An exploration of Robust Principal Component Analysis (RPCA) in the context of collaborative filtering

Jeffrey Tumminia
New York University
`jt2565@nyu.edu`

Neil Menghani
New York University
`nlm326@nyu.edu`

1 Motivation

Consider a centered matrix $M \in \mathbb{R}^{n \times m}$ that represents a dataset of n observations and m features. In a perfect world, $n >> m$, n would be sufficiently large, M would have "clean" entries (i.e. no missing entries, outliers, or noise) and most statistical methods would be able to model the data. Unfortunately this is almost never the case, and so there exist tools to handle particular circumstances. With the rise of big data and machine learning, it is becoming more common to find datasets where m is particularly large and approaching n. This leads to the curse of dimensionality, which highlights that as the dimensionality of data increases, data becomes sparser and thus statistical learning methods struggle, both in terms of model performance and computational cost to perform the methods.

A common strategy to manage this challenge is to employ dimensionality reduction techniques, which come in many forms. Perhaps the most common is Principal Component Analysis (PCA). PCA is a powerful method to represent a data matrix with many features as a denser representation of a small (pre-determined) number of new uncorrelated features that capture as much of the variance in the data as possible (i.e. minimize information loss) (Brunton and Kutz, 2017). It is commonly defined as the singular value decomposition (SVD) of the sample covariance matrix of our centered dataset. We can also cast PCA as a matrix separation problem $M = L_0 + N_0$, where L_0 is low-rank and N_0 captures the noise in the dataset. This highlights that PCA not only acts as a dimensionality reduction technique but also a method to filter out noise in our data. It is crucial to note that in this context, "noise" is expected to be independent of the features and identically distributed across each feature.

If M fits this use-case, PCA is extremely powerful. However, often M contains impurities that make standard PCA not immediately applicable without cleaning. Firstly, standard PCA is extremely sensitive to outliers. Technically even one large outlier could severely impact the performance of PCA on a dataset. In practice, a small number of outliers are easy for a data scientist to manage, but a large number of outliers or systematically produced outliers are much more difficult to resolve prior to performing dimensionality reduction. Secondly, PCA cannot handle missing data in the matrix (for any feature), which is a common flaw in some datasets and also expected in other sparse datasets where the number of features is comparable to the number of observations $n \approx m, n > m$.

In general, remedies to these problems used by data scientists are often susceptible to introducing bias into the data. If the dataset has missing values or outliers introduced by a systematic issue, imputing the values or removing the observations may lead to a flawed statistical representation of the population. Similarly, if we expect there to be missing data values, we would want a dimensionality reduction technique that can consider this situation. These situations are not just difficult, but relatively common, and this is the fundamental motivation for the large amount of research on **Robust PCA** (RPCA) (Candes et al., 2009).

2 Methodology

In RPCA we instead assume $M = L_0 + S_0$ where L_0 is the same as in PCA and S_0 is a matrix containing sparse outliers and corruption. With this we can define our separation problem as

$$\begin{aligned} \min_{L,S} \quad & rank(L) + ||S||_0 \\ \text{s.t.} \quad & M = L + S \end{aligned} \tag{1}$$

Unfortunately, this optimization problem is non-convex as neither the rank nor 0-norm of a matrix is a convex function. But by Principal Component Pursuit (PCP) (Zhou et al., 2010), (Chandrasekaran et al., 2011) we can get a convex relaxation of the problem

$$\begin{aligned} \min_{L,S} \quad & ||L||_* + \lambda||S||_1 \\ \text{s.t.} \quad & M = L + S \end{aligned} \tag{2}$$

to yield estimates $\hat{L}, \hat{S}$ where $|| \cdot ||_* = \sum_i \sigma_i(\cdot)$ or the sum of singular values, and $|| \cdot ||_1 = \sum_i j| \cdot_{ij} |$ or the sum of absolute values of the entries. A powerful result from PCP is that $\hat{L} = L_0, \hat{S} = S_0$ under certain conditions on L, S. Namely, L should not be sparse and S should not be low-rank. Under these conditions, RPCA has probability $1 - O(n^{-10})$ to find the true (i.e. exact) solution with $\lambda = \frac{1}{\sqrt{n}}$, implying there is no parameter tuning required.

These conditions can mathematically be defined as matrix $M = L + S$ having low coherence, or non-spiky/sparse singular values. Interestingly, even if these conditions are not completely met this process still recovers the low-rank matrix with an error bound proportional to the rate of corruption.

Collaborative Filtering

Collaborative Filtering is the task of inferring feature values for missing entries in your data based on the populated values. It was made popular by the Netflix Prize Competition to infer user movie ratings but has many applications. During collaborative filtering, there are a fraction of entries of M that are present, which we define as Ω.

With this we note that when there is no corruption or outlying data in M (i.e. $S_0 = \mathbf{0}$), then Robust PCA can be defined as

$$\begin{aligned} \min_{L} \quad & ||L||_* \\ \text{s.t.} \quad & M_{ij} = L_{ij} \in \Omega \end{aligned} \tag{3}$$

which is identical to the optimization problem used for exact matrix completion (Candes and Recht, 2008). Thus, when no outliers are present RPCA becomes a matrix completion task completed by minimizing the nuclear norm of L subject to the populated values of M agreeing with the corresponding values of L. This matching constraint can be relaxed in implementation, increasing solutions to the optimization problem. Through this, we see that RPCA has mechanics for handling i.i.d. noise (through standard PCA), sparse outliers (if there are enough present), and missing values (if M is fundamentally low-rank).

3 Experiment

In this experiment, we use the Steam Video Games dataset, which provides a set of users and the number of hours they have played video games in their library. Our goal is to use information about how much users have played various games to infer how much they would play other games, thus framing the task as a collaborative filtering problem. For the sake of having a matrix with sufficiently populated entries, we focus on the top video games and top users. While this problem could generalize for a sparser input matrix, there is no guarantee that such a matrix would have a low-rank representation, and the techniques being applied are computationally expensive for extremely large matrices.

In order to test different techniques, we first hold out 10% of the data to use as a test set. We then generate a 233×133 matrix populated with 8,009 entries ($\sim$ 25.8% of the data) representing the number of hours a given user (of 233 total users) has played a given game (of 133 total games). The dataset has very few outliers; there are only a few cases of a user playing more than 2000 hours, and a vast majority of users play in the 0-100 hour range. Since Robust PCA excels in the presence of corruption, we "corrupt" varying portions of the dataset (from 0% to 30% of the data) by adding random outliers between -10,000 and -1,000 and between 1,000 and 10,000. Once we have datasets with missing and varying amounts of corrupt values, we apply 3 different techniques to complete each matrix: Nuclear Norm Minimization, K-Nearest Neighbors, and Robust PCA (Menghani and Tumminia, 2021) (Ganguli, 2020).

Regardless of the amount of corruption, Robust PCA yields a low-rank component, which can be used to infer missing values, and a sparse component consisting of outliers (see Figures 2 and 4). By contrast, without outliers in the data Nuclear Norm Minimization (NNM) and K-Nearest Neighbors (kNN) reasonably infer missing values, but adding corruption causes these techniques to yield more corrupt matrices (see Figures 3 and 5). Figure 1 demonstrates that while the latter 2 techniques slightly outperform RPCA without corrupt data, adding outliers renders these techniques essentially useless while Robust PCA still performs well. Thus, Robust PCA not only excels in domains with large amounts of missing data (unlike PCA) but also outperforms other techniques in the presence of systematically corrupt data.

References

Steven L. Brunton and J. Nathan Kutz. 2017. *Data Driven Science Engineering*. Brunton Kutz.

Emmanuel J. Candes, Xiaodong Li, Yi Ma, and John Wright. 2009. Robust principal component analysis?

Emmanuel J. Candes and Benjamin Recht. 2008. Exact matrix completion via convex optimization.

Venkat Chandrasekaran, Sujay Sanghavi, Pablo A. Parrilo, and Alan S. Willsky. 2011. Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization*, 21(2):572–596.

Deep Ganguli. 2020. Robust-pca. `https://github.com/dganguli/robust-pca`.

Neil Menghani and Jeffrey Tumminia. 2021. Rpca. `https://github.com/neil-menghani/RPCA`.

Zihan Zhou, Xiaodong Li, John Wright, Emmanuel Candes, and Yi Ma. 2010. Stable principal component pursuit.

A Matrix Completion RMSE

In Figure 1 we show the Root Mean Squared Error (RMSE) on a test set for Robust PCA, Nuclear Norm Minimization (NNM) and K-Nearest Neighbors (kNN) matrix completion techniques with varying amounts of outliers in the data.

B Heatmaps with No Corruption

In Figure 2 we show the resulting components from RPCA and the original matrix M without any added corruption. In Figure 3 we show the completed matrices from NNM and kNN.

C Heatmaps with Corruption

In Figure 4 we show the resulting components from RPCA and the original matrix M with 10% added corruption. In Figure 5 we show the completed matrices from NNM and kNN.

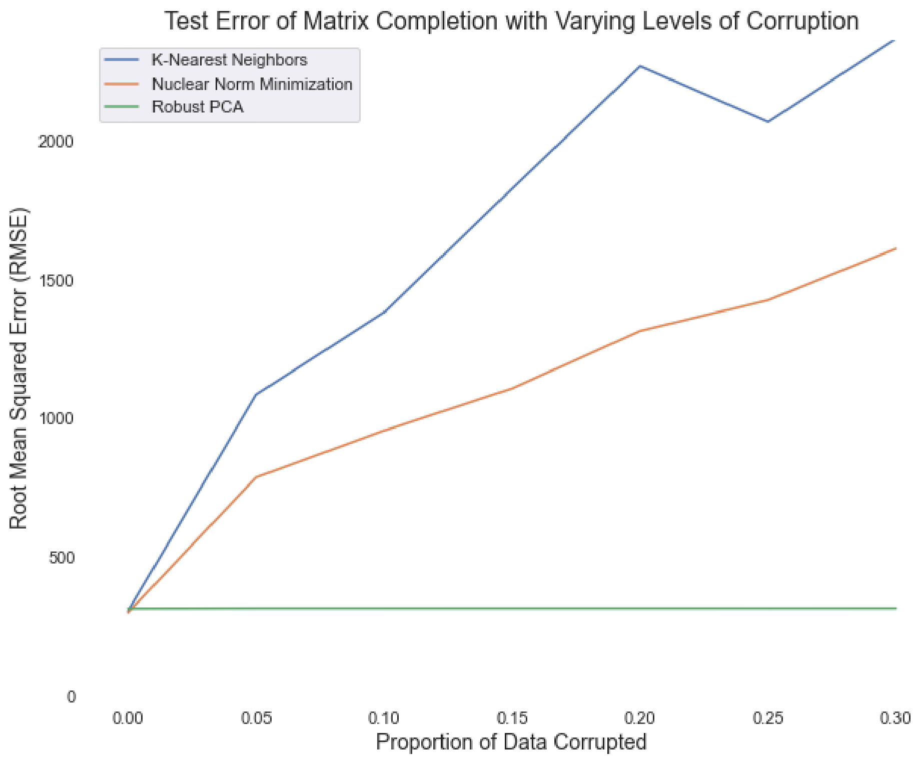
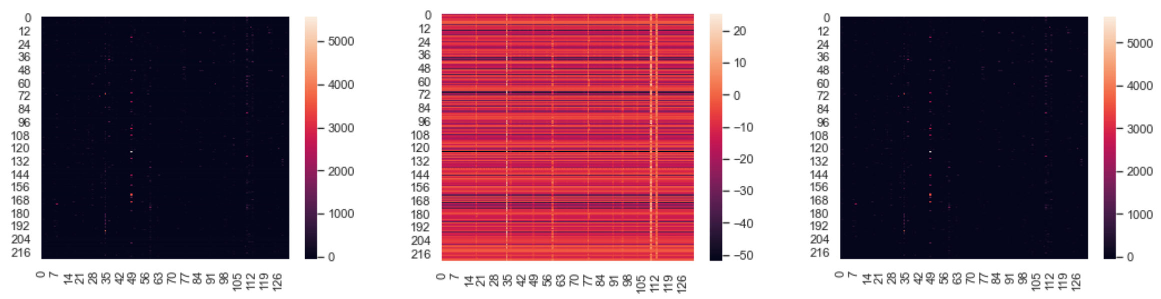


Figure 1: Plot of Root Mean Squared Error (RMSE) on test set for 3 matrix completion techniques with varying levels of corruption. K-Nearest Neighbors and Nuclear Norm Minimization slightly outperform Robust PCA on dataset with no corruption. However, performance of the former 2 techniques significantly deteriorates as corruption is added, while RPCA performs consistently in the presence of outliers.

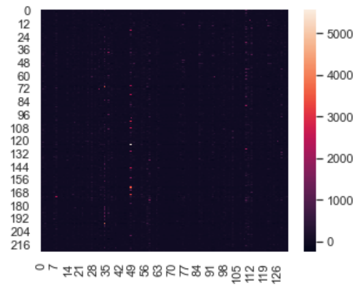


(a) Centered Data Matrix
M

(b) Low-Rank Component
L (CF Solution)

(c) Sparse Component
S (extracted outliers)

Figure 2: Heatmaps of Robust PCA output with no corruption.
Games displayed on x-axis and users on y-axis. Low-rank and sparse components extracted. Recall that RPCA minimizes objective function subject to M = L + S.


(a) Low-Rank Output L
Nuclear Norm Minimization

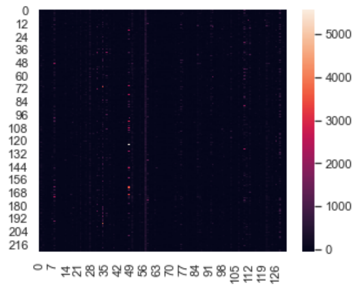
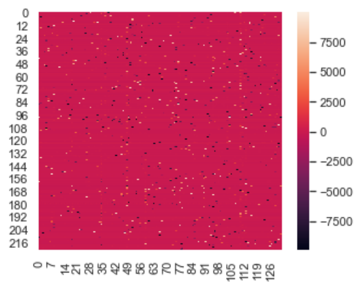
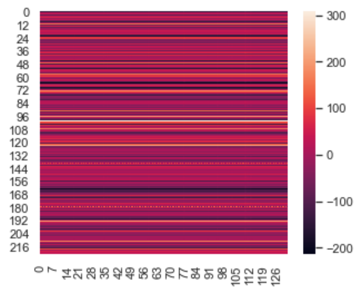

(b) Matrix Completion Solution
K-Nearest Neighbors

Figure 3: Heatmaps of traditional matrix completion techniques with no corruption. Games displayed on x-axis and users on y-axis. For certain games, entries in original matrix filled in with values inferred by the 2 techniques.


(a) Centered Matrix M
with Corruption


(b) Low-Rank Component
L (CF Solution)

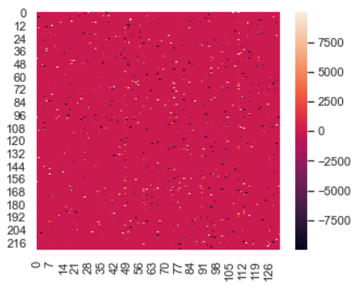
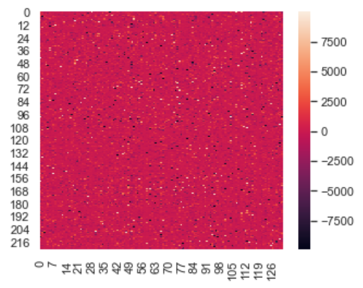

(c) Sparse Component
S (extracted outliers)

Figure 4: Heatmaps of Robust PCA output with 10% corruption. Games displayed on x-axis and users on y-axis. A coherent low-rank component is still extracted, and outliers added by corruption are pulled into the sparse component. As in Figure 3, M=L+S.


(a) Low-Rank Output L
Nuclear Norm Minimization

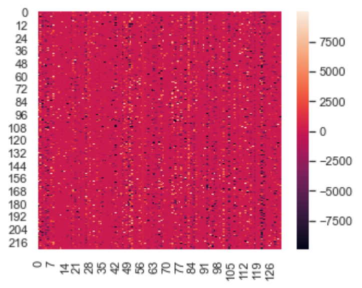

(b) Matrix Completion Solution
K-Nearest Neighbors

Figure 5: Heatmaps of traditional matrix completion techniques with 10% corruption. Games displayed on x-axis and users on y-axis. For certain games, entries in original matrix filled in with values inferred by the 2 techniques.