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

1. Motivation

- The problem of reconstruction of a matrix from an incomplete set of samples or measurements, particularly known as *matrix completion* , arises in a large area of applications including recommendation schemes, sensor network localization, collaborative filtering, quantum state tomography [1-2] etc.
- Matrix recovery based on comparisons between ratings is a very natural approach in recommendation scenarios as users are more comfortable with comparing products than giving exact ratings [3].

2. Problem Formulation

Consider a $c \times p$ rating matrix $\boldsymbol{M}$ with $[\boldsymbol{M}]_{i,j} = m_{i,j}$ with rank r and c and p denoting the number of users and the number of items, respectively. We do not observe the matrix $\boldsymbol{M}$. However, we observe a set of triplets: $\{c^{(i)}, p^{(j)}, p^{(k)}\}$ which simply illustrates a comparison: $c^{(i)}$th user prefers $p^{(j)}$th item over $p^{(k)}$th item.

We then form the one-bit observation matrix $\boldsymbol{A} \in \{-1, 0, 1\}^{d \times cp}$ with each of its row being a comparison.

For example:

$$\boldsymbol{M} = \begin{array}{c} \begin{matrix} p^{(1)} & p^{(2)} & p^{(3)} & p^{(4)} \end{matrix} \\ \begin{bmatrix} 3 & 4 & 2 & 3 \\ 4 & 5 & 3 & 2 \\ 3 & 5 & 5 & 4 \end{bmatrix} \end{array} \begin{matrix} c^{(1)} \\ c^{(2)} \\ c^{(3)} \end{matrix}$$

$$vec(\boldsymbol{M}) = [3, 4, 3 \vdots 4, 5, 5 \vdots 2, 3, 5 \vdots 3, 2, 4]^T$$

However, suppose we only have access to some comparison information:

$\{c^{(1)}, p^{(2)}, p^{(1)}\}, \{c^{(2)}, p^{(1)}, p^{(3)}\}, \{c^{(2)}, p^{(1)}, p^{(4)}\}, \{c^{(3)}, p^{(4)}, p^{(3)}\}$

We can formulate the comparison matrix $\boldsymbol{A}$:

$$\boldsymbol{A} = \left[\begin{array}{ccc|ccc|ccc|ccc} 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 \end{array}\right]$$

and, the one-bit comparison data $\boldsymbol{\lambda}$ to be:

$$\boldsymbol{\lambda} = sgn(\boldsymbol{A} \cdot vec(\boldsymbol{M})) = [-1 \quad 1 \quad 1 \quad -1]^T$$

We define $\boldsymbol{\Omega}$ to be the diagonalized matrix of $\boldsymbol{\lambda}$; i.e.

$$\boldsymbol{\Omega} = \mathrm{Diag}(\boldsymbol{\lambda})$$

Hence the problem of recovering the ranking matrix $\boldsymbol{M}$ will reduce to:

$$\begin{aligned} \text{recover} \quad & \boldsymbol{M} \\ \text{s. t.} \quad & \boldsymbol{\Omega} \cdot \boldsymbol{A} \cdot vec(\boldsymbol{M}) \geq 0 \\ & rank(\boldsymbol{M}) \leq r \\ & 0 \leq vec(\boldsymbol{M}) \leq \eta \end{aligned}$$

Goal: To identify the low-rank matrix $\boldsymbol{M}$, given the matrices $\boldsymbol{A}$ and $\boldsymbol{\Omega}$.

3. Proposed Approach

We expect the rating matrix to have a small rank which is a very common practice in collaborative filtering for practical reasons, we can formulate $\boldsymbol{M}$ as $\boldsymbol{M} = \boldsymbol{X}\boldsymbol{Y}^T$ and perform the alternating optimization over two tall matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$ of size $c \times r$ and $p \times r$, respectively.

Thus, the matrix recovery problem can be rewritten as:

$$\begin{aligned} \min_{\boldsymbol{M}, \boldsymbol{X}, \boldsymbol{Y}} \quad & \|\boldsymbol{M} - \boldsymbol{X}\boldsymbol{Y}^T\|_F^2 \\ \text{s.t.} \quad & \boldsymbol{\Omega} \cdot \mathbf{A} \cdot vec(\boldsymbol{M}) \geq 0 \\ & 0 < vec(\boldsymbol{M}) < \eta \end{aligned}$$

which can be efficiently tackled by resorting to a cyclic minimization algorithm. The optimization problem with respect to the variable $\boldsymbol{M}$ is essentially a convex linearly-constrained quadratic program (QP), leading to a low-cost solution. Moreover, the minimizers $\boldsymbol{X}$ and $\boldsymbol{Y}$ can be obtained analytically.

$$\|\boldsymbol{M} - \boldsymbol{X}\boldsymbol{Y}^T\|_F^2 = \|vec(\boldsymbol{M}) - (\boldsymbol{Y} \otimes \boldsymbol{I}) vec(\boldsymbol{X})\|_F^2$$

which yields the optimal $\boldsymbol{X}$ and $\boldsymbol{Y}$ to be:

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

4. The Rank-Quantization Bottleneck

$$\boldsymbol{M} = \boldsymbol{x}_1 \boldsymbol{y}_1^T + \boldsymbol{x}_2 \boldsymbol{y}_2^T + \ldots + \boldsymbol{x}_r\, \boldsymbol{y}_r^T$$

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. As $r(c+p)q$ bits are required to store a large rating matrix in general, we need at least $r(c+p)q$ meaningful comparisons to recover $\boldsymbol{M}$. The bottleneck is:

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

5. The Rank Determination Bound

The low-rank matrix recovery algorithms will be much more effective if an initial good estimate of the matrix rank is available. Any generic row $\boldsymbol{m}$ of $\boldsymbol{M}$ is given as a linear combination of at most r vectors $\{\boldsymbol{m}_k\}$.

$$\boldsymbol{m} = \sum_{k=1}^{r} \alpha_k \boldsymbol{m}_k$$

The data provide comparisons of different entries of $\boldsymbol{m}$ which can finally (or at the best performance of the system) lead to an *ordering* of the elements in $\boldsymbol{m}$. The number of such orderings is bounded as $\mathcal{O}(n^{2r})$ which is considerably smaller than $n!$. Such a bound will help with determining a lower bound for r.

4. Results

We first consider the reconstruction of a rank-3 target rating matrix $\boldsymbol{M}$ with $c = 20$ and $p = 30$. The matrix $\boldsymbol{M}$ is generated randomly and normalized.

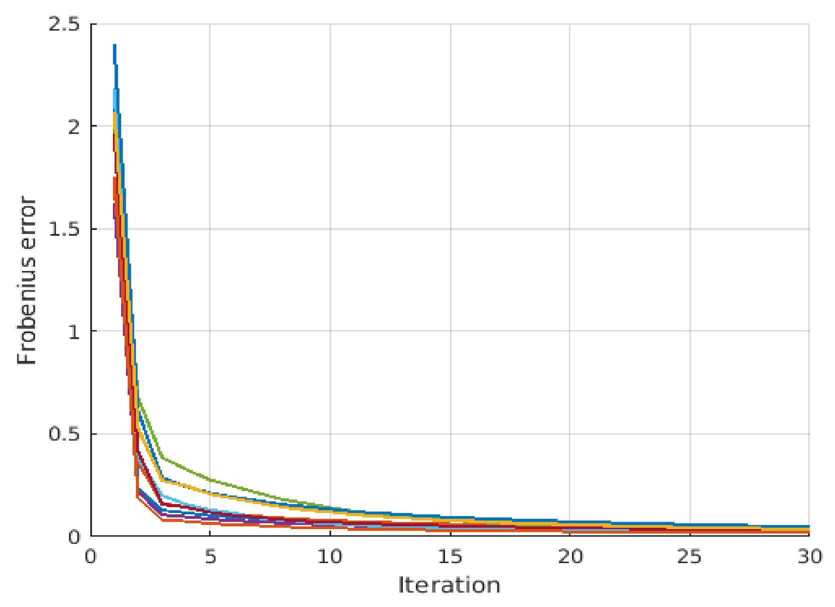

Fig. 1: Normalized Frobenius error of the low-rank matrix recovery vs. the iteration number for different random initializations with $(c;\, p;\, r) = (15;\, 20;\, 3)$

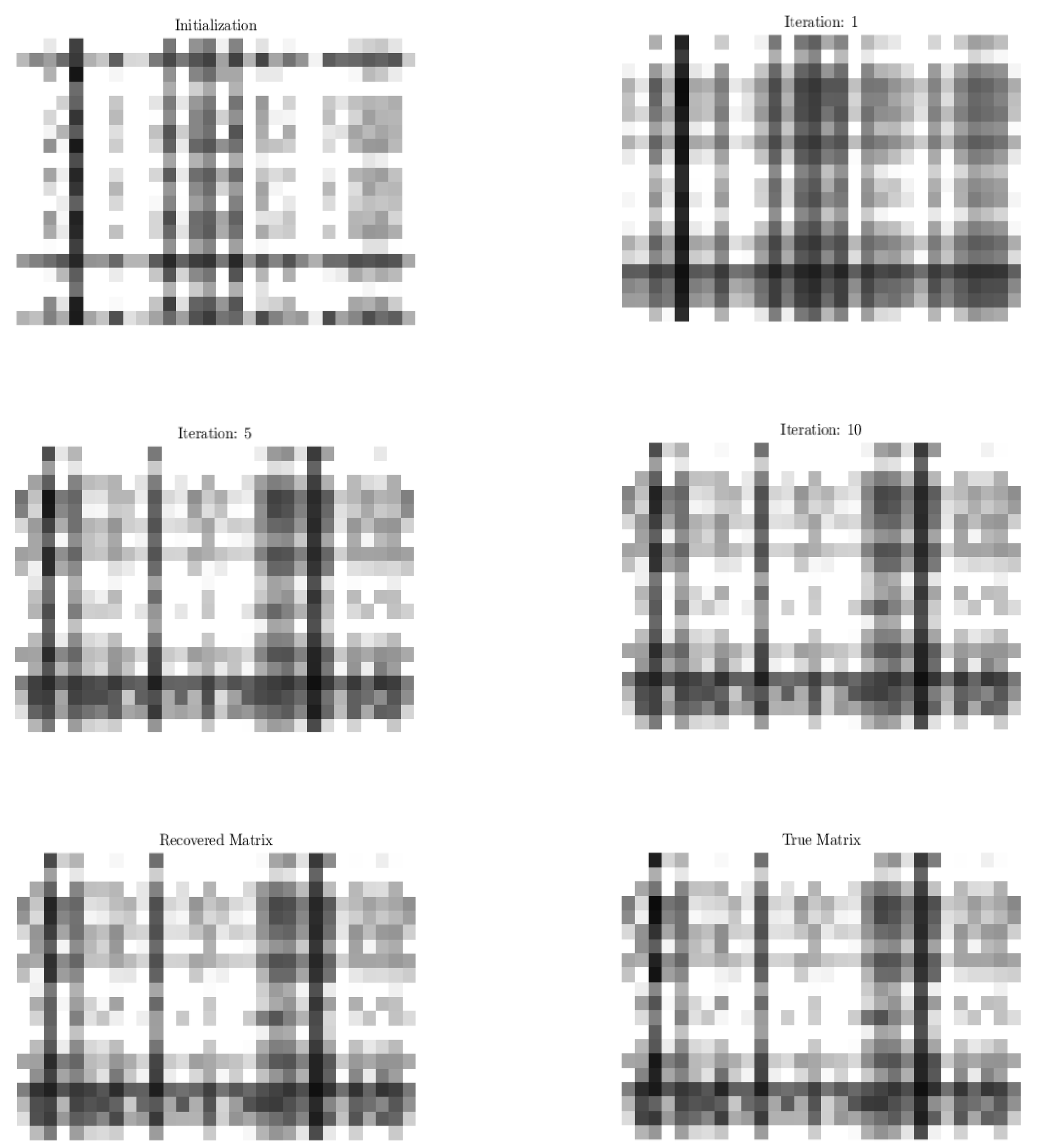

Fig. 2: An example of low-rank matrix recovery based on one-bit comparison measurements with $(c;\, p;\, r) = (15;\, 20;\, 3)$

5. References

[1] J. F. Cai, E. J. Candes, 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] 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.

[3] Y. Lu, and S. N. Negahban. “Individualized rank aggregation using nuclear norm regularization.” 2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton) (2015): 1473-1479.