LOW-RANK MATRIX RECOVERY FROM ONE-BIT COMPARISON INFORMATION

Arindam Bose, Aria Ameri, Matthew Klug, and Mojtaba Soltanalian*

Department of Electrical and Computer Engineering
University of Illinois at Chicago
Chicago, USA

ABSTRACT

In this paper, we study the problem of low-rank matrix recovery based on the information obtained by comparing matrix entries (where each comparison is represented by one-bit) and not the entries themselves. This is highly relevant in the context of recommendation systems, due to the fact that users (particularly those less familiar with the rating system) are more comfortable with comparing products than giving exact ratings. We investigate when and how a low-rank matrix (such as a rating matrix in the recommendation system) can be efficiently recovered using one-bit data, particularly by establishing the limitations of such a recovery. We devise a computational approach based on matrix factorization to accomplish the reconstruction task. The numerical examples exhibit the significant potential of the proposed approach in low-rank matrix recovery from one-bit comparison information.

Index Terms— one-bit sampling, matrix completion, low-rank matrix, matrix factorization, recommendation system

1. INTRODUCTION

The important problem of reconstruction of a matrix from an incomplete set of samples or measurements, particularly known as *matrix completion* [1–3], arises in a large area of applications including recommendation schemes, system identification, sensor network localization, collaborative filtering, quantum state tomography, and many others [4–8]. Owing to the recent developments in the field of compressed sensing, researchers now have a powerful theoretical base in matrix completion [9–12]. In particular, we now have the necessary tools to exactly and efficiently recover (with high probability) a generic $n \times n$ matrix with rank r from $\mathcal{O}(nr\text{polylog}(n))$ randomly selected measurements even in cases of noisy observations. However, there exists a fundamental gap between the problem statement of matrix completion theory and common practical applications. For example, in the "Netflix challenge", researchers were asked to reconstruct a non-complete low-rank matrix (i.e. the rating matrix) by predicting the missing elements [13]. In this challenge, while a subset of users' ratings is observed, the observations are highly quantized to the set of integers between 1 and 5 whereas the theory of matrix completion generally assumes that observations are continuous-valued. Note that although existing techniques can be utilized with discrete-valued observations by treating them as continuous-valued, we need to account for the "quantization noise" that may not be properly modeled as additive noise in some settings. Also note that, the low-rank assumption is not only key to the identifiability of the matrix and its recovery, but also fits strongly to the real world problems.

New advances in signal recovery from extremely quantized one-bit samples reveal a significant potential for one-bit measurements in matrix recovery [9, 12, 14–17]. Recent works in one-bit matrix completion literature have taken a probabilistic model into consideration for the observation matrix, where a subset of one-bit measurements of the rating matrix itself is observed [17, 18]. By resorting to a maximum likelihood (ML) formulation, several upper bounds for matrix estimation error were derived under the assumption that the matrix entries are drawn from a uniform distribution [17], or a non-uniform distribution [18].

Note that matrix recovery based on comparisons between ratings is a very natural approach in recommendation scenarios including that of the Netflix challenge. Due to the fact that users (particularly those less familiar with the rating system) are more comfortable with comparing products than giving exact ratings, such an approach is expected to make the user interface of the rating system more friendly, and even in some cases, make the ratings more precise. In this paper, we investigate the low-rank matrix recovery through the lens of such recommendation systems. The goal is to devise a reconstruction algorithm that operates only by exploiting *rating comparison information* (where each comparison is represented by one-bit) and not the actual ratings. A brief study of the recovery limitations will be presented.

The remainder of the paper is organized as follows. Section 2 presents the problem formulations and the recovery approach. Section 3 discusses the limitations of low-rank matrix recovery when only a *small error* is tolerated. Numerical results are presented in Section 4. Finally, Section 5 concludes the paper.

* Corresponding author (e-mail: *abose4@uic.edu*). This work was supported in part by U.S. National Science Foundation Grant CCF-1704401.

Notation: We use bold lowercase letters for vectors and uppercase letters for matrices. $(\cdot)^T$ denotes the vector/matrix transpose. $(\cdot)^\dagger$ denotes the Moore-Penrose pseudoinverse of the matrix argument. $\|\boldsymbol{x}\|_n$ or the l_n-norm of the vector $\boldsymbol{x}$ is defined as $\{\sum_k |x_k|^n\}^{\frac{1}{n}}$ where x_k is the k-th entry of $\boldsymbol{x}$. $\|A\|_F$ or the Frobenius norm of matrix A with entries $\{a_{i,j}\}_{i=1,j=1}^{m,n}$ is defined as $\sqrt{\sum_{i=1}^m \sum_{j=1}^n |a_{i,j}|^2}$. Finally, the symbol $\otimes$ stands for the Kronecker product of matrices.

2. THE RECOVERY APPROACH

Consider a $c \times p$ rating matrix M with $[M]_{i,j} = m_{i,j}$, rank r, and c and p denoting the number of users and number of products, respectively. Note that $m_{i,j}$ (bounded in the interval $[0, \eta]$) defines how much the user i likes a product j. Let $\text{vec}(M)$ denote the matrix M in its vectorized form. Comparisons are formulated as they are, in effect, observing M through an *observation* or a *measurement* matrix. We form the one-bit observation matrix $A \in \{-1, 0, 1\}^{d \times cp}$ with each row representing a comparison (thus d denotes the number of comparisons made). As a result of comparisons, the vector $\text{sgn}(A \cdot \text{vec}(M))$ will be acquired. We can thus form a matrix Ω as follows, that contains the comparison information:

$$\Omega = \text{Diag}(\text{sgn}(A \cdot \text{vec}(M))). \tag{1}$$

Consequently, the problem of recovering the rating matrix M can be formulated as

$$\begin{array}{ll} \text{recover} & M \\ \text{subject to} & \Omega \cdot A \cdot \text{vec}(M) \geq 0, \\ & \text{rank}(M) \leq r, \\ & 0 \leq \text{vec}(M) \leq \eta. \end{array} \tag{2}$$

Therefore, given the highly incomplete comparison information (namely the comparison matrix A and comparison outcome Ω), we aim to identify the matrix M with *small error* provided that the rank of M is relatively small.

2.1. The Proposed Algorithm

Inspired by the classical approaches to low-rank matrix estimation, we start with the optimization problem:

$$\begin{array}{ll} \min_{M_0} & \|M - M_0\|_F^2 \\ \text{subject to} & \text{rank}(M_0) \leq r, \\ & \Omega \cdot A \cdot \text{vec}(M) \geq 0, \\ & 0 < \text{vec}(M) < \eta. \end{array} \tag{3}$$

From an algorithmic point of view, one can reconstruct a low-rank matrix via alternating projections on matrix spaces with dimensions significantly reduced due to low-rank characteristic of the original matrix (note that storage becomes an issue when the rating matrix is large). Since we expect the rating matrix to have a small rank, we can always formulate M as $M = XY^T$ and perform the optimization over the two tall matrices X and Y of size $c \times r$ and $p \times r$ respectively, rather than the $c \times p$ matrix M, where usually $r \ll \min(c, p)$. The matrix recovery problem can thus be rewritten as

$$\begin{array}{ll} \min_{M,X,Y} & \|M - XY^T\|_F \\ \text{subject to} & \Omega \cdot A \cdot \text{vec}(M) \geq 0, \\ & 0 < \text{vec}(M) < \eta. \end{array} \tag{4}$$

The optimization problem stated above can be efficiently tackled by resorting to a cyclic minimization algorithm operating over its three optimization variables. In particular, the optimization problem with respect to the variable M is essentially a convex linearly-constrained quadratic program (QP), leading to a low-cost solution. On the other hand, the minimizers X and Y of (4) can be obtained analytically. One can easily verify that

$$\begin{aligned} \left\|M - XY^T\right\|_F &= \left\|\text{vec}(M) - \text{vec}(XY^T)\right\|_F \\ &= \left\|\text{vec}(M) - (Y \otimes I)\,\text{vec}(X)\right\|_F, \end{aligned} \tag{5}$$

which yields the optimal X of (4) as

$$\text{vec}(X) = (Y \otimes I)^\dagger\,\text{vec}(M). \tag{6}$$

Since $\left\|M - XY^T\right\|_F = \left\|M^T - YX^T\right\|_F$, one can conclude by symmetry that

$$\text{vec}(Y) = (X \otimes I)^\dagger\,\text{vec}(M^T) \tag{7}$$

is the minimizer Y of (4). Note that such an approach guarantees a convergence in the objective of (4) as (i) each step of the cyclic process decreases the objective, and that (ii) the objective is bounded from below.

3. IDENTIFIABILITY AND RANK DETERMINATION

Herein, we briefly investigate the fundamental limits of one-bit low-rank matrix recovery when the one-bit data are acquired by comparison. Such limitations clearly exist. For example, a scaling of the matrix does not change the comparison information, and as a result, the matrix can be obtained only up to a scalar. This issue, however, can be mitigated by having at least one rating value available. In the following, we derive a *rank-quantization bottleneck* from an *information-theoretic* viewpoint.

3.1. The Rank-Quantization Bottleneck

Let M be a rank-r matrix, represented as

$$M = \boldsymbol{x}_1\boldsymbol{y}_1^T + \boldsymbol{x}_2\boldsymbol{y}_2^T + \cdots + \boldsymbol{x}_r\boldsymbol{y}_r^T \tag{8}$$

where $\boldsymbol{x}_k \in \mathbb{R}^c$ and $\boldsymbol{y}_k \in \mathbb{R}^p$. We assume that the entries of $\{\boldsymbol{x}_k\}$ and $\{\boldsymbol{y}_k\}$ are stored via a q-bit quantization system with a predefined set of elements and a cardinality of 2^q. Note that the number of bits required to store M is given by cpq, while storing $\{\boldsymbol{x}_k\}$ and $\{\boldsymbol{y}_k\}$ will require $r(c+p)q$ bits. Storing the latter is beneficial if $r \ll \dfrac{cp}{c+p}$ which is easily satisfied in large-scale scenarios.

Moreover, as $r(c+p)q$ bits are required to store a large rating matrix in general, one will need at least the same number of bits from an alternative representation scheme such as the proposed method of comparisons. In other words, we need at least $r(c+p)q$ meaningful comparisons to recover M. A comparison is meaningful if it provides new information; for example, given $\alpha > \beta$ and $\beta > \gamma$, the comparison $\alpha > \gamma$ cannot be considered meaningful for our recovery goal. On the other hand, in the most efficient comparison scheme (which leads to a directed tree of comparisons, formed on the entries of M), we have exactly $cp - 1$ meaningful comparisons — the most one can wish for. Therefore, $r(c+p)q \leq cp - 1$ or equivalently,

$$rq \leq \frac{cp-1}{c+p}. \tag{9}$$

The above inequality represents a rank-quantization bottleneck on the recovery of M, i.e. the product of rank and quantization depth is upper bounded in a recovery that is done solely by using comparisons. Interestingly, this should not be a strong bottleneck for large-scale rating matrices. On the other hand, smaller rating matrices can be used to verify if a developed matrix recovery algorithm can approach such a fundamental limit, which can possibly be a good sign for the large-scale applications as well.

Further, the quantization-depth q gives a lower bound of error for representing M in worst-case scenarios. Let

$$\epsilon_q \triangleq \max_{\substack{M \\ 0 \leq m_{ij} \leq \eta}} \min_{\substack{\boldsymbol{x}_k \in \mathcal{S}_q^c \\ \boldsymbol{y}_k \in \mathcal{S}_q^p}} \left\| \sum_{k=1}^{r} \boldsymbol{x}_k \boldsymbol{y}_k^T - M \right\|_F \tag{10}$$

where $\mathcal{S}_q$ is the set of predefined elements in the quantization system. Particularly, one can observe that ϵ_q denotes the worst-case error in recovery of the rating matrices inherent to the quantization system. As a result, using (10) one can directly translate a limit on q to a limit on the guarantee we can provide on error-bound in the recovery of M.

Note that the number of meaningful comparisons c_m must satisfy the inequality $c_m \geq r(c+p)q$, which might alternatively imply that in an effective system of low-rank matrix recovery based on comparisons, for any new additional row or column in the matrix, we will need at least rq new comparisons, to fully update it. However, having only a few rating levels (e.g. $0, 1, \cdots, 7$ for $q = 3$) can help us significantly to reduce the number of required comparisons. Interesting enough, this is a very common approach (e.g., it is widely practiced in recommendation systems).

3.2. The Rank Determination Bound

In general, the low-rank matrix recovery algorithms will be much more effective if an initial good estimate of the matrix rank is available. Fortunately, an educated guess of the matrix rank might be obtainable by further research using the following road map: Assume the rating matrix M is of rank r. As a result, any generic row m of M is given as a linear combination of at most r vectors $\{m_k\}$:

$$m = \sum_{k=1}^{r} \alpha_k m_k. \tag{11}$$

The data provide comparisons of different entries of m which can finally (or at the best performance of the system) lead to an *ordering* of the elements in m. Now, the fundamental question which naturally arises is that how many possible orderings can appear in m if it is a product of (11), i.e. with M being rank r. Note that the ordering of entries in m depends mainly on the orderings that are available in $\{m_k\}$.

The above observations pave the way to establish that, the number of feasible orderings for m is, in fact, upper bounded by a function of r, that is considerably smaller than $n!$ (i.e. the number of all orderings). Particularly, one can show that the number of such orderings is bounded as $\mathcal{O}(n^{2r})$. A sketch of the proof is as follows: (i) We assume without loss of generality that $\alpha_k \geq 0$, turning the linear combination to a conic combination. The assumption $\alpha_k \geq 0$ is natural from a practical standpoint for rating matrices. However, the equivalence with the general case can be established by concatenating the set $\{m_k\}$ with $\{-m_k\}$, which brings the number of vectors in the combination to $2r$. (ii) Note that scaling does not change the ordering of entries in a vector. With this observation, we can limit the discussion to the set of convex combinations of $\{m_k\}$, i.e. with $\sum_{k=1}^{r} \alpha_k = 1$. (iii) For the case of $r = 2$, one can easily show that the number of feasible orderings is upper bounded by $1 + \binom{n}{2}$. To see how, note that any change of ordering in the vector $m = (1-\alpha)m_1 + \alpha m_2$ occurs when two entries of the vector become identical, which also can occur only once. As a result, by starting from m_1 with $\alpha = 0$ and then increasing α, the number of new orderings is upper bounded by the number of *pairs* that are available in the entries of the vector, namely $\binom{n}{2}$, which concludes the analysis. (iv) We further use this result to show that in the linear space produced by (11), any line between two points (shown as a convex combination of the associated vectors) can *pass through* at most $1 + \binom{n}{2}$, or $\mathcal{O}(n^2)$, orderings. (v) We have now achieved a Steiner-type [19, 20] partitioning of space that allows for counting the orderings. The upper bound on the number of orderings thus grows exponentially with dimension—leading to the bound $\mathcal{O}(n^{2r})$.

Note that such a bound will help with determining a lower bound for r. In the next section, we will show that the rating matrix can be efficiently recovered using the computational approach proposed in the paper.

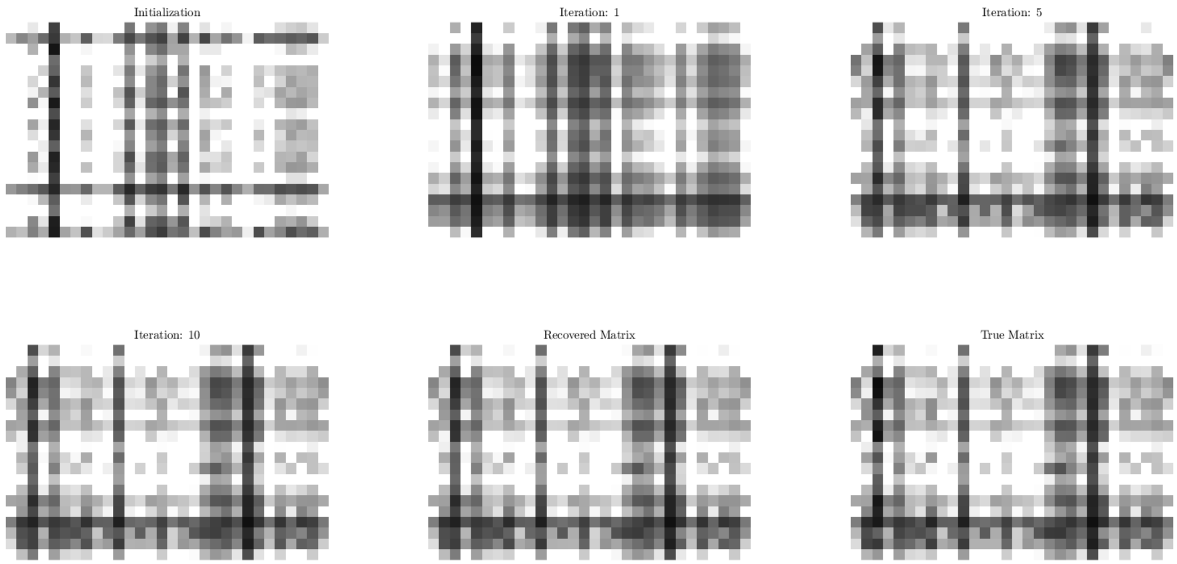

Fig. 1. An example of low-rank matrix recovery based on one-bit comparison measurements with $(c, p, r) = (15, 20, 3)$.

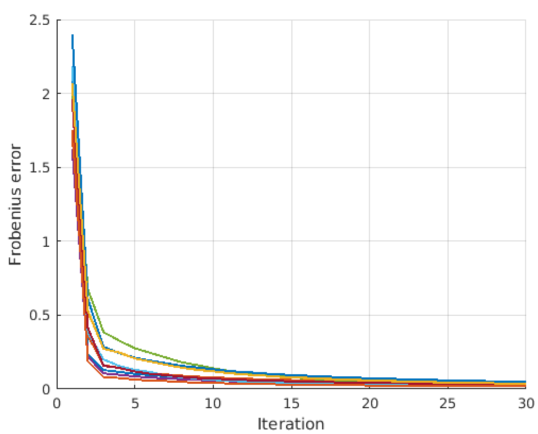

Fig. 2. Frobenius error of the low-rank matrix recovery (vs. iteration number) for different random initializations with $(c, p, r) = (15, 20, 3)$.

4. NUMERICAL EXAMPLES

In this section, we report the simulation results for low-rank matrix recovery based on one-bit comparison measurements. We tackle the recovery problem (2) by using the cyclic algorithm and the matrix factorization formulation in (4).

To show that matrix recovery from comparison information is an achievable goal, we first consider the reconstruction of a rank-3 target rating matrix M with $c = 20$ and $p = 30$. More specifically, the rating matrix M is constructed at random by generating $M = RS^T$, where R and S are of size 20×3 and 30×3 with i.i.d entries drawn uniformly from the interval [0,1]. The matrix is then scaled such that its largest entry becomes equal to one. The matrices A and Ω are formed as described in Section 2, assuming that all comparison data are available. We continue the iterations of the cyclic algorithm until convergence. Fig. 1 depicts the gradual recovery of the rating matrix after various iterations along with the true matrix M. It can be observed that the final matrix is essentially recovered from the given one-bit comparison measurements.

Next, we conduct the same experiment using different random initializations to examine the behavior of the optimization objective in (4). Fig. 2 illustrates the (relative) Frobenius error, defined as $\|M - \widehat{M}\|_F^2 / \|M\|_F^2$, where $\widehat{M}$ is the current estimate of M. The monotonically decreasing behavior of the objective is repeated for various initializations. Moreover, even with different starting points, the error appears to converge to relatively small values.

5. CONCLUSION

The problem of low-rank matrix recovery using comparison information was studied, and several results concerning the identifiability of such matrices were presented. A cyclic algorithm was devised to facilitate the recovery in such settings. Moreover, the proposed algorithm was successfully employed in our numerical investigations. The numerical examples show that the original rating matrix can be efficiently recovered even with different starting points.

6. REFERENCES

[1] J. F. Cai, E. J. Candès, and Z. Shen, "A Singular Value Thresholding Algorithm for Matrix Completion," *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, Mar. 2010.

[2] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, Aug. 2009.

[3] S. Bhojanapalli and P. Jain, "Universal matrix completion," in *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, 2014, ICML'14, pp. II–1881–II–1889.

[4] A. Anandkumar, R. Ge, and M. Janzamin, "Guaranteed Non-Orthogonal Tensor Decomposition via Alternating Rank-1 Updates," *ArXiv e-prints*, Feb. 2014.

[5] H. Sedghi and A. Anandkumar, "Provable Methods for Training Neural Networks with Sparse Connectivity," *ArXiv e-prints*, Dec. 2014.

[6] A. Anandkumar, R. Ge, D. Hsu, S. M. Kakade, and M. Telgarsky, "Tensor decompositions for learning latent variable models," *Journal of Machine Learning Research*, vol. 15, pp. 2773–2832, 2014.

[7] N. Srebro, J. Rennie, and T. S. Jaakkola, "Maximum-margin matrix factorization," in *Advances in Neural Information Processing Systems 17*, L. K. Saul, Y. Weiss, and L. Bottou, Eds., pp. 1329–1336. MIT Press, 2005.

[8] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th International Conference on World Wide Web*, New York, NY, USA, 2001, WWW '01, pp. 285–295, ACM.

[9] A. Ai, A. Lapanowski, Y. Plan, and R. Vershynin, "One-bit compressed sensing with non-gaussian measurements," *Linear Algebra and its Applications*, vol. 441, pp. 222 – 239, 2014, Special Issue on Sparse Approximate Solution of Linear Systems.

[10] P. T. Boufounos and R. G. Baraniuk, "1-bit compressive sensing," in *2008 42nd Annual Conference on Information Sciences and Systems*, Mar 2008, pp. 16–21.

[11] A. Zymnis, S. Boyd, and E. Candes, "Compressed sensing with quantized measurements," *IEEE Signal Processing Letters*, vol. 17, no. 2, pp. 149–152, Feb 2010.

[12] Y. Plan and R. Vershynin, "Robust 1-bit compressed sensing and sparse logistic regression: A convex programming approach," *IEEE Transactions on Information Theory*, vol. 59, no. 1, pp. 482–494, Jan 2013.

[13] J. Bennett and S. Lanning, "The Netflix Prize," in *Proceedings of KDD Cup and Workshop 2007*, San Jose, CA, USA, Aug 2007.

[14] C. Gianelli, L. Xu, J. Li, and P. Stoica, "One-bit compressive sampling with time-varying thresholds: Maximum likelihood and the Cramér-Rao bound," in *50th Asilomar Conference on Signals, Systems and Computers, ACSSC 2016, Pacific Grove, CA, USA, November 6-9, 2016*, 2016, pp. 399–403.

[15] C. Gianelli, L. Xu, J. Li, and P. Stoica, "One-bit compressive sampling with time-varying thresholds for sparse parameter estimation," in *2016 IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM), Rio de Janerio, Brazil, July 10-13, 2016*, 2016, pp. 1–5.

[16] S. A. Bhaskar and A. Javanmard, "1-bit matrix completion under exact low-rank constraint," in *2015 49th Annual Conference on Information Sciences and Systems (CISS)*, Mar 2015, pp. 1–6.

[17] M. A. Davenport, Y. Plan, and M. V. D. Berg, E.and Wootters, "1-bit matrix completion," *Information and Inference: A Journal of the IMA*, vol. 3, no. 3, pp. 189–223, 2014.

[18] T. Cai and W. X. Zhou, "A max-norm constrained minimization approach to 1-bit matrix completion," *J. Mach. Learn. Res.*, vol. 14, no. 1, pp. 3619–3647, Dec. 2013.

[19] E. J. Cockayne, "On the Steiner problem," *Canadian Mathematical Bulletin*, vol. 10, no. 3, pp. 431–450, 1967.

[20] J. E. Wetzel, "On the division of the plane by lines," *The American Mathematical Monthly*, vol. 85, no. 8, pp. 647–656, 1978.