Robust Singular Value Decomposition in Low-Rank Settings

Anthony Wong
Department of Computer Science
Brown University
Providence, RI 02912
anthony_g_wong@brown.edu

Preetish Juneja
Department of Computer Science
Brown University
Providence, RI 02912
preetish_juneja@brown.edu

Abstract

This paper explores the singular value decomposition (SVD) for low-rank approximations across various applications. We propose an extension to the robust covariance framework for interpretable PCA with theoretical bounds. We also extend the framework to introduce a projection-based robust regression algorithm, PrSVD, which performs well against adversarial noise scenarios in a low-rank setting. Finally, we also develop a deep learning framework for robust image filtering that reconstructs low-rank approximations from noisy images using our variant of the weighted-Huber loss. Our work for robust SVD in low-rank settings across domains provides potential future directions for research in high-dimensional low-rank environments.

1 Introduction

1.1 Singular Value Decomposition

Singular Value Decomposition (SVD) is a remarkable tool from linear algebra that has been widely used in applications such as image compression, recommender systems, and signal processing to produce low-rank approximations. Utilizing computational algorithms, SVD has been embedded in various machine learning and data pipelines across industries. However, when dealing with real-world data, it is hardly ever perfect. Moreover, it is prone to different kinds of outliers which can sometimes be rather difficult to detect, which necessitates the development of a relatively robust framework to compute these low-rank approximations. We note that despite its importance, achieving robust SVD is challenging due to the tradeoff between computational efficiency and statistical guarantees.

1.2 Our Direction and Related Works

In this work, we focus on the use of SVD to generate low-rank approximations in principal components analysis (PCA), ordinary least squares (OLS) linear regression, and image filtering. Although there has been a strong focus on devising different methods to perform robust PCA in literature following the work by Candès et al. (2011) (such as works by Brahma et al. (2017) and Rahmani and Atia (2017)), we extend the robust covariance framework to bound the principal angle to provide further insights into the technique. With respect to linear regression, we demonstrate the effectiveness of extending the robust covariance framework when estimating coefficients in a low-rank setting, rather than developing a different robust regression framework as seen in works such as Filzmoser and Nordhausen (2020) and Jambulapati et al. (2021). Lastly, for image filtering, we outline a method that employs a neural network on our proposed robust loss function to reconstruct a low-rank version of an image corrupted with arbitrary noise.

Various authors have proposed methods to find low-rank approximations by devising and solving optimization problems. For instance, Candès et al. (2011) solved an optimization problem by decomposing a data matrix into a low-rank approximation and a sparse outlier matrix. There have been other variants of this approach that have proven successful in achieving the desired low-rank approximation by Wright et al. (2013), Zhou et al. (2010) and Xu et al. (2010). Another technique that has been quite popular for robust SVD/PCA introduces the problem of finding an SVD as a minimization problem with element-wise loss. Works such as Gabriel and Zamir (1979), Liu et al. (2003), Ke and Kanade (2005) and Zhang et al. (2013) have attempted to make the problem robust by proposing loss functions such as L_1-loss or Huber's loss instead of the element-wise loss. We note that our approach for image filtering follows the same spirit in using a robust loss function within the novel framework we construct in this paper.

In Section 2, we define the mathematical notation we employ in this paper as well as clearly define the problem of robust SVD to construct a low-rank approximation. In Section 3, we delve deeper into our proposed methods for robust PCA, linear regression and image filtering as well as provide a theoretical and/or empirical analysis to

outline the technique's robustness in the considered low-rank setting. Section 4 concludes our findings and presents potential future directions to extend our work.

2 Problem Definition

In this section we formally define the problems we reference in the paper. We first define robust covariance estimation and robust SVD, which are well-studied problems in literature. We extend these problems to other robust applications, including PCA, linear regression, and image filtering, particularly in a low-rank setting.

2.1 Robust Covariance Estimation

We are given points $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_m \in \mathbb{R}^n$ sampled according to the following rule. With probability $1 - \eta$, each $\boldsymbol{x}_i$ is sampled iid from a distribution $\mathcal{D}$ with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$, and with probability η picked by an adversary. We want to estimate the mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ given the samples, especially in distributions with bounded moments, to use $\hat{\boldsymbol{\Sigma}}$ for PCA and regression.

2.2 Robust SVD

The SVD of a matrix $X \in \mathbb{R}^{n \times m}$ is given by

$$X_{n \times m} = U_{n \times p} \Sigma_{p \times p} V^T_{p \times n}$$

where $p \leq \min(n, m)$. With $\boldsymbol{u}_i, \boldsymbol{v}_i, \sigma_i$ representing the i^{th} left singular vector, right singular vector and singular value respectively, we can write the SVD as

$$X = \sum_{i=1}^{p} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T$$

with the summation of first k $(k < p)$ terms representing the k-rank approximation of X. We intend to find $\hat{U}, \hat{\Sigma}, \hat{V}$ using a perturbed matrix $X_{perturbed}$ where the rows/columns of datapoints have been sampled as in the setting of robust covariance estimation.

2.3 Robust PCA

In robust PCA, we are given a perturbed matrix, $X_{perturbed}$, and want to estimate the true principal component directions and principal component values. While often performed with SVD (since the right singular values are the principal component directions and the corresponding singular values are the magnitudes), we use the eigenvectors of the robust covariance estimate for the principal component directions. Mathematically, we have the eigendecomposition, $\hat{\Sigma} = V \Lambda V^T$, which yields the same V as the SVD decomposition. Thus, a robust top k-component PCA analysis yields V_k and $\sigma_1, \ldots, \sigma_k$, where σ_i is the i^{th} singular value of the uncontaminated matrix X.

2.4 Robust Linear Regression

Given a perturbed matrix $X_{perturbed}$, we want to estimate the coefficients $\hat{\beta}$ to closely resemble the true coefficients β. Using the pseudo-inverse, the OLS regression problem can be performed by: $\beta = X^+ y$, with the psuedo-inverse expressed as $X^+ = V \Sigma^+ U^T$, thus giving us $\beta = V \Sigma^+ U^T y$.

2.5 Robust Image Filtering

Given a perturbed image $X_{perturbed} = X + N$ where X represents the original image and N represents added adversarial noise, our goal is to recover a low-rank approximation of X from the SVD of $X_{perturbed}$.

3 Methods and Robustness Analysis

3.1 Principal Components Analysis

One of the key applications of robust SVD is principal component analysis. Thus, when evaluating the effectiveness of a robust SVD algorithm, its performance on PCA applications is an important factor. We have defined the robust PCA problem and how robust covariance can be applied in 2.

In this section, we present a new mathematical framework to quantify how effective robust PCA might be given how effective the robust covariance is. Mathematically, we extend the bounds from $||\Sigma - \hat{\Sigma}||_F$ to bounds on the difference between the eigenspaces of Σ and $\hat{\Sigma}$. Then, we compare the effectiveness of various modern robust covariance estimation in the robust PCA setting using this new framework.

So how can we quantify the error on the top-k eigenvalue subspaces for two covariance matrices, particularly if we care about low rank applications? The natural answer is using the principal angles of the subspace. For the case of PCA, the principal angle is very interpetable. It measures the difference in orientation of two subspaces in a higher-dimensional latent space. Smaller angles indicate subspaces that are closer oriented along their dimensions. These angles are scale-invariant measures. In the case of PCA, the angle is only determined by the eigenvector bases of the subspaces, and not affected by the corresponding magnitudes (the eigenvalues). This makes an angle extremely interpretable, because it decouples the effect of the eigenvectors and the eigenvalues.

3.1.1 Robust PCA evaluation with Davis-Kahan $\sin\Theta$ Theorem

Consider the robust setup where $\mathbf{E}$ is some perturbation on a true covariance matrix Σ. Let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ be the eigenvalues of Σ. Let U_k denote the subspace spanned by the top k eigenvectors of Σ, and let U_k' denote the top k eigenspace of $\Sigma + E$. Let Θ be the principal angle between U_k and U_k' Then the Davis-Kahan theorem gives an upper bound on the sin(Θ).

$$\|\sin\Theta\|_F \leq \frac{\|\mathbf{E}\|_F}{\delta} \tag{1}$$

δ is the k-th eigenvalue gap of Σ, defined by $\delta = \lambda_k - \lambda_k + 1$

For reasons that we outline below, the Davis-Kahan Theorem is an extremely natural choice for examining how perturbation affects PCA. First, notice that covariance matrices are symmetric and their eigenvectors form an orthonormal bases. Thus, the Davis-Kahan theorem directly applies for any valid covariance matrices.

This leads us into presenting our framework for evaluating the effectiveness of robust PCA for covariance estimation methods. In robust settings, we are given a perturbed matrix $X_{perturbed}$. A robust covariance estimation algorithm will return an estimated covariance $\hat{\Sigma}$. Let $\mathbf{E} = \mathbf{\Sigma} - \hat{\mathbf{\Sigma}}$ be the perturbation between the true covariance matrix $\mathbf{\Sigma}$ and the estimated covariance matrix $\hat{\mathbf{\Sigma}}$. We can now apply the Davis-Kahan Theorem 1 to bound the principal angle using the error bound

$$||E||_F = ||\Sigma - \hat{\Sigma}||_F$$

. Note that in the robust setting we cannot directly compare principal components (XV_k) because we don't have access to the unperturbed matrix X. We must therefore settle for comparing the principal component directions V_k.

3.1.2 Comparison to the Relative Error Bound

The traditional method for comparing the eigenstructure of Σ and $\hat{\Sigma}$ is to use the relative error bound given by:

$$\|\Sigma^{-1/2}\hat{\Sigma}\Sigma^{-1/2} - I\|_F,$$

This norm combines information about both eigenvalues and eigenspaces. To see this, consider the eigendecomposition of Σ and $\hat{\Sigma}$:

$$\Sigma = U\Lambda U^\top, \quad \hat{\Sigma} = V\hat{\Lambda}V^\top,$$

Then, plugging into the relative error bound, we get

$$\Sigma^{-1/2}\hat{\Sigma}\Sigma^{-1/2} = (\Sigma^{-1/2}V)\hat{\Lambda}(\Sigma^{-1/2}V)^\top.$$

This resulting matrix can be interpreted as the difference in:

1. Eigenvalues: measured by the diagonal entries of $\hat{\Lambda}$ as compared to Λ
2. Eigenspaces: $\Sigma^{-1/2}V$) measures the alignment of V with the eigenvectors of Σ

Comparing the two metrics, we see that the relative error bound has the advantage of comparing eigenvalues in the norm. However, in the robust PCA setting, especially for low rank applications, we argue that the Davis-Kahan bound is much more practical. By combining the evaluation of eigenvalues and eigenvectors, the relative error becomes less interpetable than the Davis-Kahan bound. The Davis-Kahan bound isolates the eigenspaces after

selecting an optimal value of k. On the other hand, the relative error bound evaluates all eigenvalues and eigenvectors, without including consideration for k.

To see the advantage of Davis-Kahan, consider the useful interpretation of the Spectral Gap δ. $\delta = \lambda_k - \lambda_{k+1}$ measures the separation between the U_k of interest the remaining eigenvectors using their corresponding eigenvalues. Thus δ is directly a measure of robustness, since a high δ implies that U_k remains stable under small perturbations. To see why, consider the dynamics of perturbation when choosing a k that yields a small δ. In this case, the k-th eigenvalue is close to the (k+1)-th eigenvalue; small perturbations can cause significant rotations or mixing between U_k and the remaining eigenspace. This matches the intuition that eigenvectors of closely spaced eigenvalues are more sensitive to perturbations. Thus, despite the fact that the principal angle does not account for the effect of the eigenvalues λ_k, the Davis-Theorem actually does account for this in the form of δ. This interpretation also provides a method for selecting k in robust PCA settings that maximizes δ, thus minimizing the principal angle of the resulting eigenspaces in 1.

Thus in certain settings, especially low-rank (like SVD matrix approximation), Davis-Kahan is a clear choice. Consider the PCA dimensionality reduction problem and the SVD low rank approximation problem. The quality of the dimensionality reduction depends only on how well the eigenspaces align rather than on magnitude of the eigenvalues. Davis-Kahan explicitly measures this alignment, whereas relative error bounds would conflate the eigenvalue scaling with eigenspace deviation. The Davis-Kahan measurements also provide a method for choosing the value of k, often a critical hyperparameter in these problems.

Another example where Davis-Kahan might be better is spectral clustering, where the eigenvectors of the graph Laplacian matrix are used to partition a graph into several clusters. Spectral clustering is known to depend heavily on the eigenspace alignment rather than on the eigenvalues, especially when the clusters are defined by the top k eigenvectors. In this case, Davis-Kahan's bound would provide a direct measure of the clustering accuracy after perturbing the matrix.

One advantage of the relative error bound is scale invariance, which can be a desired property in some contexts. To address this, we propose an adjusted version of Davis-Kahan that is also scale invariant. We start by normalizing the so that

$$\Sigma' = \frac{\Sigma}{\|\Sigma\|}, \quad \hat{\Sigma}' = \frac{\hat{\Sigma}}{\|\hat{\Sigma}\|}.$$

Here, $\|\Sigma\|$ would be the spectral norm ($\|\Sigma\|_2$), which equals the largest eigenvalue λ_1 of Σ. This normalization would rescale Σ and $\hat{\Sigma}$ so that $\|\Sigma'\| = 1$. Thus, the relative comparison would become invariant to the original scale of Σ, just as the relative error is. Now we can carry out Davis-Kahan on the normalized matrices, giving us

$$\|\sin\Theta\|_F \leq \frac{\|\Sigma' - \hat{\Sigma}'\|_F}{\delta} \tag{2}$$

We now compare these robust PCA bounds to 2 relatively recent robust covariance estimation papers, extending the provided bounds on $\Sigma - \hat{\Sigma}$. We give an overview of the algorithm ideas, then evaluate robust PCA error using our framework.

We start with Cheng et al. (2019), which proposes an iterative mean estimation algorithm for estimating the covariance of a normal distribution. They consider the robust covariance estimation problem as defined in 2. They define the Kronecker Products $Z_i = X_i \otimes X_i$, which represents the second-order necessary for covariance estimation. Thus, they simplify the robust covariance estimation to a high dimensional robust mean estimation.

Their algorithm starts with a rough estimation, then splits $\mathbb{R}^d$ into 3 disjoint subspaces $S_1, S_2, and S_3$. The covariance of each individual subspace is computed. These are later combined using projections Π_S to construct the final result. Algorithm 3 of Cheng et al. proposes the an iterative algorithm with steps outlined below:

- Run a rough covariance estimation algorithm to produce a rough estimate of the covariance matrix M_0. This estimation is guaranteed to have additive error of O($\sqrt{\epsilon}||\Sigma||_2$).
- Compute the eigendecomposition of M_0. Define the subspace S_1 to be the span of eigenvectors with corresponding eigenvalues greater than the threshold $C_1\sqrt{\epsilon}$.
- Input $S_1^{\perp}$ into the rough covariance estimation algorithm to produce another matrix M_1.
- Similar to before, we use the eigendecomposition of $M_1[S_1^{\perp}]$ to produce the subspace S_2 with eigenvalues greater than a smaller threshold $C_2\epsilon$. Together S_1, S_2, and S_3 form a partition of the entire $\mathbb{R}^d$.

- Use mean estimation on the projection of samples onto these subspaces, then combine the results from M_1, M_2, and M_3 to form the final covariance estimate Σ:

$$\hat{\Sigma} = M_1 + M_2 - M_2[S_2] + \sqrt{\frac{1}{\epsilon}}\big(M_3[S_3, S_3] + M_3[S_3, S_1]\big).$$

As a result, we get an error bound for the additive error (Theorem 3 of Cheng et al.) $||\Sigma - \hat{\Sigma}||_F = O\left(\epsilon \log\left(\frac{1}{\epsilon}\right)\right) \|\Sigma\|_2$ with high probability.

Using this additive error bound, we can apply our framework to get bounds for robust PCA. Let $\mathbf{E} = \Sigma - \hat{\Sigma}$ be the perturbation matrix. Substituting the bound on $\|\mathbf{E}\|_F$ and using $\delta = \lambda_k - \lambda_{k+1}$ in 1, we get the following bound on Θ, the principal angle of the top k eigenspace for Σ and $\hat{\Sigma}$

$$\|\sin\Theta\|_F \leq \frac{O\left(\epsilon \log\left(\frac{1}{\epsilon}\right)\right) \|\Sigma\|_2}{\lambda_k - \lambda_{k+1}},$$

with high probability.

It's also worth noting that Li et. al. managed to improve the algorithm, coming up with the same bounds with slightly faster runtime.

We now extend these bounds using Lai et al (2016). Their robust covariance estimation is older with worse error bounds, but they do directly apply their covariance estimation to robust SVD. The covariance estimation follows the key ideas outlined below:

- Given each sample, consider the second moment $(x - \mu)(x - \mu)^T$ as a vector in $\mathbb{R}^{n^2}$.
- Run a robust mean estimation algorithm as detailed below on these 2nd order vectors to estimate the covariance matrix.
- First, use outlier removal to mitigate adversarial noise. This entails weighting or pruning points based on the distance from a robust center (coordinate-wise median).
- Use principal component analysis to project data onto the span of the top $n/2$ principal components.
- Recursively remove outliers and robustly estimate the mean on this reduced subspace with the top n/2 directions of greatest variance.

The paper directly applies the covariance estimation to robust SVD, producing a rank k matrix $\hat{\Sigma_k}$ such that

$$\|\Sigma - \hat{\Sigma}_k\|_F \leq \|\Sigma - \Sigma_k\|_F + O\left(\sqrt{\eta \log n}\right) \|\Sigma\|_2.$$

where Σ_k is the best rank k approximation of Σ (Theorem 1.7 of Lai et al). Knowing that Σ_k is the best rank-k approximation of Σ, we have $\|\Sigma - \Sigma_k\|_F = \left(\sum_{i=k+1}^{n} \lambda_i^2\right)^{1/2}$.

The perturbation between Σ_k and $\bar{\Sigma}_k$ can be bounded as: $\|\mathbf{E}\|_F = \|\bar{\Sigma}_k - \Sigma_k\|_F \leq \|\bar{\Sigma}_k - \Sigma\|_F + \|\Sigma - \Sigma_k\|_F$.

Thus, applying the Davis Kahan theorem we get that

$$|\sin\Theta\|_F \leq \frac{\|\mathbf{E}\|_F}{\delta} \leq \frac{\left(\sum_{i=k+1}^{n} \lambda_i^2\right)^{1/2} + O\left(\sqrt{\eta \log n}\|\Sigma\|_2\right)}{\delta}$$

3.2 Ordinary Least Squares Regression

We also extend the robust covariance framework to provide an algorithm that performs robust ordinary least squares regression in a low-rank setting. This projection-based algorithm is presented as Algorithm 1 below.

We now address the key features of the algorithm to demonstrate how it works. Firstly, we notice that the symmetric covariance matrix $\hat{\Sigma}$ has the formula $\hat{\Sigma} = \frac{1}{n-1}(X - \hat{\mu})^T(X - \hat{\mu})$ where $\hat{\mu}$ is the estimated mean of each feature vector in the regression. However, we note that if we let $\hat{\mu} = 0$ by construction of feature vectors that instead solve $\hat{\Sigma} = \frac{1}{n-1} X_r^T X_r$ for some X_r. Now, using the eigendecomposition, we are able to find X_r exactly as $X_r = \sqrt{\Lambda_r} Q_r^T$ which is extremely helpful as this X_r is a reduced-sample low-rank approximation of our original perturbed data matrix X, and is expected to be relatively clean as a result of the robust covariance estimation algorithm.

Algorithm 1 PrSVD Regression

Inputs: perturbed data matrix $X \in \mathbb{R}^{n \times m}$ and perturbed outcome vector $Y \in \mathbb{R}^{n \times 1}$
1) Robust Covariance Estimation
$\hat{\Sigma} \leftarrow \texttt{RobustCovariance}(X)$ ▷ Using an algorithm from above
2) Eigendecomposition
Compute Λ, Q such that $Q\Lambda Q^T = (n-1)\hat{\Sigma}$
3) Find demeaned low-rank X_r
Since $X_r^T X_r = Q\Lambda Q^T$, compute $X_r = \sqrt{\Lambda_r} Q_r^T$
4) Project Y **accordingly**
$Y_r \leftarrow P_{X_r} Y$ where we have $P_{X_r} = X_r X^+$
5) Perform OLS of Y_r **on** X_r
Find β that minimizes $||Y_r - X_r\beta||_2^2$ for $Y_r = X_r\beta + \epsilon_r$

An interesting insight comes from the comparison of the original regression $Y = X\beta + \epsilon$ to the prospective regression of $Y_r = X_r\beta + \epsilon_r$ assuming that the coefficients represented by β do not change between the regression specifications, and that we can find a Y_r such that this new regression holds. Instead of using the traditional projector matrix $P_X = XX^+ = X(X^TX)^{-1}X^T$ to project Y onto the subspace spanned by $X\beta$, what we do is utilize a projector of the form $P_{X_r} = X_r X^+ = X_r(X^TX)^{-1}X^T$. Here, a primary challenge is that the projected error $\epsilon_r = P_{X_r}\epsilon$ does not necessarily have to be orthogonal to $X_r\beta$, but we find this to be approximately the case in a low-rank setting when testing empirically.

We use a generated dataset of $n = 10000$ samples with $m = 1000$ features where only 10 features are informative, thereby creating artificially imposing the low-rank. We further utilize the following perturbation schemes for testing to understand how robust the technique is to various kinds of adversarial attacks:

- Add large noise to a small percentage of randomly selected elements.
- Replace a fraction of rows with outlier rows.
- Replace a fraction of columns with outlier columns.
- Set a fraction of elements to extreme values.
- Add a low-rank perturbation to X.

We provide results of our experiments with different perturbations to the generated dataset with OLS, PrSVD, and other robust regression algroithms (namely Huber, RANSAC and Thiel-Sen) in the table below.

	OLS	**Huber**	**RANSAC**	**Thiel-Sen**	**PrSVD**
Large Noise	53.32%	0.41%	4.72%	25.17%	0.60%
Outlier Rows	80.71%	0.12%	0.10%	11.53%	0.22%
Outlier Columns	6.97%	7.01%	7.53%	19.41%	3.96%
Extreme Values	99.98%	DNC	DNC	56.63%	1.17%
Low-rank noise	6.32%	6.36%	6.87%	5.62%	0.91%

Table 1: Empirical results comparing our algorithm with OLS and other robust regression methods. Note that DNC indicates the algorithm in consideration did not converge despite multiple attempts. All percentages indicate relative errors in the coefficients (β) compared to OLS with the unperturbed dataset.

We observe that our algorithm performs particularly well compared to existing industry standard algorithms for robust regression. Huber and RANSAC perform better than PrSVD with outlier rows, while maintaing similar but slightly higher relative errors in coefficients for other perturbations. We also notice that all algorithms do quite poorly with outlier columns, and that PrSVD does relatively very well with low-rank noise compared to competitor algorithms.

3.3 Image Filtering

Given a perturbed image, a low-rank reconstruction using robust SVD helps to recover a relatively less noisy version of the image. Our approach is two-fold: 1) obtain robust singular vectors from the noisy image by minimizing the

impact of noise 2) obtain robust singular values corresponding to these robust singular vectors using a deep learning framework with a novel robust loss formulation. We provide the pseudocode in Algorithm 2 below.

Algorithm 2 Robust SVD for Image Filtering

Inputs: noisy image $Y \in \mathbb{R}^{64\times64}$
1) Robust Left Singular Vectors
$Y_{column} \leftarrow YD_{column}$ where $D_{column} = \text{diag}[||\text{col}_1(Y)||_2, ||\text{col}_2(Y)||_2, ..., ||\text{col}_{64}(Y)||_2]$
Find SVD of $Y_{column} = U_{column}\Sigma_{column}V_{column}^T$
$U_r \leftarrow U_{column}$ where U represents the matrix containing r robust left singular vectors
2) Robust Right Singular Vectors
$Y_{row} \leftarrow D_{row}Y$ where $D_{row} = \text{diag}[||\text{row}_1(Y)||_2, ||\text{row}_2(Y)||_2, ..., ||\text{row}_{64}(Y)||_2]$
Find SVD of $Y_{row} = U_{row}\Sigma_{row}V_{row}^T$
$V_r \leftarrow V_{row}$ where V represents the matrix containing r robust right singular vectors
3) Robust Singular Values
Train a deep neural network with data that contains both Y and X
$\Sigma_r \leftarrow \texttt{NeuralNetwork}(Y)$ where Σ_r represents the matrix containing r largest robust singular values ▷ Use the neural network to predict robust singular values

Given our input image $Y = X + N$ where Y is the perturbed image, X is the original image and N is the noise, we normalize each row of Y to create the matrix Y_{row} and normalize each column of Y to create the matrix Y_{column}. This step takes inspiration from the robust PCA technique proposed by Locantore et al. (1999) where they showed that projecting all data points in a matrix to the unit sphere naturally limits the contribution of potential outliers. As a result, performing SVD to collect the right and left singular vectors from Y_{row} and Y_{column} respectively provides vectors naturally robust to outliers. Moreover, the potential concern that there might be adversarial noise values present within the unit sphere despite the normalization due to N is partly taken care of with the low-rank approximation naturally reducing their prevalence in the output image, with the rest of it addressed through the robust loss function in the calculation of singular values. We present the architecture we use below in Figure 1.

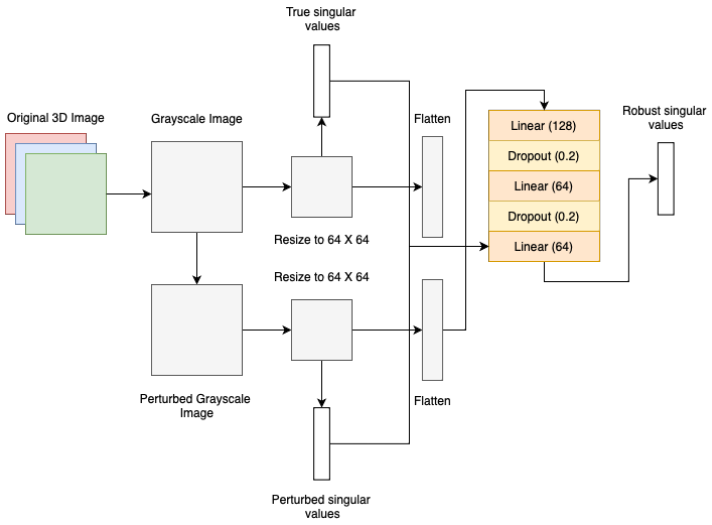

Figure 1: Neural Network Architecture

The most important addition to our neural network architecture is our robust loss function. To motivate its construction, we recognize that our primary goal is to minimize the difference between the predicted singular values s_{pred} and the true singular values s_{true}. In order to adhere to this, our the baseline loss function that we employ is a weighted-Huber loss, given by

$$L_{weighted\ Huber} = \sum_i w_i \cdot L(s_{pred,i}, s_{true,i}), \quad w_i = \frac{s_{true,i}}{||s_{true}||_2}$$

$$L(s_{pred,i}, s_{true,i}) = \begin{cases} \frac{1}{2}(s_{pred,i} - s_{true,i})^2, \text{ if } |s_{pred,i} - s_{true,i}| \leq \delta \\ \delta|s_{pred,i} - s_{true,i}| - \frac{\delta^2}{2}, \text{ otherwise} \end{cases}$$

Since our construction for the proposed neural network architecture takes in both the true and perturbed singular values as inputs for the loss function, we are able to exploit this knowledge to add a regularization term.

$$L_{noisy} = \sum_i \alpha_i \cdot (s_{pred,i} - s_{noisy,i})^2, \quad \alpha_i = \exp\left(-\frac{|s_{pred,i} - s_{noisy,i}|}{\tau}\right)$$

where $\tau > 0$ is a hyperparameter. This noisy term when added to the original Huber loss function helps to dynamically adjust the penalty to each singular value based on how close the perturbed singular value is to the true one. When close, α_i is near 1 to allow for this penalty to contribute to the loss, whereas when they are far apart, α_i is near 0 such that we don't penalize the algorithm for trying to mimic the behavior of the true singular values exactly. Therefore, with the regularizing hyperparameter λ, we have the following final form for the robust loss function

$$L_{robust} = \sum_i w_i \cdot L(s_{pred,i}, s_{true,i}) + \lambda \sum_i \alpha_i \cdot (s_{pred,i} - s_{noisy,i})^2$$

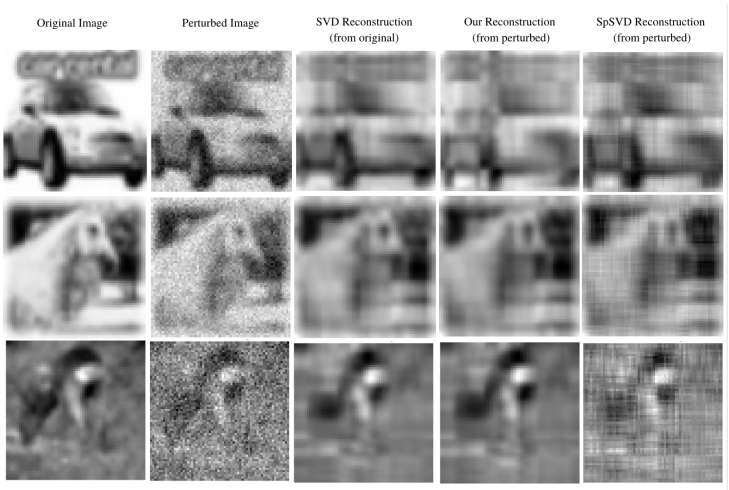

Figure 2: Original, noisy and reconstruction images

We train our neural network model on the CIFAR-10 dataset by perturbing the images with Gaussian noise, and provide results for three sample images in Figure 2. We compare our low-rank reconstruction (using the top 5 singular values i.e. rank 5) to the reconstruction of the low-rank approximation produced by performing SVD on the original image as well as the reconstruction produced by the SpSVD algorithm introduced by Han et al. (2024). As seen in the figure, our algorithm performs quite well, producing reconstructions similar to what SVD performs on the unperturbed image, while performing relatively better than the SpSVD reconstruction.

4 Conclusion and Future Works

In this paper, we extend the robust covariance framework to produce interpretable bounds for robust PCA and perform robust regression in a low-rank setting. We also introduce a robust SVD algorithm for low-rank reconstruction for image filtering. For PCA, we demonstrate theoretical bounds that display the effectiveness of a robust PCA algorithm constructed using robust covariance estimation. For regression and image filtering, results from empirical studies with datasets and image respectively display the effectiveness of our proposed frameworks for low-rank settings.

We note that further work can be performed to provide theoretical bounds for relative errors in beta coefficients for regression and to analyze the breakdown points for the algorithm. For image filtering, we note that we can work to further extend the framework to color images by attempting to perform 3 separate SVDs for each color channel within the same neural network architecture, and adding another loss term to make the overall loss robust to each channel.

References

[1] Candès, E. J., Li, X., Ma, Y., and Wright, J. (2011), "Robust principal component analysis?" *Journal of the ACM (JACM)*, 58, 1–37.

[2] Rahmani, M. and Atia, G. K. (2017), "Coherence pursuit: Fast, simple, and robust principal component analysis," *IEEE Transactions on Signal Processing*, 65, 6260–6275.

[3] Filzmoser P, Nordhausen K. "Robust linear regression for high-dimensional data: An overview." *WIREs Comput Stat*. 2021; 13:e1524. https://doi.org/10.1002/wics.1524

[4] Jambulapati, A., Li, J.Z., Schramm, T., & Tian, K. (2021). "Robust Regression Revisited: Acceleration and Improved Estimation Rates". *Neural Information Processing Systems*.

[5] Wright, J., Ganesh, A., Min, K., and Ma, Y. (2013), "Compressive principal component pursuit," *Information and Inference: A Journal of the IMA*, 2, 32–68.

[6] Zhou, Z., Li, X., Wright, J., Candes, E., and Ma, Y. (2010), "Stable principal component pursuit," in *2010 IEEE International Symposium on Information Theory*, IEEE, pp. 1518–1522.

[7] Xu, H., Caramanis, C., and Sanghavi, S. (2010), "Robust PCA via outlier pursuit," *Advances in Neural Information Processing Systems*, 23.

[8] Gabriel, K. R. and Zamir, S. (1979), "Lower rank approximation of matrices by least squares with any choice of weights," *Technometrics*, 21, 489–498.

[9] Liu, L., Hawkins, D. M., Ghosh, S., and Young, S. S. (2003), "Robust singular value decomposition analysis of microarray data," *Proceedings of the National Academy of Sciences*, 100, 13167–13172.

[10] Ke, Q. and Kanade, T. (2005), "Robust L1 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)*, IEEE, vol. 1, pp. 739–746.

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

[12] Cheng, Y., Diakonikolas, I., Ge, R., & Woodruff, D.P. (2019). "Faster Algorithms for High-Dimensional Robust Covariance Estimation". *Annual Conference Computational Learning Theory*.

[13] Lai, K.A., Rao, A.B., & Vempala, S.S. (2016). "Agnostic Estimation of Mean and Covariance". *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, 665-674.

[14] Han, S., Kim, K., and Jung s. (2024), "Robust SVD Made East: A fast and reliable algorithm for large-scale data analysis," *Proceedings of Machine Learning Research*.