Abstract

A matrix network is a family of matrices, where the relationship between them is modeled as a weighted graph. Each node represents a matrix, and the weight on each edge represents the similarity between the two matrices. Suppose that we observe a few entries of each matrix with noise, and the fraction of entries we observe varies from matrix to matrix. Even worse, a subset of matrices in this family may be completely unobserved. How can we recover the entire matrix network from noisy and incomplete observations? One motivating example is the cold start problem, where we need to do inference on new users or items that come with no information. To recover this network of matrices, we propose a structural assumption that the matrix network can be approximated by generalized convolution of low rank matrices living on the same network. We propose an iterative imputation algorithm to complete the matrix network. This algorithm is efficient for large scale applications and is guaranteed to accurately recover all matrices, as long as there are enough observations accumulated over the network.

1 Introduction

In machine learning and social network problems, information is often encoded in matrix form. User profiles in social networks can be embedded into feature matrices, item profiles in recommendation systems can also be modeled as matrices. In medical imaging we often have a stack of images. These matrices have underlying connections that can come from spatial or temporal proximity, observed similarities between the items being described, etc. In this paper, we consider weighted graphs whose nodes are matrices, and we call them matrix networks.

Due to the limitations of data acquisition processes, sometimes we can only observe a subset of entries from each data matrix. The fraction of entries we observe may vary from matrix to matrix. One extreme case is that a subset of matrices is completely unobserved, namely, none of the entries is observed for these matrices.

As an example, in the following MRI image sequence (Figure 1(a)-(c)), we observe noisy entries of each frame of the MRI images. Two frames out of a total of 88 are completely unobserved, and the other 86 frames are sampled with 20% sampling rate using an i.i.d. Bernoulli distribution. All observed entries are corrupted by independent Gaussian noise. If we perform matrix completion by nuclear norm minimization on individual frames, we are not able to recover the completely unobserved matrices.

The task of recovering all matrices from partial observations, especially inferring matrices that are totally unobserved, is crucial to solve the cold start problem in networks. The cold start problem considers new items or users in a network which does not have much information available. We need to aggregate information from the network to estimate the feature matrix of the new node in order to support inference and decisions.

We make the following contributions:
1. We define the notion of matrix network and generalized convolution on matrix networks.

2. We propose a convolutional imputation algorithm to complete the matrix network from partial and noisy observations. In particular, by exploiting the network structure, the algorithm has the ability to recover matrices that are completely unobserved.

2 Related work.

The matrix completion problem is an important field of research since the series of works by Candes, Rechet, Cai et al. [4, 5], one significant achievement is establishing that $O(nr \log^2(n))$ random samples are sufficient to exactly recover a matrix of rank $r$ via nuclear norm minimization. The noisy matrix completion problem has also attracted considerable attention [5, 8, 7]. Mazumder et al. [16] proposed the soft-impute algorithm, as a scalable method to compute the nuclear norm regularized least square problem. Another class of methods for matrix completion is called the maximum margin matrix factorization methods [18], which is a non-convex problem. Despite the theoretical difficulty, its computational efficiency attracts lots of large scale applications. Hence non-convex approaches start to attract lots of attention recently [13, 14, 20]. Hastie et al. [11] propose an algorithm as an improvement of soft-impute by computing reduced SVD via alternating least squares.
Figure 2: An example of matrix network. Each node on the graph represents a matrix. Our observations are subsets of entries from these matrices.

Tensor completion is a natural generalization of matrix completion. Two of the most often used tensor factorizations/decompositions are CP decomposition and Tucker factorization. Several works define similar versions of nuclear norm for tensors as linear combinations of the nuclear norm of its unfoldings \( [10][15][21] \). Tensor completion is then again formulated as the nuclear norm regularized least square problem. \( [9] \) reports Tucker factorization of tensor with missing data outperforms nuclear norm minimization methods when the fraction of known elements of a tensor is low. Tensor completion can be treated as a special case of matrix network completion, when the network is one dimensional. However, to our knowledge, most of the work concerning tensor completion do not consider the sampling schemes where a fraction of slices of matrices in the tensor are completely unobserved.

The idea of defining graph Fourier transform and convolution is rooted in spectral graph theory. Recent advances in the field of Convolutional Neural Network tries to extend the traditional network on Euclidean grids to handle data on graphs. \( \text{Y. LeCun et al.} [2][12] \) proposed a generalization of convolutions to graphs via the graph Laplacian. Our work extends the convolution defined therein from scalar networks to matrix networks.

3 Mathematical definitions

**Matrix network** We define a matrix network as a weighted graph whose nodes are matrices of the same size. The graph has \( N \) nodes with adjacent matrix \( W \in \mathbb{R}^{N \times N} \). The nodes are indexed \( 1, 2, \ldots, N \). A function \( A : J \to \mathbb{C}^{m \times n} \) maps each node in the graph to a matrix of size \( m \times n \), where \( J = \{1, 2, \ldots, N\} \). Therefore, \( A(i) \) is the matrix at node \( i \). Figure 2 illustrates an example of matrix network.

This definition is inspired by the fiber bundle theory in differential geometry, where the graph is the underlying base space, the weight is the metric, and the matrix network is a fiber bundle on the base, with each fiber being a linear space of matrices.

We define a \( L_2 \) norm \( \| \cdot \| \) on \( A \) as \( \| A \|^2 = \sum_{i=1}^{N} \| A(i) \|_F^2 \), where \( \| \cdot \|_F \) is the Frobenius norm.

We define a unitary transform of a matrix network, as an analog of the Discrete Fourier Transform. The unitary transformation matrix \( U \) is a \( N \times N \) matrix whose columns are eigenvectors of the normalized graph Laplacian \( L = I - D^{-1/2}WD^{-1/2} \), where \( D \) is a diagonal matrix with entries \( D_{ii} = \sum_j W_{ij} \). \( U \) is often referred to as the graph Fourier transform of this weighted graph.
For a matrix network $A$, applying the unitary transformation will give $\hat{A} = UA : J \rightarrow \mathbb{C}^{m \times n}$, where $A(k) = \sum_{i \in J} U(k, i)A(i)$. The inverse transform $U^{-1} = U^*$ will transform $\hat{A}$ back to $A$, as $A(i) = \sum_{k \in J} U^*(i, k)\hat{A}(k)$. In this paper, we refer to this transformed space as spectral space.

When the graph is a single unweighted circle where each node has degree $2$, it represents a one dimensional periodic grid, $L$ is the discrete Laplacian operator, and the eigenvectors are sine and cosine functions with different frequencies. If we represent them using roots of unity, then $U$ would be the Discrete Fourier Transform (DFT) matrix $F$.

**Generalized convolution** We can extend the definition of convolution to matrix networks. For two matrix networks $X : J \rightarrow \mathbb{C}^{m \times r}$ and $Y : J \rightarrow \mathbb{C}^{r \times n}$ on the same graph, we define their generalized convolution as $X \ast Y = U^{-1}((UX)(UY))$, namely,

$$(X \ast Y)(i) = \sum_{k \in J} U^*(i, k)((UX)(k)(UY))(k).$$

Here $(UX)(k)(UY)(k)$ is a matrix multiplication for each $k \in J$. When $g$ is a scalar network, as a matrix network of size $1 \times 1$, we define $(g \ast \hat{Y})(k)$ as scalar multiplication instead.

Since $U$ is unitary transform, $\|A\| = \|UA\|$, and $\|X \ast Y\| = \|\hat{X}\hat{Y}\|$. We can define a weighted norm with respect to a scalar network $g : J \rightarrow \mathbb{C}$ as $\|A\|_g = \|g \ast A\| = \|\hat{g}A\|_g$.

When $U$ is the Discrete Fourier Transform $F$, $X \ast Y = F^{-1}((FX)(FY))$ gives an equivalent definition of circular convolution $X \ast Y(i) = F^{-1}((FX)(FY))(i) = \sum_{j=1}^N X(j)Y(i - j + 1)$.

The second equality is due to the classical convolution theorem.

### 4 Completion

Imagine that we observe a few entries $\Omega(i)$ of each matrix $A(i)$ with noise. We define the sampling rates as $p_i = \|\Omega(i)\|/(mn)$. The sampling rate varies from matrix to matrix. Even worse, a subset of matrices may be completely unobserved, namely $p_i = 0$. The projection operator $P_\Omega$ is defined to project the full matrix network to our partial observation by only retaining entries in the set $\Omega = \bigcup \Omega(i)$.

To recover missing entries, we need structural assumptions about the matrix network $A$. We propose the assumption that $A$ can be well-approximated by the generalized convolution $X \ast Y$ of two matrix networks $X, Y$ of size $m \times r$ and $r \times n$, for some $r$ much smaller than $m$ and $n$. We will show that under this assumption, accurate completion is possible even if a significant fraction of the matrices are completely unobserved.

We formally formulate the completion problem as follows. Let $A^0 = X^0 \ast Y^0$ be a matrix network of size $m \times n$, as the ground truth, where $X^0$ and $Y^0$ are matrices of size $m \times r$ and $r \times n$ on the same network. Suppose our observations $P_\Omega(A)$ are the noisy entries of $P_\Omega(A^0)$, and satisfy the Gaussian noise model $P_\Omega(A) = P_\Omega(A^0) + P_\Omega(W)$. Here each entry of $W$ is sampled i.i.d from $N(0, \sigma^2/n)$.

After the transform, $\hat{A}^0(k)$ are rank $r$ matrices, therefore we consider the following nuclear norm minimization problem, as a convex relaxation of rank minimization problem,

$$\min_{M, \hat{M}} \sum_{k=1}^N \|\hat{M}(k)\|_* \quad \text{subject to} \quad \|P_\Omega(A - M)\|_F \leq \delta, \quad \hat{M} = UM.$$ 

Here $\delta \geq 0$ is a regularization parameter controlling the tolerance in approximating observed entries. One method to solve this problem is alternating direction method of multipliers (ADMM) [1][6]. We refer reader to a detailed explanation in the appendix. If we solve this problem via second order methods, it can become prohibitively expensive for large scale applications. Equivalently we can reformulate this constrained optimization problem in Lagrange form,

$$\min_{M, \hat{M}} \frac{1}{2}\|P_\Omega(A - M)\|^2 + \sum_{k=1}^N \lambda\|\hat{M}(k)\|_*, \quad \text{subject to} \quad \hat{M} = UM.$$
Here $\lambda$ is a regularization parameter controlling the nuclear norm of the minimizer, there is a one-one mapping between $\delta$ and $\lambda$ over their active domains. For the flexibility in applications, we generalize a single $\lambda$ to a family of regularization parameter $\lambda(k)$ on the spectral space of the network, and define the objective function $F_\lambda(M)$ as $F_\lambda(M) := \frac{1}{2} |P_\Omega (A - M)|^2 + \sum_{k=1}^{N} \lambda(k) \| \hat{M}(k) \|_p$. 

We know the following relation for the nuclear norm of $\hat{M}(k) = \hat{X}(k)\hat{Y}(k)$, where $\hat{X}, \hat{Y}$ are of size $m \times r'$ and $r' \times n$, $r'$ is the rank we pick for computation such that $r' \geq r$.

$$\| \hat{M}(k) \|_p = (1/2) \minimize_{\hat{M}=X\hat{Y}} (\|\hat{X}(k)\|_{F}^2 + \|\hat{Y}(k)\|_{F}^2),$$

Using this relation, if we define a scalar network $g_\lambda$ such that the transform $\hat{g}_\lambda(k)$ satisfies $\hat{g}_\lambda^2(k) = \lambda(k)$, we can write an objective function for the convolutional factorized form $M = X \ast Y$,

$$F_\lambda(X,Y) = \|P_\Omega (A - X \ast Y)\|^2 + \|X\|_{\hat{g}_\lambda}^2 + \|Y\|_{\hat{g}_\lambda}^2.$$ 

This objective is a generalization of the maximum margin matrix factorization methods (MMMFM) [18]. When $r' = \min(m,n)$, $F_\lambda(X,Y)$ is equivalent to convex objective $F_\lambda(M)$ for $M = X \ast Y$. Since $F_\lambda(X,Y)$ is not convex in its arguments, it is hard to prove theoretical results. However, it is computationally efficient in large scale applications. In this paper, we will stick to this objective because of its clean mathematical form, and treat $r'$ as a parameter.

In the next section we will propose a convolutional imputation algorithm that effectively finds the minimizer of $F_\lambda(X,Y)$ for a sequence of regularization parameters, based on warm restarts.

## 5 Convolutional Imputation

We use an iterative imputation algorithm to complete the missing entries, inspired by the soft-impute algorithm by Rahul Mazumder, et.al[16]. The algorithm iteratively replaces the missing entries with the current estimation $X \ast Y$, and updates the estimation of $X$ and $Y$ by optimally factorizing the imputed matrix network into convolution of lowrank matrix networks.

### Convolutional factorization via SVD

We first study the optimization problem of finding $X$ and $Y$ when the matrix network is fully observed. In this case the problem is separable after the unitary transform. We can solve the problem exactly, by taking the singular value decomposition (SVD) $A(k) = P(k)\Sigma(k)Q^*(k)$.

Then the solution is computed by singular value soft-thresholding, $P(k)(\Sigma(k) - \lambda(k)I)_{+}Q^*(k)$, where $(\cdot)_{+}$ is the projection operator on the semi-definite cone. Let $\Sigma_r(k), P_r(k), Q_r(k)$ be the matrices of the top $r$ singular values and vectors, obtained by truncating $\Sigma(k)$, $P(k), Q(k)$ for all $k$.

Then $E_r(k) = (\Sigma_r(k) - \lambda(k)I_r)_{+}^{1/2}$. We get one solution of the fully-observed case as

$$\hat{X}(k) = P_r(k)E_r(k), \quad \hat{Y}(k) = E_r(k)Q^*(k).$$

The solutions $(\hat{X}(k), \hat{Y}(k))$ are not unique, they can be rotated by any unitary matrix in $U(r)$. However, the convolution $X \ast Y$ is always unique.

### Iterative imputation algorithm

The vanilla version of our imputation algorithm iteratively performs imputation of $A^{\text{impute}} = P_\Omega(A) + P_\Omega^T(A^{\text{new}})$ and singular value soft-thresholding of $\hat{A}^{\text{impute}}$ to solve the nuclear norm regularization problem.

In the imputation algorithm above, computing the full SVD on each iteration is very expensive for large matrices. We use alternating ridge regression to compute reduced-rank SVD instead, as summarized in Algorithm 2. We can choose the rank $r' \leq r$ if we know an upper bound for the rank, we will assume $r' = r$ for simplicity. This algorithm can be easily parallelized for distributed computation, since computations on different $k$ are separable.

### Regularization path

The sequence of regularization parameters is chosen such that $\lambda_1(k) > \lambda_2(k) > \ldots > \lambda_C(k)$ for each $k$. The solution $A^{\text{new}}$ for each iteration with $\lambda_s$ is a warm start for the next iteration with $\lambda_{s+1}$. One simple choice is to choose $\lambda_1(k)$ as $\lambda_{\max}(\hat{A}^{\text{impute}}(k))$, the largest singular value for $\hat{A}^{\text{impute}}(k)$, and decay $\lambda_s$ at a constant speed $\lambda_{s+1} = c\lambda_s$. 

5
We define a surrogate $Q$ Algorithm 1

Theorem 1. The sequence of solutions of the objective function $L$ for any fixed $F$ asymptotically to a minimizer of the objective function $L$. Now we show that the solution of our imputation algorithm converges to a minimizer of $L$. Convergence analysis

Algorithm 2

Convergence analysis Now we show that the solution of our imputation algorithm converges asymptotically to a minimizer of the objective $F(X, Y)$. We treat $F(X, Y)$ as function $L(\hat{X}, \hat{Y})$ of $\hat{X}, \hat{Y}$.

$$L(\hat{X}, \hat{Y}) = \|P_{\Omega}(A - U^{-1}(\hat{X}\hat{Y}))\|^2 + \sum_{k=1}^{N} \lambda(k)(\|\hat{X}(k)\|_F^2 + \|\hat{Y}(k)\|_F^2).$$

We define a surrogate $Q(\hat{X}, \hat{Y} \mid \hat{X}^{\text{old}}, \hat{Y}^{\text{old}})$ of the above objective function as

$$\|P_{\Omega}(A) + P_{\Omega}^\perp(U^{-1}(\hat{X}^{\text{old}}\hat{Y}^{\text{old}})) - U^{-1}(\hat{X}\hat{Y})\|^2 + \sum_{k=1}^{N} \lambda(k)(\|\hat{X}(k)\|_F^2 + \|\hat{Y}(k)\|_F^2).$$

For any fixed $\lambda = (\lambda(k))_{k=1}^{N}, \lambda(\lambda) \geq 0$, define a sequence $\hat{X}^t, \hat{Y}^t$ with any starting point $\hat{X}^0, \hat{Y}^0$ by

$\hat{X}^{t+1}, \hat{Y}^{t+1} = \arg\min_{\hat{X}^t, \hat{Y}^t} Q(\hat{X}, \hat{Y} \mid \hat{X}^t, \hat{Y}^t)$. The imputation algorithm produces a sequence of solutions $\hat{X}^t, \hat{Y}^t$. We can prove that $Q$ decreases after every iteration and $\hat{X}^t, \hat{Y}^t$ converges to the optimal set of solutions of the objective function $L$. The proof of this theorem is in appendix.

**Theorem 1.** The sequence $\hat{X}^t, \hat{Y}^t$ converges to the minimizer of $L$. The proof of this theorem is in appendix.

The proof of this theorem is in appendix.

**Theorem 1.** The sequence $\hat{X}^t, \hat{Y}^t$ converges to the minimizer of $L$. The proof of this theorem is in appendix.

The proof of this theorem is in appendix.
6 Guarantee for noisy recovery

Let us now analyze the central problem: what condition is needed for the non-uniform sampling Ω such that our algorithms are guaranteed to perform accurate recovery with high probability? We prove that one condition is that average sampling rate \( p = \frac{1}{N} \sum_{i=1}^{N} p_i = |Ω|/(Nmn) \) is greater than \( O\left( \frac{1}{\log^2(nN)} \right) \).

It is worth pointing out that the condition is only about the average sampling rate, therefore it includes the interesting case that a subset of matrices is under-sampled, or in extreme case, completely unobserved.

The intuition behind this phenomenon is the following: The sampling operator \( P_{13} \) acts on \( A \), and the low rank conditions hold in \( \hat{A} = UA \), as an incoherent domain. We can imagine that the unitary transform \( U \) mixes the unbalanced sampling and make it close to uniform again. Although the sampling operator \( P_{13} \) behaves badly when a subset of matrices are completely unobserved, after the transformation the randomness is averaged, and enough measurement is made for each low rank matrix in the spectral space. To make this intuition precise, we need to introduce the incoherence condition for a family of matrices. This condition will be assumed to prove guarantee for recovery.

The incoherence condition for the matrix completion problem is first introduced by Candes and Recht \(^4\) and has become a standard assumption for low-rank matrix recovery problems. Here we define average incoherence condition, which is a weaker condition than the standard incoherence condition.

**Definition 6.1.** (Average incoherent condition.) For a matrix network \( A \), we consider the SVD of \( A(k) = P(k)E(k)Q^*(k) \). We define the average leverage score for row and column of \( A \) as

\[
\mu_h = \frac{m}{N_m} \sum_{k=1}^{N} \|P^*(k)e_h\|^2, \quad \nu_l = \frac{n}{N_m} \sum_{k=1}^{N} \|Q^*(k)e_l\|,
\]

where \( e_h \) denotes the \( h \)-th standard basis with appropriate dimension. And we define the incoherence parameter \( \mu \) of \( \hat{A} \) as \( \mu = \max_h \min_l (\mu_h, \nu_l) \).

We already proved that the solutions of our imputation algorithm asymptotically convergences to the minimizer of \( F_\gamma(X, Y) \) for any \( r' \). But \( F_\gamma(X, Y) \) is not convex in its arguments, it is hard to prove theoretical results, we can only prove this theoretical guarantee for \( F_\gamma(\hat{X}, \hat{Y}) \) with \( r' = \min(m, n) \), equivalently, the convex objective \( F_\gamma(M) \) for \( M = X \star Y \). Therefore, the guarantee we prove are only valid for our first imputation algorithm, our second imputation algorithm using alternating methods will be left.

**Theorem 2.** Suppose \( A \) satisfy the average incoherent condition with parameter \( \mu \), then with appropriate choice of \( \gamma \), there exists \( C, C_p, \gamma \) such that if the average sampling rate \( p > C \mu^{-2} \log^2(N(m+n)) \), for noiseless case \( \sigma = 0 \), the minimizer \( M \) of \( F_\gamma(M) \) is the unique minimizer \( A^0 \), with probability at least \( 1 - (Nn)^{-\gamma} \), and for the noisy case with noise level \( \sigma \), this minimizer obeys

\[
\frac{1}{\sqrt{Nmn}} \|A^0 - X \star Y\| \leq C_p \sqrt{n} \sigma.
\]

Therefore, with high probability, the minimizer gives exact recovery in noiseless case and stable recovery in noisy case, where the error is proportional to the noise level \( \sigma \). The proof of this theorem is in appendix.

7 Experimental results

7.1 Synthetic feature matrices on Facebook network

We take the ego networks from the SNAP Facebook dataset\(^17\). The combined ego networks form a connected graph with 4039 nodes and 88234 edges. All edges have equal weights. We construct synthetic feature matrices on each of the nodes by randomly generating \( \hat{X}(k), \hat{Y}(k) \in \mathbb{R}^{1 \times 50} \) in the spectral domain, and doing inverse graph Fourier transform to get \( A = U^{-1}(\hat{X}(k)\hat{Y}(k)) \). The observation is generated by sampling \( N_{\text{obs}} \) matrices at sampling rate \( p \), and adding i.i.d. Gaussian noise with mean 0 and variance \( \sigma^2/50 \) to all observed entries. Here \( N_{\text{obs}} < N = 4039 \) and the other matrices are completely unobserved.
We run our iterative imputation algorithm to recover $A$ from this observation with varying parameters $N_{\text{obs}}, p$, and $\sigma$, and calculate the MSE between our estimation and the ground truth. The results are summarized in Table 1. When there is no additive noise, we can recover all the matrices very well even with only 20% of entries observed across the matrix network. If half of the matrices are fully observed and the others are unobserved, we can recover 99% of the matrices with MSE below 0.01. When there is additive noise, the MSE between reconstruction and ground truth will grow proportionally.

### 7.2 MRI dataset

We use a normal cardiac MRI dataset called CETAUTOMATIX from http://www.osirix-viewer.com/datasets/. The dataset is has 88 frames with image dimension 512 × 416. This stack of MRI images scan through a human torso, and cardiovascular structures are highlighted. We corrupt the data by adding i.i.d. Gaussian noise with $\sigma = 0.1$, and two frames completely missing. The other 86 frames are sampled with 20% sampling rate using Bernoulli independent sampling.

We try to recover the image stack from the above observations, and compare our method with two baseline methods, the nuclear norm minimization method for matrix completion on each frame, and the tensor completion with trace norm regularization method[19]. The implementation of tensor completion from the original paper converges very slowly, and cannot recover frames that are unobserved. In Figure 1 we show reconstructed images from our method and matrix completion method for three frames out of the 88 corrupted frames, where the first and third frames are observed with sampling rate $p = 20\%$, the second frame is not observed. Our convolutional imputation is able to reconstruct the unobserved frame, which is out of the scope for matrix completion. In addition, our reconstruction for partially observed frames are visually more smooth and recovers more detailed information. A quantitative comparison is given in Figure 3. The relative MSE between reconstruction and ground truth is significantly smaller from our imputation algorithm than the baseline.

### Acknowledgments

The authors are grateful to David Donoho, Corinne Horn, Madeleine Udell, Ruoyu Sun, Yiqiao Zhong and Edgar Dobriban for a number of illuminating suggestions and help.
References

[1] S. Boyd, Parikh.N, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

[2] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun. Spectral networks and locally connected networks on graphs. *arXiv.org*, May 2014.

[3] J. Cai, E. Candes, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM J. Optim.*, 20:1956–1982, 2010.

[4] E. Candes and Y. Plan. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9:717 – 772, December 2009.

[5] E. Candes and Y. Plan. Matrix completion with noise. *Proceedings of the IEEE*, 98:925 – 936, March 2010.

[6] S. Diamond, R. Takapoui, and S. Boyd. A general system for heuristic solution of convex problems over nonconvex sets. *Working draft*, January 2016.

[7] D. Donoho and M. Gavish. The optimal hard threshold for singular values is $4\sqrt{3}$. *arXiv.org*, May 2013.

[8] D. Donoho and M. Gavish. Minimax risk of matrix denoising by singular value thresholding. *Annals of Statistics*, 42(6):2413–2440, June 2014.

[9] M. Filipovic and A. Jukic. Tucker factorization with missing data with application to low-n-rank tensor completion. *Multidimensional Systems and Signal Processing*, 26:677–692, 2015.

[10] S. Gandy, B. Recht, and I. Yamada. Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Problems*, 2011.

[11] T. Hastie, Mazumder.R, J. Lee, and R. Zadeh. Matrix completion and low-rank svd via fast alternating least squares. *Journal of Machine Learning Research*, May 2015.

[12] M. Henaff, J. Bruna, and Y. LeCun. Deep convolutional networks on graph-structured data. *arXiv.org*, June 2015.

[13] R. Keshavan, A. Montanari, and S. Oh. Matrix completion from noisy entries. *Journal of Machine Learning Research*, 11:2057–2078, 2010.

[14] R. H. Keshavan, A. Montanari, and S. Oh. Matrix completion from a few entries. *IEEE Trans. Inform. Theory*, 56:2980–2998, June 2010.

[15] J. Liu, P. Musialski, P. Wonka, and J. Ye. Tensor completion for estimating missing values in visual data. *ICCV*, page 2114–2121, 2009.

[16] Mazumder.R, T. Hastie, and R. Tibshirani. Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging. *SIAM Journal on Matrix Analysis and Applications*, (11):2287–2322, 2010.

[17] J. McAuley and J. Leskovec. Learning to discover social circles in ego networks. *Neural Information Processing Systems (NIPS)*, 2012.

[18] N. Srebro, J. Rennie, and T. Jaakkola. Maximum-margin matrix factorization. *Advances in Neural Information Processing Systems 17*, page 1329–1336, 2005.

[19] R. Tomioka, K. Hayashi, and H. Kashima. Estimation of low-rank tensors via convex optimization. *arXiv.org*, March 2011.

[20] M. Udell, C. Horn, R. Zadeh, and S. Boyd. Generalized low rank models. *arXiv.org*, page 342, October 2014.

[21] Z. Zhang and S. Aeron. Exact tensor completion using t-svd. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, February 2015.