ and the sparse noise $\mathbf{e}$ associated with each sample are optimized to minimize the loss. In the stochastic optimization, one is usually interested in the minimization of the *expected cost* overall all the samples [56],

$$f(L) \triangleq \mathbb{E}_{\mathbf{z}}[\ell(\mathbf{z}, L)] = \lim_{n\to\infty} f_n(L), \tag{4.5}$$

where the expectation is taken w.r.t. the distribution of the samples $\mathbf{z}$. In this work,

we first establish a surrogate function for this expected cost and then optimize the surrogate function for obtaining the subspace estimation in an online fashion.

4.4 Stochastic Optimization Algorithm for OR-PCA

We now present our Online Robust PCA (OR-PCA) algorithm. The main idea is to develop a stochastic optimization algorithm to minimize the empirical cost function (4.3), which *processes one sample per time instance* in an online manner. The coefficients $\mathbf{r}$, noise $\mathbf{e}$ and basis L are optimized in an alternative manner. In the t-th time instance, we obtain the estimation of the basis L_t through minimizing the cumulative loss w.r.t. the previously estimated coefficients $\{\mathbf{r}_i\}_{i=1}^t$ and sparse noise $\{\mathbf{e}_i\}_{i=1}^t$. The objective function for updating the basis L_t is defined as,

$$g_t(L) \triangleq \frac{1}{t}\sum_{i=1}^{t}\left(\frac{1}{2}\|\mathbf{z}_i - L\mathbf{r}_i - \mathbf{e}_i\|_2^2 + \frac{\lambda_1}{2}\|\mathbf{r}_i\|_2^2 + \lambda_2\|\mathbf{e}_i\|_1\right) + \frac{\lambda_1}{2t}\|L\|_F^2. \tag{4.6}$$

This is a surrogate function of the empirical cost function $f_t(L)$ defined in (4.3), i.e., it provides an upper bound for $f_t(L)$: $g_t(L) \geq f_t(L)$.

The proposed algorithm is summarized in Algorithm 3. Here, the subproblem in (4.7) involves solving a small-size convex optimization problem, which can be solved efficiently by the off-the-shelf solver (see the supplementary material). To update the basis matrix L, we adopt the block-coordinate descent with warm restarts [42]. In particular, each column of the basis L is updated individually while fixing the other columns.

The following theorem is the main theoretic result of the paper, which states that the solution from Algorithm 3 will converge to the optimal solution of the batch optimization. Thus, the proposed OR-PCA converges to the correct low-dimensional subspace even in the presence of sparse noise, as long as the batch version – PCP – works.

Theorem 11. *Assume the observations are always bounded. Given the rank of the optimal solution to* (4.5) *is provided as r, and the solution $L_t \in \mathbb{R}^{p\times r}$ provided by*

Algorithm 3 is full rank, then L_t converges to the optimal solution of (4.5) *asymptotically.*

Note that the assumption that observations are bounded is quite natural for the realistic data (such as images, videos). We find in the experiments that the final solution L_t is always full rank. A standard stochastic gradient descent method may further enhance the computational efficiency, compared with the used method here. We leave the investigation for future research.

Algorithm 3 Stochastic Optimization for OR-PCA

Input: $\{\mathbf{z}_1, \ldots, \mathbf{z}_T\}$ (observed data which are revealed sequentially), $\lambda_1, \lambda_2 \in \mathbb{R}$ (regularization parameters), $L_0 \in \mathbb{R}^{p \times r}$, $\mathbf{r}_0 \in \mathbb{R}^r$, $\mathbf{e}_0 \in \mathbb{R}^p$ (initial solutions), T (number of iterations).

for $t = 1$ to T **do**

1) Reveal the sample $\mathbf{z}_t$.

2) Project the new sample:

$$\{\mathbf{r}_t, \mathbf{e}_t\} = \arg\min \frac{1}{2}\|\mathbf{z}_t - L_{t-1}\mathbf{r} - \mathbf{e}\|_2^2 + \frac{\lambda_1}{2}\|\mathbf{r}\|_2^2 + \lambda_2\|\mathbf{e}\|_1. \tag{4.7}$$

3) $A_t \leftarrow A_{t-1} + \mathbf{r}_t\mathbf{r}_t^T$, $B_t \leftarrow B_{t-1} + (\mathbf{z}_t - \mathbf{e}_t)\mathbf{r}_t^T$.

4) Compute L_t with L_{t-1} as warm restart using Algorithm 4:

$$L_t \triangleq \arg\min \frac{1}{2}\mathrm{Tr}\left[L^T\left(A_t + \lambda_1 I\right)L\right] - \mathrm{Tr}(L^T B_t). \tag{4.8}$$

end for

Return $X_T = L_T R_T^T$ (low-rank data matrix), E_T (sparse noise matrix).

Algorithm 4 The Basis Update

Input: $L = [\mathbf{l}_1, \ldots, \mathbf{l}_r] \in \mathbb{R}^{p \times r}$, $A = [\mathbf{a}_1, \ldots, \mathbf{a}_r] \in \mathbb{R}^{r \times r}$, and $B = [\mathbf{b}_1, \ldots, \mathbf{b}_r] \in \mathbb{R}^{p \times r}$.

$\tilde{A} \leftarrow A + \lambda_1 I$.

for $j = 1$ to r **do**

$$\mathbf{l}_j \leftarrow \frac{1}{\tilde{A}_{j,j}}(\mathbf{b}_j - L\tilde{\mathbf{a}}_j) + \mathbf{l}_j. \tag{4.9}$$

end for

Return L.

4.5 Algorithm solving Problem (4.7)

For the data projection $\mathbf{r}$ and noise estimation $\mathbf{e}$, we can get the closed-form solutions for them respectively, as shown in Algorithm 5. In particular, the closed-form solution to a projection to ℓ_1-ball in updating $\mathbf{e}$ involves a soft thresholding operator $\mathcal{S}_\lambda[\cdot]$ [131], which is defined as:

$$\mathcal{S}_\lambda[x] \triangleq \begin{cases} x-\lambda, & \text{if } x > \lambda, \\ x+\lambda, & \text{if } x < -\lambda, \\ 0, & \text{otherwise.} \end{cases}$$

And it is conducted element-wisely on the involved vectors. The optimization iteration is terminated when the following convergence criterion is met:

$$\max(\|\mathbf{r}_{k+1}-\mathbf{r}_k\|/\|\mathbf{z}\|, \|\mathbf{e}_{k+1}-\mathbf{e}_k\|/\|\mathbf{z}\|) < \varepsilon.$$

Here ε is set as 1×10^{-6} throughout the simulations.

The details of the algorithm are summarized as follows,

Algorithm 5 Data Projection

Input: $L = [\mathbf{l}_1, \ldots, \mathbf{l}_r] \in \mathbb{R}^{p\times r}$ (input basis), $\mathbf{z} \in \mathbb{R}^p$, parameters λ_1 and λ_2.
$\mathbf{e} \leftarrow \mathbf{0}$.
while not converged **do**
Update the coefficient $\mathbf{r}$:

$$\mathbf{r} \leftarrow (L^T L + \lambda_1 I)^{-1} L^T (\mathbf{z}-\mathbf{e}).$$

Update the sparse error $\mathbf{e}$:

$$\mathbf{e} \leftarrow \mathcal{S}_{\lambda_2}[\mathbf{z} - L\mathbf{r}].$$

end while
Return L.

4.6 Proof Sketch

In this section we sketch the proof of Theorem 11. The details are deferred to the supplementary material due to space limit.

The proof of Theorem 11 proceeds in the following four steps: (I) we first prove that the surrogate function $g_t(L_t)$ converges almost surely; (II) we then prove that the solution difference behaves as $\|L_t - L_{t-1}\|_F = O(1/t)$; (III) based on (II) we show that $f(L_t) - g_t(L_t) \to 0$ almost surely, and the gradient of f vanishes at the solution L_t when $t \to \infty$; (IV) finally we prove that L_t actually converges to the optimum solution of the problem (4.5).

Theorem 12 (Convergence of the surrogate function g_t)**.** *Let g_t denote the surrogate function defined in* (4.6)*. Then, $g_t(L_t)$ converges almost surely when the solution L_t is given by Algorithm 3.*

We prove Theorem 12, i.e., the convergence of the stochastic positive process $g_t(L_t) > 0$, by showing that it is a quasi-martingale. We first show that the summation of the positive difference of $g_t(L_t)$ is bounded utilizing the fact that $g_t(L_t)$ upper bounds the empirical cost $f_t(L_t)$ and the loss function $\ell(\mathbf{z}_t, L_t)$ is Lipschitz. These imply that $g_t(L_t)$ is a quasi-martingale. Applying the lemma from [48] about the convergence of quasi-martingale, we conclude that $g_t(L_t)$ converges.

Next, we show the difference of the two successive solutions converges to 0 as t goes to infinity.

Theorem 13 (Difference of the solution L_t)**.** *For the two successive solutions obtained from Algorithm 3, we have*

$$\|L_{t+1} - L_t\|_F = O(1/t) \quad a.s.$$

To prove the above result, we first show that the function $g_t(L)$ is strictly convex. This holds since the regularization component $\lambda_1\|L\|_F^2$ naturally guarantees that the eigenvalues of the Hessian matrix are bounded away from zero. Notice that this is essentially different from [56], where one has to assume that the smallest eigenvalue

of the Hessian matrix is lower bounded. Then we further show that variation of the function $g_t(L)$, $g_t(L_t) - g_{t+1}(L_t)$, is Lipschitz if using the updating rule shown in Algorithm 4. Combining these two properties establishes Theorem 13.

In the third step, we show that the expected cost function $f(L_t)$ is a smooth one, and the difference $f(L_t) - g_t(L_t)$ goes to zero when $t \to \infty$. In order for showing the regularity of the function $f(L_t)$, we first provide the following optimality condition of the loss function $\ell(L_t)$.

Lemma 12 (Optimality conditions of Problem (4.4))**.** $\mathbf{r}^\star \in \mathbb{R}^r$ *and* $\mathbf{e}^\star \in \mathbb{R}^p$ *is a solution of Problem* (4.4) *if and only if*

$$
\begin{aligned}
&C_\Lambda(\mathbf{z}_\Lambda - \mathbf{e}^\star_\Lambda) = \lambda_2 \text{sign}(\mathbf{e}^\star_\Lambda),\\
&|C_{\Lambda^c}(\mathbf{z}_{\Lambda^c} - \mathbf{e}^\star_{\Lambda^c})| \leq \lambda_2, \ \textit{otherwise},\\
&\mathbf{r}^\star = (L^T L + \lambda_1 I)^{-1} L^T (\mathbf{z} - \mathbf{e}^\star),
\end{aligned}
$$

where $C = I - L(L^T L + \lambda_1 I)^{-1} L^T$ *and* C_Λ *denotes the columns of matrix* C *indexed by* $\Lambda = \{j | \mathbf{e}^\star[j] \neq 0\}$ *and* Λ^c *denotes the complementary set of* Λ*. Moreover, the optimal solution is unique.*

Based on the above lemma, we can prove that the solution $\mathbf{r}^\star$ and $\mathbf{e}^\star$ are Lipschitz w.r.t. the basis L. Then, we can obtain the following results about the regularity of the expected cost function f.

Lemma 13. *Assume the observations* $\mathbf{z}$ *are always bounded. Define*

$$
\{\mathbf{r}^\star, \mathbf{e}^\star\} = \arg\min_{\mathbf{r},\mathbf{e}} \frac{1}{2}\|\mathbf{z} - L\mathbf{r} - \mathbf{e}\|_2^2 + \frac{\lambda_1}{2}\|\mathbf{r}\|_2^2 + \lambda_2 \|\mathbf{e}\|_1.
$$

Then, 1) the function ℓ *defined in* (4.4) *is continuously differentiable and*

$$
\nabla_L \ell(\mathbf{z}, L) = (L\mathbf{r}^\star + \mathbf{e}^\star - \mathbf{z})\mathbf{r}^{\star T};
$$

2) $\nabla f(L) = \mathbb{E}_{\mathbf{z}}[\nabla_L \ell(\mathbf{z}, L)]$*; and 3)*$\nabla f(L)$ *is Lipschitz.*

Equipped with the above regularities of the expected cost function f, we can prove the convergence of f, as stated in the following theorem.

Theorem 14 (Convergence of f)**.** *Let g_t denote the surrogate function defined in* (4.2)*. Then, 1) $f(L_t) - g_t(L_t)$ converges almost surely to* 0*; and 2) $f(L_t)$ converges almost surely, when the solution L_t is given by Algorithm 3.*

Following the techniques developed in [56], we can show the solution obtained from Algorithm 3, L_∞, satisfies the first order optimality condition for minimizing the expected cost $f(L)$. Thus the OR-PCA algorithm provides a solution converging to a *stationary point* of the expected loss.

Theorem 15. *The first order optimal condition for minimizing the objective function in* (4.5) *is satisfied by L_t, the solution provided by Algorithm 3, when t tends to infinity.*

Finally, to complete the proof, we establish the following result stating that any full-rank L that satisfies the first order condition is the *global optimal solution.*

Theorem 16. *When the solution L satisfies the first order condition for minimizing the objective function in* (4.5) *, the obtained solution L is the optimal solution of the problem* (4.5) *if L is full rank.*

Combining Theorem 15 and Theorem 16 directly yields Theorem 11 – the solution from Algorithm 3 converges to the optimal solution of Problem (4.5) asymptotically.

4.7 Proof of Lemma 12

Proof. Denote the subgradient of $\|\mathbf{e}\|_1$ as $\partial\|\mathbf{e}\|_1$ and it is known that

$$\partial\|\mathbf{e}\|_1 = \{\mathbf{u}|\mathbf{u}_i = \text{sign}(\mathbf{e}_i) \text{ if } \mathbf{e}_i \neq 0, \text{ and } |\mathbf{e}_i| \leq 1 \text{ otherwise}\}.$$

The point $(\mathbf{r}^\star, \mathbf{e}^\star)$ is a global minimum of (5) if and only if the vector zero is in its subgradient at $(\mathbf{r}^\star, \mathbf{e}^\star)$:

$$\exists \mathbf{u} \in \partial \|\mathbf{e}^\star\|_1 \text{ such that } \mathbf{e}^\star + L\mathbf{r}^\star - \mathbf{z} + \lambda_2 \mathbf{u} = 0, \tag{4.10}$$

$$-L^T\mathbf{z} + L^T L\mathbf{r}^\star + L^T\mathbf{e}^\star + \lambda_1 \mathbf{r}^\star = 0. \tag{4.11}$$

From (4.11), we have,

$$\mathbf{r}^\star = (L^T L + \lambda_1 I)^{-1} L^T (\mathbf{z} - \mathbf{e}^\star).$$

This proves the third inequality in the lemma. Substituting back into (4.10) yields

$$\left(I - L(L^T L + \lambda_1 I)^{-1} L^T\right)(\mathbf{z} - \mathbf{e}^\star) = \lambda_2 \mathbf{u}, \text{where } \mathbf{u} \in \partial \|\mathbf{e}^\star\|_1.$$

Define the matrix

$$C \triangleq I - L(L^T L + \lambda_1 I)^{-1} L^T.$$

According to Woodbury matrix identity, we have

$$C = \left(I + \frac{1}{\lambda_1} L^T L\right)^{-1}.$$

Thus C is invertible. We then have

$$C(\mathbf{z} - \mathbf{e}^\star) = \lambda_2 \mathbf{u}, \text{where } \mathbf{u} \in \partial \|\mathbf{e}^\star\|_1.$$

Let $\Lambda = \{j | \mathbf{e}^\star[j] \neq 0\}$ be the index set of nonzero elements of the optimal solution $\mathbf{e}^\star$. Then we can show that

$$C_\Lambda(\mathbf{z}_\Lambda - \mathbf{e}^\star_\Lambda) = \lambda_2 \text{sign}(\mathbf{e}^\star_\Lambda)$$

Here C_Λ denotes submatrix of C consisting of the column vectors of matrix C

indexed by Λ. Then we can solve out that

$$\mathbf{e}^\star_\Lambda = (C^T_\Lambda C_\Lambda)^{-1}(C_\Lambda \mathbf{z}_\Lambda - \lambda \text{sign}(\mathbf{e}^\star_\Lambda)),$$
$$\mathbf{e}^\star_{\Lambda^c} = 0.$$

Since C is invertible, C is column full rank. Thus C_Λ is column full rank and $C^T_\Lambda C_\Lambda$ is invertible, the solution $\mathbf{e}^\star$ is unique and thus $\mathbf{r}^\star$ is also unique. □

4.8 Proof of Lemma 13

Proof. To reveal the regularity of the expected loss function f and its derivative ∇f, we need first to prove the regularity of the loss function ℓ as stated in the first claim.

Proof of the first claim

Define a function $\tilde{f}$ as

$$\tilde{f}(\mathbf{r}, \mathbf{e}, \mathbf{z}, L) \triangleq \frac{1}{2}\|\mathbf{z} - L\mathbf{r} - \mathbf{e}\|_2^2 + \frac{\lambda_1}{2}\|L\|_F^2 + \frac{\lambda_1}{2}\|\mathbf{r}\|_2^2 + \lambda_2\|\mathbf{e}\|_1.$$

Thus the loss function ℓ can be expressed as

$$\ell(\mathbf{z}, L) = \min_{\mathbf{r},\mathbf{e}} \tilde{f}(\mathbf{r}, \mathbf{e}, \mathbf{z}, L).$$

The function $\tilde{f}(\mathbf{r}, \mathbf{e}, \mathbf{z}, L)$ is continuous, and for all $\mathbf{r} \in \mathbb{R}^r, \mathbf{e} \in \mathbb{R}^p$, the function $\tilde{f}(\mathbf{r}, \mathbf{e}, \cdot, \cdot)$ is differentiable, and the derivative $\nabla_L \tilde{f}(\mathbf{r}, \mathbf{e}, \cdot, \cdot) = (L\mathbf{r} + \mathbf{e} - \mathbf{z})\mathbf{r}^T$ is continuous. Furthermore, according to Lemma 1, $\tilde{f}(\cdot, \cdot, \mathbf{z}, L)$ has unique minimizer $(\mathbf{r}^\star, \mathbf{e}^\star)$, thus Lemma 3 directly applies and we obtain that $\ell(\mathbf{z}, L)$ is differentiable in L and

$$\nabla_L \ell(\mathbf{z}, L) = \nabla_L \tilde{f}(\mathbf{r}^\star, \mathbf{e}^\star, \mathbf{z}, L) = (L\mathbf{r}^\star + \mathbf{e}^\star - \mathbf{z})\mathbf{r}^{\star T} + \lambda_1 L.$$

Thus, we complete the proof of the first claim.

Proof of the second claim

According to the first claim, the function $\ell_{\mathbf{z},L}$ is continuously differentiable, thus

$$\nabla_L f(L) = \nabla_L \mathbb{E}_{\mathbf{z}}[\ell(\mathbf{z}, L)] = \mathbb{E}_{\mathbf{z}}[\nabla_L \ell(\mathbf{z}, L)].$$

Equipped with the above two results, we are ready to prove that the derivative $\nabla_L f(L)$ is Lipschitz.

Proof of the third claim

To prove that $\nabla f(L)$ is Lipschitz, we will show that for all bounded observations $\mathbf{z}$, $\mathbf{r}^\star(\mathbf{z}, \cdot)$ and $\mathbf{e}^\star(\mathbf{z}, \cdot)$ are Lipschitz with constants independent of $\mathbf{z}$. First, the loss function $\ell(\mathbf{z}, L)$ defined in (4) is continuous in $\mathbf{r}, \mathbf{e}, L, \mathbf{z}$ and has a unique minimum (according to Lemma 1) for fixed $\mathbf{z}$ and L, thus the optimal solutions $\mathbf{r}^\star$ and $\mathbf{e}^\star$ are continuous in L and $\mathbf{z}$.

Consider a matrix L and a sample $\mathbf{z}$, and denote $\mathbf{r}^\star$ and $\mathbf{e}^\star$ as the corresponding optimal solutions. Denote by Λ the set of the indices such that $|C_\Lambda(\mathbf{z}_\Lambda - \mathbf{e}^\star_\Lambda)| = \lambda_1$ (see Lemma 1). Here the matrix C is defined as $C = I - L(L^T L + \lambda_1 I)^{-1} L^T$. Since C_Λ is nonsingular, $C_\Lambda(\mathbf{z}_\Lambda - \mathbf{e}^\star_\Lambda)$ is continuous in L and $\mathbf{z}$. Thus we consider a small perturbation of $(\mathbf{z}, L)$ in one of their open neighborhood V, such that for all $(\mathbf{z}', L')$ in V, we have if $j \notin \Lambda$, $\left|C'_j(\mathbf{z}'[j] - \mathbf{e}^{\star\prime}[j])\right| < \lambda_2$ and $\mathbf{e}^{\star\prime}[j] = 0$, where $\mathbf{e}^{\star\prime} = \mathbf{e}^\star(\mathbf{z}', L')$. Namely the support set of $\mathbf{e}^\star$ is not changed.

Based on the about continuity, we consider the following function

$$\tilde{\ell}(\mathbf{z}_\Lambda, L_\Lambda, \mathbf{r}, \mathbf{e}_\Lambda) \triangleq \frac{1}{2}\|\mathbf{z}_\Lambda - L_\Lambda \mathbf{r} - \mathbf{e}_\Lambda\|_2^2 + \frac{\lambda_1}{2}\|L_\Lambda\|_F^2 + \frac{\lambda_1}{2}\|\mathbf{r}\|_2^2 + \lambda_2\|\mathbf{e}_\Lambda\|_1.$$

Since the Hessian matrix of the function $\tilde{\ell}(\mathbf{z}_\Lambda, L_\Lambda, \cdot, \cdot)$ w.r.t. $\mathbf{r}$, $I \otimes (L_\Lambda^T L_\Lambda + \lambda_1 I)$, and the Hessian matrix w.r.t. $\mathbf{e}_\Lambda$, $I \otimes \lambda_2 I$, are positive definite, we have the function

$\tilde{\ell}(\mathbf{z}_\Lambda, L_\Lambda, \cdot, \cdot)$ is strictly convex and

$$\begin{aligned} & \tilde{\ell}(\mathbf{z}_\Lambda, L_\Lambda, \mathbf{r}^{\star\prime}, \mathbf{e}_\Lambda^{\star\prime}) - \tilde{\ell}(\mathbf{z}_\Lambda, L_\Lambda, \mathbf{r}^\star, \mathbf{e}_\Lambda^\star) \\ \geq & \ \lambda_1 \|\mathbf{r}^{\star\prime} - \mathbf{r}^\star\|_2^2 + \lambda_2 \|\mathbf{e}_\Lambda^{\star\prime} - \mathbf{e}_\Lambda^\star\|_2^2 \\ \geq & \ \min(\lambda_1, \lambda_2)(\|\mathbf{r}^{\star\prime} - \mathbf{r}^\star\|_2^2 + \|\mathbf{e}_\Lambda^{\star\prime} - \mathbf{e}_\Lambda^\star\|_2^2). \end{aligned} \tag{4.12}$$

We then show that the function $\tilde{\ell}(\mathbf{z}, L, \cdot, \cdot) - \tilde{\ell}(\mathbf{z}', L', \cdot, \cdot)$ is Lipschitz continuous. To this end, we calculate the difference of the above function:

$$\begin{aligned} & \left(\tilde{\ell}(\mathbf{z}, L, \mathbf{r}, \mathbf{e}) - \tilde{\ell}(\mathbf{z}', L', \mathbf{r}, \mathbf{e})\right) - \left(\tilde{\ell}(\mathbf{z}, L, \mathbf{r}', \mathbf{e}') - \tilde{\ell}(\mathbf{z}', L', \mathbf{r}', \mathbf{e}')\right) \\ = & \ \frac{1}{2}\left(\|\mathbf{z} - L\mathbf{r} - \mathbf{e}\|_2^2 - \|\mathbf{z}' - L'\mathbf{r} - \mathbf{e}\|_2^2\right) - \frac{1}{2}\left(\|\mathbf{z} - L\mathbf{r}' - \mathbf{e}'\|_2^2 - \|\mathbf{z}' - L'\mathbf{r}' - \mathbf{e}'\|_2^2\right) \end{aligned}$$

Define a matrix $A = [L, I]$ and a vector $\mathbf{b} = [\mathbf{r}; \mathbf{e}]$, and we have $L\mathbf{r} + \mathbf{e} = A\mathbf{b}$. Then,

$$\begin{aligned} & \left(\tilde{\ell}(\mathbf{z}, L, \mathbf{r}, \mathbf{e}) - \tilde{\ell}(\mathbf{z}', L', \mathbf{r}, \mathbf{e})\right) - \left(\tilde{\ell}(\mathbf{z}, L, \mathbf{r}', \mathbf{e}') - \tilde{\ell}(\mathbf{z}', L', \mathbf{r}', \mathbf{e}')\right) \\ = & \ \frac{1}{2}\left(\|\mathbf{z} - A\mathbf{b}\|_2^2 - \|\mathbf{z}' - A'\mathbf{b}\|_2^2\right) - \frac{1}{2}\left(\|\mathbf{z} - A\mathbf{b}'\|_2^2 - \|\mathbf{z}' - A'\mathbf{b}'\|_2^2\right) \end{aligned}$$

It is easy to show that the function $\|\mathbf{z} - A\mathbf{b}\|_2^2 - \|\mathbf{z}' - A'\mathbf{b}\|_2^2$ is Lipschitz with constant as $c_1\|A - A'\|_F + c_2\|\mathbf{z} - \mathbf{z}'\|_2$, where c_1, c_2 are constants independent of $A, A', \mathbf{z}, \mathbf{z}'$. Thus,

$$\begin{aligned} & \left(\tilde{\ell}(\mathbf{z}, L, \mathbf{r}, \mathbf{e}) - \tilde{\ell}(\mathbf{z}', L', \mathbf{r}, \mathbf{e})\right) - \left(\tilde{\ell}(\mathbf{z}, L, \mathbf{r}', \mathbf{e}') - \tilde{\ell}(\mathbf{z}', L', \mathbf{r}', \mathbf{e}')\right) \\ \leq & \ \left(c_1\|A - A'\|_F + c_2\|\mathbf{z} - \mathbf{z}'\|_2\right)\|\mathbf{b} - \mathbf{b}'\|_2 \\ = & \ \left(c_1\|L - L'\|_F + c_2\|\mathbf{z} - \mathbf{z}'\|_2\right)\left(\|\mathbf{r} - \mathbf{r}'\|_2 + \|\mathbf{e} - \mathbf{e}'\|_2\right) \end{aligned}$$

According to (3.1) in the supplementary material, and considering $(\mathbf{r}^{*\prime}, \mathbf{e}_\Lambda^{*\prime})$ min-

imizes the loss $\tilde{\ell}(\mathbf{z}', L', \cdot, \cdot)$, we have

$$\begin{aligned}
& \min(\lambda_1, \lambda_2) \left(\|\mathbf{r}^{*\prime} - \mathbf{r}^*\|_2^2 + \|\mathbf{e}_\Lambda^{*\prime} - \mathbf{e}_\Lambda^*\|_2^2 \right) \\
\leq \; & \tilde{\ell}(\mathbf{z}_\Lambda, L_\Lambda, \mathbf{r}^{*\prime}, \mathbf{e}_\Lambda^{*\prime}) - \tilde{\ell}(\mathbf{z}_\Lambda, L_\Lambda, \mathbf{r}^*, \mathbf{e}_\Lambda^*) \\
= \; & \tilde{\ell}(\mathbf{z}_\Lambda, L_\Lambda, \mathbf{r}^{*\prime}, \mathbf{e}_\Lambda^{*\prime}) - \tilde{\ell}(\mathbf{z}'_\Lambda, L'_\Lambda, \mathbf{r}^*, \mathbf{e}_\Lambda^*) + \tilde{\ell}(\mathbf{z}'_\Lambda, L'_\Lambda, \mathbf{r}^*, \mathbf{e}_\Lambda^*) - \tilde{\ell}(\mathbf{z}_\Lambda, L_\Lambda, \mathbf{r}^*, \mathbf{e}_\Lambda^*) \\
\leq \; & \tilde{\ell}(\mathbf{z}_\Lambda, L_\Lambda, \mathbf{r}^{*\prime}, \mathbf{e}_\Lambda^{*\prime}) - \tilde{\ell}(\mathbf{z}'_\Lambda, L'_\Lambda, \mathbf{r}^{*\prime}, \mathbf{e}_\Lambda^{*\prime}) + \tilde{\ell}(\mathbf{z}'_\Lambda, L'_\Lambda, \mathbf{r}^*, \mathbf{e}_\Lambda^*) - \tilde{\ell}(\mathbf{z}_\Lambda, L_\Lambda, \mathbf{r}^*, \mathbf{e}_\Lambda^*) \\
\leq \; & \left(c_1 \|L_\Lambda - L'_\Lambda\|_F + c_2 \|\mathbf{z}_\Lambda - \mathbf{z}'_\Lambda\|_2 \right) \left(\|\mathbf{r}^{*\prime} - \mathbf{r}^*\|_2 + \|\mathbf{e}_\Lambda^{*\prime} - \mathbf{e}_\Lambda^*\|_2 \right).
\end{aligned}$$

Therefore, we have,

$$\left(\|\mathbf{r}^{*\prime} - \mathbf{r}^*\|_2 + \|\mathbf{e}_\Lambda^{*\prime} - \mathbf{e}_\Lambda^*\|_2 \right) \leq \frac{1}{\min(\lambda_1, \lambda_2)} \left(c_1 \|L_\Lambda - L'_\Lambda\|_F + c_2 \|\mathbf{z}_\Lambda - \mathbf{z}'_\Lambda\|_2 \right).$$

Combining the second claim, we can conclude the third claim. □

4.9 Proof of Theorem 12

Proof. We prove the convergence of the sequence $g_t(L_t)$ by showing that the stochastic positive process

$$u_t \triangleq g_t(L_t) \geq 0,$$

is a quasi-martingale [48]. According to Lemma 4, if the sum of the positive difference of u_t is bounded, u_t is a quasi-martingale. And the sum converges almost surely. Thus, we compute the difference of u_t and obtain

$$\begin{aligned}
& u_{t+1} - u_t \\
= \; & g_{t+1}(L_{t+1}) - g_t(L_t) \\
= \; & g_{t+1}(L_{t+1}) - g_{t+1}(L_t) + g_{t+1}(L_t) - g_t(L_t) \\
= \; & g_{t+1}(L_{t+1}) - g_{t+1}(L_t) + \frac{\ell(\mathbf{z}_{t+1}, L_t) - f_t(L_t)}{t+1} + \frac{f_t(L_t) - g_t(L_t)}{t+1}. \quad (4.13)
\end{aligned}$$

Here the third equality is from the fact that $g_{t+1}(L_t) = \frac{1}{t+1}\ell(\mathbf{z}_{t+1}, L_t) + \frac{t}{t+1} g_t(L_t)$. Since L_{t+1} minimizes g_{t+1}, $g_{t+1}(L_{t+1}) - g_{t+1}(L_t) \leq 0$. Since the surrogate g_t up-

perbounds the empirical cost f_t, $g_t \geq f_t$, we also have $f_t(L_t) - g_t(L_t) \leq 0$. Thus we have

$$u_{t+1} - u_t \leq \frac{\ell(\mathbf{z}_{t+1}, L_t) - f_t(L_t)}{t+1}.$$

Since the above inequality is valid for the variation of each pair of u_{t+1} and u_t, in particular it will be valid for the positive variation,

$$[u_{t+1} - u_t]^+ \leq \frac{\ell(\mathbf{z}_{t+1}, L_t) - f_t(L_t)}{t+1},$$

where $[\cdot]^+$ denotes the positive variation.

According to Lemma 4, we consider the expectation of the variation of u_t conditioned on the past information $\mathcal{F}_t = \{\mathbf{z}_1, \ldots, \mathbf{z}_t, L_1, \ldots, L_t, \mathbf{r}_1, \ldots, \mathbf{r}_t, \mathbf{e}_1, \ldots, \mathbf{e}_t\}$ and apply the above inequality,

$$\mathbb{E}[[u_{t+1} - u_t | \mathcal{F}_t]^+] \leq \frac{\mathbb{E}[\ell(\mathbf{z}_{t+1}, L_t)|\mathcal{F}_t] - f_t(L_t)}{t+1} = \frac{f(L_t) - f_t(L_t)}{t+1} \leq \frac{\|f - f_t\|_\infty}{t+1}. \tag{4.14}$$

Here $\|f - f_t\|_\infty = \sup_{f \in F} |f - f_t|$ and $F = \{\ell(\mathbf{z}, L) : \mathcal{Z} \to \mathbb{R}, L \in \mathcal{L}\}$. To bound $\mathbb{E}[\sqrt{t}\|f - f_t\|_\infty]$, here we use the Lemma 5. It is easy to show that in our case, all the hypotheses are verified, namely, $\ell(\mathbf{z}, \cdot)$ is uniformly Lipschitz and bounded (see Lemma 2). Thus $\mathbb{E}_{\mathbf{z}}[\ell(\mathbf{z}, L)^2]$ exists and is uniformly bounded. Therefore, Lemma 5 applies and there exists a constant $\kappa > 0$ such that

$$\mathbb{E}[\sqrt{t}|f - f_t|_\infty] \leq \kappa.$$

Therefore,

$$\mathbb{E}[\mathbb{E}[u_{t+1} - u_t | \mathcal{F}_t]^+] \leq \frac{\kappa}{t^{\frac{3}{2}}}.$$

Therefore, defining δ_t as in Lemma 4:

$$\delta_t = \begin{cases} 1, & \text{if } \mathbb{E}[u_{t+1} - u_t | \mathcal{F}_t] > 0, \\ 0, & \text{otherwise,} \end{cases}$$

we have

$$\sum_{t=1}^{\infty} \mathbb{E}[\delta_t(u_{t+1} - u_t)] = \sum_{t=1}^{\infty} \mathbb{E}[\mathbb{E}[u_{t+1} - u_t|\mathcal{F}_t]^+] \leq \sum_{t=1}^{\infty} \frac{\kappa}{t^{\frac{3}{2}}} \leq +\infty.$$

Thus, we can apply Lemma 4, which proves that $u_t = g_t$ converges almost surely and that

$$\sum_{t=1}^{\infty} |\mathbb{E}[u_{t+1} - u_t|\mathcal{F}_t]| < +\infty \quad a.s.$$

Thus we complete the proof. □

4.10 Proof of Theorem 13

Proof. The Hessian matrix of $g_t(L)$ is $H = I \otimes (A_t + \lambda_1 I)$. Here $\otimes$ denotes the Kronecker production and $A_t = \sum_{i=1}^{t} \mathbf{r}_i \mathbf{r}_i^T$. The smallest eigenvalue of H is equal to the smallest eigenvalue of matrix $(A_t + \lambda_1 I)$, which must be larger than λ_1 since A_t is a semi-definite positive matrix. Thus $g_t(L)$ is strictly convex. And we have,

$$g_t(L_{t+1}) - g_t(L_t) \geq \lambda_1 \|L_{t+1} - L_t\|_F^2. \tag{4.15}$$

Since $g_{t+1}(L_{t+1}) < g_{t+1}(L_t)$ due to L_{t+1} minimizing g_{t+1}, we have

$$g_t(L_{t+1}) - g_t(L_t) \leq g_t(L_{t+1}) - g_{t+1}(L_{t+1}) + g_{t+1}(L_t) - g_t(L_t) = v_t(L_{t+1}) - v_t(L_t).$$

Here we define $v_t(L) \triangleq g_t(L) - g_{t+1}(L)$. And we have,

$$\nabla_L v_t(L) = \nabla_L g_t(L_t) - \nabla_L g_{t+1}(L_t) = \frac{1}{t}(L\tilde{A}_t - B_t) - \frac{1}{t+1}(L\tilde{A}_{t+1} - B_{t+1}).$$

Here $\tilde{A} \triangleq A + \lambda_1 I$ as defined in Algorithm 2. Therefore, by utilizing the triangle inequality and $\|AB\|_F \leq \|A\|_F \|B\|_F$, we can obtain,

$$\begin{aligned} \|\nabla_L v_t(L)\|_F &= \left\| \frac{1}{t} L \left(\tilde{A}_t - \frac{t}{t+1} \tilde{A}_{t+1} \right) - \frac{1}{t} \left(B_t - \frac{t}{t+1} B_{t+1} \right) \right\|_F \\ &\leq \frac{1}{t} \left(\|L\|_F \left\| \tilde{A}_t - \frac{t\tilde{A}_{t+1}}{t+1} \right\|_F + \left\| B_t - \frac{tB_{t+1}}{t+1} \right\|_F \right). \end{aligned}$$

Since the basis L is usually bounded $\|L\|_F < \kappa_1$ (Assumption 1), the function $v_t(L)$ is Lipschitz with constant $c_t = \frac{1}{t} \left(\kappa_1 \|\tilde{A}_t - \frac{t\tilde{A}_{t+1}}{t+1}\|_F + \|B_t - \frac{tB_{t+1}}{t+1}\|_F \right)$. Thus, we have

$$g_t(L_{t+1}) - g_t(L_t) \leq v_t(L_{t+1}) - v_t(L_t) \leq c_t \|L_{t+1} - L_t\|_F.$$

Substituting into (4.15), we can then obtain that

$$\|L_{t+1} - L_t\|_F \leq \frac{c_t}{\lambda_1}.$$

Since $c_t = O(1/t)$, we have $\|L_{t+1} - L_t\|_F = O(1/t)$. □

4.11 Proof of Theorem 14

Proof. From (4.13), we can obtain that

$$\frac{g_t(L_t) - f_t(L_t)}{t+1} \leq \frac{\ell(\mathbf{z}_{t+1}, L_t) - f_t(L_t)}{t+1} - (g_{t+1} - g_t) \leq \frac{\ell(\mathbf{z}_{t+1}, L_t) - f_t(L_t)}{t+1} + [g_{t+1} - g_t]^-$$

Taking the conditional expectation on the filtration $\mathcal{F}_t$ as in the proof of Theorem 2, we obtain

$$\mathbb{E}\left[\frac{g_t(L_t) - f_t(L_t)}{t+1} | \mathcal{F}_t \right] = \frac{g_t(L_t) - f_t(L_t)}{t+1} \leq \mathbb{E}\left[\frac{\ell(\mathbf{z}_{t+1}, L_t) - f_t(L_t)}{t+1} | \mathcal{F}_t \right] + \mathbb{E}[[g_{t+1} - g_t]^- | \mathcal{F}_t].$$

$$
\begin{aligned}
& \sum_{t=1}^{\infty} \frac{g_t(L_t) - f_t(L_t)}{t+1} \\
\leq & \sum_{t=1}^{\infty} \mathbb{E}\left[\frac{\ell(\mathbf{z}_{t+1}, L_t) - f_t(L_t)}{t+1} | \mathcal{F}_t\right] + \sum_{t=1}^{\infty} \mathbb{E}[[g_{t+1} - g_t]^- | \mathcal{F}_t] \\
\leq & \sum_{t=1}^{\infty} \frac{|f - f_t|}{t+1} + \sum_{t=1}^{\infty} \mathbb{E}[[g_{t+1} - g_t]^- | \mathcal{F}_t].
\end{aligned}
$$

Here $[\cdot]^-$ means taking negative part. The second inequality is from (4.14). According to Theorem 2, the function g_t converges almost surely. And we have

$$
\sum_{t=1}^{\infty} |\mathbb{E}[[g_{t+1} - g_t]^+] | \mathcal{F}_t| < +\infty \quad a.s.
$$

By symmetry we can also obtain similarly

$$
\sum_{t=1}^{\infty} |\mathbb{E}[[g_{t+1} - g_t]^-] | \mathcal{F}_t| < +\infty \quad a.s.
$$

According to central limit theorem, we have $\sqrt{t}|f - f_t|$ converges almost surely when $t \to \infty$. Therefore $\sum_{t=1}^{\infty} \frac{|f-f_t|}{t+1}$ converges almost surely. Then we obtain the almost sure convergence of the positive sum

$$
\sum_{t=1}^{\infty} \frac{g_t(L_t) - f_t(L_t)}{t+1} \leq \sum_{t=1}^{\infty} \frac{|f - f_t|}{t+1} + \sum_{t=1}^{\infty} |\mathbb{E}[u_{t+1} - u_t | \mathcal{F}_t]| \leq \infty.
$$

Since both g_t and f_t are Lipschitz continuous, there exists a constant $\kappa' > 0$ such that

$$
|g_{t+1}(L_{t+1}) - f_{t+1}(L_{t+1}) - (g_t(L_t) - f_t(L_t))| \leq \kappa' \|L_{t+1} - L_t\|_F.
$$

According to Theorem 3, $\|L_{t+1} - L_t\|_F = O(1/t)$. Thus it is easy to verify that the hypotheses of Lemma 6 are satisfied. Therefore,

$$
g_t(L_t) - f_t(L_t) \underset{t \to +\infty}{\longrightarrow} 0 \quad a.s.
$$

Since $g_t(L_t)$ converges almost surely, this shows that $f_t(L_t)$ converges almost surely to the same limit. Note that we have in addition $\|f_t - f\|_\infty \underset{t\to+\infty}{\longrightarrow} 0$ a.s. Therefore,

$$g_t(L_t) - f(L_t) \underset{t\to+\infty}{\longrightarrow} 0 \quad a.s.$$

and $f(L_t)$ converges almost surely. □

4.12 Proof of Theorem 15

Proof. Since the function g_t converges almost surely (see Theorem 2), $g_t = \mathrm{Tr}(L^T L\tilde{A}_t/t) - \mathrm{Tr}(L^T B_t/t)$, thus the sequences of matrices $\tilde{A}_t/t, B_t/t$ are bounded. It is possible to extract converging subsequences. Let us assume for a moment that these sequences converge respectively to two matrices A_∞ and B_∞. In that case, L_t converges to a matrix L_∞. Let U be a matrix in $\mathbb{R}^{p\times r}$. Since g_t upperbounds f_t on $\mathbb{R}^{p\times r}$, for all t,

$$g_t(L_t + U) \geq f_t(L_t + U).$$

Taking the limit when t tends to infinity,

$$g_\infty(L_\infty + U) \geq f(L_\infty + U).$$

Let $h_t > 0$ be a sequence that converges to 0. Using a first order Taylor expansion, and using the fact that ∇f is Lipschitz (see Lemma 3) and $g_\infty(L_\infty) = f(L_\infty)$ a.s. (see Theorem 4), we have

$$f(L_\infty) + \mathrm{Tr}(h_t L^T \nabla g_\infty(L_\infty)) + o(h_t L) \geq f(L_\infty) + \mathrm{Tr}(h_t L^T \nabla f(L_\infty)) + o(h_t L),$$

and it follows that

$$\mathrm{Tr}\left(\frac{1}{\|L\|_F} L^T \nabla g_\infty(L_\infty)\right) \geq \mathrm{Tr}\left(\frac{1}{\|L\|_F} L^T \nabla f(L_\infty)\right).$$

Since the above inequality is true for all L, we have $\nabla g_\infty(L_\infty) = \nabla f(L_\infty)$. Since the first-order necessary condition for L_∞ being an optimum of g_∞ is that $\nabla g_\infty = 0$. Thus at L_∞, we have $\nabla f(L_\infty) = 0$. Namely, the first-order optimum condition for f at L_∞ is also verified. □

4.13 Proof of Theorem 16

Proof. The minimization of the objective function in (6),

$$\min_L \lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} \ell(\mathbf{z}_i, L)$$

is equivalent to

$$\min_{L,R,E} \frac{1}{2}\|Z - LR^T - E\|_F^2 + \frac{\lambda_1}{2}\left(\|L\|_F^2 + \|R\|_F^2\right) + \lambda_2\|E\|. \tag{4.16}$$

Here $Z = [\mathbf{z}_1, \ldots, \mathbf{z}_n], R = [\mathbf{r}_1^T; \ldots; \mathbf{r}_n^T]$ and $E = [\mathbf{e}_1, \ldots, \mathbf{e}_n]$.

When the first order optimal condition is satisfied, we have

$$(LR^T - \tilde{Z})R + \lambda_1 L = 0, \tag{4.17}$$

$$(RL^T - \tilde{Z}^T)L + \lambda_1 R = 0, \tag{4.18}$$

$$LR^T - \tilde{Z} \in \lambda_2 \partial\|E\|_1. \tag{4.19}$$

Here $\tilde{Z} \triangleq Z - E$. Note that for any invertible matrix Q, the pair (LQ, RQ^{-1}) provides a factorization equivalent to (L, R). In particular, any solution (L, R) can be orthogonalized to a (non-unique) equivalent orthogonal solution $\bar{L} = LQ, \bar{R} = RQ^{-1}$ such that $\bar{R}^T\bar{R} = \Lambda_R$ and $\bar{L}^T\bar{L} = \Lambda_L$ are diagonal matrices [**?**]. Substituting $\bar{R}^T\bar{R} = \Lambda_R$ and $\bar{L}^T\bar{L} = \Lambda_L$ into (4.17) and (4.18), we can obtain that $\Lambda_L = \Lambda_R = \Lambda$.

Since we can always perform the orthgonalization operation on the obtained solution L and R, we focus on an orthogonal solution, where $R^TR = \Lambda \in \mathbb{R}^{r\times r}$ and $L^TL = \Lambda \in \mathbb{R}^{r\times r}$. Since L and R are full rank, the elements in the diagonal of matrix Λ are non-zero.

From (4.17) we can obtain

$$L = \tilde{Z}R(R^TR + \lambda_1 I)^{-1} = \tilde{Z}R(\Lambda + \lambda_1 I)^{-1}. \tag{4.20}$$

Substituting back into (4.18), we have

$$R\Lambda - \tilde{Z}^T L + \lambda_1 R = 0.$$

Namely,

$$R\Lambda - \tilde{Z}^T\tilde{Z}R(\Lambda + \lambda_1 I)^{-1} + \lambda_1 R = 0,$$

$$R(\Lambda + \lambda_1 I)^2 = \tilde{Z}^T\tilde{Z}R. \tag{4.21}$$

Define $R' \triangleq R(\sqrt{\Lambda})^{-1}$, then we have $R'^TR' = (\sqrt{\Lambda})^{-1}R^TR(\sqrt{\Lambda})^{-1} = I$. Namely, the matrix R' is an orthogonal matrix. From the above equation, we conclude that

$$R'\sqrt{\Lambda}(\Lambda + \lambda_1 I)^2 = \tilde{Z}^T\tilde{Z}R'\sqrt{\Lambda}.$$

$$R'(\Lambda + \lambda_1 I)^2 = \tilde{Z}^T\tilde{Z}R'.$$

Therefore, the columns of the matrix R' are the eigenvectors of the matrix $\tilde{Z}^T\tilde{Z}$. Thus the columns of the matrix R are the eigenvectors of the matrix $\tilde{Z}^T\tilde{Z}$ scaled by the square root of the matrix Λ. And the eigenvalues of the matrix $\tilde{Z}^T\tilde{Z}$ are the elements in the diagonal of matrix $(\Lambda + \lambda_1 I)^2$.

From (4.20) we have

$$\tilde{Z}\tilde{Z}^TL = \tilde{Z}\tilde{Z}^T\tilde{Z}R(\Lambda + \lambda_1 I)^{-1} \overset{(4.21)}{=} \tilde{Z}R(\Lambda + \lambda_1 I) \overset{(4.20)}{=} L(\Lambda + \lambda_1 I)^2.$$

Thus similar to R, the columns of matrix L correspond to the eigenvectors of the matrix $\tilde{Z}\tilde{Z}^T$ scaled by the square root of the matrix Λ.

Performing SVD on the matrix $\tilde{Z}$ provides $\tilde{Z} = U\Sigma V^T = U_1\Sigma_1 V_1^T + U_2\Sigma_2 V_2^T$. Here $U_1^TU_2 = 0$, $V_1^TV_2 = 0$ and $\Sigma_1 \in \mathbb{R}^{k\times k}$, $\Sigma_2 \in \mathbb{R}^{(n-k)\times(n-k)}$.

From the above results, we can obtain $L = U_1\sqrt{\Lambda}$ and $R = V_1\sqrt{\Lambda}$.

$$\tilde{Z}^T\tilde{Z} = V\Sigma^2V^T.$$

Thus

$$\Sigma_1 = \Lambda + \lambda_1 I.$$

Since the matrix L is full rank, $L^TL = \Lambda$ is positive definite. Thus $\Sigma_1 \succ \lambda_1 I$.

The obtained solution $X = LR^T = U_1\Lambda V_1^T = U_1(\Sigma_1 - \lambda_1 I)V_1^T$. We can obtain that

$$\tilde{Z} - X = U\Sigma V^T - U_1(\Sigma_1 - \lambda_1 I)V_1^T = \lambda_1 U_1V_1^T + U_2\Sigma_2V_2^T = \lambda_1(U_1V_1^T + W),$$

where $W = U_2\Sigma_2V_2^T/\lambda_1$.

Thus, it is easy to verify that

$$\tilde{Z} - X = Z - E - X \in \partial\lambda_1\|X\|_* = \{\lambda_1(U_1V_1^T + W)|U_1^TW = 0, WV_1 = 0, \|W\|_2 \leq 1\}. \tag{4.22}$$

Note that the problem in (4.16) is equivalent to the following *convex* optimization problem,

$$\min_{X,E} \frac{1}{2}\|Z - X - E\|_F^2 + \lambda_1\|X\|_* + \lambda_2\|E\|_1.$$

The first-order optimal condition is satisfied by the obtained solution as shown in (4.22) and (4.19). Since the optimization problem is convex, we can conclude that the solution is also global optimal.

□

4.14 Technical Lemmas

Lemma 14 (Corollary of Theorem 4.1 from [130])**.** *Let $f : \mathbb{R}^p \times \mathbb{R}^q \to \mathbb{R}$. Suppose that for all $\mathbf{x} \in \mathbb{R}^p$ the function $f(\mathbf{x}, \cdot)$ is differentiable, and that f and $\nabla_{\mathbf{u}} f(\mathbf{x}, \mathbf{u})$ the derivative of $f(\mathbf{x}, \cdot)$ are continuous on $\mathbb{R}^q \to \mathbb{R}$. Let $\nu(\mathbf{u})$ be the optimal value*

function $\nu(\mathbf{u}) = \min_{\mathbf{x}\in C} f(\mathbf{x},\mathbf{u})$*, where* C *is a compact subset of* $\mathbb{R}^p$*. Then* $\nu(\mathbf{u})$ *is directionally differentiable. Furthermore, if for* $\mathbf{u}_0 \in \mathbb{R}^q$*,* $f(\cdot,\mathbf{u}_0)$ *has a unique minimizer* $\mathbf{x}_0$ *then* $\nu(\mathbf{u})$ *is differentiable in* $\mathbf{u}_0$ *and* $\nabla_{\mathbf{u}}\nu(\mathbf{u}_0) = \nabla_{\mathbf{u}} f(\mathbf{x}_0,\mathbf{u}_0)$*.*

Lemma 15 (Sufficient condition of convergence for a stochastic process, [48])**.** *Let* $(\Omega,\mathcal{F},P)$ *be a measurable probability space,* u_t*, for* $t \geq 0$*, be the realization of a stochastic process and* $\mathcal{F}_t$ *be the filtration determined by the past information at time* t*. Let*

$$\delta_t = \begin{cases} 1 & \text{if } \mathbb{E}[u_{t+1} - u_t|\mathcal{F}_t] > 0, \\ 0 & \text{otherwise.} \end{cases}$$

If for all t*,* $u_t \geq 0$ *and* $\sum_{t=1}^{\infty} \mathbb{E}[\delta_t(u_{t+1} - u_t)] < \infty$*, then* u_t *is a quasi-martingale and converges almost surely. Moreover,*

$$\sum_{t=1}^{\infty} |\mathbb{E}[u_{t+1} - u_t|\mathcal{F}_t]| < +\infty \quad a.s.$$

Lemma 16 ([128])**.** *Let* $F = f_\theta : \chi \to \mathbb{R}, \theta \in \Theta$ *be a set of measurable functions indexed by a bounded subset* Θ *of* $\mathbb{R}^d$*. Suppose that there exists a constant* K *such that*

$$|f_{\theta_1}(x) - f_{\theta_2}(x)| \leq K\|\theta_1 - \theta_2\|_2,$$

for every θ_1 *and* θ_2 *in* Θ *and* x *in* χ*. Then,* F *is P-Donsker. For any* f *in* F*, let us define* $\mathbb{P}_n f$*,* $\mathbb{P}f$ *and* $\mathbb{G}_n f$ *as*

$$\mathbb{P}_n f = \frac{1}{n}\sum_{i=1}^{n} f(X_i), \quad \mathbb{P}_f = \mathbb{E}_X[f(X)],$$
$$\mathbb{G}_n f = \sqrt{n}(\mathbb{P}_n f - \mathbb{P}f).$$

Let us also suppose that for all f*,* $\mathbb{P}f^2 < \delta^2$ *and* $\|f\|_\infty < M$ *and that the random elements* $X_1, X_2, \ldots$ *are Borel-measurable. Then, we have*

$$\mathbb{E}_P\|\mathbb{G}_n\|_F = O(1),$$

where $\|\mathbb{G}_n\|_F = \sup_{f \in F} |\mathbb{G}_n f|$.

Lemma 17 (Positive converging sums, [42])**.** *Let* a_n,b_n *be two real sequences such that for all* n, $a_n \geq 0$, $b_n \geq 0$, $\sum_{n=1}^{\infty} a_n = \infty$, $\sum_{n=1}^{\infty} a_n b_n < \infty$, $\exists K > 0$ *s.t.* $|b_{n+1} - b_n| < K a_n$. *Then,* $\lim_{n \to \infty} b_n = 0$.

4.15 Simulations

4.15.1 Medium-scale Robust PCA

We here evaluate the ability of the proposed OR-PCA of correctly recovering the subspace of corrupted observations, under various settings of the intrinsic subspace dimension and error density. In particular, we adopt the batch robust PCA method, Principal Component Pursuit [44], as the batch counterpart of the proposed OR-PCA method for reference. PCP estimates the subspace in a batch manner through solving the problem in (4.1) and outputs the low-rank data matrix. For fair comparison, we follow the data generation scheme of PCP as in [44]: we generate a set of n clean data points as a product of $X = UV^T$, where the sizes of U and V are $p \times r$ and $n \times r$ respectively. The elements of both U and V are i.i.d. sampled from the $\mathcal{N}(0, 1/n)$ distribution. Here U is the basis of the subspace and the intrinsic dimension of the subspace spanned by U is r. The observations are generated through $Z = X + E$, where E is a sparse matrix with a fraction of ρ_s non-zero elements. The elements in E are from a uniform distribution over the interval of $[-1000, 1000]$. Namely, the matrix E contains gross but sparse errors.

We run the OR-PCA and the PCP algorithms 10 times under the following settings: the ambient dimension and number of samples are set as $p = 400$ and $n = 1,000$; the intrinsic rank r of the subspace varies from 4 to 200; the value of error fraction, ρ_s, varies from very sparse 0.01 to relatively dense 0.5. The trade-off parameters of OR-PCA are fixed as $\lambda_1 = \lambda_2 = 1/\sqrt{p}$. The performance is evaluated by the similarity between the subspace obtained from the algorithms and the groundtruth. In particular, the similarity is measured by the Expressed Variance

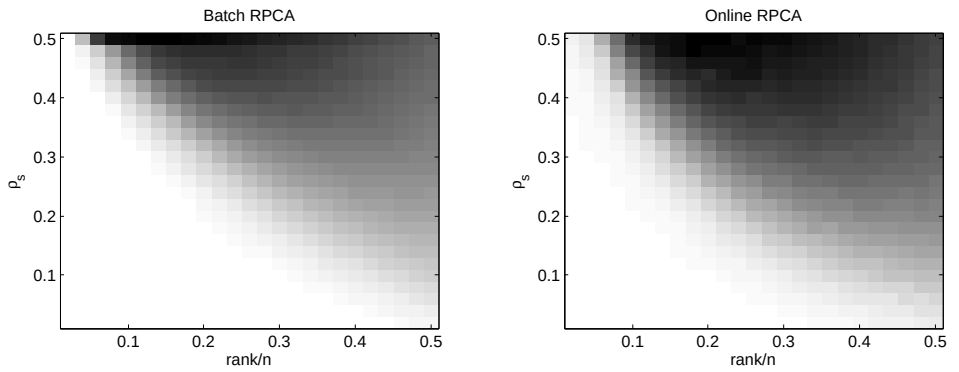
(E.V.):

$$\text{E.V.}(U, L) \triangleq \frac{\text{Tr}(L^T U U^T L)}{\text{Tr}(U U^T)},$$

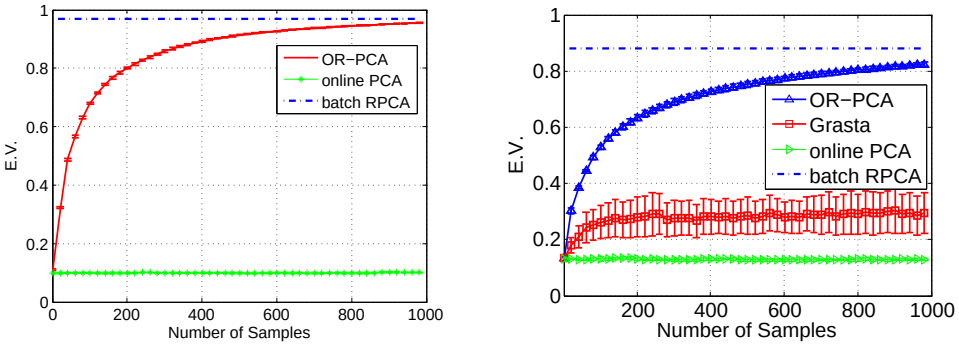
where L is from orthogonalizing the output of the OR-PCA or SVD on the output of PCP, and U is the basis of the recovered subspace. A larger value of E.V. means better subspace recovery.

We plot the averaged E.V. values of PCP and OR-PCA under different settings in a matrix form, as shown in Figure 4.1(a) and Figure 4.1(b) respectively. The results demonstrate that under relatively low intrinsic dimension (small rank/n) and sparse corruption (small ρ_s), OR-PCA is able to recover the subspace nearly perfectly (E.V.= 1). We also observe that the performance of OR-PCA is close to that of the PCP. This demonstrates that the proposed OR-PCA method achieves comparable performance with the batch method and verifies our convergence guarantee on the OR-PCA. In the relatively difficult setting (high intrinsic dimension and dense error, shown in the top-right of the matrix), OR-PCA performs slightly worse than the PCP, possibly because the number of streaming samples is not enough to achieve convergence.

To better demonstrate the robustness of OR-PCA to corruptions and illustrate how the performance of OR-PCA is improved when more samples are revealed, we plot the performance curve of OR-PCA against the number of samples in Figure 4.1(c), under the setting of $p = 400$, $n = 1,000$, $\rho_s = 0.1$, $r = 80$, and the results are averaged from 10 repetitions. We also apply GRASTA [50] to solve this RPCA problem in an online fashion as a baseline. The parameters of GRASTA are set as the values provided in the implementation package provided by the authors. We observe that when more samples are revealed, both OR-PCA and GRASTA steadily improve the subspace recovery. However, our proposed OR-PCA converges much faster than GRASTA, possibly because in each iteration OR-PCA obtains the optimal closed-form solution to the basis updating subproblem while GRASTA only takes one gradient descent step. Observe from the figure that after 200 samples are revealed, the performance of OR-PCA is already satisfactory (E.V.$>$ 0.8). How-



(a) Batch RPCA (b) OR-PCA



(c) $\rho_s = 0.1$ (d) $\rho_s = 0.3$

Figure 4.1: (a) and (b): subspace recovery performance under different corruption fraction ρ_s (vertical axis) and rank/n (horizontal axis). Brighter color means better performance; (c) and (d): the performance comparison of the OR-PCA, Grasta, and online PCA methods against the number of revealed samples under two different corruption levels ρ_s with PCP as reference.

ever, for GRASTA, it needs about 400 samples to achieve the same performance. To show the robustness of the proposed OR-PCA, we also plot the performance of the standard online (or incremental) PCA [41] for comparison. This work focuses on developing online *robust* PCA. The non-robustness of (online) PCA is independent of used optimization method. Thus, we only compare with the basic online PCA method [41], which is enough for comparing robustness. The comparison results are given in Figure 4.1(c). We observe that as expected, the online PCA cannot recover the subspace correctly (E.V.$\approx$ 0.1), since standard PCA is fragile to gross corruptions. We then increase the corruption level to $\rho_s = 0.3$, and plot the performance curve of the above methods in Figure 4.1(d). From the plot, it can be observed that the performance of GRASTA decreases severely (E.V.$\approx$ 0.3) while

OR-PCA still achieves E.V. ≈ 0.8. The performance of PCP is around 0.88. This result clearly demonstrates the robustness advantage of OR-PCA over GRASTA. In fact, from other simulation results under different settings of intrinsic rank and corruption level (see supplementary material), we observe that the GRASTA breaks down at 25% corruption (the value of E.V. is zero). However, OR-PCA achieves a performance of E.V.≈ 0.5, even in presence of 50% outlier corruption.

4.15.2 Large-scale Robust PCA

We now investigate the computational efficiency of OR-PCA and the performance for large scale data. The samples are generated following the same model as explained in the above subsection. The results are provided in Table 4.1. All of the experiments are implemented in a PC with 2.83GHz Quad CPU and 8GB RAM. Note that batch RPCA cannot process these data due to out of memory.

Table 4.1: The comparison of OR-PCA and GRASTA under different settings of sample size (n) and ambient dimensions (p). Here $\rho_s = 0.3, r = 0.1p$. The corresponding computational time (in $\times 10^3$ seconds) is shown in the top row and the E.V. values are shown in the bottom row correspondingly. The results are based on the average of 5 repetitions and the variance is shown in the parentheses.

p	1×10^3			1×10^4	
n	1×10^6	1×10^8	1×10^{10}	1×10^6	1×10^8
OR-PCA	0.013(0.0004)	1.312(0.082)	139.233(7.747)	0.633(0.047)	15.910(2.646)
	0.99(0.01)	0.99(0.00)	0.99(0.00)	0.82(0.09)	0.82(0.01)
GRASTA	0.023(0.0008)	2.137(0.016)	240.271(7.564)	2.514(0.011)	252.630(2.096)
	0.54(0.08)	0.55(0.02)	0.57(0.03)	0.45(0.02)	0.46(0.03)

From the above results, we observe that OR-PCA is much more efficient and performs better than GRASTA. In fact, the computational time of OR-PCA is linear in the sample size and nearly linear in the ambient dimension. When the ambient dimension is large ($p = 1 \times 10^4$), OR-PCA is more efficient than GRASTA with an order magnitude efficiency enhancement. We then compare OR-PCA with batch PCP. In each iteration, batch PCP needs to perform an SVD plus a thresholding operation, whose complexity is $O(np^2)$. In contrast, for OR-PCA, in each iteration, the computational cost is $O(pr^2)$, which is independent of the sample size and linear in the ambient dimension. To see this, note that in step 2) of Algorithm 3, the

computation complexity is $O(r^2 + pr + r^3)$. Here $O(r^3)$ is for computing $L^T L$. The complexity of step 3) is $O(r^2 + pr)$. For step 4) (*i.e.*, Algorithm 4), the cost is $O(pr^2)$ (updating each column of L requires $O(pr)$ and there are r columns in total). Thus the total complexity is $O(r^2 + pr + r^3 + pr^2)$. Since $p \gg r$, the overall complexity is $O(pr^2)$.

The memory cost is significantly reduced too. The memory required for OR-PCA is $O(pr)$, which is independent of the sample size. This is much smaller than the memory cost of the batch PCP algorithm ($O(pn)$), where $n \gg p$ for large scale dataset. This is quite important for processing *big data*. The proposed OR-PCA algorithm can be easily parallelized to further enhance its efficiency.

4.15.3 Robust Subspace Tracking

Besides identifying a static subspace, OR-PCA is also able to solve the subspace tracking problem [129], where the underlying subspace of the observations is time variant. In practice, several important problems can be abstracted as the subspace tracking problem, such as video surveillance with moving cameras, network monitoring. In this subsection, we investigate the performance of online RPCA for tracking the dynamic subspace which is rotated gradually, and compare its performance with the batch RPCA method. In particular, we rotate an initial subspace basis $U_0 \in \mathbb{R}^{p \times r}$ along with the time instance t through $U_t = e^{\delta t B} U_0$. Here B is a randomly generated skew-symmetric matrix[1] and δ is a parameter to control the rotation degree at each time instant. We generate one observed sample based on each basis U_t, following the data generation scheme as in the above subsection. The set of generated corrupted samples $\{\mathbf{z}_1, \ldots, \mathbf{z}_n\}$ forms the streaming samples, which are from different subspaces. In this case, the batch RPCA method will fail since it treats all the samples as from the same subspace. However, the proposed OR-PCA continuously updates the subspace estimation according to each revealed sample. Therefore, it is able to track the rotating subspace. In the simulations, we generate

[1]We use the MATLAB built-in function *skewdec* to generate the matrix B, and then normalize its elements to less than 1, *i.e.*, $B = B/\|B\|_\infty$.

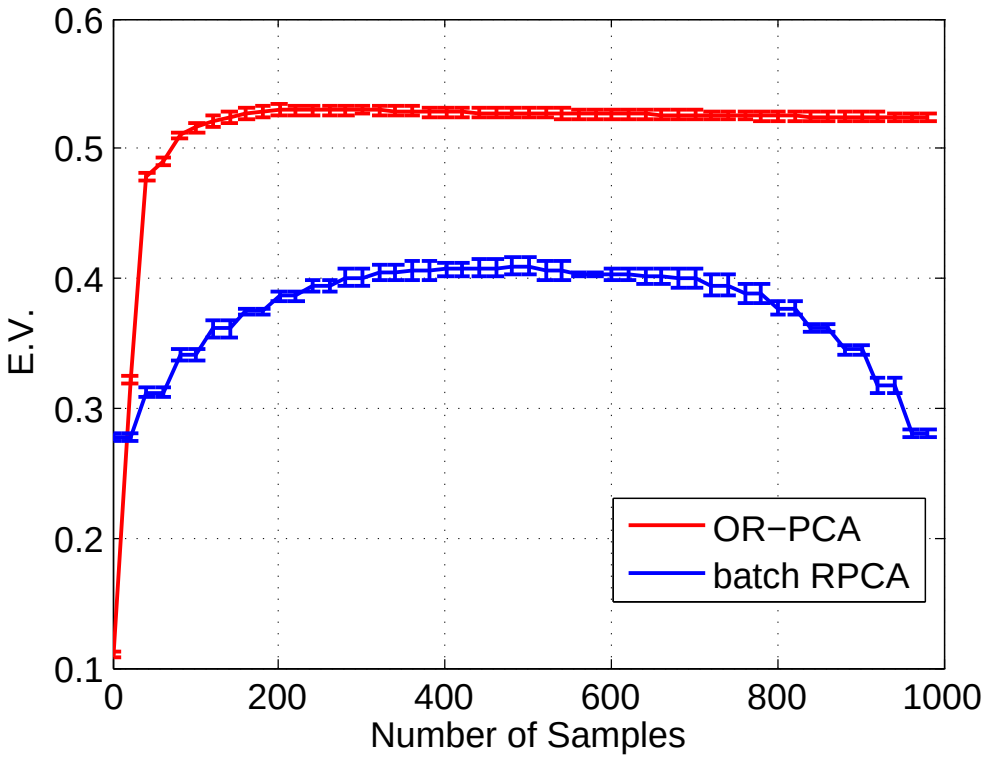


Figure 4.2: The performance comparison of the online RPCA (blue line) on rotating subspaces with the batch RPCA (red lines) method. The underlying subspace is rotated with the parameter $\delta = 1$.

$n = 1,000$ samples with $p = 400$, under the setting of the rank $r = 40$ and outlier fraction $\rho_s = 0.1$. We implement the Principal Component Pursuit over all the $1,000$ samples as the baseline, *i.e.*, batch RPCA. Both the OR-PCA and the batch RPCA are implemented 10 times under each each setting and the average E.V. and the variance are reported. Smaller δ means the subspaces change more slowly. The subspace recovery performance is also measured by E.V. as aforementioned. Note that the groundtruth subspace is different at different time instance.

We first compare the subspace tracking performance of OR-PCA with batch RPCA under the setting of $\delta = 1$, namely the subspace changes relatively fast. Their performance curves against the number of samples are plotted in Figure 4.2. From the results, we can make the following observations: (1) For the first 40 samples, the performance of OR-PCA increases very fast, from the initial E.V. of 0.1 to 0.5. This is because the initial samples are from similar subspace and can help improve

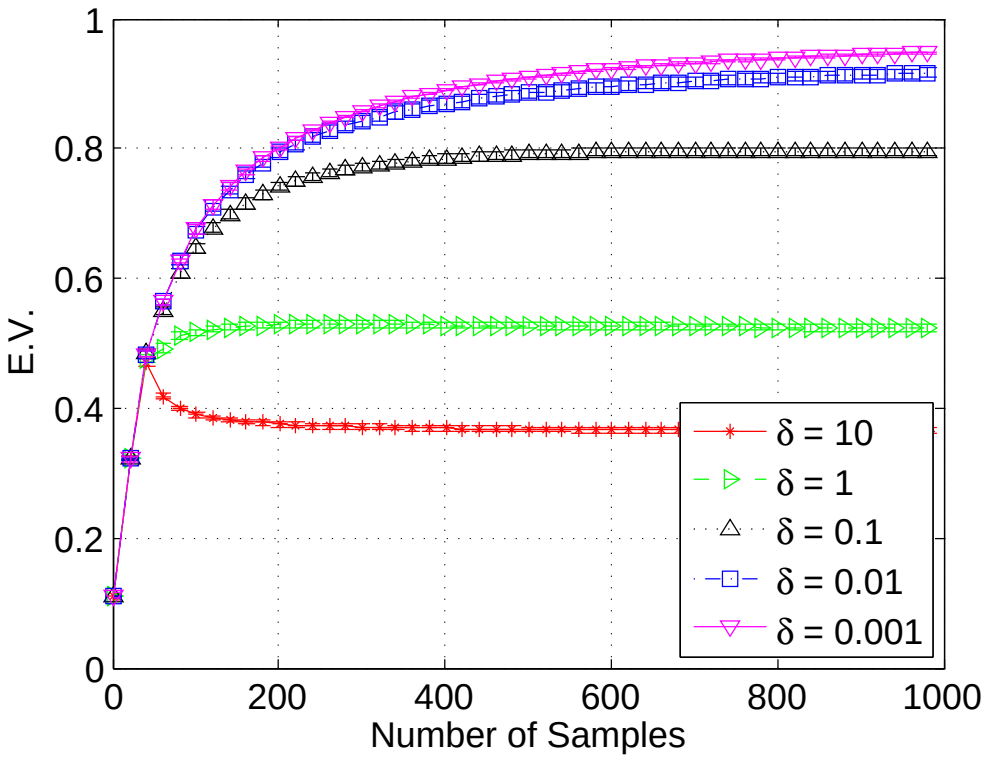


Figure 4.3: The performance of the OR-PCA on tracking rotating subspaces under different values of the changing speed parameter δ.

the subspace estimation well. Then OR-PCA enters a stable state of tracking the subspace and its performance converges to about 0.55. (2) For the batch RPCA method, due to the subspace is changing, its performance is not stable. For the first 400 samples, the performance keeps increasing. But after that, its performance breaks down soon. (3) Generally speaking, OR-PCA outperforms the batch RPCA with a performance margin of 10% under the current setting.

Intuitively, the performance of the subspace tracking methods is affected by the speed of the subspace changing. To investigate the ability of OR-PCA to track subspace with different changing speed, we conduct the experiments under the different values of the parameter $\delta = \{0.001, 0.01, 0.1, 1, 10\}$. The performance curves are shown in Figure 4.3. From the results, we can observe that the more slowly subspace rotates, the better OR-PCA performs for tracking. When the changing speed increases, *e.g.,* $\delta = 10$, the performance will drop after achieving the best performance. And finally OR-PCA converges to a relatively low performance.

4.16 Chapter Summary

In this chapter, we develop an online robust PCA (OR-PCA) method. Different from previous batch based methods, the OR-PCA need not "remember" all the past samples and achieves much higher storage efficiency. The OR-PCA objective function is formulated by decomposing the nuclear norm to an explicit product of two low-rank matrices and an stochastic fashion optimization algorithm is adopted to solve it. We provide the convergence analysis of the OR-PCA method and show that OR-PCA approximates the solution of batch RPCA asymptotically. Comprehensive simulations clearly demonstrate the outperforming ability of OR-PCA on both subspace recovering and tracking.

Chapter 5

Geometric ℓ_p-norm Feature Pooling for Image Classification

From this chapter, we introduce some applications of low-dimensional structure learning in computer vision field. In particular, we focus on the problem of image annotation and object recognition.

In this chapter, we propose to learn and utilize the low-dimensional structure in the object class space, which will embed discriminative information into the generated image representation, and boost the classification performance of a realistic system significantly.

5.1 Introduction

With the prevalence of the *Bag-of-(Visual)-Words* (BoW) model [78] for visual recognition, feature pooling has become a common practice for image/video feature representation and encoding. For a typical image classification task, local image features are first extracted and quantized according to a visual dictionary. Then, the quantization indices of all the local features are summarized to form the global feature representation. A most common summarization method is to form the histogram, *i.e.* to sum up all the occurrences of each index throughout the entire image in an orderless manner. From the viewpoint of feature pooling [67, 75], his-

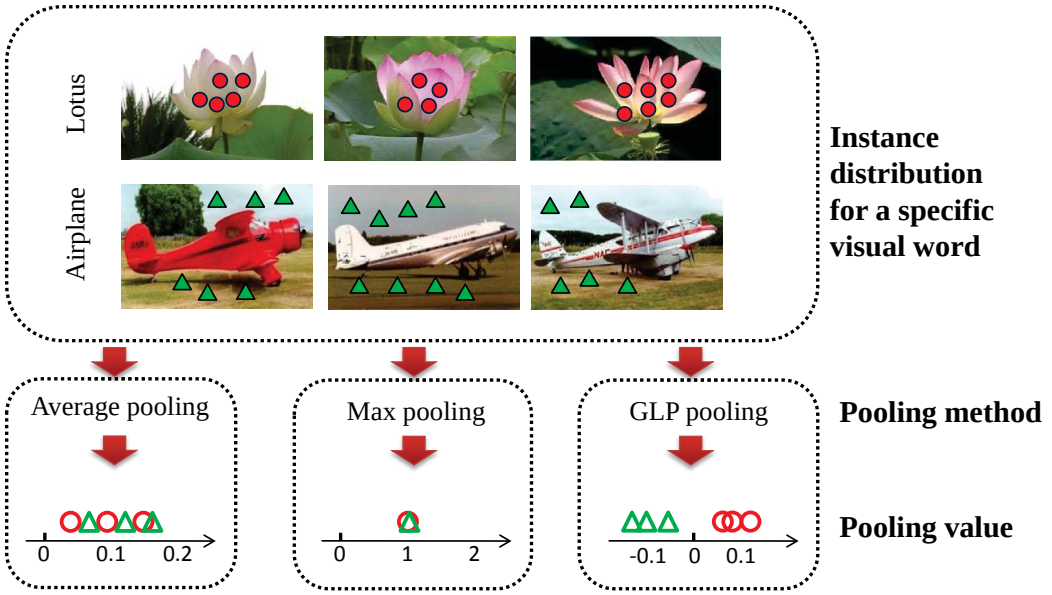


Figure 5.1: Illustration on the importance of the visual word spatial distribution for image classification purposes. In the top block, the distributions of a specific visual word in two classes are indicated by circles and triangles respectively. In the bottom blocks, circles and triangles represent the pooled statistic values of the two classes. By utilizing the class-specific local feature spatial distributions, Geometric ℓ_p-norm Pooling can generate more separable pooled values, compared with the average and max pooling.

togram representation is equivalent to *average pooling*. Despite its conceivable ease and compactness, average pooling is not immune to local feature noise. To overcome this inherent limitation, Ranzato *et al.* [83] proposed a pooling method called *max pooling*. Instead of performing averaging operation, max pooling adopts the element-wise maximum values of feature vectors over the whole image or the region of interest as the pooled features. Max pooling has shown to be more robust against local feature noise and can achieve much better classification performance [87].

However, sacrificing the rich information about spatial distribution of available features as in these two pooling methods is not always worthwhile in practice. In the image classification task, the objects/regions in the images are often well roughly aligned, *e.g.* scenery images, or can be automatically aligned via saliency and symmetry detection. Therefore, image features do possess class-specific discriminative geometric information (*i.e.* spatial distribution patterns). The simple assumption associated with average or max pooling, that the spatial distribution for each visual

feature is uniform across different classes, causes severe information loss. More seriously, such loss is irreversible and the lost information cannot be recovered in the subsequent steps once abandoned. Figure 5.1 illustrates such an issue for the average and max pooling methods. For images from a specific class, their visual features indexed by the same visual word often share similar spatial distribution. Besides, such class-specific spatial distributions are quite distinct from each other and encode discriminative information. However, as shown in this figure, neither average nor max pooling can capture the underlying difference and produce discriminative features due to the loss of the spatial information in the pooling process.

Moreover, these two deterministic pooling methods either treat all the local features uniformly or only select the most salient one, and they both assume local features are distributed independently. By comparison, an optimal pooling scheme is expected to be more flexible and able to capture the spatial correlation of features.

Motivated by the above considerations, we propose a so-called *geometric ℓ_p-norm pooling* method. Overall, the proposed method aims to learn a pooling function that implicitly encodes the class-specific geometric information of feature distribution in the form of weighted norm. This function is optimized towards best class separability, with regularization that encodes prior knowledge about correlation of local features. This geometric ℓ_p-norm pooling method possesses the following advantages:

- As the pooling function is learned by directly maximizing the class separability, it is designed to bear good discriminating capability.
- The pooling function exactly corresponds to the class specific spatial pattern of each visual word, thus the spatial distribution information of visual words is utilized to a satisfactory extent.
- It models the correlations among local features and makes a more reasonable assumption about feature distribution. Also it can naturally unify the average and max pooling in a more flexible framework.

The remainder of this chapter is organized as follows. The related literature is discussed in Section 5.2. Section 5.3 then elaborates on the geometric ℓ_p-norm feature pooling method and provides the theoretical comparison with the max and average pooling methods. An iterative optimization procedure is presented in Section 5.4 for determining the ultimate pooling function. In Section 5.5 extensive experimental results on benchmarks are presented and conclusions are drawn in Section 5.6.

5.2 Related Work

The idea of feature pooling originates in the research on complex cells in the striate cortex [70]. In [70], they proposed a model in which responses of simple cells are fed into higher complex cells through some pooling operations, thereby endowing the complex cells with phase-invariance. Inspired by this seminal work, several extensions in the direction of pooling mechanisms have been proposed afterwards and widely applied in recent computer recognition systems. In the neocognitron model [67], a sigmoid-like function is used to pool the input signals into a single output. And convolutional networks [75] take the average value of the input signals for subsequent processing. Some models of the primary visual cortex area V1 [82] also include such average pooling component. Besides, another type of pooling via max operation is used in the HMAX class of models [84]. Wang *et al.* [85] have achieved impressive classification performance on several benchmarks through such max pooling. As pointed out by Jarrett *et al.* [72], pooling type matters more for classification tasks than careful unsupervised pre-training of features. However, most of the studies on the pooling methods are purely empirical. Recently, Boureau *et al.* [65] provided a theoretical analysis on the binary feature pooling in the context of classification. Based on the i.i.d. Bernoulli distribution assumption, they demonstrated that several factors, including the pooling cardinality and the sparsity of the features, affect the discriminative powers of the pooling results. And neither the average nor max pooling can always outperform the other in classification problems.

Meanwhile a body of works have been devoted to discriminative dictionary learning [88][73], which considers the visual dictionary formation step in the BoW representation pipeline. Instead, our work focuses on the subsequent visual words spatial aggregation step and can be seamlessly combined with any dictionary formation method.

5.3 Geometric ℓ_p-norm Feature Pooling

We assume that there are n_c image classes, and the class index set is denoted as $\mathcal{C} = \{1, 2, \cdots, j, \cdots, n_c\}$. Additionally, we denote the image index set for the j-th class as $\mathcal{I}_j$ and the number of images in the j-th class is denoted as N_j. Denote the location index set as $\mathcal{M} = \{1, 2, \ldots, m, \cdots, M\}$ in an image with M feature locations, *e.g.* distributed over a regular grid. For each image I, we extract a set of d-dimensional local descriptors, *e.g.* SIFT [79], from M densely arranged locations. Then each local descriptor $\mathbf{x}$ is encoded by a pre-trained visual word dictionary $D \in \mathbb{R}^{d \times K}$ into a K-dimensional code vector $\mathbf{u}$ in a pre-defined feasible region $\mathcal{F}$:

$$\begin{aligned} &\mathbf{u} = \arg\min_{\mathbf{u}} ||\mathbf{x} - D\mathbf{u}||_2, \\ &s.t. \quad \mathbf{u} \in \mathcal{F}. \end{aligned} \tag{5.1}$$

When $\mathcal{F}$ is constrained to the set of 0-1 vectors with only a single entry equal to 1, the encoding method is known as the hard assignment. When $\mathcal{F}$ is defined as the set of neighboring bases of the local descriptor $\mathbf{x}$, the resultant $\mathbf{u}$ corresponds to the recently proposed Locality-constrained Linear Coding (LLC) [85].

Each element u_k of the code vector $\mathbf{u}$ indicates the local descriptor's response to the k-th visual word in the dictionary D. We aggregate the local descriptors' responses across all the M image locations into an M-dimensional response vector $\mathbf{v}^{(k)}$. Namely, each element $v_m^{(k)}$ of $\mathbf{v}^{(k)}$ represents the response of the local descriptor $\mathbf{x}_m$ at the m-th location to the k-th visual word.

5.3.1 Pooling Methods Revisit

Feature pooling is essentially to map the response vector $\mathbf{v}^{(k)}$ into a statistic value $f(\mathbf{v}^{(k)})$ via some spatial pooling operation f, where $f(\mathbf{v}^{(k)})$ is used to summarize the joint distribution of visual features over the region of interest. Here, for notational simplicity, we drop the visual word index k for $\mathbf{v}^{(k)}$ in all the following sections.

In modern visual classification models, there are two widely used pooling operations, *i.e.* the average pooling [65] and the max pooling [83]. Average pooling adopts the scaled ℓ_1-norm of the response vector $\mathbf{v}$ as the statistic value and its operation can be expressed as

$$f_a(\mathbf{v}) = \frac{1}{M}||\mathbf{v}||_1 = \frac{1}{M}\sum_{m=1}^{M} v_m. \tag{5.2}$$

Namely, average pooling sums up the response values throughout the entire image or the region of interest in an orderless manner. The pooling result is generally tolerant to object transformation. However, it is not selective or discriminative enough for the classification tasks [81].

Recently, inspired by the mechanism of the complex cells in the primary visual cortex, another pooling operation is proposed in [84]. The so-called max pooling operation computes the ℓ_∞-norm of the response vector,

$$f_m(\mathbf{v}) = ||\mathbf{v}||_\infty = \max_m v_m. \tag{5.3}$$

The max pooling only captures the most salient response over the whole image or the region of interest. Thus it is more selective than the average pooling and able to preserve invariance to object's spatial transformations [84].

However, both pooling methods discard the spatial distributions of local descriptors by either forcing the distribution to be uniform (the average pooling) or only adopting the most salient location (the max pooling). This information loss severely limits their discriminating capability and degrades the performance of the subsequent classification procedures.

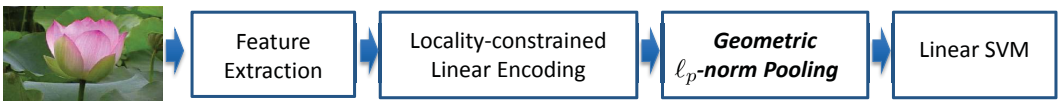


Figure 5.2: Overview of the image classification flowchart. The shown architecture has proven to perform best among the methods based on a single type of features [85]. Here we replace the original max pooling building block with our proposed geometric ℓ_p-norm pooling method, and shall show the new pipeline is better.

In fact, each visual word may exhibit certain geometric structure within individual classes since images for a certain classification task are often well roughly aligned or can be roughly aligned automatically by saliency or symmetry detection. These structures can contribute significantly to the discriminating capability once it is properly utilized as illustrated in Figure 5.1. But once lost, this useful information could never be recovered in the subsequent process. Therefore if we can well model the spatial distribution for individual visual words, the obtained pooling results will be more discriminative than those from traditional pooling methods.

5.3.2 Geometric ℓ_p-norm Pooling

As discussed, both the average and max pooling discard the geometric information of local responses and thus only maintain limited discriminating capability. To overcome this inherent issues, we propose the so-called Geometric ℓ_p-norm Pooling (GLP) method. GLP is aimed at utilizing the spatial distribution patterns of responses across different classes and meanwhile preserving the selective capability and robustness as traditional pooling methods do.

More specifically, GLP process is defined as

$$\begin{aligned} f_g(\mathbf{v};\mathbf{w}) &= \sum_{m=1}^{M} w_m v_m^p = \mathbf{w}^T \mathbf{v}^p, \\ s.t. \quad & ||\mathbf{w}||_2 = 1, \;\; p \geq 0, \end{aligned} \tag{5.4}$$

where $\mathbf{v}^p$ denotes the element-wise p-th power of the response vector $\mathbf{v}$. The geometric coefficient w_m encodes the contribution of the m-th image location for the specific visual word. Different locations are given different weights during the pool-

ing process. The parameter p determines the selection policy for locations. Note that $\mathbf{v}$ has been normalized by its ℓ_2-norm, and all the elements of $\mathbf{v}$ are smaller or equal to 1. Therefore, when the value of p equals to 1, GLP aggregates the responses over the entire region uniformly without preference to any location (same to the average pooling). When p increases to a large value, the policy changes towards winner-take-all (same to the max pooling). Namely, the value of p tunes the pooling operation to transit from the average to the max pooling. Instead of fixing the value of p, GLP adopts a more flexible one and possesses better selective capability. Moreover, in GLP method, the values of $\mathbf{w}$ and p are visual-word-specific. This enables GLP to better capture geometric information of the descriptors based on the fact that different visual features usually follow different spatial distributions among different classes.

5.3.3 Image Classification Procedure

The pipeline of a popular image classification procedure is shown in Figure 5.2. As can be seen from the figure, a multi-stage image classification architecture generally comprises four components. After local features are extracted from the input image, many methods can be used to encode the feature vectors. Here we adopt the Locality-constrained Linear Coding [85] method in our experiments, which encodes the feature vector $\mathbf{x}$ into $\mathbf{u}$ based on the dictionary D as follows,

$$\begin{aligned} &\mathbf{u} = \arg\min_{\mathbf{u}} ||\mathbf{x} - D\mathbf{u}||_2 \\ &s.t.\ \ \mathbf{u} \in \mathcal{F}, \mathcal{F} = \{\mathbf{u} | ||\mathbf{u} \circ \mathbf{d}||_2 \leq \lambda, ||\mathbf{u}||_1 = 1\}, \end{aligned} \tag{5.5}$$

where the entries of $\mathbf{d}$ are the Euclidean distances between $\mathbf{x}$ and the bases in D.

After feature encoding, pooling operations are performed to aggregate the encoded response vectors into a statistic vector to represent the whole image or the region of interest. Finally the pooled feature vector is fed into a classifier, and then further assigned to one of the pre-defined classes. Note that this work is aimed at replacing the pooling component only rather than renewing the whole classification

architecture. In fact, the proposed GLP method can be seamlessly combined with arbitrary types of local features, encoding methods and ultimate classifiers.

5.4 Towards Optimal Geometric Pooling

5.4.1 Class Separability

To determine the parameters in the GLP, we adopt the class separability as the objective function and optimize it with respect to both $\mathbf{w}$ and p. A practical choice of the class separability criterion is the Marginal Fisher Analysis (MFA) developed in [115]. MFA can well characterize the class separability of the data with more general distributions beyond the Gaussian. More specifically, the objective function is to maximize the inter-class separability scaled by the within-class compactness of the pooled features, namely,

$$\max_{\mathbf{w},p}\{\mathcal{D}(\mathbf{w},p) := \frac{\mathbf{w}^T S_b(p)\mathbf{w}}{\mathbf{w}^T S_w(p)\mathbf{w}}\}, \tag{5.6}$$

where $S_b(p)$ characterizes the separability of different classes and $S_w(p)$ describes the within-class compactness [115]. These two matrices are computed as follows,

$$\begin{aligned} S_b(p) &= \sum_i \sum_{j\in N_{k_1}^-(i)} (\mathbf{v}_i^p - \mathbf{v}_j^p)(\mathbf{v}_i^p - \mathbf{v}_j^p)^T, \\ S_w(p) &= \sum_i \sum_{j\in N_{k_2}^+(i)} (\mathbf{v}_i^p - \mathbf{v}_j^p)(\mathbf{v}_i^p - \mathbf{v}_j^p)^T. \end{aligned} \tag{5.7}$$

Here $N_{k_1}^-(i)$ means the index set for the k_1 nearest neighbors of the response vector $\mathbf{v}_i$ from different classes and $N_{k_2}^+(i)$ denotes the k_2 nearest neighbors of $\mathbf{v}_i$ from the same class as $\mathbf{v}_i$.

5.4.2 Spatial Correlation of Local Features

The previous analysis in [65] is based on the strict assumption that the response values from M locations in $\mathbf{v}$ are independent. However, this assumption is of-

ten invalid for real-world data as also mentioned in [65], since image features at adjacent locations are often strongly correlated. Ignoring this important fact may lead to degraded capability in describing images. Although there exists few prior knowledge about the exact form of the spatial correlation, an intuitive and simple constraint/prior is that the geometric coefficients of (5.4) located at adjacent image locations should exhibit similar values. To incorporate this spatial smoothness constraint, we define a spatially smooth function as follows,

$$\mathcal{S}(\mathbf{w}) = \sum_{i,j \in \mathcal{M}, i \neq j} s_{ij} \|w_i - w_j\|^2. \tag{5.8}$$

The value of weight s_{ij} is set as

$$s_{ij} = \exp\left(-\frac{\|\mathbf{a}_i - \mathbf{a}_j\|^2}{2\rho^2}\right), \tag{5.9}$$

where $\mathbf{a}_i$ denotes the spatial coordinates of the i-th feature location. Minimizing $\mathcal{S}(\mathbf{w})$ penalizes the case when adjacent elements of $\mathbf{w}$ show large numerical gap. ρ is an empirical bandwidth parameter of the neighborhood and fixed as 0.5 in all our experiments.

The smooth function can be further rewritten as

$$\mathcal{S}(\mathbf{w}) = \mathbf{w}^T L \mathbf{w}, \tag{5.10}$$

where the Laplacian matrix L is defined as $L = D - S$. The similarity matrix S is defined as $S = [s_{ij}]_{M \times M}$ and the degree matrix D is a diagonal matrix with $D_{ii} = \sum_j S_{ij}$.

5.4.3 Optimal Geometric Pooling

We combine the objectives in both (5.6) and (5.10) into a unified function with a weighting factor λ, namely,

$$\begin{aligned} \max_{\mathbf{w},p} \mathcal{Q}(\mathbf{w},p) &:= \frac{\mathbf{w}^T S_b(p)\mathbf{w}}{\mathbf{w}^T S_w(p)\mathbf{w} + \lambda \mathbf{w}^T L \mathbf{w}} \\ &= \frac{\mathbf{w}^T S_b(p)\mathbf{w}}{\mathbf{w}^T \tilde{S}_w(p)\mathbf{w}}, \end{aligned} \tag{5.11}$$

where $\tilde{S}_w$ is the regularized within-class scatter matrix, *i.e.*, $\tilde{S}_w = S_w + \lambda L$.

Though the optimization problem in (5.11) is not convex overall, but there exists closed form solution for $\mathbf{w}$ when p is fixed. Thus, we solve this optimization problem iteratively by optimizing with respect to p and $\mathbf{w}$ alternatively.

Note that when optimizing for $\mathbf{w}$, this objective function has the same form as the well-known linear discriminative analysis (LDA) [66] algorithm, where $S_b(p)$ corresponds to the between-class scatter matrix and $\tilde{S}_w(p)$ corresponds to the within-class scatter matrix. Here we borrow the analytical solution from LDA to derive the optimal solution $\mathbf{w}_{opt}$ to (5.11) with p fixed:

$$\begin{aligned} &\mathbf{w}_{opt} = \arg\max_{\mathbf{w}} \gamma, \\ &s.t. \quad S_b\mathbf{w} = \gamma \tilde{S}_w \mathbf{w}. \end{aligned} \tag{5.12}$$

The solution $\mathbf{w}_{opt}$ is the eigenvector corresponding to the largest eigenvalue.

For the optimization of (5.11) with respect to p, there is no closed-form solution. We adopt a gradient ascent process to solve p in an iterative manner. Let y denote the pooled feature $y = \mathbf{w}^T \mathbf{v}^p$, thus the between-class and within-class scatter

matrices of the pooled features can be written as

$$\begin{aligned}
\hat{S}_b(p) &= \mathbf{w}^T S_b(p)\mathbf{w} \\
&= \sum_i \sum_{j\in N_{k_1}^-(i)} (y_i - y_j)^2, \\
\hat{S}_w(p) &= \mathbf{w}^T \tilde{S}_w(p)\mathbf{w} \\
&= \sum_i \sum_{j\in N_{k_2}^+(i)} (y_i - y_j)^2 + \lambda \mathbf{w}^T L \mathbf{w}.
\end{aligned} \tag{5.13}$$

We use $\boldsymbol{\alpha}$ to denote the Hadamard product $\boldsymbol{\alpha} = \ln \mathbf{v} \circ \mathbf{v}^p$. Then the derivatives of $\hat{S}_b$ and $\hat{S}_w$ with respect to p are as follows,

$$\begin{aligned}
\frac{\partial}{\partial p}\hat{S}_b &= 2\sum_i \sum_{j\in N_{k_1}^-(i)} (y_i - y_j)\mathbf{w}^T(\boldsymbol{\alpha}_i - \boldsymbol{\alpha}_j), \\
\frac{\partial}{\partial p}\hat{S}_w &= 2\sum_i \sum_{j\in N_{k_2}^+(i)} (y_i - y_j)\mathbf{w}^T(\boldsymbol{\alpha}_i - \boldsymbol{\alpha}_j).
\end{aligned} \tag{5.14}$$

The partial derivative of the objective function (5.11) with respect to p is

$$\nabla p = \frac{\partial}{\partial p}\mathcal{Q} = \frac{1}{\hat{S}_w^2}\left(\frac{\partial \hat{S}_w}{\partial p}\hat{S}_b - \frac{\partial \hat{S}_b}{\partial p}\hat{S}_w\right). \tag{5.15}$$

Thus we update p along the gradient direction with step size β as follows,

$$p^{(t+1)} = p^{(t)} + \beta \nabla p. \tag{5.16}$$

The process will stop when the change of p is less than a pre-defined threshold θ_p or the number of iterations exceeds the permitted number N_{iter}.

5.5 Experiments

In this section, we evaluate the performance of the proposed GLP method on the image classification task and compare it with the average and max pooling methods. First, we investigate the separability of the pooling results produced by GLP

and the other two methods on a synthesized data set, which possesses distinctive spatial distribution patterns for different classes. Then we evaluate GLP along with the average and max pooling on three popular real-world datasets: Caltech-101 dataset [77], Caltech-256 dataset [69] and 15 scenes dataset [74]. The classification performances based on these three pooling methods are compared under two different feature representing schemes: one is based on the hard assignment and the other is based on the combination of Locality-constrained Linear Coding (LLC) [85] and Spatial Pyramid Matching (SPM) [74].

5.5.1 Effectiveness of Feature Spatial Distribution

A set of randomly generated data is used to investigate the effectiveness of feature spatial distribution for classification purpose. The synthesized dataset comprises two classes of data, with distinctive spatial distribution per class. There are 200 data matrices for each class. The size of the matrix is fixed as 30×30 to simulate an image with 30×30 feature locations. Each element of the matrix is a binary variable to indicate the presence of certain visual feature at the corresponding location. Random transitional noise with magnitude ranging from 1 to 20 locations are added to each datum. Figure 5.3 shows two exemplar data from different classes. We perform average pooling, max pooling and GLP on this dataset and plot the distributions of the pooling results in Figure 5.3. From the derived pooling-feature distribution, it can be seen that neither average pooling nor max pooling can well separate these two classes due to the loss of the spatial information, while GLP properly utilizes the features' class-specific spatial distributions and the resultant statistics are separable. GLP produces a discriminative pooling coefficients map as shown in Figure 5.3.

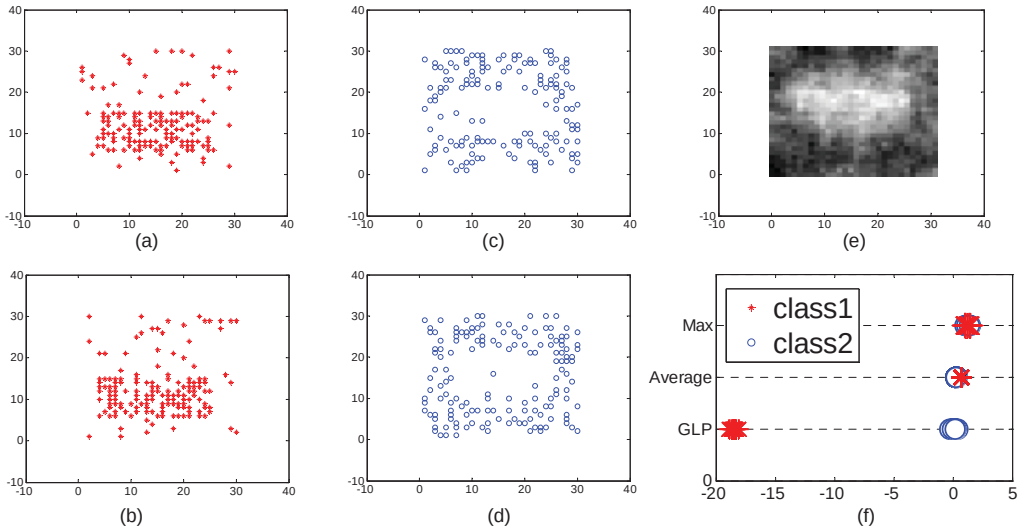


Figure 5.3: Comparison of GLP and average/max pooling over the synthesized data with distinctive feature distributions for different classes. (a), (b) and (c), (d) show the exemplar data from two different classes respectively. (e) displays the optimized geometric coefficients over the region. Brighter pixels mean that the coefficients are larger at the corresponding locations. (f) shows the pooling results distribution via the average, max and GLP poolings. It can be seen that GLP can separate the data from two classes well while average pooling and max pooling cannot.

5.5.2 Object and Scene Classification

Experiment Configurations

In this subsection, we continue with the comparison on real image datasets for object and scene classification. The purposes of the experiments are two-fold. The first one is to compare GLP directly with the other two pooling methods. The second one is to evaluate the performance of the new image classification framework which includes GLP as a new plug-in, and compare it with the state-of-the-arts methods.

These two groups of experiments follow the common experimental settings. We only use a single type of local descriptor, dense SIFT [79], throughout the experiments. The SIFT features are extracted from densely located patches centered at every 4 pixels on the images and the size of the patches is fixed as 16×16 pixels. We construct a visual word dictionary containing K words from the training samples via K-means clustering. The value of K depends on the number of samples and varies across different datasets. Each SIFT feature vector is encoded into a K-dimensional code vector based on the dictionary. Then the code vectors from

every image are pooled into a single feature vector via different pooling methods. During the training process of the GLP, we use all the training samples to calculate the MFA scatter matrices and we set $k_1 = 20$ and $k_2 = 20$. The maximum number of the alternations between optimizing $\mathbf{w}$ and p is fixed as $N_{alter} = 10$. The maximal number of iterations when solving p is $N_{iter} = 50$. The stopping threshold of updating p is $\theta_p = 0.1$. The pooled features are used to train a multi-class linear SVM. In our experiments, all the images are resized to 256×256 pixels. When GLP is directly compared with other pooling methods, the code vectors are generated by hard assignment. For the GLP, 30, 30 and 100 training samples are used for Caltech-101, Caltech-256, 15 Scenes dataset respectively.

In the second group of experiments, we apply the GLP as a new pooling component in the multi-stage image classification architecture proposed in [85]. The original architecture consists of four components: image local feature extraction, feature encoding, feature pooling and spatial pyramid matching (SPM) [74], followed by linear SVM classifier. Here, we replace the max pooling component with the GLP method, and compare the image classification performance with the original one and other state-of-the-art methods. We follow the same experimental setting as in [85]. SIFT features are encoded by Locality-constrained Linear Coding (LLC) [85] and the number of neighbors is fixed as 5. Images are hierarchically partitioned into 1×1, 2×2 and 4×4 blocks on 3 levels respectively in the SPM. We visualize the geometric coefficients in Figure 5.4, from which it can be seen that the coefficients distribution derived from GLP are able to capture the visual word spatial distributions.

Table 5.1: Accuracy comparison of image classification using hard assignment for three different pooling methods.

	Caltech-101	Caltech-256	15 Scenes
Average	44.16	22.01	56.72
Max	48.14	18.45	54.19
GLP	**55.20**	**33.12**	**65.70**

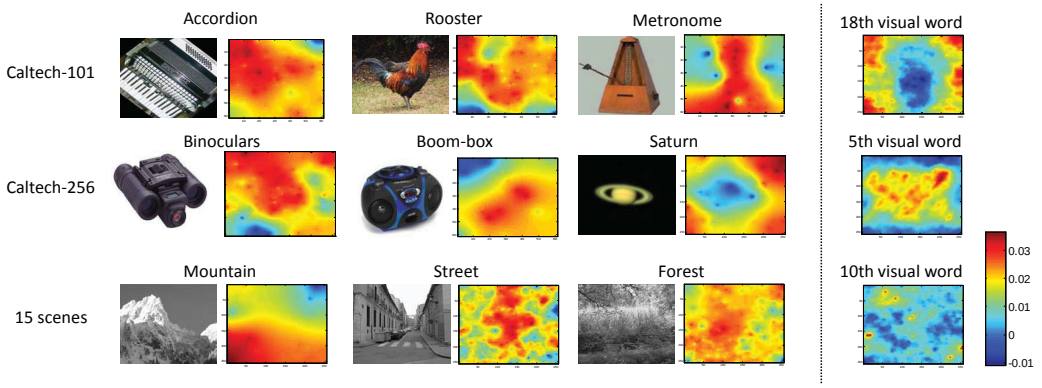


Figure 5.4: Visualization of the pursued geometric coefficient maps for each specific visual word over different classes. The left 6 columns show the exemplar images from 3 classes per dataset and their corresponding geometric coefficient distribution maps. The coefficients for one specific class are computed in one-vs-all manner. The right most column shows the geometric coefficients for one specific visual word, derived from GLP over all the classes. Each row displays for one dataset. For better view, please refer to the color version.

Results on Caltech-101 Dataset

The Caltech-101 dataset [77] contains 9144 images in total from 102 different categories, including 101 object categories and 1 additional background category. The number of images per category ranges from 31 to 800. The resolution of most images is about 300×300 pixels. Following the setting in [85] [76], we randomly select 5, 15 and 30 images respectively for training and report the classification accuracies averaged over the 102 categories. The size of visual word dictionary is set as $K = 2048$ as in [85].

Table 5.2: Classification accuracy (%) comparison on Caltech-101 dataset.

Algorithms	5 training	15 training	30 training
Zhang *et al.* [89]	46.60	59.10	66.20
KSPM [74]	–	56.40	64.60
NBNN [63]	–	65.00	70.40
ML+CORR [71]	–	61.00	69.10
KC [68]	–	–	64.16
ScSPM [87]	–	67.00	73.20
LLC [85]	51.15	65.43	73.44
GLP	**59.35**	**70.34**	**82.60**

The performance comparison of different pooling methods based on the hard-

assignment encoding scheme is shown in the first column of Table 5.1. It can be seen that GLP consistently outperforms the average and max pooling by a margin of 11% and 7% respectively. The classification accuracy of GLP combined with LLC and SPM is shown in Table 5.2. Based on the comparison with the original LLC [85], we can observe that the performance improvement brought by GLP is nearly 9% when using 30 training samples. Also GLP outperforms all the single type of feature based methods. The performance has already exceeded the best one (82.3%) ever reported on the Caltech-101 dataset in [76]. This result is very encouraging as the method in [76] utilizes the groundtruth segmentations, which is not available for real applications, to perform regression and uses 8 different types of features in total. In sharp comparison, our method only uses a single type of feature (dense SIFT) and needs no segmentation results to be provided.

Results on Caltech-256 Dataset

Caltech-256 [69] is an extension of the Caltech-101 dataset. It consists of 256 object categories and contains from 80 to 827 images per category. The total number of images is 30608. This dataset possesses larger intra-class variability than the Caltech-101 and thus is more challenging. As in [85], 15, 30 and 45 images from each category are used for training respectively, and we use a 4096-D visual word dictionary as in [85]. As can be seen from the second column of Table 5.1, GLP also consistently leads the performance compared with other pooling methods on this dataset under hard-assignment setting. Also our method outperforms the state-of-the-art method (LLC) on this dataset[1], with a margin of 2% as shown in Table 5.3.

Results on 15 Scenes Dataset

Scene-15 dataset is composed of 15 scene classes. Each class contains 200 to 400 images and there are 4485 images in total. The scene categories contain from the out-door street and industrial to the in-door kitchen and living room. As in [74] [87],

[1]Though the highest accuracy reported on this dataset is 45.3% [64] for 30 training samples setting, they only evaluate the performance on 250 classes. Thus the result is not comparable with LLC [85] and ours.

Table 5.3: Classification accuracy (%) comparison on Caltech-256 dataset.

Algorithms	15 training	30 training	45 training
KSPM [69]	–	34.10	–
ScSPM [87]	27.73	34.02	37.46
LLC [85]	34.36	41.19	45.31
KC [68]	–	27.17	–
GLP	**35.78**	**43.17**	**47.32**

we randomly select 100 images from each class as training samples to construct a 1024-D visual dictionary. The improvements brought by GLP over the average and max pooling are 9% and 11% respectively as shown in the third column of Table 5.1 under hard-assignment setting. Also from Table 5.4 we can see that GLP under the setting with LLC and SPM can improve the classification performance further with a margin of 4% compared with LLC and outperforms KSPM by nearly 2%. The highest accuracy on this dataset is 86.1%, achieved by Nakayama *et al.* [80]. But in that work, they used a much more sophisticated KL-divergence kernel SVM than our linear SVM. This encouraging result further demonstrates that the proposed GLP can enhance the pooling features' discriminating capability remarkably.

Table 5.4: Classification accuracy (%) comparison on 15 scenes dataset.

Algorithms	Accuracy
KSPM [74]	81.40
ScSPM [87]	80.28
LLC	79.24
GLP	**83.20**

5.6 Chapter Summary

In this chapter, we proposed the so-called geometric ℓ_p-norm pooling (GLP) method to perform feature pooling. Different from traditional feature pooling methods, *e.g.* the average and max pooling, the GLP method can utilize the geometric information on the feature spatial distributions and thus provide more discriminative pooling results. Moreover, it explicitly models the spatial correlations of the local

responses in a more elaborated way rather than blindly simplifying them to be flat or independent as in average and max pooling methods. Therefore, GLP is a more discriminative, flexible and information-preserving pooling method. Comprehensive experimental results on several benchmarks have demonstrated that GLP can serve as a highly effective building block for the image classification architecture and boost the performance to outperform or be comparable with the state-of-the-arts.

Note that the proposed GLP method can by no means handle all the general cases (*e.g.* PASCAL VOC datasets), especially when the dataset contains large intra-class spatial variances. This is due to the fact that GLP is tightly tied to the feature positions. A partial remedy to this for many practical scenarios is we can exploit the visual saliency or foreground detection to help roughly align the images automatically. But the more challenging cases still invite future research.

Chapter 6

Auto-grouped Sparse Representation for Visual Analysis

In this chapter, we show that how to discover the low-dimensional group structure within the image feature vectors effectively. The group structure helps provide more accurate and flexible similarity measure between images and enhance the performance of image annotation.

6.1 Introduction

Most of current image analysis methods represent images by aggregating local features into image-level features, such as bag of words model [90, 91, 92]. These methods generally ignore the fact that the local features may be from different objects and treat the image-level feature as a whole in the follow-up computation. Such over-simplified strategy may render the results of image analysis inaccurate. For example, given two multi-object images containing one common object, they should be assigned one identical annotation. Indeed they are quite similar if only considering local features from the common object. But their image-level features may differ greatly due to involving local features from non-common objects and background

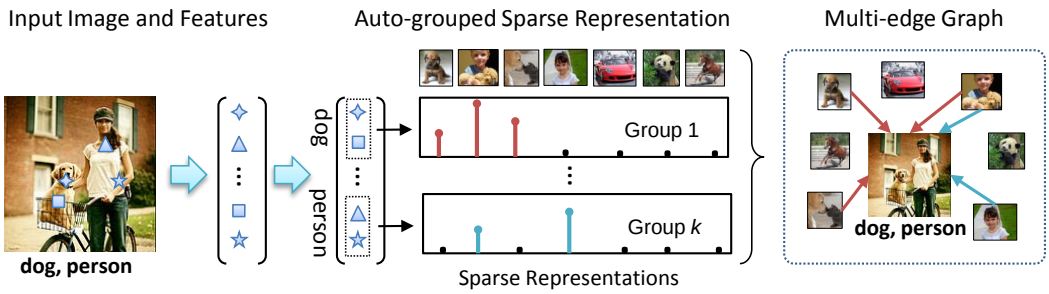


Figure 6.1: Illustration on the proposed auto-grouped sparse representation method. The elements of the image-level feature represent different visual patterns. The feature elements are divided into k groups according to their individual sparse representations. Each group represents one specific object. Based on the group-wise sparse representations, a multi-edge graph is constructed to describe the relationship between the images.

and thus mislead the image similarity computation. To handle such mutual interference of multiple objects in image analysis, several previous works propose to perform segmentation or detection as pre-processing before feature extraction [93, 94]. However, such pre-processing is quite complicated, computationally expensive and inefficient.

In order to alleviate the mutual interference of multiple objects in given image-level features, we propose to divide its elements into several independent groups, such that each group represents typical visual patterns for one object. Thus, local features from different objects can be segregated to some extent and semantically different objects are considered independently in the follow-up computations. Then we can obtain analysis result (*e.g.*, image similarity) specific for one object, which is immune to the interference from other objects and background, as desired. In this way, we can obtain more accurate image relationship description from original image-level features and improve their performance in various visual analysis tasks.

To this end, we propose a novel auto-grouped sparse representation (ASR) method to *automatically learn* the intrinsic semantic groups of an image-level feature vector. The pursuit groups should roughly reside on the identical subspace if they correspond to objects from the same class [95, 96, 97]. And ASR computes multiple sparse representations for elements of an input image-level feature vector w.r.t. an

over-complete basis to identify the subspaces (corresponding to the objects) [98]. In particular, ASR performs single sparse representation over each feature element, and meanwhile it imposes fusion-encouraging regularization to force the semantically correlated feature elements to share the same sparse representation. Thus the feature elements corresponding to the same object, namely falling on the same subspace, can be grouped together since they possess similar sparse representations.

Fig. 6.1 provides an example to illustrate the proposed ASR method. Given an input image, its feature elements representing the same object (*person* or *dog*) fall on identical subspace and thus will select identical basis images (containing *person* or *dog* respectively) in their sparse representations. Thus elements of its image-level feature can be divided into k groups according to their sparse representations. Each feature group includes several characteristic visual patterns for one specific object. In a multi-edge graph constructed based on the multiple sparse representation models, the input image is connected with the basis images via varying number of edges, reflecting the relevance degree between them. It can be seen that such relationship is more accurate and flexible.

Note that the proposed ASR is a general method and can also be applied for other intrinsic group identification tasks, such as motion segmentation. In this work, we examine the applicability of ASR in two practical visual analysis tasks. The first application is to build the multi-edge graph [99] for more accurately classifying multi-label images. Compared with conventional single edge graph, our multi-edge graph achieves the state-of-the-art performance on the NUS-WIDE-LITE database. And the second one is two-view motion segmentation. ASR segments the motion trajectories by grouping the corresponding mixture linear regression models. Compared with previously well-performed methods [100, 101], ASR significantly decreases the segmentation error rates and offers more accurate and stable segmentation results.

The remainder of this chapter is organized as follows. We discuss related work in Section 6.2. We present the problem formulation in Section 6.3 and detail the smooth approximation optimization in Section 6.4. We conduct several experiments in Section 6.5 to evaluate the proposed method. We provide some discussion in

Section **??** and conclude our work in Section 6.6.

6.2 Related Work

The proposed work aims at automatically uncovering the group structures across multiple feature entries and simultaneously calculating the underlying sparse representations within each group. The most intuitive approach to tackle this problem is the Expectation-Maximization (EM) method [100]. EM may regard the group assignments as hidden variables, and iterates the inference over hidden variables and the parameter estimation of decoupled models until a *local* optimum is reached. Gaffney *et al.* [102] applied the EM method to the trajectory clustering with the assumption that the motion trajectories are generated from a mixture regression model. The documentable limitation of the EM method is the locality of its optimization and thus the final solution is typically sensitive to initialization.

The second type of approaches may be based on convex relaxation. Recently, Quadrianto *et al.* [103] proposed to solve the regression model with mixture of several regression vectors by relaxing the assignment variables into continuous ones. Their experimental results show that the convex formulation performs better than the EM method on a number of benchmark datasets. However, their formulation seems hard to be generalized to sparse representation setting. Indeed, to the best of our knowledge, there has been no effort on solving the sparse mixture regression problem in a convex optimization framework.

Our method is directly inspired by the convex relaxation of clustering [104], where the authors employ the sparsity-inducing norms to enforce the fusion of data points. Sparsity-inducing norms have emerged as flexible tools that allow variable selection in penalized linear models [105, 106]. In this chapter, we combine these lines of research into our framework of auto-grouped sparse representation.

6.3 Problem Formulation

In this work, we propose an auto-grouped sparse representation (ASR) method to automatically identify the intrinsic group structure of a given feature vector. In particular, the elements of a feature vector $\mathbf{y} \in \mathbb{R}^p$ constitute K non-overlapped groups $\{\mathbf{y}_{\mathcal{C}_1}, \ldots, \mathbf{y}_{\mathcal{C}_K}\}$, each of which represents one specific object and admits a specific sparse representation $\boldsymbol{\omega}_k \in \mathbb{R}^n$ w.r.t. the sub-matrix of provided over-complete basis matrix $A \in \mathbb{R}^{p \times n}$. And $\mathcal{C}_k \subseteq \{1, \ldots, p\}$ is the feature element indices contained in the k^{th} group. We aim to find the groups of elements in $\mathbf{y}$ and simultaneously estimate their sparse representations by optimizing the following objective,

$$\min_{\{\mathcal{C}_k, \boldsymbol{\omega}_k\}_{k=1}^K} \left\{ \frac{1}{2} \sum_{k=1}^{K} \|\mathbf{y}_{\mathcal{C}_k} - A_{\mathcal{C}_k} \boldsymbol{\omega}_k\|_{\ell_2}^2 + \lambda \sum_{k=1}^{K} \|\boldsymbol{\omega}_k\|_{\ell_1} \right\}, \tag{6.1}$$

where $\mathbf{y}_{\mathcal{C}_k}$ denotes elements of $\mathbf{y}$ indexed by $\mathcal{C}_k$ and $A_{\mathcal{C}_k}$ denotes rows indexed by $\mathcal{C}_k$ in the matrix A. In the above optimization problem, each element of $\mathbf{y}$ is assigned to its corresponding group such that the overall loss is minimized.

The above objective function is a combinatorial optimization problem and in general computationally intractable. Following the relaxation technique introduced in [104], we relax the hard constraint on the number of groups to the fusion-encouraging constraint on the sparse representations $\{\mathbf{w}_i\}_{i=1}^p \subset \mathbb{R}^n$ of all elements in $\mathbf{y}$:[1]

$$\begin{aligned} &\min_{\{\mathbf{w}_i\}_{i=1}^p} \left\{ \frac{1}{2} \sum_{i=1}^{p} \|y_i - A_i \mathbf{w}_i\|_{\ell_2}^2 + \lambda \sum_{i=1}^{p} \|\mathbf{w}_i\|_{\ell_1} \right\}, \\ &\text{subject to :} \ \sum_{i<j} \mathbf{1}_{\mathbf{w}_i \neq \mathbf{w}_j} \leq t. \end{aligned} \tag{6.2}$$

Here y_i denotes the i^{th} element of the vector $\mathbf{y}$, A_i denotes the i^{th} row of the matrix A. The indicator function $\mathbf{1}_{\mathbf{w}_i \neq \mathbf{w}_j}$ takes value 1 if $\mathbf{w}_i$, $\mathbf{w}_j$ are unequal, and 0 otherwise; $\sum_{i<j}$ denotes $\sum_{i=1}^{p-1} \sum_{j=i+1}^{p}$. Intuitively, the constraint on the number of different vectors $\mathbf{w}_i$ serves as a proxy of constraining the number of groups. When

[1] More details and underlying rationale are referred to [104].

$t \geq p(p-1)/2$, it amounts to each entry forming an individual group. Otherwise, along with the decrease of t, more feature elements are assigned into the same group.

However, due to the non-convexity of the indicator function, Problem (6.2) remains computationally hard. Here, we replace the indicator function by ℓ_∞-norm [104], which results in the following convex optimization problem:

$$\begin{aligned} &\min_{\{\mathbf{w}_i\}_{i=1}^p} \left\{ \frac{1}{2} \sum_{i=1}^{p} \|y_i - A_i \mathbf{w}_i\|_{\ell_2}^2 + \lambda \sum_{i=1}^{p} \|\mathbf{w}_i\|_{\ell_1} \right\}, \\ &\text{subject to :} \sum_{i<j} \|\mathbf{w}_i - \mathbf{w}_j\|_{\ell_\infty} \leq t. \end{aligned}$$

The constraint imposed by the ℓ_∞-norm encourages the maximal difference between two vectors to be zero, namely fusing them together. It can be equivalently expressed in following regularization form:

$$\min_{\{\mathbf{w}_i\}_{i=1}^p} \left\{ \frac{1}{2} \sum_{i=1}^{p} \|y_i - A_i \mathbf{w}_i\|_{\ell_2}^2 + \lambda \sum_{i=1}^{p} \|\mathbf{w}_i\|_{\ell_1} + \beta \sum_{i<j} \|\mathbf{w}_i - \mathbf{w}_j\|_{\ell_\infty} \right\}. \tag{6.3}$$

The objective function of Problem (6.3) consists of a smooth loss term and two non-smooth regularization terms. In particular, we decompose the objective function $f(\mathbf{w})$ into the following two terms:

$$\begin{aligned} \hat{f}(\mathbf{w}) &:= \frac{1}{2} \sum_{i=1}^{p} \|y_i - A_i \mathbf{w}_i\|_{\ell_2}^2, \\ r(\mathbf{w}) &:= \lambda \sum_{i=1}^{p} \|\mathbf{w}_i\|_{\ell_1} + \beta \sum_{i<j} \|\mathbf{w}_i - \mathbf{w}_j\|_{\ell_\infty}. \end{aligned}$$

The problem bearing such non-smooth terms can be solved by smooth approximation [107]. We provide the optimization details in the following section. Using the proposed ASR, the feature element-wise sparse representations $\{\mathbf{w}_i\}_{i=1}^p$ are effectively recovered. In certain cases they may not exactly form distinct groups $\{\boldsymbol{\omega}_k\}_{k=1}^K$. However, it is still possible to construct reliable groups. In this work, we build an affinity graph of these representation vectors, and use a gap in the distri-

bution of eigenvalues of the corresponding Laplacian matrix to estimate the number of groups K. Then spectral clustering techniques [108] can be applied to the affinity graph to cluster the representation vectors $\{\mathbf{w}_i\}_{i=1}^p$ into K groups. And we obtain the feature elements group $\{\mathbf{y}_k\}_{k=1}^K$ accordingly. Then $\{\boldsymbol{\omega}_k\}_{k=1}^K$ can be estimated by performing sparse representation on each group individually.

6.4 Optimization Procedure

6.4.1 Smooth Approximation

According to the smooth approximation proposed in [107], the non-smooth regularization term $r(\mathbf{w})$ can be approximated by the following smoothed one,

$$r_\mu(\mathbf{w}) = \lambda \sum_{i=1}^{p} s_\mu(\mathbf{w}_i) + \beta \sum_{i<j} q_\mu(\mathbf{w}_i, \mathbf{w}_j),$$

where

$$s_\mu(\mathbf{w}_i) := \max_{\|\mathbf{v}\|_{\ell_\infty} \leq 1} \langle \mathbf{w}_i, \mathbf{v} \rangle - \frac{\mu}{2} \|\mathbf{v}\|_{\ell_2}^2, \tag{6.4}$$

$$q_\mu(\mathbf{w}_i, \mathbf{w}_j) := \max_{\|\mathbf{v}\|_{\ell_1} \leq 1} \langle \mathbf{w}_i - \mathbf{w}_j, \mathbf{v} \rangle - \frac{\mu}{2} \|\mathbf{v}\|_{\ell_2}^2. \tag{6.5}$$

Herein, μ is a parameter to control the approximation accuracy and fixed as 1×10^{-4} throughout the experiments. For a fixed $\mathbf{w}_i$, denote $\mathbf{v}(\mathbf{w}_i)$ the unique maximizer of (6.4). It is standard that $\mathbf{v}(\mathbf{w}_i) = \min\{1, \max\{-1, \mathbf{w}_i/\mu\}\}$ where operators $\max\{\cdot,\cdot\}$ and $\min\{\cdot,\cdot\}$ are performed in element-wise for the involved vectors. Moreover, $s_\mu(\mathbf{w}_i)$ is differentiable and its gradient $\nabla s_\mu = \mathbf{v}(\mathbf{w}_i)$ is Lipschitz continuous with the constant $L_s = 1/\mu$ [109]. Also, denote $\mathbf{v}(\mathbf{w}_i, \mathbf{w}_j)$ the unique maximizer of (6.5). Then $\mathbf{v}(\mathbf{w}_i, \mathbf{w}_j)$ can be easily obtained via the ℓ_1-ball projection algorithm [110]. Moreover, $q_\mu(\mathbf{w}_i)$ is differentiable and its gradient $\nabla q_\mu(\mathbf{w}_i) = \sum_{j \neq i} \mathbf{v}(\mathbf{w}_i, \mathbf{w}_j)$ is Lipschitz continuous with the constant $L_q = 1/\mu$ for each term [109].

6.4.2 Optimization of the Smoothed Objective Function

For a fixed small smoothness parameter μ, we are going to minimize the following smoothed objective function,

$$f_\mu(\mathbf{w}) := \hat{f}(\mathbf{w}) + r_\mu(\mathbf{w}). \tag{6.6}$$

It is known that $f_\mu(\mathbf{w})$ is differentiable with the gradient:

$$\nabla f_\mu(\mathbf{w}_i) = \nabla \hat{f}(\mathbf{w}_i) + \nabla r_\mu(\mathbf{w}_i), \tag{6.7}$$

where,

$$\begin{aligned} \nabla \hat{f}(\mathbf{w}_i) &= A_i^T \left(A_i \mathbf{w}_i - y_i \right), \\ \nabla r_\mu(\mathbf{w}_i) &= \mathbf{v}(\mathbf{w}_i) + \sum_{j \neq i} \mathbf{v}(\mathbf{w}_i, \mathbf{w}_j). \end{aligned}$$

It is straightforward to verify that $\nabla \hat{f}(\mathbf{w})$ is Lipschitz continuous with constant $L_f = \|A^T A\|_2$, where $\|\cdot\|_2$ denotes the spectral norm of a matrix. Combining the discussion in the previous subsection, we get that $\nabla f_\mu(\mathbf{w}_i)$ is Lipschitz continuous with the constant,

$$L_{\hat{f}_\mu} = \|A^T A\|_2 + \frac{1}{\mu} \left(\lambda + \beta \right). \tag{6.8}$$

In particular, we employ the Accelerated Proximal Gradient (APG) method [111] to optimize $f_\mu(\mathbf{w})$. The detailed optimization procedure is provided in Algorithm 6.

6.4.3 Convergence Analysis

The following theorem guarantees the convergence of Algorithm 6.

Theorem 17. *Let the sequences $\{\mathbf{w}^{(t)}\}_{t=0}^{\infty}$ be generated by Algorithm 6. Then for any $t \geq 0$, we have,*

$$f_\mu(\mathbf{w}^{(t)}) - f_\mu(\mathbf{w}^*) \leq \frac{4 L_{f_\mu} \|\mathbf{w}^*\|_{\ell_2}^2}{(t+1)(t+2)},$$

Algorithm 6 Smooth minimization for objective (6.3).

Input: $A \in \mathbb{R}^{p\times n}$, $\mathbf{y} \in \mathbb{R}^p$, λ, β, $\text{iter}_{\max}$ and ϵ.
Output: $\mathbf{w}_i, i = 1, \ldots, p$.
Initialization: Calculate L_{f_μ} according to Eqn. (6.8). Initialize $\mathbf{w}^{(0)} \in \mathbb{R}^p$, $\boldsymbol{\gamma}^{(0)} \in \mathbb{R}^p$, and let $\eta^{(0)} \leftarrow 1, t \leftarrow 0$.
repeat
  $\boldsymbol{\alpha}^{(t)} = (1-\eta^{(t)})\mathbf{w}^{(t)} + \eta^{(t)}\boldsymbol{\gamma}^{(t)}$,
  Calculate $\nabla f_\mu(\boldsymbol{\alpha}^{(t)})$ according to Eqn. (6.7),
  $\boldsymbol{\gamma}^{(t+1)} = \boldsymbol{\gamma}^{(t)} - \frac{1}{\eta^{(t)}L_{f_\mu}}\nabla f_\mu(\boldsymbol{\alpha}^{(t)})$,
  $\mathbf{w}^{(t+1)} = (1-\eta^{(t)})\mathbf{w}^{(t)} + \eta^{(t)}\boldsymbol{\gamma}^{(t+1)}$,
  $\eta^{(t+1)} = \frac{2}{t+1}, t \leftarrow t+1$.
until $t > \text{iter}_{\max}$ or $|f_\mu(\mathbf{w}^{(t+1)}) - f_\mu(\mathbf{w}^{(t)})| < \epsilon$.

where $\mathbf{w}^*$ *is an optimal solution to the problem (6.6) and* L_{f_μ} *is the Lipschitz constant of the function* $f_\mu(\cdot)$ *calculated in Eqn. (6.8).*

The above theorem can be directly derived from Theorem 2 in [107]. From Theorem 17, for a fixed μ, it can be seen that Algorithm 6 has the optimal rate of convergence $O(1/t^2)$, where t is the number of iterations. In terms of the desired residue, *i.e.*, $|f_\mu - \min f_\mu| \leq \epsilon$, the rate of convergence is $O(1/\sqrt{\epsilon})$.

6.4.4 Complexity Discussions

Despite the convexity of the proposed method, the optimization procedure is still challenging when the scale of the problem is relatively large. In ASR we individually optimize n-dimensional models for in total p feature entries, thus the number of variables for the primal problem is $O(pn)$, and $O(p^2)$ penalties are imposed. Due to using the smoothing terms, $O(p^2)$ more variables are introduced and $O(p^2)$ projections have to be performed at each iteration. In this work, to reduce the computational expense, an unsupervised entry-wise clustering (*e.g., k*-means) is performed in advance and the method turns to operate on the roughly clustered features. Suppose the number of cluster $K \ll p$, then the number of variables is reduced to $O(Kn)$ and the number of constraints is reduced to $O(K^2)$.

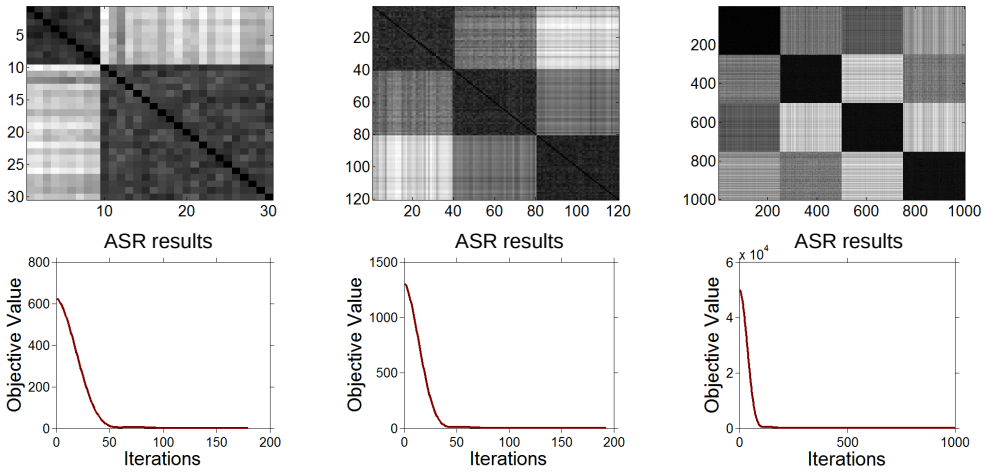
6.5 Experiments

In this section, we apply the proposed auto-grouped sparse representation (ASR) method to solve two concerned applications, including motion segmentation and multi-edge graph for multi-label image classification. We evaluate the performance of the proposed ASR method on a synthetic dataset and two realistic benchmark datasets respectively. The details on how to adapt the ASR method to the practical applications are also provided in the following subsections.

6.5.1 Toy Problem: Sparse Mixture Regression

We first apply ASR in sparse mixture regression problem [103] to verify its effectiveness in uncovering data's group structure. The observed data points $\{y_i\}_{i=1}^N$ are generated according to the linear model $y_i = \boldsymbol{\omega}_k^T \mathbf{a}_i + \varepsilon$. Here $\mathbf{a}_i$ is a given regressor vector, $\boldsymbol{\omega}_k$ is selected from a mixture of sparse linear models $\{\boldsymbol{\omega}_k\}_{k=1}^K$ and $\varepsilon \sim \mathcal{N}(0,1)$ is added Gaussian noise. Here the data points $\{y_i\}_{i=1}^N$ can be stacked into a feature vector $\mathbf{y} = [y_1, \ldots, y_N]^T$ and the regressor vectors are stacked to form the basis matrix $A = [\mathbf{a}_1, \ldots, \mathbf{a}_n]^T$. The mixture regression aims to estimate the K regression models $\{\boldsymbol{\omega}_k\}_{k=1}^K$ according to $\{y_i, \mathbf{a}_i\}_{i=1}^N$. And simultaneously data points $\{y_i\}_{i=1}^N$ are separated into K groups in which the data points are generated by the same linear model. Namely, it aims to find the group structure of the input vector $\mathbf{y}$ according to the underlying linear regression models of its elements.

In this experiment, we apply ASR on the dataset generated by varying number of linear models with $K = 2$, 3, 4. The number of data points n is respectively set as 30, 120, 1000. Data dimension p is fixed as 10. Each element of $\mathbf{a}_i$ and $\boldsymbol{\omega}_k$ is i.i.d. sampled from a uniform distribution on the unit interval. The models $\{\boldsymbol{\omega}_k\}$ are sparsified by randomly zeroing half of their elements. The value of regularization parameters are set as $\beta = 1/p^2$ and $\lambda = 0.1$. And the convergence parameters are fixed as $\epsilon = 1 \times 10^{-4}$ and $\text{iter}_{\max} = 10,000$. Fig. 6.2 shows the curves of the objective function values in Eqn. (6.3) along the optimization iterations, and the obtained data groups. The clear block diagonal structure of the ℓ_∞ distance matrix of the



(a) $K = 2$ (b) $K = 3$ (c) $K = 4$

Figure 6.2: Auto-grouped results from ASR on the synthetic datasets for sparse mixture regression. Top panel shows the ℓ_∞-distance matrices of the recovered regression models, where darker color means smaller distance. And bottom panel shows the convergence curves of the optimization processes.

uncovered linear models well demonstrates the ability of ASR to cluster the mixed data correctly. From the convergence curve, it can be seen that objective function converges within less than 200 iterations, which shows satisfying convergence rate.

6.5.2 Multi-edge Graph For Image Classification

Multi-edge graph vs. single-edge graph

A type of popular methods for image classification is to perform semi-supervised learning based on a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ [112, 113]. Here each vertex $v_i \in \mathcal{V}$ represents an image which is described by a feature vector $\mathbf{y}_i \in \mathbb{R}^p$. And the edge $e_{ij} \in \mathcal{E}$ from the i^{th} to j^{th} vertex, with weight w_{ij}, represents their similarity. In traditional graphs, such as k-NN graph and ℓ_1-graph [114], similarity of two vertices is calculated based on the feature-level measure and represented by a single edge. However, as pointed out in the introduction, multiple intrinsic groups may exist in one feature vector (corresponding to different objects or background), and more accurate similarity can be obtained based on group-wise measurement. Here, we propose to apply ASR to build a multi-edge graph $\hat{\mathcal{G}} = \{\mathcal{V}, \hat{\mathcal{E}}\}$ to more accurately and flexi-

bly describe the relationship between images, and obtain better image classification performance.

In constructing the multi-edge graph, we apply ASR for each feature vector $\mathbf{y}_i$ of vertex v_i by treating the others as basis A. Then we obtain K representation vectors $\{\boldsymbol{\omega}_i^k\}_{k=1}^K$ and the corresponding element groups of $\mathbf{y}_i$. Here the j^{th} element in $\boldsymbol{\omega}_i^k$, $\boldsymbol{\omega}_i^k(j)$, represents the similarity between v_j and v_i w.r.t. the k^{th} feature group and we construct the edge e_{ij}^k according to $\boldsymbol{\omega}_i^k(j)$. Note that for different samples, the intrinsic group structure may be different. And thus the number of edges K between two vertices may vary.

After constructing the multi-edge graph, any graph regularized semi-supervised learning method can be employed to perform multi-label image classification [113, 112]. Since most of the methods operate on the graph adjacent matrix, which bears $2D$ structure, we also need to construct an adjacent matrix for multi-edge graph $\hat{\mathcal{G}}$ through properly selecting and merging the multiple edges. In practice, we first duplicate each edge e_{ij}^k into multiple edges, and the number of duplication is equivalent to the number of the feature elements in k^{th} group. In this way, any two nodes in $\hat{\mathcal{G}}$ are directly linked by identical number of edges. Then we adopt the Marginal Fisher Analysis (MFA) ratio [115] to evaluate the discriminative capability of the edges for multiple image class. After obtaining the edges ranking on the discriminative capability, we select the top $t = 100$ edges and combine them into single edge by summing their weights directly. The produced graph adjacent matrix is used for the semi-supervised learning on image classification.

Results

We compare the multi-edge graph with the k-NN and ℓ_1-graph [114] on the multi-label image classification task. The evaluations are performed on the public NUS-WIDE-LITE dataset [116], which consists of $55,615$ images and 81 different semantic labels. Here, we use $27,807$ images as labeled data and the remains are unlabeled as in [117]. The used 634-D feature is the concatenation of 225-D block-wise color moments (CM), 128-D wavelet texture (WT), 73-D edge direction histogram (ED),

64-D color histogram (CH) and 144-D color correlogram (CC). The k-NN graph is constructed by selecting $k = 3000$ nearest neighbors. For the ℓ_1-graph, the regularization parameter $\lambda = 0.1$ is selected from $[0.001, 10]$. In building the multi-edge graph, the parameters are fixed as $\lambda = 0.1$, $\beta = 2 \times 10^{-4}$, $\text{iter}_{\max} = 200$ and $\epsilon = 1 \times 10^{-4}$. In our experiments, it takes about 15 seconds to build the graph for one vertex on a PC with Quad CPU 2.83GHz and 8GB memory. An exemplar sub-graph of $\hat{\mathcal{G}}$ is shown in Fig. 6.3. In this sub-graph, several vertices are linked to the query vertex v_1 via 7 edges (the estimated number of groups $K = 7$), each of which measures the similarity between corresponding vertex and v_1 based on a certain feature group, as indicated in the legend. It can be seen from Fig. 6.3 that more semantically similar vertices (*e.g.,* v_2, v_4) have larger number of edges with larger weights to the vertex v_1. This is because these vertices (images) contain more similar objects to v_1, which is captured by ASR in constructing the multi-edge graph.

After duplicating the edges between two vertices into 634 edges (total dimension of adopted feature), the MFA ratio is calculated based on 20 positive and negative nearest neighbors of each vertex in $\hat{\mathcal{G}}$ [115], and top $t = 100$ edges are selected and combined. Then the popular Random Walk (RW) [113] and Entropic Graph Semi-Supervised Classification (EGSSC) [112] are used to perform semi-supervised learning on the multi-edge graph and baseline single-edge graphs. For EGSSC, the parameters are searched in the sets $\mu \in \{1 \times 10^{-8}, 1 \times 10^{-4}, 0.01, 0.1\}$ and $\nu \in \{1 \times 10^{-8}, 1 \times 10^{-6}, 1 \times 10^{-4}, 0.01, 0.1\}$ as in [112]. Classification performance is measured by the Mean Average Precision (MAP) [117] and shown in Table 6.1. It can be seen that the multi-edge graph significantly improves the multi-label image classification performance, for both two semi-supervised learning methods. In particular, compared with the state-of-the-art performance from LELR [117], the improvement achieves 3.3% for multi-edge graph + RW and 4.1% for multi-edge graph + EGSSC.

Besides, we also compare ASR with k-means + ℓ_1-graph. In particular, the elements of feature vectors are clustered into 7 groups by k-means along the feature

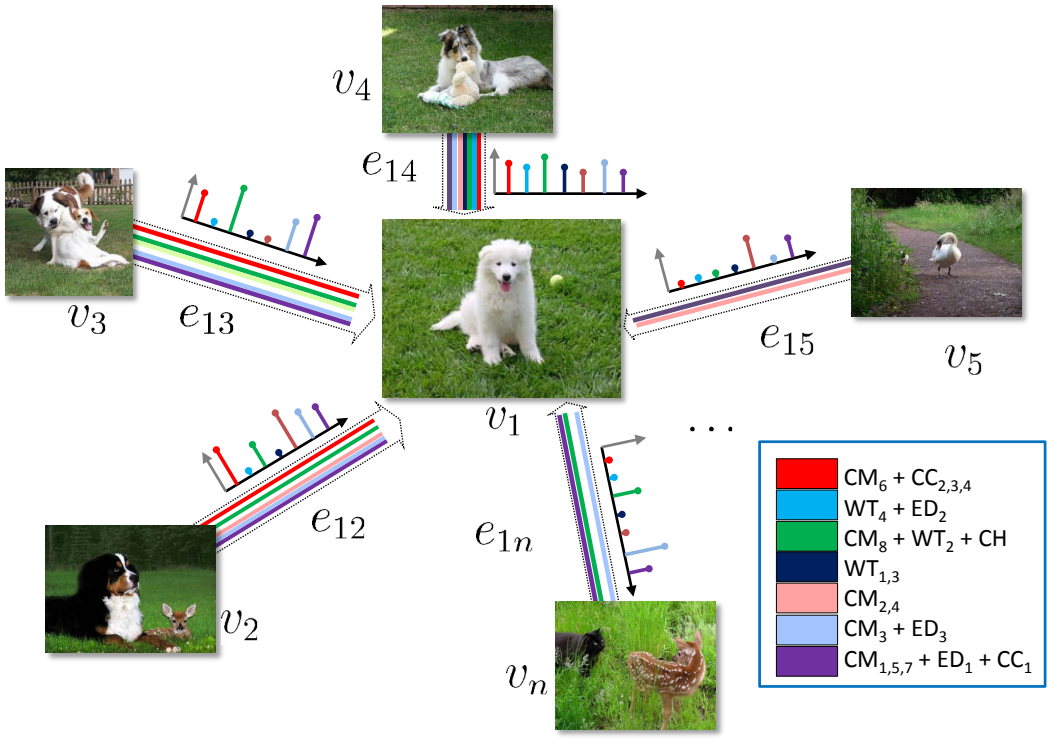


Figure 6.3: A subgraph of the constructed multi-edge graph. Here 5 types of features are used. Note that for ease of display, each type of feature is shown in groups, as indicated by the subscripts in legend. The groups of these feature elements clusters obtained by ASR are shown in legend. In the multi-edge graph, the edges' weights are shown in a histogram form.

Table 6.1: MAP (%) of label propagation on different graphs.

Graph	RW [113]	EGSSC [112]
kNN-graph	21.62	20.83
ℓ_1-graph	23.36	23.76
ℓ_1-graph_Comb	22.60	23.55
Multi-edge graph	**29.09**	**29.95**
LELR [117]	25.79	

dimension. And we construct ℓ_1-graph for each feature element cluster. These ℓ_1-graphs are then combined into a 7-edge graph ℓ_1-graph_Comb for fairly comparing with our ASR multi-edge graph. From Table 6.1, it is shown that multi-edge graph outperforms ℓ_1-graph_Comb graph by about 6% MAP. This further demonstrates that ASR's ability to find reasonable feature groups with more discriminative information, benefitting from its accordance with the intrinsic structure of features.

6.5.3 Motion Segmentation

Two-view motion segmentation

Motion segmentation is aimed to assign multiple well tracked motion trajectories to the corresponding moving rigid objects. From epipolar geometry, given two corresponding points $\mathbf{p}$ and $\mathbf{q}$ from two images $(\mathbf{p}, \mathbf{q} \in \mathbb{R}^3)$[2], they satisfy the following equation [118],

$$\mathbf{p}^T F \mathbf{q} = 0. \tag{6.9}$$

Here the fundamental matrix F encapsulates the intrinsic projective geometry between two views. Trajectories on the same object have identical fundamental matrix. And when K different rigid objects are moving independently, there are K different fundamental matrices $\{F_k\}_{k=1}^K$.

Here we apply ASR to the two view motion segmentation problem, where the tracked trajectories are only from two frames. We first rewrite the epipolar equation (6.9) for one corresponded pair as $(\mathbf{p} \otimes \mathbf{q})^T \boldsymbol{\omega} = 0$, where $\otimes$ denotes the Kronecker product and the vector $\boldsymbol{\omega}$ is formed by concatenating the columns of F. By removing the homogeneous coordinate (last element of $\boldsymbol{\omega}$) to the right hand side, the epipolar equation for N corresponding points can be written as $\mathbf{a}_i^T \boldsymbol{\omega}_k = 1$, where $\mathbf{a}_i$ consists of the first 8 elements of the vector $\mathbf{p}_i \otimes \mathbf{q}_i$. Here, we also denote the first 8 elements of original $\boldsymbol{\omega}$ as $\boldsymbol{\omega}$ without confusion. Then we stack the vectors $\mathbf{a}_i$'s into basis matrix $A = [\mathbf{a}_1, \ldots, \mathbf{a}_n]^T$ and the corresponding input vector is $\mathbf{y} = [1, \ldots, 1]^T \in \mathbb{R}^n$. Similar to the mixture regression, we can solve it through ASR

[2] Actually, the point coordinates are in the projective plane, namely $\mathbf{p}, \mathbf{q} \in \mathbb{P}^2$

Table 6.2: Segmentation errors (%) for sequences with 2 motions.

Method	GPCA [101]	RANSAC	EM	ASR
Checkerboard: 78 sequences				
Mean	11.01	12.43 ± 0.26	37.44 ± 0.58	**9.07**
Median	7.51	8.22 ± 0.93	39.26 ± 0.82	**4.13**
Traffic: 31 sequences				
Mean	**7.75**	14.60 ± 1.12	41.24 ± 0.41	9.42
Median	**1.95**	10.54 ± 2.28	42.91 ± 0.52	2.32
Articulated: 11 sequences				
Mean	16.11	20.15 ± 0.61	33.77 ± 1.27	**6.15**
Median	14.14	17.28 ± 2.51	32.37 ± 4.08	**0.99**
All: 120 sequences				
Mean	10.63	13.70 ± 0.32	38.08 ± 0.42	**8.89**
Median	6.68	9.05 ± 0.98	40.33 ± 0.60	**3.07**

as in Eqn. (6.1). Thus, we can obtain the segmentation of the motion trajectories according to their estimated fundamental matrix F_k, which is expressed as vector $\boldsymbol{\omega}_k$ in ASR.

Results

We use the Hopkins155 dataset [119] to evaluate ASR for the two-view motion segmentation task. The dataset consists of 155 video sequences of two or three motions, which are divided into three categories: checkerboard, traffic, and articulated. We use the trajectories from the first 2 frames of each sequence as the input of the two-view motion segmentation.

We compare our method with three popular motion segmentation methods. The first one is Generalized Principal Component Analysis (GPCA) [101], which first projects the data points to a 4 dimensional subspace, and then groups the estimated normal vectors of the subspaces for data segmentation. The second method is the Expectation-Maximization (EM) [100], which is widely used but only provides a *local* optimum solution. The last one is the RANdom SAmple Consensus (RANSAC) which solves model fitting problem by random data sampling and evaluation [120]. In the experiments, both the EM and RANSAC are run 20 times and their average errors are reported. For ASR, the parameters are set as $\lambda = 0$ due to the fundamental

Table 6.3: Segmentation errors (%) for sequences with 3 motions.

Method	GPCA [101]	RANSAC	EM	ASR
Checkerboard: 26 sequences				
Mean	32.27	56.02 ± 0.29	46.88 ± 1.02	**25.53**
Median	30.92	57.47 ± 0.76	47.66 ± 1.85	**20.04**
Traffic: 7 sequences				
Mean	**17.58**	48.61 ± 1.24	47.56 ± 2.19	26.48
Median	**18.54**	51.37 ± 0.98	51.31 ± 1.32	29.92
Articulated: 2 sequences				
Mean	26.14	61.70 ± 3.83	43.27 ± 5.60	**10.05**
Median	26.14	61.70 ± 3.83	43.27 ± 5.60	**10.05**
All: 35 sequences				
Mean	28.86	54.62 ± 0.32	46.81 ± 1.11	**24.83**
Median	24.32	57.07 ± 0.79	48.28 ± 1.84	**22.62**

matrix is not sparse, $\beta = 1/N^2$, $\text{iter}_{\max} = 1,000$ and $\epsilon = 1 \times 10^{-4}$. Note that here we do not compare the proposed method with multiple sample based methods, *e.g.*, sparse subspace clustering [121], since they only apply for multi-view motion segmentation. And they do not estimate the fundamental matrices since they vary across multiple frames.

The segmentation errors are provided in Table 6.2 for two motions and Table 6.3 for three motions respectively. It can be seen that ASR and GPCA significantly outperform the EM and RANSAC methods owning to their convexity. Compared with GPCA, the proposed ASR achieves smaller segmentation errors in most of the sequences, and brings 1.74% and 3.91% overall improvement for two and three motions respectively. More accurate segmentation results achieved by ASR well demonstrate its superior ability in uncovering the underlying data group structure.

6.6 Chapter Summary

In this chapter, we proposed auto-grouped sparse representation (ASR) to automatically obtain the underlying group structures of the correlated feature elements. In ASR, each uncovered group represents a certain semantically meaningful pattern. We applied a convex relaxation to the primal intractable objective function to guarantee a global solution and further introduced smooth approximations to

ease the optimization process. Furthermore, two realistic applications of ASR were considered besides the evaluations on synthetic data. For multi-label image classification, ASR achieves remarkable performance improvement over the state-of-the-art methods owing to its ability to more accurately describe the semantic relationship between images by building informative multi-edge graph. And for two view motion segmentation, ASR significantly reduces segmentation errors compared with previous methods. Our proposed ASR need include a set of pair-wise regularizations which may be inefficient for large-scale problems.

Chapter 7

Conclusions

This thesis studies low-dimensional structure learning for big data in the spirit of online learning and robust statistical learning. In this thesis, we investigate both theoretical properties and practical applications of robust structure learning in machine learning and computer vision fields. Section 1.2 gives a fairly detailed account of the contribution of this thesis. In this chapter we provide a brief overview of what we have learnt and what issues are open, and need to be addressed in future research.

7.1 Summary of Contributions

In Chapter 2-4, we addresses two important problems in low-dimensional structure learning for big data, namely, how to guarantee robustness of the inference results to the data inherent noise and outliers, and how to scale the learning algorithms to large-scale data.

We proposed a deterministic robust PCA algorithm for high-dimensional data corrupted by arbitrary outliers. The algorithm alternates between a classical PCA and decrease of weight coefficients on all the data points. To the best of our knowledge, this is the first deterministic algorithm that achieves the maximal robustness in the high-dimensional regime. This not only provides an acceleration of its random counterpart method, but also suggests a new scheme for enhancing the efficiency

of other randomness based methods by exploiting to process the samples in batch manner.

We devised an online robust PCA (online RPCA) algorithm for samples corrupted by outliers. The online RPCA alternates between standard PCA for updating PCs and probabilistic selection of the new samples which alleviates the impact of outliers. This is the first work to investigate such online robust PCA problem with theoretical performance guarantee. This work also provides a general framework for robustifying the existing online learning methods, and it gives a promising and practical solution to the challenges imposed by the modern big data analysis.

We developed the method of online optimizing the matrix norm through reformulating the nuclear norm as an explicit product of two low-rank matrices. After the reformulation, an stochastic fashion optimization algorithm is ready to apply to solve the problem, with significant efficacy gain for processing large scale data. We also provided a convergence analysis of the OR-PCA method. More importantly, OR-PCA suggests a general optimization and analysis framework of online optimizing certain matrix norm through the factorization.

In Chapter 5-6, we demonstrated two low-dimensional structure learning methods for the applications in object recognition and image classification. We provided the methods to discover the class discriminative structure and sub-group structure of the image description vectors.

We proposed the so-called geometric ℓ_p-norm pooling (GLP) method to perform feature pooling. Different from traditional feature pooling methods, *e.g.* the average and max pooling, the GLP method can utilize the geometric structural information on the feature class and spatial distributions and thus provide more discriminative pooling results. GLP is a more discriminative, flexible and information-preserving pooling method.

We proposed auto-grouped sparse representation (ASR) to automatically obtain the underlying group structures of the correlated feature elements. In ASR, each uncovered group represents a certain semantically meaningful visual pattern. Compared with traditional clustering based methods, ASR provided a more robust way

for finding the sub-group structure. ASR was used to build multi-edge graph which more accurately describes semantic relationship between images and yielded better image content annotation performance.

7.2 Open Problems and Future research

The work reported in this thesis has raised many problems to be studied in the future. We list in this section some of the immediate questions to direct future works after this thesis.

Distributed robust learning algorithms. In this thesis, we propose online learning as a major solution to scaling robust low-dimensional structure learning methods to big data. Besides online learning, another promising solution is the distributed learning. Under distributed learning framework, the data are distributed and stored in different computation nodes. The learning is executed in each node separately, and communications among different nodes are allowed. Distributed learning is efficient for processing large-scale data and has been adopted in some practical computation architectures, such as MapReduce developed in Google. However, how to design efficient communication between different nodes to guarantee the robustness of the solutions is still an open problem. In the future, we can apply the algorithmic and analysis framework proposed in Chapter 3 and Chapter 4 to the distributed robust learning algorithms design and analysis.

Applications for robust regression and classification. In this thesis, we focus on the problems of learning the low-dimensional structure of observed data. However, the developed techniques in this thesis are not restricted to these problems. Instead, the techniques can be readily extended to other linear model learning problems, among which the most important one may be the linear regression and classification. Due to the ubiquitous noises and outliers are in realistic data, robustifying the linear regression and classification are also heavily desired for making predictions on realistic data. In the future, we will investigate the regression and classification problems along this direction.

Bibliography

[44] Candes, E.J., Li, X., Ma, Y., and Wright, J. Robust principal component analysis? *ArXiv:0912.3599*, 2009.

[19] Croux, C. and Ruiz-Gazen, A. A fast algorithm for robust principal components based on projection pursuit. In *COMPSTAT: Proceedings in computational statistics*, 1996.

[20] Croux, C. and Ruiz-Gazen, A. High breakdown estimators for principal components: the projection-pursuit approach revisited. *Journal of Multivariate Analysis*, 2005.

[21] d'Aspremont, A., Bach, F., and Ghaoui, L. Optimal solutions for sparse principal component analysis. *JMLR*, 2008.

[5] Donoho, D.L. Breakdown properties of multivariate location estimators. Technical report, 1982.

[6] Donoho, D.L. High-dimensional data analysis: The curses and blessings of dimensionality. *AMS Math Challenges Lecture*, 2000.

[29] Hubert, M., Rousseeuw, P.J., and Verboven, S. A fast method for robust principal components with applications to chemometrics. *Chemometrics and Intelligent Laboratory Systems*, 2002.

[52] Hubert, M., Rousseeuw, P.J., and Branden, K.V. Robpca: a new approach to robust principal component analysis. *Technometrics*, 2005.

[30] Li, G. and Chen, Z. Projection-pursuit approach to robust dispersion matrices and principal components: primary theory and monte carlo. *Journal of the American Statistical Association*, 1985.

[32] Maronna, R.A. Robust m-estimators of multivariate location and scatter. *The annals of statistics*, 1976.

[57] Pearson, K. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 1901.

[36] Rousseeuw, P.J. Least median of squares regression. *Journal of the American statistical association*, 1984.

[37] Rousseeuw, P.J. and Leroy, A.M. *Robust regression and outlier detection*. John Wiley & Sons Inc, 1987.

[14] Schölkopf, B., Smola, A., and Müller, K.R. Kernel principal component analysis. *Artificial Neural Networks*, 1997.

[61] Xu, H., Caramanis, C., and Mannor, S. Principal component analysis with contaminated data: The high dimensional case. In *COLT*, 2010.

[16] Xu, H., Caramanis, C., and Sanghavi, S. Robust PCA via outlier pursuit. In *NIPS*, 2010.

[17] J.R. Bunch and C.P. Nielsen. Updating the singular value decomposition. *Numerische Mathematik*, 1978.

[18] E.J. Candes, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *ArXiv:0912.3599*, 2009.

[19] C. Croux and A. Ruiz-Gazen. A fast algorithm for robust principal components based on projection pursuit. In *COMPSTAT*, 1996.

[20] C. Croux and A. Ruiz-Gazen. High breakdown estimators for principal components: the projection-pursuit approach revisited. *Journal of Multivariate Analysis*, 2005.

[21] A. d'Aspremont, F. Bach, and L. Ghaoui. Optimal solutions for sparse principal component analysis. *JMLR*, 2008.

[22] J. Feng, H. Xu, and S. Yan. Robust PCA in high-dimension: A deterministic approach. In *ICML*, 2012.

[23] David Gross and Vincent Nesme. Note on sampling without replacing from a finite collection of matrices. *arXiv preprint arXiv:1001.2738*, 2010.

[24] P. Hall, D. Marshall, and R. Martin. Merging and splitting eigenspace models. *TPAMI*, 2000.

[25] Jun He, Laura Balzano, and John Lui. Online robust subspace tracking from partial information. *arXiv preprint arXiv:1109.3827*, 2011.

[26] P. Honeine. Online kernel principal component analysis: a reduced-order model. *TPAMI*, 2012.

[27] P.J. Huber, E. Ronchetti, and MyiLibrary. *Robust statistics*. John Wiley & Sons, New York, 1981.

[28] M. Hubert, P.J. Rousseeuw, and K.V. Branden. Robpca: a new approach to robust principal component analysis. *Technometrics*, 2005.

[29] M. Hubert, P.J. Rousseeuw, and S. Verboven. A fast method for robust principal components with applications to chemometrics. *Chemometrics and Intelligent Laboratory Systems*, 2002.

[30] G. Li and Z. Chen. Projection-pursuit approach to robust dispersion matrices and principal components: primary theory and monte carlo. *Journal of the American Statistical Association*, 1985.

[31] Y. Li. On incremental and robust subspace learning. *Pattern recognition*, 2004.

[32] R.A. Maronna. Robust m-estimators of multivariate location and scatter. *The annals of statistics*, 1976.

[33] S. Ozawa, S. Pang, and N. Kasabov. A modified incremental principal component analysis for on-line learning of feature space and classifier. *PRICAI*, 2004.

[34] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 1901.

[35] C. Qiu, N. Vaswani, and L. Hogben. Recursive robust pca or recursive sparse recovery in large but structured noise. *arXiv preprint arXiv:1211.3754*, 2012.

[36] P.J. Rousseeuw. Least median of squares regression. *Journal of the American statistical association*, 1984.

[37] P.J. Rousseeuw and A.M. Leroy. *Robust regression and outlier detection*. John Wiley & Sons Inc, 1987.

[38] M.K. Warmuth and D. Kuzmin. Randomized online pca algorithms with regret bounds that are logarithmic in the dimension. *JMLR*, 2008.

[39] H. Xu, C. Caramanis, and S. Mannor. Principal component analysis with contaminated data: The high dimensional case. In *COLT*, 2010.

[40] H. Zhao, P.C. Yuen, and J.T. Kwok. A novel incremental principal component analysis and its application for face recognition. *TSMC-B*, 2006.

[41] M. Artac, M. Jogan, and A. Leonardis. Incremental pca for on-line visual learning and recognition. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 3, pages 781–784. IEEE, 2002.

[42] D.P. Bertsekas. *Nonlinear programming*. Athena Scientific, 1999.

[43] Samuel Burer and Renato Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Math. Progam.*, 2003.

[44] E.J. Candes, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *ArXiv:0912.3599*, 2009.

[45] V. Chandrasekaran, S. Sanghavi, P.A. Parrilo, and A.S. Willsky. Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization*, 21(2):572–596, 2011.

[46] M. Fazel. *Matrix rank minimization with applications*. PhD thesis, PhD thesis, Stanford University, 2002.

[47] J. Feng, H. Xu, and S. Yan. Robust PCA in high-dimension: A deterministic approach. In *ICML*, 2012.

[48] D.L. Fisk. Quasi-martingales. *Transactions of the American Mathematical Society*, 1965.

[49] N. Guan, D. Tao, Z. Luo, and B. Yuan. Online nonnegative matrix factorization with robust stochastic approximation. *Neural Networks and Learning Systems, IEEE Transactions on*, 23(7):1087–1099, 2012.

[50] Jun He, Laura Balzano, and John Lui. Online robust subspace tracking from partial information. *arXiv preprint arXiv:1109.3827*, 2011.

[51] P.J. Huber, E. Ronchetti, and MyiLibrary. *Robust statistics*. John Wiley & Sons, New York, 1981.

[52] M. Hubert, P.J. Rousseeuw, and K.V. Branden. Robpca: a new approach to robust principal component analysis. *Technometrics*, 2005.

[53] Y. Li. On incremental and robust subspace learning. *Pattern recognition*, 2004.

[54] Z. Lin, M. Chen, and Y. Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv preprint arXiv:1009.5055*, 2010.

[55] Z. Lin, A. Ganesh, J. Wright, L. Wu, M. Chen, and Y. Ma. Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix. *Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, 2009.

[56] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *JMLR*, 2010.

[57] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 1901.

[58] C. Qiu, N. Vaswani, and L. Hogben. Recursive robust pca or recursive sparse recovery in large but structured noise. *arXiv preprint arXiv:1211.3754*, 2012.

[59] B. Recht, M. Fazel, and P.A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM review*, 52(3):471–501, 2010.

[60] Jasson Rennie and Nathan Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *ICML*, 2005.

[61] H. Xu, C. Caramanis, and S. Mannor. Principal component analysis with contaminated data: The high dimensional case. In *COLT*, 2010.

[62] H. Xu, C. Caramanis, and S. Sanghavi. Robust pca via outlier pursuit. *Information Theory, IEEE Transactions on*, 58(5):3047–3064, 2012.

[63] O. Boiman, E. Shechtman, and M. Irani. In defense of nearest-neighbor based image classification. In *CVPR*, 2008.

[64] A. Bosch, A. Zisserman, and X. Muñoz. Image classification using random forests and ferns. In *ICCV*, 2007.

[65] Y. Boureau, J. Ponce, and Y. LeCun. A theoretical analysis of feature pooling in visual recognition. In *ICML*, 2010.

[66] R. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 1936.

[67] K. Fukushima and S. Miyake. Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position. *Pattern Recognition*, 1982.

[68] J. Gemert, J. Geusebroek, C. Veenman, and A. Smeulders. Kernel codebooks for scene categorization. In *ECCV*, 2008.

[69] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical report, California Institute of Technology, 2007.

[70] D. Hubel and T. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J Physiol*, 1962.

[71] P. Jain, B. Kulis, and K. Grauman. Fast image search for learned metrics. In *CVPR*, 2008.

[72] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun. What is the best multi-stage architecture for object recognition? In *ICCV*, 2009.

[73] S. Lazebnik and M. Raginsky. Supervised learning of quantizer codebooks by information loss minimization. *TPAMI*, 2009.

[74] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.

[75] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel. Handwritten digit recognition with a back-propagation network. In *NIPS*, 1989.

[76] F. Li, J. Carreira, and C. Sminchisescu. Object recognition as ranking holistic figure-ground hypotheses. In *CVPR*, 2010.

[77] F. Li, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In *CVPR workshop*, 2004.

[78] F. Li and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *CVPR*, 2005.

[79] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004.

[80] H. Nakayama, T. Harada, and Y. Kuniyoshi. Global gaussian approach for scene categorization using information geometry. In *CVPR*, 2010.

[81] B. Ni, S. Yan, and A. Kassim. Contextualizing histogram. In *CVPR*, 2009.

[82] N. Pinto, D. Cox, and J. DiCarlo. Why is real-world visual object recognition hard. *PLoS Computational Biology*, 2008.

[83] M. Ranzato, Y. Boureau, and Y. LeCun. Sparse feature learning for deep belief networks. In *NIPS*, 2007.

[84] T. Serre, L., and T. Poggio. Object recognition with features inspired by visual cortex. In *CVPR*, 2005.

[85] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *CVPR*, 2010.

[86] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, and S. Lin. Graph embedding and extensions: A general framework for dimensionality reduction. *TPAMI*, 2007.

[87] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, 2009.

[88] L. Yang, R. Jin, R. Sukthankar, and F. Jurie. Unifying discriminative visual codebook generation with classifier training for object category reorganization. In *CVPR*, 2008.

[89] H. Zhang, A. Berg, M. Maire, and J. Malik. Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In *CVPR*, 2006.

[90] Fei-Fei, L., Perona, P.: A bayesian hierarchical model for learning natural scene categories. In: CVPR. (2005)

[91] Perronnin, F., Sánchez, J., Mensink, T.: Improving the fisher kernel for large-scale image classification. In: ECCV. (2010)

[92] Yang, J., Yu, K., Huang, T.: Efficient highly over-complete sparse coding using a mixture model. ECCV (2010)

[93] Li, F., Carreira, J., Sminchisescu, C.: Object recognition as ranking holistic figure-ground hypotheses. In: CVPR. (2010)

[94] Song, Z., Chen, Q., Huang, Z., Hua, Y., Yan, S.: Contextualizing object detection and classification. In: CVPR. (2011)

[95] Wright, J., Yang, A., Ganesh, A., Sastry, S., Ma, Y.: Robust face recognition via sparse representation. TPAMI (2008)

[96] Yuan, X., Yan, S.: Visual classification with multi-task joint sparse representation. In: CVPR. (2010)

[97] Wright, J., Ma, Y., Mairal, J., Sapiro, G., Huang, T., Yan, S.: Sparse representation for computer vision and pattern recognition. P. IEEE (2010)

[98] Elhamifar, E., Vidal, R.: Sparse subspace clustering. In: CVPR. (2009)

[99] Liu, D., Yan, S., Rui, Y., Zhang, H.: Unified tag analysis with multi-edge graph. In: MM. (2010)

[100] Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the em algorithm. J Roy. Stat. Soc. B Met. (1977)

[101] Vidal, R., Ma, Y., Sastry, S.: Generalized principal component analysis (gpca). TPAMI (2005)

[102] Gaffney, S., Smyth, P.: Trajectory clustering with mixtures of regression models. In: KDD. (1999)

[103] Quadrianto, N., Caetano, T., Lim, J., Schuurmans, D.: Convex relaxation of mixture regression with efficient algorithms. In: NIPS. (2009)

[104] Hocking, T., Vert, J., Bach, F., Joulin, A.: Clusterpath: an algorithm for clustering using convex fusion penalties. In: ICML. (2011)

[105] Shen, X., Huang, H.: Grouping pursuit through a regularization solution surface. J Am. Stat. Assoc. (2010)

[106] Vert, J., Bleakley, K.: Fast detection of multiple change-points shared by many signals using group lars. In: NIPS. (2010)

[107] Nesterov, Y.: Smooth minimization of non-smooth functions. Math. Program. (2005)

[108] Ng, A., Jordan, M., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: NIPS. (2001)

[109] Boyd, S., Vandenberghe, L.: Convex optimization. Cambridge Univ. Pr. (2004)

[110] Duchi, J., Shalev-Shwartz, S., Singer, Y., Chandra, T.: Efficient projections onto the ℓ_1-ball for learning in high dimensions. In: ICML. (2008)

[111] Tseng, P.: On accelerated proximal gradient methods for convex-concave optimization. submitted to SIAM J Optimiz. (2008)

[112] Subramanya, A., Bilmes, J.: Entropic graph regularization in non-parametric semi-supervised classification. In: NIPS. (2009)

[113] Zhu, X., Ghahramani, Z.: Learning from labeled and unlabeled data with label propagation. Tech. Rep. (2002)

[114] Yan, S., Wang, H.: Semi-supervised learning by sparse representation. In: SDM. (2009)

[115] Yan, S., Xu, D., Zhang, B., Zhang, H., Yang, Q., Lin, S.: Graph embedding and extensions: A general framework for dimensionality reduction. TPAMI (2007)

[116] Chua, T., Tang, J., Hong, R., Li, H., Luo, Z., Zheng, Y.: Nus-wide: a real-world web image database from national university of singapore. In: CIVR. (2009)

[117] Chen, X., Yuan, X., Chen, Q., Yan, S., Chua, T.: Multi-label visual classification with label exclusive context. In: ICCV. (2011)

[118] Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge Univ. Press (2000)

[119] Tron, R., Vidal, R.: A benchmark for the comparison of 3-d motion segmentation algorithms. In: CVPR. (2007)

[120] Fischler, M., Bolles, R.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM (1981)

[121] Elhamifar, E., Vidal, R.: Sparse subspace clustering. In: CVPR. (2009)

[122] Niyogi, X.: Locality preserving projections. In: NIPS. (2004)

[123] Tenenbaum, Joshua B., De Silva, Vin, Langford, John C. A global geometric framework for nonlinear dimensionality reduction. Science (2000).

[124] Roweis, Sam T., Saul, Lawrence K. Nonlinear dimensionality reduction by locally linear embedding. Science (2000).

[125] Belkin, Mikhail, Niyogi, Partha. Laplacian eigenmaps and spectral techniques for embedding and clustering. In: NIPS (2001).

[126] Fisher, Ronald A. The use of multiple measurements in taxonomic problems. Annals of eugenics (1936).

[127] Chen, Yudong and Caramanis, Constantine and Mannor, Shie Robust Sparse Regression under Adversarial Corruption. In: ICML (2013)

[128] Van der Vaart, A.W. *Asymptotic statistics*, volume 3. Cambridge university press, 2000.

[129] Crammer, K. Online tracking of linear subspaces. *Learning Theory*, pp. 438–452, 2006.

[130] Bonnans, J Frédéric and Shapiro, Alexander. Optimization problems with perturbations: A guided tour. *SIAM review*, 40(2):228–264, 1998.

[131] Hale, E.T., Yin, W., and Zhang, Y. Fixed-point continuation for \ell_1-minimization: Methodology and convergence. *SIAM Journal on Optimization*, 19(3):1107–1130, 2008.

List of Publications

1. **Jiashi Feng**, Zhouchen Lin, Huan Xu, Shuicheng Yan. Robust Subspace Segmentation with Laplacian Constraint. Submitted to *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 2014.

2. **Jiashi Feng**, Stefanie Jegelka, Trevor Darrell, Huan Xu, Shuicheng Yan. Learning Discriminative Scalable Attributes from Sample Relatedness. Submitted to *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 2014.

3. **Jiashi Feng**, Huan Xu, Shuicheng Yan. Stochastic Optimization for Robust PCA. *Neural Information Processing Systems (NIPS)* 2013.

4. **Jiashi Feng**, Huan Xu, Shie Mannor, Shuicheng Yan. Online PCA for Big Contaminated Data. *Neural Information Processing Systems (NIPS)* 2013.

5. Canyi Lu, **Jiashi Feng**, Shuicheng Yan. Correlation Adaptive Subspace Segmentation by Trace Lasso. *International Conference on Computer Vision (ICCV)*, 2013.

6. **Jiashi Feng**, Huan Xu, and Shuicheng Yan. Robust PCA in High-dimension: A Deterministic Approach. *International Conference on Machine Learning (ICML)*, 2012.

7. **Jiashi Feng**, Xiaotong Yuan, Zilei Wang, Huan Xu, Shuicheng Yan. Auto-grouped Sparse Representation for Visual Analysis. *European Conference on Computer Vision (ECCV)*, 2012.

8. **Jiashi Feng**, Bingbing Ni, Qi Tian, and Shuicheng Yan. Geometric ℓ_p-norm Feature Pooling for Image Classification. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

9. **Jiashi Feng**, Bingbing Ni, Dong Xu, and Shuicheng Yan. Histogram Contextualization. *IEEE Transaction on Image Processing (TIP)*, 21(2): 778-788, 2011.

10. **Jiashi Feng**, Xiaotong Yuan, Zilei Wang, Huan Xu, Shuicheng Yan. Auto-grouped Sparse Representation for Visual Analysis. Submitted to *IEEE Transactions on Image Processing (TIP)* .

11. Jian Dong, Wei Xia, Qiang Chen, **Jiashi Feng**, Zhongyang Huang, Shuicheng Yan. Subcategory-Aware Object Classification. *Computer Vision and Pattern Recognition (CVPR)* 2013.

12. Zilei Wang, **Jiashi Feng**. Multi-class Learning from Class Proportions. *Neurocomputing*, 119(7), 273-280, 2013.

13. Congyan Lang, **Jiashi Feng**, Guangcan Liu, Jinghui Tang, Shuicheng Yan, Jiebo Luo. Improving Bottom-up Saliency Detection by Looking into Neighbors. *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, 23(6), 1016 - 1028, 2013.

14. Zilei Wang, **Jiashi Feng**, Shuicheng Yan, Hongsheng Xi. Linear Distance Coding for Image Classification. *IEEE Transaction on Image Processing (TIP)*, 22(2): 534-548, 2013.

15. Zilei Wang, **Jiashi Feng**, Shuicheng Yan, Hongsheng Xi. Image Classification Via Object-aware Holistic Superpixel Selection. Accepted by *IEEE Transactions on Image Processing (TIP)*, 2013.

16. Saining Xie, **Jiashi Feng**, Shuicheng Yan, Hongtao Lu. DP^{3}s: Dual Perception Preserving Projections. *British Machine Vision Conference (BMVC)*, 2013.(Oral)

17. Wei Xia, Zheng Song, **Jiashi Feng**, Loong Fah Cheong, and Shuicheng Yan. Segmentation over Detection by Coupled Global and Local Sparse Representations. *European Conference on Computer Vision (ECCV)*, 2012.

18. Xiaobai Liu, **Jiashi Feng**, Shuicheng Yan, Liang Lin, and Hai Jin. Segment an Image by Looking into an Image Corpus. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

19. Xiaobai Liu, **Jiashi Feng**, and Shuicheng Yan. Image Segmentation with Patch-pair Density Prior. *ACM International Conference on Multimedia (MM)*, 2010.

20. Congyan Lang, **Jiashi Feng**, Songhe Feng, Shuicheng Yan, Qi Tian. Dual Low Rank Pursuit: Salient Features for Saliency Detection. Submitted to *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*.

21. Jian Dong, Qiang Chen, **Jiashi Feng**, Zhongyang Huang, Shuicheng Yan. Subcategory-aware Object Classification. Submitted to *IEEE Transactions on Image Processing (TIP)*.

22. **Jiashi Feng**, Yuzhao Ni, Jian Dong, Zilei Wang, and Shuicheng Yan. Purposive Hidden-Object-Game: Embedding Human Computation in Popular Game. *IEEE Transaction on Multimedia (TMM)*, 14(5): 1496 - 1507, 2012.

23. **Jiashi Feng**, Bingbing Ni, and Shuicheng Yan. Auto-generate Professional Background Music for Homemade Videos. *ACM International Conference on Internet Multimedia Computing and Service (ICIMCS)*, 2010. (Best paper candidate)

24. **Jiashi Feng**, Yantao Zheng, and Shuicheng Yan. Towards a Universal Detector by Mining Concepts with Small Semantic Gaps. *ACM International Conference on Multimedia (MM)*, 2010.

25. Si Liu, **Jiashi Feng**, Zheng Song, Tianzhu Zhang, Hanqing Lu, Changsheng Xu, and Shuicheng Yan. "Hi, Magic Closet, Tell Me What to Wear!". *ACM*

Conference on Multimedia (MM), 2012.

26. Si Liu, Tam V. Nguyen, **Jiashi Feng**, Meng Wang, Shuicheng Yan: "Hi, magic closet, tell me what to wear!". *ACM Conference on Multimedia (MM)*, 2012. (Best Demo Award).

27. Ruchir Srivastava, **Jiashi Feng**, Sujoy Roy, Terence Sim, and Shuicheng Yan. Don't Ask Me What I'm Like, Just Watch and Listen. *ACM Conference on Multimedia (MM)*, 2012.

28. Jian Dong, Yuzhao Ni, **Jiashi Feng**, and Shuicheng Yan, Purposive hidden-object game (P-HOG) towards imperceptible human computation, *ACM International Conference on Multimedia (MM)*, 2011.

29. Yuzhao Ni, Jian Dong, **Jiashi Feng**, and Shuicheng Yan, Purposive hidden-object-game: embedding human computation in popular game, *ACM International Conference on Multimedia (MM)*, 2011.

30. Zheng Wang, **Jiashi Feng**, and Shuicheng Yan. Learning to Rank Tags. *ACM International Conference on Image and Video Retrieval (CIVR)*, 2010.

31. Si Liu, **Jiashi Feng**, Csaba Domokos, Junshi Huang, Zhenzhen Hu, Shuicheng Yan. Segment Skirt from "Red Skirt". Accepted by *IEEE Transactions on Multimedia (TMM)*.

32. Tam V. Nguyen, **Jiashi Feng**, Shuicheng Yan. Seeing Human Weight from a Single RGB-D Image. Submitted to *ACM Transactions on Intelligent Systems and Technology (TIST)*.

33. Jian Dong, **Jiashi Feng**, Ju Sun, Shuicheng Yan. Towards Automatic 3D Glasses Removal in Interactive Media. Submitted to *ACM Transactions on Intelligent Systems and Technology (TIST)*.