Ranking Recovery from Limited Comparisons using Low-Rank Matrix Completion

Tal Levy\textsuperscript{a}, Alireza Vahid\textsuperscript{b}, and Raja Giryes\textsuperscript{a}

\textsuperscript{a}School of Electrical Engineering, Tel-Aviv University
\textsuperscript{b}Electrical Engineering Department, University of Colorado Denver

Abstract—This paper proposes a new method for solving the well-known rank aggregation problem from pairwise comparisons using the method of low rank matrix completion. The partial and noisy data of pairwise comparisons is transformed into a matrix form. We then use tools from matrix completion, which has served as a major component in the low-rank completion solution of the Netflix challenge, to construct the preference of the different objects. In our approach, the data of multiple solutions of the Netflix challenge, to construct the preference into a matrix form. We then use tools from matrix completion, partial and noisy data of pairwise comparisons is transformed using the method of low rank matrix completion. The well-known rank aggregation problem from pairwise comparisons has a known rank of one. An alternating minimization algorithm, in which the target matrix takes a bilinear form, is then used in combination with maximum likelihood estimation for both factors. The reconstructed matrix is used to obtain the true underlying preference intensity. This work demonstrates the improvement of our proposed algorithm over the current state-of-the-art in both simulated scenarios and real data.

I. INTRODUCTION

The problem of rank aggregation is common in a wide variety of tasks such as recommendation systems \cite{2}, crowd sourcing \cite{3}, ranking of chess players or online gamers (e.g. MSR’s TrueSkill system) and many more. In most scenarios, the preference of each object (rating) is of interest as well as the global ranking of objects: Understanding the intensity of object preferences allows us to make predictions under the assumption that the preferences do not change dramatically over time. In some scenarios, however, we are only given partial information over a collection of objects. Moreover, this information can be inconsistent due to noise. A common example of the problem is a small dataset of noisy pairwise comparisons from which the preferences needs to be inferred.

As the problem of group ranking in the presence of only partial pair-wise comparisons appears in many applications, it is of great importance to understand the reciprocal relations between pairs that have rare or no direct interaction between them. To do so, we draw a novel link between the problem of ranking and matrix completion that allows using tools from the latter to solve problems in the former with better accuracy compared to other solutions. We demonstrate our approach on various problems including the ranking of national soccer teams showing its advantage over other approaches.

This paper addresses this challenge by introducing a new algorithm based on low-rank matrix completion \cite{11} in an effort to reconstruct the preference intensity. The framework of low-rank matrix completion has many powerful methods proposed for exact reconstruction from few entries \cite{4}, \cite{5}, \cite{7}, \cite{8}, \cite{9} based on convex relaxation \cite{6}, \cite{10} even when the entries are corrupted by noise \cite{11}. The performance of our algorithm is tested on a popular pairwise preference-based model, Bradley-Terry-Luce \cite{12}, and is compared with the current state-of-the-art techniques. For further analysis, data from weather readings is used to evaluate the error on a simple pairwise partial dataset. To conclude, the method is examined on a non-trivial (complicated model) data of soccer scores from FIFA world cup, UEFA Euro and the Olympic games to create a ranking, which is then shown to be better than FIFA’s men ranking in the prediction of future results as shown in Section VII-D.

We present here a timely ranking before the 2018 FIFA World Cup tournament. Figure 4 presents our current ranking of the top national soccer teams including all the qualified teams to the 2018 tournament. This estimation is based on 7.5 years data of all the matches between international teams up to April 2018. We used the FIFA top 100 teams as the basis for the ranking. Therefore, teams that were not ranked in the top 100 teams by FIFA on 12.04.2018 were not included. A detailed comparison to the FIFA men’s ranking appears hereafter in Section VII-D.

II. THE RANKING PROBLEM

Consider the problem of rank aggregation as a simple tournament (without ties), where in each match the players (or teams) compete until one wins. Aggregating over several past matches, part of which may repeat, provides multiple comparisons between only a subset of the possible pairs. Assuming there is a latent preference score to the players, our task is therefore to recover a consistent ordering of all players based on the partially revealed comparison data. Our assumption is based on the Bradley-Terry-Luce model (BTL) that postulates a set of latent scores underlying all items, where the odds of paired comparisons depend only on the relative scores of the players (or teams) involved.

The above problem can be described by the following model. Assume (without loss of generality) the following set of preference scores

\[ \omega_1 \geq \omega_2 \geq \ldots \geq \omega_n > 0, \]
Fig. 1. Ranking based on all match results up to April 2018 in the past 7.5 years. In bold: Teams that qualified to 2018 FIFA World Cup.

and a given edge set for a comparison graph:

\[ i \& j \text{ are compared} \iff (i, j) \in \mathcal{E}, \]

where an edge is contained in the edge set \( \mathcal{E} \) with some probability \( p_{\text{obs}} \). For each edge in this set, we observe \( L \) repeated comparisons. In the BTL, model the \( l \)th comparison between items \( i \) and \( j \), denoted by \( y_{ij}^{(l)} \), is

\[
y_{ij}^{(l)} = \begin{cases} 
1 & : \text{ w.p. } p_{i/j} = \frac{\omega_i}{\omega_i + \omega_j} \\
0 & : \text{ otherwise }
\end{cases},
\]

where \( y_{ij}^{(l)} \) equals 1 indicates a win for item \( i \) over \( j \) in the \( l \)th match. In this model, it is assumed that a match result is binary, where each item either wins or loses a specific match, thus, \( y_{ij}^{(l)} = 1 - y_{ji}^{(l)} \). A naive estimator for the probability \( p_{i/j} \) can be obtained by:

\[
y_{ij} = \frac{1}{L} \sum_{l=1}^{L} y_{ij}^{(l)}. \tag{2}
\]

It is clear that this is an unbiased estimator that converges to \( p_{i/j} \) as \( L \to \infty \). Throughout this work we will assume that the graph of the match graph describing the comparisons (representing the edge set \( \mathcal{E} \)) obeys an Erdős Rényi model \( G(n, p_{\text{obs}}) \), where the graph is constructed by randomly connecting nodes, with each edge having probability \( p_{\text{obs}} \) of appearing in the graph independently of the other edges.

III. LOW RANK MATRIX COMPLETION (LRMC) APPROACH

By defining the following ratio estimator, the original problem may take a matrix form:

\[
R_{i/j} = \frac{1}{y_{j/i}} - 1. \tag{3}
\]

Notice that

\[
\lim_{L \to \infty} R_{i/j} = \frac{\omega_i + \omega_j}{\omega_j} - 1 = \frac{\omega_i}{\omega_j}.
\]
We define the ratio matrix $M$ in the following way:

$$M_{ij} = \begin{cases} R_{i,j} & : (i, j) \in \mathcal{E} \\ 0 & : \text{otherwise} \end{cases} \quad . \quad (4)$$

Note that all its diagonal entries are equal to one and $M_{ij} = 1/M_{ji}$ for $(i, j) \in \mathcal{E}$. In the case $L \to \infty$ we get the noiseless and complete matrix $\hat{M}$, which is rank-1 and can be constructed as:

$$\hat{M} = \tilde{\omega}^T \times \frac{\tilde{I}}{\omega},$$

$$\tilde{\omega} = (\omega_1, \omega_2, \ldots, \omega_n),$$

$$\frac{\tilde{I}}{\omega} = \left( \frac{1}{\omega_1}, \frac{1}{\omega_2}, \ldots, \frac{1}{\omega_n} \right).$$

Clearly, a reconstruction of $\hat{M}$ leads also to a recovery of the ranking $\tilde{\omega}$. Thus, the original problem can now be formulated as recovering $\hat{M}$ from a partial and noisy $M$, as this allows us to find the latent preferences $\tilde{\omega}$. A suitable framework that takes advantage of the structure of the ratio matrix $M$ to solve this problem is matrix completion.

Low rank matrix completion (LRMC) is the problem of completing a partial matrix using the lowest rank matrix fitting the observed items. For a partially observed matrix $M$ and an observed edge set $\mathcal{E}$ the problem can be formalized as:

$$\text{Minimize : } \text{Rank}(X)$$

$$\text{Subject to: } X_{ij} = M_{ij} : (i, j) \in \mathcal{E}$$

This is typically a non-convex problem and difficult to solve. When the true rank is known, a simpler problem to solve is:

$$\text{Find } X \text{ s.t. } \text{Rank}(X) = r$$

$$\text{Subject to: } X_{ij} = M_{ij} : (i, j) \in \mathcal{E}$$

Since the target matrix $X \in \mathbb{R}^{n \times n}$ is of known rank $r$, it can be written in a bilinear form, which will later prove to be more suitable to solve.

The matrix $X$ can be parametrized in the following form:

$$X = UV^T,$$

where $U \in \mathbb{R}^{m \times r}$ and $V \in \mathbb{R}^{n \times r}$. This parameterization is common and can be found for example in sparse PCA [13] and clustering [14]. In the rank recovery problem this form is advantageous since a solution of the form $V = 1/U^T$ is optimal and the rank $r$ of the matrix $M$ is equal to one.

The presented definition of the LRMC in the noiseless case for a known rank requires equality to known entries. For the ranking problem this translates to recovering $\hat{M}$ from its partial version. However, when $L$ is finite we have in $M$ a noisy version of the entries of $\hat{M}$. In this case a weaker condition (that is better suited for noisy data) needs to be defined in a sense of minimal error rather than equality. This is achieved by demanding a minimal Frobenius norm on the residual matrix (of observed entries) instead of the original equality. By defining the operator $P_\mathcal{E}$

$$P_\mathcal{E}(X)_{ij} = \begin{cases} X_{ij} & : (i, j) \in \mathcal{E} \\ 0 & : \text{Else} \end{cases} \quad . \quad (5)$$

$\hat{M}$ may be reconstructed by solving the following problem

$$\min_{U, V} \|P_\mathcal{E}(UV^T) - P_\mathcal{E}(M)\|_F^2.$$  

However, this results in a non-convex problem in general.

A popular approach to solve [3] has been to alternately keep such a low-rank projection in the entries of $\hat{M}$ and some desired resolution for the weights estimation $\omega$. For the rank-1 case, equation (8) simply becomes:

$$\hat{V}^{(t)} = \min_{V} \|P_\mathcal{E}(\hat{U}^{(t-1)}V^T) - P_\mathcal{E}(M)\|_F^2 : \text{Given } \hat{U}^{(t-1)},$$

$$\hat{U}^{(t)} = \min_{U} \|P_\mathcal{E}(\hat{U}V^{(t)}\hat{V}) - P_\mathcal{E}(M)\|_F^2 : \text{Given } \hat{V}^{(t)}.$$  

A good way to initialize this process (finding $\hat{U}^{(0)}$) is to take the top-$r$ left singular vectors of $\frac{1}{\rho_{\text{obs}}} P_\mathcal{E}(M)$ using SVD [11].

A solution to the minimization problem in (6) may be found by first defining the “row-wise” operator:

$$P_\mathcal{E}^{(s)}(X)_{ij} = \begin{cases} X_{ij} & : (i, s) \in \mathcal{E} \\ 0 & : \text{Else} \end{cases}$$

followed by computing the following $r \times r$ matrix:

$$\hat{I}^{(s)} = P_\mathcal{E}^{(s)}(\hat{U})^T P_\mathcal{E}^{(s)}(\hat{U}) \in \mathbb{R}^{r \times r}.$$  

Thus, the solution to (6) can be calculated by

$$\hat{V}^{(t)}_s = (\hat{I}^{(s)})^{-1} \hat{U}^T P_\mathcal{E}(M)_{ls}, \quad (8)$$

which has a complexity of $O(|\mathcal{E}| \cdot r^2 + n \cdot r^3)$ for $l \in [1, r]$ and $s \in [1, n]$. For the rank-1 case, equation (8) simply becomes:

$$\hat{V}^{(t)}_s = \frac{\sum_{i,s} M_{i,s} \cdot U_i}{\sum_{i,s} U_i^2}.$$

IV. ALGORITHM FOR PARTIAL NOISELESS DATA

For the case of $L \to \infty$ all the non-zero entries of the matrix $M$ obtained using (3) & (4) are noiseless (identical to $\hat{M}$). Defining the observation matrix $Y$ as $Y_{ij} = y_{i,j}$, we present in Algorithm 1 a strategy to recover the ranking from $Y$, the edge set $\mathcal{E}$, the maximal weights ratio $R_{\text{Max}}$ and some desired resolution for the weights estimation $\Delta \omega_{\text{Min}}$.

Algorithm 1 Noisless LRMC Ranking

1: procedure RANKINGMC($P_\mathcal{E}(Y), \mathcal{E}, R_{\text{Max}}, \Delta \omega_{\text{Min}}$)

2: Set: $M_{ij} = (1/Y_{ij} - 1)$

3: Set: $M_{ii} = 1$

4: Set: $T = \text{round}(4 \cdot \ln(n/2 \cdot \Delta \omega_{\text{Min}})/\ln(16))$

5: Set: $\hat{U}^{(0)}$ as the top left singular vector of $\frac{1}{\rho_{\text{obs}}} P_\mathcal{E}(M)$

6: Clipping: set $\hat{U}^{(0)}$ where $|\hat{U}^{(0)}_{ij}| \geq 2 \cdot R_{\text{Max}}$ to zero

7: Normalization: normalize $\hat{U}^{(0)}$ to $\hat{U}^{(0)}/||\hat{U}^{(0)}||_2$

8: for $t = 1 \to T$ do

9: $\hat{V}^{(t)} \leftarrow \arg \min_V \|P_\mathcal{E}(\hat{U}^{(t-1)} V^T - M)\|_F^2$

10: $\hat{U}^{(t)} \leftarrow \arg \min_U \|P_\mathcal{E}(U \hat{V}^{(t)} - M)\|_F^2$

11: $\hat{U}^{(t)} = \hat{U}^{(t)}/\max_i (\hat{U}^{(t)})$  

12: return $\hat{U}^{(t)}$


This algorithm is based on the method presented in [1]. Following the steps in the proof of theorem 2.5 in [1], we get that with probability at least $1 - n^{-\gamma}$ for a sampling probability ($p_{obs}$) obeying:

$$p_{obs} \geq \frac{8}{3} \cdot (\gamma + 1) \cdot R_{\text{Max}}^2 \cdot \frac{\ln(n)}{n} \cdot \ln \left( \frac{n}{2 \cdot \Delta \omega_{\text{Min}}} \right) \cdot \delta_2^{-2},$$

(10)

for some constant $\delta_2 \leq 1/12$, we have:

$$\max(|U - \hat{U}(\text{Finall})|) \leq \Delta \omega_{\text{Min}},$$

(11)

V. RANKING USING LIMITED COMPARISONS

The above algorithm performs well when $L \to \infty$. However, for a finite set of comparisons $L$ it encounters a few problems. The first is that we may get zero values in the observation matrix $Y$, which will lead to infinite values in the matrix $M$. To solve this problem we need to truncate the values of $M$. For that purpose we use either an estimation of $R_{\text{Max}}$ or the actual value of $R_{\text{Max}}$, if it is known, to limit the value of $M$. Defining the minimal value of $y_{i/j}$ as

$$y_{\text{Min}} = \frac{1}{1 + C_R \cdot R_{\text{Max}}},$$

the truncated observation is defined as

$$\hat{y}_{i/j} = \begin{cases} y_{i/j} : y_{i/j} \geq y_{\text{Min}} \\ y_{\text{Min}} : \text{Else} \end{cases}.$$

(13)

now we can define the truncated ratio matrix $M$ using

$$M_{i/j} = \begin{cases} \frac{1}{\hat{y}_{i/j}} - 1 : (i, j) \in \mathcal{E} \\ 0 : \text{Else} \end{cases},$$

(14)

where the constant $C_R \geq 1$ and we use $R_{\text{Max}}$ for the given or estimated value of $R_{\text{Max}}$. Note that the maximal value of any entry in the matrix $M$ is now $C_R \cdot R_{\text{Max}}$. Though the largest value in the true matrix $M$ cannot exceed $R_{\text{Max}}$, in $M$ we may have several entries grater than $R_{\text{Max}}$. Because their order contains some information, we do not truncate exactly at $R_{\text{Max}}$ but rather at $C_R \cdot R_{\text{Max}}$, where $C_R$ is a relaxation constant. The selection of this constant trade-off the keeping of the order between those larger values and the arithmetic stability of the algorithm that is affected by the extreme values in $M$.

Another problem in Algorithm [1] is that for a finite $L$ the entries in $M$ become biased. Because the value $1/\hat{y}_{i/j}$ is bounded in the range $[1, 1/y_{\text{Min}}]$ its expected value exists. Since $\phi(x) = 1/x$ is a strictly convex function in the range $[y_{\text{Min}}, 1]$, for a non degenerate distribution of $X$ we know from Jensen’s inequality that

$$E\left[ \frac{1}{\hat{y}_{i/j}} \right] > \frac{1}{E[\hat{y}_{i/j}]}.$$

Thus, from this inequality we have

$$E[M_{i/j}] = E\left[ \frac{1}{\hat{y}_{i/j}} \right] - 1 > \frac{1}{E[\hat{y}_{i/j}]} - 1.$$

A second bias factor comes from the truncation and thus

$$\frac{1}{E[\hat{y}_{i/j}]} < \frac{1}{E[\hat{y}_{i/j}]} = \frac{1}{p_{i/j}}.$$

Even though the two bias factors are opposite to each other they will rarely cancel each other, so the expected value $E[M_{i/j}]$ will most likely remain biased. Minimizing a squared error function such as the Frobenius norm on an estimated matrix $M$, is a biased estimator for the true matrix $M$ with partially observed entries, will result in a biased estimation of the ratios $\omega_{i/j}$.

The third problem in Algorithm [1] is the different variance of different entries in $M$, which strongly depends on the probabilities $p_{i/j}$. To resolve the last two problems we replace the Frobenius norm minimization in [6] and [7] with the maximum likelihood estimator for $U$ and $V$ given $M, \mathcal{E}$ and the result obtained from the previous iteration. Since each pair may be compared a different number of times against each other, in the analysis we will replace the number of comparisons $L$ with a matrix $L$ containing the number of comparison per each pair. The likelihood function is given by

$$L(UV, \hat{U}^{(t-1)}) = \prod_{(i, j) \in \mathcal{E}} \left( \frac{L_{i/j}}{p_{i/j}} \right)^{k_{i/j}} (1 - p_{i/j})^{L_{i/j} - k_{i/j}},$$

(15)

where $p_{i/j} = 1/(1 + M_{i/j})$, $M_{i/j} = U_{i} V_{j}$ and $k_{i/j} = L_{i/j} \cdot Y_{i,j}$. For a given factor $\hat{U}^{(t-1)}$ that approximates $U$, we may define $p_{i/j}$ in terms of $V^{(t)}$ using the following

$$p_{i/j} = \frac{1}{1 + \hat{U}^{(t-1)}_{i} \hat{V}^{(t)}_{i}}.$$

(16)

Plugging (16) to (15) leads to the following log-likelihood objective function (without terms dependent of $V$)

$$\mathcal{L} = \sum_{(i, j) \in \mathcal{E}} (L_{ij} - k_{ij}) \log \left( \frac{\hat{U}^{(t-1)}_{i} \hat{V}^{(t)}_{i}}{L_{ij} \cdot \hat{U}^{(t-1)}_{i} \hat{V}^{(t)}_{i}} \right) - L_{ij} \log \left( 1 + \hat{U}^{(t-1)}_{i} \hat{V}^{(t)}_{i} \right).$$

To maximize $\mathcal{L}$ with respect to $\hat{V}^{(t)}_{q}$, we take the derivative

$$\frac{d\mathcal{L}}{d\hat{V}^{(t)}_{q}} = \sum_{j: (q, j) \in \mathcal{E}} \left( \frac{L_{aq} - k_{aq}}{\hat{V}^{(t)}_{q}} - \frac{L_{aq} \cdot \hat{V}^{(t-1)}_{q}}{1 + \hat{U}^{(t-1)}_{i} \hat{V}^{(t)}_{q}} \right) = 0.$$

(17)

By some arithmetical operations, (17) becomes

$$\frac{1}{\hat{V}^{(t)}_{q}} \cdot \sum_{j: (q, j) \in \mathcal{E}} \left( \frac{L_{aq}}{1 + \hat{U}^{(t-1)}_{j} \hat{V}^{(t)}_{q}} - k_{aq} \right) = 0.$$

(18)

Since $\hat{V}^{(t)}_{q} > 0$, we obtain the following equation

$$\sum_{j: (q, j) \in \mathcal{E}} \frac{L_{aq}}{1 + \hat{U}^{(t-1)}_{j} \hat{V}^{(t)}_{q}} = \sum_{j: (q, j) \in \mathcal{E}} L_{aq} \cdot \frac{k_{aq}}{L_{aq}}.$$

(19)

Dividing both sides by $n$ and assigning $k_{aq} = L_{aq} \cdot y_{q/j}$

$$\frac{1}{n} \cdot \sum_{j: (q, j) \in \mathcal{E}} \frac{L_{aq}}{1 + \hat{U}^{(t-1)}_{j} \hat{V}^{(t)}_{q}} = \frac{1}{n} \cdot \sum_{j: (q, j) \in \mathcal{E}} L_{aq} \cdot y_{q/j}.$$

(20)
or in terms of the ratio matrix \( M \)
\[
\frac{1}{n} \sum_{j: (q,j) \in E} \frac{L_{qj}}{1 + \hat{U}_{qj}^{(t-1)}\hat{V}_q^{(t)}} = \frac{1}{n} \sum_{j: (q,j) \in E} \frac{L_{qj}}{1 + M_{qj}}, \quad (21)
\]

Defining the weighting factor
\[
W_{qj} = \frac{L_{qj}}{\max_j(L_{qj})},
\]
and dividing \(^{21}\) by \( \max_j(L_{qj}) \) we have:
\[
\frac{1}{n} \sum_{j: (q,j) \in E} \frac{W_{qj}}{1 + \hat{U}_{qj}^{(t-1)}\hat{V}_q^{(t)}} = \frac{1}{n} \sum_{j: (q,j) \in E} \frac{W_{qj}}{1 + M_{qj}}.
\]

By defining the constant
\[
S_q = \frac{1}{n} \sum_{j: (q,j) \in E} \frac{W_{qj}}{1 + M_{qj}}, \quad (24)
\]
and the transformation
\[
Z = \frac{1}{\hat{V}_q^{(t)}},
\]
where both \( Z \) and \( S_q \) are in the range \([0, 1]\), \(^{23}\) becomes:
\[
f(Z) = \frac{1}{n} \sum_{j: (q,j) \in E} \frac{W_{qj}}{\hat{V}_q^{(t-1)}} = S_q.
\]

Note that the maximal valid value of \( f(Z) \), when \( Z = 1 \) is
\[
\max_Z(f(Z)) = \frac{1}{n} \sum_{j: (q,j) \in E} \frac{W_{qj}}{1 + \hat{U}_{qj}^{(t-1)}}.
\]

Therefore, if the calculated constant \( S_q \) is larger than \(^{26}\) we return \( Z_0 = 1 \), since values of \( Z \) grater than one are not valid. Otherwise, we use the fact that the \( f(Z) \) is a strictly monotone function in the range \( Z \in [0, 1] \) and define the function
\[
g(Z) = f(Z) - S_q
\]
which is also strictly monotone and has one root in the interval \( Z \in [0, 1] \). The root can be easily found using either the simple bisection method or the faster Brent’s method. Even for the simple bisection and a required accuracy \( \delta_Z \), we will find a good approximation for the root in \( O(\log_2(1/\delta_Z)) \) steps. The solution for \( \hat{V}_q^{(t)} \) in this case is obtained using the root approximation \( (Z_0) \) for the function \( g(Z) \) as \( \hat{V}_q^{(t)} = 1/Z_0 \).

We will define the process of calculating the MLE of \( \hat{V}_q^{(t)} \) as
\[
\hat{V}_q^{(t)} = \text{FactorMLE}(M,E,\hat{U}_q^{(t-1)},\delta_Z, L).
\]

For the MLE of \( \hat{U}_q^{(t)} \) we define
\[
S_q = \frac{1}{n} \sum_{i: (i,q) \in E} \frac{W_{iq}}{1 + \hat{U}_q^{(t)}\hat{V}_q^{(t)}}, \quad (28)
\]
where \( W_{iq} \) is as defined in \(^{22}\) Now we have to solve
\[
f(Z) = \frac{1}{n} \sum_{i: (i,q) \in E} \frac{W_{iq}}{1 + Z\hat{V}_q^{(t)}} = S_q;
\]

In this case, the minimal value of \( f(Z) \) is
\[
\min_Zf(Z) = \frac{1}{n} \sum_{j: (q,j) \in E} \frac{W_{qj}}{1 + \hat{V}_q^{(t)}}.
\]

obtained for \( Z = 1 \). Thus, if we have
\[
S_q \leq \frac{1}{n} \sum_{i: (i,q) \in E} \frac{W_{iq}}{1 + \hat{V}_q^{(t)}}, \quad (31)
\]
we will assign \( \hat{U}_q^{(t)} = 1 \). Otherwise, we use the fact that \( f(Z) \) is strictly monotone to get a numerical solution by the method described above for \( \hat{V}_q^{(t)} \), we define the entire process as
\[
\hat{U}_q^{(t)} = \text{FactorMLE}(M,E,\hat{V}_q^{(t)},\delta_Z, L).
\]

For low values in \( L \), the entries of \( M \) are very noisy and may get extreme values. To avoid such values from affecting the result, we propose to truncate the values of \( U \) and \( V \) at each iteration. The values to be truncated are the values of \( \hat{U} \), which estimates \( \hat{V} \), that are smaller than \( 1/(C_R \cdot R_{Max}) \), and the value of \( \hat{V} \), which estimates \( 1/\hat{\omega} \), that are larger than \( C_R \cdot R_{Max} \).

Since we know that for the optimal solution we have \( U_q = 1/V_q \), we can force the consistency of the solution for \( U \) and \( V \) at each iteration with the optimal solution by applying the steps
\[
\hat{R}_q^{(t)} = \frac{\hat{U}_q^{(t)} + 1}{V_q^{(t)}}, \quad \hat{U}_q^{(t)} = \hat{R}_q^{(t)}, \quad \hat{V}_q^{(t)} = \frac{1}{\hat{R}_q^{(t)}}. \quad (32)
\]

We will define the set of assignments in \(^{32}\) as
\[
(\hat{V}_q^{(t)}, \hat{U}_q^{(t)}) \leftarrow \text{ForceConsistency}(\hat{V}_q^{(t)}, \hat{U}_q^{(t)}).
\]

In order to perform truncation, the initial estimation \( \hat{U}^{(0)} \) needs to have a correct sign, which is obtained by defining
\[
\text{VecSign}(U) = \begin{cases} +1 & : \sum_{i=1}^{n} \text{sign}(U_i) \geq 0 \\ -1 & : \text{Else} \end{cases} \quad (33)
\]

Multiplying \( \hat{U}^{(0)} \) by \( \text{VecSign}(\hat{U}^{(0)}) \) assures the correct sign.

VI. ALGORITHM FOR NOISY DATA

The improved version of Algorithm \(^{14}\) that includes all the changes discussed in the previous section appears in Algorithm \(^{2}\). It better handles noise in the initial matrix \( M \) caused by limited pairwise comparisons.
The main difference between Algorithm 2 and Algorithm 1 is where we replaced the Frobenius norm minimization with the MLE (maximum likelihood estimator) of each factor given the previously estimated factor (either $U$ or $V$) and the matrix $M$. The MLE takes into account the probabilities of all possible values in $M$ and therefore it does not need the estimates of $M$ to be unbiased as is the case with the Frobenius norm minimization, which minimizes the error around these entries.

Another difference between the algorithms is the solving of the equation in the next iteration is consistent with the known optimal form of the solution for $U$ and $V$ (line 17 of Algorithm 2). This reduces the effect of errors in the values of these factors at initial iterations. Because this step may not be helpful in the case of constant $L_{ij} = L_{V(i,j)}$ and $L \to \infty$ where the entries in $M$ are exact, it may happen that forcing consistency at an early stage will slow down the convergence of the algorithm. Another difference is the truncation steps (lines 8, 15, and 16 of Algorithm 2) added to ensure that spurious values at early iterations do not interfere with the convergence of the algorithm. Clearly, these steps are also unnecessary in the case of $L \to \infty$.

Estimating $R_{\text{Max}}$. Under certain assumptions, if $R_{\text{Max}}$ is unknown, it can be estimated from the observation matrix $Y$. If we assume that the preference scores $\omega$ are uniformly distributed in the range $[\omega_{\text{Min}}, 1]$, then if we calculate the probability of the weakest item (corresponding to $\omega_{\text{Min}}$) to win $k$ times against a random item, we get from the law of total probability that

$$p(k|\omega_{\text{Min}}, L) = \int_{\omega_{\text{Min}}}^{1} \frac{1}{1 - \omega_{\text{Min}}} \cdot p(k|\omega, \omega_{\text{Min}}, L) \cdot d\omega.$$ 

Now, we can insert the probability that the weakest item will win $k$ times against an item with a preference score $\omega$, which is simply the binomial distribution with a probability $p = \omega_{\text{Min}}/(\hat{\omega}_{\text{Min}} + \omega)$ and $L$ games

$$p(k|\omega_{\text{Min}}, L) = \int_{\omega_{\text{Min}}}^{1} \frac{1}{1 - \omega_{\text{Min}}} \int_{\omega_{\text{Min}}}^{L-k} \frac{\omega}{\omega_{\text{Min}} + \omega} \cdot (\omega_{\text{Min}} + \omega)^{L-k} \cdot d\omega.$$ 

Using the variable change $x = \omega/\omega_{\text{Min}}$, we get

$$p(k|\omega_{\text{Min}}, L) = \frac{L}{k} \cdot \frac{\omega_{\text{Min}}^{-1}}{1 - \omega_{\text{Min}}} \int_{1}^{\omega_{\text{Min}}} \int_{1}^{x+1} \frac{1}{x} \cdot dx.$$ 

By defining the constant

$$A = \frac{\omega_{\text{Min}}}{1 - \omega_{\text{Min}}},$$

and taking the expectation with respect to $k$

$$E[k] = A \sum_{k=0}^{L} \frac{\omega_{\text{Min}}^{-1}}{1 - \omega_{\text{Min}}} \int_{1}^{\omega_{\text{Min}}} \int_{1}^{x+1} \frac{1}{x} \cdot dx. \quad (34)$$

Swapping the integral and sum we have

$$E[k] = A \cdot \int_{1}^{\omega_{\text{Min}}} \frac{1}{x+1} \frac{L}{1} \sum_{k=0}^{L} \frac{\omega_{\text{Min}}^{-1}}{k} \frac{1}{k} \cdot dx. \quad (35)$$

Solving for the sum and placing the result we have

$$E[k] = L \cdot A \cdot \int_{1}^{\omega_{\text{Min}}} \frac{1}{x+1} \frac{L}{1} \frac{1}{x+1} \cdot \frac{1}{1} \cdot dx. \quad (36)$$

By calculating the integral and dividing by $L$ we get:

$$E \left[ \frac{k}{L} \right] = \frac{\omega_{\text{Min}}}{1 - \omega_{\text{Min}}} \cdot \ln \left( \frac{1 + \omega_{\text{Min}}}{2 \cdot \omega_{\text{Min}}} \right). \quad (37)$$

Notice that each entry in $Y$ is a proxy of $k_{ij}/L$ where $k_{ij}$ is the number of times item $i$ won a match with item $j$. Thus, the average value of entries in a certain row in $Y$ (excluding $Y_{ii}$) is simply an estimate for $E \left[ \frac{k}{L} \right]$ for the $i$th item, which we will refer to as

$$\hat{E}_i = \frac{1}{(|\{(i,j) \in E \land j \neq i\}| \cdot \sum_{j:(i,j) \in E \land j \neq i} Y_{ij}}. \quad (38)$$

Since we want the expectation of the row corresponding to $\omega_{\text{Min}}$ (weakest item), we use the minimal value across items

$$\hat{E} = \min_{i} (\hat{E}_i). \quad (39)$$

We define the strictly monotonic (for $Z \in [0, 1]$) function

$$g(Z) = \frac{Z}{1 - Z} \cdot \ln \left( \frac{1 + Z}{2 \cdot Z} \right) - \hat{E}, \quad (40)$$

for which the root gives us the estimation for $\omega_{\text{Min}}$ since it represent the preference score that best explains the expected number of wins for the weakest item in the group. The root of $g(Z)$ is easy to find up to an arbitrary required precision $\delta_Z$ in $O(\log(1/\delta_Z))$ steps. After finding the root (there is only one) of the function $g(Z)$, which we denote as $Z_0$, the estimation
for the ratio $R_{Max}$, for our selected normalization of $\omega_{Max} = 1$, is given based on the definition in (9) by the ratio
\[
\hat{R}_{Max} = \frac{1}{Z_0},
\]
(41)

VII. EXPERIMENTS
A. Comparing the noiseless LRMC and noisy MC-MLE

In this section we test the contributions of the modification presented in Algorithm 2 to the straightforward adaptation (Algorithm 1) of the LRMC algorithm presented in [1] in two cases.

The first case is a scan over values of $L$ for different values of $p_{obs}$. MC-MLE represent the final algorithm as presented in Algorithm 2 and LRMC represent the noiseless algorithm as presented in Algorithm 1. A 95% confidence bound for the rank error is calculated by fitting a generalized linear regression for binomial distribution to the empirical rank error the rank error is calculated by fitting a generalized linear regression for binomial distribution to the empirical rank error.

To test the modifications, 500 iterations were used for each value of $L$ and $p_{obs}$. The vector $\vec{u}$ has $N_T$ preference scores which always include two values $R_{max}$ and 1. The rest of the $N_T - 2$ preference scores are randomized, at each iteration, using a uniform distribution $U(0, 1)$. The uniform distribution yields a random vector $\vec{u}$ of size $N_T - 2$, then by applying
\[
\vec{\omega} = R_{max}^{-1} + (1 - R_{max}^{-1}) \cdot \vec{u} / \max(\vec{u}) - \min(\vec{u}),
\]
(42)
we get the remaining $N_T - 2$ values of the preference scores.

In Fig. 2 a value of $R_{max} = 8$ is fixed for all $L$ values and the number of preference scores ($N_T$) is set to 50. The value of the constant $C_R$ is chosen to be 1.4 and we set $\delta = \Delta \omega_{min} / (20 \cdot N_T)$ to assure sufficient accuracy.

The second case scans over values of $p_{obs}$ without noise ($L \to \infty$) for different values of $R_{max} = \omega_{max} / \omega_{min}$. Because the 95% confidence bound on the rank error appears to be very small for this scenario, the probability of a ranking error is presented instead. This case is tested for several values of $N_T$ (the number of items). In this case, there is no need for truncation so $C_R$ is chosen to be 1000 such that the truncation has no effect. Another change is that the forcing of the factors to the form of the optimal solution at the initial iterations causes the algorithm to converge to a small error larger than zero for all $p_{obs}$ values. Therefore, when $L \to \infty$ we remove line [17] from Algorithm 2.

As can be seen in Fig. 3 the MC-MLE ranking algorithm shows significant improvement (over the simple adaptation of LRMC) for low values of $p_{obs}$ even in the simple completion problem without noise. The improvement becomes more significant as higher values of $R_{max}$ are used. In Fig. 2 we can see that the modified algorithm MC-MLE performs better than the simple LRMC on all value of $L$ and for all values of $p_{obs}$ tested. The difference between the algorithms becomes smaller as the value of $L$ increases. However, as can be seen in Fig. 3 this is only true for a high enough value of $p_{obs}$.

For both scenarios the algorithm received the value of $R_{max}$ as an input. However as we tested in the following simulations the value of $R_{max}$ can be estimated from the data matrix $Y$ instead of being inserted as input to the algorithm.

B. Comparison to current state-of-the-art

A comparison to other ranking methods is presented here. We compare Rank Centrality (RC) [16] and Spectral MLE (SMLE) [17] algorithms to our suggested MC-MLE algorithm.

For SMLE we used the constant $c3 = 0.1$ instead of $c3 = 1$ as used in [17] in order to get better results for SMLE. In this simulation 200 Monte-Carlo trials were used with $R_{max} = 2$. The reported results are obtained by averaging over all the Monte-Carlo trials. The preference scores ($\vec{C}$) are randomized uniformly as previously described in the testing of the algorithm modifications. For all the following results the value of $R_{max}$ is estimated from the data matrix $Y$ and inserted to the MC-MLE algorithm, so it is not an input of the overall algorithm. The value of $\Delta \omega_{min}$ is $1e-6$ for all the following simulations, and as before $\delta = \Delta \omega_{min} / (20 \cdot N_T)$.

We observed that at low enough values of $L$, choosing a small value for $C_R$ may result in a too strict truncation at initialization and so a homogeneous vector of equal scores. Thus, the value of $C_R$ needs to slightly increase for larger
\( p_{\text{obs}} \) and \( L \), but also for very low values of \( L \). For simulation we used:

\[
C_R = \begin{cases} 
  1.2 & : p_{\text{obs}} \leq 0.2 \\
  1.4 & : p_{\text{obs}} > 0.2 , L \geq 10 \\
  1.8 & : p_{\text{obs}} > 0.2 , L < 10.
\end{cases}
\]  

(43)

As can be seen in Fig. 4, MC-MLE achieves better performance than both RC and SMLE for all values of \( L \) in this simulation. The performance difference is slightly reduced for the smaller observation probability, and perhaps an even better choice of \( C_R \) can improve this result as we did not optimize over this value. This improvement is consistent over different values of \( R_{\text{max}} \) as can be demonstrated in Fig. 6. In this simulation a value of 100 is used for the number of items \( N_T \).

![Fig. 4. Scanning over different number of repeated comparisons \( L \)](image)

Further decreasing the value of \( \Delta \omega_{\text{min}} \) was tested for improved results and displayed no significant benefit so we fixed the value of \( \Delta \omega_{\text{min}} \) throughout all the following simulations.

![Fig. 5. Scanning over different observation probabilities \( p_{\text{obs}} \)](image)

As can be seen in Fig. 5, MC-MLE achieves better performance than both RC and SMLE for all values of \( p_{\text{obs}} \) in this simulation (apart from one scenario where \( L = 5 \) and \( p_{\text{obs}} = 0.2 \)). Note that the performance difference increases as the number of comparisons \( L \) is reduced.

![Fig. 6. Scanning over different weights ratio \( (R_{\text{max}}) \) values](image)

In Fig. 6 we can observe that the performance improvement of MC-MLE compared with RC and SMLE is independent of the choice of \( R_{\text{max}} \) even though SMLE takes as input the value of \( R_{\text{max}} \) and in the case of MC-MLE it is estimated from the observation matrix \( Y \).

![Fig. 7. Scanning over different number of items \( N_T \)](image)

In Fig. 7 we can see that the performance of MC-MLE depends strongly on \( N_T \) and that as the value of \( p_{\text{obs}} \) decreases a larger value of \( N_T \) is required to achieve better performance than SMLE for a constant value of \( L = 20 \) and \( R_{\text{max}} = 2 \). Perhaps an optimization of \( C_R \) could also help in this case.

### C. Evaluation on weather data-base

The LRMC ranking algorithm was tested on a weather database of monthly measurements from 45 states over several years starting from July 2001. For each pair of states in each month a match was performed by adding a score of 1 point for the state with the higher temperature and a score of 0.5 to both in cases of ties. The probability estimator \( y_{i/j} \) was calculated by normalizing each total score by the total amount of points given.
D. Analysis of the football (soccer) data

The MC-MLE ranking algorithm was also applied to football matches data using the Olympic, European championship (UEFA) and FIFA world cup games. The probability estimator for \( i \) over \( j \) was calculated by giving 3 points for each win of \( i \) over \( j \) regardless of the match score and one point to both \( i \) and \( j \) in cases of a tie score. The sum of the points for both \( i \) and \( j \) are then normalized by the total amount of points distributed to both. In order to compare with the FIFA ranking algorithm, the FIFA ranking was taken for the top 50 teams of each year from 2008 to 2016. The MC-MLE ranking was constructed for each year on the top 50 teams from the FIFA ranking of December of the same year using the data of the previous years. The number of previous years taken into the estimation was tested on another time window (years 1999-2008) to find an optimal window size that includes enough data to properly compare all teams but at the same time does not take into account too old matches that may be irrelevant to the teams’ current status.

To compare the algorithms, for each year, a score was given to each algorithm by looking at all the games of that year (involving the FIFA top 50 teams). For every game that ended with a team winning, each algorithm that ranked the winning team higher than the losing team had received a point. For each tie, the algorithm that ranked the teams closer (than the other algorithm) had received half a point. In tie cases where the distance in ranking is identical for both algorithms, each received half a point. For each year the MC-MLE algorithm ranked the top 50 teams from FIFA’s ranking of December of the previous year using data from previous years.

Though in some years the results are close, MC-MLE scored better in 9 out of the 10 years tested, tied in one out of 10 years and scored lower in one out of 10 years. The optimal window size chosen is 8 years of backward data. If we use a larger window of 9 years MC-MLE has a higher score on 6 years and two tie scores out of 10 years, and if we use a smaller window of 7 years MC-MLE has a higher score on 7 years and two tied scores out of 10 years. The choice of the window is important. However, the MC-MLE algorithm is still better on a smaller and larger choice of window. On the validation data (years 1999-2008), the MC-MLE performed better in the prediction of the results of 7 years and tied on one year based on the 8 years window size. The MC-MLE algorithm also performed better (on the validation set) when a window size of 7 or 9 years is applied. The optimal window on the validation is 9 years of backward look having better results than FIFA’s method of ranking on 8 of the 10 years tested.

VIII. CONCLUSIONS

In this paper, we proposed a new method for rank recovery based on a matrix completion approach. We presented a strategy for rank recovery from partial observations that supports limited comparisons, which introduces noise to the matrix completion model. The proposed approach was tested in both a limited comparisons scenario and a noiseless scenario and had shown improvement over the state-of-the-art. We tested the
stability of our proposed method under different observation probabilities on a weather data-set assembled from limited comparisons. Finally, we compared FIFA's ranking to our ranking and evaluated them by testing their performance in predicting match results on successive years (for ten years). Our solution achieved better performance than FIFA's method for team ranking.

Our matrix completion approach can also be adapted to other models. For instance, to the model proposed in [18]:

\[ P(i \text{ beats } j) = \omega_i/(\omega_i + \theta \cdot \omega_j), \]
\[ P(j \text{ beats } i) = \omega_j/(\theta \cdot \omega_i + \omega_j), \]
\[ P(i \text{ ties } j) = (\theta^2 - 1) \cdot (\omega_i + \theta \cdot \omega_j) / (\omega_i + \theta \cdot \omega_j) \]

which includes tie results. For this model we construct the incomplete low-rank matrix as:

\[ M_{ij} = 1/P(j \text{ beats } i) - 1 = \theta \cdot \omega_i \]
\[ M_{ji} = 1/P(i \text{ beats } j) - 1 = \theta \cdot \omega_j \]

This is equivalent to the original problem we defined in this paper up to a scale (which does not affect our solution). After recovering the preference scores vector \( \omega \), we may use all the tie probabilities estimators and the preference scores estimation to recover the factor \( \theta \). We leave further analysis of this model to a future work.

ACKNOWLEDGMENT

This work is partially supported by ERC-StG SPADE PI Giryes.

REFERENCES

[1] P. Jain, P. Netrapalli and S. Sanghavi, Low-rank Matrix Completion using Alternating Minimization, STOC’13 Proceedings of the forty-fifth annual ACM symposium on Theory of computing, pp. 665-674 STOC, 2013.
[2] L. Baltrunas, T. Makcinskas and F. Ricci, Group recommendations with rank aggregation and collaborative filtering, ACM conference on Recommender systems, pp. 119-126. ACM, 2010.
[3] X. Chen, P.N. Bennett, K. Collins-Thompson and E. Horvitz, Pairwise ranking aggregation in a crowdsourced setting, WSDM, pp. 193-202, 2009.
[4] R. H. Keshavan, A. Montanari, S. Oh, Matrix Completion from a Few Entries, IEEE Transactions on Information Theory, 56 (6): 2980-2998, 2010.
[5] B. Recht, A Simpler Approach to Matrix Completion, Journal of Machine Learning Research. 12: 3413-3430, 2009.
[6] E.J. Cands, T. Tao, The Power of Convex Relaxation: Near-Optimal Matrix Completion, IEEE Transactions on Information Theory. 56 (5): 2053-2080, 2010.
[7] J.F. Cai, E.J. Cands, Z. Shen, A Singular Value Thresholding Algorithm for Matrix Completion, SIAM Journal on Optimization. 20 (4): 1956-1982, 2010.
[8] D.L. Donoho, M. Gavish and A. Montanari, The phase transition of matrix recovery from Gaussian measurements matches the minimax MSE of matrix denoising, PNAS May 21, 2013. 110 (21) 8405-8410; https://doi.org/10.1073/pnas.1306110110.

[9] B. Eriksson, L. Balzano and R. Nowak, High-Rank Matrix Completion and Subspace Clustering with Missing Data, 2013. [arXiv:1112.5629]
[10] E.J Cands, B. Recht, Exact Matrix Completion via Convex Optimization, Foundations of Computational Mathematics. 9 (6): 717-772, 2009.
[11] E.J Cands, Y. Plan, Matrix Completion with Noise, Proceedings of the IEEE. 98 (6): 925-936, 2010.
[12] R.A. Bradley and M.E. Terry, Rank analysis of incomplete block designs: I. the method of paired comparisons, Biometrika, pp. 324-345, 1952.
[13] H. Zou, T. Hastie and R. Tibshirani, Sparse principal component analysis, JCGS, 15(2):2622-286, 2006.
[14] J. Kim and H. Park, Sparse nonnegative matrix factorization for clustering, Technical Report GT-CSE-08-01, Georgia Institute of Technology, 2008.
[15] Y. Koren, R. M. Bell and C. Volinsky, Matrix factorization techniques for recommender systems, IEEE Computer, 42(8):3037, 2009.
[16] S. Negahban, S. Oh and D. Shah, Rank centrality: Ranking from pairwise comparisons, 2012. URL: http://arxiv.org/abs/1209.1688
[17] Y. Chen and S. Changho, Spectral MLE: Top-K Rank Aggregation from Pairwise Comparisons, Proceedings of the International Conference on Machine Learning, 2015. JMLR: W&CP volume 37.
[18] P.V. Rao, L.L. Kupper Ties in Paired-Comparison Experiments: A Generalization of the Bradley-Terry Model, 2012, Journal of the American Statistical Association, 62:317, 194-204, DOI: 10.1080/01621459.1967.10482901.
[19] Y. Koren, R.M. Bell and C. Volinsky, Matrix factorization techniques for recommender systems, IEEE Computer, 42(8):3037, 2009.