Minimax Rank-1 Matrix Factorization

Julien M. Hendrickx UCLouvain
Alex Olshevsky Boston University
Venkatesh Saligrama Boston University

Abstract

We consider the problem of recovering a rank-one matrix when a perturbed subset of its entries is revealed. We propose a method based on least squares in the log-space and show its performance matches the lower bounds that we derive for this problem in the small-perturbation regime, which are related to the spectral gap of a graph representing the revealed entries. Unfortunately, we show that for larger disturbances, potentially exponentially growing errors are unavoidable for any consistent recovery method. We then propose a second algorithm relying on encoding the matrix factorization in the stationary distribution of a certain Markov chain. We show that, under the stronger assumption of known upper and lower bounds on the entries of the true matrix, this second method does not have exponential error growth for large disturbances. Both algorithms can be implemented in nearly linear time.

1 Introduction

We consider the problem of finding a rank-one approximation xy^T of a matrix $A \in \mathbb{R}^{m\times n}$ when only a subset Ω of the entries of A are revealed. We do not impose any stochastic assumptions on the support set Ω (i.e., the entries in Ω do not need to be randomly chosen) nor assume any structure on the underlying matrix A. We are looking for stable algorithms: this means that if what is revealed is not $\{A_{ij} \mid (i,j) \in \Omega\}$ but rather the perturbed entries $\{A_{ij} + \Delta_{ij}\}$, then we need to be able to bound the error in the recovered matrix as a function of the size of the perturbations $||\Delta||_F := \sum_{(i,j)\in\Omega} \Delta_{ij}^2$.

In particular, we are interested in analyzing this question in a minimax framework. We would like to understand, for a given error size $||\Delta||_F$, how large can the error in recovering A can be. Moreover, we would like to know which algorithm(s) can always guarantee this minimax level of performance. Finally, we would like to understand how these quantities depend on the support set Ω.

We will be making only the minimal requirement on the support Ω: the only condition we will impose is that the true matrix A be identifiable, meaning that it is possible in principle to complete the matrix A from the unperturbed entries $\{A_{ij} \mid (i,j) \in \Omega\}$. These conditions were worked out in Király et al. [2015], Bonald and Combes [2017], Cosse and Demanet [2017] and we describe them next.

First, it is natural to assume that A should have no zero entries. Indeed, if A has zero entries, then it may be impossible to complete A even from a very large number of unperturbed revealed entries Ω. For example, consider the case where every matrix entry except the $(1,1)$ entry is revealed, and equals zero; it is impossible to know what A_{11} is even though the set of revealed entries is only a single entry away from being complete.

Provided that A has no zero entries, a simple graph-theoretic condition exists for identifiability. Specifically, we associate the support set Ω with an undirected bipartite graph, G, with node set $\mathcal{I}_x \cup \mathcal{I}_y$, where $I_x = \{1,\dots,m\}$ and $I_y = \{1,\dots,n\}$ (recall that $A \in \mathbb{R}^{m\times n}$), with nodes $i \in I_x$ and $j \in I_y$ connected if the ijth entry is an element of Ω. If A has no zero entries, identifiability is equivalent to the connectivity of this bipartite graph (see Király et al. [2015] as well as discussion in Cosse and Demanet [2017]). *Thus, henceforth it will be standing assumptions that A has no zero entries and that G is connected.*

Our motivation stems from several practical applications ranging from worker skill estimation in crowdsourcing (Bonald and Combes [2017], Ma et al. [2018], Dawid and Skene [1979], Dalvi et al. [2013], Zhang et al. [2014]), inferring latent information from limited observations in collaborative filtering and recommender systems (Rennie and Srebro [2005]), and in other matrix completion applications such as global positioning

Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108.

and system identification (Candes and Plan [2010]), all of which can be formulated in terms of rank-1 matrix completion.

2 Related Work

Our work is broadly related to a number of other works that either utilize rank-one matrix completion in the context of crowd-sourcing, collaborative filtering or deal with low-rank matrix completion (Rennie and Srebro [2005], Candes and Plan [2010], Keshavan et al. [2010], Dalvi et al. [2013], Zhang et al. [2014], Li et al. [2016], Bonald and Combes [2017], Ge et al. [2016], Cosse and Demanet [2017], Ma et al. [2018], Kleindessner and Awasthi [2018]). Apart from Cosse and Demanet [2017], Ma et al. [2018], much of this literature assumes some form of incoherence on the matrices, a probabilistic model for Ω, or other structures on what indices of Ω are revealed. *This separates it from the present work, which does not use any of these assumptions.*

Many of the methods described in these contexts (Candes and Plan [2010], Keshavan et al. [2010], Dalvi et al. [2013], Ge et al. [2016], Li et al. [2016]) reduce to the fact that spectral decomposition is approximately preserved even though the matrices are only partially observed. Unlike these papers and like Cosse and Demanet [2017], Ma et al. [2018], we impose no such structure and so such spectral properties can no longer be leveraged for recovery.

In the unperturbed case, it is easy to complete a matrix A_{ij} from a subset of revealed entries using a "propagation" approach; this consists in fixing one entry, say $x_1 = 1$, and then solving for entries of y from the revealed entries in the first row, and then iterating this scheme as more entries are fixed. In particular, this was discussed in Bonald and Combes [2017]. Unfortunately, Cosse and Demanet [2017] points out that this technique performs very poorly in the presence of perturbations even on some very simple examples. Relatedly, Kleindessner and Awasthi [2018] also consider the possibility that the observations are not rank-one; they introduce other assumptions such as that the entries are observed at random and that various moments can be estimated among the different observed components can be estimated.

Other techniques proposed for matrix completion include nuclear norm minimization, see Candes and Plan [2010]. Unfortunately, nuclear norm minimization fails to solve our problem, as it will in almost all cases output a higher-rank matrix, even when there is no disturbance and sufficiently many entries are revealed, as shown in Cosse and Demanet [2017]. Ridge-regression based approaches have also been considered Ma et al. [2018], Mnih and Salakhutdinov [2008], and appear natural for our setting, since Tikhonov regularization typically provides stable solutions. Nevertheless, as pointed out in Cosse and Demanet [2017], even these approaches are unstable. Moreover, they require solving non-convex optimization problems with a potentially high number of local minima of the form

$$\min_{x,y} \left|\left|(xy^T - A^R)_\Omega\right|\right|_F + \lambda(||x||_2 + ||y||_2), \quad (1)$$

where Ω selects the entries for which data is available, and A^R denotes the revealed entries. More recently alternative minimization (ALM) methods, wherein the two vectors x and y are alternately updated to optimize $\left|\left|(xy^T - A^R)_\Omega\right|\right|_F$, have been proposed to handle the computational bottleneck of optimizing over low-rank matrices. Li et al. [2016], following up on a long line of works establishes recovery guarantees, under strong-coherence assumptions. As these authors point out, ALM methods leverage the key property "that the spectral property only need to hold in an average sense" to guarantee recovery of ALM based methods. In contrast to these methods, we impose no constraints either on the ground-truth matrix or assume random sampling of revealed entries.

Our work is closely related to Cosse and Demanet [2017]. Like that work, we require our algorithms be stable. On the other hand, we differ from their method in several ways. First, theirs is based on solving an SDP relaxation involving a matrix whose size grows quadratically with that of the matrix to be recovered leading in the best-case scenario to a fourth-order complexity. In addition, Cosse and Demanet [2017] provides guarantees on the relative error on a matrix of moments that is related to, but different from, the initial matrix to be recovered, and assumes knowing some bound on the magnitude of the perturbation. As we will discuss later in the paper, our methods actually run in linear time (up to log factors), and are thus considerably faster than methods based on SDP relaxation.

2.1 Our contributions

In this paper, we develop two efficient and stable approximation methods for rank-one estimation of a partially observed matrix.

Our first scheme is based on formulating the problem as weighted least squares in a certain logarithmically transformed space. *Our first contribution is to demonstrate that, for small perturbations, the performance of this scheme matches the fundamental lower bounds that we derive for this problem*, and which are related to the spectral gap of the bipartite graph G associated with the revealed entries.

Unfortunately, the recovery error of the weighted log-least squares method will scale exponentially in $||\Delta||_F$.

While this may be acceptable for small perturbations Δ, it makes the minimax performance quite poor if $||\Delta||_F$ is not small. Unfortunately, *our second contribution is to show that this is unavoidable.* Specifically, we consider the class of consistent algorithms, defined those methods that require correct recovery of A when $\Delta_{ij} = 0$ for all $(i,j) \in \Omega$. We show that any consistent scheme must suffer an estimation error that scales exponentially in $||\Delta||_F$.

This negative result leads us to consider a minor modification of our problem. Specifically, we consider the setting where we additionally know upper and lower bounds on the entries of A. We propose a method, based on the encoding of the rank-one factors x and y (from the decomposition $A = xy^T$) into a stationary distribution of a suitable Markov chain, and whose parameters leverage these known lower and upper bounds. *Our final contribution is to give an estimate of the recovery error associated with this method and show that it does not scale exponentially in* $||\Delta||_F$.

3 The first algorithm: weighted log-least squares

We begin with a heuristic derivation of our method. Let us first consider the unperturbed case, i.e., when $\Delta_{ij} = 0$ for all $(i,j) \in \Omega$.

We begin with the observation that it suffices to deal with the case where A is positive. Indeed, if $A = xy^T$ is rank-one with no zero entries, then the same holds for $|A| = |x||y^T|$. Any method which works in the positive case can be used to compute $|A|$ by taking the absolute value of the revealed entries $\{|A_{ij}| \mid (i,j) \in \Omega\}$. Having obtained the full-matrix A, we can then easily compute $|x|$ and $|y|$ by a standard rank-1 factorization. Finally, we can use "sign propagation" to figure out the sign of all the entries of x and y: we fix $\text{sign}(x_1) = +$, and repeatedly figure out the signs other elements of x and y by inspecting the revealed entries A_{ij}. It is easy to see that this process will work provided the graph G is connected: specifically, the process will result either in the recovery of x and y, or in the recovery of $-x$ and $-y$, which amounts to the same thing.

Now in the case the rank-1 matrix $A = xy^T$ is positive, it follows that x and y can be taken to be positive vectors. We define

$$\begin{aligned} z_i &= x_i \text{ for all } i \in I_x \\ z_j &= y_j^{-1} \text{ for all } j \in I_y \end{aligned}$$

In terms of these new variables we have that

$$A_{ij} = z_i/z_j \text{ for all } (i,j) \in \Omega,$$

or

$$\log A_{ij} = \log z_i - \log z_j \text{ for all } (i,j) \in \Omega. \qquad (2)$$

The assumption of positivity of A_{ij} was necessary in order to be able to take logarithm in the last equation. We now observe that these equations are linear in the quantities $\log z_i$. This leads to a natural idea: we can solve the linear system of Eq. (2) for the quantities $\log z_i$, and then find x, y by exponentiating.

We now come back to the case where the perturbations Δ_{ij} are not necessarily zero. Provided the perturbations are not so large as to change the sign of the entries, we can proceed as before by taking the absolute values of the revealed matrix and recovering the signs during post-processing using a sign-propagation step.

However, simply solving Eq. (2) approximately is no longer the best thing to do, because different entries of the matrix display different levels of sensitivity to perturbations. Indeed, observe that if we solve

$$\begin{aligned} \log z_i - \log z_j &= \log(A_{ij} + \Delta_{ij}) \\ &= \log A_{ij} + (\log(A_{ij} + \Delta_{ij}) - \log A_{ij}) \\ &:= \log A_{ij} + D_{ij}, \end{aligned}$$

we see that the same disturbance Δ_{ij} might create a larger or smaller D_{ij} depending on the matrix entry A_{ij}. Specifically, if A_{ij} is smaller, a disturbance Δ_{ij} of the same magnitude can result in a larger D_{ij}. Informally speaking, an adversary with a fixed budget for disturbances might choose to perturb smaller entries.

Our approach to deal with this is to re-weigh the equations so that an adversary could not take advantage of this, at least when the perturbations Δ_{ij} are small relative to A_{ij}. Indeed, observe that using first-order approximations

$$\begin{aligned} \frac{D_{ij}}{\Delta_{ij}} &= \frac{\log(A_{ij} + \Delta_{ij}) - \log A_{ij}}{\Delta_{ij}} \\ &\approx \frac{1}{A_{ij}} \approx \frac{1}{A_{ij} + \Delta_{ij}}, \end{aligned}$$

where the first approximation used that $(\log x)' = 1/x$ while the second one used that Δ_{ij} should be small.

This string of equations leads to a natural heuristic: we can simply multiply the equation in Eq. (2) corresponding to (i,j) by the revealed entry $A_{ij} + \Delta_{ij}$. If we do that, small perturbations Δ_{ij} will have the same effect on every equation. Thus adopting the shorthand

$$A^R_{ij} := A_{ij} + \Delta_{ij} \text{ for all } (i,j) \in \Omega,$$

where the superscript "R" stands for "revealed", our algorithm is to solve the system of equations

$$A^R_{ij}(\log z_i - \log z_j) = A^R_{ij} \log A^R_{ij}, \qquad \forall (i,j) \in \Omega \quad (3)$$

in the least square sense. Naturally, this is not the same as solving Eq. (2) in the least-squares sense, since

multiplying the (i,j)'th equation by A^R_{ij} effectively "weights" each equation in Eq. (2) differently.

We conclude this section by explaining how this system of equations is a Laplacian linear system, which allows us to leverage existing results to show it can be solved in time nearly linear in the size of Ω. We begin by introducing a particular weighted version of the bipartite graph G: G_{W^R} has the same node set and edges as G with the weight of the edge (i,j) being $(A^R_{ij})^2$. Let L_{W^R} be the Laplacian of this weighted graph, and let B be its incidence matrix. It is standard that

$$L_{W^R} = BW^RB^T, \tag{4}$$

where $W^R \in \mathbb{R}^{|\Omega|\times|\Omega|}$ a diagonal matrix collecting all the weights $(A^R_{i,j})^2$. The linear system of Eq. (3) can then be expressed as

$$(W^R)^{\frac{1}{2}}B^T \log z = (W^R)^{\frac{1}{2}} \log A^R_\Omega$$

where A^R_Ω denotes the vector collecting the revealed entries A^R_{ij}. Using Eq. (4), least squares solutions of this systems are solutions of the linear system

$$L_{W^R} \log z = BW^R \log A^R_\Omega.$$

For example, one least-squares solution is

$$\log \hat{z} = L^\dagger_{W^R} BW^R \log A^R_\Omega \tag{5}$$

where † denotes the Monroe-Penrose inverse.

Computational Efficiency. The main advantage of this rewriting is that it now follows from the now-classic results of Spielman and Teng [2014] that Eq. (5), being a system of equations with a graph Laplacian, can be solved in near linear time (up to log terms) in the size of Ω. More precisely, a solution with precision ϵ can be obtained in $O(|\Omega| \log^\kappa(n+m)\log(\epsilon^{-1}))$ operations for some constant $\kappa > 0$.

The pseudocode of the weighted log-least squares method is given below.

Algorithm 1 Weighted Log-Least Squares Method

1: **Input:** Positive revealed entries $\{A^R_{ij} \mid (i,j) \in \Omega\}$.
2: Solve Eq. (5) for $\hat{z}$.
3: For $i \in I_x$, set $\hat{x}_i = z_i$ and for $j \in I_y$, set $\hat{y}_j = z_j^{-1}$.
4: Return $\hat{A} = \hat{x}\hat{y}^T$.

3.1 Accuracy Results

We now move to a discussion of our results. Our first theorem gives an error bound for the performance of the weighted log-least squares method.

Theorem 1. *Suppose that A is positive and*

$$A_{ij} + \Delta_{ij} > 0 \text{ for all } (i,j) \in \Omega.$$

Suppose that the disturbances further satisfy

$$\Delta_{ij} \le (c-1)A_{ij} \text{ for } c > 1 \text{ and all } (i,j) \in \Omega.$$

Then the weighted-log least squares method returns an estimate $\hat{A}$ satisfying the error bound

$$\left|\left|\hat{A} - A\right|\right|^2_F \le c^2 \lambda_{\max}(K_W L^\dagger_{W^R})\, ||\Delta||^2_F\, e^{2c\sqrt{R_{W^R,\max}}||\Delta||_F},$$

where $R_{W^R,\max}$ is the largest pairwise resistance between any pair of nodes in the weighted bipartite graph G_{W^R}, and K_{W^R} is the Laplacian of the complete bipartite graph on $I_x \times I_y$ where the edge of weight (i,j) is the true entry A_{ij}.

To parse this theorem, note that the positivity of $A_{ij} + \Delta_{ij}$ effectively bounds Δ from below, while the condition $\Delta_{ij} < (c-1)A_{ij}$ bounds it from above. The latter condition is just another way of stating that the revealed entry $A_{ij} + \Delta_{ij}$ is not more than c times the actual entry A_{ij}. Finally, as discussed earlier, we can consider the positive case without loss of generality due to the trick of taking the absolute value of revealed entries and using sign propagation.

For a formal definition and discussion of the electrical resistance of graphs, we refer the reader to Levin and Peres [2017]. Informally, the resistance of a graph is defined as the largest resistance in an electrical circuits where each edge is replaced by a resistor with resistance *inversely* proportional to the weight of that edge.

The assumption that the revealed entry A^R_{ij} has the same sign as A_{ij} is unavoidable. To see why, consider the rank-1 matrix

$$A = \begin{pmatrix} \epsilon & \epsilon \\ \epsilon & \epsilon \end{pmatrix}$$

Supposing that

$$A^R = \begin{pmatrix} \epsilon^2 & -1 \\ -1 & * \end{pmatrix}$$

where the star represents unrevealed entries, we see that the revealed entries of A^R are consistent with the matrix

$$\begin{pmatrix} \epsilon \\ -\epsilon^{-1} \end{pmatrix} \begin{pmatrix} \epsilon & -\epsilon^{-1} \end{pmatrix}$$

Thus even though $||\Delta||_F$ is constant, the recovery error will scale at least as ϵ^{-2}. Choosing a sufficiently small ϵ, we can obtain an arbitrarily large error.

We note that the necessity of the same-sign assumption is not particularly dependent on the choice of method,

as this pair of matrices is a counterexample for all methods which return a rank-1 matrix completion of A^R whenever it is available (in our next section, we will prove lower bounds on the performance of such methods, which we call *consistent*).

For small $||\Delta||_F$, both the exponential factor and the constant c in Theorem 1 approach one, so that we have

$$\lim_{||\Delta||_F \to 0} \frac{||\hat{A} - A||_F^2}{||\Delta||_F^2} \leq \lambda_{\max}(K_{W^0} L^{\dagger}_{W^R}). \qquad (6)$$

Theorem 1 thus identifies the key graph-theoretic quantity that governs robustness in the small-perturbation regime. Because it may be difficult to trace how this quantity scales in the number of nodes or other graph-theoretic quantities, we provide a corollary that gives a bound in terms of the more usual graph characteristics.

Corollary 1. *Let $\bar{\alpha}$ be an upper bound on the entries of the matrix A and let $\underline{\alpha}^R$ be a lower bounds on the revealed entries A^R_{ij}. Under the same assumption as in Theorem 1, the estimate produced by the weighted log-least squares method satisfies*

$$\left|\left|\hat{A} - A\right|\right|_F^2 \leq c^2 \left(\frac{\bar{\alpha}}{\underline{\alpha}^R}\right)^2 \frac{m+n}{\lambda_2(L)} ||\Delta||_F^2 \, e^{2c\sqrt{R_{\max}}||\Delta||_F/\underline{\alpha}^R}$$
$$\leq \frac{c^2}{4} \left(\frac{\bar{\alpha}}{\underline{\alpha}^R}\right)^2 (m+n)^3 ||\Delta||_F^2 \, e^{2c\sqrt{m+n}||\Delta||_F/\underline{\alpha}^R},$$

where $R_{\max}$ is the maximal pairwise resistance in the unweighted bipartite graph G, $\lambda_2(L)$ is the second-smallest eigenvalue of the Laplacian L corresponding to this graph, and, as before, $A \in \mathbb{R}^{m \times n}$.

4 Lower bounds

It is natural to wonder to what extent the upper bounds we have derived in the previous section is optimal. The following theorem considers the limiting case when the perturbation is small. We provide a lower bound under the assumption that the algorithm only uses the revealed entries A^R_{ij} to compute an estimate $\hat{A}$. Note that this assumption applies to the weighted log-least squares method, but will be violated by the algorithm we will propose in the next section.

Theorem 2. *Consider any algorithm that computes an estimate $\hat{A}$ of A based solely on $\{A^R_{ij} \mid (i,j) \in \Omega\}$. Then for any entry-wise positive rank-1 matrix A^R and mask Ω, one can find a matrix A such that*

$$\left|\left|\hat{A} - A\right|\right|_F^2 \geq \lambda_{\max}(K_W L^{\dagger}_{W^R}) \, ||\Delta||_F^2 + o\left(||\Delta_\Omega||_F^2\right),$$

with $\Delta_{ij} = A^R_{ij} - A_{ij}$ for $(i,j) \in \Omega$.

Combining this theorem with Eq. (6), *we obtain our first main result: that the weighted log-least squares method is optimal for small disturbances, and that the relevant graph-theoretic quantity governing performance is $\lambda_{\max}(K_W L^{\dagger}_{W^R})$.*

We next turn to the question of what happens when disturbances are not small. Inspecting Theorem 1, we see that the error bound grows exponentially in the size of the disturbance $||\Delta||_F$. There is also an exponential scaling in terms of the largest resistance in the graph G with weights coming from W^R. The latter is also concerning, as resistances will often scale polynomially in the number of nodes (for example, on a line of n nodes resistance is linear in n). And since the resistance of a weighted graph scales inversely in the weights, the upper bound may also blow up for certain classes of problems where specific revealed entries go to zero.

It is natural to ask whether these poor scalings are avoidable. Unfortunately, our next main result answers this negatively under a plausible assumption.

That assumption is *consistency*, which says that when the revealed entries are the unperturbed entries of a rank-1 matrix A, the algorithm should recover A exactly. Consistency is a natural condition for algorithms that estimate A based purely on revealed entries. Looking forward, however, we note that consistency is *not* a natural assumption for algorithms that are allowed to use additional information. For example, later on in the paper we will consider algorithms that know upper and lower bounds on the entries of A. Indeed, when revealed entries of a rank-1 matrix A with the property that some entries of A lie outside of these bounds, such algorithms should not simply return A.

Our second main result, presented as the following theorem, says that, in the worst-case, any consistent method suffers from exponentially poor scaling in $||\Delta||_F$ and the largest resistance of G.

Theorem 3.
(a) Fix any positive constant c. There exists a family of matrices $A_n \in \mathbb{R}^{n \times n}$, support sets $\Omega_n \subset \{1,\dots,n\} \times \{1,\dots,n\}$, and disturbances Δ_n satisfying $||\Delta_n|| \leq c$, with uniformly bounded A^R_{ij} and A_{ij}, for which we have

$$\left|\left|\hat{A} - A\right|\right|_F^2 \geq \left(\exp\left(c\sqrt{R_{\max}\left(\frac{1}{2} - O(n^{-1/2})\right)}\right) - 1\right)^2$$

for any consistent algorithm. Here $R_{\max}$ is the largest pairwisse resistance of the (unweighted) graph G.

(b) For every even n, there exists a family of square matrices A^R, A, support sets Ω with $||\Delta||_F$, $\max A_{ij}$, $c = 1 + \max \Delta_{ij}/A_{ij}$ bounded uniformly independently

of n such that for any consistent algorithm,

$$\left|\left|\hat{A} - A\right|\right|_F^2 \geq \frac{n^2}{9}(\min A_{ij}^R)^{-2}.$$

5 The second algorithm: Markov chain stationary distributions

We now provide an algorithm that avoids the exponential scaling discussed in the previous section. This does not contradict Theorem 3 because we now assume we have lower and upper bounds on the entries of A: $\underline{\alpha} \leq (A)_{ij} \leq \overline{\alpha}$ for all $i \in I_x, j \in I_y$, and these quantities $\underline{\alpha}$ $\overline{\alpha}$ are known to the algorithm. For small disturbances, however, the guarantees on performance of this method will, in the worst-case, be weaker than the asymptotically optimal algorithm in Section 3.

In the sequel, we will find it convenient to define the quantities $\mu = \sqrt{\bar{\alpha}\underline{\alpha}}$ and $\rho = \sqrt{\bar{\alpha}/\underline{\alpha}}$, so that the interval $[\underline{\alpha}, \bar{\alpha}]$ can be re-expressed as $[\mu\rho^{-1}, \mu\rho]$.

5.1 Algorithm Description

Since we know that every $(A)_{ij}$ lies in $[\underline{\alpha}, \bar{\alpha}]$, we will begin by projecting all revealed entries on that interval. Note that this step can only reduce the disturbances.

The algorithm consists in computing the stationary distribution of a continuous time Markov chain defined on the graph G. Specifically, we define the matrix $M^R \in \mathbb{R}^{(m+n)\times(m+n)}$ as

$$\begin{aligned}
(M^R)_{ij} &= \frac{\mu}{\mu + (A^R)_{ij}} && (i,j) \in \Omega, \\
(M^R)_{ji} &= \frac{(A^R)_{ij}}{\mu + (A^R)_{ij}} && (i,j) \in \Omega, \\
(M^R)_{ii} &= -\sum_{j \in I_y} (M^R)_{ij} && i \in I_x, \\
(M^R)_{jj} &= -\sum_{i \in I_x} (M^R)_{ji} && j \in I_y,
\end{aligned}$$

and set all other entries are 0. The matrix M is defined in the same way, replacing A^R by A. For background on continuous-time Markov chains, we refer the reader to Levin and Peres [2017].

The motivation for this method is captured by the following simple observation. Recalling that $A = xy^T$, it turns out that the vector z defined as

$$z_i = \frac{x_i}{\sqrt{\mu}} \text{ for } i \in I_x, \;\; z_j = \frac{\sqrt{\mu}}{y_j} \text{ for } j \in I_y$$

is proportional to the stationary distribution of the continuous time Markov chain M. This fact can be verified immediately by observing that the "balance equations" $M_{ij}z_i = M_{ji}z_j$ hold.

In other words, in the case where the perturbations are zero, computing the stationary distribution of M immediately lets us recover x and y, and therefore the matrix A. When the perturbations are nonzero, one might hope that the stationary distribution of M^R will depend smoothly on the perturbations Δ, so that it will be possible to bound the recovery error.

This trick is very similar to the approach used by Negahban et al. [2012, 2016] for the problem of estimating an unknown set of weights from a collection of noisy pairwise comparisons. One difference is that we add a projection step; this seemingly minor difference allows us to bypass the lower bounds of Theorem 3. The pseudocode for the method is given below.

Algorithm 2 Projected Eigenvector Algorithm

1: Project all revealed entries onto $[\mu\rho^{-1}, \mu\rho]$.
2: Compute $\pi^R \in \mathbb{R}^{m+n}$, the principal left-eigenvector of M^R, normalized so that $e^T\pi^R = 1$
3: Let $\hat{\pi}$ be obtained by projecting each entry of π^R onto $[\frac{\rho^{-2}}{m+n}, \frac{\rho^2}{m+n}]$.
4: Return the matrix $\hat{A} \in \mathbb{R}^{m\times n}$ defined as $\hat{A}_{ij} = \mu^2\hat{\pi}_i/\hat{\pi}_j$.

Finally, the stationary distribution π^R is simply the eigenvector of M^R corresponding to the zero eigenvalue. It can be computed in nearly-linear time as a consequence of the recent results of Cohen et al. [2017].

5.2 Accuracy Results

The accuracy guarantee of this algorithm are naturally expressed in terms of total variation (i.e., l^1) norm rather than a quadratic loss. We thus introduce the "first-order Frobenius norm" defined as

$$||M||_{F:1} = \sum_{i,j} |M_{ij}| .$$

Note that $||M||_F \leq ||M||_{F:1}$, so any upper bound on the first-order Frobenius norm is also an upper bound on the ordinary Frobenius norm.

The error guarantee for the projected eigenvector method is given in the following theorem.

Theorem 4. *The estimate $\hat{A}$ computed by Algorithm 1 satisfies*

$$\begin{aligned}
\left|\left|\hat{A} - A\right|\right|_{F:1} &\leq 3(m+n)^2\rho^4\frac{\log\rho\sqrt{m+n}}{\lambda_2(M)}\max\left(||\Delta||_\infty, ||\Delta||_1\right) \\
&\leq 3(m+n)^{2.5}\rho^4\frac{\log\rho\sqrt{m+n}}{\lambda_2(M)}||\Delta||_F ,
\end{aligned}$$

where $\rho = \bar{\alpha}/\underline{\alpha}$ and $\lambda_2(M)$ is the second-smallest eigenvalue of M, which is real.

The previous theorem implicitly depends on the revealed submatrix A^R, since the matrix M is built from A^R. The following corollary provides a bound which depends only on the graph and not the revealed entries.

Corollary 2. *The estimate $\hat{A}$ computed by Algorithm 1 satisfies*

$$\left|\left|\hat{A} - A\right|\right|_{F:1} \leq 6(m+n)^{2.5}\rho^7 \frac{\log \rho\sqrt{m+n}}{\lambda_2(L)} \left|\left|\Delta\right|\right|_F .$$

Since it is elementary that $\lambda_2(L)^{-1}$ is at most polynomial in $m+n$ (for example, the bound $\lambda_2(L)^{-1} \leq O\left((m+n)^2\right)$ follows from Theorem 6.2 in Mohar [1991b]) we come to *our third result: the projected eigenvector method avoids the exponential scaling faced by all the consistent methods.*

6 Synthetic Experiments

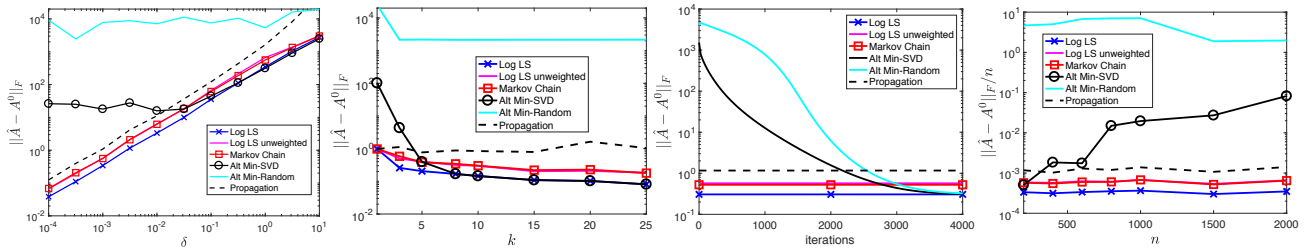
In this section, we conduct a number of synthetic experiments to highlight similarities and differences between the algorithms proposed here and prior works.

Metrics. Recall that our goal is to approximate the unknown ground-truth rank-one matrix A^0 by a rank-one matrix xy^T, when the observations are perturbed, $A^R = A_0 + \Delta$, $\frac{1}{n}\|\Delta\|_F \leq \delta$, and the revealed entries are sampled components of A^R. We will demonstrate that, under various scenarios, such as different perturbation levels (δ); different samplings of revealed entries: either random or structured; different matrix sizes, either small or large; our algorithm recovers a *stable* solution rapidly with computational time scaling with the number of revealed entries.

Competing Rank-One Approximation Methods. As pointed in our related work, nuclear norm based methods do not guarantee rank-one recovery and so we omit them in our experiments. Ridge-regression Eq. 1 is demonstrably unstable (see Appendix Fig. 4). In summary, we are left with three algorithms, propagation (Bonald and Combes [2017]), Alternative Minimization with/without clipping and with SVD and with random initialization (Li et al. [2016]), and our weighted Log-LS method, unweighted Log-LS method, and the Markov chain method, which we report here. The clipping method of Li et al. [2016] was found to be sub-optimal, and in general Alternative minimization without clipping out-performed clipping for our rank-one setting. For this reason we omitted results with clipping. We report results for Alternative Minimization as Alt-Min-SVD and Alt-Min-Rand.

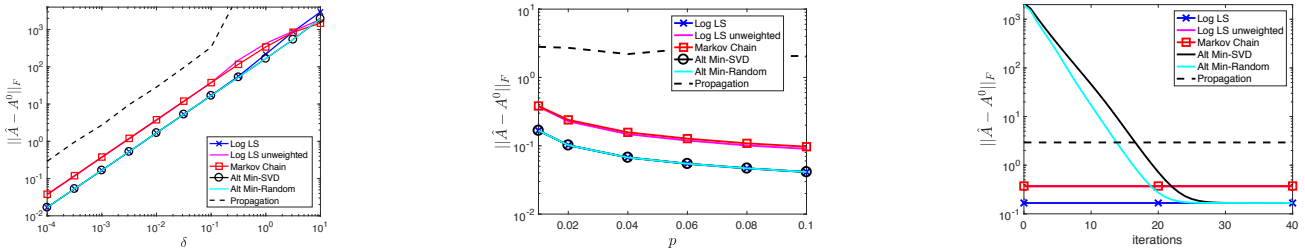
Datasets.
Ground truth matrices A^0: The matrices were generated by taking random vectors x, y, with $\log x_i, \log y_j$ uniformly distributed in $[-(\log \rho)/2, (\log \rho)/2]$, for $\rho = 10$. As a consequence, all entries lie in $[10^{-1}, 10]$. Unless stated otherwise, matrices were of size 1000×1000.
Perturbation on revealed entries (A^R, Δ). The perturbation applied to revealed entries consist of i.i.d. random variable uniformly distributed in $[-\delta/2, \delta/2]$, with $\delta = 10^{-3}$ unless stated otherwise. Other typical perturbations were not observed to lead to significantly different behaviors.
Masks and revealed entries. Two sort of masks were considered. (i) **Random mask:** each entry is revealed or not according to i.i.d. random variable of probability p, equal to 0.01 unless stated otherwise. This corresponds thus to an average of 10 revealed entries per row or column. Masks that did not lead to a connected graph G were discarded, as this is a necessary condition to reconstruct A^0 even in the absence of perturbation. (ii) **Star mask:** the first k rows and columns are revealed. Note that revealing a single row or column corresponds to a star graph, and is the minimal number of elements required to complete a rank-one matrix Ma et al. [2018]. Unless otherwise specified, we conduct experiments with 1% revealed entries, and 3-rows/columns for Star masks.

Experiments and Key Findings. All of the reported results are mean values averaged over 50 trials. We found Alt-Min-SVD and Alt-Min-Rand exhibited high variance over the course of the iterations. We report variances in the appendix (see Figure 5).
Effect of Perturbation Size. We first determine how the different methods scale with size of perturbation under different structures (random & star), with fixed number of revealed entries ($\sim 1\%$) for a matrix of size 1000×1000, for 1000 Alt-Min iterations. Figures 1(a) and 2(a) reveal that, unsurprisingly, accuracy degrades with perturbation size. For small perturbation, we notice that proposed weighted Log-LS dominates our other proposals. Nevertheless, for large perturbations, Markov-Chain error saturates and thus performs better than competing methods. Propagation performs poorly against other competing methods even with relatively small perturbation. Alt-Min and Log-LS achieve somewhat the same accuracy with 1% randomly revealed entries (see Fig. 2(a)).
Influence of Sampling. Figures 1 and 2 provide a qualitative comparison between random vs. structured star-mask sampling. Alt-Min performs well with random sampling, but its performance significantly degrades with star-masks. This points to the fact that Alt-Min performs well only when the spectral properties are preserved, see Li et al. [2016]. Propagation is somewhat robust to different types of sampling.
Effect of # Revealed Entries. We varied number of revealed entries, keeping all other variables (matrix size, perturbation level, # Alt-Min iterations) fixed for ran-



(a) Perturbation Size (b) # Revealed Columns. (c) Convergence. (d) Matrix Size.

Figure 1: Influence of "star-graph" sampling on proposed and competing methods. On Star graphs our Log-LS dominates other methods with perturbation size, # Revealed Columns, Convergence Speed, and Matrix Size. For each experiment, when one of the variables was varied (for instance perturbation) then other variables were fixed (with matrix size $n = 1000$, 3-column/row star-graph sampling, and 1000 iterations for Alt-Min methods).



(a) Perturbation Size (b) # Revealed Fraction. (c) Convergence.

Figure 2: Comparisons for ideal random sampling scenario. Matrix size is omitted as no discernible features between Alt-Min and Log-LS were found. Experiments were conducted as in Figure 1, by varying one parameter and holding other parameters fixed. Qualitatively Log-LS and Alt-Min perform similarly. In contrast to star-graph sampling Alt-Min converges rapidly to optimal solution.

dom and star-mask sampling (see Fig. 1(b) 2(b)). For star-mask we varied the number of rows/ columns. Both Alt-Min-SVD and Alt-Min-Rand initially performed worse than proposed methods. Furthermore, Alt-Min-Rand, could not stably recover the ground-truth even with sufficiently large number of revealed entries. In contrast, Alt-Min performed as well as our Log-LS method for random sampling. This is not surprising, since the assumptions for Alt-Min are satisfied.

Computational Scaling & Convergence. We refer to Sec. 3 for details on computational scaling of our proposed methods, where we claimed linear scaling with number of revealed entries. Propagation method has a similar scaling. In contrast, Alt-Min is iterative, and for each iteration, scales at least linearly with number of entries. For this reason, we also conduct experiments to compare convergence speed for the various algorithms. Fig. 1(c) reveals that Alt-Min converges relatively slowly, and exhibits high variance under star-mask sampling. It is indeed surprising that it takes over 30 iterations to converge even under the ideal random-sampling scenario (Fig. 2(c)).

Matrix Size. We also experimented with matrix size ranging from small values to 4000×4000 size matrices. Results are presented in Fig. 1(d). Surprisingly, both Alt-Min-SVD and Alt-Min-Rand degrades with size of the matrix, when all other parameters are kept constant, while proposed method and propagation are robust to matrix size.

7 Conclusions

We have presented two algorithms for rank-1 approximation based on a set of revealed entries. Both are computationally very efficient, in that a nearly linear time implementation exists. Our first method, based on weighted log least-squares, was shown to achieve the minimax bound for small disturbances. Unfortunately, it scales exponentially in the size of the disturbance for large disturbances. We have shown that this is unavoidable because any consistent algorithm has this property. Our second algorithm avoids this exponential scaling by further assuming lower and upper bounds on the matrix entries are known. However, its performance guarantees are worse for small disturbances. We conducted a number of synthetic experiments to highlight salient aspects of our method relative to competing methods. We showed that both in ideal and non-ideal sampling situations, with other varying parameters, such as matrix size and perturbation size, our method is computationally efficient and statistically stable.

Acknowledgements

This work of J. Hendrickx was supported by a *WBI.World Excellence Fellowship* and by the *Incentive Grant for Scientific Research (MIS)* "Learning from Pairwise Data" of the F.R.S.-FNRS. The work of A. Olshevsky was supported partly by the National Science Foundation Grant ECCS-1933027 and the Office of Naval Research Grant N000014-16-1-2245. The work of V. Saligrama was supported partly by grants from the National Science Foundation Grant 1527618, the Office of Naval Research Grant N0014-18-1-2257 and by a gift from the ARM corporation.

References

Arpit Agarwal, Prathamesh Patil, and Shivani Agarwal. Accelerated spectral ranking. In *International Conference on Machine Learning*, pages 70–79, 2018.

Thomas Bonald and Richard Combes. A Minimax Optimal Algorithm for Crowdsourcing. In *Neural Information Processing Systems Conference NIPS*, Los Angeles, United States, 2017.

E. J. Candes and Y. Plan. Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936, June 2010. ISSN 0018-9219. doi: 10.1109/JPROC.2009.2035722.

Michael B Cohen, Jonathan Kelner, John Peebles, Richard Peng, Anup B Rao, Aaron Sidford, and Adrian Vladu. Almost-linear-time algorithms for markov chains and new spectral primitives for directed graphs. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 410–419. ACM, 2017.

Augustin Cosse and Laurent Demanet. Stable rank one matrix completion is solved by two rounds of semidefinite programming relaxation. *arXiv preprint arXiv:1801.00368*, 2017.

Nilesh Dalvi, Anirban Dasgupta, Ravi Kumar, and Vibhor Rastogi. Aggregating crowdsourced binary ratings. In *Proceedings of the 22nd international conference on World Wide Web*, pages 285–294. ACM, 2013.

A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Applied Statistics*, 28(1):20–28, 1979.

R. Ge, J. Lee, and T. Ma. Matrix completion has no spurious local minimum. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2973–2981. Curran Associates, Inc., 2016.

Tosio Kato. *Perturbation theory for linear operators*, volume 132. Springer Science & Business Media, 2013.

R. H. Keshavan, A. Montanari, and S. Oh. Matrix completion from a few entries. *IEEE Transactions on Information Theory*, 56(6):2980–2998, June 2010. ISSN 0018-9448. doi: 10.1109/TIT.2010.2046205.

Franz J Király, Louis Theran, and Ryota Tomioka. The algebraic combinatorial approach for low-rank matrix completion. *The Journal of Machine Learning Research*, 16(1):1391–1436, 2015.

Matthaeus Kleindessner and Pranjal Awasthi. Crowdsourcing with arbitrary adversaries. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2708–2717, 2018.

David A Levin and Yuval Peres. *Markov chains and mixing times*, volume 107. American Mathematical Soc., 2017.

Yuanzhi Li, Yingyu Liang, and Andrej Risteski. Recovery guarantee of weighted low-rank approximation via alternating minimization. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, 2016.

Y. Ma, A. Olshevsky, V. Saligrama, and C. Szepesvari. Gradient descent for sparse rank-one matrix completion for crowd-sourced aggregation of sparsely interacting workers. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

Andriy Mnih and Ruslan R Salakhutdinov. Probabilistic matrix factorization. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1257–1264. Curran Associates, Inc., 2008.

Bojan Mohar. Eigenvalues, diameter, and mean distance in graphs. *Graphs and combinatorics*, 7(1): 53–64, 1991a.

Bojan Mohar. The laplacian spectrum of graphs. *Graph theory, combinatorics, and applications*, 2(871-898): 12, 1991b.

Sahand Negahban, Sewoong Oh, and Devavrat Shah. Iterative ranking from pair-wise comparisons. In *Advances in neural information processing systems*, pages 2474–2482, 2012.

Sahand Negahban, Sewoong Oh, and Devavrat Shah. Rank centrality: Ranking from pairwise comparisons. *Operations Research*, 65(1):266–287, 2016.

Jasson D. M. Rennie and Nathan Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the 22Nd International Conference on Machine Learning*, ICML '05, pages 713–719, New York, NY, USA, 2005. ACM. ISBN 1-59593-180-5. doi: 10.1145/1102351.1102441. URL `http://doi.acm.org/10.1145/1102351.1102441`.

Daniel A Spielman and Shang-Hua Teng. Nearly linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems. *SIAM Journal on Matrix Analysis and Applications*, 35(3): 835–885, 2014.

Nisheeth K Vishnoi. $Lx = b$. *Foundations and Trends in Theoretical Computer Science*, 8(1–2):1–141, 2013.

Y. Zhang, X. Chen, D. Zhou, and M.I. Jordan. Spectral methods meet EM: A provably optimal algorithm for crowdsourcing. In *NIPS*, pages 1260–1268, 2014.