An Overview of Low-rank Matrix Completion using Convex Optimization Techniques

Ramiro Deo-Campo Vuong Yibo Wen David Haolong Lee
Aditya Prasad

May 13, 2022

1 Introduction

In this paper, we explore the use of convex optimization in compressed sensing. Specifically, we examine the application of convex programming in the problem of low-rank matrix and tensor reconstruction from sparse sampling.

We begin by giving a brief overview of the linear algebra techniques that will be used frequently throughout our paper. These include singular value decomposition (SVD) of matrices, semi-unitary matrices and their properties, and orthogonal projection matrices and their properties. We provide an explanation of each topic and review key properties that will be necessary for our proofs.

Next, we give an overview of the proof that low-rank matrix and tensor completion can be represented as a convex optimization problem (specifically one of minimizing nuclear norm). Then we provide the main theorem of matrix completion, which gives a lower bound on the number of samples necessary to exactly reconstruct the original matrix with high probability.

We then use matrix reconstruction in two applications: NBA Finals Prediction and 3D model completion. Finally, we apply our algorithm for matrix reconstruction and examine its performance.

2 Linear Algebra Overview

This section provides an overview of the Linear Algebra concepts used in this report. The explanation of these concepts extensively references the book *Matrix Algebra*, written by Abadir and Magnus (2005).

2.1 Complex Conjugate

The complex conjugate is a transformation on matrices. To calculate the complex conjugate of $\boldsymbol{A}$, take the transpose $\boldsymbol{A}^T$ and then replace each entry with its complex conjugate. For example, the entry $a + bi$ would have a complex conjugate of $a - bi$. We denote the complex conjugate as $\boldsymbol{A}^*$ in this survey.

2.2 Semi-Unitary Matrices

A matrix $\boldsymbol{A}$ is semi-unitary if and only if $\boldsymbol{A}^*\boldsymbol{A} = \boldsymbol{I}$ or $\boldsymbol{A}\boldsymbol{A}^* = \boldsymbol{I}$. If $\boldsymbol{A} \in \mathbb{R}^{n_1 \times n_2}$ where $n_1 > n_2$ then $\boldsymbol{A}^*\boldsymbol{A} = \boldsymbol{I}$. In this paper, this property will only ever be applied on matrices with more rows than columns.

2.3 Orthogonal Projection Matrices

An orthogonal projection matrix $\boldsymbol{P}$ satisfies $\boldsymbol{P}^2 = \boldsymbol{P} = \boldsymbol{P}^*$. Orthogonal projection matrices can also be used to project vectors onto a subspace. Let the columns of $\boldsymbol{S}$ be the basis of some subspace $\mathcal{S}$. Then $\boldsymbol{P}_\mathcal{S}$, the orthogonal projection matrix onto the subspace $\mathcal{S}$ is defined as follows:

$$\boldsymbol{P}_\mathcal{S} = \boldsymbol{S}(\boldsymbol{S}^*\boldsymbol{S})^{-1}\boldsymbol{S}^*$$

Note that $\boldsymbol{P}_\mathcal{S}\boldsymbol{x}$ will be in the subspace $\mathcal{S}$ ($\boldsymbol{P}_\mathcal{S}\boldsymbol{x}$ can be expressed as a linear combination of the columns of $\boldsymbol{S}$).

2.4 Compact Singular Value Decomposition (SVD)

Compact singular value decomposition decomposes any matrix into three matrices: a left singular vector matrix, a singular value matrix, and a right singular vector matrix. The matrix $\boldsymbol{M} \in \mathbb{R}^{n_1 \times n_2}$ can be expressed as:

$$\boldsymbol{M} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*$$

Both $\boldsymbol{U}$ and $\boldsymbol{V}$ are unitary matrices. $\boldsymbol{\Sigma}$ is a diagonal matrix: $\text{diag}(\sigma_1, \ldots, \sigma_r)$.[1] Compact SVD removes all zero singular values along with corresponding columns of $\boldsymbol{U}$ and $\boldsymbol{V}$, so $\boldsymbol{U} \in \mathbb{R}^{n_1 \times r}, \boldsymbol{\Sigma} \in \mathbb{R}^{r \times r}$, and $\boldsymbol{V} \in \mathbb{R}^{n_2 \times r}$ where r is the rank of matrix M. The above formulation is equivalent to

$$\boldsymbol{M} = \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^*$$

where $[r] = \{0, \ldots, r\}$, $\boldsymbol{U} = \begin{bmatrix}\boldsymbol{u}_1 & \cdots & \boldsymbol{u}_r\end{bmatrix}$, and $\boldsymbol{V} = \begin{bmatrix}\boldsymbol{v}_1 & \cdots & \boldsymbol{v}_r\end{bmatrix}$. Useful properties of compact SVD that will be used later is that the projection matrices onto the column space ($\boldsymbol{P}_U$) and row space ($\boldsymbol{P}_V$) are given by

$$\boldsymbol{P}_U = \boldsymbol{U}(\boldsymbol{U}^*\boldsymbol{U})^{-1}\boldsymbol{U}^* = \boldsymbol{U}\boldsymbol{U}^*$$
$$\boldsymbol{P}_V = \boldsymbol{V}(\boldsymbol{V}^*\boldsymbol{V})^{-1}\boldsymbol{V}^* = \boldsymbol{V}\boldsymbol{V}^*$$

2.5 Useful Norms

Nuclear Norm $\|\boldsymbol{M}\|_*$ is the sum of the singular values of a matrix $\boldsymbol{M}$.

Spectral Norm $\|\boldsymbol{M}\|$ is the largest singular value of a matrix $\boldsymbol{M}$.

[1] We assume for the rest of this paper that singular values are sorted in descending order, so $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r$.

2.6 Von Neumann's Trace Inequality

$$\text{tr}(\boldsymbol{Y}^\top \boldsymbol{X}) \leq \sum_{i=1}^{q} \sigma_i(\boldsymbol{Y})\sigma_i(\boldsymbol{X})$$

This provides an upper bound on the trace of the LHS. This inequality will be useful for converting the conjugate of the rank function to a more usable state, as seen later in the paper.

3 Matrix Completion as Convex Optimization Problem

3.1 Formulation

In matrix completion, we would like to recover the entries of a low-rank matrix by sampling a sparse set of entries from the original matrix. Let $\phi(\boldsymbol{X})$ denote the rank of matrix $\boldsymbol{X}$, and let $\boldsymbol{M}$ be the matrix that we are trying to find and $\boldsymbol{X}$ be variable of the convex program. If we have a masking matrix $\Omega \in \{0,1\}^{n_1 \times n_2}$ which represents the positions that we sample and a function $\mathcal{P}_\Omega : \mathbb{R}^{n_1 \times n_2} \to \mathbb{R}^{n_1 \times n_2}$ that applies the mask, our problem can be written as:

$$\begin{aligned} &\text{minimize} && \phi(\boldsymbol{X}) \\ &\text{subject to} && \mathcal{P}_\Omega(\boldsymbol{X}) = \mathcal{P}_\Omega(\boldsymbol{M}) \end{aligned} \tag{1}$$

Unfortunately, rank is not a convex function, so this is not a convex program. We solve this problem by taking the double conjugate of the rank function. By the properties of conjugates, we know this will yield the the convex envelope of the rank function, which is guaranteed to be convex.

Lemma 1. *The nuclear norm $\|\cdot\|_*$ is the convex envelope of the rank function ϕ.*

This section uses information from Fazel (2002). First, we will examine the conjugate of the rank function:

$$\begin{aligned} \phi^*(\boldsymbol{Y}) &= \sup_{\|\boldsymbol{X}\| \leq 1} \left(\langle \boldsymbol{Y}, \boldsymbol{X} \rangle - \phi(\boldsymbol{X})\right) \\ &= \sup_{\|\boldsymbol{X}\| \leq 1} \left(\text{tr}(\boldsymbol{Y}^\top \boldsymbol{X}) - \phi(\boldsymbol{X})\right) \end{aligned} \tag{2}$$

Note that we must bound the spectral norm of $\boldsymbol{X}$ ($\|\boldsymbol{X}\| \leq 1$), because otherwise we could pick an $\boldsymbol{X}$ with an arbitrarily high spectral norm and the first term of the supremum would always be ∞. For the remainder of this proof, let $q = \min(n_1, n_2)$. Subsection 2.6 states:

$$\text{tr}(\boldsymbol{Y}^\top \boldsymbol{X}) \leq \sum_{i=1}^{q} \sigma_i(\boldsymbol{Y})\sigma_i(\boldsymbol{X}) \tag{3}$$

As stated previously, Inequality 3 gives an upper bound on the first term in Equation 2. Fazel showed that when taking the SVD of $\boldsymbol{X} = \boldsymbol{U}_X \boldsymbol{\Sigma}_X \boldsymbol{V}_X^\top$ and $\boldsymbol{Y} = \boldsymbol{U}_Y \boldsymbol{\Sigma}_Y \boldsymbol{V}_Y^\top$, setting $\boldsymbol{U}_X = \boldsymbol{U}_Y$ and $\boldsymbol{V}_X = \boldsymbol{V}_Y$ will result in $\text{tr}(\boldsymbol{Y}^\top \boldsymbol{X}) = \sum_{i=1}^{q} \sigma_i(\boldsymbol{Y})\sigma_i(\boldsymbol{X})$ because $\mathbf{U}$ and $\mathbf{V}$ are orthonormal bases, so solving the following supremum is equivalent:

$$\phi^*(\boldsymbol{Y}) = \sup_{\|\boldsymbol{X}\| \leq 1} \left(\sum_{i=1}^{q} (\sigma_i(\boldsymbol{Y})\sigma_i(\boldsymbol{X})) - \phi(\boldsymbol{X}) \right)$$

It turns out that this simplifies to two cases:

$$\phi^*(\boldsymbol{Y}) = \begin{cases} 0, & \text{when } \|\boldsymbol{Y}\| \leq 1 \\ \sum_{i=1}^{r} [\sigma_i(\boldsymbol{Y}) - 1], & \text{where } \sigma_r(\boldsymbol{Y}) > 1 \text{ and } \sigma_{r+1}(\boldsymbol{Y}) \leq 1 \end{cases}$$

In other words, the conjugate of the rank is the sum of the r singular values that are greater than 1 (each elementwise subtracted by 1). To get the convex envelope of rank, we take the double conjugate:

$$\begin{aligned} \phi^{**}(\boldsymbol{Z}) &= \sup_{\boldsymbol{Y}} (\text{tr}(\boldsymbol{Z}^\top \boldsymbol{Y}) - \phi^*(\boldsymbol{Y})) \\ &= \sup_{\boldsymbol{Y}} \left(\sum_{i=1}^{q} (\sigma_i(\boldsymbol{Z})\sigma_i(\boldsymbol{Y})) - \phi^*(\boldsymbol{Y}) \right) \end{aligned}$$

By a very similar series of steps as we did for the first conjugate. We will examine the following three cases.

- If $\|\boldsymbol{Z}\| > 1$, then by writing out the full expression for $\phi^*(\boldsymbol{Y})$ and factoring, we see that the supremum will be ∞.

$$\begin{aligned} \sup_{\boldsymbol{Y}} \left(\sum_{i=1}^{q} (\sigma_i(\boldsymbol{Z})\sigma_i(\boldsymbol{Y})) - \phi^*(\boldsymbol{Y}) \right) &= \sup_{\boldsymbol{Y}} (\sigma_1(\boldsymbol{Z})\sigma_1(\boldsymbol{Y}) - \sigma_1(\boldsymbol{Y}) + ... - r) \\ &= \sup_{\boldsymbol{Y}} (\sigma_1(\boldsymbol{Y})(\sigma_1(\boldsymbol{Z}) - 1) + ... - r) \end{aligned}$$

  We can simply let $\sigma_1(\boldsymbol{Y})$ approach ∞ (since it is multiplied by a positive value) and achieve a supremum of infinity. Therefore, in order to achieve any meaningful results, we must bound $\|\mathbf{Z}\| \leq \mathbf{1}$.

- If $\|\boldsymbol{Z}\| \leq 1$ and $\|\boldsymbol{Y}\| \leq 1$, then we know that $\phi(\mathbf{Y}) = \mathbf{0}$, so to maximize, we choose a $\mathbf{Y}$ such that all the singular values $\sigma_1 = ... = \sigma_q = 1$.

$$\phi^{**}(\mathbf{Z}) = \sum_{\mathbf{i=0}}^{\mathbf{q}} \sigma_{\mathbf{i}}(\mathbf{Z}) = \|\mathbf{Z}\|_*$$

- If $\|\boldsymbol{Z}\| \leq 1$ and $\|\boldsymbol{Y}\| > 1$, then we have

$$\begin{aligned} \sum_{i=1}^{q} \sigma_i(\boldsymbol{Z})\sigma_i(\boldsymbol{Y}) - \phi^*(\boldsymbol{Y}) &= \sum_{i=1}^{q} \sigma_i(\boldsymbol{Z})\sigma_i(\boldsymbol{Y}) - \sum_{i=1}^{r} (\sigma_i(\boldsymbol{Y}) - 1) + \sum_{i=1}^{q} \sigma_i(\boldsymbol{Z}) - \sum_{i=1}^{q} \sigma_i(\boldsymbol{Z}) \\ &= \sum_{i=1}^{r} (\sigma_i(\boldsymbol{Z})\sigma_i(\boldsymbol{Y}) - \sigma_i(\boldsymbol{Z}) - \sigma_i(\boldsymbol{Y}) + 1) + \sum_{i=r+1}^{q} \sigma_i(\boldsymbol{Z})(\sigma_i(\boldsymbol{Y}) - 1) + \sum_{i=1}^{q} \sigma_i(\boldsymbol{Z}) \\ &= \sum_{i=1}^{r} ((\sigma_i(\boldsymbol{Z}) - 1)(\sigma_i(\boldsymbol{Y}) - 1)) + \sum_{i=r+1}^{q} \sigma_i(\boldsymbol{Z})(\sigma_i(\boldsymbol{Y}) - 1) + \sum_{i=1}^{q} \sigma_i(\boldsymbol{Z}) \quad (4) \\ &\leq \sum_{i=1}^{q} \sigma_i(\boldsymbol{Z}) \\ &= \|\boldsymbol{Z}\|_* \quad (5) \end{aligned}$$

  We know that line (4) is less than line (5) since the first two terms in the summation are always negative (remember we defined the first r singular values of $\mathbf{Y}$ to be greater than 1 and the less to be less than 1.

Therefore, when we bound $\|\mathbf{X}\| < \mathbf{1}$ and limit ourselves to cases 2 and 3, (and rename the variable $\mathbf{Z}$ to $\mathbf{X}$) we achieve:

$$\begin{array}{ll} \text{minimize} & \|\boldsymbol{X}\|_* \\ \text{subject to} & \mathcal{P}_\Omega(\boldsymbol{X}) = \mathcal{P}_\Omega(\boldsymbol{M}) \end{array} \tag{6}$$

Later, we show that the optimal solution to this Nuclear Norm Minimization problem is equivalent to the original matrix M under certain conditions.

3.2 Coherence

This section uses information presented in Candès and Recht (2009).

Coherence is a measure of how close a subspace is to containing a standard basis. We will see that coherence is a factor determining how many entries sampled uniformly at random are needed for exact matrix reconstruction. Before formally exploring how coherence relates to matrix reconstruction, consider the following definition: the coherence of a subspace U with dimension r is given by

$$\mu(U) = \frac{n}{r} \max_{1 \le i \le n} \|\boldsymbol{P}_U \boldsymbol{e}_i\|^2$$

where $\boldsymbol{P}_U \in \mathbb{R}^{n \times n}$ is the projection from $\mathbb{R}^n$ onto the subspace U. Notice that a subspace U containing a standard basis vector $\boldsymbol{e}_i$ will have

$$\begin{aligned} \mu(U) &= \frac{n}{r} \max \|\boldsymbol{P}_U \boldsymbol{e}_i\|^2 \\ &= \frac{n}{r} \|\boldsymbol{e}_i\|^2 \\ &= \frac{n}{r} \end{aligned}$$

This is clearly an upper bound because the projection of a unit vector onto a subspace cannot have length greater than 1. Candès and Recht also note that the minimum bound for coherence has a value of 1, and occurs when the entries of $\boldsymbol{U}$ have magnitude $\frac{1}{\sqrt{n}}$.

A matrix is considered low coherence if its column and row spaces have low coherence. It turns out that matrices with low coherence can be reconstructed with fewer entries sampled uniformly at random. We will explain this in detail in the next section, but this result is intuitively correct because matrices with high coherence will have rows that look very similar to standard basis vectors, and thus we need to sample more to guarantee that we select the non-zero entry.

3.3 Coherence and Matrix Completion

For intuition about why sampling uniformly at random is more effective for low coherence matrices, consider the SVD formulation of the target matrix $\boldsymbol{M}$

$$\boldsymbol{M} = \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^*$$

Let $\boldsymbol{U}$ be a matrix with columns $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_r$ and let $\boldsymbol{V}$ be a matrix with columns $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_r$. By the properties of SVD, $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$ form the basis for the column space and row space, respectively. Take a basis vector of the column space to be some arbitrary standard basis vector ($\boldsymbol{u}_k = \boldsymbol{e}_j$); the matrix has high coherence because the column space has high coherence. Notice that $\sigma_k \boldsymbol{u}_k \boldsymbol{v}_k^* = \sigma_k \boldsymbol{e}_j \boldsymbol{v}_k^*$,

which is a matrix with zeroes in every entry except the j^{th} row. If none of the j^{th} row entries are sampled, then there is no distinction between $\mathcal{P}_\Omega(\sum_{i=1}^r \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^*)$ and $\mathcal{P}_\Omega(\sum_{i=1}^r [\sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^*] - \sigma_k \boldsymbol{u}_k \boldsymbol{v}_k)$. As a result, $\boldsymbol{M}$ cannot be an optimal solution to the nuclear norm minimization problem (6). Let $\boldsymbol{Y} = \sum_{i=1}^r [\sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^*] - \sigma_k \boldsymbol{u}_k \boldsymbol{v}_k$, $\boldsymbol{M} = \sum_{i=1}^r \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^*$, and OPT denote the set of optimal solutions to (6). Then:

$$\|\boldsymbol{Y}\|_* < \|\boldsymbol{M}\|_* \text{ and } \mathcal{P}_\Omega(\boldsymbol{Y}) = \mathcal{P}_\Omega(\boldsymbol{M}) \implies \boldsymbol{M} \notin OPT$$

The nuclear norm of $\boldsymbol{Y}$ is less than that of $\boldsymbol{M}$ because it has one less singular value and the nuclear norm is the summation of singular values.

For the above reason, some assumptions must be made about the coherence of matrix $\boldsymbol{M}$ to tightly bound the number of necessary samples for matrices of different coherence.

Assumption 1 (A1)**.**

$$\max(\mu(\boldsymbol{U}), \mu(\boldsymbol{V})) \leq \mu_0$$

where matrix $\boldsymbol{U}$ has rows $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_r$ and matrix $\boldsymbol{V}$ has rows $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_r$.

Assumption 2 (A2)**.** Every entry of the matrix $\sum_{i \in [r]} \boldsymbol{u}_i \boldsymbol{v}_i^* \in \mathbb{R}^{n_1 \times n_2}$ is less than or equal to $\mu_1 \sqrt{\frac{r}{n_1 n_2}}$

Both, μ_0 and μ_1 measure how distributed the influence of singular values and rows are on entries of matrix $\boldsymbol{M}$. Recall the example highly coherent singular vectors had on entries of $\boldsymbol{M}$. As coherence increases, singular vectors and values begin to be expressed on fewer and fewer entries of $\boldsymbol{M}$ and sampling uniformly at random may miss this smaller set of entries. The main theorem will establish a lower bound on the necessary number of sampled entries in terms of n, r, μ_0, and μ_1 to account for matrix size, rank, and coherence.

For note, these assumptions are made by Candès and Recht (2009). Other papers, like Candes and Tao (2010) make different variations assumptions that produce slightly different bounds on the necessary number of entries to sample.

3.4 Main Theorem

This uses information presented in Candès and Recht (2009) and Candes and Tao (2010).

Theorem 1. *Take $\boldsymbol{M} \in \mathbb{R}^{n_1 \times n_2}$ satisfying incoherence assumptions A1 and A2 with parameters μ_1 and μ_2. Let $n = \max(n_1, n_2)$ and let r be the rank of $\boldsymbol{M}$. Given m entries of $\boldsymbol{M}$ sampled uniformly at random, where m satisfies*

$$m \geq C \max(\mu_1^2, \mu_0^{1/2} \mu_1, \mu_0 n^{1/4}) \cdot nr(\beta \log n)$$

for some constants C and c, the matrix $\boldsymbol{M}$ will be the unique optimal solution to the nuclear norm minimization problem (6) *with probability $1 - cn^{-\beta}$ for some $\beta \geq 2$.*

This theorem provides lower bound on the necessary number of samples m. Interestingly, there are very strong guarantees: if enough entries of $\boldsymbol{M}$ are known, $\boldsymbol{M}$ can be reconstructed exactly with very high probability.

Other variations of this theorem also take the form $m \geq poly(nr \cdot \log n)$. Notably, at least $O(n \log n)$ samples are needed. Candes and Tao attribute this to the "coupon collection effect", in which one

must draw $O(n \log n)$ times uniformly at random from a set of n unique items to collect all items (assuming replacement). With respect to entry sampling, every row and column must be sampled to reconstruct M. The sampling must be a "complete collection" of the rows and columns (Candès & Recht, 2009).

3.5 Proof Outline of Main Theorem

This survey only provides a summary of the proofs required for the main theorem. Candes and Recht provide the exhaustive proof with all steps (Candès & Recht, 2009).

The nuclear norm minimization problem (6) has been shown to be convex. By duality, the existence of a dual certificate $\boldsymbol{Y}$ can be used to confirm the optimally the primal $\boldsymbol{X} = \boldsymbol{M}$ for (6). Take the Lagrangian of (6):

$$L(\boldsymbol{X}, \boldsymbol{Y}) = \|\boldsymbol{X}\|_* + \langle \boldsymbol{Y}, (\mathcal{P}_\Omega(\boldsymbol{M}) - \mathcal{P}_\Omega(\boldsymbol{X}))\rangle$$

Although nuclear norm is not differentiable across the entire domain, it has a sub-gradient at every point. Let $\partial\|\boldsymbol{X}\|_*$ be the set of all sub-gradients of the nuclear norm at $\boldsymbol{X}$. If $\boldsymbol{M}$ is an optimal solution, then by the KKT conditions, $\exists \boldsymbol{Y} \in \mathbb{R}^{n_1 \times n_2}$ such that

$$\begin{aligned} \mathbf{0} \in \partial\|\boldsymbol{M}\| + \Delta\langle \boldsymbol{Y}, (\mathcal{P}_\Omega(\boldsymbol{M}) - \mathcal{P}_\Omega(\boldsymbol{X}))\rangle &\equiv \mathbf{0} \in \partial\|\boldsymbol{M}\|_* + \langle \boldsymbol{Y}, -\mathcal{P}_\Omega\rangle \\ &\equiv \mathbf{0} \in \partial\|\boldsymbol{M}\|_* - \mathcal{P}_\Omega(\boldsymbol{Y}) \\ &\equiv \mathcal{P}_\Omega(\boldsymbol{Y}) \in \partial\|\boldsymbol{M}\|_* \end{aligned}$$

The set $\partial\|\boldsymbol{M}\|_*$ is known to include $\sum_{i=1}^{r} \boldsymbol{u}_i \boldsymbol{v}_i^* + \boldsymbol{W}$ if and only if

- The column space of $\boldsymbol{W}$ is orthogonal to the column space of M and the row space of $\boldsymbol{W}$ is orthogonal to the row space of M
- $\|\boldsymbol{W}\| \leq 1$

To simplify the above, define an orthogonal decomposition $\mathcal{P}_T \bigoplus \mathcal{P}_{T^\perp}$, where

$$\begin{aligned} \mathcal{P}_T(\boldsymbol{X}) &= \boldsymbol{P}_U \boldsymbol{X} + \boldsymbol{X}\boldsymbol{P}_{\boldsymbol{V}} - \boldsymbol{P}_U \boldsymbol{X} \boldsymbol{P}_V \\ \mathcal{P}_{T^\perp}(\boldsymbol{X}) &= (\boldsymbol{\mathcal{I}} - \mathcal{P}_T)(\boldsymbol{X}) \end{aligned}$$

This decomposition is designed in such a way that $\mathcal{P}_\Omega$ restricted to the domain T is injective. Thus, the main theorem can be proven by showing:

- $\mathcal{P}_T(\boldsymbol{Y}) = \sum_{i=1}^{r} \boldsymbol{u}_k \boldsymbol{v}_k^*$
- $\|\mathcal{P}_{T\perp}\boldsymbol{Y}\| \leq 1$

Note that a $\boldsymbol{Y}$ satisfying the above will be in the set $\partial\|\boldsymbol{M}\|_*$. The entire main theorem to this point has been reduced to showing that the above is satisfied with high probability. From here, Candès and Recht, 2009 identify a matrix satisfying the equality constraints and show its spectral norm is less than one with high probability. Candes and Tao, 2010 employ moment methods to show random matrices satisfy these conditions with high probability.

4 Applications

4.1 NBA Game Prediction

In an effort to explore the power of low-rank matrix completion in realistic settings, we make use of low-rank matrix completion in predicting the results of NBA games during regular seasons. We evaluate the performance of low-rank matrix completion subject to different interpretation of the NBA data set and different formulations of the low-rank matrix completion problem.

The data set consists of the statistics of NBA teams during the season 2018-2019 obtained from www.basketball-reference.com. The data is obtained via a web-scraper written in Python. The source also defines the following related metrics.

Offensive Rating (ORtg) The ORtg of a team measures the number of "points scored per 100 possessions".

Pace The pace factor is an estimate of the number of possessions per 48 minutes by a team.

As usual, let $\boldsymbol{M} \in \mathbb{R}^{n_1 \times n_2}$ be the target matrix and $\boldsymbol{X} \in \mathbb{R}^{n_1 \times n_2}$ be our variable. Furthermore, let $\boldsymbol{O} \in \mathbb{R}^{n_1 \times n_2}$ denote for each entry $\boldsymbol{O}_{ij}$ the average offensive rating of team i when playing against team j, and $\boldsymbol{P} \in \mathbb{R}^{n_1 \times n_2}$ denote for each entry $\boldsymbol{P}_{ij}$ the pace of the game between team i and j. Note that $n_1 = n_2 = 30$ since there are 30 teams in the NBA. We begin with an extreme simplification of the games, using the following formula, which we call "score potential," to evaluate how well team i does against team j

$$\boldsymbol{M}_{ij} = \frac{\boldsymbol{O}_{ij}\boldsymbol{P}_{ij}}{100} \tag{7}$$

The reasoning behind this equation is as follows: offensive rating gives us the points a team scores in 100 possessions, so we divide by 100 to get the average points scored per possession. Then, we multiply by the estimated number of possessions per game (pace) to get the predicted score potential for the match-up. We assume that if $\boldsymbol{M}_{ij} > \boldsymbol{M}_{ji}$, then team i wins. Conversely, team j wins.

We attempt to find $\boldsymbol{M}$ by recovering $\boldsymbol{O}$ and $\boldsymbol{P}$ respectively given some $\mathcal{P}_\Omega(\boldsymbol{O})$ and $\mathcal{P}_\Omega(\boldsymbol{P})$. Let K denote the proportion of entries observed in $\boldsymbol{O}$ and $\boldsymbol{P}$.

4.1.1 Naive Nuclear Norm Minimization

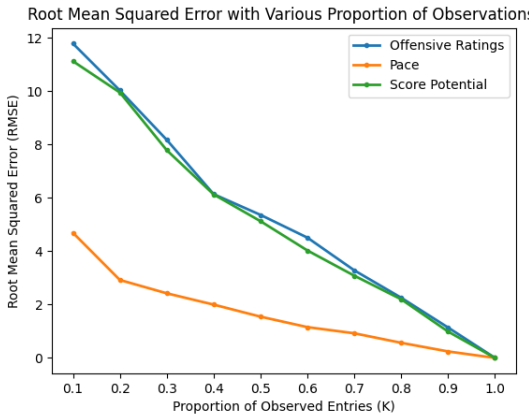
In this section, we explore the accuracy of the original nuclear norm minimization SDP described in (6).

The root mean squared error of the recovered matrix $\hat{\boldsymbol{O}}$ (and similarly for $\hat{\boldsymbol{P}}$ and $\hat{\boldsymbol{M}}$) is calculated using the following formula:

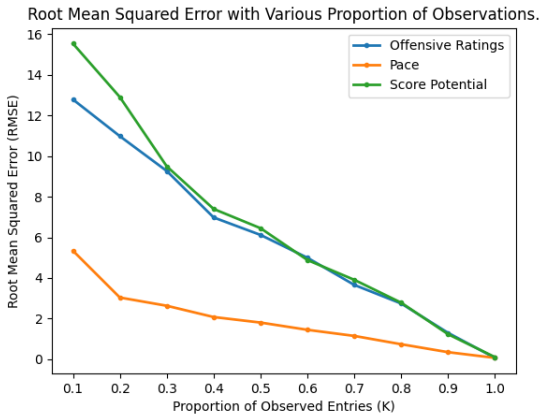
$$RMSE(\hat{\boldsymbol{O}}) = \frac{1}{n_1 n_2} \sum_{i=0}^{n_1} \sum_{j=0}^{n_2} \sqrt{(\hat{\boldsymbol{O}}_{ij} - \boldsymbol{O}_{ij})^2}$$

As shown in Figure 1a, RMSE and K exhibit an approximately linear relationship.

However, this approach would potentially lead to over-fitting of the model (Mazumder et al., 2010), because the constraint $\mathcal{P}_\Omega(\boldsymbol{X}) = \mathcal{P}_\Omega(\boldsymbol{M})$ from (6) ensures that our model performs perfectly on observed inputs, but makes no guarantees on the unobserved inputs.



(a) $\hat{\boldsymbol{O}}$ and $\hat{\boldsymbol{P}}$ is computed using (6).



(b) $\hat{\boldsymbol{O}}$ and $\hat{\boldsymbol{P}}$ is computed using (9).

Figure 1: Root mean squared error (RMSE) of offensive ratings $\hat{\boldsymbol{O}}$ and pace $\hat{\boldsymbol{P}}$ with respect to $\boldsymbol{O}$ and $\boldsymbol{P}$, respectively. $\hat{\boldsymbol{O}}$ and $\hat{\boldsymbol{P}}$ is computed using different formulation of nuclear norm minimization using the common mask Ω that samples Kn_1n_2 observations uniformly at random. The score potential is computed using (7) with input $\hat{\boldsymbol{O}}$ and $\hat{\boldsymbol{P}}$.

4.1.2 Relaxed Nuclear Norm Minimization

To prevent our model from over-fitting, we motivate the following convex relaxation of (6) described by Fazel (2002).

$$\begin{aligned} &\text{minimize} && \|\boldsymbol{X}\|_* \\ &\text{subject to} && \sum_{(i,j)\in\Omega} (\boldsymbol{M}_{ij} - \boldsymbol{X}_{ij})^2 \le \delta \end{aligned} \tag{8}$$

where $\delta \ge 0$ is a parameter. As opposed to (6), the constraint of (8) allows for a square loss of at most δ, allowing us to prevent the model from over-fitting.

According to Mazumder et al. (2010), (8) can be rewritten in the following Lagrangian form

$$\text{minimize} \quad \tfrac{1}{2} \sum_{(i,j)\in\Omega} (\boldsymbol{M}_{ij} - \boldsymbol{X}_{ij})^2 + \lambda \|\boldsymbol{X}\|_*. \tag{9}$$

where λ is a hyper-parameter controlling the trade-off between nuclear norm and square loss. We attempt to tune the model by solving (9) with various λ input, assuming that $K = 0.9$. The result of the model tuning is shown in Figure 2. We conclude that the $\lambda = 25$ is optimal for predicting the score potential with $K = 0.9$. Although ideally we should find optimal λ for all K, doing so will be computationally expensive and time-consuming, so instead we adopt $\lambda = 25$ as the optimal for all K. Thus, we will use $\lambda = 25$ as our hyper-parameter for any solving of (9).

As shown in Figure 3, (6) out-performs (9) for almost every K and every metric (ORtg, pace, and score potential). However, in theory, (9) should perform much better when there is a significant difference in the cardinality of the testing and training.

4.1.3 Normalizing the Data

In an effort to reduce the rank of the data, we attempt to normalize the entries of $\boldsymbol{O}$. Previously, the values of $\boldsymbol{O}$ were around 100, and its singular values ranged from $\sigma_1(\boldsymbol{O}) = 3000$ to about

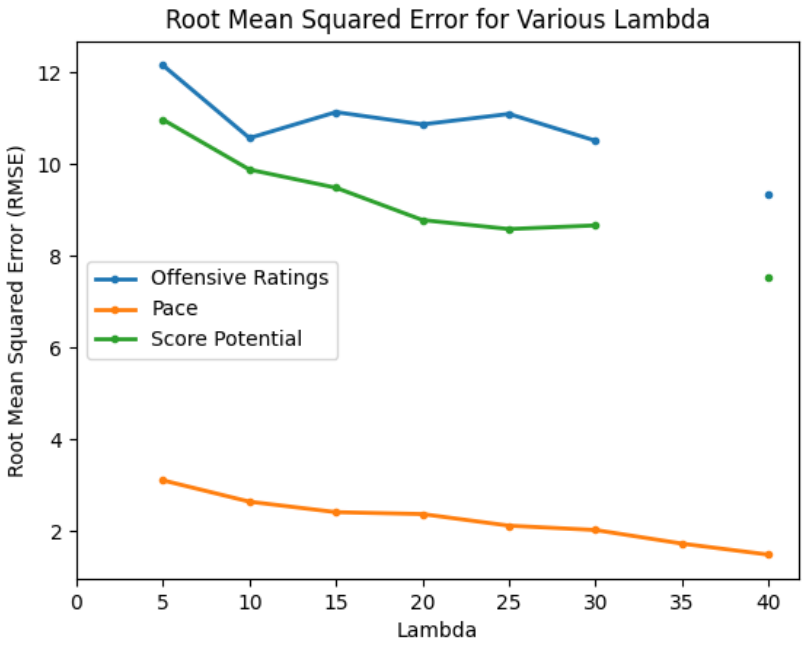


Figure 2: The difference between the RMSE of the two formulations (naive – relaxed). Note that a point above $y = 0$ indicates that naive performed better for this metric.

100. Our normalized matrix $\boldsymbol{O}_N$ had a maximum singular value of $\sigma_1(\boldsymbol{O}_N) = 22$ with most of the smaller singular values staying around 1. By normalizing the original data, we hoped to reduce the magnitude of the singular values to create an effectively low rank matrix that we could predict on. Then, if we successfully reconstructed the normalized matrix, we would simply re-scale the predicted value by our normalization factor to get the original value.

Unfortunately, our hypothesis backfired. While normalizing the original matrix did lower the singular values and did, in some sense, create a lower rank matrix which might be more plausible to predict, we weren't able to decrease the rank by enough (we wanted more singular values to be close to 0, not 1). Further, when we re-scaled our predicted values, any small error in normalized prediction was suddenly scaled up one hundred-fold, resulting in an even greater error than before, as can be seen in 4a and 4b.

4.1.4 Discussion

We suspect that our formulation of the NBA prediction problem is not inherently low-rank because, unlike the Netflix problem, the offensive side and defensive side of a basketball game does not fall into common categories. In particular, the offensive rating and the pace of a game does not depend on a few common factors. In fact, there are many "surprise" factors that are difficult to categorize, which complicates the rank of our matrix. As an example, our formulation of the problem replying on score potential does not take into account injuries, home-advantage, time of season, and improvement of teams over time.

We believe that for competitive sports, it is difficult to apply low-rank matrix completion for predicting outcomes. This is primarily because team's have an *incentive to change*. During a NBA season, teams are encouraged to improve their skills by making changes to their play-style,

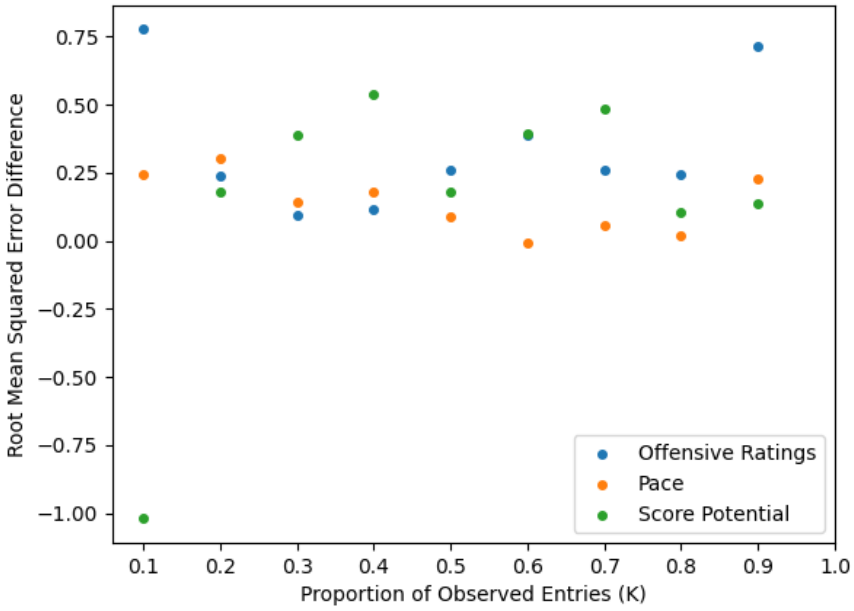


Figure 3: The RSME for various λ. Note that due to technical difficulties, we are not able to produce results for some values of λ

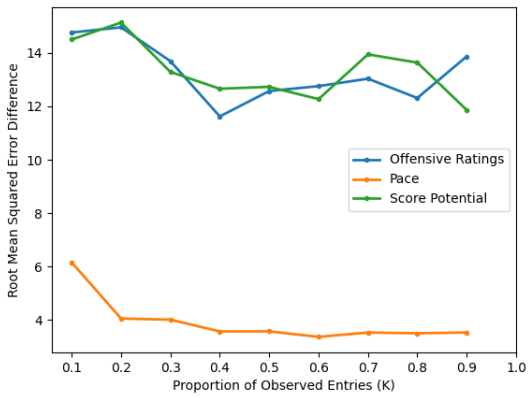
team composition, etc. This makes it especially hard to group teams into common offensive and defensive categories, as it is possible for teams to switch to a more favorable category.

If the NBA data were to be low rank, we would expect teams to be categorized into distinct offensive and defensive archetypes that have clear advantages over one another. For example, a team that relied heavily on shooting three pointers should always have an advantage over a team with bad three pointer defense. This is not what occurs in real life. Teams that are good at shooting may shoot bad some games or have a different game plan. More importantly, the teams must not change their play-style during the season.

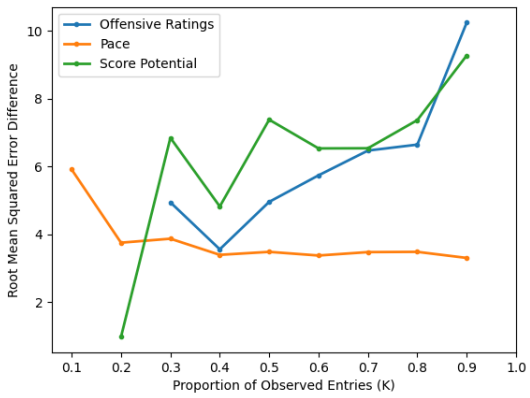
On the other hand, for data that are commonly considered low-rank such as movie preferences, social networks, medical records (Udell & Townsend, 2019), the participating parties have no incentive to change. For instance, in the Netflix problem, there is no notion of a "better" preferences for the users, and thus the lack of incentive for a user to change their preferences.

Moreover, an additional factor that is not in our favor is the relatively small data set that we have to work with. Recall that $\boldsymbol{M}$ was a 30×30 matrix, which is very small in the scheme of data sets. Udell and Townsend (2019) showed the rank of a matrix, in general, increases slower as the matrix size increases. Hence, big data sets are more likely to be of low-rank.

We are unable to show rigorously the relationship between *incentive to change* and high-rank matrices. However, this is an interesting future direction to explore as it makes claim about difficulty of prediction for all data sets where competition between parties is involved.



(a) $\hat{\boldsymbol{O}}$ and $\hat{\boldsymbol{P}}$ is computed using (6)



(b) $\hat{\boldsymbol{O}}$ and $\hat{\boldsymbol{P}}$ is computed using (9)

Figure 4: Root mean squared error (RMSE) of offensive ratings $\hat{\boldsymbol{O}}$ and pace $\hat{\boldsymbol{P}}$ with respect to normalized $\boldsymbol{O}$ and the original $\boldsymbol{P}$, respectively. $\hat{\boldsymbol{O}}$ and $\hat{\boldsymbol{P}}$ is computed using different formulation of nuclear norm minimization using the common mask Ω that samples Kn_1n_2 observations uniformly at random. The score potential is computed using (7) with input $\hat{\boldsymbol{O}}$ and $\hat{\boldsymbol{P}}$.

4.2 3D Model Completion

Inspired by applications of matrix completion for image recovery, we extend our results from matrix to 3D tensor and apply it to 3D model recovery. For a given 3D model (represented by polygon mesh or point cloud), we first compute a color voxel grid represented by three 3D tensor where each entry contains information of its RGB value (between 0 to 255) and a density voxel grid represented by one 3D tensor where each entry is either 1 or 0. Then we randomly remove part of the color voxel grid by clearing some of the RGB value to 0. Finally, we run our tensor completion algorithm on the three 3D tensor and reconstruct the model with completed color voxel grid and original density voxel grid. We analyze and compare the reconstruction results with ground truth on two different models with 30%, 50% and 80% missing data.

4.2.1 Definition and Notation

To extend from matrix to 3D tensor, we use similar definitions and notations as mentioned in Kolda and Bader (2009):

3D Rank-one Tensor A 3D Tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{M\times N\times K}$ is *rank – one* if it can be written as the outer product of three vectors, i.e.,

$$\boldsymbol{\mathcal{X}} = \mathbf{a}_1 \circ \mathbf{a}_2 \circ \mathbf{a}_3.$$

The symbol $\circ$ represents the vector outer product.

3D Tensor Rank The *rank* of a 3D tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{M\times N\times K}$, denoted by $\phi(\boldsymbol{\mathcal{X}})$, is the smallest number of rank-one tensors that sum up to $\boldsymbol{\mathcal{X}}$.

Unfortunately, unlike matrix rank, determining the rank of a 3D tensor is NP-hard (Håstad, 1990).

kth-mode unfolding Let matrix $\boldsymbol{X}_{(k)} \in \mathbb{R}^{n_k \times (\prod_{l \neq k} n_l)}$ denote the kth-mode unfolding (k=1,2,3) of a 3D tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{M \times N \times K}$.

The matrix from kth-mode unfolding of a 3D tensor is obtained by considering the kth-mode as the first dimension and collapsing the other two modes into the second dimension.

4.2.2 Nuclear Norm Extension: Low-Rank Tensor Completion

For a partially observed 3D tensor $\boldsymbol{\mathcal{M}} \in \mathbb{R}^{M \times N \times K}$, the tensor completion problem can be formulated as follows:

$$\begin{aligned} \text{minimize} \quad & \phi(\boldsymbol{\mathcal{X}}) \\ \text{subject to} \quad & \mathcal{P}_\Omega(\boldsymbol{\mathcal{X}}) = \mathcal{P}_\Omega(\boldsymbol{\mathcal{M}}) \end{aligned}$$

As mentioned previously, this problem is NP-hard. In order to make it computationally tractable, many research tried to extend the matrix nuclear norm to obtain a reasonable approximation for tensor rank. While there are a few variations of tensor nuclear norm, we select a rather straightforward approach with a weighted sum of multiple matrix nuclear norms proposed by Liu et al. (2013) and replace the problem as follows:

$$\begin{aligned} \text{minimize} \quad & \textstyle\sum_{k=1}^{3} \alpha_k \|\boldsymbol{X}_{(k)}\|_* \\ \text{subject to} \quad & \mathcal{P}_\Omega(\boldsymbol{\mathcal{X}}) = \mathcal{P}_\Omega(\boldsymbol{\mathcal{M}}), \\ & \textstyle\sum_{k=1}^{3} \alpha_k = 1 \end{aligned}$$

Since the voxel grid has the same dimensions in all three mode, we should treat all three unfoldings equally and set $\alpha_k = \frac{1}{3}$.

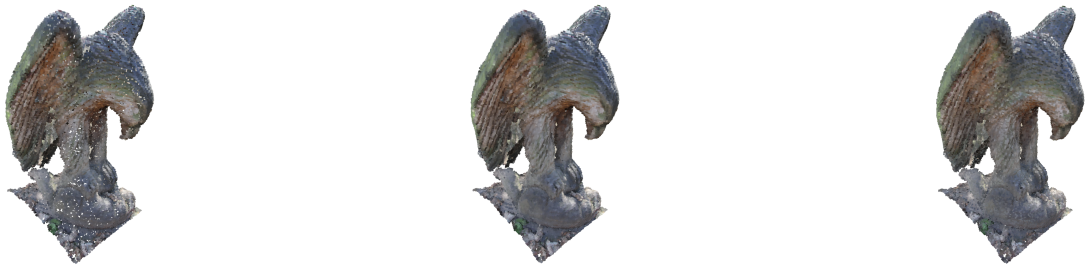
4.2.3 Results Analysis

We compare the reconstruction results with ground truth with 30%, 50%, 80% color data removed. We first use a simple banana model with 80% of its color removed to verify the feasibility of our algorithm and achieve a final RMSE of 10.82.

(a) 80% removed (b) Ground Truth (c) Recovered

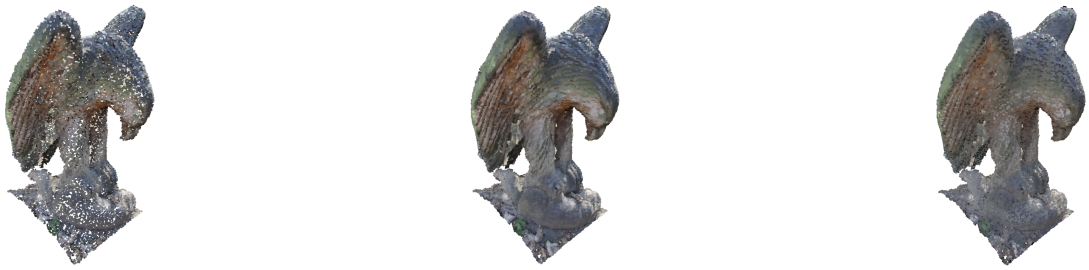
Figure 5: Reconstruction of the banana model with 80% missing color which reached a final RMSE of 10.82. The size of voxel grid is 200×200×200.

We then apply the algorithm to a much more complicated eagle model with more color details. With more missing data, the reconstructed model loses more details and tries to recover the color with larger regional patterns. The three models achieve a final RMSE of 14.50, 22.48, 36.22 respectively.



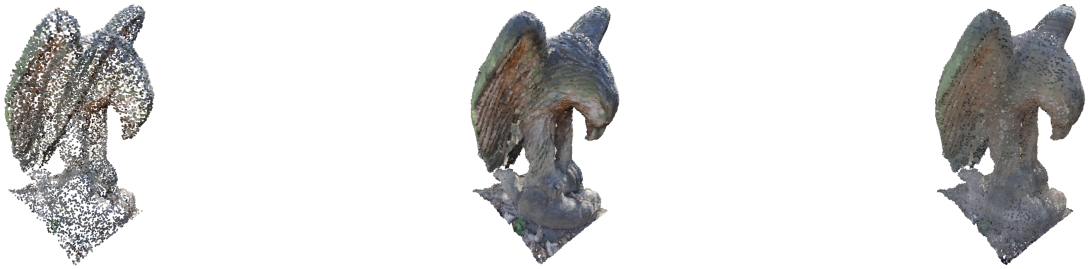
(a) 30% removed (b) Ground Truth (c) Recovered

Figure 6: Reconstruction of the eagle model with 30% missing color which reached a final RMSE of 14.50. The size of voxel grid is 200×200×200.



(a) 50% removed (b) Ground Truth (c) Recovered

Figure 7: Reconstruction of the eagle model with 50% missing color which reached a final RMSE of 22.48. The size of voxel grid is 200×200×200.



(a) 80% removed (b) Ground Truth (c) Recovered

Figure 8: Reconstruction of the eagle model with 80% missing color which reached a final RMSE of 36.22. The size of voxel grid is 200×200×200.

4.2.4 Discussion

Because we use dense voxel grid to represent finer models, the computational speed significantly slows down when we increase the size of the grid. Alternatively, if we have the mesh representation of a 3D model, we can just store the known vertex location and color in a coarser grid and use trilinear interpolation to fill in colors of unknown vertices after reconstruction.

As we can see in the reconstructed model, some have artifacts only along certain axis. A possible reason is that unfolding a 3D tensor will break the multidimensional structure of the original data and lead to degraded reconstruction result. To avoid unfolding, other definitions of tensor nuclear norm and even some nonconvex approximation for tensor rank such as truncated nuclear norm (Chen et al., 2020) or Schatten p-norm (Kong et al., 2018) can be explored and compared for this specific application.

Other than randomly removing data, more experiments can be done with non-random missing which might be more common in cases like surface erosion or overexposure during model capture. Non-random recovery should be more challenging since the missing data is more correlated and would work better when model has a clear pattern.

Although we only defined nuclear norm for 3D tensor, the approach can be extended to higher dimension tensor. One natural application might be recovering missing information for spatial data that changes over time.

References

Abadir, K. M., & Magnus, J. R. (2005). *Matrix algebra*. Cambridge University Press.

Candes, E. J., & Tao, T. (2010). The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, *56*(5), 2053–2080. https://doi.org/10.1109/TIT.2010.2044061

Candès, E. J., & Recht, B. (2009). Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, *9*(6), 717–772.

Chen, X., Yang, J., & Sun, L. (2020). A nonconvex low-rank tensor completion model for spatiotemporal traffic data imputation. *Transportation Research Part C: Emerging Technologies*, *117*, 102673. https://doi.org/10.1016/j.trc.2020.102673

Fazel, M. (2002). *Matrix rank minimization with applications* (Doctoral dissertation). PhD thesis, Stanford University.

Håstad, J. (1990). Tensor rank is np-complete. *Journal of Algorithms*, *11*(4), 644–654. https://doi.org/https://doi.org/10.1016/0196-6774(90)90014-6

Kolda, T. G., & Bader, B. W. (2009). Tensor decompositions and applications. *SIAM Review*, *51*(3), 455–500. https://doi.org/10.1137/07070111X

Kong, H., Xie, X., & Lin, Z. (2018). T-schatten-p norm for low-rank tensor recovery. *IEEE Journal of Selected Topics in Signal Processing*, *12*(6), 1405–1419. https://doi.org/10.1109/JSTSP.2018.2879185

Liu, J., Musialski, P., Wonka, P., & Ye, J. (2013). Tensor completion for estimating missing values in visual data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(1), 208–220. https://doi.org/10.1109/TPAMI.2012.39

Mazumder, R., Hastie, T., & Tibshirani, R. (2010). Spectral regularization algorithms for learning large incomplete matrices. *The Journal of Machine Learning Research*, *11*, 2287–2322.

Udell, M., & Townsend, A. (2019). Why are big data matrices approximately low rank? *Society for Industrial and Applied Mathematics*, *1*(1), 44–160. https://doi.org/10.1137/18M1183480