Improving Rank Aggregation via Matrix Completion

Dimitris Papadimitriou[⋆], **Nicholas Platias**[†], **David Saltzman**[⋆]

Abstract

We consider the problem of determining a global ranking of objects by looking at pairwise comparisons between them, an area of active research due to its vast applications. In this paper we present a novel method for improving the global ranking of the objects in cases where there are limited pairwise comparisons available. Our approach is based on using Matrix Completion algorithms to predict missing pairwise comparisons. We implement our method in real and synthetic data sets under realistic assumptions. We obtain encouraging results which demonstrate that enhancing the Hodge Decomposition ranking algorithm with Matrix Completion via Nuclear Norm Minimization yields significantly improved performance over Rank Centrality and Hodge Decomposition ranking without Matrix Completion.

1. Introduction

Rank aggregating algorithms are very frequently used to produce a global ranking from partial comparisons between objects. This problem is generally impossible to solve as shown by the celebrated Arrow's impossibility theorem (Arrow, 1950). However it is increasingly important for applications like web searching and advertising. Thus, many polynomial-time approximation algorithms have been proposed that give suboptimal total rankings.

There exist algorithms that can efficiently and almost optimally rank items when all of the items have been compared many times, but when the nodes haven't all been compared, the algorithms' performance decreases. We are interested in predicting missing entries from the ones already obtained by using Matrix Completion algorithms in order to improve the resulting rankings. This method that we will study in our project can have a significant impact on web advertising firms since they can rank their adds efficiently from partial information, which enables them to rank their adds faster since they don't need to wait in order to gather enough user clicks.

The rest of the paper is organized as follows. First we present two recently proposed rank aggregation algorithms that output a global ranking of n items by using information for their pairwise comparisons. In section 3 we implement a number of Matrix Completion algorithms and report their performance in predicting missing values by comparing the error that we obtain. Section 4 contains the results that we acquired from the data sets that we used to implement our method.

1.1. Notation

In the rest of this paper we use M and $\hat{M}$ to denote the matrix with the given entries and the predicted matrix respectively. We denote with Ω_T and Ω_S the training and the testing sets of the matrix entries, that are used to train and test our algorithms. The frobenius norm of a matrix is given by $\|A\|_F = \sum_{i,j} |a_{i,j}|^2$ and the nuclear norm of a matrix is $\|A\|_* = \sum_k \sigma_k$ where σ_i's are the singular values of matrix A.

2. Ranking Algorithms

Nowadays, search engines want to select the top adds and rank them based on the expected number of clicks they will receive in order to maximize their profit. For this reason ranking algorithms are being extensively studied and many innovative methods have been proposed (Lu & Boutilier, 2011). In this project, we will use two algorithms for rank aggregation, called Rank Centrality and Hodge Ranking.

2.1. Rank Centrality

Rank Centrality is a recently proposed algorithm that produces a score for a set of objects $[n] = \{1, \dots, n-1\}$ by looking at comparisons between pairs $S \subset [n] \times [n]$. (Negahban et al., 2012). Intuitively, the algorithm performs a random walk on a graph $G([n], E, A)$ where (i, j) is an edge if there are comparisons between objects i and j and has weight A_{ij} the fraction of times that j was preferred to i. The produced score vector is the stationary distribution π of this random walk.

2.1.1. Bradley-Terry-Luce (BTL) Model

For the theoretical study of the algorithm it is assumed that the comparisons follow from the Bradley-Terry-Luce model. We will use the same assumption later to construct the synthetic data set. More specifically the model assumes that items $[n] = \{1, \dots, n-1\}$ are associated with latent weights $w_i \in \mathbb{R}_+$. The matrix of comparisons Y is constructed as follows:

$$Y_{ij}^{\ell} = \begin{cases} 1, & \text{w.p. } \frac{w_j}{w_i + w_j}, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

⋆ Department of Electrical Engineering, Stanford University
† Department of Mathematics, Stanford University

where ℓ denotes the ℓ_{th} comparison between the objects i and j. The random variables Y_{ij}^{ℓ} are considered independent given the weights w. After summing comparison results and normalizing as described in (Negahban et al., 2012) we obtain the stochastic matrix P which is the input to the algorithm.

The algorithm takes as its input the stochastic matrix P and outputs the stationary distribution, which when sorted gives the resulting ranking.

Algorithm 1 Rank Centrality

Input: $G = ([n], E, A)$
Output: Stationary distribution π
Compute transition matrix P
Initialize vector p
repeat $p_{t+1}^T = p_t^T P$
until convergence

If we make a fixed k number of comparisons between pairs in S and $k = O(logn)$, $|S| = O(nlogn)$ Rank Centrality converges to a stationary distribution π such that $\pi_i = w_i / \sum_{j=1}^{n} w_j$ which uniquely identifies the ranking of the objects.

2.2. Hodge Decomposition

Computing a global ranking using Hodge theory is done by decomposing the matrix of pairwise comparison into three components.

$$Y = -grad(s) \oplus H \oplus curl^*(\Phi) \tag{2}$$

where Y is the comparison matrix, s is the potential that minimizes the distance between Y and $-grad(s)$, H is the global inconsistency, and Φ is the vector potential such that $curl^*(\Phi)$ is the local inconsistency (Jiang et al., 2011). The consistent global potential s gives the global ranking of the elements; the inconsistency metrics are not used here.

Algorithm 2 Hodge Ranking

Input: $G = ([n], E, A)$
Output: Global score $\hat{s}$
1. Vectorize pairwise comparisons into w
2. Compute gradient δ_0
3. $div = \delta_0 * w$
4. $\hat{s} = lsqr(\delta_0^T \delta_0, div)$

3. Matrix Completion Algorithms

Matrix Completion algorithms are used to recover an unknown low rank matrix from a few given entries and have applications in many fields, such as recommendation systems. The general formulation of the problem is:

$$\begin{aligned} &\text{minimize} && rank(\hat{M}) \\ &\text{s.t.} && \hat{M}_{i,j} = M_{i,j}, \ \ \forall (i,j) \in \Omega_T \end{aligned} \tag{3}$$

Depending on the setting of various parameters this setting can be NP-hard. However, there are a number of standard methods used to solve it (Asendorf et al.). Lately some new techniques have been proposed, some of which succeed even perfect reconstruction of the matrix under certain conditions, as in (Jain et al., 2013), (Candès & Recht, 2009),(Candès & Tao, 2010) and (Keshavan et al., 2010).

3.1. Low Rank Approximation

In Low Rank Approximation (LRA) we first construct a training matrix M_T by subtracting from M's nonzero elements the mean of their row. Then we project M_T on the set of matrices of *rank* at most r. More formally we solve the problem:

$$\begin{aligned} &\underset{\hat{M}}{\text{minimize}} && \|M_T - \hat{M}\|_F \\ &\text{subject to} && rank(\hat{M}) \leq r \end{aligned} \tag{4}$$

We find the projection via the SVD of $M = U\Sigma V^*$: $P_r(M_T) = U_r \Sigma_r V_r^*$. Our prediction is $\hat{M} = A + \frac{1}{\alpha} P_r(M_T)$ where A_{ij} is the mean of row i and $\alpha > 0$ is a regularizartion parameter typically set to be the fraction of M's entries that are known. (Montanari, 2013),(Michenková, 2011). The method of subtracting first order structure (in this case row means) before computing higher order structure is very common and gives improved results.

3.2. Non-negative Matrix Factorization

The Non-negative Matrix Factorization (NMF) is another useful decomposition for multivariate data with applications in information retrieval and document clustering (Paatero & Tapper, 1994),(Seung & Lee, 2001). This method decomposes a non-negative matrix $M \in \mathbb{R}_+^{n \times n}$ into a product of two non-negative matrices $W \in \mathbb{R}_+^{n \times r}$ and $H \in \mathbb{R}_+^{r \times n}$ with rank r such that $M \approx WH$. The corresponding optimization problem is:

$$\text{minimize } \|M - WH\|^2, \ \text{ s.t. } W, H \geq 0 \tag{5}$$

We solve this problem by gradient descent using multiplicative update rules as described in (Seung & Lee, 2001).

3.3. Nuclear Norm Minimization

Nuclear Norm Minimization (NNM) is a recently proposed method for tight relaxation of the initial optimization problem (Matrix Completion) described above. (Wen et al., 2012), (Seung & Lee, 2001),(Candès & Recht, 2009) and (Candès & Tao, 2010). The problem reduces to solving the following convex optimization problem:

$$\begin{aligned} &\underset{\hat{M} \in \mathbb{R}^{n \times n}}{\text{minimize}} && \|\hat{M}\|_* \\ &\text{subject to} && \hat{M}_{i,j} = M_{i,j}, \ \ \forall (i,j) \in \Omega_T \end{aligned} \tag{6}$$

This formulation is guaranteed to recover the missing entries of a low rank matrix almost surely from $m \geq Cn^{1.2} r \log n$

samples for some constant C under certain assumptions. We solve this optimization problem by using the CVX package (Grant & Boyd, 2013).

4. Results

We first present the performance of the above Matrix Completion algorithms in predicting missing entries. We then explore all combinations of Matrix Completion-Ranking algorithms and compare performance before and after Matrix Completion.

4.1. Data Sets

In this project we use two different data sets on which we apply the proposed method for ranking improvement. The first one is a data set of real pairwise comparisons between political entities and the second is a synthetic data set that we constructed.

4.1.1. Real Data Set

The real data set that we will use was collected from an online polling in Washington Post[1]. It is poll data asking who had the worst year in Washington and it is organized in pairwise comparisons between 67 political entities (Negahban et al., 2012). On average for each pair there have been about k=20 comparisons. For this data set we learn the correct ranking by running Rank Centrality on the complete graph.

4.1.2. Synthetic Data Set

We construct a matrix of pairwise comparisons according to the BTL model. Our matrix has $n \times n$ dimension where we vary n and show respective results. Between each pair that we ultimately compare we perform an average number of $k = 20$ comparisons. This choice for k is done primarily in order to be close to the value of k for the real data set and also to be realistic for real applications, since a very large number of comparisons is not always available. A small k can also be interpreted as adding noise on the real weights of the graph. By construction the true ranking (in ascending order) is $1, \ldots, n-1$.

4.2. Performance in Data Sets

Here we will compare the performance of our algorithms LRA, NMF and NNM in the two data sets. For the evaluation of Matrix Completion we used $O(nlogn)$ pairwise comparisons in the initial matrix, from which we then held out 20% of the pairs of entries uniformly at random for testing. In order to compare the performance of the algorithms we implement we will use the Root Mean Square Error (RMSE) metric. The RMSE is given by:

$$\text{RMSE}(\hat{M}, M) = \sqrt{\frac{1}{|\Omega_S|} \sum_{(i,j)\in\Omega_S} (\hat{M}_{i,j} - M_{i,j})^2} \qquad (7)$$

and gives a good sense of the correctness of the estimation.

Since we uniformly at random delete or create edges, depending on the type of the data set, we will perform 20 trials for each algorithm on each data set. The resulting range of the RMSEs that we obtained can be seen in table 1.

Table 1. RMSE values obtained by testing the three Matrix Completion algorithms on the data sets.

Data set	LRA	NMF	NNM
Synthetic Data	4.1± 0.5	3.6± 0.3	2.4±0.5
Real Data	14.0± 4.0	12.0± 3.5	11.0± 2.5

We estimated the parameters r, α of LRA and r of NMF by cross-validation. In LRA we chose $\alpha = 0.2$ for which we obtained the optimal RMSE, which is close to the fraction of known entries to total entries. For the rank of approximation r in LRA and NMF we chose $\text{rank}_{LRA} = 15$ and $\text{rank}_{NMF} = 15$ also by cross-validation. We observe that the RMSE values for the real data are larger than the ones for the synthetic. One possible explanation is that the number of comparisons k in the real data set varied a lot, while k was constant in the synthetic data set.

4.3. Post-processing of Data

The resulting $\hat{M}$ matrix that we obtain from the completion algorithms is a full matrix. Some of its entries are less than zero in some cases or greater than one and in general they don't accurately satisfy $\hat{M}_{ij} + \hat{M}_{ji} = 1$. Consequently we appropriately threshold and weigh the values in order to be in accordance with the theory. We experimented with a number of post processing procedures. For instance we tried replacing pairs M_{ij} and M_{ji} with the mean values of the corresponding rows, when one of the two or both turned out to be negative in the completed matrix. We finally observed that the following heuristic produced best results: we set $M_{ij}, M_{ji} = 0$, i.e. as if there were no comparisons, when $\hat{M}_{ij}, \hat{M}_{ji} < 0$ and set $M_{ij} = 1$, $M_{ji} = 0$ when $\hat{M}_{i,j} > 0$, $\hat{M}_{j,i} < 0$. After the post processing phase we use our data as input for the Ranking Algorithm and obtain the resulting ranking.

4.3.1. Metrics for Ranking Comparisons

In order to test the robustness of our method we will use two different metrics to compare the rankings we acquire with the correct global ranking. More specifically we will use the Spearman's Footrule (SF) and the Kendall tau (KT) metrics (Kumar & Vassilvitskii, 2010). Both metrics operate on a permutation σ of the true ranking.

[1] 1 `http://www.washingtonpost.com/wp-srv/ /interactivity/worst-year-voting.html`

4.3.2. Spearmans Footrule (SF) Metric

$SF(\sigma) = \sum_i |i - \sigma(i)|$ i.e. SF counts the total displacement of objects as a result of σ.

4.3.3. Kendall's tau (KT) Metric

$KT(\sigma) = \sum_{i<j} \mathbb{1}(\sigma(i) > \sigma(j))$ i.e KT counts the number of inversions in σ.

The ordering of the distances of the rankings produced by the different algorithm combinations from the true rankings was the same under SF and KT hence we only report results for SF for the sake of simplicity, because all permutations σ satisfy $KT(\sigma) \leq SF(\sigma) \leq 2KT(\sigma)$.

4.4. Comparison of Matrix Completion Algorithms

As long as there are many edges and the number of pairwise comparisons k between each pair is sufficiently large, the Rank Centrality algorithm is guaranteed to give an optimal global ranking. Consequently, we are interested in studying cases that are also abundant in practice, in which fewer nodes have been compared, and these nodes have been compared relatively few times.

We first tested each Matrix Completion algorithm with each ranking algorithm to determine which Matrix Completion algorithms improved performance. The results in figures 1 and 2 show the effects of matrix completion on Rank Centrality and on Hodge Ranking, respectively, on the synthetic dataset. The synthetic dataset was constructed with $n = 100$ nodes. The threshold, which determined what percent of nodes were compared, was varied, and at each threshold the average distance from the true ranking across 10 Monte Carlo trials was calculated. The results show that Matrix Completion decreases performance for Rank Centrality, but can increase performance for Hodge Ranking. In particular, Nuclear Norm Minimization improves performance the most, particularly at thresholds between 0.15 and 0.65.

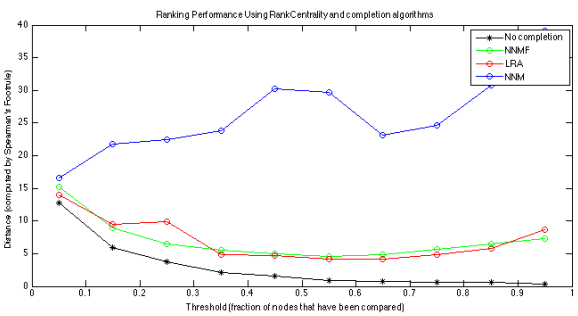

Figure 1. Rank Centrality distance from true ranking on synthetic data (n=100), with and without performing Matrix Completion.

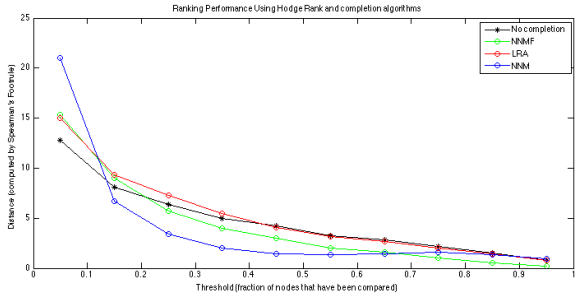

Figure 2. Hodge Ranking distance from true ranking on synthetic data (n=100), with and without performing Matrix Completion.

This is probably related to the combinatorial nature of the Hodge Ranking algorithm as opposed to the Markovian nature of Rank Centrality.

4.5. Final Results

To profile when Matrix Completion improves performance, we compared the best-performance algorithms Rank Centrality, Hodge Ranking, and Hodge Ranking with Nuclear Norm Minimization (NNM). Figure 3 shows the distances from the true ranking when the number of nodes n was fixed to 100 and the percentage of nodes that were compared (the threshold) was varied. It shows that for thresholds between 0.2 and 0.8, NNM improves the accuracy of Hodge Ranking; for thresholds between 0.2 and 0.5, this improvement is enough to outperform Rank Centrality. Figure 4 shows the distances when the threshold is fixed to 0.2, and the number of nodes is varied from 20 to 200. As n increases, Hodge Ranking with NNM performs relatively better. After n=70, NNM improves the performance of Hodge Ranking, and after n=95, Hodge Ranking with NNM outperforms Rank Centrality. As n increases, the performance difference increases.

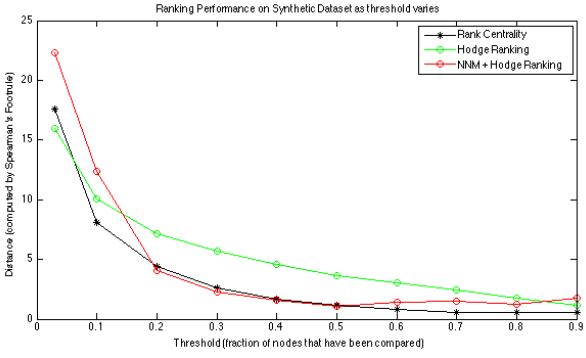

Figure 3. Distance from true ranking of Rank Centrality, Hodge Ranking, and NNM+Hodge Ranking on synthetic data as threshold changes

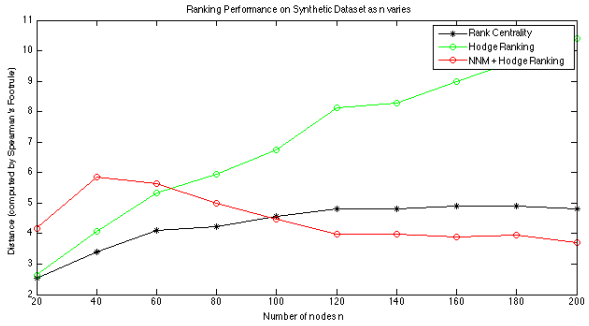

Figure 4. Distance from true ranking of Rank Centrality, Hodge Ranking, and NNM+Hodge Ranking on synthetic data as number of nodes changes

These three algorithms were also tested on the political dataset. For this dataset, $n = 67$, and the comparison matrix is almost full. A baseline ranking is performed, then on each trial some percent of comparisons are deleted and a new ranking is calculated. Figure 5 shows the distances to the baseline as the threshold (the percent of comparisons kept) is varied. It is consistent with the synthetic dataset at a similar number of nodes - NNM slightly increases the performance of Hodge Ranking for thresholds between about 0.3 and 0.8, but not enough to outperform Rank Centrality.

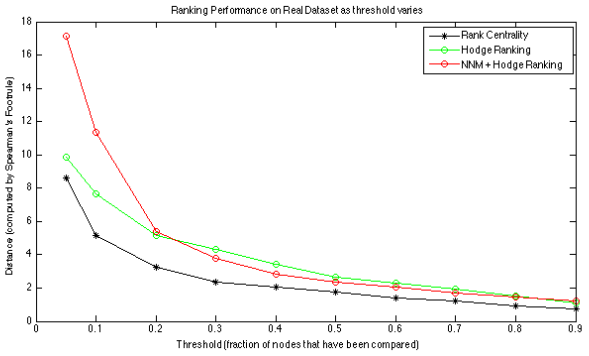

Figure 5. Distance from baseline ranking of Rank Centrality, Hodge Ranking, and NNM+Hodge Ranking on real data as threshold changes

Conclusions

In this project we proposed a method for improving the global ranking of items out of a limited number of pairwise comparisons between them by using Matrix Completion techniques; it was shown that this can improve performance. Matrix Completion helps Hodge Ranking but hurts Rank Centrality; this may be because the Hodge decomposition uses the additional structure generated by Matrix Completion for the global ranking, and the noise from uncertainty in Matrix Completion is captured in the uncertainty terms of the decomposition. The best performance was obtained by applying Nuclear Norm Minimization prior to Hodge Ranking when there were over 80 nodes and between about 15% and 60% of them had been compared. When ranking such a dataset, this is a promising algorithm to give a more accurate global ranking.

References

Arrow, Kenneth J. A difficulty in the concept of social welfare. *The Journal of Political Economy*, 58(4):328–346, 1950.

Asendorf, Nick, McGaffin, Madison, Prelee, Matt, and Schwartz, Ben. Algorithms for completing a user ratings matrix.

Candès, Emmanuel J and Recht, Benjamin. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717–772, 2009.

Candès, Emmanuel J and Tao, Terence. The power of convex relaxation: Near-optimal matrix completion. *Information Theory, IEEE Transactions on*, 56(5):2053–2080, 2010.

Grant, Michael and Boyd, Stephen. CVX: Matlab software for disciplined convex programming, version 2.0 beta. http://cvxr.com/cvx, September 2013.

Jain, Prateek, Netrapalli, Praneeth, and Sanghavi, Sujay. Low-rank matrix completion using alternating minimization. In *Proceedings of the 45th annual ACM symposium on Symposium on theory of computing*, pp. 665–674. ACM, 2013.

Jiang, Xiaoye, Lim, Lek-Heng, Yao, Yuan, and Ye, Yinyu. Statistical ranking and combinatorial hodge theory. *Mathematical Programming*, 127(1):203–244, 2011.

Keshavan, Raghunandan H, Montanari, Andrea, and Oh, Sewoong. Matrix completion from a few entries. *Information Theory, IEEE Transactions on*, 56(6):2980–2998, 2010.

Kumar, Ravi and Vassilvitskii, Sergei. Generalized distances between rankings. In *Proceedings of the 19th international conference on World wide web*, pp. 571–580. ACM, 2010.

Lu, Tyler and Boutilier, Craig. Learning mallows models with pairwise preferences. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pp. 145–152, 2011.

Michenková, Marie. Numerical algorithms for low-rank matrix completion problems, 2011.

Montanari, Andrea. ee378b: Inference, estimation and information processing. University Lectures, 2013.

Nan, Feng et al. *Low rank matrix completion*. PhD thesis, Massachusetts Institute of Technology, 2009.

Negahban, Sahand, Oh, Sewoong, and Shah, Devavrat. Iterative ranking from pair-wise comparisons. *arXiv preprint arXiv:1209.1688*, 2012.

Paatero, Pentti and Tapper, Unto. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126, 1994.

Seung, D and Lee, L. Algorithms for non-negative matrix factorization. *Advances in neural information processing systems*, 13:556–562, 2001.

Wen, Zaiwen, Yin, Wotao, and Zhang, Yin. Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. *Mathematical Programming Computation*, 4(4):333–361, 2012.