4-2022

Robust L1-norm Singular-Value Decomposition and Estimation

Duc H. Le
dhl3772@rit.edu

RIT

Robust L1-norm Singular-Value Decomposition and Estimation

by

Duc H. Le

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Electrical Engineering

Supervised by
Dr. Panos P. Markopoulos
Department of Electrical and Microelectronic Engineering
Kate Gleason College of Engineering
Rochester Institute of Technology, Rochester, NY
April, 2022

Approved by:

Dr. Panos P. Markopoulos, Associate Professor
Thesis Advisor, Department of Electrical and Microelectronic Engineering

---

Dr. Sohail A. Dianat, Professor
Committee Member, Department of Electrical and Microelectronic Engineering

---

Dr. Majid Rabbani, Professor of Practice
Committee Member, Department of Electrical and Microelectronic Engineering

---

Dr. Ferat E. Sahin, Professor
Department Head, Department of Electrical and Microelectronic Engineering

Thesis Release Permission Form

Rochester Institute of Technology
Kate Gleason College of Engineering

Title:

Robust L1-norm Singular-Value
Decomposition and Estimation

I, Duc H. Le, hereby grant permission to the Wallace Memorial Library to reproduce my thesis in whole or part.

Duc H. Le

Date

Dedication

I dedicate this Thesis to my dear Mother (Mẹ), Phạm Đỗ Quyên, my Father (Ba), Lê Hữu Danh, and my younger brother Lê Hữu Trung for their neverending love and support from halfway across the globe. I also dedicate this work to my partner, Lê Hà Lan Nhi, for having accompanied me through important milestones in my life.

Acknowledgements

There are a number of people without whose support this Thesis would never have been possible and I am forever grateful for them.

I thank my advisor, Professor Panos Markopoulos, for being a patient and listening guide, a passionate and extraordinary teacher, and a kind and caring person.

I thank all of my family members, for being supportive through every up and down.

I thank my late grandparents, whose blessings reassured me through hard times.

I thank all of my teachers and friends for having given me invaluable lessons and shaped me as a person.

I thank Dr. Mishkat Bhattacharya for having given me the unique opportunity to be exposed to the practice of research as early as Sophomore year, without which this work could not have been completed smoothly.

I thank my labmates at the MILOS Lab for their advice and directions on not only my research but also my career path.

I thank my Thesis Committee members, Dr. Dianat and Dr. Rabbani for taking their valuable time to review my Thesis.

Abstract

Robust L1-norm Singular-Value Decomposition and Estimation

Duc H. Le

Supervising Professor: Dr. Panos P. Markopoulos

Singular-Value Decomposition (SVD) is a ubiquitous data analysis method in engineering, science, and statistics. Singular-value estimation, in particular, is of critical importance in an array of engineering applications, such as channel estimation in communication systems, EMG signal analysis, and image compression, to name just a few. Conventional SVD of a data matrix coincides with standard Principal-Component Analysis (PCA). The L2-norm (sum of squared values) formulation of PCA promotes peripheral data points and, thus, makes PCA sensitive against outliers. Naturally, SVD inherits this outlier sensitivity. In this work, we present a novel robust method for SVD based on a L1-norm (sum of absolute values) formulation, namely L1-norm compact Singular-Value Decomposition (L1-cSVD). We then propose a closed-form algorithm to solve this problem and find the robust singular values with cost $\mathcal{O}(N^3K^2)$. Accordingly, the proposed method demonstrates sturdy resistance against outliers, especially for singular values estimation, and can facilitate more reliable data analysis and processing in a wide range of engineering applications.

List of contributions

- Novel preliminary algorithm for L1 and L2-norm Compact Singular-Value Decomposition (L1L2-cSVD).
- Novel proposed algorithm for L1-norm Compact Singular-Value Decomposition (L1-cSVD).
- Numerical studies with synthetic data on outlier-resistant singular values estimation and low-rank approximation.
- Experimental study on using L1-cSVD for a Bayesian Classifier on a dataset from PMLB.
- Experimental study on using L1-cSVD for preprocessing in the ℓ1-SVD algorithm for direction-of-arrival estimation.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Singular-Value Decomposition

The Singular-Value Decomposition (SVD) has established itself as a powerful tool ubiquitous in various engineering applications. One example is multiple-input multiple-output (MIMO) channel capacity estimation, where applying SVD onto the channel matrix decomposes the MIMO channel into multiple single-input single-output (SISO) channels with gains corresponding to singular values (SVs), enabling efficient power allocation and channel capacity estimation [2, 3]. Furthermore, SVD has been used for watermarking [4, 5], direction of arrival (DOA) estimation [1, 6, 7], restructuring of deep neural network acoustic models [8], electromyography (EMG) signal analysis [9], etc.

The SVD decomposes a $D \times N$ matrix into [10, 11]

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \tag{1.1}$$

where $\mathbf{U}\,(D \times d)$ and $\mathbf{V}\,(N \times d)$, defined as the left and right singular vectors respectively,

are orthonormal matrices while $\mathbf{\Sigma}\,(d \times d)$ is a diagonal matrix whose diagonal elements are the SVs, $(\cdot)^T$ denotes the transpose operation, and $d = \text{rank}(\mathbf{X})$. Throughout this thesis, we will work with "fat" matrices with $D < N$, where D can be understood as the number of dimensions and N can be understood as the number of data points. Eq. (1.1) refers to the "compact" SVD (cSVD) where the left or right singular vectors corresponding to zero SVs are disregarded [10, 11]. For simplicity reason, we will refer to the "compact" SVD as SVD from now on.

The SVD can be easily found by performing the Eigenvalue Decomposition on $\mathbf{X}\mathbf{X}^T = \mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^T$ and $\mathbf{X}^T\mathbf{X} = \mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^T$, whose eigenvectors are $\mathbf{U}$ and $\mathbf{V}$, respectively, and whose eigenvalues are the square of the SVs.

1.2 Principal-Component Analysis

In this section, we look at another mathematical tool closely related to SVD, which is the Principal-Component Analysis (PCA). PCA seeks to find K unit vectors such that the k^{th} vector, or principal component (PC), maximizes the variance of the data projected on it while being orthogonal to the first $(k-1)$ PCs. PCA is mainly used for dimensionality reduction by only keeping the first few PCs as the new data dimensions, thus preserving most of the data variance while having the benefit of working with fewer dimensions. Therefore, it has found use in a number of disciplines such as machine learning, signal processing, and pattern recognition [12, 13, 14].

The standard SVD is very closely related to PCA, since the first $K\,(K \leq d)$ left singular vectors of $\mathbf{U}$ are also the first K PCs of $\mathbf{X}$, which satisfy the following L2-norm (or Frobenius

norm) optimization problem

$$\mathbf{Q}_{L2} = \underset{\mathbf{Q} \in \mathbb{S}^{D \times K}}{\operatorname{argmax}} ||\mathbf{Q}^T \mathbf{X}||_{2,2}, \tag{1.2}$$

where $\mathbb{S}$ denotes the Stiefel manifold, i.e, $\mathbf{Q} \in \mathbb{R}^{D \times K}, \mathbf{Q}^T \mathbf{Q} = \mathbf{I}_K$ and the $L_{p,q}$ norm is defined for a matrix $\mathbf{A} \in \mathbb{R}^{D \times N}$ to be

$$||\mathbf{A}||_{p,q} = \left(\sum_{j=1}^{N} \left(\sum_{i=1}^{D} |a_{i,j}|^p \right)^{q/p} \right)^{1/q}. \tag{1.3}$$

This problem essentially maximizes the L2-norm of the dimensionality-reduced data $\mathbf{Q}^T \mathbf{X}$. Thus, the variance of the original data is actively preserved in the projected data. Alternatively, PCA can be formulated as

$$\mathbf{Q}_{L2} = \underset{\mathbf{Q} \in \mathbb{S}^{D \times K}}{\operatorname{argmin}} ||\mathbf{X} - \mathbf{Q}\mathbf{Q}^T \mathbf{X}||_{2,2}, \tag{1.4}$$

which minimizes the energy of the residue between the original data matrix $\mathbf{X}$ and its projection onto the low-rank subspace spanned by $\mathbf{Q}$. Finally, the PCA problem can also be written as an L2-norm closest rank-K matrix problem

$$\mathbf{R}_{L2}, \mathbf{Z}_{L2} = \underset{\mathbf{R} \in \mathbb{R}^{D \times K}, \mathbf{Z} \in \mathbb{R}^{K \times N},}{\operatorname{argmin}} ||\mathbf{X} - \mathbf{R}\mathbf{Z}||_{2,2}. \tag{1.5}$$

Under the L2-norm, the 3 mentioned optimization problems are equivalent, meaning that $\mathbf{R}_{L2}\mathbf{Z}_{L2}$ from Eq. (1.5) is equal to $\mathbf{Q}_{L2}\mathbf{Q}_{L2}^T\mathbf{X}$ from Eq. (1.4), where $\mathbf{Q}_{L2}$ is also the solution of Eq. (1.2). This is known as the Projection Theorem [10, 11].

1.3 Mathematical properties of SVD and PCA

1.3.1 Low-rank approximation

One important application of SVD is its ability for low-rank approximation. Specifically, the product of the matrices $\mathbf{U}_K, \mathbf{\Sigma}_K, \mathbf{V}_K$ containing the first K left singular vectors, singular values and right singular vectors, respectively, is the closest rank-K matrix to $\mathbf{X}$ in the L2 sense, i.e.

$$\mathbf{U}_K, \mathbf{\Sigma}_K, \mathbf{V}_K = \underset{\mathbf{U}\in\mathbb{S}^{D\times K}, \mathbf{\Sigma}\text{ diag}, \mathbf{V}\in\mathbb{S}^{N\times K},}{\operatorname{argmin}} ||\mathbf{X} - \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T||_{2,2}. \tag{1.6}$$

This is known as the Eckart-Young theorem [15].

1.3.2 Scalability

In addition, the PCs, or left singular vectors, found by solving the projection maximization problem Eq. (1.2) with $K = k_1$ are also the first k_1 PCs found by solving the same problem with $K = k_2$ where $k_1 < k_2$. In other words, the PCs are not affected by the number of PCs to be found, or equivalently the matrix rank to be approximated down to. This is known as the scalability property of PCA.

1.3.3 Diagonality

Furthermore, the PCs $\mathbf{Q}_{L2}$ found by solving Eq. (1.2) can also diagonalize $\mathbf{X}$, meaning $\mathbf{Q}_{L2}^T\mathbf{X}\mathbf{X}^T\mathbf{Q}_{L2}$ is a diagonal matrix, or equivalently the projected coordinates of $\mathbf{X}$ in the orthonormal basis defined by $\mathbf{Q}_{L2}$, i.e., $\mathbf{Q}_{L2}^T\mathbf{X}$, is readily orthogonal. In this work, an "orthogonal" matrix is one whose column vectors are orthogonal but not necessarily normalized. Interestingly, $\mathbf{Q}_{L2}$ is the only orthonormal matrix that can diagonalize $\mathbf{X}$ in that manner.

This property enables a simple extension from PCA to SVD, since according to Eq. (1.1), $\mathbf{U}^T\mathbf{X} = \mathbf{\Sigma}\mathbf{V}^T$, who are conveniently both orthogonal.

1.4 Issues with traditional PCA and SVD

The traditional PCA method seeks to maximize the L2-norm of the variance of the projected coordinates on the PCs or equivalently to minimize the Euclidean distance between the original data points and their projection onto the subspace spanned by the PCs. However, because of its emphasis on the square of residues or projections, any data analysis methods that involve the L2-norm are prone to corruption from outliers.

As a result, there have has been considerable research effort in reformulating the conventional PCA problem to make it more robust, including weighted PCA (WPCA) [16, 17], where a weight is assigned to every entry of the residual matrix, which is supposed to converge to zero for outliers and and missing data and thus remove their adverse effect on the low-rank approximation. Another approach is Robust Subspace Learning [18], which is a continuous optimization framework based on robust M-estimator.

However, among all robust PCA approaches, the simple replacement of the L2-norm in traditional PCA formulation by the elementwise L1-norm (sum of absolute values) stands out as the approach with the most straightforward algebraic expression, which makes room for closed-form solutions and algorithms [19, 20, 21, 22]. By not enforcing squared emphasis on the residuals or projections, the effect of extreme data points are curtailed and the PCs found are thus more outlier-resistant.

Another line of research with a comparably simple mathematical formulation is the Robust Principal-Component Analysis (RPCA) [23], which decomposes an outlier-corrupted data

matrix into a low-rank and sparse component, whose nuclear and L1- norms are maximized, respectively. This approach has been proven to be the state-of-the-art in the world of robust low-rank approximation and will be compared against throughout this thesis.

Chapter 2

Technical background

2.1 L1-norm Principal-Component Analysis (L1-PCA)

This project builds on existing progress in the field of L1-norm based PCA, which is mathematically stated by replacing the L2-norms in the 3 optimization problems in section 1.1 with the L1-norms, under which they are no longer equivalent due to loss of the Projection Theorem [21]. One formulation that has attracted considerable attention is the L1-norm projection maximization problem, which will be referred to as "L1-PCA" going forward. The problem is written to be

$$\mathbf{Q}_{L1} = \underset{\mathbf{Q} \in \mathbb{S}^{D \times K}}{\operatorname{argmax}} ||\mathbf{Q}^T \mathbf{X}||_{1,1}. \tag{2.1}$$

2.1.1 Properties of L1-PCA

The L1-PCs found by L1-PCA no longer inherit certain attractive properties of the conventional PCs. First of all, it is no longer scalable, meaning that the values of the individual PCs now depend on the number of PCs being found. For example, the first PC $\mathbf{q}_1$ found

by maximizing $||\mathbf{q}_1^T\mathbf{X}||_{1,1}$ is not necessarily one column vector of $\mathbf{Q}_{L1}$ from Eq. (2.1) for any $K > 1$. Thus, there now exists a chasm between a greedy solution where each PCs are found individually knowing the previous PCs and a non-greedy solution where the PCs are found jointly.

In addition, the diagonality property no longer holds true, i.e., $\mathbf{Q}_{L1}^T\mathbf{X}\mathbf{X}^T\mathbf{Q}_{L1}$ is not diagonal or $\mathbf{Q}_{L1}^T\mathbf{X}$ is not orthogonal. This is due to the conventional PCs $\mathbf{Q}_{L2}$ being the only orthonormal matrix that can diagonalize $\mathbf{X}$ in that fashion.

2.1.2 Optimal Algorithm for L1-PCA

Problem Eq. (2.1) is non-convex due to the orthonormality constraint on $\mathbf{Q}$. However, a path to the optimal solution can be revealed by proving that the problem is equivalent to [21]

$$\mathbf{B}_{\text{opt}} = \underset{\mathbf{B}\in\{\pm 1\}^{N\times K}}{\text{argmax}} ||\mathbf{X}\mathbf{B}||_*, \tag{2.2}$$

where $\mathbf{B}_{\text{opt}}$ is an antipodal binary matrix and $(\cdot)_*$ denotes the nuclear norm, which is the L1-norm of the SVs of the argument. The optimal $\mathbf{Q}_{L1}$ for Eq. (2.1) can be found by $\mathbf{Q}_{L1} = \mathcal{U}(\mathbf{X}\mathbf{B}) = \mathbf{U}'\mathbf{V}'^T$ where $(\mathbf{U}', \mathbf{\Sigma}', \mathbf{V}') = \text{SVD}(\mathbf{X}\mathbf{B}_{\text{opt}})$ and $\mathcal{U}(\cdot)$ means to find the closest orthonormal matrix using the Procrustes theorem [11].

From Eq. (2.2), the L1-PCA problems simply becomes a combinatorial problem by testing the metric $||\mathbf{Q}^T\mathbf{X}||_{1,1}$ on 2^{NK} possible instances of $\mathbf{B}$. However, the exponential complexity of such a brute-force approach renders it infeasible even for small data size.

Thus, to reduce the exponential complexity of the optimal algorithm, Markopoulos et. al. [21] proposed a method to reduce the search range of $\mathbf{B}$ to a subset of $\{\pm 1\}^{N\times K}$ in which an

optimal solution must exists. This optimal algorithm incurs a lower cost of $\mathcal{O}(N^{DK-K+1})$.

2.1.3 Suboptimal Algorithms to find the first L1-PC

Alternatively, optimality has been sacrificed in favor of lower computational complexity. This section looks at such algorithms for the simplest case of $K = 1$. When finding the first L1-PC, the problem in Eq. (2.1) becomes a vector optimization problem

$$\mathbf{q}_{L1} = \underset{\mathbf{q}\in\mathbb{S}^{D\times 1}}{\operatorname{argmax}} ||\mathbf{q}^T\mathbf{X}||_{1,1}, \tag{2.3}$$

which is similarly proven to be equivalent to [21]

$$\mathbf{b}_{\text{opt}} = \underset{\mathbf{b}\in\{\pm 1\}^{N\times 1}}{\operatorname{argmax}} ||\mathbf{X}\mathbf{b}||_2. \tag{2.4}$$

The optimal $\mathbf{q}_{L1}$ can be approximated from $\mathbf{b}_{\text{opt}}$ by

$$\mathbf{q}_{L1} = \frac{\mathbf{X}\mathbf{b}_{\text{opt}}}{||\mathbf{X}\mathbf{b}_{\text{opt}}||_2}. \tag{2.5}$$

Fixed-point iterative algorithm

Kwak [19] proposed an iterative algorithm to solve Eq. (2.3) via finding the antipodal binary vector $\mathbf{b}_{\text{opt}}$ from Eq. (2.4). The algorithm can be summarized to be

$$\mathbf{b}^{(t)} = \operatorname{sgn}\left(\mathbf{X}^T\mathbf{X}\mathbf{b}^{(t-1)}\right), \qquad t = 2, 3, 4, ..., \tag{2.6}$$

where $\mathbf{b}^{(1)} \subset \{\pm 1\}^N$ is a binary vector that can be randomly initialized. The iteration is guaranteed to converge to a fixed $\mathbf{b}$ as soon as $\mathbf{b} = \operatorname{sgn}\left(\mathbf{X}^T\mathbf{X}\mathbf{b}\right)$. The first L1-PC is then found following Eq. (2.5). The time complexity of this algorithm is $\mathcal{O}(MDN)$ where M is

the maximum number of iterations. If M is considered to be bounded by N, the complexity becomes $\mathcal{O}(DN^2)$ [19, 22].

Bit-flip algorithm

Markopoulos et al. [22] proposed an algorithm that calculates the effect of flipping any of the N bits of the binary vector $\mathbf{b}$ on the optimization metric of Eq. (2.4). First, it is proven that the metric in Eq.(2.4) is equivalent to

$$||\mathbf{X}\mathbf{b}_{\text{opt}}||_2 = ||\mathbf{Y}\mathbf{b}_{\text{opt}}||_2, \tag{2.7}$$

where $\mathbf{Y} = \mathbf{\Sigma}\mathbf{V}^T$ and $\mathbf{U}, \mathbf{\Sigma}, \mathbf{V} = \text{SVD}(\mathbf{X})$. The effect of flipping the n^{th} bit of $\mathbf{b}$ is

$$\delta_{||\mathbf{Y}\mathbf{b}||_2}(n) = 2\left(b_n \mathbf{y}_n^T \mathbf{Y}\mathbf{b} - ||\mathbf{y}_n||_2^2\right), \tag{2.8}$$

which can be calculated at a lower cost than directly finding the difference of $||\mathbf{Y}\mathbf{b}||_2$ at 2 different $\mathbf{b}$ values. Then, the algorithm decides to flip the bit with the highest $\delta_{||\mathbf{Y}\mathbf{b}||_2}(n)$ among all N possible bit-flips, until no further bit-flips can increase $||\mathbf{X}\mathbf{b}||_2$.

2.1.4 Suboptimal Algorithms to find multiple L1-PCs

Greedy algorithms based on Successive Nullspace Projection

When finding multiple L1-PCs ($K > 1$), one can use a greedy approach in which the k^{th} PC can be successively found on the projection of $\mathbf{X}$ onto the nullspace of the previous $k-1$ L1-PCs, i.e. solving [19]

$$\mathbf{q}_k = \underset{\mathbf{q} \in \mathbb{S}^{D\times 1}}{\text{argmax}} \left|\left| \mathbf{q}^T \left(\mathbf{I}_D - \sum_{i=1}^{k-1} \mathbf{q}_i \mathbf{q}_i^T \right) \mathbf{X} \right|\right|_{1,1}, \tag{2.9}$$

using any of the approaches in section 2.1.3. It is important to note that because L1-PCA is not scalable, meaning the PCs themselves are dependent on the number of PCs being found, the greedy approaches are less optimal than finding the PCs $\mathbf{q}_i$ jointly, whose algorithms will be discussed in the next sections.

Iterative Alternating Algorithm

Nie et. al [20] made the first significant contribution to finding the column vectors of $\mathbf{Q}$ jointly. The iterative algorithm can be summarized to be

$$\mathbf{B}^{(t)} = \text{sgn}(\mathbf{X}^T\mathbf{Q}^{(t-1)}), \mathbf{Q}^{(t)} = \mathcal{U}(\mathbf{X}\mathbf{B}^{(t)}) \qquad (t = 2, 3, 4, ...), \tag{2.10}$$

and $\mathbf{B}^{(1)} \subset \{\pm 1\}^{N\times K}$ is an antipodal binary matrix that can be randomly initialized. Note that for $K = 1$, the algorithm is identical to the approach of [19] in the previous section. The complexity of this algorithm is $\mathcal{O}[M(DN + K^2)]$ where M is the maximum number of iterations. By considering M to be of the same order of magnitude as NK, the complexity becomes $\mathcal{O}(DN^2K + NK^3)$ [20, 22].

Bit-flipping Algorithm

The bit-flipping algorithm [22] can be extended to jointly find the K L1-PCs. Similarly, the effect off flipping the (n, k) bit of the now $N \times K$ antipodal binary matrix $\mathbf{B}$ on the optimization metric $||\mathbf{XB}||_* = ||\mathbf{YB}||_*$ is

$$\delta_{||\mathbf{YB}||_*}(n, k) = ||\mathbf{YB} - 2\mathbf{B}_{n,k}\mathbf{y}_n\mathbf{e}_{k,K}^T||_*, \tag{2.11}$$

where $\mathbf{e}_{k,K}$ is the k^{th} column of the identity matrix $\mathbf{I}_K$. Similarly, the algorithm decides to flip the bit with the highest $\delta_{||\mathbf{YB}||_*}(n,k)$ among all possible NK bit-flips. This algorithm converges to the optimal L1-PCs with high frequency and achieves the highest value in the optimization metric of Eq. (2.1) than any previously known L1-PCA algorithms with similar computational costs. The complexity of this algorithm is $\mathcal{O}(ND\min\{N,D\}+N^2K^2(K^2+d))$ where $d=\text{rank}(\mathbf{X})$.

2.2 L1-low-rank approximation (L1-LR)

As mentioned in the previous sections, an alternative approach to find the L1-PCs is solving the L1-norm equivalence of Eq. (1.5), i.e.,

$$\mathbf{R}_{L1}, \mathbf{Z}_{L1} = \underset{\mathbf{R}\in\mathbb{R}^{D\times K}, \mathbf{Z}\in\mathbb{R}^{K\times N},}{\text{argmin}} ||\mathbf{X}-\mathbf{RZ}||_{1,1}, \tag{2.12}$$

which we shall refer to as the L1-factorization or L1-low-rank approximation problem (L1-LR).

Alternating convex optimization

The problem in Eq. (2.12) is non-convex when $\mathbf{R}$ and $\mathbf{Z}$ are found concurrently. Thus, Ke and Kanade [24] came up with an alternating algorithm to solve Eq. (2.12) suboptimally. When $\mathbf{R}$ is fixed to find $\mathbf{Z}$, the problem becomes

$$\mathbf{Z} = \underset{\mathbf{Z}\in\mathbb{R}^{K\times N},}{\text{argmin}} ||\mathbf{X}-\mathbf{RZ}||_{1,1}, \tag{2.13}$$

which is convex and can be solved using standard convex optimization methods [25]. As soon as an optimal $\mathbf{Z}$ is found, it can be used to find $\mathbf{R}$ by solving the same convex optimization problem, whose result is then used to update $\mathbf{Z}$. The algorithm is iterated until $\mathbf{R}$ and $\mathbf{Z}$ converge.

Uniform feature preservation

Tsagkarakis et. al. [26] proposed another suboptimal algorithm to solve Eq. (2.12) by taking advantage of the fact that any column vector of the optimal solution $[\mathbf{R}_{L1}\mathbf{Z}_{L1}]_{:,n}$ will be equal to $\mathbf{x}_n$ in at least K entries. The algorithm is simplified to achieve low computational complexity by enforcing uniform feature preservation, i.e., forcing the indices d where $[\mathbf{R}_{L1}\mathbf{Z}_{L1}]_{d,n} = x_{d,n}$ to be the same across all N columns, which is not always the case but has been nevertheless observed to happen with high frequency. In doing so, the search range for the indices combination can be substantially limited, thus reducing the computational complexity.

2.3 Robust Principal-Component Analysis (RPCA)

Another line of research to the problem of robustly estimating a low-rank matrix from a corrupted matrix is the Robust Principal-Component Analysis (RPCA) [23]. This approach seeks to decompose a matrix $\mathbf{X}$ into a low-rank component $\mathbf{L}$ and a sparse component $\mathbf{S}$, the latter of which models the outliers that we refer to throughout this work. Ideally, these criteria can be satisfied by solving the problem

$$\mathbf{L}_{\text{opt}}, \mathbf{S}_{\text{opt}} = \underset{\mathbf{L},\mathbf{S},\mathbf{L}+\mathbf{S}=\mathbf{X}}{\text{argmin}} \ \text{rank}(\mathbf{L}) + \lambda ||\mathbf{S}||_{0,0}, \tag{2.14}$$

where the zero-norm $||\cdot||_{0,0}$ gives the number of non-zero entries. Unfortunately, both the rank function and the zero-norm are both known to be non-convex and the problem is NP-hard, leading to no efficient solutions existing in literature. However, the problem can be reformulated by replacing the rank function with the nuclear norm $(\cdot)_*$ and the zero-norm by the L1-norm, which are both convex, i.e.,

$$\mathbf{L}_{\text{opt}}, \mathbf{S}_{\text{opt}} = \underset{\mathbf{L},\mathbf{S},\mathbf{L}+\mathbf{S}=\mathbf{X}}{\operatorname{argmin}} ||\mathbf{L}||_* + \lambda||\mathbf{S}||_{1,1}. \tag{2.15}$$

The problem essentially promotes the sparsity of $\mathbf{S}$ by minimizing its L1-norm and minimizes the rank of $\mathbf{L}$ by enforcing sparsity of its SVs by minimizing their L1-norm, or equivalently the nuclear norm of $\mathbf{L}$. A popular choice for the parameter λ is $1/\sqrt{M}$ where M is the larger dimension of $\mathbf{X}$ and we will use this parameter value when implementing RPCA [23].

The optimization problem in Eq. (2.15) is convex and thus solvable by standard convex optimization methods [25]. Some algorithms for RPCA can be found in [23, 27, 28, 29, 30].

RPCA can be used for robust SVD by simply taking the conventional SVD of the extracted low-rank component $\mathbf{L}$.

2.4 L1-norm Singular Spectrum Analysis

One approach that aims to utilize the robustness of the L1-norm for SVs estimation is the L1-norm Singular Spectrum Analysis (L1-SSA) [31], in which conventional SVD is first performed on the data matrix $\mathbf{X}$ to obtain the left and right singular vectors $\mathbf{U} \in \mathbb{S}^{D\times K}$ and $\mathbf{V} \in \mathbb{S}^{N\times K}$. Then, the L1-SSA method robustifies the SVs in case outliers are present by

solving the following L1-norm minimization problem

$$\Sigma_{\mathrm{L1-SSA}} = \underset{\Sigma \in \mathrm{diag}(\mathbb{R}^K)}{\operatorname{argmin}} ||\mathbf{X} - \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T||_{1,1}. \tag{2.16}$$

This is a convex problem and is solvable by standard convex optimization methods [25].

Chapter 3

Proposed solution

Besides making the PCs, or equivalently the left singular vectors of SVD, more robust against sparse and gross outliers, a robust, outlier-resistant acquisition of singular values is also of great interest. There currently exists some robust SVD approaches, including one that utilizes the least-trimmed square (the sum of squares of a subset of data points) to eliminate the influence of erroneous data points [32], a robust regularized SVD that develops a fast iterative reweighted least square algorithm [33], and an Lp-norm SVD ($0 < p < 1$) where the Lp-norm of the weighted residual matrix is minimized [34].

However, these robust SVD methods mainly focus on finding the low-rank matrix without any elaborate discussion on the robustness of SVs estimation, which is the main focus of this work. In addition, we aim to derive an algebraically straightforward SVs estimation scheme akin to the existing L1-PCA approaches and attempt to find a closed-form algorithm to tackle the formulated problem.

Regretfully, the orthonormal basis $\mathbf{Q}_{L1}$ found by L1-PCA, while robust against outliers, does not possess the attractive property of its L2-norm counterpart to diagonalize the data matrix $\mathbf{X}$ [35], thus making the extension from L1-PCA to SVs estimation non-trivial.

First, we formulate our L1-norm based SVD approach to be

$$\mathbf{X} \approx \mathbf{U}_{L1}\mathbf{\Sigma}_{L1}\mathbf{V}_{L1}^{T}, \tag{3.1}$$

where the left and right singular vectors $\mathbf{U}_{L1}$ and $\mathbf{V}_{L1}$ are still orthonormal and $\mathbf{\Sigma}_{L1}$ is still diagonal. As a result, this decomposition has to be approximated because the only exact decomposition with such constraints on $\mathbf{U}_{L1}, \mathbf{\Sigma}_{L1}$ and $\mathbf{V}_{L1}$ would be the solution of conventional SVD due to its uniqueness property. We carry over the property of SVD that the left singular vectors are also the PCs and make $\mathbf{U}_{L1}$ the L1-PCs, which has been thoroughly examined in the previous section. The main contribution of this thesis will be to find the SVs $\mathbf{\Sigma}_{L1}$ and the right singular vectors $\mathbf{V}_{L1}$.

3.1 Naive approaches

3.1.1 Direct extension of L1-PCA

Naively, as soon as the L1-PCs are found by solving Eq. (2.1) using one of the mentioned method, one may perform conventional SVD on the data projected on the L1-PCs, i.e., $\mathbf{Q}_{L1}\mathbf{Q}_{L1}^{T}\mathbf{X} = \mathbf{U}_{L1}\mathbf{\Sigma}_{L1}\mathbf{V}_{L1}^{T}$. We will refer to this SVD approach as "L1-PCA" in the rest of the discussion.

However, the SVs $\mathbf{\Sigma}_{L1}$ found using this method is very close to the SVs found by the conventional L2-cSVD and thus are not robust against outliers, as will be demonstrated in section 4, which calls for more sophisticated methods to find new $(\mathbf{\Sigma}_{L1}, \mathbf{V}_{L1})$ combinations.

3.1.2 Direct extension of L1-low-rank approximation (L1-LR)

In addition, from the solutions to the L1-low-rank approximation problem in Eq. (2.12), we can simply perform the conventional SVD on the low-rank matrix $\mathbf{R}_{L1}\mathbf{Z}_{L1}$ closest to $\mathbf{X}$ in the L1-norm, i.e. $\mathbf{U}_{L1}, \mathbf{\Sigma}_{L1}, \mathbf{V}_{L1} = \text{SVD}(\mathbf{R}_{L1}\mathbf{Z}_{L1})$ to obtain our L1-norm based SVD. However, like L1-PCA, this simplistic approach will be empirically proven incapable of robustly estimating SVs.

3.2 Preliminary algorithm: L1L2-cSVD

In this section we experiment with a simple alternative to finding the SVs from the left singular vectors $\mathbf{U}_{L1}$, which is chosen to be the L1-PCs from the L1-projection maximization problem

$$\mathbf{U}_{L1} = \underset{\mathbf{U} \in \mathbb{S}^{D \times K}}{\text{argmax}} ||\mathbf{U}^T \mathbf{X}||_{1,1}. \tag{3.2}$$

This choice of $\mathbf{U}_{L1}$ ensures that the subspace found is robust against outliers [21, 22]. As mentioned in section 2.1, the problem in Eq. (3.2) has been studied extensively and there are multiple algorithms available to choose from, the importance of which will be discussed in detail in section 3.4. For now, we assume that a good $\mathbf{U}_{L1}$ can be found.

As discussed in section 1.3.3, conventional SVD can diagonalize $\mathbf{X}$, i.e., $\mathbf{U}_{L2}^T \mathbf{X} = \mathbf{\Sigma}_{L2} \mathbf{V}_{L2}^T$ is an orthogonal matrix, while an orthonormal $\mathbf{U}_{L1}$ from Eq. (3.2) generally cannot. On the other hand, the definition of L1-cSVD in Eq. (3.1) also requires that $\mathbf{\Sigma}_{L1} \mathbf{V}_{L1}^T$ be orthogonal. Thus, the problem of L1-cSVD becomes finding the closest orthogonal matrix $\mathbf{\Sigma}_{L1} \mathbf{V}_{L1}^T$ to $\mathbf{U}_{L1}^T \mathbf{X}$. In this algorithm, we experiment with a quick approach to optimize this criteria with

the L2-norm, hence the name "L1L2-cSVD",

$$(\mathbf{\Sigma}_{L1}, \mathbf{V}_{L1}) = \underset{\mathbf{V}\in\mathbb{S}^{N\times K}, \Sigma\in\text{diag}(\mathbb{R}^K)}{\text{argmin}} ||\mathbf{U}_{L1}^T\mathbf{X} - \mathbf{\Sigma}\mathbf{V}^T||_{2,2}. \tag{3.3}$$

This is a non-convex problem due to the orthonormality constraint on $\mathbf{V}_{L1}$ [25]. Thus, we will employ an alternating method to find a solution for a diagonal $\mathbf{\Sigma}_{L1}$ and an orthonormal $\mathbf{V}_{L1}$. When $\mathbf{V}$ is fixed to find $\mathbf{\Sigma}$, the problem becomes K individual problems

$$\sigma_i = \underset{\sigma_i}{\text{argmin}} ||(\mathbf{X}^T\mathbf{U}_{L1})_{:,i} - \sigma_i\mathbf{v}_i||_2 \qquad (i = 1, 2, ..., K), \tag{3.4}$$

where $\sigma_i = \mathbf{\Sigma}_{i,i}$ is the ith SV. This problem is simply asking for a multiplying coefficient σ_i to minimize the Euclidean distance between 2 vectors $(\mathbf{X}^T\mathbf{U}_{L1})_{:,i}$ and $\sigma_i\mathbf{v}_i$. The solution is achieved when the projection of the latter on the former is exactly the same as former, in other words

$$\sigma_i = \frac{[(\mathbf{X}^T\mathbf{U}_{L1})_{:,i}]^T\mathbf{v}_i}{||\mathbf{v}_i||_2} \qquad \forall\, i = 1, 2, \ldots, K. \tag{3.5}$$

On the other hand, when $\mathbf{\Sigma}$ is fixed to find $\mathbf{V}$, the problem in Eq. (3.3) becomes

$$\mathbf{V} = \underset{\mathbf{V}\in\mathbb{S}^{N\times K}}{\text{argmin}} ||\mathbf{X}^T\mathbf{U}_{L1} - \mathbf{V}\mathbf{\Sigma}||_{2,2}, \tag{3.6}$$

which can be proven to be equivalent to

$$\mathbf{V} = \underset{\mathbf{V}\in\mathbb{S}^{N\times K}}{\text{argmin}} ||\mathbf{X}^T\mathbf{U}_{L1}\mathbf{\Sigma}^{-1} - \mathbf{V}||_{2,2}. \tag{3.7}$$

This problem asks to find an orthonormal matrix $\mathbf{V}$ closest to $\mathbf{X}^T\mathbf{U}_{L1}\mathbf{\Sigma}^{-1}$, better known

as the Orthogonal Procrustes problem, whose solution is $\mathbf{V} = \mathbf{U}'\mathbf{V}'^T$ where $(\mathbf{U}', \mathbf{\Sigma}', \mathbf{V}') = \text{SVD}(\mathbf{X}^T\mathbf{U}_{L1}\mathbf{\Sigma}^{-1})$ [11]. Thus, the algorithm can be summarized in the pseudocode provided below.

Algorithm 1 L1L2-cSVD

Input: Data matrix $\mathbf{X}_{D\times N}$, number of SVs K,
1: $\mathbf{U} = \text{L1PCA}(\mathbf{X})$
2: $\mathbf{A} = \mathbf{X}^T\mathbf{U}$
3: **initialization** $\mathbf{\Sigma} = \text{zeros(K, K)}$, orthonormal $\mathbf{V}$
4: **while** not converged **do**
5: **for** i = 1 to K **do**
6: $[\mathbf{\Sigma}]_{\text{i,i}} = [\mathbf{A}]_{:,\text{i}}^{\text{T}}[\mathbf{V}]_{:,\text{i}}/|||[\mathbf{V}]_{:,\text{i}}||_2$
7: **end for**
8: $(\mathbf{U}', \mathbf{\Sigma}', \mathbf{V}') = \text{SVD}(\mathbf{A}\mathbf{\Sigma}^{-1})$
9: $\mathbf{V} = \mathbf{U}'\mathbf{V}'^{\text{T}}$
10: **end while**
Output: $\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}$

Running K instances to find the SVs σ_i costs $\mathcal{O}(KN)$. The SVD of the $N \times K$ matrix $\mathbf{X}^T\mathbf{U}\mathbf{\Sigma}^{-1}$ and the subsequent matrix multiplication $\mathbf{U}'\mathbf{V}'^T$ to solve the Procrustes problem costs an extra $\mathcal{O}(NK^2)$. Thus, the complexity of finding $\mathbf{\Sigma}_{L1}$ and $\mathbf{V}_{L1}$ is $\mathcal{O}(MNK^2)$, where M is the number of iterations. Consider M to be bounded by NK, the total complexity of L1L2-cSVD is $\mathcal{O}(N^2K^3)$ in addition to the complexity of the L1-PCA algorithm chosen to find $\mathbf{U}_{L1}$.

3.3 Proposed Algorithm: L1-cSVD

In this section, we detail the proposed algorithm that further improves the robustness of $\mathbf{\Sigma}_{L1}$ and $\mathbf{V}_{L1}$ by finding the closest orthogonal matrix to the projected coordinates on the L1-PCs $\mathbf{U}_{L1}\mathbf{X}$ in the L1-norm instead of the Frobenius norm like in L1L2-cSVD, i.e.,

$$(\mathbf{\Sigma}_{L1}, \mathbf{V}_{L1}) = \underset{\mathbf{V}\in\mathbb{S}^{N\times K}, \Sigma\in\text{diag}(\mathbb{R}^K)}{\text{argmin}} ||\mathbf{X}^T\mathbf{U}_{L1} - \mathbf{V}\mathbf{\Sigma}||_{1,1}. \tag{3.8}$$

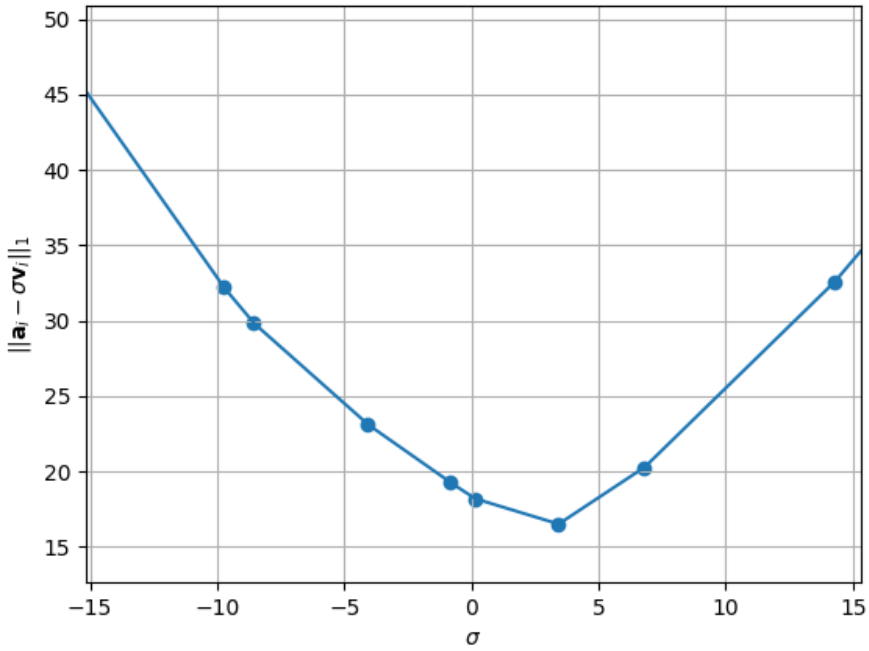

Figure 3.1: An example of how the L1-norm metric in Eq. (3.9) changes with different σ. The blue circles mark the candidates σ, in between which the metric can be seen to change linearly and thus cannot attain a minimum. Note that $\mathbf{A} = \mathbf{X}^T\mathbf{U}_{L1}$.

We call our algorithm L1-cSVD to emphasize that we only collect $K \leq D \leq N$ SVs and singular vectors from $\mathbf{X}$. We will solve for the matrices $\mathbf{\Sigma}_{L1}$ and $\mathbf{V}_{L1}$ suboptimally by an alternating method. When fixing $\mathbf{V}$ to find $\mathbf{\Sigma}$, the problem can be decomposed into K individual problems

$$\sigma_i = \underset{\sigma_i}{\operatorname{argmin}} ||(\mathbf{X}^T\mathbf{U}_{L1})_{:,i} - \sigma_i\mathbf{v}_i||_1, \qquad (i = 1, 2, ..., K), \tag{3.9}$$

where $\sigma_i = \mathbf{\Sigma}_{i,i}$ is the i^{th} SV. This problem is simply asking for a scaling factor σ_i that minimizes the L1-distance between 2 vectors $(\mathbf{X}^T\mathbf{U}_{L1})_{:,i}$ and $\sigma_i\mathbf{v}_i$. This problem is proven

in [21] to be equivalent to

$$j_{\text{opt}} = \underset{j\in\{1:N\}}{\text{argmin}} \left|\left|(\mathbf{X}^T\mathbf{U}_{L1})_{:,i} - \frac{(\mathbf{X}^T\mathbf{U}_{L1})_{j,i}}{v_{j,i}}\mathbf{v}_i\right|\right|_1 \qquad (i = 1, 2, ..., K), \tag{3.10}$$

which performs exhaustive search on N candidates σ_i chosen such that $(\mathbf{X}^T\mathbf{U}_{L1})_{:,i}$ is equal to $\sigma_i\mathbf{v}_i$ on the j^{th} entry. From the N candidates, the one that returns the least L1 error will be chosen to be the optimal σ_i. On the other hand, when $\mathbf{\Sigma}$ is fixed to find $\mathbf{V}$, the problem in Eq. (3.8) becomes

$$\mathbf{V} = \underset{\mathbf{V}\in\mathbb{S}^{N\times K}}{\text{argmin}} ||\mathbf{X}^T\mathbf{U}_{L1} - \mathbf{V}\mathbf{\Sigma}||_{1,1}, \tag{3.11}$$

which is essentially a L1-Procrustes problem. A solution to this problem using a smoothed version of the L1-norm has been studied in [36]. However, in this work, for lower computational complexity, we will use the solution to the L2-Procrustes problem instead, since it is empirically observed that the L1-Procrustes solution gives similar results to the L2- counterpart while taking much longer to solve. Thus, similarly from section 3.2, $\mathbf{V} = \mathbf{U}'\mathbf{V}'^T$ where $(\mathbf{U}', \mathbf{\Sigma}', \mathbf{V}') = \text{SVD}(\mathbf{X}^T\mathbf{U}_{L1}\mathbf{\Sigma}^{-1})$. The iterations are continued until $\mathbf{\Sigma}$ and $\mathbf{V}$ converge. The algorithm can be summarized in the pseudocode below.

Finding $\mathbf{A}$ costs $\mathcal{O}(NDK)$. Finding $||\mathbf{A}_{:,i} - s\mathbf{V}_{:,i}||_1$ costs $\mathcal{O}(N)$ for a candidate s. Since there are N candidates for K SVs, finding $\mathbf{\Sigma}_{L1}$ costs $\mathcal{O}(N^2K)$ in total. Finally, $\mathbf{V}$ is found with cost $\mathcal{O}(NK^2)$. Because $N \geq D \geq K$, the complexity of finding $\mathbf{\Sigma}_{L1}$ and $\mathbf{V}_{L1}$ is $\mathcal{O}(MKN^2)$, where M is the number of iterations. By considering M to be bounded by NK, the complexity of this L1-cSVD algorithm is $\mathcal{O}(N^3K^2)$ in addition to the cost of the L1-PCA algorithm chosen to find $\mathbf{U}_{L1}$.

Algorithm 2 L1-cSVD

Input: Data matrix $\mathbf{X}_{D\times N}$, number of SVs K

$\mathbf{U}$ = L1PCA ($\mathbf{X}$)
$\mathbf{A} = \mathbf{X}^T\mathbf{U}$
initialization $\mathbf{\Sigma}$ = zeros(K,K), orthonormal $\mathbf{V}$
while not converged **do**
    for i = 1 to K **do**
        for j = 1 to N **do**
            $\mathrm{s_j} = ([\mathbf{A}]_{\mathrm{j,i}}/[\mathbf{V}]_{\mathrm{j,i}})$
            $\mathrm{M_j} = ||[\mathbf{A}]_{:,\mathrm{i}} - \mathrm{s}[\mathbf{V}]_{:,\mathrm{i}}||_1$
        end for
        $j^{\mathrm{opt}} = \underset{j\in[1:N]}{\mathrm{argmin}}\{M_j\}$
        $[\mathbf{\Sigma}]_{\mathrm{i,i}} = \mathrm{s}_{\mathrm{j^{opt}}}$
    end for
    $(\mathbf{U}', \mathbf{\Sigma}', \mathbf{V}') = \mathrm{SVD}(\mathbf{A}\mathbf{\Sigma}^{-1})$
    $\mathbf{V} = \mathbf{U}'\mathbf{V}'^T$
end while

Output: $\mathbf{U}_{L1} = \mathbf{U}, \mathbf{\Sigma}_{L1} = \mathbf{\Sigma}, \mathbf{V}_{L1} = \mathbf{V}$

3.4 Importance of Choosing Left Singular Vectors $\mathbf{U}_{L1}$: Joint vs Greedy

As previously mentioned, there are many algorithms to solve the L1-PCA problem of Eq. (3.2) to find the left singular vectors $\mathbf{U}_{L1}$, on which $\mathbf{\Sigma}_{L1}$ and $\mathbf{V}_{L1}$ are dependent. The main difference is between the Greedy solutions, such as in [19] and Non-greedy or Joint solutions, such as in [20] and [22]. Because L1-PCA is not scalable, the Joint solutions have a higher, more optimal $||\mathbf{U}_{L1}^T\mathbf{X}||_{1,1}$ metric. However, it is observed that while the Joint solution finds a better subspace, the L1-PCs found from the outliers-corrupted matrix, is not necessarily more aligned to the L2-PCs of the clean data. The reason for this is that maximizing the L1-norm in Eq. (3.2) promotes balance in $||\mathbf{u}_i^T\mathbf{X}||_1$, meaning that apart from maximizing the data L1-projection, the Joint L1-PCA algorithms also inadvertently rotate the basis vectors, i.e., the individual L1-PCs, to make the L1-projections more balanced. On the other hand, this issue is ameliorated by the Greedy solution since it focuses on maximizing the L1-norm of the projection to one particular L1-PC without having to balance with other L1-PCs.

Coincidentally, since this work is concerned with SVD, particularly the approximation of SVs under corruption by outliers, finding good PCs should be given a priority to finding a good subspace. The reason is that for SVD, the SVs are directly tied to their corresponding PCs, so the SVs found with good PCs are more meaningful than the SVs found in a good subspace where the bases have been rotated, which is often the case with Joint L1-PCA. As a result, we elect to choose the Greedy approach to find $\mathbf{U}_{L1}$ in Eq. (3.2). The proof for this discussion using a numerical study is available in section 4.2.3.

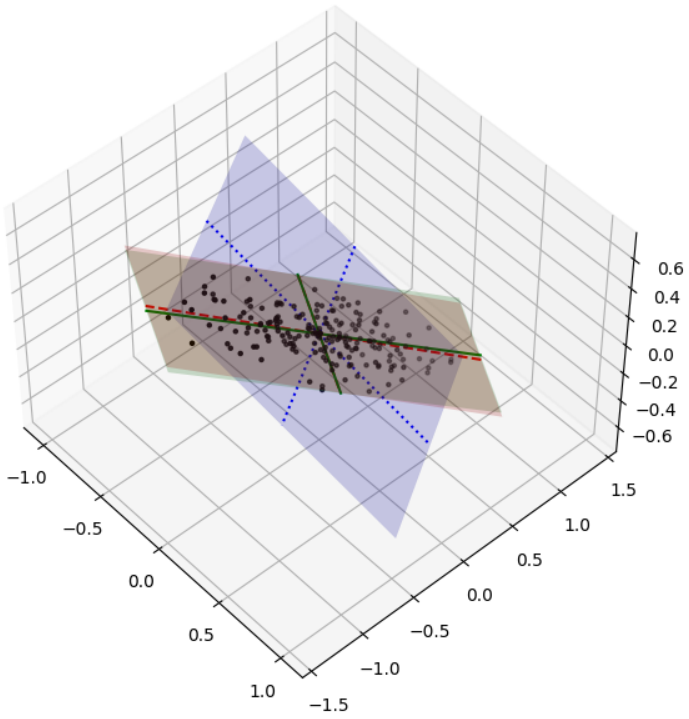

Figure 3.2: The 2 PCs found by PCA (red, dashed), Joint L1-PCA (blue, dotted) and Greedy L1-PCA (green) for the case of $D = 3, N = 100$ and the data has rank $K = 2$. The subspaces spanned by the 2 PCs (in this case the planes) using 3 approaches follow the same color code. The length of the PCs are scaled by the respective SVs found by SVD or L1-cSVD.

For example, in Fig. 3.2, the Greedy L1-PCs are aligned almost perfectly with the conventional L2-norm based PCs. On the other hand, the Joint L1-PCs, despite spanning the

same subspace as the data, are rotated by about 45° to balance the projections on either PCs due to the L1-norm. As a result, the subsequent SVs found from the Joint L1-PCs will be almost equal, not reflecting the actual structure of the data.

Chapter 4

Experimental Studies

4.1 Algorithm Analysis

Convergence

To assess the convergence of the preliminary L1L2-cSVD algorithm, we define the normalized performance measurement

$$\mathrm{M_P(L1L2-cSVD)} = \frac{||\mathbf{U}^T\mathbf{X} - \mathbf{\Sigma}\mathbf{V}^T||_{2,2}}{||\mathbf{U}^T\mathbf{X}||_{1,1}} \tag{4.1}$$

and plot its evolution for 4 different initializations on the same 8×50 data matrix $\mathbf{X}$ ($K = 5$ SVs are obtained) in Fig. 4.2, according to which the L1-cSVD does converge to the same level.

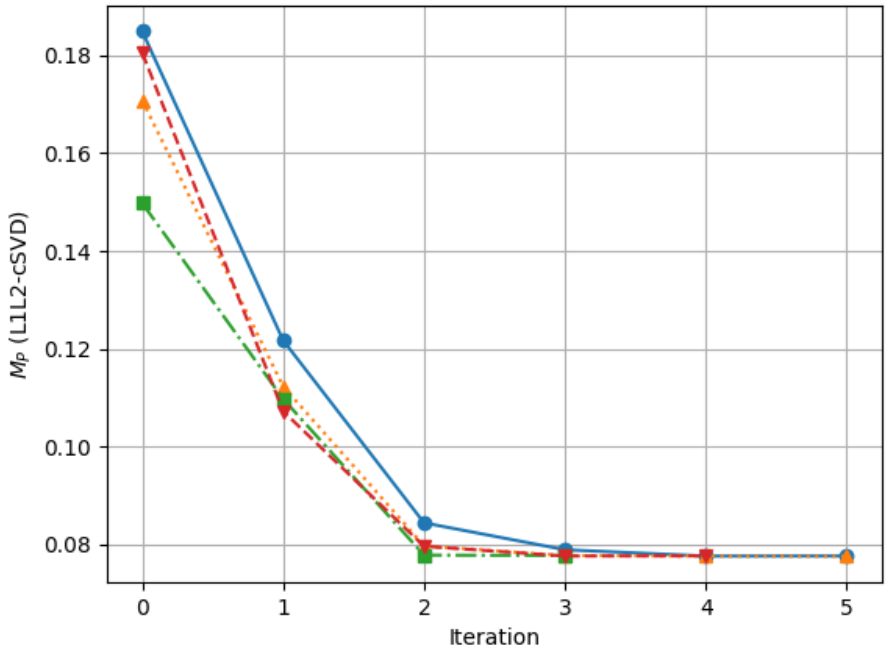

Figure 4.1: Evolution of the performance metric M_P for the L1L2-cSVD algorithm

Similarly, to assess the convergence of the proposed L1-cSVD algorithm, we define the normalized performance measurement,

$$\mathrm{M_P(L1-cSVD)} = \frac{||\mathbf{U}^T\mathbf{X} - \mathbf{\Sigma}\mathbf{V}^T||_{1,1}}{||\mathbf{U}^T\mathbf{X}||_{1,1}} \tag{4.2}$$

and plot its evolution for the same setup.

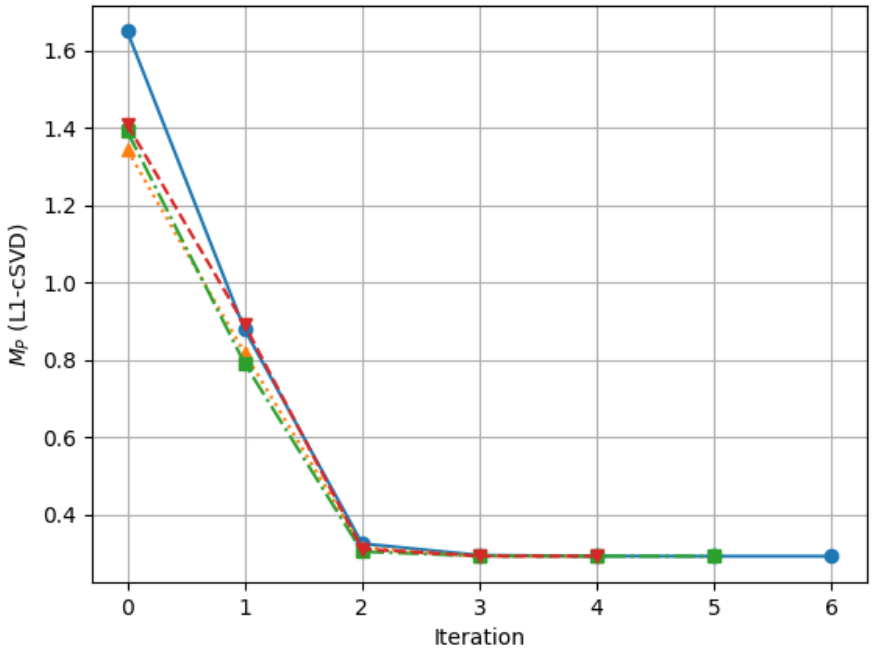

Figure 4.2: Evolution of the performance metric M_P for the L1-cSVD algorithm

4.2 Performance Analysis with Synthetic Dataset

In this section, we will compare the SVs estimation criterion of the preliminary L1L2-cSVD and the proposed L1-cSVD algorithms against the conventional SVD, the simplistic L1-PCA and L1-LR approaches, and the state-of-the-art RPCA algorithm.

4.2.1 Signal Model

Suppose there is a clean data matrix $\mathbf{X}^{\text{clean}} \in \mathbb{R}^{D \times N}$ with a rank-K structure ($K \leq D \leq N$). The clean data lies in the subspace spanned by an orthonormal $\mathbf{U}_0 \in \mathbb{S}^{D \times K}$, which is kept constant for the experiment. $\mathbf{V}_0 \in \mathbb{R}^{N \times K}$ is a random orthonormal matrix and the common logarithm of the SVs $\mathbf{\Sigma}_0$ are drawn from a uniform distribution between 0 and 1. First, $\mathbf{X}^{\text{clean}}$ is corrupted by Gaussian noise $\mathbf{N}$ with a signal-to-noise ratio defined as

SNR $= ||\mathbf{N}||_{2,2}^2 / ||\mathbf{\Sigma}_0||_{2,2}^2$. Then, the noisy data matrix is further corrupted by outlier $\mathbf{O}$.

$$\mathbf{X}^{\text{corrupted}} = \mathbf{X}^{\text{clean}} + \mathbf{N} + \mathbf{O} = \mathbf{U}_0\mathbf{\Sigma}_0\mathbf{V}_0^T + \mathbf{N} + \mathbf{\Gamma} \odot \mathbf{R}_o\mathbf{S}_o. \tag{4.3}$$

The outlier $\mathbf{O}$ is modelled as subspace outlier coming from a subspace spanned by an orthonormal $\mathbf{R}_o \in \mathbb{S}^{D \times K_o}$, which is also kept constant for the experiment. The outlier corrupts random data points (column vectors in $\mathbf{X}^{\text{corrupted}}$) with probability P_o. Thus, $\mathbf{\Gamma} \in \{0,1\}^{D \times N}$ is a matrix with P_o chance of a column vector being $\mathbf{1}$ while the rest are $\mathbf{0}$. The outlier strength is also defined by an outlier-to-signal ratio defined as OSR $= ||\mathbf{O}||_{2,2}^2 / ||\mathbf{\Sigma}_0||_{2,2}^2$.

4.2.2 Algorithms to compare

Name	Problem formulation
SVD (conventional)	$\mathbf{U}, \mathbf{\Sigma}, \mathbf{V} = \text{SVD}(\mathbf{X})$
L1-PCA	$\mathbf{Q} = \underset{\mathbf{Q} \in \mathbb{S}^{D \times K}}{\text{argmax}} ||\mathbf{Q}^T\mathbf{X}||_{1,1}$. then $\mathbf{U}, \mathbf{\Sigma}, \mathbf{V} = \text{SVD}(\mathbf{Q}\mathbf{Q}^T\mathbf{X})$
L1-LR	$\mathbf{R}, \mathbf{Z} = \underset{\mathbf{R} \in \mathbb{R}^{D \times K}, \mathbf{Z} \in \mathbb{R}^{K \times N},}{\text{argmin}} ||\mathbf{X} - \mathbf{R}\mathbf{Z}||_{1,1}$ then $\mathbf{U}, \mathbf{\Sigma}, \mathbf{V} = \text{SVD}(\mathbf{R}\mathbf{Z})$
RPCA	$\mathbf{L}, \mathbf{S} = \underset{\mathbf{L}, \mathbf{S}, \mathbf{L}+\mathbf{S}=\mathbf{X}}{\text{minimize}} ||\mathbf{L}||_* + \lambda ||\mathbf{S}||_{1,1}$ then $\mathbf{U}, \mathbf{\Sigma}, \mathbf{V} = \text{SVD}(\mathbf{L})$
L1L2-cSVD (preliminary)	$\mathbf{U} = \underset{\mathbf{U} \in \mathbb{S}^{D \times K}}{\text{argmax}} ||\mathbf{U}^T\mathbf{X}||_{1,1}$. then $\mathbf{\Sigma}, \mathbf{V} = \underset{\mathbf{\Sigma} \in \text{diag}(\mathbb{R}^K), \mathbf{V} \in \mathbb{S}^{N \times K}}{\text{argmin}} ||\mathbf{X}^T\mathbf{U} - \mathbf{V}\mathbf{\Sigma}||_{2,2}$
L1-cSVD (proposed)	$\mathbf{U} = \underset{\mathbf{U} \in \mathbb{S}^{D \times K}}{\text{argmax}} ||\mathbf{U}^T\mathbf{X}||_{1,1}$. then $\mathbf{\Sigma}, \mathbf{V} = \underset{\mathbf{\Sigma} \in \text{diag}(\mathbb{R}^K), \mathbf{V} \in \mathbb{S}^{N \times K}}{\text{argmin}} ||\mathbf{X}^T\mathbf{U} - \mathbf{V}\mathbf{\Sigma}||_{1,1}$

Table 4.1: Brief summary of all SVD algorithms to be compared in this numerical study

In this section, we detail and motivate the algorithms being compared in the following experimental study. The main baseline algorithm is of course the conventional (L2-norm based) SVD. We then look at the performance of the L1-PCA and L1-LR approaches for SVD, which are the natural extension of existing L1-PCA and L1-low-rank approximation algorithms, in order to demonstrate that an extension to finding the L1-norm based SVs and right singular vectors is not trivial.

The main robust SVD approach in the literature to compare against is RPCA, whose

robustly extracted low-rank component is readily decomposable to find the SVs. The preliminarily developed algorithm L1L2-cSVD using a L2-norm to find $\mathbf{\Sigma}$ and $\mathbf{V}$ using the L2-norm is compared against the proposed L1-cSVD algorithm, which uses the L1-norm for the same purpose, to emphasize the effect of the L1-norm in finding robust SVs.

4.2.3 Subspace and PCs robustness

To corroborate the discussion in section 3.4 on the merits of different approaches, the robustness against outlier of the subspace spanned by the PCs found by different PCA methods detailed in Table 4.2 is evaluated.

Name	Problem formulation to find the PCs $\mathbf{U}^{\text{opt}}$
PCA (conventional)	$\mathbf{U}^{\text{opt}} = \underset{\mathbf{U}\in\mathbb{S}^{D\times K}}{\text{argmax}} \|\|\mathbf{U}^T\mathbf{X}\|\|_{2,2}$
RPCA	$\mathbf{L}^{\text{opt}}, \mathbf{S}^{\text{opt}} = \underset{\mathbf{L},\mathbf{S},\mathbf{L}+\mathbf{S}=\mathbf{X}}{\text{argmin}} \|\|\mathbf{L}\|\|_* + \lambda\|\|\mathbf{S}\|\|_{1,1}$, then $\mathbf{U}^{\text{opt}}, \mathbf{\Sigma}^{\text{opt}}, \mathbf{V}^{\text{opt}} = \text{SVD}(\mathbf{L}^{\text{opt}})$
L1-LR	$\mathbf{R}^{\text{opt}}, \mathbf{Z}^{\text{opt}} = \underset{\mathbf{R}\in\mathbb{R}^{D\times K},\mathbf{R}\in\mathbb{Z}^{K\times N},}{\text{argmin}} \|\|\mathbf{X}-\mathbf{R}\mathbf{Z}\|\|_{1,1}$, then $\mathbf{U}^{\text{opt}}, \mathbf{\Sigma}^{\text{opt}}, \mathbf{V}^{\text{opt}} = \text{SVD}(\mathbf{R}^{\text{opt}}\mathbf{Z}^{\text{opt}})$
L1-PCA (Joint)	$\mathbf{U}^{\text{opt}} = \underset{\mathbf{U}\in\mathbb{S}^{D\times K}}{\text{argmax}} \|\|\mathbf{U}^T\mathbf{X}\|\|_{1,1}$
L1-PCA (Greedy)	$\mathbf{u}_k^{\text{opt}} = \underset{\mathbf{u}\in\mathbb{S}^{D\times 1}}{\text{argmax}} \left\|\left\|\mathbf{u}^T\left(\mathbf{I}_D - \sum_{i=1}^{k-1}\mathbf{u}_i\mathbf{u}_i^T\right)\mathbf{X}\right\|\right\|_{1,1}$

Table 4.2: Brief summary of all PCA approaches to be compared in this numerical study

The L1-PCA problems are solved using the bit-flipping algorithm [22] for both the $K = 1$ (Greedy) and $K > 1$ (Joint) cases, since it converges to the optimal result with the highest frequencies among all suboptimal L1-PCA algorithms of similar computational complexity.

To evaluate the robustness of the subspace spanned by $\mathbf{U}$, the normalized difference R_U (defined below) between the projection matrix $\mathbf{U}\mathbf{U}^T$ evaluated with corrupted data and clean data is plotted at different outlier strengths.

$$R_{\text{U}} = \frac{||(\mathbf{U}\mathbf{U}^T)^{\text{corrupted}} - (\mathbf{U}\mathbf{U}^T)^{\text{clean}}||_{2,2}}{||(\mathbf{U}\mathbf{U}^T)^{\text{clean}}||_{2,2}}. \tag{4.4}$$

In addition, since the SVs are tied to the PCs, it is also important to examine the

robustness of the individual PCs, instead of the subspace they define as a whole. To that end, we defined the normalized PCs alignment metric, which takes the absolute values of the dot product between the PCs pairs with the same significance found from the corrupted and clean data, downscaled by K to ensure R_{PC} does not exceed 1.

$$R_{\text{PC}} = \frac{1}{K} \sum_{i=1}^{K} \left| \left(\mathbf{u}_i^{\text{corrupted}} \right)^T \mathbf{u}_i^{\text{clean}} \right|. \tag{4.5}$$

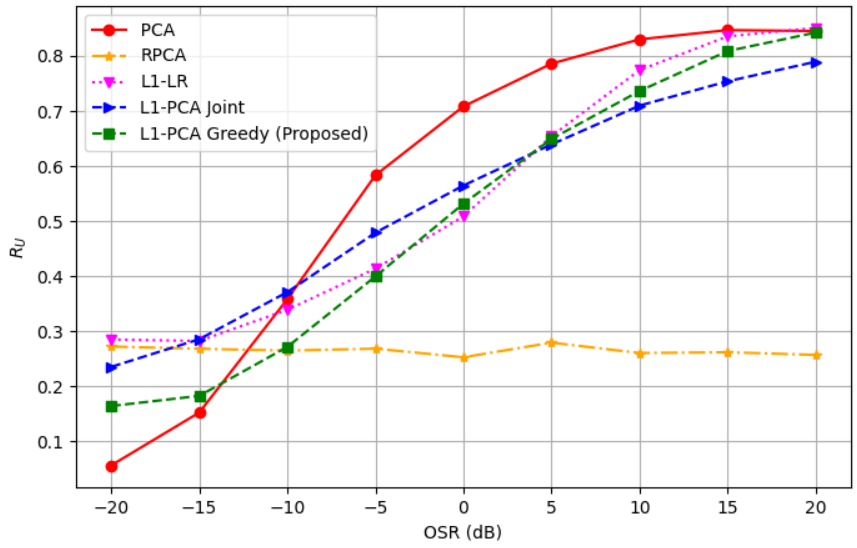

Figure 4.3: The normalized subspace error R_{U} for the different PCA approaches at multiple OSR dB values averaged over 200 experiments. The synthetic dataset has dimension $D = 10$, number of data points $N = 50, K = 4$ SVs are captured, and noise level is SNR = 10 dB. Probability of corruption is $P_o = 0.04$ and outliers are drawn from a 4-dimensional subspace ($K_o = 4$).

From Fig. 4.3, it can be observed that all L1-norm based approaches find a more outlier resistant subspace than the conventional PCA at high OSR, with Greedy L1-PCA over-performing L1-LR at most OSR values and Joint L1-PCA at OSR < 5 dB. RPCA has a practically constant R_{U} at every outlier strength, overtaking all L1-norm approaches at -10 dB. However, RPCA is known to be significantly more computationally expensive than

L1-PCA, so its PCs are not used to find the L1-SVs.

In addition, despite achieving a higher $||\mathbf{U}_{L1}^T\mathbf{X}||_{1,1}$ metric than the Greedy approach, the Joint L1-PCA does not necessarily find a more robust subspace. Yet, the main issue with the Joint approach resides in the robustness of individual PCs, as exhibited next.

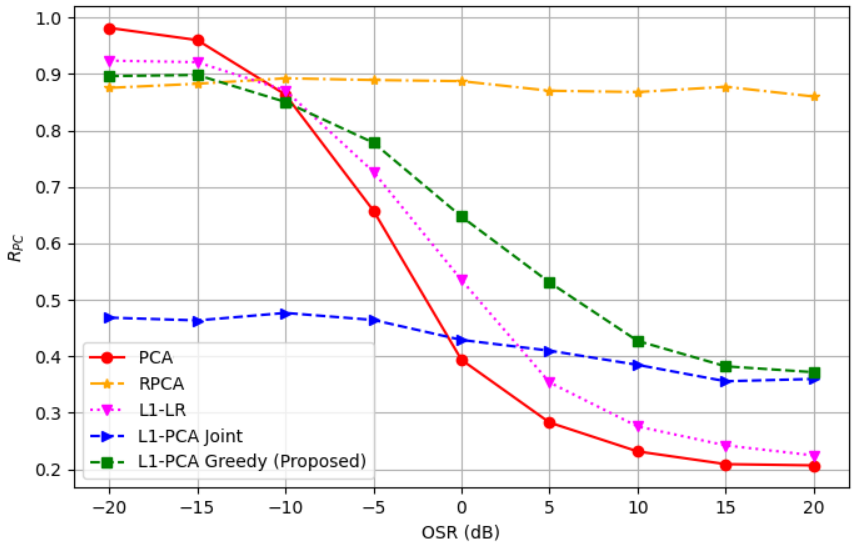

Figure 4.4: The normalized PCs alignment metric R_{PC} for the different PCA approaches at multiple OSR dB values for the same experiment setup as Fig. 4.3

Fig. 4.4 shows the poor alignment between the jointly found L1-PCs of the corrupted data with the clean PCs, consistent with the prediction in section 3.4. On the other hand, the Greedy L1-PCA approach, being capable of finding decisively more robust PCs than L1-LR and PCA at high outlier strength while maintaining reasonably low computational complexity, is consequently chosen to find the L1-SVs.

4.2.4 Singular Values Preservation

All SVs

We then define the normalized SVs estimation error metric to evaluate how well the different algorithms can preserve the SVs of the clean dataset when corrupted with subspace outlier

$$R_{\text{sv}} = \frac{||\mathbf{\Sigma}^{\text{estimated}} - \mathbf{\Sigma}^{\text{clean}}||_{2,2}}{||\mathbf{\Sigma}^{\text{clean}}||_{2,2}}, \tag{4.6}$$

where $\mathbf{\Sigma}^{\text{clean}}$ is calculated by applying the conventional SVD on the clean data matrix $\mathbf{X}^{\text{clean}}$ while $\mathbf{\Sigma}^{\text{estimated}}$ is the estimated SVs from the corrupted dataset $\mathbf{X}^{\text{corrupted}}$ by applying the different SVD algorithms.

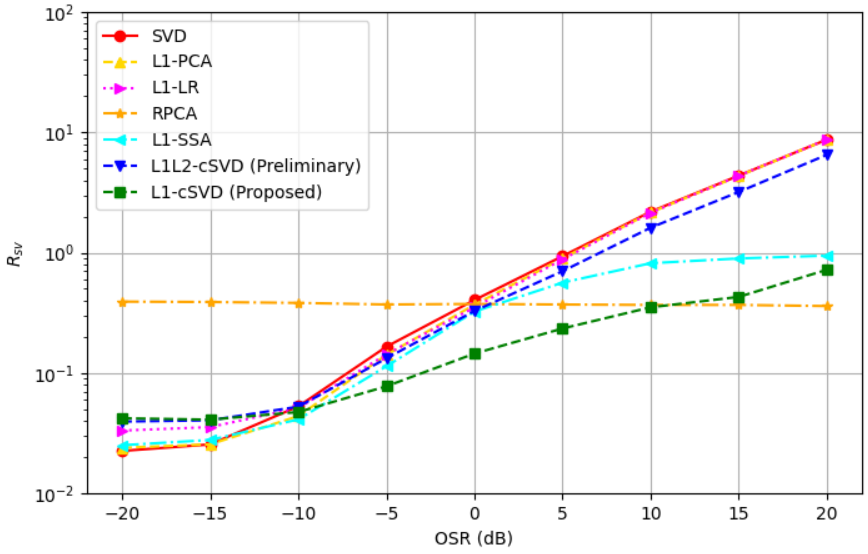

Figure 4.5: The normalized total SVs error R_{sv} for the different SVD approaches at different OSR dB values averaged over 200 experiments. The synthetic dataset has dimension $D = 10$, number of data points $N = 50, K = 4$ SVs are captured, and noise level is SNR = 10 dB. Probability of corruption is $P_o = 0.04$ and outliers are drawn from a 4-dimensional subspace ($K_o = 4$).

From Fig. 4.5, the deviation from clean SVs of SVs found by L1-PCA and L1-LR from corrupted data is very close to that of the conventional SVD. Thus, the robustness of the L1-

PCs or the L1-low-rank approximation is not readily transferable to the SVs. The preliminary L1L2-cSVD does provide marginally better performance in estimating SVs from corrupted data. However, the OSR where its SVs estimation error starts to spiral out of control is still relatively low, hindering any practical applications. Thus, an L2-norm to orthogonalize $\mathbf{U}_{L1}^T\mathbf{X}$ to find the SVs and right singular vectors is not sufficiently robust.

On the other hand, L1-cSVD has much better performance, maintaining a normalized SVs approximation error of $5-25\%$ for OSR between -10dB and 5dB, in which the SVs estimated by SVD start to deviate strongly. In addition, L1-cSVD follows SVD very closely at low OSR, indicating that it is in effect the same as SVD at this regime.

RPCA has the same performance at every OSR due to its ability to separate the sparse component effectively. However, the reconstructed low-rank component is not necessarily robust, since its SVs estimation error is about 40%. This can be attributed to $\mathbf{X}^{\text{corrupted}}$ being noisy, since the formulation in Eq. (2.15) does not take into account noise, which is neither low-rank nor sparse. In addition, the SVs found by L1-SSA are robust at high OSR, but still suffer roughly the same SVs estimation error as SVD before OSR = 0 dB.

Individual SVs

We are also interested in comparing the preservation of the individual SVs, defined by

$$R_{\text{sv,i}} = \frac{(\sigma_i^{\text{estimated}} - \sigma_i^{\text{clean}})^2}{(\sigma_i^{\text{clean}})^2} \qquad (i = 1, 2, ..., K). \tag{4.7}$$

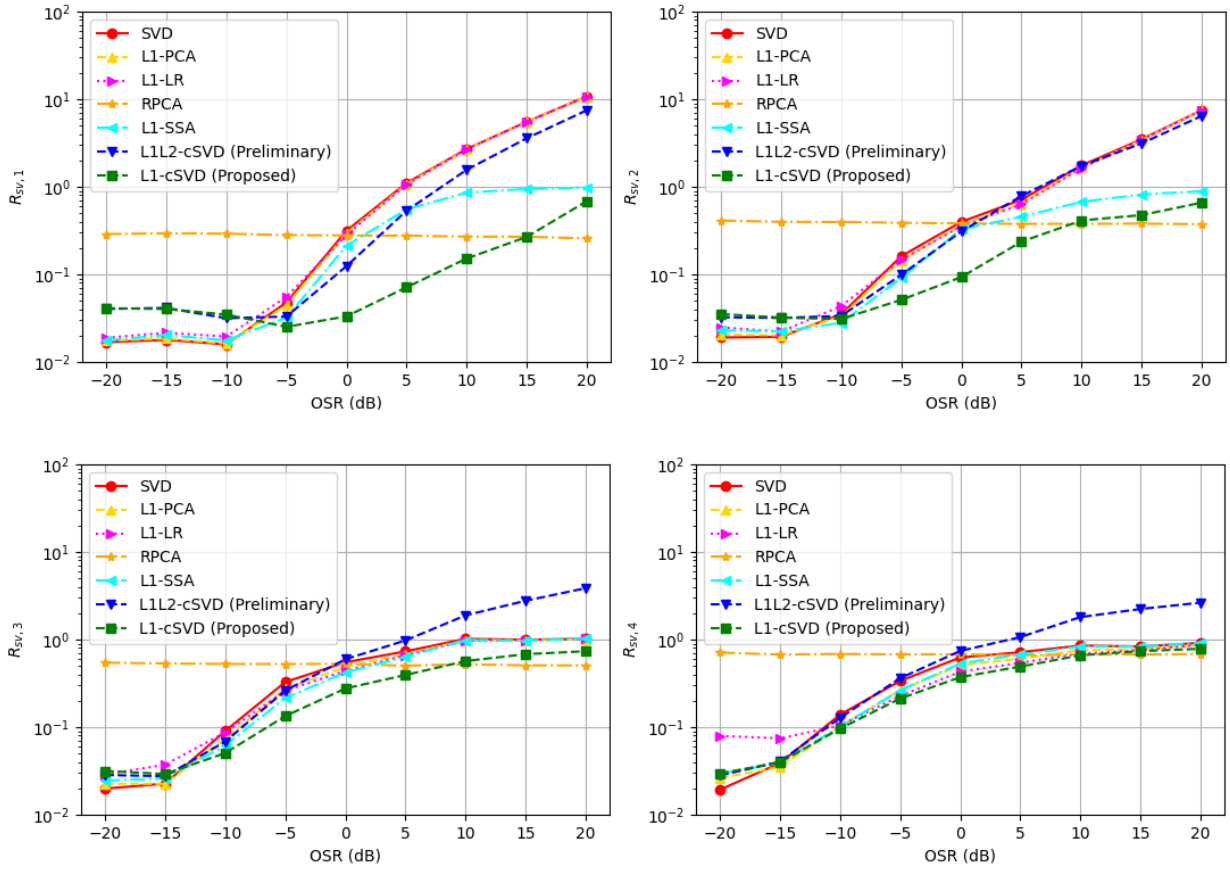

Figure 4.6: The normalized first, second, third and fourth SV errors $R_{\text{sv},1}$, $R_{\text{sv},2}$, $R_{\text{sv},3}$ and $R_{\text{sv},4}$ for the different SVD approaches.

The most significant finding is that the robustness of L1-cSVD is clearly demonstrated in the estimation of the first and also most important SV, which only deviates less than 10% from the clean SV for OSR up to almost 10dB. On the other hand, RPCA incurs a constant 30-70% error on all SVs across different OSR values.

The estimation of subdominant SVs by L1-cSVD is less robust because the task of finding subdominant SVs is more challenging than the first SV, explainable by the fact they they are tied to the subdominant L1-PCs. In Greedy L1-PCA, if the first L1-PC is not recovered perfectly, it is unlikely that the second L1-PC is recovered perfectly either because it has to be orthogonal to the 1$^{\text{st}}$ L1-PC, which is not readily orthogonal to the 2$^{\text{nd}}$ clean PC as

the 1st and 2nd clean PCs are already orthogonal. Nevertheless, L1-cSVD still consistently attains lower subdominant SVs estimation error than the other approaches.

4.2.5 Low-rank Approximation

Another criterion to evaluate the different SVD algorithms is how well $\mathbf{U\Sigma V}^T$ low-rank approximates $\mathbf{X}$. To that end, we will define a low-rank approximation error metric

$$R_{LR} = \frac{(\mathbf{U\Sigma V}^T)^{\text{corrupted}} - \mathbf{X}^{(\text{clean})}}{\mathbf{X}^{(\text{clean})}}. \tag{4.8}$$

It is important to clarify that the main objective of this Thesis is to robustly estimate SVs from corrupted data instead of approximating the whole low-rank data itself. Nevertheless, it is still interesting to observe how the robustness of SVs extends to the low-rank approximated matrix.

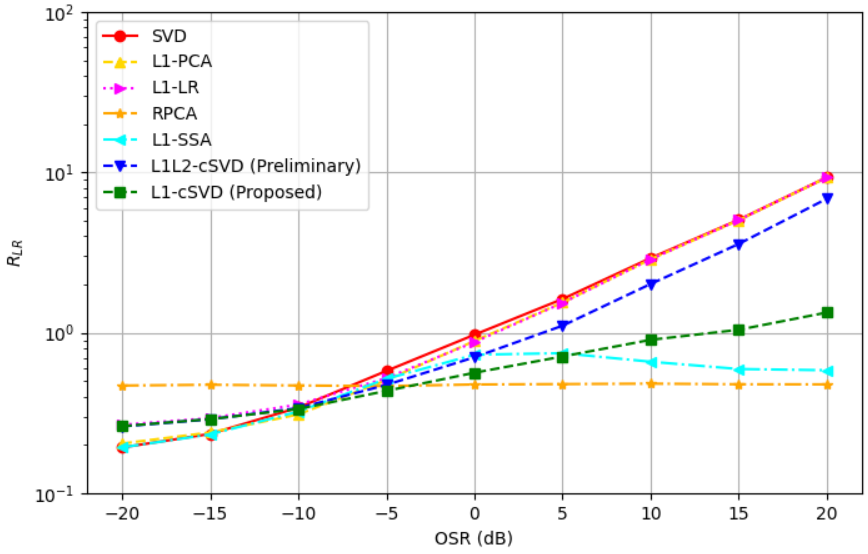

Figure 4.7: The normalized low-rank approximation error metric R_{LR} for the different SVD approaches.

From Fig. 4.7, the L1-PCA and L1-LR approach does not offer any robustness in the

low-rank approximation of the corrupted data compared to conventional SVD. Once again, L1L2-cSVD only shows marginally better performance in this criterion.

On the other hand, the proposed L1-cSVD algorithm indeed provides a much more robust low-rank approximation of the corrupted data matrix than conventional SVD. RPCA also picks out a consistent low-rank component unaffected by the outlier strength. However, for this criterion, the crossover between the performance of RPCA and L1-cSVD happens at a lower OSR of -5 dB, meaning that RPCA arguably does a better job in low-rank approximation than SVs estimation.

4.3 Performance Analysis with Real World Dataset

In this section, the robust L1-cSVD method is applied to the Bayesian classifier for the "Vowel" dataset from Penn Machine Learning Benchmarks (PMLB), which includes hundreds of datasets to evaluate supervised machine learning algorithms [37]. The chosen dataset has $C = 11$ vowels to be classified using $D = 11$ numerical features available. There are $N = 990$ samples to be used as either training or test data.

We will apply a Bayesian Classifier on the 11 features given without using the kernel trick. Of the 90 samples of each of the 11 vowels, 75 is used for training and 15 is reserved for testing. For each training dataset $\mathbf{X}^{(i)}$ of the ith vowel ($i = 1, 2, ..., 11$), we can obtain the median vector $\mathbf{m}_i \in \mathbb{R}^D$ (chosen instead of the conventional mean to soften the effect of outliers) and the SVD of $\mathbf{X}^{(i)} = \mathbf{U}^{(i)}\mathbf{\Sigma}^{(i)}\mathbf{V}^{(i)T}$. It is important to note that $\mathbf{U}^{(i)}$ and $\mathbf{\Sigma}^{(i)}$ are also the eigenvectors and the square root of the eigenvalues of the covariance matrix of $\mathbf{X}^{(i)}$.

According to the Bayesian Classifier, a given test data point $\mathbf{y}$ is classified to the vowel

whose distribution it has the smallest Mahalanobis distance d_i from, where [14]

$$d_i = \sqrt{\sum_{j=1}^{D} \left(\frac{\mathbf{u}_j^{(i)T}(\mathbf{y} - \mathbf{m}^{(i)})}{\sigma_j^{(i)}/\sqrt{N}} \right)^2}. \tag{4.9}$$

The robustness of L1-cSVD is then evaluated by corrupting 3 out of 75 samples from each vowel of the clean dataset $\mathbf{X}$ with outliers coming from a Gaussian distribution $\mathcal{N}(0, \sigma^2)^D$. We choose the outlier strength σ^2 to be 25 times the average energy of entries in $\mathbf{X}$, which means the outlier has the same energy as the clean data since the OSR in this case can be calculated to be 0 dB.

The vowels are then classified based on the parameters trained by the corrupted training data using the left singular vectors $\mathbf{U}^{(i)}$ and SVs $\mathbf{\Sigma}^{(i)}$ from either SVD or L1-cSVD and compare the correct prediction ratios. From Fig. (4.8), it can be observed that the SVs of corrupted data are much better estimated by L1-cSVD compared to the traditional SVD. Importantly, the first SV is reconstructed almost exactly by L1-cSVD. On the other hand, RPCA tends to underestimate the SVs, which can be attributed to the overpromotion of the sparsity of SVs. Because of its high computational cost, RPCA is only used to observe the SVs instead of training the classifier.

Thus, we proceed to train the corrupted dataset with L1-cSVD and compare its performance SVD. Fig. (4.9) shows the decisively higher correct prediction ratio that L1-cSVD can achieve compared to conventional SVD, demonstrating its robustness against gross and sparse outliers.

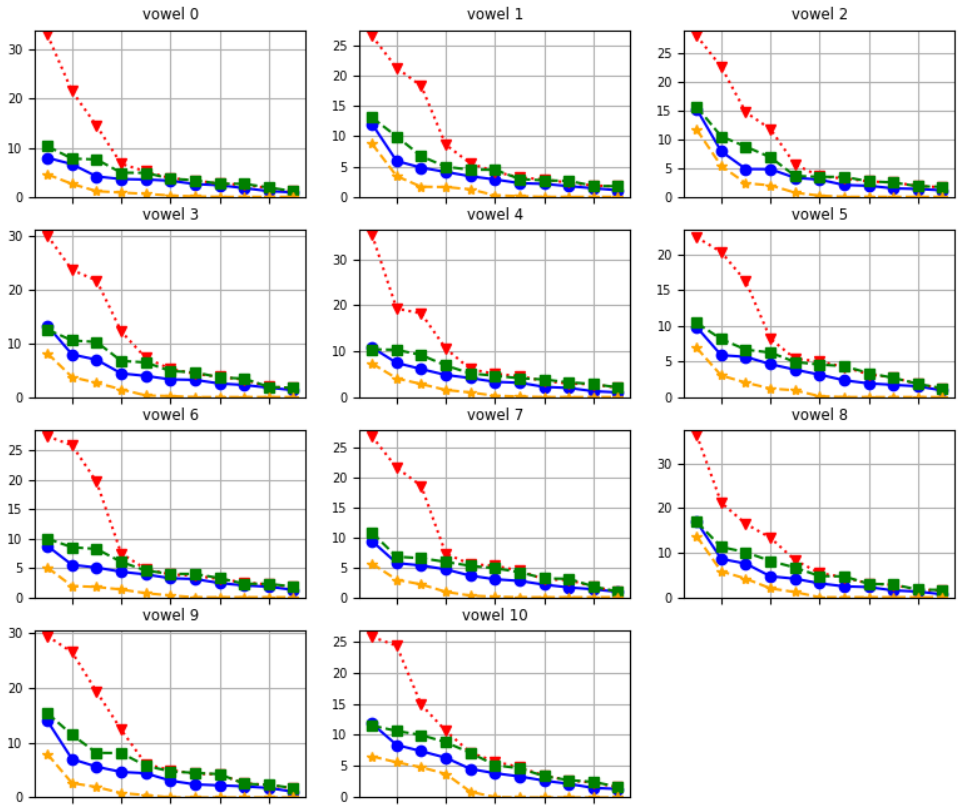

Figure 4.8: The SVs of each vowel training dataset for clean data using SVD (blue, circles), compared to the SVs estimated from corrupted data using SVD (red, triangles), RPCA (orange, stars) and L1-cSVD (green, squares).

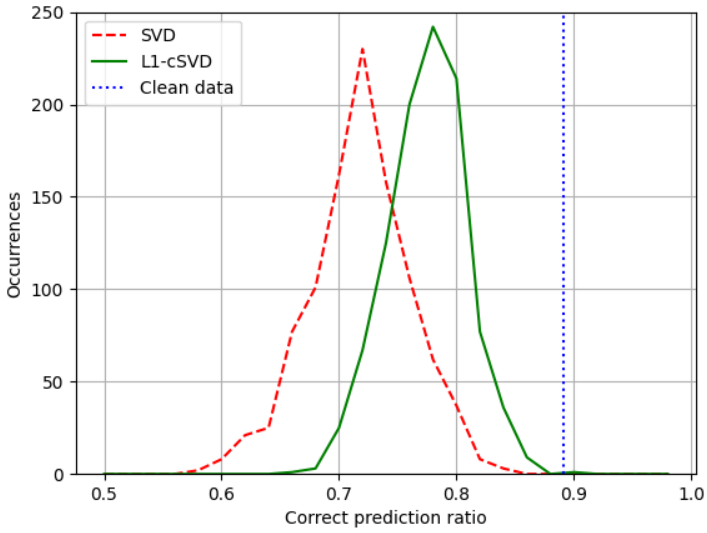

Figure 4.9: The histogram of the correct prediction ratio when the corrupted data is trained with the traditional SVD (green) and L1-cSVD (red, dashed) for 1000 experiments with different corruption realizations. The correct prediction ratio using SVD on clean data is marked by the blue dotted line.

4.4 Direction-of-arrival estimation

In this experiment, we choose a linear array with $M = 8$ sensors uniformly spaced by $\lambda/2$ taking $T = 200$ snapshots of 3 incoming signals with directions of arrival (DOAs) $-45°, 0°$ and $60°$. The received signal at the sensor array can be written to be [1]

$$\mathbf{Y} = \mathbf{AS} + \mathbf{N}, \tag{4.10}$$

where $\mathbf{Y} \in \mathbb{C}^{M \times T}$ is a complex matrix whose rows are the received signal at each sensor and whose columns represent the different time snapshots, $\mathbf{A} \in \mathbb{C}^{M \times N_\theta}$ is the array manifold matrix, which describes the gain information from each of the N_θ DOAs to each of the M sensors. Here, we choose a grid of DOAs from $-90°$ to $90°$ with $1°$ spacing, so $N_\theta = 180$. The (m, k) element of the array manifold matrix can be written to be

$$a_{m,k} = \exp\left[-jm\pi \sin(\theta_k)\right]. \tag{4.11}$$

The noise is represented by $\mathbf{N}$ with SNR = 10 dB. The task of DOA estimation is to reconstruct the matrix $\mathbf{S} \in \mathbb{C}^{N_\theta \times T}$, which includes the amplitude of the incoming signals from N_θ DOAs at T time snapshots. To better distinguish between spatially close sources, Malioutov et. al [1] proposed a method that enforces sparsity within every column of $\mathbf{S}$, since signal sources can be considered sparse in space but not in time, using the L1-norm in the columns and L2-norm in the rows of $\mathbf{S}$.

$$\underset{\mathbf{S} \in \mathbb{C}^{N_\theta \times T}}{\text{minimize}} ||\mathbf{Y} - \mathbf{AS}||_{2,2} + \lambda ||\mathbf{S}||_{1,2}, \tag{4.12}$$

where λ is a regularization parameter. However, solving for $\mathbf{S}$ can be expensive since the

number of time snapshots can be very large. Thus, a dimensionality reduction on the received signal matrix $\mathbf{Y}$ using conventional SVD has been proposed [1], i.e., $\mathbf{Y} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$. Then, $\mathbf{Y}$ in Eq. (4.12) is replaced with $\mathbf{Y}^{\mathrm{SV}} = \mathbf{U}_K\mathbf{\Sigma}_K$, where $\mathbf{U}_K$ includes the first K left singular vectors, $\mathbf{\Sigma}_K$ includes the first K SVs, and $K = 3$ is the expected number of sources. Eq. (4.12) is rewritten to be

$$\underset{\mathbf{S}^{\mathrm{SV}} \in \mathbb{C}^{N_\theta \times K}}{\text{minimize}} ||\mathbf{Y}^{\mathrm{SV}} - A\mathbf{S}^{\mathrm{SV}}||_{2,2} + \lambda||\mathbf{S}^{\mathrm{SV}}||_{1,2}, \tag{4.13}$$

which is a convex problem solvable by standard convex optimization methods [25]. This method is called ℓ1-SVD [1], since it uses an L1-norm to enforce spatial sparsity and SVD for dimensionality reduction. It is not to be confused with L1-cSVD in this work, which formulates a compact SVD scheme using the L1-norm.

Unfortunately, using conventional SVD for dimensionality reduction means that $\mathbf{Y}^{\mathrm{SV}}$ can be sensitive to outliers. To demonstrate, we corrupt $\mathbf{Y}$ with jammer signals coming from DOAs $-30°, 30°$ and $50°$. Each jammer corrupts 10 time snapshots at random with power of 20 times the signal of interest power, so that the OSR as defined in earlier sections is 0 dB. We first reconstruct $\mathbf{S}^{\mathrm{SV}}$ using the corrupted received signal with the ℓ1-SVD method in [1] to obtain a baseline result.

We then propose to amend the original ℓ1-SVD method by using our L1-cSVD for dimensionality reduction so that $\mathbf{Y}^{\mathrm{SV}}$ is less affected by the jammers. Because our L1-cSVD is developed for real data, we define a real $2M \times T$ matrix $\tilde{\mathbf{Y}} = [\mathrm{Re}(\mathbf{Y}); \mathrm{Im}(\mathbf{Y})]$ concatenating the real and imaginary components of $\mathbf{Y}$, on which L1-cSVD is applied to obtain the dimensionality reduced $\tilde{\mathbf{Y}}^{\mathrm{SV}} \in \mathbb{R}^{2M \times K}$. We similarly define the real array manifold matrix

$\tilde{\mathbf{A}} = [\mathrm{Re}(\mathbf{A}); \mathrm{Im}(\mathbf{A})] \in \mathbb{R}^{2M \times N_\theta}$. Finally, we solve

$$\underset{\mathbf{S}^{\mathrm{SV}} \in \mathbb{C}^{N_\theta \times K}}{\text{minimize}} ||\tilde{\mathbf{Y}}^{\mathrm{SV}} - \tilde{\mathbf{A}}\mathbf{S}^{\mathrm{SV}}||_{2,2} + \lambda ||\mathbf{S}^{\mathrm{SV}}||_{1,2} \tag{4.14}$$

to obtain the robust spatial spectrum.

In Fig. 4.10 (top), we plot the spatial spectrum for the clean signal, showing 3 peaks at the expected DOAs. Then, we plot the spatial spectrum when jammers are on using the original ℓ1-SVD methods with conventional SVD (middle), which is clearly affected by the jammers evident by the extra peaks at $-30°$ and $50°$. On the other hand, by using L1-cSVD for dimensionality reduction before applying ℓ1-SVD for DOA estimation (bottom), the jammers' peaks in the spectrum are efficiently suppressed. The reconstructed power of the 3 signals of interest (height of the peaks) is also well-preserved by using L1-cSVD for preprocessing.

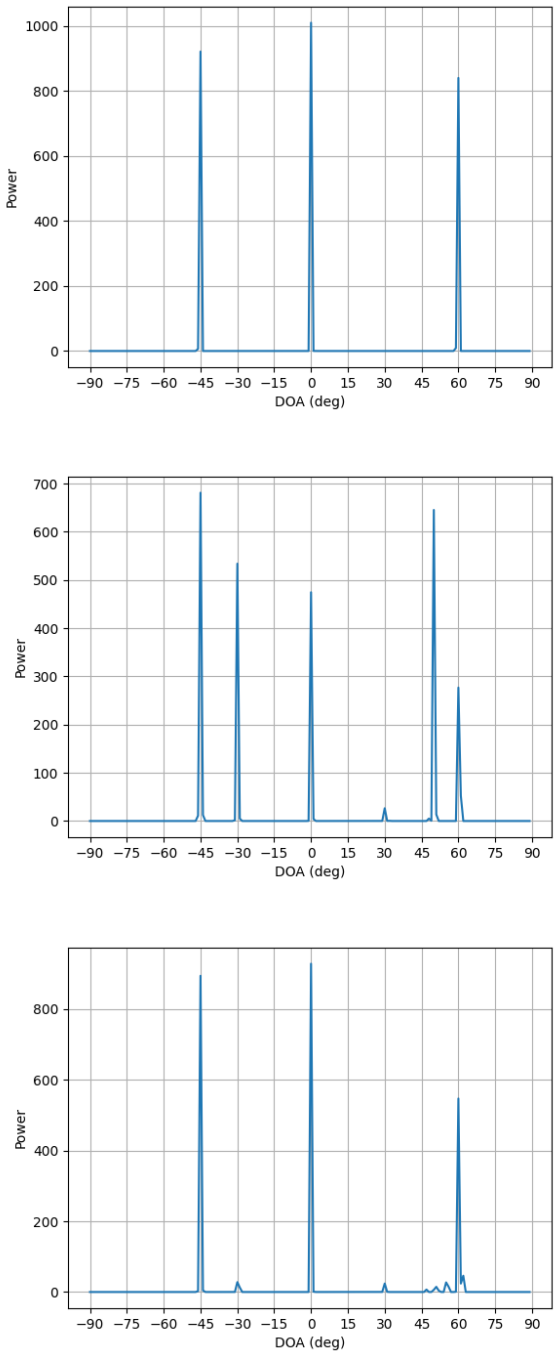

Figure 4.10: Spatial spectra produced by the $\ell 1$-SVD method [1] for uncorrelated sources at DOAs $-45°, 0°$ and $60°$ with (top) no jammers, (middle) jammers at DOAs $-30°, 30°$ and $50°$, using conventional SVD for dimensionality reduction, and (bottom) the same jammers, using L1-cSVD for dimensionality reduction.

Chapter 5

Conclusion

We present a novel L1-norm based SVD algorithm which, for the first time, extends the robustness against outliers of the well-studied L1-PCA problem to finding robust SVs. We prove that such an extension is not obvious and propose a problem formulation to achieve more robust SVs estimation. We experiment with orthogonalizing $\mathbf{U}_{L1}^T\mathbf{X}$ by an L2-norm to find the SVs corresponding to the L1-PCs in a preliminary algorithm called L1L2-cSVD, which is later found to not provide enough robustness in SVs estimation.

Thus, we came up the L1-cSVD algorithm, which utilizes the more robust L1-norm to find the closest orthogonal matrix $\mathbf{\Sigma}_{L1}\mathbf{V}_{L1}^T$ to $\mathbf{U}_{L1}^T\mathbf{X}$. We propose an iterative algorithm that finds $\mathbf{\Sigma}_{L1}$ and $\mathbf{V}_{L1}$ alternatingly with closed-form solutions. The proposed algorithm has an additional complexity of $\mathcal{O}(N^3K^2)$ on top of L1-PCA. We also provide an in depth discussion on the pros and cons of different L1-PCs estimation schemes for more consistent SVs estimation and conclude to use the Greedy approach.

Our algorithms are tested on a synthetic dataset with corruption coming from subspace outliers and noise along with a real world data experiment where the PCs and SVs estimated are applied to a Bayesian Classifier with a dataset corrupted by outliers. Both experiments

demonstrate the robustness of the L1-cSVD approach in SVs estimation compared to the conventional SVD, the direct extension from L1-PCA and L1-low-rank approximation, and the state-of-the-art RPCA. We expect our algorithm to find use in applications where the SVs play a strong role.

Bibliography

[1] D. Malioutov, M. Cetin, and A. S. Willsky, "A sparse signal reconstruction perspective for source localization with sensor arrays," *IEEE transactions on signal processing*, vol. 53, no. 8, pp. 3010–3022, 2005.

[2] G. Lebrun, J. Gao, and M. Faulkner, "Mimo transmission over a time-varying channel using svd," *IEEE Transactions on wireless Communications*, vol. 4, no. 2, pp. 757–764, 2005.

[3] R. F. Fischer, C. Windpassinger, A. Lampe, and J. B. Huber, "Space-time transmission using tomlinson-harashima precoding," *ITG FACHBERICHT*, pp. 139–148, 2002.

[4] K.-L. Chung, W.-N. Yang, Y.-H. Huang, S.-T. Wu, and Y.-C. Hsu, "On svd-based watermarking algorithm," *Applied Mathematics and Computation*, vol. 188, no. 1, pp. 54–57, 2007.

[5] C.-C. Chang, P. Tsai, and C.-C. Lin, "Svd-based digital image watermarking scheme," *Pattern Recognition Letters*, vol. 26, no. 10, pp. 1577–1586, 2005.

[6] N. Tsagkarakis, P. P. Markopoulos, and D. A. Pados, "Direction finding by complex l 1-principal-component analysis," in *2015 IEEE 16th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 475–479, IEEE, 2015.

[7] P. Markopoulos, N. Tsagkarakis, D. Pados, and G. Karystinos, "Direction finding with l1-norm subspaces," in *Compressive Sensing III*, vol. 9109, p. 91090J, International Society for Optics and Photonics, 2014.

[8] J. Xue, J. Li, and Y. Gong, "Restructuring of deep neural network acoustic models with singular value decomposition.," in *Interspeech*, pp. 2365–2369, 2013.

[9] O. Iqbal, S. A. Fattah, and S. Zahin, "Hand movement recognition based on singular value decomposition of surface emg signal," in *2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*, pp. 837–842, IEEE, 2017.

[10] C. D. Meyer, *Matrix analysis and applied linear algebra*, vol. 71. Siam, 2000.

[11] G. H. Golub and C. F. Van Loan, *Matrix computations*. JHU press, 2013.

[12] R. Vidal, Y. Ma, and S. S. Sastry, "Principal component analysis," in *Generalized principal component analysis*, pp. 25–62, Springer, 2016.

[13] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*, vol. 4. Springer, 2006.

[14] R. O. Duda, P. E. Hart, and D. G. Stork, "Pattern classification 2nd edition," *New York, USA: John Wiley&Sons*, p. 35, 2001.

[15] C. Eckart and G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, no. 3, pp. 211–218, 1936.

[16] N. Srebro and T. Jaakkola, "Weighted low-rank approximations," in *Proceedings of the 20th international conference on machine learning (ICML-03)*, pp. 720–727, 2003.

[17] M. Irani and P. Anandan, "Factorization with uncertainty," in *European Conference on Computer Vision*, pp. 539–553, Springer, 2000.

[18] F. De La Torre and M. J. Black, "A framework for robust subspace learning," *International Journal of Computer Vision*, vol. 54, no. 1, pp. 117–142, 2003.

[19] N. Kwak, "Principal component analysis based on l1-norm maximization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, no. 9, pp. 1672–1680, 2008.

[20] F. Nie, H. Huang, C. Ding, D. Luo, and H. Wang, "Robust principal component analysis with non-greedy ℓ1-norm maximization," in *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[21] P. P. Markopoulos, G. N. Karystinos, and D. A. Pados, "Optimal algorithms for $l\_\{1\}$-subspace signal processing," *IEEE Transactions on Signal Processing*, vol. 62, no. 19, pp. 5046–5058, 2014.

[22] P. P. Markopoulos, S. Kundu, S. Chamadia, and D. A. Pados, "Efficient l1-norm principal-component analysis via bit flipping," *IEEE Transactions on Signal Processing*, vol. 65, no. 16, pp. 4252–4264, 2017.

[23] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?," *Journal of the ACM (JACM)*, vol. 58, no. 3, pp. 1–37, 2011.

[24] Q. Ke and T. Kanade, "Robust l/sub 1/norm factorization in the presence of outliers and missing data by alternative convex programming," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, pp. 739–746, IEEE, 2005.

[25] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[26] N. Tsagkarakis, P. P. Markopoulos, and D. A. Pados, "On the l1-norm approximation of a matrix by another of lower rank," in *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 768–773, IEEE, 2016.

[27] Z. Lin, A. Ganesh, J. Wright, L. Wu, M. Chen, and Y. Ma, "Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix," *Coordinated Science Laboratory Report no. UILU-ENG-09-2214, DC-246*, 2009.

[28] V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky, "Sparse and low-rank matrix decompositions," *IFAC Proceedings Volumes*, vol. 42, no. 10, pp. 1493–1498, 2009.

[29] H. Xu, C. Caramanis, and S. Sanghavi, "Robust pca via outlier pursuit," *Advances in neural information processing systems*, vol. 23, 2010.

[30] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma, "Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization," *Advances in neural information processing systems*, vol. 22, 2009.

[31] M. Kalantari, M. Yarmohammadi, and H. Hassani, "Singular spectrum analysis based on l 1-norm," *Fluctuation and Noise Letters*, vol. 15, no. 01, p. 1650009, 2016.

[32] L. Liu, D. M. Hawkins, S. Ghosh, and S. S. Young, "Robust singular value decomposition analysis of microarray data," *Proceedings of the National Academy of Sciences*, vol. 100, no. 23, pp. 13167–13172, 2003.

[33] L. Zhang, H. Shen, and J. Z. Huang, "Robust regularized singular value decomposition with application to mortality data," *The Annals of Applied Statistics*, pp. 1540–1561, 2013.

[34] K. G. Quach, K. Luu, C. N. Duong, and T. D. Bui, "Robust l p-norm singular value decomposition," in *NIPS Workshop on Non-convex Optimization for Machine Learning: Theory and Practice*, 2015.

[35] A. Gang and W. U. Bajwa, "A linearly convergent algorithm for distributed principal component analysis," *Signal Processing*, vol. 193, p. 108408, 2022.

[36] N. T. Trendafilov, "On the $\ell 1$ procrustes problem," *Future Generation Computer Systems*, vol. 19, no. 7, pp. 1177–1186, 2003.

[37] R. S. Olson, W. La Cava, P. Orzechowski, R. J. Urbanowicz, and J. H. Moore, "Pmlb: a large benchmark suite for machine learning evaluation and comparison," *BioData mining*, vol. 10, no. 1, pp. 1–13, 2017.