UNIVERSITY OF CALIFORNIA,
IRVINE

Guaranteed and Efficient Learning Algorithms for Ranking and Matrix Factorization

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Computer Science

by

Niranjan Uma Naresh

Dissertation Committee:
Professor Gopi Meenakshisundaram, Chair
Professor Nader Bagherzadeh
Assistant Professor Shuang Zhao

2016
DEDICATION

Family, Friends, Nature
# TABLE OF CONTENTS

| LIST OF FIGURES | vi |
| LIST OF TABLES | vii |
| ACKNOWLEDGMENTS | viii |
| CURRICULUM VITAE | ix |
| ABSTRACT OF THE DISSERTATION | xii |

## 1 Inductive Pairwise Ranking
1.1 Introduction ................................................. 3
1.1.1 Related Work and Background ........................... 4
1.2 Model and Algorithm ........................................... 7
1.2.1 Notation and Preliminaries ............................... 7
1.2.2 Feature Low Rank Model ................................. 8
1.2.3 Algorithm ................................................... 12
1.3 Analysis ....................................................... 14
1.4 Experimental Results ......................................... 18
1.4.1 Synthetic Simulations .................................... 18
1.4.2 Real-data Simulations .................................... 20
1.5 Discussion and Future Directions ........................... 22
1.6 Proofs ......................................................... 23
1.6.1 Proof of Proposition 1.1 ................................. 23
1.6.2 Proof of Proposition 1.2 ................................. 23
1.6.3 Proof of Proposition 1.3 ................................. 23
1.6.4 Proof of Lemma 1.1 ....................................... 24

## 2 Robust Pairwise Ranking
2.1 Introduction ................................................... 26
2.1.1 Related Work .............................................. 27
2.2 Problem Setup ................................................ 30
2.2.1 Notation .................................................... 30
2.2.2 Characterization of the Adversary ....................... 31
2.3 Fully Observed Adversarial Setting ......................... 34
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.3.1</td>
<td>Algorithm</td>
<td>34</td>
</tr>
<tr>
<td>2.3.2</td>
<td>Analysis</td>
<td>34</td>
</tr>
<tr>
<td>2.4</td>
<td>Partially Observed Adversarial Setting</td>
<td>39</td>
</tr>
<tr>
<td>2.5</td>
<td>Generalization to Other Models</td>
<td>41</td>
</tr>
<tr>
<td>2.6</td>
<td>Experiments</td>
<td>43</td>
</tr>
<tr>
<td>2.7</td>
<td>Conclusion</td>
<td>45</td>
</tr>
<tr>
<td>3</td>
<td>Streaming Robust PCA</td>
<td>46</td>
</tr>
<tr>
<td>3.1</td>
<td>Introduction</td>
<td>48</td>
</tr>
<tr>
<td>3.1.1</td>
<td>Our Contribution</td>
<td>48</td>
</tr>
<tr>
<td>3.1.2</td>
<td>Related work</td>
<td>48</td>
</tr>
<tr>
<td>3.2</td>
<td>Problem Setup</td>
<td>51</td>
</tr>
<tr>
<td>3.2.1</td>
<td>Model</td>
<td>51</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Notations and Assumptions</td>
<td>51</td>
</tr>
<tr>
<td>3.2.3</td>
<td>Algorithm</td>
<td>52</td>
</tr>
<tr>
<td>3.3</td>
<td>Analysis</td>
<td>55</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Convergence of the Innermost Loop</td>
<td>56</td>
</tr>
<tr>
<td>3.3.2</td>
<td>Concentration Properties in the Middle Loop</td>
<td>59</td>
</tr>
<tr>
<td>3.3.3</td>
<td>Convergence of the Outermost Loop</td>
<td>61</td>
</tr>
<tr>
<td>3.4</td>
<td>Conclusion and Future Work</td>
<td>65</td>
</tr>
<tr>
<td>4</td>
<td>Inductive Robust PCA</td>
<td>66</td>
</tr>
<tr>
<td>4.1</td>
<td>Introduction</td>
<td>67</td>
</tr>
<tr>
<td>4.1.1</td>
<td>Robust Inductive Learning: Motivation</td>
<td>68</td>
</tr>
<tr>
<td>4.1.2</td>
<td>Related Work</td>
<td>69</td>
</tr>
<tr>
<td>4.1.3</td>
<td>Our Contributions</td>
<td>71</td>
</tr>
<tr>
<td>4.2</td>
<td>Problem Setup</td>
<td>73</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Notation and Preliminaries</td>
<td>73</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Assumptions</td>
<td>74</td>
</tr>
<tr>
<td>4.2.3</td>
<td>Sample Complexity</td>
<td>76</td>
</tr>
<tr>
<td>4.2.4</td>
<td>Algorithm</td>
<td>76</td>
</tr>
<tr>
<td>4.2.5</td>
<td>Computational Complexity</td>
<td>77</td>
</tr>
<tr>
<td>4.3</td>
<td>Analysis</td>
<td>79</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Proof Outline</td>
<td>79</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Symmetric Noiseless Case</td>
<td>80</td>
</tr>
<tr>
<td>4.3.3</td>
<td>Symmetric Noisy Case</td>
<td>83</td>
</tr>
<tr>
<td>4.3.4</td>
<td>Asymmetric Case</td>
<td>84</td>
</tr>
<tr>
<td>4.4</td>
<td>Experiments</td>
<td>86</td>
</tr>
<tr>
<td>4.4.1</td>
<td>Synthetic Simulations</td>
<td>86</td>
</tr>
<tr>
<td>4.4.2</td>
<td>Real-data Simulations</td>
<td>87</td>
</tr>
<tr>
<td>4.5</td>
<td>Conclusion</td>
<td>89</td>
</tr>
<tr>
<td>4.6</td>
<td>Proofs: Noisy Case</td>
<td>90</td>
</tr>
<tr>
<td>4.6.1</td>
<td>Proof of Theorem 4.2</td>
<td>90</td>
</tr>
<tr>
<td>4.6.2</td>
<td>Proof of Lemma 4.3</td>
<td>90</td>
</tr>
<tr>
<td>4.6.3</td>
<td>Proof of Lemma 4.4</td>
<td>91</td>
</tr>
</tbody>
</table>
4.7 Proofs: Asymmetric Case .................................................. 93
4.7.1 Proof of Claim 4.1 .............................................. 93

Bibliography ................................................................. 95
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Ranking results of LRPR and IPR: fixing $d = 20$ and $K = 50 \lceil d^2 \log^2(n)/n^2 \rceil$ while varying $m$.</td>
<td>18</td>
</tr>
<tr>
<td>1.2</td>
<td>Ranking results of LRPR and IPR: fixing $d = 20$ and $m = \lceil d^2 \log(n) \rceil$ while varying $K$.</td>
<td>18</td>
</tr>
<tr>
<td>1.3</td>
<td>Ranking results of LRPR and IPR: fixing $K = 50 \lceil d^2 \log^2(n)/n^2 \rceil$ and $m = \lceil d^2 \log(n) \rceil$ while varying $d$.</td>
<td>19</td>
</tr>
<tr>
<td>1.4</td>
<td>Ranking results of LRPR and IPR on Sushi and Car datasets: we fix $m = \lceil d^2 \log(n) \rceil$ while varying $K$; we fix $K = 50 \lceil d^2 \log^2(n)/n^2 \rceil$ while varying $m$.</td>
<td>20</td>
</tr>
<tr>
<td>2.1</td>
<td>Robust recovery results of the BTL model: we fix $\nu = 2$ and vary $d$ in the left plot; we fix $d = 100$ and vary $nu$ in the left plot.</td>
<td>43</td>
</tr>
<tr>
<td>4.1</td>
<td>Comparing of RPCA algorithms in terms of running time to reach a solution of a given accuracy. For $n = 1000$, we vary each problem parameter while fixing the others. Specifically, we vary:</td>
<td>83</td>
</tr>
<tr>
<td>4.2</td>
<td>Comparison of robust PCA algorithms on the MovieLens data: running time and recovery error.</td>
<td>88</td>
</tr>
</tbody>
</table>
# LIST OF TABLES

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Comparison of this work to previous works in non-active pairwise ranking: extended-BTL model is due to [7], chest-blade model is due to [9] and low-rank model is due to [48]. Here, <em>pairs</em> and <em>per</em> denote the state-of-the-art bounds known regarding the number of pairs compared and the number of comparisons per pair respectively. Note that, we consider the practically important regime of $d \ll n$.</td>
<td>4</td>
</tr>
<tr>
<td>4.1</td>
<td>Comparison of this work to previous robust PCA works. For simplicity and brevity, we let $n_1 = n_2 = n$ and $d_1 = d_2 = d$; let the number of non-zeros per row/column of $S^<em>$ be $z$ and the number of non-zero entries in $S^</em>$ be $m$; we use $\tilde{O}$ to suppress log factors. Note that we consider the practically important regime of $d \ll n$.</td>
<td>68</td>
</tr>
</tbody>
</table>
ACKNOWLEDGMENTS

Thanks to my advisor and my defense committee members. Thanks to Xerox Research Centre India and the University of California at Irvine for their support. Thanks to my family and friends. Nature is awe-inspiring.
CURRICULUM VITAE

Niranjan Uma Naresh

EDUCATION

Doctor of Philosophy in Computer Science 2011–2016
University of California Irvine

Master of Science in Computer Science 2011–2013
University of California Irvine

Bachelor of Technology in Information Technology 2007–2011
National Institute of Technology Karnataka

RESEARCH EXPERIENCE

Graduate Student Researcher 2011–2016
University of California Irvine

TEACHING EXPERIENCE

Teaching Assistant 2011–2013, 2016
University of California Irvine

WORK EXPERIENCE

Research Intern 2016
Xerox Research Centre India

Research Intern 2014
Microsoft Research India
REFEREED PUBLICATIONS

1. Provable Inductive Robust PCA via Iterative Hard Thresholding. 2017
   U N Niranjan, A Rajkumar, T Tulabandhula. *Submitted to AISTATS.*

2. Provably Robust Pairwise Ranking. 2017
   U N Niranjan, A Rajkumar. *Submitted to AAAI.*

3. Inductive Pairwise Ranking: Going Beyond the $n \log n$ Barrier. 2017
   U N Niranjan, A Rajkumar. *Submitted to AAAI.*

4. Perception-optimal Low-rank Approximation for Image Compression. 2017
   U N Niranjan, M Gopi. *Submitted to ICASSP.*

5. Streaming Robust PCA. 2017
   U N Niranjan, Y Shi. *Submitted to SODA.*

6. Tensor Contractions with Extended BLAS Kernels on CPU and GPU. 2016
   Y Shi, U N Niranjan, A Anandkumar, C Cecka. *HiPC.*

   A Anandkumar, P Jain, Y Shi, U N Niranjan. *AISTATS.*

8. FEAST at Play: Feature ExtrAction using Score function Tensors. 2015
   H Sedghi, M Janzamin, U N Niranjan, A Anandkumar. *NIPS Workshop on Feature Extraction.*

   L Li, J Snyder, I Pelaschier, J Huang, U N Niranjan, P Duncan, M Rupp, K R Müller, K Burke. *Int J of Quantum Chemistry.*

    F Huang, U N Niranjan, M U Hakeem, A Anandkumar. *JMLR.*

11. Non-convex Robust PCA. 2014
    P Netrapalli, U N Niranjan, S Sanghavi, A Anandkumar, P Jain. *NIPS.*

13. Puzzhull: Cavity and protrusion hierarchy to fit conformal polygons. 2017
S Mistry, U N Niranjan, M Gopi. GD/SPM.

Awards and Honors

Finalist, Adobe PhD Fellowship. 2016
Travel Award, NIPS. 2014
Chair’s Fellowship, CS Department, UC Irvine. 2011–2013

Programming Skills

MATLAB, C, C++, Linux, \LaTeX, R, Python, SQL, HTML, CUDA, CULA, Eigen.

Talks

XRCl.

Robust Tensor Decomposition. 2015
NIPS.

Support Vector Machines. 2012
UCI.

Toeplitz Operators. 2010
NITK.

Other Accomplishments

Concert Violinist
Specialization in Carnatic Classical Music.
ABSTRACT OF THE DISSERTATION

Guaranteed and Efficient Learning Algorithms for Ranking and Matrix Factorization

By

Niranjan Uma Naresh

Doctor of Philosophy in Computer Science

University of California, Irvine, 2016

Professor Gopi Meenakshisundaram, Chair

In this dissertation, two central problems in computer science are considered:

1. ranking \( n \) items from pairwise comparisons focusing on -
   (a) handling adversarial noise
   (b) incorporating feature information

2. robust matrix factorization algorithms focusing on -
   (a) space-efficient computation
   (b) using feature information

Motivated by several open theoretical and practical questions, novel solutions to the above problems are explored in detail; to be specific, the contributions are summarized below:

(1) Ranking:

(Part-1a) In the presence of adversarial noise, many popular ranking algorithms, such as maximum likelihood and rank centrality, for estimating standard models, fail to work well. Robustifying many existing models such as the binary choice models, an algorithm is devised

\footnote{It is to be noted that this PhD thesis is \textit{not} based on publications 6, 7, 8, 10, 11 and 12.}

xii
to provably recover an $\epsilon$-accurate ranking with high probability while allowing as large as $O(n)$ grossly corrupted comparison results per item.

(Part-1b) Typically, items have associated features. Subsuming several models including the classic Bradley-Terry-Luce and Thurstone models, a feature-based low-rank model is proposed and characterized. In learning this model, the $O(n \log n)$ sorting barrier is provably improved upon due to the proposed sample-efficient polynomial time algorithm that systematically uses features and feature correlations. Via empirical benchmarking, it is shown that the proposed approach consistently outperforms the state-of-the-art algorithms that do not use feature information.

(2) Matrix factorization:

(Part-2a) While principal component analysis and its variants are very well-studied, existing algorithms are either space-efficient or noise-tolerant; in contrast, incorporating both these aspects in solving eigen-problems, a space-efficient noise-tolerant algorithm with finite sample guarantees is developed and analyzed in the framework of the perturbed spiked covariance model under weaker assumptions than previous works; a key differentiating aspect from previous works is that the algorithm is streaming. Moreover, the derived computational and statistical guarantees are near optimal (up to logarithmic factors).

(Part-2b) While robust subspace identification methods, both convex and non-convex, are well-studied, they are often oblivious to utilizing feature information that is available in practice; in contrast to existing (feature-oblivious) convex and non-convex approaches, a simple iterative method with a fast (linear) convergence property is proposed and studied - a key aspect that differentiates this from previous works is how side information is incorporated in achieving better robust subspace recovery.
Chapter 1

Inductive Pairwise Ranking

Abstract

We study the problem of ranking \( n \) items from non-actively chosen pairwise comparisons when the items have associated \( d \)-dimensional feature vectors. We propose and characterize a very broad class of preference matrices giving rise to the Feature Low Rank (FLR) model, which subsumes several models ranging from the classic Bradley–Terry–Luce (BTL) [4] and Thurstone [51] models to the recently proposed blade-chest [9] and generic low-rank preference [48] models. Using the technique of matrix completion in the presence of side information, we develop the Inductive Pairwise Ranking (IPR) algorithm to provably learn a good ranking under the FLR model. Specifically, we prove that our algorithm needs to make only \( \Omega(\max(\log n/\epsilon^2, d^4 \log^3 n/\epsilon^3 n^2)) \) pairwise comparisons chosen uniformly at random to output an \( \epsilon \)-close ranking with high probability. We note that when \( d \ll n \), which is often the case in practice, our approach gives a significant gain in the sample complexity as compared to the \( \Omega(n \text{ poly log}(n)) \) pairwise comparisons needed by previous results to obtain similar guarantees. In practice, through systematic synthetic simulations, we confirm our
theoretical findings regarding improvements sample complexity by using feature information. Moreover, on popular real-world preference learning datasets, with as less as 10% sampling of the pairwise comparisons, our method recovers a good ranking.
1.1 Introduction

Ranking from pairwise comparisons or preferences is an ubiquitous problem in machine learning, statistics and theoretical computer science. In the so-called non-active setting, one is given comparison results of $m$ pairs pre-selected from among all pairs of $n$ items where each pair is compared at least $K$ times. Particularly, the learner does not get to choose which pairs are to be compared. The goal is then to estimate a suitable ordering of the items, using the observed comparison results, that conforms to the true ordering, assuming one exists, up to the desired error $\epsilon$. In practice, we often have side information associated with the items that need to be ranked – such a scenario is referred to as the inductive setting. Motivated by this, we wish to leverage the available side information to compute an ordering more efficiently than existing techniques.

**Our Contributions:** To the best of our knowledge, our work is the first to derive a provable and efficient method for ranking in the non-active inductive setting. Our novelty and technical contributions can be summarized along the following axes:

1. **Model:** We generalize existing models so that we can incorporate (a) features, and (b) feature correlations associated with the items to be ranked. We show that our model subsumes many existing and popular ranking models.

2. **Algorithm:** Our algorithm uses two key subroutines namely, (a) noisy inductive matrix completion, and (b) approximate pairwise ranking algorithm [12].

3. **Guarantee:** We derive the guarantee that our algorithm obtains, with high probability, an $\epsilon$-accurate recovery using $\Omega(\max(\log n/\epsilon^2, d^4 \log^3 n/\epsilon^6 n^2))$ independent pairwise comparisons chosen uniformly at random.

4. **Experiments:** We substantiate our theoretical results by demonstrating sample com-
Table 1.1: Comparison of this work to previous works in non-active pairwise ranking: extended-BTL model is due to [7], chest-blade model is due to [9] and low-rank model is due to [48]. Here, \textit{pairs} and \textit{per} denote the state-of-the-art bounds known regarding the number of pairs compared and the number of comparisons per pair respectively. Note that, we consider the practically important regime of \(d \ll n\).

<table>
<thead>
<tr>
<th>Model</th>
<th>(F)</th>
<th>(w)</th>
<th>(W)</th>
<th>\textit{pairs}</th>
<th>\textit{per}</th>
</tr>
</thead>
<tbody>
<tr>
<td>BTL</td>
<td>(F = I)</td>
<td>(w \in \mathbb{R}^n)</td>
<td>(W = 0)</td>
<td>(\Omega(n \log n))</td>
<td>(\Omega(\log n))</td>
</tr>
<tr>
<td>Item-feature</td>
<td>(F \in \mathbb{R}^{d \times n})</td>
<td>(w \in \mathbb{R}^d)</td>
<td>(W = 0)</td>
<td>many</td>
<td>many</td>
</tr>
<tr>
<td>Chest-blade</td>
<td>(F = I)</td>
<td>(w \in \mathbb{R}^n)</td>
<td>(\text{rank}(W) = O(d))</td>
<td>many</td>
<td>many</td>
</tr>
<tr>
<td>Low-rank</td>
<td>(F = I)</td>
<td>(w \in \mathbb{R}^n)</td>
<td>(W \in \mathbb{R}^{n \times n})</td>
<td>(\Omega(nr \log n))</td>
<td>(\Omega(r \log n))</td>
</tr>
<tr>
<td>This work</td>
<td>(F \in \mathbb{R}^{d \times n})</td>
<td>(w \in \mathbb{R}^d)</td>
<td>(W \in \mathbb{R}^{d \times d})</td>
<td>(\Omega(d^2 \log n))</td>
<td>(\Omega(d^2 \log^2 n/n^2))</td>
</tr>
</tbody>
</table>

We would like to emphasize upfront that it is the sole focus of this chapter to study the practically motivated regime of \(d \ll n\) in detail. Furthermore, we note that our sample complexity results do not violate the standard \(\Omega(n \log n)\) lower bounds for comparison-based sorting algorithms since we develop an algorithm that effectively ranks in the \feature space\ rather than the \item space\.

1.1.1 Related Work and Background

We now give a brief overview of relevant work in ranking models followed by a brief background regarding tools from inductive matrix completion theory which will be crucial in proving our sample complexity bounds.

**Ranking Models** : In the simplest terms, the ranking problem involves estimating the \best\ ordering items according to some observed preferences. A early thread of ranking literature has its beginnings in economics involving choice models [36]. A certain deterministic version of the ranking problem is also studied as the \sorting\ problem which is central in theoretical computer science.

1. **Random Utility (RU) models**: Starting with the seminal work of [4], the Bradley-Terry-
Luce (BTL) model has become a landmark model for ranking. In the vanilla version of this model, the probability that item $i$ beats item $j$ is given by $P_{ij} = \frac{w_i}{w_i + w_j}$ where $w \in \mathbb{R}^n_+$ is the parameter vector to be estimated from data; the $i^{th}$ entry in $w$ denotes the score associated with item $i$. Thurstone [51] model is also a well-known statistical model; here, $P_{ij} = \Phi(s_i - s_j)$ where $\Phi$ is the standard normal Cumulative Distribution Function (CDF) and $s \in \mathbb{R}^n$ is the score vector. These classic models fall under the so-called Random Utility (RU) Models [37].

2. **Item Feature (IF) models:** Extending the BTL model, statistical models that utilize side information are presented in [7]. Recently, [9] presented the blade-chest ranking model which studied the stochastic intransitive setting. Their algorithm involves regularized maximum likelihood estimation for which tight sample complexity properties are not known. Despite the above works, to the best of our knowledge, there are no known models utilizing feature information while have provable sample-efficient algorithms for estimation and ranking.

3. **Low Rank (LR) models:** Recently, [48] – unifying classic models such as BTL and Thurstone models – defined a generic class of preference matrices which have low rank under transformations involving suitable link functions. Upon such a transformation, connections of the ranking problem to matrix completion theory become clear. Subsequently, they use well-known matrix completion results to derive sample complexity guarantees for ranking. However, their model does not utilize side information that may be available.

We position the present work in relation to the above relevant previous works in Table 1.1. This list is by no means exhaustive; while there exist several other ranking methods (eg, ranking-SVM[30]), there are no known sample complexity guarantees associated with these.

**Inductive Matrix Completion:** The matrix completion task [6] is to fill-in the missing entries of a partially observed matrix, which is possible efficiently under a low-rank assum-
tion on the underlying matrix. Oftentimes, side information may be available which further makes this task potentially easier. This is the Inductive Matrix Completion (IMC) problem which is formally defined as the optimization problem, $\hat{Z} = \arg\min_Z \ell((A^T Z B)_{ij}, M_{ij})$ where $A \in \mathbb{R}^{d_1 \times n_1}$ and $B \in \mathbb{R}^{d_2 \times n_2}$ are known feature matrices, $Z \in \mathbb{R}^{d_1 \times d_2}$ is a rank-$r$ unknown latent parameter matrix, $(i, j) \in \Xi \subseteq [n] \times [n]$ is the support set corresponding to the (uniformly sampled) observed entries and $\ell$ is any loss function, the squared loss being the most commonly chosen one. Once the estimate $\hat{Z}$ is obtained using the training set indexed by $\Xi$, predictions may then be performed as $\hat{M}_{ij} = (A^T \hat{Z} B)_{ij}$ for any $(i, j) \in \Xi^c$. The known solution techniques with recovery guarantees are:

1. **Non-convex algorithm (via alternating minimization):** This approach entails parameterizing $Z = UV^\top$ and performing alternating projected least squares updates on $U$ and $V$. The tightest known guarantee for this approach involves a sample complexity of $\Omega(d^2 r^3 \kappa^2 \log(d))$ and a convergence rate of $O(\log(1/\epsilon))$ [24].

2. **Convex relaxation (via trace-norm formulation):** This approach entails relaxing the rank constraints to a nuclear norm penalty. Existence of a unique optimum can be shown with high probability [54] and is characterized a sample complexity of $\Omega(dr \log(d) \log(n))$. Despite the non-smoothness, a sub-gradient descent algorithm provably converges with a rate of $O(1/\sqrt{\epsilon})$ [27]. Noisy features are handled in [10].
1.2 Model and Algorithm

1.2.1 Notation and Preliminaries

General notation: Unless stated otherwise, we use lower-case letters for scalars, upper-case letters for universal constants, lower-case bold-face letters for vectors and upper-case bold-face letters for matrices; specifically, $P$ denotes a preference matrix. For any matrix $M \in \mathbb{R}^{a \times b}$, let $\|M\|_\infty = \max_{i,j} |M_{ij}|$, $\|M\|_* = \sum_{i=1}^{\min\{a,b\}} \sigma_i(M)$ where $\sigma_i(M)$ are the singular values of $M$ and $\|M\|_F = \sqrt{\sum_{i=1}^{a} \sum_{j=1}^{b} M_{ij}^2}$. $I$ denotes the identity matrix whose dimensions would be implied from the context; similarly, depending on the context, $\mathbf{0}$ denotes a vector or matrix of zeros of the appropriate dimension. Next, let $P_{\min} = \min_{i \neq j} P_{ij}$ and $\Delta = \min_{i \neq j} |\psi(P_{ij}) - \psi(1/2)|$. Let $\Xi$ be the support set of the observed entries of a matrix and let $m = |\Xi|$. Define projection of a matrix on the support set $B = R_{\Xi}(A)$ as: $B_{ij} = A_{ij}$ if $(i, j) \in \Xi$ and $B_{ij} = 0$ if $(i, j) \notin \Xi$.

Items and features: Let $n$ be the number of items to be ranked. Let $S_n$ denote the symmetric group on $n$ items. Let each item have a $d$-dimensional feature vector associated with it, ie, $f_i \in \mathbb{R}^d, \forall i \in [n]$; concatenating these, we obtain the feature matrix $F = [f_1, \ldots, f_n] \in \mathbb{R}^{d \times n}$.

Link functions: Any $\psi : [0, 1] \to \mathbb{R}$ which a strictly increasing bijective function is a valid link function. For example, $\psi$ could be the logit function, which is the inverse of the sigmoid function, defined as, $\psi(x) := \log \left( \frac{x}{1-x} \right)$ for $x \in [0, 1]$; another example is the probit function defined as $\psi(x) = \Phi^{-1}(x)$ where $\Phi$ is the standard normal CDF. When we apply the link function to a matrix, we mean that the transformation applied entry-wise.

Preference matrices: Let $\mathcal{P}_n := \{P \in [0, 1]^{n \times n}\}^{P_{ij} + P_{ji} = 1}$ denote the set of all pairwise preference matrices over $n$ items. Let the set of stochastic-transitive matrices be $\mathcal{P}_n^{ST} := \{P \in \mathcal{P}_n | P_{ij} > 1/2, P_{jk} > 1/2 \implies P_{ik} > 1/2\}$ and the set of stochastic-intransitive matrices be $\mathcal{P}_n^{SI} := \{P \in \mathcal{P}_n | P_{ij} > 1/2, P_{jk} > 1/2 \implies P_{ik} < 1/2\}$. Let $\mathcal{P}_n^{RU}$ be the set of preference matrices associated with unary random utility models (which are described in
the next section). Let $\mathcal{P}_n^{IF} := \{ P \in \mathcal{P}_n | P_{ij} = \psi^{-1}(w^\top(f_i - f_j)) \}$ for some $w \in \mathbb{R}^d$.

Let $r \leq n$. Define the set of preference matrices having rank-$r$ under the link function $\psi$ as $\mathcal{P}_n(\psi, r) := \{ P \in \mathcal{P}_n | \text{rank} (\psi(P)) \leq r \}$. Next, define the set of preference matrices having rank-$r$ under the link function $\psi$ with the associated feature matrix $A \in \mathbb{R}^{d \times n}$ as $\mathcal{P}_n(\psi, r, A) := \{ P \in \mathcal{P}_n(\psi, r) | \psi(P) = A^\top L A \}$ where $L \in \mathbb{R}^{d \times d}$ is an unknown rank-$r$ latent matrix (which is a function of the parameters of the ranking model) and $A = [a_1, \ldots, a_n] \in \mathbb{R}^{d \times n}$ is the known feature matrix whose $i^{th}$ column is the feature vector corresponding to the $i^{th}$ item. Let $\kappa = \sigma_{\min}(A)/\sigma_{\max}(A)$ be the inverse condition number of the feature matrix $A$. Let $i \succ_P j$ iff $P_{ij} > 1/2$. Denoting the indicator function by $1$, we define the distance between a permutation $\sigma \in S_n$ and a preference matrix $P \in \mathcal{P}_n$ as:

$$\text{dist}\,(\sigma, P) := \left( \begin{array}{c} n \\ 2 \end{array} \right)^{-1} \sum_{i<j} 1 ((i \succ_P j) \land (\sigma(i) \succ \sigma(j)))$$

$$+ \left( \begin{array}{c} n \\ 2 \end{array} \right)^{-1} \sum_{i<j} 1 ((j \succ_P i) \land (\sigma(j) \succ \sigma(i)))$$

Note that the above distance measure essentially counts the fraction of pairs on which $\sigma$ and $P$ disagree, and can be thought of as a normalized $0 - 1$ loss function.

**1.2.2 Feature Low Rank Model**

Random Utility (RU) models, arising in discrete choice theory, dating back to [37], characterize the probability of an item $i$ beating item $j$, $P_{ij}$, using a prior on the (latent) score associated with those items, $w_i \in \mathbb{R}$ and $w_j \in \mathbb{R}$. The most popular pairwise ranking models including BTL and Thurstone models fit in this framework. In particular, it is well-known
that if $w_i \sim \text{Gumbel}(0, 1)$, we obtain the BTL model; for completeness, we justify it below:

$$P_{ij} = \Pr(w_i > w_j) = \Pr(w_i - w_j > 0) = \frac{\xi_1 e^{-(w_i - w_j)}}{1 + e^{-(w_i - w_j)}}$$

where $\xi_1$ follows from the fact that the difference of two independent standard Gumbel distributed random variables follows the standard logistic distribution. Similarly, if $w_i \sim \mathcal{N}(0, 1)$, we obtain the Thurstone model. The underlying commonality in these models is the simple observation that the prior distribution is on the scores, which are unary terms. Notably, the recent result by [48] shows that under the inverse transformation of the CDF of the difference of the latent score variables, the preference probability matrix is low-rank for BTL and Thurstone models. Further, they extended this result to a broader class of low-rank models in which the preference matrices are low-rank when the link function is set to be this inverse CDF.

One angle of motivation for this chapter stems from the intuitive thought that the scores associated with an item $i$ in RU models can be generalized to functions involving, not just unary terms but also, pairwise terms, ie, the score of item $i$ with respect to item $j$ is given by an energy function $E_{ij}$ that has a bilinear form. From this point onwards, for simplicity, we detail the generalization of the RU models encompassing the BTL model, ie, we posit that $E_{ij}$ has a standard Gumbel distribution and consequently, we choose the link function $\psi$ to be the logit function. It is noteworthy that our results will hold under any link function for the corresponding prior.

We now propose the energy-based generative model, which we call Feature Low Rank (FLR) model, defined via the preference matrix specified as follows:

$$P_{ij} = \frac{e^{-E_{ij}}}{e^{-E_{ij}} + e^{-E_{ji}}}$$

(1.1)

Here, we define the energy function associated with the pair of items $(i, j)$ to be of the form
Subroutine 1 IMC: Inductive Matrix Completion

Input: $M_{ij}$ for $(i, j) \in \Xi \subseteq [n] \times [n]$, feature matrix $F$.
Output: Completed matrix \( \hat{M} \).

1: Solve the convex program:
\[
\hat{Z} = \arg \min_{Z} \| \mathcal{R}_\Xi (M - F^T L) ||^2_F \quad \text{s.t.} \quad \| Z \|_* \leq C_L
\]

2: return $\psi(\hat{P}) \leftarrow F^T \hat{Z} F$.

\[
E_{ij} := f_i^T w + f_i^T W f_j \quad \text{where} \quad w \in \mathbb{R}^d \quad \text{and} \quad W \in \mathbb{R}^{d \times d} \quad \text{are the unknown latent parameters (vector and matrix parameters repectively) to be estimated, and} \quad f_i \quad \text{and} \quad f_j \quad \text{are the known feature vectors associated with items} \quad i \quad \text{and} \quad j \quad \text{respectively. It is clear from Equation (1.1) that a key advantage of the proposed model is the additional ability to incorporate side information in the form of feature vectors and feature correlations in a latent space described by} \quad W. \quad \text{In matrix notation,}
\]
\[
\psi(P) = (1 g^T + F^T W^T F) - (g 1^T + F^T W F)
\]
\[
= (\Sigma V^T)^T L (\Sigma V^T)
\]

where $g := F^T w$ is column vector in $\mathbb{R}^n$, $1 \in \mathbb{R}^n$ is the all-ones column vector, $F = U \Sigma V^T$ is the full SVD of $F$ (such that $U \in \mathbb{R}^{d \times d}$, $V \in \mathbb{R}^{n \times n}$ are orthonormal matrices with $\Sigma \in \mathbb{R}^{d \times n}$ as the $d \times d$ diagonal matrix of singular values padded with zeros) and $L := U^T (1 w^T - w 1^T + W^T - W) U$ (such that $\Sigma_i^{-1} = \sigma^{-1}_i$ and $\Sigma_i^{-1}$ = 0 if $i \neq j$). It is now clear that the sufficient condition for $P \in \mathcal{P}_n(\psi, r, \Sigma V^T)$ is that $\text{rank}(L) \leq r$. Now, we describe the generality of the FLR model in Equation (1.1) by showing that it subsumes many existing models and has much more expressiveness.

**Proposition 1.1.** The LR model is a special case of the FLR model, ie, $\mathcal{P}_n(\psi, r, \Sigma V^T) \subseteq \mathcal{P}_n(\psi, r, A)$.

**Corollary 1.1.** Let $F = I$ and $\psi$ be the logit link function. From Proposition 1.1, it is easy to see the following special cases from Equation (1.1).

10
Subroutine 2 PR: Pairwise Ranking (Copeland Procedure)

**Input:** Preference matrix $\mathbf{M} \in \mathbb{R}^{n \times n}$.

**Output:** Ranking $\hat{\sigma}$.

1. **Threshold:** $\forall (i,j), \quad \tilde{M}_{ij} \leftarrow 1(M_{ij} > 1/2)$.
2. Compute row-sum of $\tilde{\mathbf{M}}$: $\mathbf{v} \leftarrow \tilde{\mathbf{M}} \mathbf{1}$.
3. **return** $\hat{\sigma} \leftarrow \text{Sort}(\mathbf{v})$.

1. Let $\mathbf{W} = \mathbf{xy}^\top$. If $\mathbf{w} = 0$, then $\mathbf{P} \in \mathcal{P}_n(\psi, 2)$. If $\mathbf{w} \neq 0$, then $\mathbf{P} \in \mathcal{P}_n(\psi, 4)$.
2. If $\mathbf{W}$ is symmetric, then $\mathbf{W} - \mathbf{W}^\top = 0$ and hence $\mathbf{P} \in \mathcal{P}_n(\psi, 2)$.
3. Let $\mathbf{A}$ be a diagonal $r \times r$ matrix; let $\{\mathbf{X}, \mathbf{Y}\} \in \mathbb{R}^{n \times r}$ be orthonormal matrices. If $\mathbf{W} = \mathbf{X}\mathbf{A}_{r \times r} \mathbf{Y}^\top + \mathbf{M}$ where $\mathbf{M}$ is a symmetric matrix, then $\psi(\mathbf{P}) \in \mathcal{P}_n(\psi, 2r + 2)$.

**Proposition 1.2.** The unary RU models are special cases of the FLR model, ie, $\mathcal{P}_n^{RU} \subseteq \mathcal{P}_n(\psi, r, A)$.

**Corollary 1.2.** The BTL and Thurstone models are obtained as special cases of the FLR model under the logit and the probit transormations of $\mathbf{P}$ respectively. This follows from Proposition 1.1 (or Corollary 1.1-part (1)) above together with Propositions 6 and 7 of [48].

**Proposition 1.3.** Regression-based models with item-specific features in [7] are special cases of the FLR model, ie, $\mathcal{P}_n^{IF} \subseteq \mathcal{P}_n(\psi, r, \mathbf{A})$.

**Corollary 1.3.** Let $d \ll n$. Then we recover the blade-chest model [9] as a special case of the FLR model by setting $\text{rank}(\mathbf{W}) = O(d)$ and $\mathbf{w} = 0$. Next, when $d \geq n$, it is clear from Theorem 1 of [9] that such preference matrices degenerate into matrices in $\mathcal{P}_n(\psi, n, \mathbf{A})$ where $\psi$ is the logit function. Moreover, it is easy to see that the FLR model admits both stochastic-transitive and stochastic-intransitive preference matrices.

Due to space constraints, proofs of Propositions 1.1, 1.2 and 1.3 are given in the appendix. To summarize, we have shown how to instantiate several previously proposed ranking models as special cases of our FLR model in Table 1.1.
Algorithm 3 IPR: Inductive Pairwise Ranking

**Input:** Set of comparison results $S = \{(i,j, \{y_{ij}^k\})\}$, feature matrix $F$, link function $\psi$, target rank $r$.

**Output:** Ranking of $n$ items, $\sigma \in S_n$.

1: Construct the partially observed empirical preference matrix using $S$ as:

\[
\hat{P}_{ij} = \begin{cases} 
\frac{1}{K} \sum_{k=1}^{K} y_{ij}^k & \text{if } (i,j) \in \Xi \\
\frac{1}{K} \sum_{k=1}^{K} (1 - y_{ij}^k) & \text{if } (j,i) \in \Xi \\
\frac{1}{2} & \text{if } i = j \text{ or } (i,j) \notin \Xi
\end{cases}
\]

2: Compute SVD of $F = U\Sigma V^T$ and set $A \leftarrow \Sigma V^T$.
3: Use a noisy inductive matrix completion subroutine: $\psi(\hat{P}) \leftarrow \text{IMC}(\psi(\hat{P}), A)$.
4: Take the inverse inductive matrix completion subroutine: $Q \leftarrow \psi^{-1}(P_r(\psi(\hat{P})))$.
5: Using a pairwise ranking subroutine: $\hat{\sigma} \leftarrow \text{PR}(Q)$.
6: return $\hat{\sigma}$.

1.2.3 Algorithm

We present our main algorithm for inductive ranking in Algorithm 3. The input data consist of the set of pairwise comparison results $S = \{(i,j, \{y_{ij}^k\})\}$, $(i,j) \in \Xi \subseteq [n] \times [n]$, $k \in [K]$, $y_{ij}^k \in \{0,1\}$ and the feature matrix $F \in \mathbb{R}^{d \times n}$. The algorithm assumes the link function and the rank as input parameters. The subroutines used are:

1. **Noisy matrix completion with features (Subroutine 1):** Note that to solve our ranking problem and derive the associated recovery guarantee, it suffices, as we have done, to use the specified trace-norm program as a black-box method; hence, we assume that we have access to an oracle that gives us the solution to the convex program. The details of how the solution to this program may be found numerically is beyond the scope of this work – for further details regarding some possible sub-gradient algorithms, we refer the reader to [10] and [27].

2. **$\gamma$-approximate pairwise ranking procedure (Subroutine 2):** Let $\hat{\sigma} \in S_n$ be the output of any Pairwise Ranking (PR) procedure with respect to an underlying prefer-
ence matrix $\mathbf{P}$. For a constant $\gamma > 1$, $\hat{\sigma}$ is said to be $\gamma$-approximate if $\text{dist}(\hat{\sigma}, \mathbf{P}) \leq \gamma \min_{\sigma \in \mathcal{S}_n} \text{dist}(\sigma, \mathbf{P})$. Any constant factor approximate ranking procedure may be used. Specifically, we use the Copeland procedure [12] as a black-box method which has a 5-approximation guarantee [13]. This method involves simply sorting the items according to a score which is computed for every item $i$ as $\sum_{j=1}^{n} \mathbb{1}(\overline{P}_{ij} > 1/2)$. 
1.3 Analysis

In this section, we state and prove our main result.

**Theorem 1.1** (Guaranteed rank aggregation with sub-linear sample complexity using item features). Let \( P \in P_n(\psi, r, A) \) be the true underlying preference matrix according to which the pairwise comparison dataset \( S = \{(i, j, \{y^k_{ij}\})\} \) is generated. Let \( \psi \) be \( L \)-Lipschitz in \([P_{\min}^2, 1 - P_{\min}^2]\). Let \( \Xi \) be the set of pairs of items compared such that the number of pairs compared is \( |\Xi| = m > \frac{48C^2d^2\log(n)(1+\gamma)^2}{\kappa^8\epsilon^2\Delta^4} \) where \( \Xi \) is chosen uniformly at random from among all possible subsets of item pairs of size \( m \). Let each pair in \( \Xi \) be compared independently \( K \geq \frac{16(1+\gamma)mL^2\log(n)}{n^2\Delta^2\epsilon} \) times where \( \Delta = \min_{i \neq j} |\psi(P_{ij}) - \psi(1/2)| \). Then, with probability at least \( 1 - 3/n^3 \), Algorithm 3 returns an estimated permutation \( \hat{\sigma} \) such that \( \text{dist}(\hat{\sigma}, P) \leq \epsilon \).

**Remark 1.1.** The key take-away message in Theorem 1.1 is the reduction in sample complexity possible due to efficient utilization of features and feature correlations, associated with the items to be ranked, by Algorithm 3. For instance, when \( d = O(1) \), which is often the case in practice, we reduce the required total number of comparisons to be made to \( \Omega(\log(n)) \). Thus, we achieve a very significant gain since the total number of comparisons is logarithmic as opposed to quadratic in the number of items. This is especially crucial in large-scale machine learning applications.

**Remark 1.2.** Another point to be noted from Theorem 1.1 is that, under the uniform sampling assumption, when features associated with items are known, it is more important that we compare sufficient (precisely, \( \Omega(\log(n)) \)) number of different pairs rather than high number of comparisons per pair. Furthermore, he total number of comparisons needed in Theorem 1.1 is given by the product \( mK \) which is \( \Omega(\max(\log n/\epsilon^2, d^4\log^3 n/\epsilon^3 n^2)) \).

We now present the proof of Theorem 1.1. We shall prove the theorem under the Bernoulli sampling model (where each entry of an \( n \times n \) matrix is observed independently with prob-
ability $1/n^2$) rather than the uniform sampling model (wherein $\Xi$ is chosen uniformly at random from among all possible subsets of item pairs of size $m$); the equivalence between the two is well-known (see, for instance, Section 7.1 of [5]).

**Proof.** Let $\hat{P}_{ij}$ be the empirical probability estimate of $P_{ij}$. Note that we compute $\hat{P}_{ij} = \frac{1}{K} \sum_{k=1}^{K} y_{ij}^k$ for $(i, j) \in \Xi$ from the given pairwise comparison dataset, $S = \{(i, j, \{y_{ij}^k\})\}$. From Equation (1.2), $\psi(P) = A^\top L A$ where $A = \Sigma V^\top$. Since we use the empirical estimate for $P_{ij}$, we have noise due to sampling error only over $\Xi$, ie, $\psi(\hat{P}) = \psi(P) + N = A^\top L A + N$ where

$$|N_{ij}| = \begin{cases} 0 & \text{if } (i, j) \notin \Xi \\ |\psi(\frac{1}{K} \sum_{k=1}^{K} y_{ij}^k) - \psi(P_{ij})| & \text{if } (i, j) \in \Xi \end{cases}$$

Now, we solve the trace-norm regularized convex program corresponding to the noisy inductive matrix completion problem:

$$\{\bar{L}, \bar{N}\} = \arg \min_{Z_N, Z_L} \left\| R_{\Xi}(\psi(\hat{P}) - (A^\top L A + Z_N)) \right\|_F^2 + \lambda_L \left\| Z_L \right\|_* + \lambda_N \left\| Z_N \right\|_*$$

and let $\psi(\bar{P}) = A^\top \bar{L} A + \bar{N}$ be the link-transformed completed (estimate) matrix where $\bar{N}$ be the estimated noise matrix. This is equivalent to solving the problem:

$$\{\bar{L}, \bar{N}\} = \arg \min_{Z_N, Z_L} \left\| R_{\Xi}(\psi(\hat{P}) - (A^\top L A + Z_N)) \right\|_F^2$$

s.t. $\left\| Z_L \right\|_* \leq C_L$, $\left\| Z_N \right\|_* \leq C_N$

We set $C_N = 0$ and $C_L = \left\| (A^\top)^{\dagger} \psi(\hat{P}) (A)^{\dagger} \right\|_*$ which may be upper bounded, by Lemma 3 of [10] as $C_L \leq \frac{d}{C' \kappa^4}$ for a constant $C'$. We now recall Theorem 1 from [10]. Let $\delta < 1/d$ and $A$ be well-conditioned, specifically, $\kappa_A^{4} \leq C_2 d$ for constant $C_2$. The expected squared loss under
Bernoulli sampling is bounded as, with probability at least $1 - \delta$:

$$
\frac{\|\psi(\bar{\mathbf{P}}) - \psi(\hat{\mathbf{P}})\|_F^2}{n^2} \leq C_1 \min \left( C_N \sqrt{\frac{\log(2n)}{m}}, \sqrt{C_N \frac{\sqrt{n}}{m}} \right)
+ \frac{C_2 d \kappa^4}{\kappa^4} \sqrt{\frac{\log(2/\delta)}{m}}
$$

(1.3)

where $C_1$ and $C_2$ are constants. By triangle inequality,

$$
\|\psi(\bar{\mathbf{P}}) - \psi(\hat{\mathbf{P}})\|_F = \|\psi(\bar{\mathbf{P}}) - (\psi(\mathbf{P}) + \mathbf{N})\|_F
\geq \|\psi(\bar{\mathbf{P}}) - \psi(\mathbf{P})\|_F - \|\mathbf{N}\|_F
$$

Using $C_N = 0$ in Equation (1.3), with probability at least $1 - \delta$,

$$
\frac{1}{n} \|\psi(\bar{\mathbf{P}}) - \psi(\mathbf{P})\|_F \leq \left( \frac{C_2 d}{\kappa^4} \sqrt{\frac{\log(2/\delta)}{m}} \right)^{1/2} + \frac{1}{n} \|\mathbf{N}\|_F
$$

Let $K \geq \frac{mL^2 \log(n)}{\gamma^2}$ where $\tau = n \sqrt{\frac{\epsilon}{1 + \gamma} \Delta}$. Substituting the bounds for the $\mathbf{N}$ terms from Lemma 1.1 and using the union bound, with probability at least $1 - \delta - 1/n^3$,

$$
\|\psi(\bar{\mathbf{P}}) - \psi(\mathbf{P})\|_F \leq n \left( \frac{C_2 d}{\kappa^4} \sqrt{\frac{\log(2/\delta)}{m}} \right)^{1/2} + \tau
\leq n \left( \frac{C_2 d}{\kappa^4} \sqrt{\frac{\log(2/\delta)}{m}} \right)^{1/2} + n \sqrt{\frac{\epsilon}{1 + \gamma} \frac{\Delta}{4}}
$$

Now, setting $m > \frac{16C_2^2 d^2 \log(2/\delta)(1+\gamma)^2}{\epsilon^2 \Delta^4}$ and $\delta = 2/n^3$, we obtain, with probability $1 - 3/n^3$,

$$
\|\psi(\bar{\mathbf{P}}) - \psi(\mathbf{P})\|_F \leq n \sqrt{\frac{\epsilon}{1 + \gamma} \frac{\Delta}{4}} + n \sqrt{\frac{\epsilon}{1 + \gamma} \frac{\Delta}{4}}
= n \sqrt{\frac{\epsilon}{1 + \gamma} \frac{\Delta}{2}}
$$

Then using similar arguments as proof of Theorem 13 in [48], we obtain our result. □
Lemma 1.1 (Characterization of noise due to finite-sample effects). Under the conditions of Theorem 1.1, let $m$ item pairs be compared such that the number of comparisons per item pair is $K \geq \frac{mL^2 \log(n)}{\tau^2}$. Then, with probability at least $1 - 1/n^3$, $\|N\|_F \leq \tau$. 
Figure 1.1: Ranking results of LRPR and IPR: fixing $d = 20$ and $K = 50 \lceil d^2 \log^2(n)/n^2 \rceil$ while varying $m$.

Figure 1.2: Ranking results of LRPR and IPR: fixing $d = 20$ and $m = \lceil d^2 \log(n) \rceil$ while varying $K$.

1.4 Experimental Results

In this section, we conduct a systematic empirical investigation of the performance of our ranking method and justify our theoretical claim in the previous section. The goal of this study is two-fold: (a) to verify the correctness of our algorithm, and (b) to show that by using features and feature correlations, our IPR algorithm has a better sample complexity thereby improving upon the LRPR algorithm that does not take into account the available side information.

1.4.1 Synthetic Simulations

For a given set of $n = 500$ items, we consider three main problem parameters:
Figure 1.3: Ranking results of LRPR and IPR: fixing $K = 50[d^2 \log^2(n)/n^2]$ and $m = [d^2 \log(n)]$ while varying $d$.

1. $m$ – the number of item pairs compared (Figure 1.1).
2. $K$ – the number of comparisons per pair (Figure 1.2).
3. $d$ – the dimensionality of features (Figure 1.3).

We study the performance of both IPR and LRPR algorithms by varying each of the problem parameters while fixing the others. We note that by making use of side information, IPR outperforms LRPR in all the cases as shown in the sample complexity plots. All the accuracy results presented are obtained by averaging over five runs.

Data generation: We consider three representative preference matrices derived from Equation (1.1): (a) Model-1: we set $W = 0$, (b) Model-2: we construct a general $W$; here, we generate $W_{ij} \sim U(0,1)$, and (c) Model-3: we construct a low-rank $W$, ie, rank($W$) = 2 < $d$; here we generate $W_{ij} \sim U(0,1)$ and then truncating $W$ by setting all but its top two singular values to zero. In all the three models, we generate $w_i \sim U(0,1)$. The features are generate as $F_{ij} \sim U(0,1)$; to ensure that the features are well-conditioned, we perform the full SVD of feature matrix $F$ and set all its singular values to 1.

Parameter settings: For IPR, we choose $\lambda_L = 10^{-2}$ and $\lambda_N = 10^2$. Note that LRPR allows for the rank of the problem to be automatically determined. In the same spirit, though Step-5 of Algorithm 3 requires the knowledge of the true rank, we choose not to perform this
Figure 1.4: Ranking results of LRPR and IPR on Sushi and Car datasets: we fix $m = \lceil d^2 \log(n) \rceil$ while varying $K$; we fix $K = 50 \lceil d^2 \log^2(n)/n^2 \rceil$ while varying $m$.

truncation step thereby including the error induced by the smaller singular values resulting from noise due to sampling in our distance estimate – even then, IPR outperforms LRPR.

1.4.2 Real-data Simulations

We apply our method on two popular preference learning datasets. We briefly describe the data and the results (Figure 1.4) we obtain below:

1. *Sushi:* This data [31] was collected by surveying the preferences of 5000 customers. Each customer orders 10 sushi dishes according to their preferences. The goal, then,
is to estimate a global ranking of these sushi dishes using these observations from customers. Each sushi has six features such as price, taste and so on. We construct the complete preference matrix $P \in [0, 1]^{10 \times 10}$ using the preferences of all the customers and consider this to be ground truth preference matrix. By systematically varying $m$ and $K$, we noticed that IPR succeeds consistently as compared to LRPR. An interesting observation was that, over five runs of the algorithms, IPR gets two out of the top four sushi dishes right most of the times namely, ‘amaebi’ and ‘ikura’; on the other hand, LRPR does not succeed in recovering these always.

2. Car: The task in this dataset [1] is find an order of preference among ten cars. This data was collected by surveying 60 customers regarding their preferences among pairs of cars drawn from the set of ten cars. Each car has four features including engine, transmission and so on. We construct the ground truth preference matrix $P \in [0, 1]^{10 \times 10}$ by aggregating the the pairwise preferences of all the customers. By systematically varying $m$ and $K$, we noticed that IPR performs quite well when compared to LRPR. An interesting trend we found was that customers generally preferred sedans over SUVs and non-hybrid vehicles over hybrid vehicles.
1.5 Discussion and Future Directions

In this chapter, we have proposed and characterized the FLR model together with the guaranteed IPR algorithm that utilizes available side information of the items to be ranked in the form of features and feature correlations to provably reduce the sample complexity for ranking from $\Omega(n \log n)$ to possibly as low as $\Omega(\log n)$ – this is an important advantage of our model over previous models since, oftentimes, $d \ll n$ in practice. Some future research directions include:

1. It is of interest to see if mixture models for ranking such the mixture of BTL models and the recently proposed topic modeling based approach [16] could fit into a similar feature-based framework while admitting sample-efficient estimation algorithms.
2. Higher-order models involving comparisons of $\nu$-tuples of items for $\nu > 2$ are of interest, especially, in applications such as ranking in multi-party games.
3. Models and guaranteed algorithms for ranking with non-independent comparisons, including user-specific features, are also of interest.
1.6 Proofs

1.6.1 Proof of Proposition 1.1

Proof. We prove this by showing that every \( P \in \mathcal{P}_n(\psi, r) \) is in \( \mathcal{P}_n(\psi, r, A) \) but not the other way around. By the definition of a preference matrix corresponding to the LR model, if \( P \in \mathcal{P}_n(\psi, r) \), then \( \text{rank}(\psi(P)) \leq r \). Similarly, for the FLR model, if \( P \in \mathcal{P}_n(\psi, r, A) \), then \( \psi(P) = A^\top L A \) and \( \text{rank}(\psi(P)) \leq r \); in other words, \( \text{rank}(L) \leq r \). Now setting \( A = I \), we have \( \mathcal{P}_n(\psi, r) = \mathcal{P}_n(\psi, r, A) \). On the other hand, if \( A \neq I \), we have \( \mathcal{P}_n(\psi, r) \subsetneq \mathcal{P}_n(\psi, r, A) \). \( \square \)

1.6.2 Proof of Proposition 1.2

Proof. Let \( w \) be the unary score vector in RU models. The result then follows by setting the energy function of item \( i \) with respect to item \( j \) in the FLR model to be the unary score corresponding to item \( i \) in the RU model, ie, by simply setting \( F = I \) and \( W = 0 \) which leads to \( E_{ij} = w_i \). \( \square \)

1.6.3 Proof of Proposition 1.3

Proof. This is immediate by setting \( W = 0 \). For concreteness, we choose \( \psi \) to be the logit link function. Setting \( W = 0 \) in Equation (1.1), we obtain

\[
P_{ij} = \frac{e^{-w^\top f_i}}{e^{-w^\top f_i} + e^{-w^\top f_j}} \tag{1.4}
\]

Observe that \( \psi(P_{ij}) = w^\top f_j - w^\top f_i \). Writing this in matrix notation, \( \psi(P) = 1w^\top F - F^\top w1^\top \). Note that \( \psi(P) \in \mathbb{R}^{n \times n} \) is a rank-2 skew-symmetric matrix. Suppose that \( L := \)
\((V\Sigma^{-1})^T\psi(P)V\Sigma^{-1}) = U^T(1w^T - w1^T)U\). Now, note that \(L \in \mathbb{R}^{d \times d}\) is also a rank-2 skew-symmetric matrix. Thus, \(P \in \mathcal{P}_n(\text{logit, } 2, \Sigma V^T)\) since \(\psi(P) = (\Sigma V^T)^T L (\Sigma V^T)\). In addition, note that the FLR model accounts for feature correlations when \(W \neq 0\). □

### 1.6.4 Proof of Lemma 1.1

**Proof.** For any support \(\Xi\), define the following event:

\[
\mathcal{G}_\Xi := \left( \left| \hat{P}_{ij} - P_{ij} \right| < \frac{P_{\text{min}}}{2} \quad \forall (i, j) \in \Xi \right)
\]

By Hoeffding’s bound, \(\Pr(\mathcal{G}_\Xi) \geq 1 - \frac{1}{2n^3}\) whenever \(K \geq 11 \log(n)/P_{\text{min}}^2\). Let \(L\) be the Lipschitz constant of \(\psi\) and set \(K \geq \frac{mL^2 \log(n)}{\tau^2}\). Using the inequality that \(\|N\|_F \leq \sqrt{m}\|N\|_\infty\), we have

\[
\Pr(\|N\|_F \geq \tau) \leq \Pr\left(\|N\|_\infty \geq \frac{\tau}{\sqrt{m}}\right)
\]

\[
= \Pr\left(\exists (i, j) \in \Xi : \left| \psi(\hat{P}_{ij}) - \psi(P_{ij}) \right| \geq \frac{\tau}{\sqrt{m}}\right)
\]

\[
\leq \sum_{(i,j) \in \Xi} \Pr\left(\left| \psi(\hat{P}_{ij}) - \psi(P_{ij}) \right| \geq \frac{\tau}{\sqrt{m}}\bigg| \mathcal{G}_\Xi\right) \Pr(\mathcal{G}_\Xi)
\]

\[
+ \Pr(\mathcal{G}_\Xi^c)
\]

\[
\leq \sum_{(i,j) \in \Xi} \Pr\left(\left| \hat{P}_{ij} - P_{ij} \right| \geq \frac{\tau}{L\sqrt{m}}\bigg| \mathcal{G}_\Xi\right) \Pr(\mathcal{G}_\Xi) + \frac{1}{2n^3}
\]

\[
\leq \sum_{(i,j) \in \Xi} \Pr\left(\left| \hat{P}_{ij} - P_{ij} \right| \geq \frac{\tau}{L\sqrt{m}}\right) + \frac{1}{2n^3}
\]

\[
\leq \frac{1}{2n^3} + \frac{1}{2n^3} = \frac{1}{n^3}
\]

□
Chapter 2

Robust Pairwise Ranking

Abstract

Learning to rank $n$ items from pairwise comparisons is a well-studied problem. In the presence of adversarial noise, many existing models fail to work well – and provably so – we quantify this precisely for a generic definition of an ‘adversary’. On the other hand, existing robust ranking heuristics do not have statistical and computational guarantees associated with them. In this chapter, we devise a guaranteed polynomial time ranking algorithm which robustifies several existing models such as the classic Bradley–Terry–Luce (BTL) [4] model and certain generalizations of it. To the best of our knowledge, our present work is the first to propose an algorithm that provably recovers an $\epsilon$-optimal ranking with high probability while allowing as large as $O(n)$ perturbed pairwise comparison results per item. Furthermore, we show robust recovery results in the partially observed setting. We extend our technique for other binary choice models as well. Experiments confirm that our algorithm handles adversarial noise well and outperforms several popular methods including maximum likelihood and rank centrality.
2.1 Introduction

Ranking from pairwise comparisons is a well-studied problem in machine learning with a rich and long history. Starting with the seminal work of [4], the Bradley–Terry–Luce (BTL) model has become a standard and widely used statistical model for ranking. A key challenge while applying these models and estimating a ranking from datasets in practice is the undesirable effect of grossly corrupted data samples which we aim to understand in this chapter. We propose a novel noise modeling approach followed by proposing a solution that has interesting connections to robust matrix factorization. To this end, we consider the following learning problem. Suppose there are \( n \) items we wish to order based on a notion of comparison, between every pair of items, with probabilistic outcomes. Further, we are given a set, \( \mathcal{K} = \{(i, j, \{y_{ij}^k\})\} \), consisting of \( K \) independent pairwise comparison outcomes, denoted by \( \{y_{ij}^k\} \in \{0, 1\}, k \in [K] \), between pairs of items \( (i, j) \subseteq [n] \times [n] \), a significant proportion of which might be corrupted by an adversary. In this passive learning setting, the concrete questions we wish to address are:

1. Is it possible to identify the pairs whose comparison results were corrupted by an adversary?
2. Having identified the corrupted results, as desired, is it possible to filter them out while computing a global ranking of the \( n \) items?
3. Is this task tractable statistically and computationally?
4. If so, is it possible to construct a provably correct and efficient algorithm, and what are the associated properties?
5. Further, does it work well in practice?

We systematically answer the above questions in the affirmative. Specifically, our contributions are as follows. We give a generic definition of (additive) adversarial noise which can be handled for a broad class of statistical models including the classic BTL model and also
certain extensions of it such as the recently proposed general Low-Rank (LR) models [48]. First, as is the case with standard estimation techniques, if the noise is not modeled and handled well, we show that the quality of estimated ranking could be quite bad, by quantifying the error of the estimated ranking with respect to the best possible ranking. Next, under certain (information-theoretically tight) identifiability assumptions on the properties of the adversary, we develop a correct and efficient ranking method that guarantees $\epsilon$-accurate high-probability learnability in a manner that is ‘robust’ and oblivious to the effects of the adversary. Our learning algorithm is provably characterized by polynomial time computational complexity. Next, in practice, it is often the case that not all pairs are compared and even the observed pairwise comparison data could be adversarially corrupted – we also characterize the conditions for guaranteed robust recovery in this scenario. Finally, we support our theoretical results by showing robust ranking results in experiments.

### 2.1.1 Related Work

We now briefly present relevant work in: (1) ranking models and ranking algorithms that handle noise, and (2) robust subspace recovery methods which will be needed for us to prove recovery results for ranking.

**Ranking Models**: In the BTL model, item $i$ has an associated score $w_i$; then, the probability that item $i$ is preferred over $j$ is given by $P_{ij} = e^{-w_i}/(e^{-w_i} + e^{-w_j})$ where $\mathbf{w} \in \mathbb{R}^n$ is the BTL parameter vector to be estimated from data; here, $\mathbf{P} \in \mathbb{R}^{n \times n}$ is called the ‘preference matrix’. A closely related model is the recently proposed LR model [48] wherein a generic class of preference matrices is characterized to be those having low rank under transformations using certain functions; specifically, for BTL-like models, the logit function defined as $\psi(x) = \log(x/(1-x))$ turns out to right choice as shown in their paper. However, while their model accounts for missing information, they do not consider the harder problem
of handling adversarial noise. Several robust ranking heuristics have been proposed (for example, [53, 56]) but these approaches do not have theoretical guarantees associated with them. Recently, the Sync-Rank algorithm, for handling different noise models as compared to the one considered in the preset work, was proposed in [14] and is based on spectral techniques. Another related work is [47] which proposes the so-called ‘Generalized Low-Noise’ (GLN) condition that \( \forall i \neq j, P_{ij} > P_{ji} \implies \sum_{h=1}^{n} \alpha_h P_{hj} > \sum_{h=1}^{n} \alpha_h P_{hi} \) for \( \alpha \in \mathbb{R}^n \).

When \( \alpha_h = 1, \forall h \) they analyze the sample complexity and show convergence properties of various popular ranking algorithms like:

1. Maximum Likelihood (ML): this entails solving \( \operatorname{arg\,max}_w \sum_{i<j} (\hat{P}_{ij}(w_j - w_i) - \log(1 + \exp(w_j - w_i))) \) where \( w \in \mathbb{R}^n \) is the BTL parameter vector and \( \hat{P}_{ij} \) is the empirical preference matrix.

2. Rank Centrality (RC) [42]: here, one sorts items by their scores which are computed as the stationary distribution of an appropriately normalized empirical preference matrix; this approach has a known sample complexity guarantee of \( O(n \log(n)) \).

3. Borda Count (BC) [28]: this heuristic involves ranking an item according to the fraction of times it beats other items.

For the general case \( \alpha \) (which previous methods fail to handle), they also propose a noise-tolerant SVM-based method for rank aggregation. However, in the adversarial setting we consider in this chapter, GLN could be violated and hence requires a different algorithmic approach and analysis.

**Robust Subspace Recovery:** It is well-known that Principal Component Analysis (PCA), an ubiquitous technique for subspace identification, is not robust to outliers; this may be attributed to the fact that PCA is an \( L_2 \) optimization problem due to which grossly corrupted data points may perturb and skew the eigenvectors spanning the maximum variance subspace of the data points significantly.
The Robust PCA (RPCA) problem [44] addresses the following question: suppose we are given a data matrix \( \mathbf{M} \) which is the sum of an unknown low-rank matrix \( \mathbf{L} \) and an unknown sparse matrix \( \mathbf{S} \), can we recover each of the component matrices? While several works [55, 23] analyze this problem, it is shown in [44] that, under information-theoretically tight assumptions, a simple iterative algorithm based on non-convex alternating projections of appropriate residuals provably yields an \( \epsilon \)-accurate solution in \( O(\log(1/\epsilon)) \) iterations with an overall computational complexity of \( O(n^2r^2 \log(1/\epsilon)) \) where \( r \) is the rank of \( \mathbf{L} \). We will use this result, in particular, to derive guarantees for our ranking problem.
2.2 Problem Setup

2.2.1 Notation

We first define some notation. We denote the set of all permutations of \( n \) items as \( S_n \). If not specifically defined, we use lower-case letters for scalars, upper-case letters for global constants, lower-case bold-face letters for vectors and upper-case bold-face letters for matrices; specifically, \( P \) denotes a preference matrix. Let \( P_n := \{ P \in [0,1]^{n \times n} | P_{ij} + P_{ji} = 1 \} \) denote the set of all pairwise preference matrices over \( n \) items. Let the set of stochastic-transitive matrices be \( P^{ST}_n := \{ P \in P_n | P_{ij} > 1/2, P_{jk} > 1/2 \implies P_{ik} > 1/2 \} \). Let the set of preference matrices described by the BTL model be \( P^{BTL}_n := \{ P \in P_n | \exists w \in \mathbb{R}^n \text{ s.t. } e^{-w_i}/(e^{-w_i} + e^{-w_j}) \} \). Let \( \psi : [0,1] \mapsto \mathbb{R} \) be a strictly increasing bijective \( L \)-Lipschitz function and define the class of low-rank preference matrices with respect to \( \psi \) as \( P^{LR(\psi,r)}_n = \{ P \in P_n | \text{rank}(\psi(P)) \leq r \} \) where \( r \in [n] \); when we apply such a transformation to a matrix, it is applied entry-wise. In this chapter, we take \( \psi \) to be the logit function.

For any matrix \( M \in \mathbb{R}^{n \times n} \), let the infinity norm be denoted by \( \| M \|_\infty = \max_{i,j} |M_{ij}| \), the Frobenius norm be denoted by \( \| M \|_F = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} M_{ij}^2} \), the spectral norm be denoted by \( \| M \|_2 = \max_{x,y} x^\top M y \). Denoting the indicator function by \( 1 \), define the zero norm of a matrix to be the maximum number of non-zero elements in any row/column, ie, \( \| M \|_0 = \max(\max_i \sum_{j=1}^{n} 1(M_{ij} \neq 0), \max_j \sum_{i=1}^{n} 1(M_{ij} \neq 0)) \). Let the Singular Value Decomposition (SVD) of a square matrix be given by \( M = U \Sigma V^\top \) where \( U, V \in \mathbb{R}^{n \times r} \) are orthonormal matrices (whose columns are singular vectors) and \( \Sigma \in \mathbb{R}^{r \times r} \) is the diagonal matrix of singular values. Now, \( M \) is said to be \( \mu \)-incoherent if \( \max(\max_i \| e_i^\top U \|_2, \max_i \| e_i^\top V \|_2) \leq \mu \sqrt{r/n} \) where \( e_i \) denotes the \( i^{th} \) basis vector in \( \mathbb{R}^n \). Also, let \( \sigma_{\max} := \max_i \Sigma_{ii} \) and \( \sigma_{\min} := \min_i \Sigma_{ii} \).
We define the distance between a permutation $\sigma \in S_n$ and a preference matrix $P \in P_n$ as:

$$\text{dist}(\sigma, P) := \left(\begin{array}{c} n \\ 2 \end{array}\right)^{-1} \sum_{i<j} 1 \left( (P_{ij} > 1/2) \land (\sigma(i) \succ \sigma(j)) \right)$$

$$+ \left(\begin{array}{c} n \\ 2 \end{array}\right)^{-1} \sum_{i<j} 1 \left( (P_{ji} > 1/2) \land (\sigma(j) \succ \sigma(i)) \right)$$

Note that the above loss function basically is the number of pairs on which the ordering with respect $\sigma$ and $P$ differ divided by the number of ways to choose two out of $n$ items. Finally, let $P_{\text{min}} = \min_{i \neq j} P_{ij}$ and $\Delta = \min_{i \neq j} |\psi(P_{ij}) - \psi(1/2)|$.

### 2.2.2 Characterization of the Adversary

The following (weak) assumption characterizes the properties of the adversary. We shall see in the next section that it is information-theoretically tight in order to guarantee recovery in the solution approach that we propose. Note that this is a deterministic assumption; in particular, we do not have any distributional assumptions regarding the locations, the signs or the magnitudes of the corruptions, and hence is very general.

**Assumption 2.1.** The (additive) adversarial noise which corrupts a $\mu$-incoherent preference matrix $P \in P_n^{LR(\psi,r)}$ is modeled by a skew-symmetric sparse matrix $S$ so that the corrupted preference matrix $P^c \in P_n$ is given by $P^c = P + S$. We assume the (deterministic) bounded degree condition that $\|S\|_0 \leq d < n$ such $d < n/512 \mu^2 r$ where $r \leq n$.

So, why do existing non-robust algorithms provably fail when not considering the effects of the adversary? This question is answered by the following proposition which is a negative result that precisely quantifies how bad a ranking could be when an algorithm uses the corrupted pairwise preference matrix.
Procedure 4 RPCA: Robust Principal Component Analysis

**Input:** $M = L^* + S^*$, rank $r$ of $L^*$.

**Output:** $\hat{L}, \hat{S}$.

1. Solve the following optimization problem using Algorithm 1 of [44]:

   \[
   \{\hat{L}, \hat{S}\} = \arg\min_{L,S} \|M - L - S\|_F \\
   \text{s.t.} \ \text{rank}(L) \leq r, \|S\|_0 \leq d
   \]

2. return $\hat{L}, \hat{S}$.

Procedure 5 PR: ($\gamma$-approximate) Pairwise Ranking

**Input:** Preference matrix $M \in \mathbb{R}^{n \times n}$.

**Output:** Ranking $\hat{\sigma}$.

1. Compute $\forall i, \ v_i = \sum_{j=1}^{n} \mathbb{1}(M_{ij} > 1/2)$.

2. return $\hat{\sigma} \leftarrow \text{Sort}(v)$.

Claim 2.1 (Worst-case estimation error). Under Assumption 2.1 it is possible that $\text{dist}(\hat{\sigma}, P^c) = O(1)$.

Proof. Assume that we are exactly given the entries of the preference matrix as opposed to sampling them. Note that in order to estimate a ranking from a given preference matrix, we still need to use a pairwise ranking procedure. Let $\hat{\sigma} \in S_n$ be the output of any Pairwise Ranking (PR) procedure with respect to an underlying preference matrix $Q \in \mathcal{P}_n$. For a constant $\gamma > 1$, $\hat{\sigma}$ is said to be $\gamma$-approximate if $\text{dist}(\hat{\sigma}, Q) \leq \gamma \min_{\sigma \in S_n} \text{dist}(\sigma, Q)$. Define the following distance which measure the fraction of item pairs over which two preference matrices $\{Q, R\} \in \mathcal{P}_n$ disagree.

\[
\text{dist} (Q, R) := \left( \begin{array}{c} n \\ 2 \end{array} \right)^{-1} \sum_{i<j} \mathbb{1} (Q_{ij} > 1/2) \wedge (R_{ij} < 1/2) \\
+ \left( \begin{array}{c} n \\ 2 \end{array} \right)^{-1} \sum_{i<j} \mathbb{1} (Q_{ij} < 1/2) \wedge (R_{ij} > 1/2)
\]
By Lemma 20 of [48], for $Q \in \mathcal{P}^{ST}_n$ and $R \in \mathcal{P}_n$, we have $\text{dist}(\hat{\sigma}, Q) \leq (1 + \gamma) \text{dist}(Q, R)$. But note that it is possible that $\text{dist}(Q, R) = 1$ as it is easy to construct by $R$ that disagrees with $Q$ in every entry by simply setting $R = Q^\top$. Now, we may set $Q = P$ and $R = P^c$ for any algorithm that uses $P^c$ for ranking; specifically, for the adversary satisfying Assumption 2.1, we can see by a direct counting argument that $\text{dist}(Q, R) \leq \frac{d(2n-1-d)}{n(n-1)}$ which proves the claim. \qed
2.3 Fully Observed Adversarial Setting

2.3.1 Algorithm

In this section, we answer Question 4 We present our main algorithm for robust passive ranking from pairwise comparisons in the presence of adversarial noise in Algorithm 6. The input data consist of the set of pairwise comparison results \( \mathcal{R} = \{(i, j, \{y_{ij}^k\})\}, (i, j) \in [n] \times [n], k \in [K], y_{ij}^k \in \{0, 1\} \). The algorithm assumes the true rank of \( \psi(P) \) as an input parameter; specifically, for the BTL model, we set \( r = 2 \). Algorithm 6 calls the following procedures:

1. **Robust PCA (Procedure 4):** Note that Step 3 of Algorithm 6 uses a matrix low-rank plus sparse decomposition subroutine. To obtain our recovery guarantee, it is sufficient to use the robust PCA problem as a black-box method; for the precise details of this algorithm, we refer the reader to [44]. In particular, for our analysis, we use the noise-case guarantees in their paper. This is characterized by a (strongly-polynomial) running time of \( O(n^2 r^2 \log(1/\epsilon)) \) and guarantees \( \epsilon \)-recovery of the component matrices under the conditions of Assumption 2.1 and Lemma 2.3.

2. **\( \gamma \)-approximate pairwise ranking procedure (Subroutine 5):** Step 4 of Algorithm 6 calls a constant factor approximate ranking procedure. Specifically, we use the Copeland procedure [12] which has a 5-approximation guarantee [13] and involves sorting the items according to a score of item \( i \) given by \( \sum_{j=1}^{n} 1(\bar{P}_{ij} > 1/2) \).

2.3.2 Analysis

We begin with a useful short result followed by the statement and the proof of our main result that, with high probability, we achieve \( \epsilon \)-accurate ranking in polynomial time using polynomial number of samples, despite the presence of adversarial noise. Precisely, The-
Algorithm 6 RPR: Robust Pairwise Ranking

**Input:** Comparison dataset $\mathcal{N} = \{(i,j, \{y^k_{ij}\})\}$, true rank $r$.

**Output:** Ranking of $n$ items, $\hat{\sigma} \in S_n$.

1: Estimate entries of $\hat{P}$ for $i \leq j$ as:

$$
\hat{P}_{ij} = \begin{cases} 
\frac{1}{K} \sum_{k=1}^{K} y^k_{ij} & \text{if } i < j \\
1/2 & \text{if } i = j
\end{cases}
$$

2: Set $\hat{P}_{ij} = 1 - \hat{P}_{ji}$ for all $i > j$.

3: Perform robust PCA: $\{\psi(\hat{P}), \hat{S}\} \leftarrow \text{RPCA}(\psi(\hat{P}), r)$.

4: Using a pairwise ranking procedure after taking the inverse transform: $\hat{\sigma} \leftarrow \text{PR}(\hat{P})$.

5: return $\hat{\sigma}$.

Lemma 2.1 (Some properties of the logit function). Let $a, b, c \in (0, 1)$ such that $c = a + b$. Then, we have,

1. $\psi(c) = \psi(a) + \psi(a + b) + \psi(1 - a)$
2. $\psi(a) + \psi(1 - a) = 0$.

Proof. Both follow by using the definition of the logit function that $\psi(a) = \log(a/(1 - a))$ and using the property that $\log(ab) = \log(a) + \log(b)$.

Theorem 2.1 (Provably good estimation of ranking in LR models in the presence of adversarial noise). Let $P \in \mathcal{P}_n^{LR(\psi, r)}$ be the true preference matrix according to which the pairwise comparison dataset $\mathcal{N} = \{(i,j, \{y^k_{ij}\})\}$ is generated for all items pairs $(i,j)$ such that $k \in [K]$. Let $\hat{P}$ be the empirical preference matrix computed using $\mathcal{N}$. Let $S \in [0, 1]^{n \times n}$ be the adversarial matrix that additively corrupts $\hat{P}$. Let $\psi$ be $L$-Lipschitz in $[\frac{P_{\min}}{2}, 1 - \frac{P_{\min}}{2}]$ and $\psi(P)$ be $\mu$-incoherent. Let each pair be compared independently $K \geq 16384\mu^2(1 + \ldots$
\( \gamma L^2n^2 \log^2(n)/\epsilon \Delta^2 \) times where \( \Delta = \min_{i \neq j} |\psi(P_{ij}) - \psi(1/2)| \). Then, with probability at least \( 1 - 1/n^3 \), Algorithm 6 returns an estimated permutation \( \hat{\sigma} \) such that \( \text{dist}(\hat{\sigma}, P) \leq \epsilon \).

**Remark 2.1 (Computational complexity).** In Algorithm 6, Step 1 takes \( O(n^2K) = O(n^4 \log^2 n/\epsilon) \) time, Step 3 takes \( O(n^2 r^2 \log(1/\epsilon)) \), and Step 4 takes \( O(n^2 + n \log n) \) time. Thus, putting together the cost of these main steps, the overall computational complexity of our robust ranking algorithm for \( P \in \mathcal{P}_n^{LR(\psi, r)} \) is \( O(n^4 \log^2 n/\epsilon) \).

**Remark 2.2 (Identifying adversarially corrupted pairwise comparisons).** From Step 3 of Algorithm 6, using Theorem 2 of [44], we also have \( \text{Supp}(\hat{S}) \subseteq \text{Supp}(S) \) and thus we can identify the corrupted pairwise comparison results.

**Remark 2.3 (Missing data versus adversarially corrupted data).** Note that the adversarial sparse noise we consider subsumes the setting when comparison results for certain pairs are missing as in [48] and hence directly applies in that situation. Moreover, since the support and magnitude of the corrupted entries of the preference matrix are unknown, the problem considered in this paper is harder; consequently, our sample complexity is \( O(n^2) \) as opposed to \( O(n \text{ poly log } n) \) in their work.

**Proof.** Let \( \tilde{P}_{ij} \) be the empirical probability estimate of \( P_{ij} \). Note that we compute \( \tilde{P}_{ij} = \frac{1}{K} \sum_{k=1}^{K} y_{ij}^k \) from the given pairwise comparison dataset, \( \mathcal{X} = \{(i, j, \{y_{ij}^k\})\} \). Now, \( \tilde{P} = \tilde{P} + S \).

By Lemma 2.1, we may write the adversarially corrupted empirical probability estimate as \( \psi(\tilde{P}) = \psi(\tilde{P}) + \tilde{S} \) where \( \tilde{S} = \psi(\tilde{P} + S) + \psi(1 - \tilde{P}) \). We have \( \psi(\tilde{P}) = \psi(P) + \tilde{N} \) where \( \tilde{N} = \psi(\tilde{P}) - \psi(P) \). Now, this noise, \( \tilde{N} \), is purely due to finite-sample effects which can be controlled (using concentration arguments given in the inequality \( \xi_3 \) below) by driving it down to as small a value as we want by ensuring large enough number of comparisons for each pair. Note that we input \( \psi(\tilde{P}) = \psi(P) + \tilde{S} + \tilde{N} \) to Subroutine 4 and obtain \( \psi(\tilde{P}) \) as the output in Step 3 of Algorithm 6. Hence, using Theorem 2 from [44], if \( \|\tilde{N}\|_\infty \leq \sigma_{\min}(\psi(P))/100n \),
we have,

\[ \| \psi(\bar{P}) - \psi(P) \|_F \leq \epsilon' + 2\mu^2 r \left( 7 \| \tilde{N} \|_2 + \frac{8n}{r} \| \tilde{N} \|_\infty \right) \]

after \( T \geq 10 \log(3\mu^2 r \sigma_1 / \epsilon') \) iterations associated with Step 1 of Subroutine 4. Next, we have, with probability at least \( 1 - 1/n^3 \),

\[ \| \psi(\bar{P}) - \psi(P) \|_F \leq \epsilon' + 2\mu^2 r \left( 7 \| \tilde{N} \|_2 + \frac{8n}{r} \| \tilde{N} \|_\infty \right) \]

\[ \leq \epsilon' + 32\mu^2 n \| \tilde{N} \|_2 \leq \epsilon' + 32\mu^2 n \tau \]

\[ \xi_3 \leq n \sqrt{\frac{\epsilon}{1 + \gamma} \frac{\Delta}{2}} \]

where \( \xi_1 \) follows by using \( r \leq n \) and \( \| \tilde{N} \|_\infty \leq \| \tilde{N} \|_2 \), \( \xi_2 \) follows by substituting for \( \tilde{N} \) from Lemma 2.2 with \( K \geq \frac{L^2 n^2 \log n}{\tau^2} \), and \( \xi_3 \) is obtained using \( \epsilon' = n \sqrt{\frac{\epsilon}{1 + \gamma} \frac{\Delta}{4}} \) and \( \tau = \min \left( \frac{\sigma_{\min}(\psi(P))}{100}, \frac{\epsilon}{1 + \gamma} \frac{128\mu^2}{n} \right) \). Then using similar arguments as proof of Theorem 13 in [48], we obtain our result.

\[ \square \]

**Lemma 2.2 (Concentration of sampling noise).** Under the conditions of Theorem 2.1, let each item pair be compared such that the number of comparisons per item pair is \( K \geq \frac{L^2 n^2 \log(n)}{\tau^2} \); with probability at least \( 1 - 1/n^3 \), \( \| \tilde{N} \|_2 \leq \tau \).

**Proof.** Let \( L \) be the Lipschitz constant of \( \psi \) and set \( K \geq \frac{L^2 n^2 \log(n)}{\tau^2} \). Using the inequality
that $\|\tilde{N}\|_2 \leq n \|\tilde{N}\|_\infty$,

$$\Pr\left(\|\tilde{N}\|_2 \geq \tau\right) \leq \Pr\left(\|\tilde{N}\|_\infty \geq \frac{\tau}{n}\right)$$

$$= \Pr\left(\exists(i, j) : |\psi(\hat{P}_{ij}) - \psi(P_{ij})| \geq \frac{\tau}{n}\right)$$

$$\leq \sum_{i,j} \Pr\left(|\psi(\hat{P}_{ij}) - \psi(P_{ij})| \geq \frac{\tau}{n}\right)$$

$$\leq \sum_{i,j} \Pr\left(|\hat{P}_{ij} - P_{ij}| \geq \frac{\tau}{nL}\right) \leq \frac{1}{n^3} \tag*{\blacksquare}$$

Next, for completeness, we recall the following lemma (proved in Theorem 8 and Lemma 14 of [48]) which characterizes the incoherence constant $\mu$ of $P \in (P_n^{LR(\psi, 2)} \cap P_n^{ST})$ in Assumption 2.1.

**Lemma 2.3 (Incoherence of BTL and LR models).** We have $P \in (P_n^{LR(\psi, 2)} \cap P_n^{ST})$ if and only if $\psi(P) = uv^\top - vu^\top$ for $u \in \mathbb{R}_+^n$ and $v \in \mathbb{R}^n$ where $u^\top v = 0$. Moreover, $\psi(P)$ is $\mu$-incoherent where $\mu = \sqrt{\frac{n}{2} \left(\frac{u_{\max}^2}{u_{\min}^2} + \frac{v_{\max}^2}{v_{\min}^2}\right)}^{1/2}$ where $u_{\min} = \min_i |u_i|$, $u_{\max} = \max_i |u_i|$, $v_{\min} = \min_i |v_i|$ and $v_{\max} = \min_i |v_i|$. We also have $P_n^{BTL} \subset (P_n^{LR(\psi, 2)} \cap P_n^{ST})$ since we may set $u = 1$ where $1$ is the all-ones vector and $v = w$ where $w$ is the BTL parameter vector. In this case, we may rewrite $\mu = \sqrt{\frac{n}{2} \left(1 + \frac{(u_{\max} - w)^2}{(u_{\min} - w)^2}\right)}$ where $w = \frac{1}{n} \sum_{i=1}^n w_i$.

The following corollary makes precise our claim that up to $O(n^2)$ item pairs may be subject to adversarial corruption but our RPR algorithm still recovers a good ranking.

**Corollary 2.1 (Recovery result for BTL model).** Consider $P \in P_n^{BTL}$. Using Assumption 2.1, let the adversarial matrix be $S \in [0, 1]^{n \times n}$ satisfying $\|S\|_0 \leq n/1024\mu^2$ where $\mu$ is characterized as in Lemma 2.3. Then, with probability $1 - 1/n^3$, the output of Algorithm 6 with input $\hat{P}$ computed using $\mathcal{X} = \{(i, j, \{Y_{ij}^k\})\}$ satisfies and $r = 2$, $\text{dist}(\hat{\sigma}, P) \leq \epsilon$. 

38
Algorithm 7 RPOPR: Robust Partially Observed Pairwise Ranking

**Input:** Comparison dataset $\mathcal{R} = \{(i, j, \{y^k_{ij}\})\}$, true rank $r$.

**Output:** Ranking of $n$ items, $\hat{\sigma} \in S_n$.

1. Estimate entries of $\tilde{P}$ for $i \leq j$ as:
   
   $$
   \tilde{P}_{ij} = \begin{cases} 
   \frac{1}{K} \sum_{k=1}^{K} y^k_{ij} & \text{if } i < j \text{ and } (i, j) \in \Omega \\
   1/2 & \text{if } i = j \text{ and } (i, j) \in \Omega \\
   1/2 & \text{if } (i, j) \notin \Omega
   \end{cases}
   $$

2. Set $\hat{P}_{ij} = 1 - \hat{P}_{ji}$ for all $i > j$.

3. Set $R \leftarrow \text{OptSpace}(\psi(\tilde{P})_{\Omega})$.

4. Use a robust PCA procedure: $\psi(\tilde{P}) \leftarrow \text{RPCA}(R)$.

5. Using a pairwise ranking procedure after taking the inverse transform: $\hat{\sigma} \leftarrow \text{PR}(\tilde{P})$.

6. return $\hat{\sigma}$.

### 2.4 Partially Observed Adversarial Setting

In practice, there are two important scenarios, i.e., it might be the case that (1) not all pairs of items are compared (2) some information is available regarding which of the comparison results are adversarially corrupted (for example, by observing user behavior during the data collection process). Motivated by these, we consider the partially observed setting in this section. Both of these scenarios may be modeled in a unified manner by setting the corresponding missing entries of the preference matrix to zero (or a specific constant to account for numerical stability). We present our robust ranking algorithm for this setting in Algorithm 7 – this essentially involves using the ‘OptSpace’ matrix completion algorithm of [32] followed by using the robust PCA algorithm of [44] as sub-routines. We now derive the recovery guarantees as follows.

**Theorem 2.2** (Provably good estimation of ranking in BTL model in the presence of adversarial noise as well as missing data). Consider a similar notations as in Theorem 2.1 but let $P \in \mathcal{P}_n^{BTL}$. Let $\Omega \subseteq [n] \times [n]$ be a the set of item pairs that are compared. Assume $\Omega$ is drawn uniformly from all subsets of $[n] \times [n]$ of size $|\Omega|$ such that $|\Omega| \geq C''n \log(n)$ and let the sparse noise satisfy $\|S\|_{\infty} \leq \Delta_w \frac{\log(n)}{C\Delta_w}$ where $\Delta_w := \min_{i,j} |w_i - w_j|$. Let the
number of comparisons per pair be $K \geq cn^4/\Delta_w$. Then with probability at least $1 - 2/n^3$, Algorithm 7 returns a ranking that satisfies $\text{dist}(\hat{\sigma}, P) \leq \epsilon$.

**Remark 2.4 (Robust Estimation of BTL Model in the Partially Observed Case).**

For the BTL model, Theorem 2.2 says $O(n \log n)$ pairs suffice to estimate the BTL model which matches bounds from [48]. Further, even in this incomplete comparison data case, we are able tolerate uniformly random additive sparse noise its maximum absolute entry scaling as the order of the BTL ‘score-gap’ divided by the number of items upto logarithmic factors, ie, $\tilde{O}(\Delta_w/n)$.

**Proof.** From Lemma 2.3, we have $\psi(P) = 1w^T - w1^T$ for the BTL model where $\psi$ is the logit function. Clearly, in this case, $\psi(P)$ is a real skew-symmetric matrix of rank $r = 2$. Since it is skew-symmetric, its eigenvalues, which are the roots of its characteristic polynomial, are of the form $\pm \lambda i$ for some $\lambda \in \mathbb{R}$ and $i = \sqrt{-1}$, and hence, $\sigma_{\min}(\psi(P)) = \sigma_{\max}(\psi(P))$, ie, the condition number of $\psi(P)$, $\kappa = 1$. Now, we recall the spectral-lower bound from Corollary 2 of [22],

$$\sigma_{\min}(\psi(P)) \geq \frac{\|\psi(P)\|_F}{\sqrt{r(r-1)}} \geq \sqrt{\frac{n(n-1)}{2}} \Delta_w$$  \hfill (2.1)

where $\Delta_w = \min_{i,j} |w_i - w_j|$.

Let $\Omega \subseteq [n] \times [n]$ be a subset of all the item pairs with comparison results among which some might be corrupted by sparse noise, ie, $\psi(\hat{P}_\Omega) = \psi(P_\Omega) + \tilde{S}_\Omega + \tilde{N}_\Omega$. Let $T := \tilde{S}_\Omega + \tilde{N}_\Omega$. From Theorem 1.2 of [32], we have $\frac{1}{n} \|\psi(\hat{P}) - \psi(P)\|_F = \frac{1}{n} \|T + M\|_F \leq C\kappa^2 \frac{n\sqrt{\tau}}{||T||_2}$ where $M$ is the noise matrix after obtaining the completed matrix $\psi(\hat{P})$ from $\psi(\hat{P}_\Omega)$ using OptSpace.
Using triangle inequality and noting that $|\Omega| \geq C'' n \log(n)$, the noise may be bounded as

$$
\|\tilde{N}_\Omega + M\|_\infty \leq \|\tilde{N}_\Omega + M\|_F \leq \|T\|_2 \frac{\sqrt{2}Cn^2}{|\Omega|} + \|\tilde{S}_\Omega\|_F
$$

$$
\leq C' \frac{n}{\log(n)} \|\tilde{S}_\Omega\|_2^{1/2}
$$

(2.2)

where $C'$, $C''$ and $C'''$ are constants and $\zeta_1$ is obtained by using the triangle inequality that

$$
\|T\|_2 \leq \|\tilde{S}_\Omega\|_2 + \|\tilde{N}_\Omega\|_2^{1/2},
$$

followed by setting $K \geq cn^4/\Delta_w$ for constant $c$ and finally using

$$
\|\tilde{S}_\Omega\|_F \leq \sqrt{n} \|\tilde{S}_\Omega\|_2^{1/2}.
$$

Then, combining Equations 2.2 and 2.1, we have if

$$
\log(n) \Delta_w \geq \|\tilde{S}_\Omega\|_2 = \|\psi(\tilde{P}) - \psi(\tilde{P})\|_2
$$

$$
\geq \|\psi(\tilde{P}) - \psi(\tilde{P})\|_\infty \geq L \|\tilde{P} - \tilde{P}\|_\infty \geq \|S\|_\infty
$$

where $C_\Delta$ is a global constant and using Lemma 2.2, then we have the guarantee (along similar lines as that of Theorem 2.1 that Algorithm 7 returns an estimated permutation which satisfies dist$(\tilde{\sigma}, P) \leq \epsilon$. □

### 2.5 Generalization to Other Models

Related to the BTL model are many other binary choice models [19] such as the Thurstonian model [51]. In such models, the preference matrix has been shown to be low-rank under appropriate choices of $\psi$; for instance, for the Thurstonian models, the probit function turns out to be the right choice. For further details, we refer the reader to the work of [48].

Let $a, b, c \in (0, 1)$ such that $c = a + b$. Then, for any general non-linear $L$-Lipschitz function, we write $\psi(c) = \psi(a + b) = \psi(a) + \psi(a + b) - \psi(a)$. The error may be lower bounded by $|\psi(a + b) - \psi(a)| \geq Lb$. Thus, for any adversarial model wherein we have $P' = P + S$, we
have:

$$\psi(P^c) = \psi(P) + (\psi(P + S) - \psi(P)) = \psi(P) + \tilde{S}$$

where $\tilde{S}$ is also a deterministic sparse corruption matrix with the absolute value of the non-zero entries lower bounded by $L \min_{i,j} S_{ij}$. With the appropriate $\psi$, $\psi(P)$ will be a low-rank matrix and hence Algorithm 6 and the associated recovery guarantee of Theorem 2.1 holds.
2.6 Experiments

In this section, we answer Question 5. We now perform simulations in order to understand the performance of our robust ranking approach in practice. First, we begin with the BTL model. We generate synthetic pairwise comparison data and also adversarial sparse matrix as follows. We generate the entries of the BTL parameter vector \( w \) from \( \mathcal{N}(0, \nu^2) \) followed by generating the ground truth preference matrix from with \( y_{ij}^k \) is sampled for all item pairs \((i, j)\) for a fixed \( K \). The adversarial sparse matrix \( S \) is generated as a skew-symmetric matrix where each entry is non-zero independently with probability \( d/n \) followed by generating a value for an entry from \( U(5, 10) \) and then setting the sign to be positive with probability \( 1/2 \); this corruption matrix is then added to the \( \psi(P) \) to give \( \psi(P^c) \) which is then input to our algorithm; the same \( P^c \) is used for the other algorithms as well.

We take the number of items to be \( n = 500 \). In plots in Figure 2.1, we compare the performance of our RPR approach using Algorithm 6 against well-known ranking algorithms, such as Rank Centrality (RC [42], Maximum Likelihood (ML) and Borda Count (BC) count [28], with special attention to robustness to the noise model that we consider in this chapter. We
vary two parameters namely, $\nu$, spread of the BTL scores, and $d$, the density of adversarial corruption matrix. All our results averaged over five runs. We observe that our algorithm maintains low recovery error in spite of increasing the problem hardness, thus outperforming previous approaches in all cases.
2.7 Conclusion

In this chapter, we have proposed a robust algorithm for provably correct and efficient ranking in the BTL, LR and general binary choice models. Our algorithm had two key components namely, using the robust PCA subroutine and also a standard pairwise ranking procedure. Further, we also handled the partially observed setting, wherein only some item pairs are compared, by integrating matrix completion techniques into our robust learning algorithm. In all cases, we provided statistical and computational guarantees using novel techniques. Some future research directions include (1) tightening the recovery results for the partially observed setting under weaker conditions (possibly using noisy-case extensions of [55]), (2) exploring other notions of adversarial noise, and (3) understanding the minimax optimal rates for ranking estimators under various noise models.
Chapter 3

Streaming Robust PCA

Abstract

In this chapter, we consider the problem of robust PCA in the streaming setting with space constraints. Specifically, we adopt the popular spiked-covariance model and focus on the rank-1 case. The problem can be stated as follows: at time $t$, we are given a $n$-dimensional data vector $x_t = uz_t + s_t + w_t$ where $u$ is a fixed vector, $z_t$ is a Gaussian random variable, $s_t$ is an arbitrary sparse perturbation and $w_t$ is the usual dense noise vector with a given variance. Without storing samples, we wish to recover $u$ and subsequently also separate the sparse perturbation $s_t$ from each sample. A key challenge is that $s_t$ is a sparse vector with unknown magnitude and support. While recovery guarantees for this problem are known with stronger assumptions, to the best of our knowledge, our result is the first to obtain finite-sample guarantees while having the weakest assumption on the sparse perturbation, namely, deterministic support, and a standard identifiability assumption on the low-rank component, namely, incoherence. Essentially, our algorithm performs simple iterative hard-thresholding followed by stochastic block power method. Our algorithm also has the optimal
space complexity of $O(n)$ and a sample complexity of $O(n \log n)$. 
3.1 Introduction

The robust PCA problem addresses the following question: suppose we are given a data matrix which is the sum of an unknown low-rank matrix and an unknown sparse matrix, can we recover each of the component matrices? Despite the inherent non-convexity of the problem, recent advances have provided algorithms with near-optimal convergence guarantees. However, these bounds hold only in the batch setting, ie, when the entire data matrix is known. In the present work, we analyze robust PCA in the streaming setting, focusing on the rank-1 case where we would like to recover the top eigenvector of the true covariance without the perturbation effect due to sparse corruptions.

3.1.1 Our Contribution

To the best of our knowledge, we obtain the first-known convergence guarantees from streaming robust PCA while having finite sample complexity of $O(n \log n)$ and also having optimal space complexity of $O(n)$ where $n$ is the dimension; the precise result is stated in Theorem 3.1. The assumptions that we use are natural identifiability assumptions used in the batch case as well, the details of which are presented in Section 3.3. At a high level, our algorithm performs alternating hard-thresholding followed by stochastic block power method. Two specific improvements from earlier works are: (1) we have the weaker deterministic assumption for the sparse perturbation (2) We do not need incoherence of the intermediate updates in our analysis.

3.1.2 Related work

Recently, provable non-convex optimization is gaining a lot of research focus since a lot of practically important problems in big data analysis and large-scale machine learning can
naturally be modeled as non-convex problems. However, with this representational power comes computational and statistical intractability, in general. In many cases, these problems can be solved via algorithms operating in non-convex spaces and have been empirically successful. However, convergence guarantees are less well-known for such algorithms.

Matrix factorization, a central technique in many domains, is a class of non-convex optimization problems for which empirically successful algorithms exist. Recent advances in theoretical machine learning have led to novel analysis techniques and global convergence guarantees for many matrix factorization problems. This chapter is an attempt to add to this growing collection.

**PCA:** Principal Component Analysis (PCA) is an ubiquitous unsupervised learning algorithm and has a rich history. Oja’s algorithm is a classical method for streaming PCA [45]. Though the convergence and empirical performance were known, the asymptotic convergence rate was first provided in [2]. Improving on the analysis of [2], linear convergence is presented in [50] but with the requirement that the initialization vector must have a constant correlation with the true eigenvector. The convergence of block stochastic power method is considered in [39] for PCA in the streaming setting. Recently, a tighter analysis for Oja’s algorithm is provided in [25]. Also, Alecton is a SGD algorithm for low-rank matrix problems presented in [15]. Their analysis is based on control of martingales to achieve $O\left(\frac{1}{\epsilon}\right)$ convergence rate where $\epsilon$ is the desired numerical error.

**Matrix completion:** Convergence guarantees for non-convex alternating minimization algorithm for matrix completion in the batch setting are known (for example, see [20] and the references therein). Recently, for streaming matrix completion, convergence guarantees are presented in [40]. They assume random support drawn according to the Bernoulli model and the algorithm is essentially the block power method performed on an unbiased estimate
of the true covariance. Their sample complexity is $O(n \log n)$ and space complexity is $O(nr)$. Instead of receiving incomplete data vectors at each time instance, [29] recently obtained guarantees for an online matrix completion model wherein, at each time instance, some entry of the matrix is revealed.

**Robust PCA:** The convergence of the non-convex alternating projections based method was analyzed in [44] in the batch setting. Recently, a projected gradient method on factorized matrices was presented in [55]. They also match the time complexity lower bound of $\tilde{O}(rn^2)$ in the fully observed setting and also provide guarantees under the partially observed setting, however, only in the batch setting. For the online setting, the work by He et al [21] presented an algorithm based on online $\ell_1$-minimization which also had good empirical performance. Using martingale techniques, the convergence of an online algorithm for the convex formulation of the robust PCA is presented in [18] but the result assumes the correctness of the batch convex formulation. The ReProCS algorithm and its variants are presented in [35] and the references therein. They study online robust PCA and online matrix completion under a unified framework but more restrictive assumptions are imposed on the low-rank and the sparse parts such as a slow changes in the support set of the outliers.

Note that the references mentioned above are by no means exhaustive - such a list is beyond the scope of this paper. They are merely a sample to provide some background and are representative of the state-of-the-art results.
3.2 Problem Setup

3.2.1 Model

We consider the popular spiked-covariance model with sparse perturbations in $n$-dimensions, ie, $x_t = A z_t + w_t + s_t$ where $A$ is an unknown $n \times r$ matrix of rank-$r$, $s_t$ is deterministic sparse perturbation with unknown support and magnitude, and $w_t$ is independent Gaussian noise with a given variance $\sigma^2 I$. Given a sample $x_t$ at time $t$, we wish to recover $s_t$ and with finite such samples, we wish to find the space spanned by the columns of $A$ upto a fixed numerical accuracy $\epsilon$. In other words, if $A = U \Sigma V^T$ is the SVD, we wish to find the eigenvectors $U$.

In this chapter, we will focus only on the rank-1 case, ie, at time instance $t$, we are given the data vector at time is given by $x_t = uz_t + s_t + w_t$ where $\|u\|_2 = 1$.

3.2.2 Notations and Assumptions

First, for simplicity and clarity, we will start by presenting the rank-1 case where the dense noise is absent, ie, $w_t = 0$. Specifically, the data vector at time is given by $x_t = uz_t + s_t$. Note that for simplicity we have assumed that the eigenvalue corresponding to the top eigenvector is 1. Note that we can decompose $u_h$ as $u_h = \pm (\sqrt{1 - \alpha_h} u + \sqrt{\alpha_h} v_h)$, where $u \perp v_h$ and $\alpha_h \in (0,1)$, for every $h \geq 1$. Now, we introduce natural (standard) conditions, similar to [44] under which the problem is identifiable:

1. Low-rank part: $u$ is $\mu$-incoherent, ie, $\|u\|_\infty \leq \frac{\mu}{\sqrt{r}}$ and $z_t \sim \mathcal{N}(0,1)$. We note that we may relax this generative assumption on $z_t$ to a random variable such that $\mathbb{E}[z_t] = 0$, $\mathbb{E}[z_t^2] = 1$, $|z_t| \leq Z_{\text{max}}$ almost surely with a little care.

2. Sparse part: we have a deterministic sparsity condition, ie, $\|s_t\|_0 \leq d_h$ and also without loss of generality, we assume $\|s_t\|_\infty \leq \frac{s_{\text{max}}}{\sqrt{n}}$. We assume that the sparsity increases
with epochs as \( d_h < \min \left( \frac{1}{100\mu^2 + 3\mu a_{h-1}}, \frac{1}{250Z_{\max}^2/\alpha_{h-1}} \right) \) where \( d_h \) is the number of non-zeros in \( s_t \) that appears at epoch \( h \). As a consequence of this inequality, we have the interpretation that as the number of epochs increases, we are able to tolerate more perturbation, ie, as \( u_h \) gets closer to \( u \), the thresholding become more effective.

Let \( b_i \) denote the \( i \)th basis vector in \( n \) dimensions. Define the entry-wise hard-thresholding operation of a vector \( v \), denoted as \( \text{Thresh}_a(v) \) as follows: for every \( i \),

\[
\text{Thresh}_a(b_i^Tv) = \begin{cases} 
  b_i^Tv, & \text{if } |b_i^Tv| > a \\
  0, & \text{else}
\end{cases}
\]

We define quantities \( Z_{h-1} = Z_{\max} \left( a_{h-1} \frac{a_{h-1}^2}{\sqrt{n}} + a_{h-1} (1 - a_{h-1}) \right) \) and for all \( \tau \), \( e_t^\tau = s_t - \hat{s}_t^\tau \) where \( \hat{s}_t^\tau \) is our estimate of \( s_t \) resulting from the \( \tau \)-th thresholding step. Let \( \hat{s}_t = \hat{s}_t^T \), \( e_t = e_t^T \) for every \( t \) and denote by \( E_h \) the matrix whose columns are \( e_t \) with \( t \) ranging from \( B(h-1)+1 \) to \( Bh \) (within epoch \( h \)). We note that the true covariance \( \Sigma = \mathbb{E}[(uz_t)(uz_t)^T] = uu^T \). Let \( \Sigma_h = \sum_{t=B(h-1)+1}^{Bh} \frac{1}{B}(x_t - \hat{s}_t)(x_t - \hat{s}_t)^T \) denote our estimate of the true covariance at epoch \( h \) and define \( \Delta_h = \Sigma - \Sigma_h \). Our aim is to show that \( \alpha_h \to 0 \) as \( h \) grows, with high probability.

We use \( C \), \( C_1 \), \( C_2 \), etc to denote global constants. \( \| \cdot \|_2 \) denotes the two-norm of a vector or the spectral norm of a matrix, \( \| \cdot \|_\infty \) denotes the maximum of the absolute values of the entries of a vector or a matrix, \( \| \cdot \|_1 \) is the sum of absolute values of entries of a vector, \( \| \cdot \|_0 \) is the number of non-zeros in a vector.

### 3.2.3 Algorithm

We present our algorithm for the rank-1 case in Algorithm 8. There are three key loops in Algorithm 8 namely, (1) (innermost) \( \tau \)-loop which we call alternations, (2) (middle) \( t \)-loop which we call iterations, and (3) the (outermost) \( h \)-loop which we call epochs. Our algorithm
uses random initialization for our eigenvector estimate, which is also very easy in practice. Intuitively, the $\tau$-loop is performs denoising via iterative hard-thresholding, ie, it solves the optimization problem:

$$\{\hat{z}_t, \hat{s}_t\} = \arg\min_{a \in \mathbb{R}, b \in \mathbb{R}^n} \|x_t - (ua + b)\|_2 \quad \text{s.t.} \quad \|b\|_0 \leq d_h$$

From this, we obtain an estimate of the sparse perturbation vector and consequently, the scaling factor associated with $u$. By subtracting this out, we obtain vector which is close to our desired subspace. Using a block of $B$ such vectors, the $t$-loop accumulates the sample covariance matrix. Finally, the $h$-loop performs a noisy power method update on the accumulated covariance matrix until our estimate reach the desired numerical accuracy $\epsilon$ with respect to the true eigenvector. Note that this is effectively a block version of the usual power method but the key challenge is to control the perturbation in the sample covariance estimate due to (1) the error induced by thresholding (ie, running only a finite number of alternations), and (2) the error in our estimate of the top eigenvector in the current epoch. Additionally, we note that samples are never revisited and hence this is a one-pass algorithm.

As described in Section 3.3, note that $s_{\max}$ is a constant which may be assumed to be known, without loss of generality.

**Remark 3.1.** Note that we don’t know $\alpha_{h-1}$ in practice and hence don’t know the exact bound on $Z_{h-1}$ but we will see that from Theorem 3.4 that a simple rule is to set $Z_{h-1} = C_1 \sqrt{n} C_2^{-(h-1)/2}$ where $C_1, C_2 > 0$ are constants. Algorithm 8 requires an access to an oracle that tracks the progress of the algorithm by giving us the value of $\alpha_{h-1}$ at epoch $h$. Though, for convenience, our analysis in Section 3.3 assume we have this oracle access, we may do away with this oracle requirement by carefully controlling the failure probability of the thresholding operation due to not knowing $\alpha_{h-1}$. To be precise, we may construct the inductive hypothesis that $Pr(\alpha_{h-1} \geq 0.9 \alpha_{h-2}) \leq (\frac{h-1}{3H^2})$, use this to bound the probability of the event $\|E_h\|_\infty \geq 4Z_{h-1} + \epsilon$ and reflect the same in the Section 3.3.2 for $\Delta_h$. Finally, while unrolling
the recursion for $\alpha_h$ in Theorem 3.4, we may use the union bound and obtain the failure probability as $Pr(\alpha_h \geq C^4 C_3^h n) \leq \frac{HC + \sum_{h=1}^{H} \alpha_h}{3H^2} = 1 - o(1)$.

**Algorithm 8** Block Stochastic Power Method with Hard Thresholding

1: **Input**: Samples $\{x_1, \ldots, x_T\} \in \mathbb{R}^n$ such that $x_t = u_{zt} + s_t$

2: **Output**: Leading eigenvector of the denoised samples $u_H$

3: $u'_0 \sim \mathcal{N}(0, I_{n \times n})$

4: $u_0 \leftarrow \frac{u'_0}{\|u'_0\|_2}$

5: **for** $h = 1, \ldots, H = \frac{T}{B}$ **do**

6: $u'_h \leftarrow 0$

7: **for** $t = B(h - 1) + 1, \ldots, Bh$ **do**

8: $\hat{s}^0_t \leftarrow 0$

9: **for** $\tau = 1, \ldots, T$ **do**

10: $\zeta^\tau_t \leftarrow 2Z_{h-1} + \frac{1}{5} \left( \frac{1}{10} \right) \tau \frac{s_{\max}}{\sqrt{n}}$

11: $\hat{z}^\tau_t \leftarrow u^\tau_{h-1}(x_t - \hat{s}^\tau_{t-1})$

12: $\hat{s}^\tau_t \leftarrow \text{Thresh}_{\zeta^\tau_t}(x_t - u^\tau_{h-1} \hat{z}^\tau_t)$

13: **end for**

14: $\hat{s}_t \leftarrow \hat{s}^T_t$

15: $u'_h \leftarrow u'_h + \frac{1}{B}(x_t - \hat{s}_t)(x_t - \hat{s}_t)^\top u_{h-1}$

16: **end for**

17: $u_h \leftarrow \frac{u'_h}{\|u'_h\|_2}$

18: **end for**

19: **return** $\hat{z}_1, \ldots, \hat{z}_T, \hat{s}_1, \ldots, \hat{s}_T, u_{T/B}$
3.3 Analysis

For simplicity and concreteness, we consider the analysis of only the noiseless rank-1 case. The noisy case analysis is very similar except that the result involves the noise level $\sigma$ of $w_t$ and concentration properties associated with that. We now present the main result for the noiseless case and present the proof details in Sections 3.3.1, 3.3.2, 3.3.3 that follow.

**Theorem 3.1.** Under the assumptions in Section 3.2.2, if $B \geq \frac{32Cn(\log H)^2}{\epsilon^2}$, $H \geq C_5 \log \left( \frac{n}{\epsilon} \right)$, $T > \log_{10} \left( \frac{C_1 \sqrt{n} \log(H) s_{\text{max}}}{\epsilon \sqrt{B}} \right)$, with probability at least $1 - 6C$, Algorithm 8 yields an $\epsilon$-close solution in the sense that $\alpha_H \leq \epsilon$.

**Proof.** This follows from Theorems 3.2, 3.3 and 3.4 and the initialization property in Lemma 3.1.

**Remark 3.2.** Note that we have a constant failure of probability. To get a high probability statement, we may boost our result by paying an extra $\log_{1/6C}(n)$ factor in space or sample complexity where $C < 1/6$. For example, if we have $\log_{1/6C}(n)$ independent runs, the probability that all of them fail is at most $(6C)^{\log_{1/6C}(n)} = \frac{1}{n}$. So, $\alpha_H \leq \epsilon$ with probability at least $1 - \frac{1}{n}$.

**Proof outline:** At a high level, the proof of convergence involves analyzing the three loops in Algorithm 8, namely: (1) convergence of (innermost) $\tau$-loop (alternations), (2) concentration properties in (middle) $t$-loop (iterations), and (3) convergence of (outermost) $h$-loop (epochs). We wish to emphasize that the concentration arguments are different from [39, 40] since we do not have any randomness assumptions on the support of the sparse perturbation. Lemma 3.1 quantifies the property of our initialization that is proved in Lemma 6 of [39] but we provide it here for completeness.

**Lemma 3.1.** The initialization given by Steps 3 and 4 of Algorithm 8 yields a vector $w_0$ such that $\sqrt{1 - \alpha_0} = O(1/\sqrt{n})$ with probability $1 - o(1)$. 

55
3.3.1 Convergence of the Innermost Loop

The main result of this section is the validity of the hard-thresholding operation, stated as:

**Theorem 3.2.** For every $t$, after $T > \log_{10} \left( \frac{s_{\max}}{\epsilon \sqrt{n}} \right)$ alternations we have $\|e_t\|_\infty \leq 4Z_{h-1} + \epsilon$.

**Proof.** This follows from Lemmas 3.2 and 3.3. 

**Lemma 3.2.** We have the following useful short results.

1. $\| (I - u_{h-1}u_{h-1}^\top) u \|_\infty \leq \alpha_{h-1} \frac{\mu}{\sqrt{n}} + \sqrt{\alpha_{h-1} (1 - \alpha_{h-1})}$.

2. $Z_{h-1} \leq 2Z_{\max} \sqrt{\alpha_{h-1}}$.

3. When $\alpha_{h-1} > \frac{1}{n}$, $\| u_{h-1} \|_\infty^2 \leq \frac{\mu^2 + 3\mu \alpha_{h-1}}{n}$. Else, $\| u_{h-1} \|_\infty^2 \leq \frac{4\mu^2}{n}$.

**Proof.**

1. Note that $|u^\top u_{h-1}| = \sqrt{1 - \alpha_{h-1}}$ and $\text{sign}(u^\top u_{h-1}), \text{sign}(b_i^\top u_{h-1}) = \text{sign}(b_i^\top u)$.

Thus,

$$\| u - (u_{h-1}^\top u) u_{h-1} \|_\infty = \| u - \sqrt{1 - \alpha_{h-1}} \left( \sqrt{1 - \alpha_{h-1}} u + \sqrt{\alpha_{h-1} (1 - \alpha_{h-1})} v_{h-1} \right) \|_\infty$$

$$\leq \alpha_{h-1} \| u \|_\infty + \sqrt{\alpha_{h-1} (1 - \alpha_{h-1})} \| v_{h-1} \|_\infty$$

$$\leq \alpha_{h-1} \frac{\mu}{\sqrt{n}} + \sqrt{\alpha_{h-1} (1 - \alpha_{h-1})}$$

2. We can obtain an upper bound on $Z_{h-1}$ as follows:

$$Z_{h-1} = Z_{\max} \left( \alpha_{h-1} \frac{\mu}{\sqrt{n}} + \sqrt{\alpha_{h-1} (1 - \alpha_{h-1})} \right)$$

$$\leq Z_{\max} \sqrt{\alpha_{h-1}} \left( \sqrt{\alpha_{h-1} \frac{\mu}{\sqrt{n}}} + \sqrt{1 - \alpha_{h-1}} \right)$$

By noting that $\alpha_{h-1} \in (0, 1)$ and $\mu \leq \sqrt{n}$, we obtain $Z_{h-1} \leq 2Z_{\max} \sqrt{\alpha_{h-1}}$. 

56
3. Since $\alpha_{h-1} \in (0, 1)$, we have $1 - \alpha_{h-1} < 1$. Using this in $\xi_1$,

$$
\|u_{h-1}\|_\infty^2 = \left\| \left( \sqrt{1 - \alpha_{h-1}} u + \sqrt{\alpha_{h-1}} v_{h-1} \right) \right\|_\infty^2 \\
\leq (1 - \alpha_{h-1}) \|u\|_\infty^2 + \alpha_{h-1} \|v_{h-1}\|_\infty^2 + 2\sqrt{\alpha_{h-1} (1 - \alpha_{h-1})} \|u\|_\infty \|v_{h-1}\|_\infty \\
\leq (1 - \alpha_{h-1}) \frac{\mu^2}{n} + \alpha_{h-1} + 2\sqrt{\alpha_{h-1} (1 - \alpha_{h-1})} \frac{\mu}{\sqrt{n}} \\
\xi_1 \frac{\mu^2}{n} + \alpha_{h-1} + 2\sqrt{\alpha_{h-1}} \frac{\mu}{\sqrt{n}} = \frac{\mu^2 + n\alpha_{h-1} + 2\mu \sqrt{n\alpha_{h-1}}}{n}
$$

When $\alpha_{h-1} > \frac{1}{n}$, we have $n\alpha_{h-1} > \sqrt{n\alpha_{h-1}}$. So, $\|u_{h-1}\|_\infty^2 \leq \frac{\mu^2 + 3n\alpha_{h-1}}{n}$ by noting $\mu \geq 1$.

\[ \square \]

Lemma 3.3. If $\|e_t^{-1}\|_\infty \leq 4Z_{h-1} + \left( \frac{1}{10} \right)^{\tau-1} \frac{s_{\max}}{\sqrt{n}}$, then we have:

1. $|b_t^\top (u z_t - u_{h-1} \hat{z}_t^\tau)| \leq \frac{26}{25} Z_{h-1} + \left( \frac{1}{10} \right)^{\tau+1} \frac{s_{\max}}{\sqrt{n}}$
2. $\|e_t^\tau\|_\infty \leq 4Z_{h-1} + \left( \frac{1}{10} \right)^{\tau} \frac{s_{\max}}{\sqrt{n}}$
3. Moreover, $\text{Supp} (e_t^\tau) \subseteq \text{Supp} (e_t^{-1}) \subseteq \text{Supp} (s_t)$. 

57
Proof. 1. $x_t-u_{h-1} \hat{z}_t^\tau = (uz_t+s_t)-u_{h-1} \hat{z}_t^\tau \implies |b_i^\top (x_t-u_{h-1} \hat{z}_t^\tau -s_t)| = |b_i^\top (uz_t-u_{h-1} \hat{z}_t^\tau)|$.

$$|b_i^\top (uz_t-u_{h-1} \hat{z}_t^\tau)| \overset{\xi_1}{=} |b_i^\top (uz_t-u_{h-1}u_{h-1}^\top (x_t-s_t))|$$

$$\overset{\xi_2}{=} |b_i^\top (uz_t-u_{h-1}u_{h-1}^\top (uz_t+e_i^{\tau-1}))|$$

$$\overset{\xi_3}{=} |b_i^\top (uz_t-u_{h-1}u_{h-1}^\top uz_t)| + |b_i^\top u_{h-1}u_{h-1}^\top e_i^{\tau-1}|$$

$$\leq Z_{\max} \max_i |u - (u_{h-1}^\top u)u_{h-1}| + \max_i |b_i^\top u_{h-1}u_{h-1}^\top e_i^{\tau-1}|$$

$$\overset{\xi_4}{=} Z_{\max} \left( \alpha_{h-1} \frac{\mu}{\sqrt{n}} + \sqrt{\alpha_{h-1} (1-\alpha_{h-1})} \right) + d_h \|u_{h-1}\|_\infty \|e_i^{\tau-1}\|_\infty$$

$$\leq Z_{h-1} + \frac{1}{100} \|e_i^{\tau-1}\|_\infty \overset{\xi_5}{\leq} Z_{h-1} + \frac{1}{100} \left( 4Z_{h-1} + \left( \frac{1}{10} \right)^{\tau-1} \frac{s_{\max}}{\sqrt{n}} \right)$$

$$\leq \frac{26}{25} Z_{h-1} + \left( \frac{1}{10} \right)^{\tau+1} \frac{s_{\max}}{\sqrt{n}}$$

where $\xi_1$ is by substituting $\hat{z}_t^\tau = u_{h-1}^\top (x_t-s_t)$, $\xi_2$ by recalling the definition that $e_i^{\tau-1} = s_t-s_t^{\tau-1}$, $\xi_3$ by triangle inequality, $\xi_4$ by using $|\langle a, b \rangle| \leq \|a\|_\infty \|b\|_1$, $\xi_5$ by Lemma 3.2-(1) and noting that $\|e_i^{\tau-1}\|_1 \leq d_h \|e_i^{\tau-1}\|_\infty$, $\xi_6$ by using the definition of $Z_{h-1}$ and the assumption on $d_h$, $\xi_7$ by inductive hypothesis that $\|e_i^{\tau-1}\|_\infty \leq 4Z_{h-1} + \left( \frac{1}{10} \right)^{\tau-1} \frac{s_{\max}}{\sqrt{n}}$.

2. Next, to complete the induction over $\tau$, let us calculate $\|e_i^\tau\|_\infty$. We have two cases

(a) **Case 1 ($|b_i^\top (x_t-u_{h-1} \hat{z}_t^\tau)| > \zeta_i^\tau$):**

$$|b_i^\top e_i^\tau| = |b_i^\top (s_t-s_t^{\tau})| = |b_i^\top (s_t-(x_t-u_{h-1} \hat{z}_t^\tau))|$$

$$= |b_i^\top uz_t-u_{h-1} \hat{z}_t^\tau| \leq \frac{26}{25} Z_{h-1} + \left( \frac{1}{10} \right)^{\tau+1} \frac{s_{\max}}{\sqrt{n}}$$

(b) **Case 2 ($|b_i^\top (x_t-u_{h-1} \hat{z}_t^\tau)| \leq \zeta_i^\tau$: $b_i^\top \hat{z}_t^\tau = 0 \implies b_i^\top e_i^\tau = b_i^\top s_t$ and $|b_i^\top (x_t-u_{h-1} \hat{z}_t^\tau)| = \frac{26}{25} Z_{h-1} + \left( \frac{1}{10} \right)^{\tau+1} \frac{s_{\max}}{\sqrt{n}}$**
\[ |b_i^\top(uz_t + s_t - u_{h-1}z_t^\tau)| \leq \zeta_t^\tau. \] So, we have

\[ |b_i^\top e_i^\tau| = |b_i^\top s_t| \leq \zeta_t^\tau + |b_i^\top(uz_t - u_{h-1}z_t^\tau)| \]

\[ \leq \left( 2Z_{h-1} + \frac{1}{5} \left( \frac{1}{10} \right)^\tau s_{\max} \sqrt{n} \right) + \left( \frac{26}{25}Z_{h-1} + \left( \frac{1}{10} \right)^{\tau+1} s_{\max} \sqrt{n} \right) \]

\[ = \frac{76}{25}Z_{h-1} + \left( \frac{1}{5} + \frac{1}{10} \right) \left( \frac{1}{10} \right)^{\tau} s_{\max} \sqrt{n} \leq 4Z_{h-1} + \left( \frac{1}{10} \right)^{\tau} s_{\max} \sqrt{n}. \]

3. If \( b_i^\top s_t = 0 \), then we have \( b_i^\top e_i^\tau = I_{\{b_i^\top(uz_t - u_{h-1}z_t^\tau)\} > \zeta_t^\tau (b_i^\top(uz_t - u_{h-1}z_t^\tau)) \} \) but note that

\[ |b_i^\top(uz_t - u_{h-1}z_t^\tau)| \leq \frac{26}{25}Z_{h-1} + \left( \frac{1}{10} \right)^{\tau+1} s_{\max} \sqrt{n} < 2Z_{h-1} + \frac{1}{5} \left( \frac{1}{10} \right)^{\tau} s_{\max} \sqrt{n} = \zeta_t^\tau. \]

This is a contradiction since the indicator is inactive at location \( i \), so \( b_i^\top e_i^\tau = 0. \)

\[ \square \]

### 3.3.2 Concentration Properties in the Middle Loop

We analyze the concentration properties of many iterations within a single epoch, ie, with enough number of samples, the covariance within a block concentrates. In this step, it is essential to show that the covariance in a single epoch converges to the true covariance plus a perturbation that depends on the sparse perturbation and is decaying as epochs proceed, so that this estimate may then be used for block power method updates. We note that the index \( t \) in this section runs from \( B(h-1)+1 \) to \( Bh \) and to simplify notation, we will omit this range in the summations. Thus, the main result of this section is:

**Theorem 3.3.** Setting \( T > \log_{10} \left( \frac{C_1 \sqrt{n} \log(H) s_{\max}}{\epsilon \sqrt{B}} \right) \) for every \( t \), letting \( B \geq \frac{32Cn(\log H)^2}{\epsilon^2} \), with probability \( 1 - \frac{6C}{H} \), we have \( \|\Delta_h\|_2 \leq \epsilon + 100d_h Z_{\max}^2 \alpha_{h-1}. \)
Proof. We have $\Delta_h = \Sigma_h - \Sigma$.

$$\|\Sigma_h - \Sigma\|_2 = \left\| \sum_{t=1}^{B} \frac{1}{B} (x_t - \hat{s}_t)(x_t - \hat{s}_t)^\top - uu^\top \right\|_2 = \left\| \frac{1}{B} \sum_t (uz_t + e_t)(uz_t + e_t)^\top - uu^\top \right\|_2 \leq \frac{1}{B} \left\| uu^\top \sum_t (z_t^2 - 1) \right\|_2 + \frac{1}{B} \left\| u \sum_t z_te_t^\top \right\|_2 + \frac{1}{B} \left\| \sum_t z_te_t^\top \right\|_2 + \frac{1}{B} \left\| \sum_t e_t^\top e_t \right\|_2$$

The second step is by triangle inequality on the spectral norm of the perturbed matrix. Now, we bound each of the terms using similar techniques as [39].

**Term-1:** Using tail bounds for sub-Gaussian random variables from [52], with probability at least $1 - \frac{2C}{H}$

$$\frac{1}{B} \left\| uu^\top \sum_t (z_t^2 - 1) \right\|_2 \leq \frac{1}{B} \sum_t (z_t^2 - 1) \left\| uu^\top \right\|_2 \leq \sqrt{\frac{C \log H}{B}}$$

**Term-2:** As spectral norm is sub-multiplicative, $\|\sum_t uz_te_t^\top\|_2 = \|u(E_hz)^\top\|_2 \leq \|u\|_2 \|E_hz\|_2$. Now, for every $i$, since $z_i \sim \mathcal{N}(0, 1)$, we have $b_i^\top E_hz \sim \mathcal{N}(0, \sigma_e^2)$ where $\sigma_e^2 \leq B \|E_h\|_\infty^2$; this is because $\text{var}(b_i^\top E_hz) = \text{var} \left( \sum_{j=1}^{B} b_i^\top E_hz_j^\top z \right) \leq B \|E_h\|_\infty^2$. Hence, with probability $1 - \frac{2C}{H}$

$$\|E_hz\|_2 = \sqrt{\sum_{i=1}^{n} (b_i^\top E_hz)^2} \leq \sqrt{n (b_i^\top E_hz)^2} \leq \sqrt{n} \sqrt{2\sigma_e \log \left( \frac{H}{C} \right)}$$

where the last line was obtained by using the Hoeffding bound, i.e., tail bound for $X \sim \mathcal{N}(0, \sigma_e^2)$ is given by $Pr(-t \leq X \leq t) \leq 1 - 2 \exp \left( -\frac{t^2}{2\sigma^2} \right)$ and noting that $\sqrt{X^2}$ is half-normal distribution satisfying this bound. Further simplifying by substituting $\|E_h\|_\infty$, using Theorem 3.2 and Lemma 3.2-(2), we get

$$\|E_hz\|_2 \leq \sqrt{2nB \|E_h\|_\infty \log \left( \frac{H}{C} \right)} \leq \sqrt{2nB \log \left( \frac{H}{C} \right)} (8Z_{\max} \sqrt{\alpha_{h-1}} + \epsilon)$$
Dividing both sides by $B$, we obtain \( \frac{1}{B} \| E_h z \|_2 \leq \sqrt{\frac{2n}{B}} \log \left( \frac{H}{C} \right) \left( 8 Z_{\text{max}} \sqrt{\alpha_{h-1}} + \epsilon \right) \).

**Term-3:** Same as Term-2.

**Term-4:** Let $\epsilon \leq Z_{h-1} \leq 2 Z_{\text{max}} \sqrt{\alpha_{h-1}}$. Using triangle inequality, sub-multiplicative property, Theorem 3.2 and Lemma 3.2-(4), we have

\[
\frac{1}{B} \left\| \sum_t e_t e_t^\top \right\|_2 \leq \frac{1}{B} \sum_t \| e_t e_t^\top \|_2 \leq \frac{1}{B} \sum_t \| e_t \|_2 \leq \frac{1}{B} B \left( \sqrt{d} \| e_t \|_\infty \right)^2 \leq d_h \| E_h \|_\infty
\]

\[
\leq d_h (4 Z_{h-1} + \epsilon)^2 \leq d_h (25 Z_{h-1}^2) \leq 100 d_h Z_{\text{max}}^2 \alpha_{h-1}
\]

Note that by setting $T > \log_{10} \left( \frac{C_1 \sqrt{n \log(H) s_{\text{max}}}}{\epsilon \sqrt{B}} \right)$ we have $\| e_t \|_\infty \leq 4 Z_{h-1} + \frac{\epsilon \sqrt{B}}{C_1 n \log H}$ where $C_1$ is a constant. Using this, combining all the terms, letting $B \geq 32 C_1 n (\log H)^2$ and assuming $\alpha_{h-1} \leq \frac{1}{2^{56} Z_{\text{max}}}$, we obtain, with probability $1 - \frac{6C}{H}$:

\[
\| \Delta_h \|_2 = \| \Sigma_h - \Sigma \|_2
\]

\[
\leq \sqrt{\frac{C \log H}{B}} + 2 \sqrt{\frac{2n}{B}} \log \left( \frac{H}{C} \right) \left( 8 Z_{\text{max}} \sqrt{\alpha_{h-1}} + \frac{\epsilon \sqrt{B}}{C_1 n \log H} \right) + 100 d_h Z_{\text{max}}^2 \alpha_{h-1}
\]

\[
\leq \epsilon + 100 d_h Z_{\text{max}}^2 \alpha_{h-1}
\]

\[\square\]

### 3.3.3 Convergence of the Outermost Loop

The goal here is to show that $\alpha_h \to 0$ by quantifying the improvement (decrease) of $\alpha_h$ over $\alpha_{h-1}$. The main result for this section is:

**Theorem 3.4.** If $H \geq C_5 \log \left( \frac{n}{\epsilon} \right)$ with probability atleast $1 - 6C$, we obtain $\alpha_H \leq \epsilon$ where $C_5$ and $C$ are constants.
Applying Lemmas 2, 6 of [39], we get
\[ \alpha \]
Note that if \( H \) holds with probability at least \( 1 - \epsilon \), using Lemma 3.4 and assuming \( 10\epsilon < \sqrt{\alpha_{\Delta h}} \), we have,

\[ \alpha_h = \langle u_h, v_h \rangle^2 = \left\langle \frac{u'_h}{\|u'_h\|_2}, v_h \right\rangle^2 = \frac{\langle u'_h, v_h \rangle^2}{\|u'_h\|_2^2} = \frac{\langle u'_h, v_h \rangle^2}{\langle u'_h, u \rangle^2 + \langle u'_h, v_h \rangle^2} \]

\[ \leq \frac{(\epsilon + 100d_hZ^2_{\max}\alpha_{\Delta h})^2}{(\sqrt{1 - \alpha_{\Delta h}} \left( 1 - \epsilon - \sqrt{\frac{\alpha_{\Delta h}}{1-\alpha_{\Delta h}}}100d_hZ^2_{\max}\alpha_{\Delta h} \right))^2 + (\epsilon + 100d_hZ^2_{\max}\alpha_{\Delta h})^2} \]

The second inequality above is obtained by noting that \( \frac{x}{c+x} \) is an increasing function in \( x \) for positive \( x \) and \( c \). Let \( C_3 = \left( (0.1 + 100d_hZ^2_{\max}\alpha_{\Delta h}) / \left( 0.9 - \sqrt{\frac{\alpha_{\Delta h}}{1-\alpha_{\Delta h}}}100d_hZ^2_{\max}\sqrt{\alpha_{\Delta h}} \right) \right)^2 \).

Note that if \( d_h < 1/250Z^2_{\max}\sqrt{\alpha_{\Delta h}} \), we note that the constant \( C_3 < 1 \). Using this and also applying Lemmas 2, 6 of [39], we get \( \alpha_h \leq \frac{\xi_1}{1 - \alpha_{\Delta h}} \leq \frac{\xi_2}{1 - \alpha_{\Delta h}} \leq C_4C_3^h n \) where \( \xi_1 \) holds with probability at least \( 1 - \frac{6C}{H} \) and where \( \xi_2 \) holds with probability at least \( 1 - \frac{9hC}{H} \)
where the factor of \( h \) comes by accounting for the failure of at least one epoch followed by applying the union bound. Hence, if \( H \geq \log_{1/C_3}(C_4C_3^h) \) with probability at least \( 1 - 6C \), we obtain \( \alpha_H \leq \epsilon \).

**Lemma 3.4.** We have the following upper and lower bounds.

1. \( \langle u'_h, v_h \rangle \leq \epsilon + 100d_hZ^2_{\max}\alpha_{\Delta h} \).
2. \( \langle u'_h, u \rangle \geq \sqrt{1 - \alpha_{\Delta h}} \left( 1 - \epsilon - \sqrt{\frac{\alpha_{\Delta h}}{1-\alpha_{\Delta h}}}100d_hZ^2_{\max}\alpha_{\Delta h} \right) \).

**Proof.**

1. Recall that \( \langle u, v_h \rangle = 0 \). Now,

\[ \langle u'_h, v_h \rangle = v_h^\top \Sigma_h u_{\Delta h} = v_h^\top (\Sigma + \Delta_h) u_{\Delta h} = v_h^\top uu^\top u_{\Delta h} + v_h^\top \Delta_h u_{\Delta h} \]

\[ \leq 0 + \|v_h\|_2 \|\Delta_h\|_2 \|u_{\Delta h}\|_2 = \epsilon + 100d_hZ^2_{\max}\alpha_{\Delta h} \]
2. Next, we have lower bound the following term since it would appear in the denominator.

\[
\langle u_h', u \rangle = u^\top \Sigma_h u_{h-1} = u^\top \frac{1}{B} \sum_t (x_t - \hat{s}_t) (x_t - \hat{s}_t)^\top u_{h-1}
\]

\[
= \frac{1}{B} \sum_t (u^\top (uz_t + e_t)) \left( (uz_t + e_t)^\top \left( \sqrt{1 - \alpha_{h-1}} u + \sqrt{\alpha_{h-1}} v_{h-1} \right) \right)
\]

\[
= \frac{1}{B} \sum_t (z_t + u^\top e_t) \left( \sqrt{1 - \alpha_{h-1}} z_t + \sqrt{1 - \alpha_{h-1}} e_t^\top u + \sqrt{\alpha_{h-1}} e_t^\top v_{h-1} \right)
\]

\[
= \frac{\sqrt{1 - \alpha_{h-1}}}{B} \sum_t (z_t + u^\top e_t)^2 + \frac{\sqrt{\alpha_{h-1}}}{B} \sum_t (z_t + u^\top e_t) (e_t^\top v_{h-1})
\]

**Term-5:** With probability \(1 - \frac{2C}{H}\), using the settings for \(B\) and \(T\) from Section 3.3.2, and the upper bound for Term-2 with a negative sign (since this is an absolute value of scalar),

\[
\frac{1}{B} \sum_t (z_t + u^\top e_t)^2 = \frac{1}{B} \sum_t z_t^2 + \frac{1}{B} \sum_t (u^\top e_t)^2 + \frac{2}{B} \sum_t z_t u^\top e_t
\]

\[
\geq 1 - \sqrt{\frac{C \log H}{B}} + 0 + \frac{2}{B} u^\top E_h z \geq 1 - \frac{\epsilon}{4} - \frac{2}{B} \|u\|_2 \|E_h z\|_2
\]

\[
\geq 1 - \frac{\epsilon}{2}
\]

**Term-6:** This is similar to spectral norm upper bounds in Step-2 but with a negative sign, ie,

\[
\frac{1}{B} \sum_t (z_t + u^\top e_t) (e_t^\top v_{h-1}) \leq \frac{1}{B} \|v_{h-1}^\top E_h z\| + \frac{1}{B} \|u^\top E_h E_h^\top v_{h-1}\|
\]

\[
\leq \frac{\epsilon}{2} + 100d_h z_{\text{max}}^2 \alpha_{h-1}
\]
Thus, from Terms-5 and 6, and noting $\alpha_{h-1} \leq \frac{1}{4}$, we have,

\[
\langle u'_h, u \rangle \geq \sqrt{1 - \alpha_{h-1}} \left( 1 - \frac{\epsilon}{2} \right) - \sqrt{\alpha_{h-1}} \left( \frac{\epsilon}{2} + 100d_h Z_{\text{max}}^2 \alpha_{h-1} \right) \\
= \sqrt{1 - \alpha_{h-1}} \left( 1 - \left( 1 + \sqrt{\frac{\alpha_{h-1}}{1 - \alpha_{h-1}}} \right) \frac{\epsilon}{2} - \sqrt{\frac{\alpha_{h-1}}{1 - \alpha_{h-1}}} 100d_h Z_{\text{max}}^2 \alpha_{h-1} \right) \\
\geq \sqrt{1 - \alpha_{h-1}} \left( 1 - \epsilon - \sqrt{\frac{\alpha_{h-1}}{1 - \alpha_{h-1}}} 100d_h Z_{\text{max}}^2 \alpha_{h-1} \right)
\]

\[\square\]
3.4 Conclusion and Future Work

In this chapter, we have presented the first convergence result for the robust PCA problem in the streaming setting under the most general assumptions compared to previous works mentioned in Section 3.1.2. Extending the results in this paper to the rank-$r$ case should be possible along similar lines but we note two points: (1) a naïve analysis would lead to loose bounds and hence care must be taken in accounting for the $r$-eigenvectors while using the distance between subspaces to track the progress of the algorithm, and (2) we suspect that the convergence guarantee for the rank-$r$ analogue of Algorithm 8 (ie, via block stochastic orthogonal iteration with hard thresholding) will have a sub-optimal dependence on the condition number and hence one plausible fix would be to consider the streaming version of the stage-wise algorithm in [44]. We defer these analyses of the rank-$r$ case to future work. Though the main focus of this paper was to obtain convergence guarantees, we wish to note that potential applications include real-time background-foreground separation in videos and real-time subspace tracking similar to [21].
Chapter 4

Inductive Robust PCA

Abstract

The robust PCA problem, wherein, given an input data matrix that is the superposition of a low-rank matrix and a sparse matrix, we aim to separate out the low-rank and sparse components, is a well-studied problem in machine learning. One natural question that arises is that, as in the inductive setting, if features are provided as input as well, can we hope to do better? Answering this in the affirmative, the main goal of this chapter is to study the robust PCA problem while incorporating feature information. In contrast to previous works in which recovery guarantees are based on the convex relaxation of the problem, we propose a simple iterative algorithm based on hard-thresholding of appropriate residuals. Under weaker assumptions than previous works, we prove the global convergence of our iterative procedure; moreover, it admits a much faster convergence rate and lesser computational complexity per iteration. In practice, through systematic synthetic and real data simulations, we confirm our theoretical findings regarding improvements obtained by using feature information.
4.1 Introduction

Principal Component Analysis (PCA) [46] is a very fundamental and ubiquitous technique for unsupervised learning and dimensionality reduction; basically, this involves finding the best low-rank approximation to the given data matrix. To be precise, one common formulation of PCA is the following:

\[
\hat{L} = \arg \min_{L} \| M - L \|_F \quad \text{s.t.} \quad \text{rank}(L) \leq r
\]

(4.1)

where \( M \in \mathbb{R}^{n_1 \times n_2} \) is the input data matrix, where \( \| . \|_F \) denotes the Frobenius norm of a matrix and \( 1 \leq r \leq \min(n_1, n_2) \). It is well-known that the constrained optimization problem given by Equation (4.1) can be solved via the Singular Value Decomposition (SVD) and truncating the resultant decomposition to the top-\( r \) singular values and singular vectors yields the optimal solution [17]. While this machine learning technique has umpteen number of applications, one of its main shortcomings is that it not robust to the presence of gross outliers since the optimization involves just an \( \ell_2 \) objective. To address this issue, the robust PCA technique – given \( M \) such that \( M = L^* + S^* \), our aim is to find \( L^* \) and \( S^* \) which are low-rank and sparse matrix components respectively – was developed. Precisely, one hopes to solve the following problem (or its equivalent formulations):

\[
\{ \hat{L}, \hat{S} \} = \arg \min_{L, S} \| M - L - S \|_F \\
\text{s.t.} \quad \text{rank}(L) \leq r, \quad \| S \|_0 \leq z_0
\]

(4.2)

where \( \| . \|_0 \) denotes the number of non-zero entries in a matrix, \( 0 \leq r \leq \min(n_1, n_2) \) and \( 0 \leq z_0 \leq n_1 n_2 \). While Equation (4.2) may not be always well-posed, under certain identifiability conditions, many recent works over the past decade have advanced our understanding of this problem; we briefly recap some of the existing relevant results in Section 4.1.2.
Table 4.1: Comparison of this work to previous robust PCA works. For simplicity and brevity, we let $n_1 = n_2 = n$ and $d_1 = d_2 = d$; let the number of non-zeros per row/column of $S^*$ be $z$ and the number of non-zero entries in $S^*$ be $m$; we use $\tilde{O}$ to suppress log factors. Note that we consider the practically important regime of $d \ll n$.

<table>
<thead>
<tr>
<th>Work</th>
<th>Features</th>
<th>Approach</th>
<th>Incoherence</th>
<th>Sparsity</th>
<th>Sample complexity</th>
<th>Comp. complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>[5]</td>
<td>×</td>
<td>Convex</td>
<td>Strong</td>
<td>Random</td>
<td>$m = O(n^2), r = O(n)$</td>
<td>$O\left(\frac{n^2}{\epsilon}\right)$</td>
</tr>
<tr>
<td>[23]</td>
<td>×</td>
<td>Convex</td>
<td>Weak</td>
<td>Deterministic</td>
<td>$z = O\left(\frac{n}{\epsilon}\right)$</td>
<td>$O\left(\frac{n^2}{\epsilon}\right)$</td>
</tr>
<tr>
<td>[44]</td>
<td>×</td>
<td>Non-convex</td>
<td>Weak</td>
<td>Deterministic</td>
<td>$z = O\left(\frac{n}{\epsilon}\right)$</td>
<td>$O\left(rn^2 \log\left(\frac{1}{\epsilon}\right)\right)$</td>
</tr>
<tr>
<td>[55]</td>
<td>×</td>
<td>Non-convex</td>
<td>Weak</td>
<td>Deterministic</td>
<td>$z = O\left(\frac{n}{\epsilon}\right)$</td>
<td>$O\left(rn^2 \log\left(\frac{1}{\epsilon}\right)\right)$</td>
</tr>
<tr>
<td>[11]</td>
<td>✓</td>
<td>Convex</td>
<td>Strong</td>
<td>Random</td>
<td>$m = O(n^2), r = \tilde{O}\left(\frac{n^2}{\epsilon}\right)$</td>
<td>$O\left(dn^2 + d^2 r \log\left(\frac{1}{\epsilon}\right)\right)$</td>
</tr>
</tbody>
</table>

**4.1.1 Robust Inductive Learning: Motivation**

A key point to be noted is that Equation (4.2) does not incorporate feature information; this is the so-called transductive setting. In practical applications, we often have feature information available in the form of feature matrices $F_1$ and $F_2$. In the low-rank matrix recovery literature, this is often incorporated as a bilinear form, $L^* = F_1^\top W^* F_2$, which models the feature interactions via the latent space characterized by matrix $W^* \in \mathbb{R}^{d_1 \times d_2}$; this is the so-called inductive setting. We now present a motivating real-life situation.

**Example 4.1 (Using features for collaborative filtering with grossly corrupted observations).** In recommendation systems, it is often the case that we have user-product ratings matrix along with side information in the form of features corresponding to each user and product. It is common in large-scale machine learning applications that the number of products and users is very large compared to the features available for each user or product. Though a user might not have used a product, we would like to infer a how the user might rate that product given the user and product features – unlike the transductive setting this is possible in, and is a key application of, the inductive learning setting. Moreover, the ratings matrix is subject to various kinds of noise including erasures and outliers – in this work, we consider a general noise model using which robust recovery of ratings is possible.

It is the goal of this chapter to focus on the practically useful regime of $\max(d_1,d_2) \ll
\[
\min(n_1, n_2).
\]

## 4.1.2 Related Work

We now present the related work in both transductive and inductive settings.

**Transductive setting:** This is the relatively more well-explored setting. There are two main solution approaches that have been considered in the literature namely, the convex and the non-convex methods.

Convex methods entail understanding the properties of the convex relaxation of Equation (4.2) given by:

\[
\begin{align*}
\{\hat{L}, \hat{S}\} = \arg \min_{L, S} & \|L\|_* + \lambda \|S\|_1 \\
\text{s.t.} & \quad M = L + S
\end{align*}
\]

(4.3)

The works of [8] and [23] characterize the recovery properties of the convex program assuming a weak deterministic assumption on the support of the sparse matrix that the fraction of corrupted entries; the tightest bounds are that this fraction scales as \(O(1/r)\). Under a stronger model of the sparse matrix namely, uniformly sampled support, [5] show that it is possible to have \(r = O(n/\log(n))\) when \(z_0 = O(n^2)\) for exact recovery with high probability. Numerically, the convex program in Equation (4.3) is most commonly solved by variants of sub-gradient descent (involving iterative soft-thresholding); the convergence rate known for trace-norm programs is \(O(1/\sqrt{\epsilon})\) [27] for an \(\epsilon\)-close solution.

The underlying theme in non-convex methods involves retaining the formulation in Equation (4.2), starting with a suitable initialization and performing alternating projections onto non-convex sets (involving iterative hard-thresholding) until convergence. The work of [44]
provides recovery guarantees under the weaker deterministic support assumptions matching the conditions of [23]. However, the computational complexity of their algorithm scales with rank quadratically – to improve this, [55] propose a (non-convex) projected gradient approach while paying a cost in the permissible number of sparse corruptions, i.e., $O(1/r^{1.5})$ as opposed to $O(1/r)$. A consequence of the analysis of these non-convex methods is that they admit a faster convergence rate – specifically, $O(\log(1/\epsilon))$ iterations for an $\epsilon$-close solution – as opposed to convex methods.

It is noteworthy that the matrix completion problem (see, for instance, [49] and [26]), where the goal is to recover an incomplete low-rank matrix, is a special case of the robust PCA problem where $S^*$ is taken to be $-L^*$ for the non-observed entries. Finally, we note that the robust PCA problem has been invoked in several applications including topic modeling [38], object detection [34] and so on.

**Inductive setting:** To the best of our knowledge, currently, there is only one other work due to [11] which considers the robust PCA problem in the inductive setting and presents a guaranteed convex optimization procedure for solving it; incorporating additional feature information is into the robust PCA problem, they solve the following convex program, known as $PCPF$:

$$\{\hat{W}, \hat{S}\} = \arg\min_{W,S} \|W\|_* + \lambda \|S\|_1$$

s.t. $M = F_1^TWF_2 + S$  \hspace{1cm} (4.4)

For this paragraph, let $m := \|S^*\|_0$, $W^* = U_{W^*} \Sigma_{W^*} V_{W^*}^T$ be the SVD of $W^*$, $F_1F_1^T = I$, $F_2F_2^T = I$ and $e_i$ denote the $i^{th}$ standard basis vector in $\mathbb{R}^n$; the key recovery guarantee states that $r = O(n^2/d \log(n) \log(d))$ and $m = O(n^2)$; most notably, these guarantees are derived under stronger assumptions namely, (1) strong incoherence property, i.e., $\|U_{W^*} V_{W^*}^T\|_\infty \leq$
\[ \mu \sqrt{r/n_1n_2}, \max_j \|U_{W^T}^T F_1 e_j\|_2 \leq \mu_0 \sqrt{r/n_1}, \max_j \|V_{W^T}^T F_2 e_j\|_2 \leq \mu_0 \sqrt{r/n_2}, \max_j \|F_1 e_j\|_2 \leq \mu_{F_1} \sqrt{d/n_1}, \max_j \|F_2 e_j\|_2 \leq \mu_{F_2} \sqrt{d/n_2} \]

(2) random sparsity, i.e., the support of \( S^* \) is drawn uniformly at random from all subsets of \([n_1] \times [n_2]\) of size \( m \). Note that assumptions such as uniform support sampling may not be realistic in practice. In contrast, as we explain in Sections 4.1.3 and 4.2.2, our work relaxes the assumptions they require while admitting a simpler algorithm, novel analysis approach and faster convergence result.

In this context, it is also to be mentioned that for the related problem of inductive matrix completion is relatively better understood; recovery guarantees are known for both the convex (see, for instance, [54] and [10]) and the non-convex (e.g., [24]) approaches.

To summarize, we position this chapter with respect to other works in Table 4.1. While we have highlighted the most relevant existing results, note that the list provided here is by no means comprehensive – such a list is beyond the scope of this work.

### 4.1.3 Our Contributions

To the best of our knowledge, our work is the first to derive a provable and efficient non-convex method for robust PCA in the inductive setting. Our novelty and technical contributions can be summarized along the following axes:

1. **Assumptions (Section 4.2.2):** We use the weakest assumptions, i.e., (1) weak incoherence conditions on only the feature matrices and (2) (weak) deterministic support of the sparse matrix.
2. **Algorithm (Section 4.2.4):** Our algorithm (IRPCA-IHT) performs simple steps involving spectral and entry-wise hard-thresholding operations.
3. ** Guarantees (Sections 4.3.2, 4.3.3 and 4.3.4):** We show \( \epsilon \)-close recovery in both the noiseless and noisy cases for problems of general size, feature dimension, rank and
sparsity; moreover, our method has the fast (linear) convergence property.

4. *Experiments (Section 4.4)*: We substantiate our theoretical results by demonstrating gains on both synthetic and real-world experiments.
4.2 Problem Setup

4.2.1 Notation and Preliminaries

Let \( M = L^* + S^* \), ie, \( \{M, L^*, S^*\} \in \mathbb{R}^{n_1 \times n_2} \) are matrices such that the input data matrix \( M \) is the superposition of two component matrix signals namely, the low-rank component \( L^* \) and the sparse component \( S^* \). Here, \( S^* \) is a sparse perturbation matrix with unknown (deterministic) support and arbitrary magnitude. In our inductive setting, side information or features are present in the bilinear form specified \( L^* = F_1^\top W F_2 \). The feature matrices are denoted as \( F_1 \in \mathbb{R}^{d_1 \times n_1} \) and \( F_2 \in \mathbb{R}^{d_2 \times n_2} \). Note that the feature dimensions are \( d_1 \) and \( d_2 \) such that \( \max(d_1, d_2) \ll \min(n_1, n_2) \) and \( W^* \in \mathbb{R}^{d_1 \times d_2} \) is the rank-\( r \) latent matrix to be estimated where \( r \leq \min(d_1, d_2) \); intuitively, this latent matrix parameter describes the interaction and correlation among the feature vectors. Now, our optimization problem is given by:

\[
\{\hat{W}, \hat{S}\} = \arg \min_{W, S} \|M - F_1^\top W F_2 - S\|_F
\]

\[
\text{s.t. \ rank}(W) \leq r, \|S\|_{0,\infty} \leq z_2, \|S\|_{\infty,0} \leq z_1
\]

(4.5)

Here, for a matrix \( A \in \mathbb{R}^{n_1 \times n_2} \), we define the relevant norms, \( \|A\|_{0,\infty} := \max_i \sum_j \mathbb{1}(A_{ij} \neq 0) \), \( \|A\|_{\infty,0} := \max_i \sum_j \mathbb{1}(A_{ij} \neq 0) \), \( \|A\|_{\infty} := \max_{ij} |A_{ij}| \), Frobenius norm \( \|A\|_F := \sqrt{\sum_{i=1}^{n_1} \sum_{j=1}^{n_2} A_{ij}^2} \), spectral norm \( \|A\|_2 = \max_{\|x\|_2=1,\|y\|_2=1} x^\top A y \) for unit vectors \( x \in \mathbb{R}^{n_1} \) and \( y \in \mathbb{R}^{n_2} \). Next, for a matrix \( A \), we denote its maximum and minimum singular value by \( \sigma_{\max}(A) \) and \( \sigma_{\min}(A) \) respectively, and further the condition number of \( A \) is denoted by \( \kappa(A) := \sigma_{\max}(A)/\sigma_{\min}(A) \). The pseudoinverse of a matrix \( A \) is denoted by \( B = A^\dagger \) and is computed as \( B := (A^\top A)^{-1} A^\top \) where \( A \) is assumed to be of full rank. Let \( I \) denote the identity matrix whose size is clear from the context. Finally, we use \( e_i \) to denote the \( i^{th} \) standard basis vector in the appropriate dimension, which is also clear from the context.
Remark 4.1 (Noisy case: motivation and setup). Note that, so far, for simplicity and clarity, we have been focusing on the case when \( M = L^* + S^* \) where \( L^* = F_1^T W^* F_2 \). This model posits that \( W^* \) is exactly a rank-\( r \) matrix and \( S^* \) is exactly a sparse matrix which might not be the case in practice. Our approach, for solving Equation 4.5, in terms of both the algorithm and the analysis, also handles the noisy case \( M = F_1^T W^* F_2 + S^* + N^* \) wherein \( N^* \) is some generic bounded additive noise that renders \( L^* \) approximately low-rank or \( S^* \) approximately sparse.

### 4.2.2 Assumptions

We now state and explain the intuition behind the (by-now standard) identifiability assumptions on the quantities involved in our optimization problem so that it is well-posed. Also, we re-emphasize specifically that Assumptions 2 and 3 are much weaker and generic than previous works such as [11].

1. **Feasibility condition:** We assume that \( \text{row}(L^*) \subseteq \text{row}(F_2) \) and \( \text{col}(L^*) \subseteq \text{col}(F_1^\top) \).

2. **Weak incoherence of the feature matrices:** Let \( F_1 = U_{F_1} \Sigma_{F_1} V_{F_1}^\top \) be the SVD of the feature matrix \( F_1 \) such that \( U_{F_1} \in \mathbb{R}^{d_1 \times d_1}, V_{F_1} \in \mathbb{R}^{d_1 \times n_1} \) are the matrices of left and right singular vectors respectively, and \( \Sigma_{F_1} \in \mathbb{R}^{d_1 \times d_1} \) is the diagonal matrix of singular values. Then, we assume \( \max_i \| e_i^\top V_{F_1} \|_2 \leq \mu_{F_1} \sqrt{d_1/n_1} \) where \( \mu_{F_1} \) is called the incoherence constant of matrix \( F_1 \). Similarly, we assume incoherence of \( F_2 \) as well.

3. **Bounded deterministic sparsity:** Let the number of non-zeros per row of the sparse matrix \( S \) satisfy \( z_1 \leq n_1/20\mu^2 d_1 \kappa \); similarly, let the number of non-zeros per column of the sparse matrix \( S \) satisfy \( z_2 \leq n_2/20\mu^2 d_2 \kappa \). Here, \( \mu = \max(\mu_{F_1}, \mu_{F_2}) \) and \( \kappa = \max(\kappa(F_1), \kappa(F_2)) \).

Having side information always need not help; otherwise, we may always generate random features and obtain improvement over transductive learning. In disallowing this, Assump-
Algorithm 9 IRPCA-IHT: Inductive Robust PCA via Iterative Hard Thresholding

1: **Input:** Grossly corrupted data matrix $M \in \mathbb{R}^{n_1 \times n_2}$, feature matrices $F_1 \in \mathbb{R}^{d_1 \times n_1}, F_2 \in \mathbb{R}^{d_2 \times n_2}$, true rank $r$, noise parameter $\nu$.
2: **Output:** Estimated latent matrix $\hat{W} \in \mathbb{R}^{d_1 \times d_2}$ and sparse perturbation matrix $\hat{S} \in \mathbb{R}^{n_1 \times n_2}$.
3: Initialize $L_0 \leftarrow 0$ and $\zeta_0 \leftarrow 5 \|M\|_\infty + \nu$.
4: for $t = 1, \ldots, T$ do
5: Set threshold $\zeta_t \leftarrow \|M\|_\infty + \nu$.
6: $S_t \leftarrow \mathcal{P}_{\zeta_t}(M - L_{t-1})$.
7: $W_t \leftarrow \mathcal{P}_r\left((F_1^\top)^\dagger(M - S_t)(F_2)^\dagger\right)$.
8: $L_t \leftarrow F_1^\top W_t F_2$.
9: end for
10: Set $\hat{W} \leftarrow W_T$ and $\hat{S} \leftarrow S_T$.
11: return $\hat{W}, \hat{S}$.

Section 1 is a necessary condition that we have informative features $F_1$ and $F_2$ in the sense that they are correlated meaningfully in the latent space given by $W^*$. In order to make the low-rank component not too sparse and distinguishable from the sparse perturbation, we make the weak incoherence assumption on the feature matrices which says that the energy of the right singular vectors of the matrices is well-spread with respect to all the co-ordinate axes. This is precisely quantified by Assumption 2.

In our problem setup we assume that a generic (possibly adversarial) deterministic sparse perturbation is added to the low-rank matrix. This is quantified by Assumption 3. In particular, we do not have any specific distributional assumptions on the support of the sparse matrix, and the magnitudes and signs of its non-zero entries.

Remark 4.2 (Noisy case: assumptions). To obtain recovery guarantees for the noisy case described in Remark 4.1, the only assumption on $N^*$ we have is that it is suitably well-behaved – this is quantified by assuming $\|N^*\|_\infty \leq 1/40 \mu^2 d \kappa^2$. 

75
4.2.3 Sample Complexity

In this work, as given in Table 1.1, we refer to the rank-sparsity trade-off in Assumption 3 as ‘sample complexity’ – this is the allowable extent to which the model is robust to gross outliers while retaining identifiability, i.e., the number of non-zeros in the sparse corruption matrix. Note that, by using features, we are always able to tolerate $\Omega(\frac{n_1}{d_1})$ (resp. $\Omega(\frac{n_2}{d_2})$) gross corruptions per row (resp. column). This is a gain over the transductive setting as in [44] where the permissible number of outliers is $O(\frac{n_1}{r})$ (resp. $O(\frac{n_2}{r})$) per row (resp. column) and $r$ could be potentially $O(n)$.

4.2.4 Algorithm

Our method, presented in Algorithm 9, uses two non-convex projection operations as building blocks. Our algorithm essentially applies these projections to the low-rank and sparse residuals in an alternating manner until convergence, i.e., at the $t^{th}$ iteration, the residuals $M - L_{t-1}$ and $M - S_t$ are projected onto the set of sparse and low-rank matrices respectively via the following hard-thresholding operations:

1. **Spectral hard thresholding:** This is used for projecting a matrix onto the set of low-rank matrices. It is achieved via the truncated-SVD operation and is denoted by $B = P_r(A)$. Here, we are finding a matrix rank-$r$ matrix $B$ which best-approximates $A$.

2. **Entry-wise hard thresholding:** This is used for projecting a matrix onto the set of sparse matrices. We compute a matrix $B = P_a(A)$ where $B_{ij} = A_{ij}$ if $|A_{ij}| > a$ and $B_{ij} = 0$ if $|A_{ij}| \leq a$.

Note that the above hard thresholding operations result in in rank-restricted and sparsity-restricted matrices for appropriate choices of $r$ and $a$. It is noteworthy that our algorithm, unlike many non-convex optimization procedures, employs the very simple initialization scheme...
of setting the initial iterates to the all-zeros matrix while achieving global convergence.

The algorithm needs (a) the true rank \( r \) of \( W^* \), and (b) the noise parameter \( \nu \) (for which it suffices to have the knowledge of a reasonable bound on \( \| N^* \|_\infty (1+3\mu^2dn^2) \)). In practice, the knowledge of \( r \) and \( \|N^*\|_\infty \) can be obtained using cross-validation, grid search or leveraging domain knowledge of the specific application; for instance, in the noiseless setting, \( N^* = 0 \) and hence, \( \nu \) is set to zero.

A key difference from related approaches in the transductive setting [44] is the more efficient spectral hard thresholding that is possible due to the available feature information, i.e., our approach involves a truncated SVD operation in the feature space rather than the ambient space which is computationally inexpensive. Specifically, since \( L^* = F_1^T W^* F_2 \), in Step 7 of Algorithm 9, we find the best matrix \( W_t \) such that \( M - S_t \approx F_1^T W_t F_2 \) for every \( t \). This is achieved via a simple bilinear transformation of the residual \( M - S_t \) given by \((F_1^T)^\dagger(M - S_t)(F_2)^\dagger\) followed by a truncated \( r \)-SVD of the resulting \( d_1 \times d_2 \) matrix \( W_t \) to obtain \( W_t \). Note that the low-rank iterates may then be computed as \( L_t = F_1^T W_t F_2 \); specifically, \( \hat{L} = F_1^T \hat{W} F_2 \).

### 4.2.5 Computational Complexity

We now infer the per-iteration computational complexity from Algorithm 9, specifically Steps 6-8. The entry-wise hard-thresholding in Step 6 has a time complexity of \( O(n_1n_2) \). The spectral hard-thresholding in Step 7 has a time complexity of \( O(\max(n_1^2d_1, n_2^2d_2) + d_1d_2r) \) due to the involved matrix multiplication followed by the truncated SVD operation. Step 8 has a complexity of \( O(n_1n_2 \max(d_1, d_2)) \). Unlike previous [11] trace norm based approaches in the inductive setting, we directly perform rank-\( r \) SVD in Step 7 leading to a complexity of just \( O(d_1d_2r) \) as opposed to \( O(d_1d_2 \min(d_1, d_2)) \); this is a significant gain when \( r \ll \min(d_1, d_2) \).

In the transductive setting as well, our method has significant computational gains over the
state-of-the art AltProj algorithm of [44], especially in the regime $\max(d_1, d_2) < r^2$ while maintaining the sample complexity guarantees as in Section 4.2.3.
4.3 Analysis

4.3.1 Proof Outline

For simplicity, we first begin with the symmetric noiseless case (Section 4.3.2). Upon presenting the convergence result for this case, we show how to extend our analysis and result to general cases including the noisy case (Section 4.3.3) and the asymmetric matrix case (Section 4.3.4).

The key steps in the proof of convergence of Algorithm 9 involve analyzing the two main hard-thresholding operations and controlling the error decrease, in terms of a suitably chosen potential function, as a result of performing these operations. Since we care about recovering every entry of both the low-rank and the sparse matrix components, we choose the infinity norm of appropriate error matrices as our potential function to track the progress of our algorithm. Bounds in the infinity norm are trickier to obtain than the more usual spectral norm. Consequently, our guarantees are stronger as opposed to showing faithful recovery in the spectral or Frobenius norms. Specifically, for a given $t$, we show that $\|L^* - L_t\|_\infty \leq 2\|S^* - S_t\|_\infty \leq \frac{1}{5}\|L^* - L_{t-1}\|_\infty$. Upon showing this geometric reduction in error, we use induction to stitch up argument across iterations.

At a high level, the proof techniques involved for a fixed $t$ are as follows:

1. **Entry-wise hard thresholding**: The are two aspects here. First, given that $L_{t-1}$ is close to $L^*$, we show, by using a case-by-case argument, that $S_t$ is also close to $S^*$. Second, we show, by contradiction, that the $S_t$ does not have any spurious entries that are not present in $S^*$ originally.

2. **Spectral hard thresholding**: Given that $S_t$ is close to $S^*$, we show that $L_t$ gets closer to $L^*$ than $L_{t-1}$. There are three aspects here. First, we use the weak incoherence property of features to obtain infinity norm bounds. Second, we use Weyl’s eigenvalue
perturbation lemma to quantify how close the estimate $W_t$ is to the true latent matrix $W^*$. Third, we bound the spectral norm of a sparse matrix tightly in terms of its infinity norm.

For the noisy case, using Remark 4.1, we simply account for the noise terms as well in the error reduction argument. Extension to the asymmetric case proceeds via the standard symmetric embedding technique, both for the noiseless and the noisy setting, as detailed in Section 4.3.4; a key point to be noted here is that we maintain the rank-sparsity conditions in the symmetrized matrix.

4.3.2 Symmetric Noiseless Case

Let $N^* = 0$, $W^* = (W^*)^\top$ and $S^* = (S^*)^\top$. For simplicity, let the features be equal i.e., $F_1 = F_2 = F$. Further, let $d_1 = d_2$, $z_1 = z_2 = z$ and $n_1 = n_2 = n$. Also, recall that $\nu = 0$ in the noiseless case. We now state our main result.

**Theorem 4.1 (Noiseless case: fast and correct convergence).** Under the assumptions of Section 4.2.2, after $T > \lceil \log_5 (2 \|M\|_\infty / \epsilon) \rceil + 1$ iterations of Algorithm 9, we have $\|L^* - \hat{L}\|_\infty \leq \epsilon$, $\text{rank}(\hat{L}) \leq r$, $\|S^* - \hat{S}\|_\infty \leq \epsilon$ and $\text{Supp}(\hat{S}) \subseteq \text{Supp}(S^*)$.

**Remark 4.3.** Several implications are immediate from Theorem 4.1: (1) our algorithm converges to the true parameters at a linear rate; (2) we have faithful latent space recovery as well as outlier detection; (3) assumptions used for deriving the recovery guarantee are weaker than previous works in the inductive setting; (4) we achieve improved sample complexity; (5) guarantees for the transductive robust PCA problem are recovered if the features are identity matrices and $W^* = L^*$; in particular, our sample complexity bounds match up to a factor of $d/r$.

We now prove Theorem 4.1.
Proof. We prove this by induction over \( t \). Note that Step 3 of Algorithm 9 initializes \( \zeta_0 = 5\|M\|_\infty \) (as \( N^* = 0 \)) and sets \( \zeta_t = \zeta_{t-1}/5 \) for all \( t \geq 1 \). For \( t = 1 \), since \( L_0 = 0 \) by our initialization, it is clear that \( \|L^* - L_0\|_\infty \leq \|M\|_\infty \) and hence the base case holds. Next, for \( t \geq 1 \), by using Lemma 4.1, we have \( \|S^* - S_t\|_\infty \leq 2\|M\|_\infty /5^{t-1} \) and \( \text{Supp}(S_t) \subseteq \text{Supp}(S^*) \) and further, by Lemma 4.2, we have \( \|L^* - L_t\|_\infty \leq \|M\|_\infty /5^t \). Moreover, setting \( T > \lceil \log_5(2\|M\|_\infty /\epsilon) \rceil + 1 \), we have \( \|L^* - L_T\|_\infty \leq \epsilon \) and \( \|S^* - S_T\|_\infty \leq \epsilon \). \( \square \)

Lemma 4.1 (Noiseless case: faithful support recovery due to entry-wise hard thresholding). Let \( L_{t-1} \) satisfy the error condition that \( \|L^* - L_{t-1}\|_\infty \leq \|M\|_\infty /5^{t-1} \). Then, we have \( \|S^* - S_t\|_\infty \leq 2\|M\|_\infty /5^{t-1} \) and \( \text{Supp}(S_t) \subseteq \text{Supp}(S^*) \).

Proof. Note that \( S_t = \mathcal{P}_{\zeta_t}(M - L_{t-1}) = \mathcal{P}_{\zeta_t}(L^* - L_{t-1} + S^*) \). By the definition of our entry-wise hard thresholding operation, we have the following:

1. Term \( e_t^\top S_te_j = e_t^\top (M - L_{t-1})e_j = e_t^\top (L^* + S^* - L_{t-1})e_j \) when \( |e_t^\top (M - L_{t-1})e_j| > \zeta_t \). Thus \( |e_t^\top (S^* - S_t)e_j| = |e_t^\top (L^* - L_{t-1})e_j| \leq \|M\|_\infty /5^{t-1} \).

2. Term \( e_t^\top S_te_j = 0 \) when \( |e_t^\top (M - L_{t-1})e_j| = |e_t^\top (L^* + S^* - L_{t-1})e_j| \leq \zeta_t \). Using the triangle inequality, we have \( |e_t^\top (S^* - S_t)e_j| \leq |e_t^\top S^*e_j| \leq \zeta_t + |e_t^\top (L^* - L_{t-1})e_j| \leq 2\|M\|_\infty /5^{t-1} \).

Thus, the above two cases show the validity of the entry-wise hard thresholding operation. To show correct support recovery, we show that for any given \( (i, j) \), if \( e_t^\top S^*e_j = 0 \) then \( e_t^\top S_te_j \) is also zero for all \( t \). Noting that \( M = L^* + S^* \) and \( e_t^\top S^*e_j = 0 \), \( e_t^\top S_te_j = e_t^\top (M - L_{t-1})e_j = e_t^\top (L^* - L_{t-1})e_j \neq 0 \) iff \( |e_t^\top (L^* - L_{t-1})e_j| > \zeta_t \). But this is a contradiction since \( |e_t^\top (L^* - L_{t-1})e_j| \leq \|M\|_\infty /5^{t-1} = \zeta_t \) by the inductive assumption. \( \square \)

Lemma 4.2 (Noiseless case: error decay due to spectral hard thresholding). Let \( S_t \) satisfy the error condition that \( \|S^* - S_t\|_\infty \leq 2\|M\|_\infty /5^{t-1} \). Then, we have \( \|L^* - L_t\|_\infty \leq \|M\|_\infty /5^t \) and \( \text{rank}(L_t) \leq r \).
Proof. Using the fact that $L^* = F^T W^* F$ and $L_t = F^T W_t F$, we have

$$\|L^* - L_t\|_\infty = \|F^T (W^* - W_t) F\|_\infty$$

$$= \max_{i,j} |e_i^T F^T (W^* - W_t) F e_j|$$

$$\leq \max_{i,j} |e_i^T V_F \Sigma_F^T U_F^T (W^* - W_t) U_F \Sigma_F V_F^T e_j|$$

$$\leq \left( \max_i \left\| e_i^T V_F \Sigma_F^T \right\|_2 \right)^2 \|U_F^T (W^* - W_t) U_F\|_2,$$

(4.6)

where $\xi_1$ follows by substituting the SVD of $F$, i.e., $F = U_F \Sigma_F V_F^T$ and $\xi_2$ follows from the sub-multiplicative property of the spectral norm. Now, from Assumption 2, we have

$$\max_i \left\| e_i^T V_F \Sigma_F^T \right\|_2 \leq \mu \sqrt{\frac{d}{n} \sigma_{\max}(F)}.$$ 

(4.7)

Recall from Step 7 of Algorithm 9 that $W_t$ is computed as $\mathcal{P}_r \left( (F_1^T)^\dagger (M - S_t) (F_2)^\dagger \right)$ where $M = F_1^T W^* F_2 + S^*$. Let $E_t := S^* - S_t$. Further, let $Q\Lambda Q^T + Q_\perp A_\perp Q_\perp^T$ be the full SVD of $W^* + G^T E_t G$, where $Q$ and $Q_\perp$ span orthogonal sub-spaces of dimensions $r$ and $d - r$ respectively, and $G := F^\dagger$ is the pseudoinverse of $F$. Next, using these and the unitary invariance property of the spectral norm, we have

$$\|U_F^T (W^* - W_t) U_F\|_2 \leq \|W^* - W_t\|_2$$

$$\leq \|W^* - \mathcal{P}_r (G^T (F^T W^* F + E_t) G)\|_2$$

$$\leq \|Q \Lambda Q^T + Q_\perp A_\perp Q_\perp^T - G^T E_t G - Q \Lambda Q^T\|_2$$

$$\leq \|G^T E_t G\|_2 + \|Q_\perp A_\perp Q_\perp^T\|_2$$

$$\leq 2 \|G^T E_t G\|_2 \leq 2 \|G\|_2^2 \|E_t\|_2$$

$$\leq 2 \|E_t\|_2 \leq \frac{\xi_6}{\sigma_{\min}(F)^2} \leq \frac{2 \varepsilon \|E_t\|_\infty}{\sigma_{\min}(F)^2},$$

(4.8)
Figure 4.1: Comparing of RPCA algorithms in terms of running time to reach a solution of a given accuracy. For \( n = 1000 \), we vary each problem parameter while fixing the others. Specifically, we vary:

(a) sparsity  
(b) rank  
(c) feature dimension  
(d) condition number

where \( \xi_3 \) is obtained by substituting \( W^* = QAQ^T + Q_\perp \Lambda_\perp Q_\perp^T - G^T E_t G \), \( \xi_4 \) by triangle inequality. Inequality \( \xi_5 \) is obtained by using Weyl’s eigenvalue perturbation lemma [3], which is:

\[
\|Q_\perp \Lambda_\perp Q_\perp^T\|_2 = \|\Lambda_\perp\|_\infty \leq \|G^T E_t G\|_2.
\]

Finally, inequality \( \xi_6 \) is obtained by using Lemma 4 of [44]. Combining Equations (4.6), (4.7) and (4.8), we have

\[
\|L^* - L_t\|_\infty \leq 2\mu^2 d\kappa^2 \|E_t\|_\infty / n \leq \|E_t\|_\infty / 10, \tag{4.9}
\]

where \( \kappa = \sigma_{\text{max}}(F) / \sigma_{\text{min}}(F) \) and \( \xi_7 \) is due to Assumption 3. Substituting the result \( \|E_t\|_\infty = \|S^* - S_t\|_\infty \leq 2 \|M\|_\infty / 5^{l-1} \) from Lemma 4.1 in Equation (4.9) completes the proof.

### 4.3.3 Symmetric Noisy Case

Next we consider the general noisy case of \( M = L^* + S^* + N^* \), where \( L^* = F^T W^* F \), \( S^* \) and \( N^* \) are symmetric and \( N^* \) is a bounded additive noise matrix satisfying properties as given in Remark 4.2. Note that, in practice, by setting \( \nu = c.d \) for a suitably chosen constant \( c \), Algorithm 9 works unchanged. However, in order to establish convergence in theory, the key
challenge is to be able to control the perturbation effects of $N^*$ in each iteration. In making this precise, we now state our main result for this section whose proof is given in Section 4.6 due to space limitations.

**Theorem 4.2 (Noisy case: fast and correct convergence).** Under the assumptions of Section 4.2.2, setting $T > \lceil \log_5(2\|M\|_{\infty}/\epsilon) \rceil + 1$ in Algorithm 9, we have $\|L^* - \hat{L}\|_{\infty} \leq \epsilon + 3\mu^2d\kappa^2\|N^*\|_{\infty}$, $\text{rank}(\hat{L}) \leq r$, $\|S^* - \hat{S}\|_{\infty} \leq \epsilon + 8\mu^2d\kappa^2\|N^*\|_{\infty}$ and $\text{Supp}(\hat{S}) \subseteq \text{Supp}(S^*)$.

To prove the above theorem, we need the following key lemmas whose proofs are given in Section 4.6 as well.

**Lemma 4.3 (Noisy case: faithful support recovery due to entry-wise hard thresholding).** Let $L_{t-1}$ satisfy the error condition that $\|L^* - L_{t-1}\|_{\infty} \leq \frac{\|M\|_{\infty}}{5^{t-1}} + 3\mu^2d\kappa^2\|N^*\|_{\infty}$. Then, we have $\|S^* - S_t\|_{\infty} \leq \frac{2\|M\|_{\infty}}{5^{t-1}} + 2(3\mu^2d\kappa^2 + 1)\|N^*\|_{\infty}$ and $\text{Supp}(S_t) \subseteq \text{Supp}(S^*)$.

**Lemma 4.4 (Noisy case: error decay due to spectral hard thresholding).** Let $S_t$ satisfy the error condition that $\|S^* - S_t\|_{\infty} \leq \frac{2\|M\|_{\infty}}{5^{t-1}} + 2(3\mu^2d\kappa^2 + 1)\|N^*\|_{\infty}$. Then, we have $\|L^* - L_t\|_{\infty} \leq \frac{\|M\|_{\infty}}{5^{t-1}} + 3\mu^2d\kappa^2\|N^*\|_{\infty}$ and $\text{rank}(L_t) \leq r$.

### 4.3.4 Asymmetric Case

We now show how to extend our analysis for any general asymmetric matrix, both in the noiseless and the noisy inductive settings. Let $M \in \mathbb{R}^{n_1 \times n_2}$ be the input data matrix. The main result can be stated as:

**Claim 4.1.** Let $M = L^* + S^* + N^*$ where $L^* = F_1^TW^*F_2$ such that $n_1 \neq n_2$ and $d_1 \neq d_2$. Algorithm 9 executed on this $M$ satisfies the guarantees in Theorem 4.1 (resp. Theorem 4.2) for the noiseless case where $N^* = 0$ (resp. noisy case where $N^*$ satisfies the properties in Remark 4.2).
Consider the standard symmetric embedding of a matrix given by:

$$\text{Sym}(M) := \begin{pmatrix} 0 & M \\ M^T & 0 \end{pmatrix}. $$

With $\text{Sym}(M)$ as input, the intermediate iterates of our algorithm also have a similar form. Moreover, note that this embedding preserves the rank, incoherence and sparsity properties – due to space constraints, these details which are needed as the key components of the proof of Claim 4.1 are deferred to Section 4.7.
4.4 Experiments

In this section, we conduct a systematic empirical investigation of the performance of our robust subspace recovery method (IRPCA-IHT) and justify our theoretical claims in the previous sections. Specifically, the goal of this study is to show: (1) the correctness of our algorithm, (2) that informative features and feature correlations are indeed useful, and (3) that our algorithm is computationally efficient.

4.4.1 Synthetic Simulations

We set the problem size as $n_1 = n_2 = n = 1000$; for simplicity, we take $d_1 = d_2 = d$, $z_1 = z_2 = z$ and $F_1 = F_2 = F$; let $\kappa$ be the condition number of the feature matrix $F$.

First, we generate approximately well-conditioned weakly incoherent feature matrices by computing $F = U_F \Sigma_F V_F^\top$ where the entries of $U_F \in \mathbb{R}^{d \times d}$ and $V_F \in \mathbb{R}^{d \times n}$ are drawn iid from the standard normal distribution followed by row normalization, and the diagonal entries of $\Sigma_F$ are set to one. Next, the latent matrix $W^*$ is generated by sampling each entry independently and uniformly at random from the interval $(0, 1)$, performing SVD of this sampled matrix and retaining its top $r$ singular values. The low-rank component $L^*$ is then computed as $F^\top W^* F$. Note that this also ensure the feasibility condition in Assumption 1.

Next, we generate the sparse matrix as follows. We first choose the support according to the Bernoulli sampling model, i.e., each entry is chosen to be included in the support with probability $z/n$ and then its value is chosen independently and uniformly at random from $(-10r/n, -5r/n) \cup (5r/n, 10r/n)$.

There are four main parameters in the problem namely, (a) the sparsity level $z$ of $S^*$, (b) the rank $r$ of $W^*$, (c) the feature dimension $d$, and (d) condition number $\kappa$ of the feature matrix $F$; we vary each of these while fixing the others. We compare the performance of our algo-
algorithm to that of two existing algorithms namely, (i) the convex relaxation approach ‘PCPF’ due to [11] which is a state-of-the-art robust PCA method in the inductive setting, and (ii) ‘AltProj’ due to [44] which is a state-of-the-art robust PCA method in the transductive setting. We execute these algorithms until an accuracy of \( \| M - \hat{L} - \hat{S} \|_F / \| M \|_F \leq 10^{-3} \) is achieved and time them individually. All the results presented in the running time plots in Figure 4.1 are obtained by averaging over five runs.

We note that our algorithm outperforms PCPF and AltProj consistently while increasing the problem hardness in three situations (Figures 1-(a), 1-(b) and 1-(c)) in terms of running time. The gain in terms of scalability of our method over the convex PCPF method is attributed to the fact that the soft thresholding operation for solving the nuclear-norm objective involves computing the partial-SVD of the intermediate iterates which could of potentially much higher rank than \( r \) – this leads to \( O(d^3) \) worst-case time complexity for the SVD step in PCPF as opposed to our algorithm which has \( O(d^2r) \) worst-case complexity for spectral hard thresholding. The time gain over the transductive AltProj method is attributed to the fact that our spectral hard-thresholding is performed in the \( d \)-dimensional (feature) space rather than the \( n \)-dimensional (ambient) space; moreover, another factor that adds to the running time of AltProj is that it proceeds in stages unlike Algorithm 9. An interesting point to be noted from the relatively flat plot in Figure 4.1-(d) is that the condition number dependence in Assumption 3 is merely an artifact of our analysis and is not inherent to the problem; we leave tightening this bound in theory to future work.

### 4.4.2 Real-data Simulations

As described in Example 4.1, we consider an important application of our method – to robustify estimation in recommendation systems while leveraging feature information; specifically, the task is to predict user-movie ratings accurately despite the presence of gross sparse cor-
Figure 4.2: Comparison of robust PCA algorithms on the MovieLens data: running time and recovery error.

ruptions. We take the MovieLens \(^1\) dataset which consists of 100,000 ratings from \(n_1 = 943\) users on \(n_2 = 1682\) movies. The ground-truth in this dataset is, per se, unavailable. Hence, as the first step, we apply matrix completion techniques (specifically, using the OptSpace algorithm of \([32]\)) to obtain a baseline complete user-movie ratings matrix, \(L^*\); we take \(r = 3\). Next, we form features while ensuring the feasibility condition. For this, we compute the SVD of the baseline matrix, 

\[ L^* = U_L \Sigma_L V_L^T \]

followed by setting 

\[ F_1 = U_L Q_U \] (resp. 

\[ F_2 = V_L Q_V \]) where 

\[ Q_U \in SO(d_1) \] (resp. 

\[ Q_V \in SO(d_2) \]) are random rotation matrices; we take \(d_1 = 20\) and \(d_2 = 25\). Note that forming features using the SVD result, as we have done here, is a common technique in inductive matrix estimation problems (see, for instance, \([41]\)). We then add a sparse perturbation matrix whose each entry is chosen to be included in the support with probability \(z/n\) and the entries are chosen independently and uniformly at random from 

\[ (-10r/\sqrt{n_1n_2}, -5r/\sqrt{n_1n_2}) \cup (5r/\sqrt{n_1n_2}, 10r/\sqrt{n_1n_2}) \]  

We compare the performance of PCPF, AltProj and our IRPCA-IHT algorithms; we consider two evaluation metrics, running time and relative recovery error (the latter is measured by 

\[ \|\hat{S} - S^*\|_F / \|S^*\|_F \]). Varying \(z\) and averaging over five runs, we note that our algorithm outperforms (Figure 4.2) both PCPF and AltProj by achieving about an order of magnitude of gain in terms of both the running time as well as the recovery error.

---

\(^1\)http://grouplens.org/datasets/movielens/
4.5 Conclusion

In this chapter, we have presented a novel approach for inductive robust subspace identification by leveraging available informative feature information. We hope our results motivate similar studies of other learning problems in the inductive setting leading to improved statistical and computational performance. Keeping this in mind, some future directions with respect to this work include understanding the following:

1. Minimax rates, both tight lower and upper bounds for learning problems in the inductive setting, are of interest. Relevant techniques include the works by [43] and [33] in the transductive setting.

2. We note that the sample complexity in Assumption 3 is still sub-optimal by a factor of $d/r$ which is significant when $r \ll d$. In addition to this, removing the condition number dependence in Assumption 3 and also obtaining $\epsilon$-independent results as in matrix completion (see for instance, [26]) are of interest.
4.6 Proofs: Noisy Case

4.6.1 Proof of Theorem 4.2

Proof. We prove this by induction over \( t \). Note that Step 3 of Algorithm 9 initializes \( \zeta_0 = 5 \|M\|_\infty + \nu \) and sets \( \zeta_t = \|M\|_\infty / 5^{t-1} + \nu \) for all \( t \geq 1 \). Let \( \nu = (3\mu^2d\kappa^2 + 1) \|N^*\|_\infty \).

For \( t = 1 \), since \( L_0 = 0 \) by our initialization, it is clear that \( \|L^* - L_0\|_\infty \leq \|M\|_\infty \) and hence the base case holds. Next, for \( t \geq 1 \), by using Lemma 4.3, we have \( \|S^* - S_t\|_\infty \leq \frac{2\|M\|_\infty}{\sqrt{5}} + 2(3\mu^2d\kappa^2 + 1) \|N^*\|_\infty \) and further, by Lemma 4.4, we have \( \|L^* - L_t\|_\infty \leq \frac{\|M\|_\infty}{\sqrt{5}} + 3\mu^2d\kappa^2 \|N^*\|_\infty \). Moreover, setting \( T > \lceil \log_5(2 \|M\|_\infty / \epsilon) \rceil + 1 \), we obtain the result. \( \square \)

4.6.2 Proof of Lemma 4.3

Proof. Recall that \( S_t = \mathcal{P}_{\zeta_t}(M - L_{t-1}) = \mathcal{P}_{\zeta_t}(L^* - L_{t-1} + S^* + N^*) \). By the definition of our entry-wise hard thresholding operation, we have the following:

1. Term \( e_i^T S_t e_j = e_i^T (M - L_{t-1})e_j = e_i^T (L^* + S^* + N^* - L_{t-1})e_j \) when \( |e_i^T (M - L_{t-1})e_j| > \zeta_t \). Thus, \( |e_i^T (S^* - S_t)e_j| = |e_i^T (L^* - L_{t-1})e_j| + |e_i^T N^*e_j| \leq \frac{\|M\|_\infty}{\sqrt{5}} + 3\mu^2d\kappa^2 \|N^*\|_\infty + \|N^*\|_\infty \).

2. Term \( e_i^T S_t e_j = 0 \) when \( |e_i^T (M - L_{t-1})e_j| = |e_i^T (L^* + S^* + N^* - L_{t-1})e_j| \leq \zeta_t \). Now, using the triangle inequality, we have \( |e_i^T (S^* - S_t)e_j| = |e_i^T S^*e_j| \leq \zeta_t + |e_i^T (L^* - L_{t-1})e_j| + |e_i^T N^*e_j| \leq 2 \left( \frac{\|M\|_\infty}{\sqrt{5}} + 3\mu^2d\kappa^2 \|N^*\|_\infty + \|N^*\|_\infty \right) \).

Thus, the above two cases show the validity of the entry-wise hard thresholding operation.

Next, we show that for any given \((i, j)\), if \( e_i^T S^*e_j = 0 \) then \( e_i^T S_t e_j \) is also zero for all \( t \). Noting that \( M = L^* + S^* + N^* \) and \( e_i^T S^*e_j = 0 \), \( e_i^T S_t e_j = e_i^T (M - L_{t-1})e_j = e_i^T (L^* + N^* - L_{t-1})e_j \neq 0 \) iff \( |e_i^T (L^* + N^* - L_{t-1})e_j| > \zeta_t \). But this is a contradiction since \( |e_i^T (L^* + N^* - L_{t-1})e_j| \leq |e_i^T (L^* - L_{t-1})e_j| + |e_i^T N^*e_j| \leq \frac{\|M\|_\infty}{\sqrt{5}} + 3\mu^2d\kappa^2 \|N^*\|_\infty + \|N^*\|_\infty = \zeta_t \). \( \square \)
4.6.3 Proof of Lemma 4.4

Proof. Using the fact that \( F_1 = F_2 \), \( L^* = F^\top W^* F \) and \( L_t = F^\top W_t F \), we have

\[
\|L^* - L_t\|_\infty = \|F^\top (W^* - W_t) F\|_\infty \\
= \max_{i,j} |e_i^\top F^\top (W^* - W_t) F e_j| \\
\xi_{11} \leq \max_{i,j} |e_i^\top V F \Sigma F^\top U F (W^* - W_t) U F \Sigma F^\top V e_j| \\
\xi_{12} \leq \left( \max_i \|e_i^\top V F \Sigma F^\top \|_2 \right)^2 \|U F (W^* - W_t) U F\|_2
\]

(4.10)

where \( \xi_{11} \) follows by substituting the SVD of \( F = U F \Sigma_F V^\top_F \) and \( \xi_{12} \) follows from the sub-multiplicative property of the spectral norm. Similar to the proof of Lemma 4.2, using Assumption 2 we have:

\[
\max_i \|e_i^\top V F \Sigma F^\top \|_2 \leq \mu \sqrt{\frac{d}{n}} \sigma_{\max}(F).
\]

(4.11)

Let the residual sparse perturbation be defined as \( E_t := S - S_t \). Let \( Q \Lambda Q^\top + Q_\perp \Lambda_\perp Q_\perp^\top \) be the full SVD of \( W^* + G^\top (E_t + N^*) G \) where \( Q \) and \( Q_\perp \) span orthogonal sub-spaces of dimensions \( r \) and \( d - r \) respectively, and \( G = F^\dagger \) is the pseudoinverse. Also, recall that from Step 7 of Algorithm 9 that \( W_t \) is computed as \( \mathcal{P}_r \left( (F_1^\top)^\dagger (M - S_t)(F_2)^\dagger \right) \) where \( M = F_1^\top W^* F_2 + S^* + N^* \). Using these and the unitary invariance property of the spectral norm.
norm, we have

\[ \|U_F^T(W^* - W_t)U_F\|_2 \leq \|W^* - W_t\|_2 \]
\[ \leq \|W^* - P_r(G^T(F^T W^* F + E_t + N^*) G)\|_2 \]
\[ \leq \|Q\Lambda Q^T + Q\perp\Lambda\perp Q^T\perp - G^T(E_t + N^*) G - Q\Lambda Q^T\|_2 \]
\[ \leq \|G^T(E_t + N^*)\|_2 + \|Q\perp\Lambda\perp Q^T\perp\|_2 \]
\[ \leq 2\|G^T(E_t + N^*)\|_2 \leq 2\|G\|_2^2 \|E_t + N^*\|_2 \]
\[ \leq \frac{2\|E_t + N^*\|_2^2}{[\sigma_{\text{min}}(F)]^2} \leq \frac{2\|E_t\|_\infty^2}{[\sigma_{\text{min}}(F)]^2} + \frac{2\|N^*\|_2^2}{[\sigma_{\text{min}}(F)]^2} \]  
(4.12)

where \( \xi_{13} \) is obtained by substituting \( W^* = Q\Lambda Q^T + Q\perp\Lambda\perp Q^T\perp - G^T(E_t + N^*) G \), \( \xi_{14} \) by triangle inequality, \( \xi_{15} \) by using Weyl’s eigenvalue perturbation lemma, ie,

\[ \|Q\perp\Lambda\perp Q^T\perp\|_2 = \|\Lambda\perp\|_\infty \leq \|G^T(E_t + N^*)\|_2 \]

and \( \xi_{16} \) by using Lemma 4 of [44] along with triangle inequality. Now, combining Equations (4.10), (4.11) and (4.12), we have

\[ \|L^* - L_t\|_\infty \leq 2\mu^2 d\kappa^2 \left( z \|E_t\|_\infty + \|N^*\|_2 \right) \]
\[ \leq \frac{\|E_t\|_\infty}{10} + 2\mu^2 d\kappa^2 \|N^*\|_\infty \]  
(4.13)

where \( \xi_{17} \) follows by using Assumption 3 and the inequality that \( \|N^*\|_2 \leq n \|N^*\|_\infty \). Using the inequality \( \|S^* - S_t\|_\infty \leq 2 \left( \frac{\|M\|_\infty}{d\kappa^2} + (3\mu^2 d\kappa^2 + 1) \|N^*\|_\infty \right) \) from Lemma 4.3 in Equation (4.13) completes the proof.
4.7 Proofs: Asymmetric Case

4.7.1 Proof of Claim 4.1

Proof. Applying the symmetric embedding transformation to our data matrix, we get $\text{Sym}(M) = \text{Sym}(L^*) + \text{Sym}(S^*)$. Now we characterize the properties of this symmetric embedding and show that it satisfies Assumptions 1, 2 and 3. First, we have

$$\text{Sym}(L^*) = \begin{pmatrix} 0 & L^* \\ L^{\top} & 0 \end{pmatrix} = \begin{pmatrix} 0 & F_1^\top W^* F_2 \\ F_2^\top W^* F_1 & 0 \end{pmatrix} = \begin{pmatrix} F_1^\top & 0 \\ 0 & F_2^\top \end{pmatrix} \begin{pmatrix} 0 & W^* \\ W^{\top} & 0 \end{pmatrix} \begin{pmatrix} F_1 & 0 \\ 0 & F_2 \end{pmatrix}.$$ 

Thus, $\text{Sym}(L^*)$ is of the form $\tilde{F}^\top \tilde{W}^* \tilde{F}$. If the SVD of $W^*$ is $U_{W^*} \Sigma_{W^*} V_{W^*\top}$, then the eigenvalue decomposition of $\tilde{W}^*$ is given by

$$\tilde{W}^* = \begin{pmatrix} 0 & W^* \\ W^{\top} & 0 \end{pmatrix} = \begin{pmatrix} 0 & U_{W^*} \Sigma_{W^*} V_{W^*\top} \\ V_{W^*} \Sigma_{W^*} U_{W^*\top} & 0 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} U_{W^*} & U_{W^*} \\ V_{W^*} & -V_{W^*} \end{pmatrix} \begin{pmatrix} \Sigma_{W^*} & 0 \\ 0 & -\Sigma_{W^*} \end{pmatrix} \begin{pmatrix} U_{W^*} & U_{W^*} \\ V_{W^*} & -V_{W^*} \end{pmatrix}^\top,$$

implying that $\text{rank}(\tilde{W}^*) = 2 \cdot \text{rank}(W^*)$. Next, let the SVDs of $F_1$ and $F_2$ be $U_{F_1} \Sigma_{F_1} V_{F_1\top}$ and $U_{F_2} \Sigma_{F_2} V_{F_2\top}$ respectively; also, without loss of generality, let $\sigma_{\text{min}}(F_1) > \sigma_{\text{min}}(F_2)$. Then, the
SVD of $\tilde{F} = U_{\tilde{F}} \Sigma_{\tilde{F}} V_{\tilde{F}}^T$ is given by

$$\tilde{F} = \begin{pmatrix} F_1 & 0 \\ 0 & F_2 \end{pmatrix} = \begin{pmatrix} U_{F_1} & 0 \\ 0 & U_{F_2} \end{pmatrix} \begin{pmatrix} \Sigma_{F_1} & 0 \\ 0 & \Sigma_{F_2} \end{pmatrix} \begin{pmatrix} V_{F_1}^T \\ 0 \end{pmatrix}$$

Now, we verify that the right singular vectors of this new feature matrix $\tilde{F}$ satisfies weak incoherence property. Specifically, we expect that the following holds:

$$\max_j \| V_e e_j \|_2 \leq \mu_{\tilde{F}} \sqrt{\frac{d_1 + d_2}{n_1 + n_2}} \quad (4.14)$$

On the other hand, we actually have

$$\max_j \| V e_j \|_2 \leq \max \left( \mu_{F_1} \sqrt{\frac{d_1}{n_1}}, \mu_{F_2} \sqrt{\frac{d_2}{n_2}} \right). \quad (4.15)$$

Wlog, let $\mu_{F_1} \sqrt{d_1/n_1} > \mu_{F_2} \sqrt{d_2/n_2}$. Then, combining Equations (4.14) and (4.15), we want

$$\frac{\mu_{\tilde{F}}}{\mu_{F_1}} \leq \sqrt{\frac{1 + n_2/n_1}{1 + n_2/d_1}}.$$  In particular, when $n_2/n_1 = d_2/d_1$, the incoherence constant for $\tilde{F}$ satisfies $\mu_{\tilde{F}} = \mu_{F_1}$.

Next, note that $\text{Sym}(S^*)$ is also sparse; specifically, $\| S^* \|_{0,\infty} \leq z$ and $\| S^* \|_{\infty,0} \leq z$ where $z = \max(z_1, z_2)$.

Finally, our algorithm and guarantees hold for general matrices with noise, similar to noiseless case, due to the following observation: $\| \text{Sym}(N^*) \|_\infty = \| N^* \|_\infty$.  \qed
Bibliography


