Rank Aggregation via Nuclear Norm Minimization

David F. Gleich*
Sandia National Laboratories†
Livermore, CA
dfgleic@sandia.gov

Lek-Heng Lim‡
University of Chicago
Chicago, IL
lekheng@uchicago.edu

ABSTRACT

The process of rank aggregation is intimately intertwined with the structure of skew-symmetric matrices. We apply recent advances in the theory and algorithms of matrix completion to skew-symmetric matrices. This combination of ideas produces a new method for ranking a set of items. The essence of our idea is that a rank aggregation describes a partially filled skew-symmetric matrix. We extend an algorithm for matrix completion to handle skew-symmetric data and use that to extract ranks for each item. Our algorithm applies to both pairwise comparison and rating data. Because it is based on matrix completion, it is robust to both noise and incomplete data. We show a formal recovery result for the noiseless case and present a detailed study of the algorithm on synthetic data and Netflix ratings.

Categories and Subject Descriptors

H.3.5 [**Information Storage and Retrieval**]: On-line information Services—*Web-based services*; G.1.3 [**Numerical analysis**]: Numerical Linear Algebra—*Singular value decomposition*

General Terms

Algorithms

Keywords

nuclear norm, skew symmetric, rank aggregation

1. INTRODUCTION

One of the classic data mining problems is to identify the important items in a data set; see Tan and Jin [2004] for an interesting example of how these might be used. For this task, we are concerned with rank aggregation. Given a series of votes on a set of items by a group of voters, rank aggregation is the process of permuting the set of items so that the first element is the best choice in the set, the second element is the next best choice, and so on. In fact, rank aggregation is an old problem and has a history stretching back centuries [Condorcet, 1785]; one famous result is that any rank aggregation requires some degree of compromise [Arrow, 1950]. Our point in this introduction is not to detail a history of all the possible methods of rank aggregation, but to give some perspective on our approach to the problem.

Direct approaches involve finding a permutation explicitly – for example, computing the Kemeny optimal ranking [Kemeny, 1959] or the minimum feedback arc set problem. These problems are NP-hard [Dwork et al., 2001, Ailon et al., 2005, Alon, 2006]. An alternate approach is to assign a score to each item, and then compute a permutation based on ordering these items by their score, e.g. Saaty [1987]. In this manuscript, we focus on the second approach. A key advantage of the computations we propose is that they are *convex* problems and efficiently solvable.

While the problem of rank aggregation is old, modern applications – such as those found in web-applications like Netflix and Amazon – pose new challenges. First, the data collected are usually cardinal measurements on the quality of each item, such as 1–5 stars, received from voters. Second, the voters are neither experts in the rating domain nor experts at producing useful ratings. These properties manifest themselves in a few ways, including skewed and indiscriminate voting behaviors [Ho and Quinn, 2008]. We focus on using aggregate pairwise data about items to develop a score for each item that predicts the pairwise data itself. This approach eliminates some of the issues with directly utilizing voters ratings, and we argue this point more precisely in Section 2.

To explain our method, consider a set of n items, labeled from 1 to n. Suppose that each of these items has an unknown intrinsic quality $s_i : 1 \leq i \leq n$, where $s_i > s_j$ implies that item i is better than item j. While the s_i's are unknown, suppose we are given a matrix $\boldsymbol{Y}$ where $Y_{ij} = s_i - s_j$. By finding a rank-2 factorization of $\boldsymbol{Y}$ (there is no rank-1 skew-symmetric factorization), for example

$$\boldsymbol{Y} = \mathbf{s}\mathbf{e}^T - \mathbf{e}\mathbf{s}^T, \tag{1}$$

we can extract unknown scores. The matrix $\boldsymbol{Y}$ is skew-symmetric and describes any score-based global pairwise ranking. (There are other possible rank-2 factorizations

*Supported by the Natural Sciences and Engineering Research Council of Canada and the Dept. of Energy's John von Neumann fellowship.

†Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

‡Supported by NSF CAREER Award DMS 1057064

KDD'11, August 21–24, 2011, San Diego, California, USA.

Table 1: Notation for the paper.

Sym.	Interpretation
$\mathcal{A}(\cdot)$	a linear map from a matrix to a vector
$\mathbf{e}$	a vector of all ones
$\mathbf{e}_i$	a vector with 1 in the ith entry, 0 elsewhere
$\|\cdot\|_*$	the nuclear norm
$\boldsymbol{R}$	a rating matrix (voters-by-items)
$\boldsymbol{Y}$	a fitted or model pairwise comparison matrix
$\hat{\boldsymbol{Y}}$	a measured pairwise comparison matrix
Ω	an index set for the known entries of a matrix

of a skew-symmetric matrix, a point we return to later in Section 3.1.)

Thus, given a measured $\hat{\boldsymbol{Y}}$, the goal is to find a minimum rank approximation of $\hat{\boldsymbol{Y}}$ that models the elements, and ideally one that is rank-2. Phrased in this way, it is a natural candidate for recent developments in the theory of matrix completion [Candès and Tao, 2010, Recht et al., 2010]. In the matrix completion problem, certain elements of the matrix are presumed to be known. The goal is to produce a low-rank matrix that respects these elements – or at least minimizes the deviation from the known elements. One catch, however, is that we require matrix completion over skew-symmetric matrices for pairwise ranking matrices. Thus, we must solve the matrix completion problem inside a structured class of matrices. This task is a novel contribution of our work. Recently, Gross [2010] also developed a technique for matrix completion with Hermitian matrices.

With a "completed" matrix $\boldsymbol{Y}$, the norm of the residual $\|\hat{\boldsymbol{Y}} - \boldsymbol{Y}\|$ gives us a certificate for the validity of our fit – an additional piece of information available in this model.

To continue, we briefly summarize our main contributions and our notational conventions.

Our contributions.

- We propose a new method for computing a rank aggregation based on matrix completion, which is tolerant to noise and incomplete data.
- We solve a structured matrix-completion problem over the space of skew-symmetric matrices.
- We prove a recovery theorem detailing when our approach will work.
- We perform a detailed evaluation of our approach with synthetic data and an anecdotal study with Netflix ratings.

Notation.

We try to follow standard notation conventions. Matrices are bold, upright roman letters, vectors are bold, lowercase roman letters, and scalars are unbolded roman or Greek letters. The vector $\mathbf{e}$ consists of all ones, and the vector $\mathbf{e}_i$ has a 1 in the ith position and 0's elsewhere. Linear maps on matrices are written as script letters. An index set Ω is a group of index pairs. Each $\omega \in \Omega$ is a pair (r, s) and we assume that the ω's are numbered arbitrarily, i.e. $\Omega = \{\omega_1, \ldots, \omega_k\}$. Please refer to Table 1 for reference.

Before proceeding further, let us outline the rest of the paper. First, Section 2 describes a few methods to take voter-item ratings and produce an aggregate pairwise comparison matrix. Additionally, we argue why pairwise aggregation is a superior technique when the goal is to produce an ordered list of the alternatives. Next, in Section 3, we describe formulations of the noisy matrix completion problem using the nuclear norm. In our setting, the LASSO formulation is the best choice, and we use it throughout the remainder. We briefly describe algorithms for matrix completion and focus on the SVP algorithm [Jain et al., 2010] in Section 3.1. We then show that the SVP algorithm preserves skew-symmetric structure. This process involves studying the singular value decomposition of skew-symmetric matrices. Thus, by the end of the section, we've shown how to formulate and solve for a scoring vector based on the nuclear norm. The following sections describe alternative approaches and show our recovery results. At the end, we show our experimental results. In summary, our overall methodology is

$$\begin{array}{c} \text{Ratings } (= \boldsymbol{R}) \\ \Downarrow\ (\S 2) \\ \text{Pairwise comparisons } (= \boldsymbol{Y}) \\ \Downarrow\ (\S 3) \\ \text{Ranking scores } (= \mathbf{s}) \\ \Downarrow\ (\text{sorting}) \\ \text{Rank aggregations.} \end{array}$$

An example of our rank aggregations is given in Table 2. We comment further on these in Section 6.3.

Finally, we provide our computational and experimental codes so that others may reproduce our results:
`https://dgleich.com/projects/skew-nuclear`

2. PAIRWISE AGGREGATION METHODS

To begin, we describe methods to aggregate the votes of many voters, given by the matrix $\boldsymbol{R}$, into a measured pairwise comparison matrix $\hat{\boldsymbol{Y}}$. These methods have been well-studied in statistics [David, 1988]. In the next section, we show how to extract a score for each item from the matrix $\hat{\boldsymbol{Y}}$.

Let $\boldsymbol{R}$ be a voter-by-item matrix. This matrix has m rows corresponding to each of the m voters and n columns corresponding to the n items of the dataset. In all of the applications we explore, the matrix $\boldsymbol{R}$ is highly incomplete. That is, only a few items are rated by each voter. Usually all the items have a few votes, but there is no consistency in the number of ratings per item.

Instead of using $\boldsymbol{R}$ directly, we compute a pairwise aggregation. Pairwise comparisons have a lengthy history, dating back to the first half of the previous century [Kendall and Smith, 1940]. They also have many nice properties. First, Miller [1956] observes that most people can evaluate only 5 to 9 alternatives at a time. This fact may relate to the common choice of a 5-star rating (e.g. the ones used by Amazon, eBay, Netflix, YouTube). Thus, comparing pairs of movies is easier than ranking a set of 20 movies. Furthermore, only pairwise comparisons are possible in certain settings such as tennis tournaments. Pairwise comparison methods are thus natural for analyzing ranking data. Second, pairwise comparisons are a relative measure and help reduce bias from the rating scale. For these reasons, pairwise comparison methods have been popular in psychology, statistics, and social choice theory [David, 1988, Arrow, 1950]. Such methods have also been adopted by the learning to rank community; see the contents of Li et al. [2008]. A final advantage of pairwise methods is

Table 2: The top 15 movies from Netflix generated by our ranking method (middle and right). The left list is the ranking using the mean rating of each movie and is emblematic of the problems global ranking methods face when infrequently compared items rocket to the top. We prefer the middle and right lists. See Section 6 and Figure 4 for information about the conditions and additional discussion. LOTR III appears twice because of the two DVDs editions, theatrical and extended.

Mean	Log-odds (all)	Arithmetic Mean (30)
LOTR III: Return ...	LOTR III: Return ...	LOTR III: Return ...
LOTR I: The Fellowship ...	LOTR I: The Fellowship ...	LOTR I: The Fellowship ...
LOTR II: The Two ...	LOTR II: The Two ...	LOTR II: The Two ...
Lost: Season 1	Star Wars V: Empire ...	Lost: S1
Battlestar Galactica: S1	Raiders of the Lost Ark	Star Wars V: Empire ...
Fullmetal Alchemist	Star Wars IV: A New Hope	Battlestar Galactica: S1
Trailer Park Boys: S4	Shawshank Redemption	Star Wars IV: A New Hope
Trailer Park Boys: S3	Star Wars VI: Return ...	LOTR III: Return ...
Tenchi Muyo! ...	LOTR III: Return ...	Raiders of the Lost Ark
Shawshank Redemption	The Godfather	The Godfather
Veronica Mars: S1	Toy Story	Shawshank Redemption
Ghost in the Shell: S2	Lost: S1	Star Wars VI: Return ...
Arrested Development: S2	Schindler's List	Gladiator
Simpsons: S6	Finding Nemo	Simpsons: S5
Inu-Yasha	CSI: S4	Schindler's List

that they are much more complete than the ratings matrix. For Netflix, $\boldsymbol{R}$ is 99% incomplete, whereas $\boldsymbol{Y}$ is only 0.22% incomplete and most entries are supported by *many* comparisons. See Figure 1 for information about the number of pairwise comparisons in Netflix and MovieLens.

More critically, an incomplete array of user-by-product ratings is a strange matrix – not every 2-dimensional array of numbers is best viewed as a matrix – and using the rank of this matrix (or its convex relaxation) as a key feature in the modeling needs to be done with care. Consider, if instead of rating values 1 to 5, 0 to 4 are used to represent the exact same information, the rank of this new rating matrix will change. Furthermore, whether we use a rating scale where 1 is the best rating and 5 is worst, or one where 5 is the best and 1 is the worst, a low-rank model would give the exact same fit with the same input values, even though the connotations of the numbers is reversed.

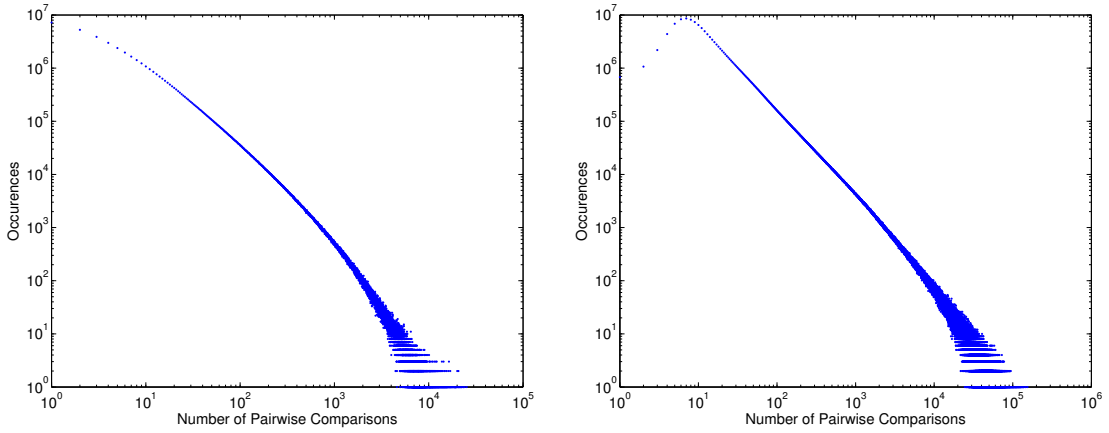
On the other hand, the pairwise ranking matrix that we construct below is invariant under monotone transformation of the rating values and depends only on the degree of relative preference of one alternative over another. It circumvents the previously mentioned pitfalls and is a more principled way to employ a rank/nuclear norm model.

We now describe five techniques to build an aggregate pairwise matrix $\hat{\boldsymbol{Y}}$ from the rating matrix $\boldsymbol{R}$. Let α denote the index of a voter, and i and j the indices of two items. The entries of $\boldsymbol{R}$ are $R_{\alpha i}$. To each voter, we associate a pairwise comparison matrix $\hat{\boldsymbol{Y}}^{\alpha}$. The aggregation is usually computed by something like a mean over $\hat{\boldsymbol{Y}}^{\alpha}$.

1. **Arithmetic mean of score differences** The score difference is $Y_{ij}^{\alpha} = R_{\alpha j} - R_{\alpha i}$. The arithmetic mean of all voters who have rated both i and j is
$$\hat{Y}_{ij} = \frac{\sum_{\alpha}(R_{\alpha i} - R_{\alpha j})}{\#\{\alpha \mid R_{\alpha i}, R_{\alpha j} \text{ exist}\}}.$$
These comparisons are translation invariant.
2. **Geometric mean of score ratios** Assuming $\boldsymbol{R} > 0$, the score ratio refers to $Y_{ij}^{\alpha} = R_{\alpha j}/R_{\alpha i}$. The (log) geometric mean over all voters who have rated both i and j is
$$\hat{Y}_{ij} = \frac{\sum_{\alpha}(\log R_{\alpha i} - \log R_{\alpha j})}{\#\{\alpha \mid R_{\alpha i}, R_{\alpha j} \text{ exist}\}}.$$
These are scale invariant.
3. **Binary comparison** Here $Y_{ij}^{\alpha} = \text{sign}(R_{\alpha j} - R_{\alpha i})$. Its average is the probability difference that the alternative j is preferred to i than vice versa
$$\hat{Y}_{ij} = \Pr\{\alpha \mid R_{\alpha i} > R_{\alpha k}\} - \Pr\{\alpha \mid R_{\alpha i} < R_{\alpha j}\}.$$
These are invariant to a monotone transformation.
4. **Strict binary comparison** This method is almost the same as the last method, except that we eliminate cases where users rated movies equally. That is,
$$\hat{Y}_{ij}^{\alpha} = \begin{cases} 1 & R_{\alpha i} > R_{\alpha j} \\ - & R_{\alpha i} = R_{\alpha j} \\ -1 & R_{\alpha i} < R_{\alpha j}. \end{cases}$$
Again, the average Y_{ij} has a similar interpretation to binary comparison, but only among people who expressed a strict preference for one item over the other. Equal ratings are ignored.
5. **Logarithmic odds ratio** This idea translates binary comparison to a logarithmic scale:
$$\hat{Y}_{ij} = \log \frac{\Pr\{\alpha \mid R_{\alpha i} \geq R_{\alpha j}\}}{\Pr\{\alpha \mid R_{\alpha i} \leq R_{\alpha j}\}}.$$

3. RANK AGGREGATION WITH THE NUCLEAR NORM

Thus far, we have seen how to compute an aggregate pairwise matrix $\hat{\boldsymbol{Y}}$ from ratings data. While $\hat{\boldsymbol{Y}}$ has fewer missing entries than $\boldsymbol{R}$ – roughly 1-80% missing instead of almost 99% missing – it is still not nearly complete. In this section, we discuss how to use the theory of matrix completion to estimate the scoring vector underlying the comparison matrix $\hat{\boldsymbol{Y}}$. These same techniques apply even when $\hat{\boldsymbol{Y}}$ is



(a) MovieLens - 85.49% of total pairwise comparisons

(b) Netflix - 99.77% of total pairwise comparisons

Figure 1: A histogram of the number of pairwise comparisons between movies in MovieLens (left) and Netflix (right). The number of pairwise comparisons is the number of users with ratings on both movies. These histograms show that most items have more than a small number of comparisons between them. For example, 18.5% and 34.67% of all possible pairwise entries have more than 30 comparisons between them. Largely speaking, this figure justifies dropping infrequent ratings from the comparison. This step allows us to take advantage of the ability of the matrix-completion methods to deal with incomplete data.

not computed from ratings and is measured through direct pairwise comparisons.

Let us now state the matrix completion problem formally [Candès and Recht, 2009, Recht et al., 2010]. Given a matrix $\boldsymbol{A}$ where only a subset of the entries are known, the goal is to find the lowest rank matrix $\boldsymbol{X}$ that agrees with $\boldsymbol{A}$ in all the non-zeros. Let Ω be the index set corresponding to the known entries of $\boldsymbol{A}$. Now define $\mathcal{A}(\boldsymbol{X})$ as a linear map corresponding to the elements of Ω, i.e. $\mathcal{A}(\boldsymbol{X})$ is a vector where the ith element is defined to be

$$[\mathcal{A}(\boldsymbol{X})]_i = X_{\omega_i}, \tag{2}$$

and where we interpret X_{ω_i} as the entry of the matrix $\boldsymbol{X}$ for the index pair $(r, s) = \omega_i$. Finally, let $\mathbf{b} = \mathcal{A}(\boldsymbol{Y})$ be the values of the specified entries of the matrix $\boldsymbol{Y}$. This idea of matrix completion corresponds with the solution of

$$\begin{aligned} &\text{minimize} && \text{rank}(\boldsymbol{X}) \\ &\text{subject to} && \mathcal{A}(\boldsymbol{X}) = \mathbf{b}. \end{aligned} \tag{3}$$

Unfortunately, like the direct methods at permutation minimization, this approach is NP-hard [Vandenberghe and Boyd, 1996].

To make the problem tractable, an increasingly well-known technique is to replace the rank function with the nuclear norm [Fazel, 2002]. For a matrix $\boldsymbol{A}$, the nuclear norm is defined

$$\|\boldsymbol{A}\|_* = \sum_{i=1}^{\text{rank}(\boldsymbol{A})} \sigma_i(\boldsymbol{A}) \tag{4}$$

where $\sigma_i(\boldsymbol{A})$ is the ith singular value of $\boldsymbol{A}$. The nuclear norm has a few other names: the Ky-Fan n-norm, the Schatten 1-norm, and the trace norm (when applied to symmetric matrices), but we will just use the term nuclear norm here. It is a convex underestimator of the rank function on the unit spectral-norm ball $\{\boldsymbol{A} : \sigma_{\max}(\boldsymbol{A}) \leq 1\}$, i.e. $\|\boldsymbol{A}\|_* \leq \text{rank}(\boldsymbol{A})\sigma_{\max}(\boldsymbol{A})$ and is the largest convex function with this property. Because the nuclear norm is convex,

$$\begin{aligned} &\text{minimize} && \|\boldsymbol{X}\|_* \\ &\text{subject to} && \mathcal{A}(\boldsymbol{X}) = \mathbf{b} \end{aligned} \tag{5}$$

is a convex relaxation of (3) analogous to how the 1-norm is a convex relaxation of the 0-norm.

In (5) we have $\mathcal{A}(\boldsymbol{X}) = \mathbf{b}$, which is called a noiseless completion problem. Noisy completion problems only require $\mathcal{A}(\boldsymbol{X}) \approx \mathbf{b}$. We present four possibilities inspired by similar approaches in compressed sensing. For the compressed sensing problem with noise:

$$\text{minimize } \|\mathbf{x}\|_1 \quad \text{subject to } \boldsymbol{A}\mathbf{x} \approx \mathbf{b}$$

there are four well known formulations: LASSO [Tibshirani, 1996], QP [Chen et al., 1998], DS [Candès and Tao, 2007] and BPDN [Fuchs, 2004]. For the noisy matrix completion problem, the same variations apply, but with the nuclear norm taking the place of the 1-norm:

LASSO

$$\begin{aligned} &\text{minimize} && \|\mathcal{A}(\boldsymbol{X}) - \mathbf{b}\|_2 \\ &\text{subject to} && \|\boldsymbol{X}\|_* \leq \tau \end{aligned}$$

DS

$$\begin{aligned} &\text{minimize} && \|\boldsymbol{X}\|_* \\ &\text{subject to} && \sigma_{\max}(\mathcal{A}^*(\mathcal{A}(\boldsymbol{X}) - \mathbf{b})) \leq \mu \end{aligned}$$

QP Mazumder et al. [2009]

$$\text{minimize} \quad \|\mathcal{A}(\boldsymbol{X}) - \mathbf{b}\|_2^2 + \lambda \|\boldsymbol{X}\|_*$$

BPDN Mazumder et al. [2009]

$$\begin{aligned} &\text{minimize} && \|\boldsymbol{X}\|_* \\ &\text{subject to} && \|\mathcal{A}(\boldsymbol{X}) - \mathbf{b}\|_2 \leq \sigma \end{aligned}$$

Returning to rank-aggregation, recall the perfect case for the matrix $\boldsymbol{Y}$: there is an unknown quality s_i associated with each item i and $\boldsymbol{Y} = \mathbf{s}\mathbf{e}^T - \mathbf{e}\mathbf{s}^T$. We now assume that

the pairwise comparison matrix computed in the previous section approximates the true $\boldsymbol{Y}$. Given such a $\hat{\boldsymbol{Y}}$, our goal is to complete it with a rank-2 matrix. Thus, our objective:

$$\begin{aligned} &\text{minimize} && \|\mathcal{A}(\boldsymbol{X}) - \mathbf{b}\|_2 \\ &\text{subject to} && \|\boldsymbol{X}\|_* \leq 2 \quad \text{and} \quad \boldsymbol{X} = -\boldsymbol{X}^T \end{aligned} \tag{6}$$

where $\mathcal{A}(\cdot)$ corresponds to the filled entries of $\hat{\boldsymbol{Y}}$. We adopt the LASSO formulation because we want $\text{rank}(\boldsymbol{X}) = 2$, and $\|\boldsymbol{X}\|_*$ underestimates rank as previously mentioned. This problem only differs from the standard matrix completion problem in one regard: the skew-symmetric constraint. With a careful choice of solver, this additional constraint comes "for-free" (with a few technical caveats). It should also be possible to use the skew-Lanczos process to exploit the skew-symmetry in the SVD computation. The problem remains convex because these new equality constraints are linear.

3.1 Algorithms

Algorithms for matrix completion seem to sprout like wildflowers in spring: Lee and Bresler [2009], Cai et al. [2008], Toh and Yun [2009], Dai and Milenkovic [2009], Keshavan and Oh [2009], Mazumder et al. [2009], Jain et al. [2010]. Each algorithm fills a slightly different niche, or improves a performance measure compared to its predecessors.

We first explored crafting our own solver by adapting projection and thresholding ideas used in these algorithms to the skew-symmetrically constrained variant. However, we realized that many algorithms do not require any modification to solve the problem with the skew-symmetric constraint. This result follows from properties of skew-symmetric matrices we show below.

Thus, we use the SVP algorithm by Jain et al. [2010]. For the matrix completion problem, they found their implementation outperformed many competitors. It is scalable and handles a LASSO-like objective for a fixed rank approximation. For completeness, we restate the SVP procedure in Algorithm 1.

Algorithm 1 Singular Value Projection [Jain et al., 2010]: Solve a matrix completion problem. We use the notation $\Omega(\boldsymbol{X})$ to denote output of $\mathcal{A}(\boldsymbol{X})$ when $\mathcal{A}(\cdot)$ is an index set.

INPUT index set Ω, target values $\mathbf{b}$, target rank k, maximum rank k, step length η, tolerance ε

1: Initialize $\boldsymbol{X}^{(0)} = 0$, $t = 0$
2: REPEAT
3: Set $\boldsymbol{U}^{(t)}\boldsymbol{\Sigma}^{(t)}{\boldsymbol{V}^{(t)}}^T$ to be the rank-k SVD of a matrix with non-zeros Ω and values $\Omega(\boldsymbol{X}^{(t)}) - \eta(\Omega(\boldsymbol{X}^{(t)}) - \mathbf{b})$
4: $\boldsymbol{X}^{(t+1)} \leftarrow \boldsymbol{U}^{(t)}\boldsymbol{\Sigma}^{(t)}{\boldsymbol{V}^{(t)}}^T$
5: $t \leftarrow t + 1$
6: UNTIL $\|\Omega(\boldsymbol{X}^{(k)}) - \mathbf{b}\|_2 > \varepsilon$

If the constraint $\mathcal{A}(\boldsymbol{X}) = \mathbf{b}$ comes from a skew-symmetric matrix, then this algorithm produces a skew-symmetric matrix as well. Showing this involves a few properties of skew-symmetric matrices and two lemmas.

We begin by stating a few well-known properties of skew-symmetric matrices. Let $\boldsymbol{A} = -\boldsymbol{A}^T$ be skew-symmetric. Then all the eigenvalues of $\boldsymbol{A}$ are pure-imaginary and come in complex-conjugate pairs. Thus, a skew-symmetric matrix must always have even rank. Let $\boldsymbol{B}$ be a square real-valued matrix, then the closest skew-symmetric matrix to $\boldsymbol{B}$ (in any norm) is $\boldsymbol{A} = (\boldsymbol{B} - \boldsymbol{B}^T)/2$. These results have elementary proofs. We continue by characterizing the singular value decomposition of a skew-symmetric matrix.

LEMMA 1. *Let $\boldsymbol{A} = -\boldsymbol{A}^T$ be an $n \times n$ skew-symmetric matrix with eigenvalues $i\lambda_1, -i\lambda_1, i\lambda_2, -i\lambda_2, \ldots, i\lambda_j, -i\lambda_j$, where $\lambda_i > 0$ and $j = \lfloor n/2 \rfloor$. Then the SVD of $\boldsymbol{A}$ is given by*

$$\boldsymbol{A} = \boldsymbol{U} \begin{bmatrix} \lambda_1 & & & & & & \\ & \lambda_1 & & & & & \\ & & \lambda_2 & & & & \\ & & & \lambda_2 & & & \\ & & & & \ddots & & \\ & & & & & \lambda_j & \\ & & & & & & \lambda_j \end{bmatrix} \boldsymbol{V}^T \tag{7}$$

for $\boldsymbol{U}$ and $\boldsymbol{V}$ given in the proof.

PROOF. Using the Murnaghan-Wintner form of a real matrix [Murnaghan and Wintner, 1931], we can write

$$\boldsymbol{A} = \boldsymbol{X}\boldsymbol{T}\boldsymbol{X}^T$$

for a *real-valued* orthogonal matrix $\boldsymbol{X}$ and *real-valued* block-upper-triangular matrix $\boldsymbol{T}$, with 2-by-2 blocks along the diagonal. Due to this form, $\boldsymbol{T}$ must also be skew-symmetric. Thus, it is a block-diagonal matrix that we can permute to the form:

$$\boldsymbol{T} = \begin{bmatrix} 0 & \lambda_1 & & & & \\ -\lambda_1 & 0 & & & & \\ & & 0 & \lambda_2 & & \\ & & -\lambda_2 & 0 & & \\ & & & & \ddots & \end{bmatrix}.$$

Note that the SVD of the matrix

$$\begin{bmatrix} 0 & \lambda_1 \\ -\lambda_1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_1 \end{bmatrix} \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}.$$

We can use this expression to complete the theorem:

$$\boldsymbol{A} = \underbrace{\boldsymbol{X} \begin{bmatrix} 0 & 1 & & & \\ 1 & 0 & & & \\ & & 0 & 1 & \\ & & 1 & 0 & \\ & & & & \ddots \end{bmatrix}}_{=\boldsymbol{U}} \begin{bmatrix} \lambda_1 & & & & \\ & \lambda_1 & & & \\ & & \lambda_2 & & \\ & & & \lambda_2 & \\ & & & & \ddots \end{bmatrix} \underbrace{\begin{bmatrix} -1 & 0 & & & \\ 0 & 1 & & & \\ & & -1 & 0 & \\ & & 0 & 1 & \\ & & & & \ddots \end{bmatrix} \boldsymbol{X}^T}_{=\boldsymbol{V}^T}.$$

Both the matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ are real and orthogonal. Thus, this form yields the SVD of $\boldsymbol{A}$. □

We now use this lemma to show that – under fairly general conditions – the best rank-k approximation to a skew-symmetric matrix is also skew-symmetric.

LEMMA 2. *Let $\boldsymbol{A}$ be an n-by-n skew-symmetric matrix, and let $k = 2j$ be even. Let $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_j > \lambda_{j+1}$ be the magnitude of the singular value pairs. (Recall that the previous lemma showed that the singular values come in pairs.) Then the best rank-k approximation of $\boldsymbol{A}$ in an orthogonally invariant norm is also skew-symmetric.*

PROOF. This lemma follows fairly directly from Lemma 1. Recall that the best rank-k approximation of $\boldsymbol{A}$ in an orthogonally invariant norm is given by the k largest singular values and vectors. By assumption of the theorem, there is a gap in the spectrum between the kth and k+1-st singular value. Thus, taking the SVD form from Lemma 1 and truncating to the k largest singular values produces a skew-symmetric matrix. □

Finally, we can use this second result to show that the SVP algorithm for the LASSO problem preserves skew-symmetry in all the iterates $\boldsymbol{X}^{(k)}$.

Algorithm 2 Nuclear Norm Rank Aggregation. The SVP subroutine is given by Algorithm 1.

INPUT ranking matrix $\boldsymbol{R}$, minimum comparisons c

1: Compute $\boldsymbol{Y}$ from $\boldsymbol{R}$ by a procedure in Section 2.
2: Discard entries in $\boldsymbol{Y}$ with fewer than c comparisons
3: Let Ω be the index set for all retained entries in $\boldsymbol{Y}$ and $\mathbf{b}$ be the values for these entries
4: $\boldsymbol{U}, \boldsymbol{S}, \boldsymbol{V}$ = SVP(index set Ω, values $\mathbf{b}$, rank 2)
5: Compute $\mathbf{s} = (1/n)\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T\mathbf{e}$

THEOREM 3. *Given a set of skew-symmetric constraints $\mathcal{A}(\cdot) = \mathbf{b}$, the solution of the* LASSO *problem from the* SVP *solver is a skew-symmetric matrix $\boldsymbol{X}$ if the target rank is even and the dominant singular values stay separated as in the previous lemma.*

PROOF. In this proof, we revert to the notation $\mathcal{A}(\boldsymbol{X})$ and use $\mathcal{A}^*(\mathbf{z})$ to denote the matrix with non-zeros in Ω and values from $\mathbf{z}$. We proceed by induction on the iterates generated by the SVP algorithm. Clearly $\boldsymbol{X}^{(0)}$ is skew-symmetric. In step 3, we compute the SVD of a skew-symmetric matrix: $\mathcal{A}^*(\mathcal{A}(\boldsymbol{X}^{(k)}) - \mathbf{b})$. The result, which is the next iterate, is skew-symmetric based on the previous lemma and conditions of this theorem. □

The SVP solver thus solves (6) for a fixed rank problem. A final step is to extract the scoring vector $\mathbf{s}$ from a rank-2 singular value decomposition. If we had the exact matrix $\boldsymbol{Y}$, then $(1/n)\boldsymbol{Y}\mathbf{e} = \mathbf{s} - (\mathbf{s}^T\mathbf{e})/n\mathbf{e}$, which yields the score vector centered around 0. The outcome that a rank-2 $\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ from SVP is not of the form $\mathbf{s}\mathbf{e}^T - \mathbf{e}\mathbf{s}^T$ is quite possible because there are many rank-2 skew-symmetric matrices that do not have $\mathbf{e}$ as a factor. If we had a full-but-inconsistent pairwise comparison matrix $\boldsymbol{Y}$, then using a Borda count $\mathbf{s} = (1/n)\boldsymbol{Y}\mathbf{e}$ provides *the best least-squares approximation* to $\mathbf{s}$ [Jiang et al., 2010]. Formally, $(1/n)\boldsymbol{Y}\mathbf{e} = \operatorname{argmin}_{\mathbf{s}} \left\|\boldsymbol{Y}^T - (\mathbf{s}\mathbf{e}^T - \mathbf{e}\mathbf{s}^T)\right\|$. This discussion justifies using the scoring vector $\mathbf{s} = (1/n)\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T\mathbf{e}$ derived from this completed matrix.

Our complete ranking procedure is given by Algorithm 2.

4. OTHER APPROACHES

Now, we briefly compare our approach with other techniques to compute ranking vectors from pairwise comparison data. An obvious approach is to find the least-squares solution $\min_{\mathbf{s}} \sum_{(i,j)\in\Omega}(Y_{i,j} - (s_i - s_j))^2$. This is a linear least squares method, and is exactly what Massey [1997] proposed for ranking sports teams. The related Colley method introduces a bit of regularization into the least-squares problem [Colley, 2002]. By way of comparison, the matrix completion approach has the same ideal objective, however, we compute solutions using a two-stage process: first complete the matrix, and then extract scores.

A related methodology with skew-symmetric matrices underlies recent developments in the application of Hodge theory to rank aggregation [Jiang et al., 2010]. By analogy with the Hodge decomposition of a vector space, they propose a decomposition of pairwise rankings into *consistent*, *globally inconsistent*, and *locally inconsistent* pieces. Our approach differs because our algorithm applies without restriction on the comparisons. Freeman [1997] also uses an SVD of a skew-symmetric matrix to discover a hierarchical structure in a social network.

We know of two algorithms to directly estimate the item value from ratings [de Kerchov and van Dooren, 2007, Ho and Quinn, 2008]. Both of these methods include a technique to model voter behavior. They find that skewed behaviors and inconsistencies in the ratings require these adjustments. In contrast, we eliminate these problems by using the pairwise comparison matrix. Approaches using a matrix or tensor factorization of the rating matrix directly often have to determine a rank empirically [Rendle et al., 2009].

The problem with the mean rating from Netflix in Table 2 is often corrected by requiring a minimum number of rating on an item. For example, IMDB builds its top-250 movie list based on a Bayesian estimate of the mean with at least 3000 ratings (`imdb.com/chart/top`). Choosing this parameter is problematic as it directly excludes items. In contrast, choosing the minimum number of comparisons to support an entry in $\boldsymbol{Y}$ may be easier to justify.

5. RECOVERABILITY

A hallmark of the recent developments on matrix completion is the existence of theoretical *recoverability* guarantees (see Candès and Recht [2009], for example). These guarantees give conditions under which the solution to the optimization problems posed in Section 3 *is or is nearby* the low-rank matrix from whence the samples originated. In this section, we apply a recent theoretical insight into matrix completion based on operator bases to our problem of recovering a scoring vector from a skew-symmetric matrix [Gross, 2010]. We only treat the noiseless problem to present a simplified analysis. Also, the notation in this section differs slightly from the rest of the manuscript, in order to match the statements in Gross [2010] better. In particular, Ω is not necessarily the index set, $\imath$ represents $\sqrt{-1}$, and most of the results are for the complex field.

The goal is this section is to apply Theorem 3 from Gross [2010] to skew-symmetric matrices arising from score difference vectors. We restate that theorem for reference.

THEOREM 4 (THEOREM 3, GROSS [2010]). *Let $\boldsymbol{A}$ be a rank-r Hermitian matrix with coherence ν with respect to an operator basis $\{\boldsymbol{W}_i\}_{i=1}^{n^2}$. Let $\Omega \subset [1, n^2]$ be a random set of size $|\Omega| > O(nr\nu(1+\beta)(\log n)^2)$. Then the solution of*

$$\begin{aligned} &\text{minimize} && \|\boldsymbol{X}\|_* \\ &\text{subject to} && \operatorname{trace}(\boldsymbol{X}^*\boldsymbol{W}_i) = \operatorname{trace}(\boldsymbol{A}^*\boldsymbol{W}_i) \quad i \in \Omega \end{aligned}$$

is unique and is equal to $\boldsymbol{A}$ with probability at least $1 - n^{-3}$.

The definition of coherence follows shortly. On the surface, this theorem is useless for our application. The matrix we wish to complete is not Hermitian, it's skew-symmetric. However, given a real-valued skew-symmetric matrix $\boldsymbol{Y}$, the matrix $\imath\boldsymbol{Y}$ is Hermitian; and hence, we will work to apply this theorem to this particular Hermitian matrix.

The following theorem gives us a condition for recovering the score vector using matrix completion. As stated, this theorem is not particularly useful because $\mathbf{s}$ may be recovered from noiseless measurements by exploiting the special structure of the rank-2 matrix $\boldsymbol{Y}$. For example, if we know $Y_{i,j} = s_i - s_j$ then given s_i we can find s_j. This argument may be repeated with an arbitrary starting point as long as the known index set corresponds to a connected set over the

indices. Instead we view the following theorem as providing intuition for the noisy problem.

Consider the operator basis for Hermitian matrices:

$$\begin{aligned}
&\mathcal{H} = \mathcal{S} \cup \mathcal{K} \cup \mathcal{D} \text{ where} \\
&\mathcal{S} = \{1/\sqrt{2}(\mathbf{e}_i\mathbf{e}_j^T + \mathbf{e}_j\mathbf{e}_i^T) : 1 \le i < j \le n\}; \\
&\mathcal{K} = \{\imath/\sqrt{2}(\mathbf{e}_i\mathbf{e}_j^T - \mathbf{e}_j\mathbf{e}_i^T) : 1 \le i < j \le n\}; \\
&\mathcal{D} = \{\mathbf{e}_i\mathbf{e}_i^T : 1 \le i \le n\}.
\end{aligned}$$

THEOREM 5. *Let* $\mathbf{s}$ *be centered, i.e.,* $\mathbf{s}^T\mathbf{e} = 0$. *Let* $\boldsymbol{Y} = \mathbf{s}\mathbf{e}^T - \mathbf{e}\mathbf{s}^T$ *where* $\theta = \max_i s_i^2/(\mathbf{s}^T\mathbf{s})$ *and* $\rho = ((\max_i s_i) - (\min_i s_i))/\|\mathbf{s}\|$. *Also, let* $\Omega \subset \mathcal{H}$ *be a random set of elements with size* $|\Omega| \ge O(2n\nu(1+\beta)(\log n)^2)$ *where* $\nu = \max((n\theta+1)/4, n\rho^2)$. *Then the solution of*

$$\begin{aligned}
&\text{minimize} \quad \|\boldsymbol{X}\|_* \\
&\text{subject to} \quad \text{trace}(\boldsymbol{X}^*\boldsymbol{W}_i) = \text{trace}((\imath\boldsymbol{Y})^*\boldsymbol{W}_i), \quad \boldsymbol{W}_i \in \Omega
\end{aligned}$$

is equal to $\imath\boldsymbol{Y}$ *with probability at least* $1 - n^{-\beta}$.

The proof of this theorem follows directly by Theorem 4 if $\imath\boldsymbol{Y}$ has coherence ν with respect to the basis $\mathcal{H}$. We now show this result.

DEFINITION 6 (COHERENCE, GROSS [2010]). *Let* $\boldsymbol{A}$ *be* $n \times n$, *rank-*r, *and Hermitian. Let* $\boldsymbol{U}\boldsymbol{U}^*$ *be an orthogonal projector onto* range$(\boldsymbol{A})$. *Then* $\boldsymbol{A}$ *has coherence* ν *with respect to an operator basis* $\{\boldsymbol{W}_i\}_{i=1}^{n^2}$ *if both*

$$\begin{aligned}
&\max_i \text{trace}(\boldsymbol{W}_i\boldsymbol{U}\boldsymbol{U}^*\boldsymbol{W}_i) \le 2\nu r/n, \text{ and} \\
&\max_i \text{trace}(\text{sign}(\boldsymbol{A})\boldsymbol{W}_i)^2 \le \nu r/n^2.
\end{aligned}$$

For $\boldsymbol{A} = \imath\boldsymbol{Y}$ with $\mathbf{s}^T\mathbf{e} = 0$:

$$\boldsymbol{U}\boldsymbol{U}^* = \frac{\mathbf{s}\mathbf{s}^T}{\mathbf{s}^T\mathbf{s}} - \frac{1}{n}\mathbf{e}\mathbf{e}^T \text{ and } \text{sign}(\boldsymbol{A}) = \frac{1}{\|\mathbf{s}\|\sqrt{n}}\boldsymbol{A}.$$

Let $\boldsymbol{S}_p \in \mathcal{S}$, $\boldsymbol{K}_p \in \mathcal{K}$, and $\boldsymbol{D}_p \in \mathcal{D}$. Note that because sign$(\boldsymbol{A})$ is Hermitian with no real-valued entries, both quantities trace(sign$(\boldsymbol{A})\boldsymbol{D}_i)^2$ and trace(sign$(\boldsymbol{A})\boldsymbol{S}_i)^2$ are 0. Also, because $\boldsymbol{U}\boldsymbol{U}^*$ is symmetric, trace$(\boldsymbol{K}_i\boldsymbol{U}\boldsymbol{U}^*\boldsymbol{K}_p) = 0$. The remaining basis elements satisfy:

$$\begin{aligned}
\text{trace}(\boldsymbol{S}_p\boldsymbol{U}\boldsymbol{U}^*\boldsymbol{S}_p) &= \frac{1}{n} + \frac{s_i^2 + s_j^2}{2\mathbf{s}^T\mathbf{s}} \le (1/n) + \theta \\
\text{trace}(\boldsymbol{D}_p\boldsymbol{U}\boldsymbol{U}^*\boldsymbol{D}_p) &= \frac{1}{n} + \frac{s_i^2}{\mathbf{s}^T\mathbf{s}} \le (1/n) + \theta \\
\text{trace}(\text{sign}(\boldsymbol{A})\boldsymbol{K}_p)^2 &= \frac{2(s_i - s_j)^2}{n\mathbf{s}^T\mathbf{s}} \le (2/n)\rho^2.
\end{aligned}$$

Thus, $\boldsymbol{A}$ has coherence ν with ν from Theorem 5 and with respect to $\mathcal{H}$. And we have our recovery result. Although, this theorem provides little practical benefit unless both θ and ρ are $O(1/n)$, which occurs when $\mathbf{s}$ is nearly uniform.

6. RESULTS

We implemented and tested this procedure in two synthetic scenarios, along with Netflix, movielens, and Jester joke-set ratings data. In the interest of space, we only present a subset of these results for Netflix.

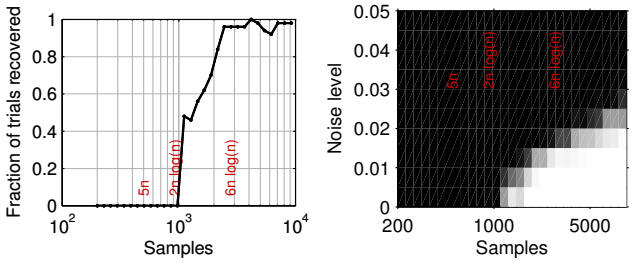


Figure 2: An experimental study of the recoverability of a ranking vector. These show that we need about 6n log n entries of $\boldsymbol{Y}$ to get good recovery in both the noiseless (left) and noisy (right) case. See §6.1 for more information.

6.1 Recovery

The first experiment is an empirical study of the recoverability of the score vector in the noiseless and noisy case. In the noiseless case, Figure 2 (left), we generate a score vector with uniformly distributed random scores between 0 and 1. These are used to construct a pairwise comparison matrix $\boldsymbol{Y} = \mathbf{s}\mathbf{e}^T - \mathbf{e}\mathbf{s}^T$. We then sample elements of this matrix uniformly at random and compute the difference between the true score vector $\mathbf{s}$ and the output of steps 4 and 5 of Algorithm 2. If the relative 2-norm difference between these vectors is less than 10^{-3}, we declare the trial recovered. For $n = 100$, the figure shows that, once the number of samples is about $6n\log n$, the correct $\mathbf{s}$ is recovered in nearly all the 50 trials.

Next, for the noisy case, we generate a uniformly spaced score vector between 0 and 1. Then $\boldsymbol{Y} = \mathbf{s}\mathbf{e}^T - \mathbf{e}\mathbf{s}^T + \varepsilon\boldsymbol{E}$, where $\boldsymbol{E}$ is a matrix of random normals. Again, we sample elements of this matrix randomly, and declare a trial successful if the *order* of the recovered score vector is identical to the true order. In Figure 2 (right), we indicate the fractional of successful trials as a gray value between black (all failure) and white (all successful). Again, the algorithm is successful for a moderate noise level, i.e., the value of ε, when the number of samples is larger than $6n\log n$.

6.2 Synthetic

Inspired by Ho and Quinn [2008], we investigate recovering item scores in an item-response scenario. Let a_i be the center of user i's rating scale, and b_i be the rating sensitivity of user i. Let t_i be the intrinsic score of item j. Then we generate ratings from users on items as:

$$R_{i,j} = L[a_i + b_i t_j + E_{i,j}]$$

where $L[\alpha]$ is the discrete levels function:

$$L[\alpha] = \max(\min(\text{round}(\alpha), 5), 1)$$

and $E_{i,j}$ is a noise parameter. In our experiment, we draw $a_i \sim N(3,1)$, $b_i \sim N(0.5, 0.5)$, $t_i \sim N(0.1, 1)$, and $E_{i,j} \sim \varepsilon N(0,1)$. Here, $N(\mu, \sigma)$ is a standard normal, and ε is a noise parameter. As input to our algorithm, we sample ratings uniformly at random by specifying a desired number of average ratings per user. We then look at the Kendall τ correlation coefficient between the true scores t_i and the output of our algorithm using the arithmetic mean pairwise aggregation. A τ value of 1 indicates a perfect ordering correlation between the two sets of scores.

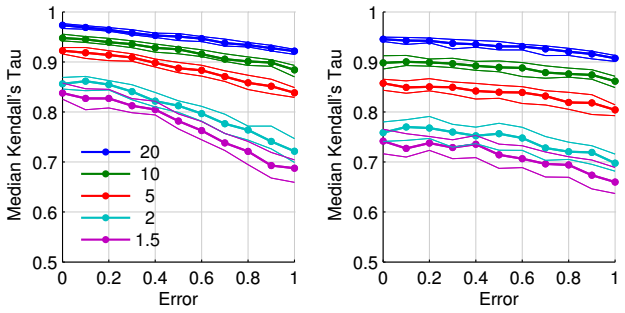


Figure 3: The performance of our algorithm (left) and the mean rating (right) to recovery the ordering given by item scores in an item-response theory model with 100 items and 1000 users. The various thick lines correspond to average number of ratings each user performed (see the in place legend). See §6.2 for more information

Figure 3 shows the results for 1000 users and 100 items with $1.1, 1.5, 2, 5$, and 10 ratings per user on average. We also vary the parameter ε between 0 and 1. Each thick line with markers plots the median value of τ in 50 trials. The thin adjacency lines show the 25th and 75th percentiles of the 50 trials. At all error levels, our algorithm outperforms the mean rating. Also, when there are few ratings per-user and moderate noise, our approach is considerably more correlated with the true score. This evidence supports the anecdotal results from Netflix in Table 2.

6.3 Netflix

See Table 2 for the top movies produced by our technique in a few circumstances using all users. The arithmetic mean results in that table use only elements of $\boldsymbol{Y}$ with at least 30 pairwise comparisons (it is a `am all 30` model in the code below). And see Figure 4 for an analysis of the residuals generated by the fit for different constructions of the matrix $\hat{\boldsymbol{Y}}$. Each residual evaluation of Netflix is described by a code. For example, `sb all 0` is a strict-binary pairwise matrix $\hat{\boldsymbol{Y}}$ from all Netflix users and $c = 0$ in Algorithm 2 (i.e. accept all pairwise comparisons). Alternatively, `am 6 30` denotes an arithmetic-mean pairwise matrix $\hat{\boldsymbol{Y}}$ from Netflix users with at least 6 ratings, where each entry in $\hat{\boldsymbol{Y}}$ had 30 users supporting it. The other abbreviations are `gm`: geometric mean; `bc`: binary comparison; and `lo`: log-odds ratio.

These residuals show that we get better rating fits by only using frequently compared movies, but that there are only minor changes in the fits when excluding users that rate few movies. The difference between the score-based residuals $\left\|\Omega(\mathbf{s}\mathbf{e}^T - \mathbf{e}\mathbf{s}^T) - \mathbf{b}\right\|$ (red points) and the SVP residuals $\left\|\Omega(\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T) - \mathbf{b}\right\|$ (blue points) show that excluding comparisons leads to "overfitting" in the SVP residual. This suggests that increasing the parameter c should be done with care and good checks on the residual norms.

To check that a rank-2 approximation is reasonable, we increased the target rank in the SVP solver to 4 to investigate. For the arithmetic mean (6,30) model, the relative residual at rank-2 is 0.2838 and at rank-4 is 0.2514. Meanwhile, the nuclear norm increases from around 14000 to around 17000. These results show that the change in the fit is minimal and our rank-2 approximation and its scores should represent a reasonable ranking.

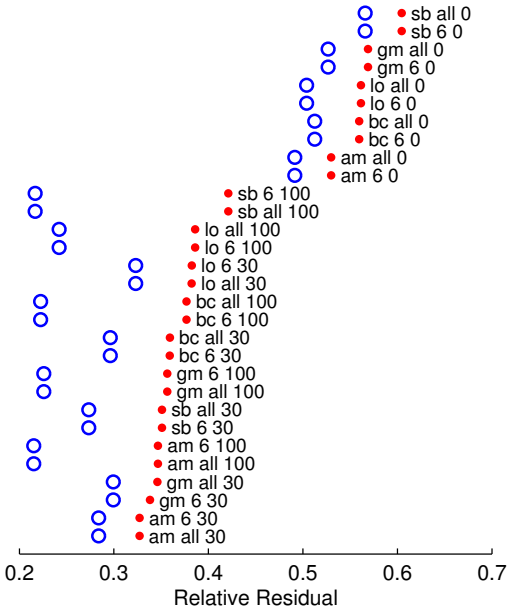


Figure 4: The labels on each residual show how we generated the pairwise scores and truncated the Netflix data. Red points are the residuals from the scores, and blue points are the final residuals from the SVP algorithm. Please see the discussion in §6.3.

7. CONCLUSION

Existing principled techniques such as computing a Kemeny optimal ranking or finding a minimize feedback arc set are NP-hard. These approaches are inappropriate in large scale rank aggregation settings. Our proposal is (i) measure pairwise scores $\hat{\boldsymbol{Y}}$ and (ii) solve a matrix completion problem to determine the quality of items. This idea is both principled and functional with significant missing data. The results of our rank aggregation on the Netflix problem (Table 2) reveal popular and high quality movies. These are interesting results and could easily have a home on a "best movies in Netflix" web page. Such a page exists, but is regarded as having strange results. Computing a rank aggregation with this technique is not NP-hard. It only requires solving a convex optimization problem with a unique global minima. Although we did not record computation times, the most time consuming piece of work is computing the pairwise comparison matrix $\boldsymbol{Y}$. In a practical setting, this could easily be done with a MapReduce computation.

To compute these solutions, we adapted the SVP solver for matrix completion [Jain et al., 2010]. This process involved (i) studying the singular value decomposition of a skew-symmetric matrix (Lemmas 1 and 2) and (ii) showing that the SVP solver preserves a skew-symmetric approximation through its computation (Theorem 3). Because the SVP solver computes with an explicitly chosen rank, these techniques work well for large scale rank aggregation problems.

We believe the combination of pairwise aggregation and matrix completion is a fruitful direction for future research. We plan to explore optimizing the SVP algorithm to exploit the skew-symmetric constraint, extending our recovery result to the noisy case, and investigating additional data.

Acknowledgements. The authors would like to thank Amy Langville, Carl Meyer, and Yuan Yao for helpful discussions.

References

N. Ailon, M. Charikar, and A. Newman. Aggregating inconsistent information: ranking and clustering. In *STOC '05*, pages 684–693, 2005. doi: 10.1145/1060590.1060692.

N. Alon. Ranking tournaments. *SIAM J. Discret. Math.*, 20 (1):137–142, 2006. doi: 10.1137/050623905.

K. J. Arrow. A difficulty in the concept of social welfare. *J. Polit. Econ.*, 58(4):328–346, 1950. http://jstor.org/stable/1828886.

J.-F. Cai, E. J. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. *arXiv*, math.OC:0810.3286v1, 2008. http://arxiv.org/abs/0810.3286.

E. Candès and T. Tao. The Dantzig selector: Statistical estimation when p is much larger than n. *Ann. Stat.*, 35(6): 2313–2351, 2007. doi: 10.1214/009053606000001523.

E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Found. Comput. Math.*, 9(6):717–772, 2009. doi: 10.1007/s10208-009-9045-5.

E. J. Candès and T. Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Trans. Inform. Theory*, 56(5):2053–2080, 2010. doi: 10.1109/TIT.2010.2044061.

S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM J. Sci. Comp.*, 20(1): 33–61, 1998. doi: 10.1137/S1064827596304010.

W. N. Colley. Colley's bias free college football ranking method: The Colley matrix explained. Technical report, Princeton University, 2002.

J.-A.-N. d. C. Condorcet. *Essai sur l'application de l'analyse à la probabilité des décisions...* de L'imprimerie Royale, Paris, 1785. http://gallica2.bnf.fr/ark:/12148/bpt6k417181.

W. Dai and O. Milenkovic. Set: an algorithm for consistent matrix completion. *arXiv*, 2009. http://arxiv.org/abs/0909.2705.

H. A. David. *The method of paired comparisons*. Number 41 in Griffin's Statistical Monographs and Courses. Charles Griffin, 1988. ISBN 0195206169.

C. de Kerchov and P. van Dooren. Iterative filtering for a dynamical reputation system. *arXiv*, cs.IR:0711.3964, 2007. http://arXiv.org/abs/0711.3964.

C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *WWW '01*, pages 613–622, New York, NY, USA, 2001. ACM. ISBN 1-58113-348-0. doi: 10.1145/371920.372165.

M. Fazel. *Matrix rank minimization with applications*. PhD thesis, Stanford University, 2002. http://faculty.washington.edu/mfazel/thesis-final.pdf.

L. C. Freeman. Uncovering organizational hierarchies. *Computational and Mathematical Organization Theory*, 3 (1):5–18, 1997. doi: 10.1023/A:1009690520577.

J.-J. Fuchs. Recovery of exact sparse representations in the presence of noise. In *ICASSP '04*, volume 2, pages ii–533–6 vol.2, 2004. doi: 10.1109/ICASSP.2004.1326312.

D. Gross. Recovering low-rank matrices from few coefficients in any basis. *arXiv*, cs.NA:0910.1879v5, 2010. http://arxiv.org/abs/0910.1879.

D. E. Ho and K. M. Quinn. Improving the presentation and interpretation of online ratings data with model-based figures. *Amer. Statist.*, 62(4):279–288, 2008. doi: 10.1198/000313008X366145.

P. Jain, R. Meka, and I. Dhillon. Guaranteed rank minimization via singular value projection. In *NIPS 23*, pages 937–945, 2010. http://books.nips.cc/papers/files/nips23/NIPS2010_0682.pdf.

X. Jiang, L.-H. Lim, Y. Yao, and Y. Ye. Statistical ranking and combinatorial hodge theory. *Mathematical Programming*, 127(1):1–42, 2010. doi: 10.1007/s10107-010-0419-x.

J. G. Kemeny. Mathematics without numbers. *Daedalus*, 88 (4):577–591, 1959. http://jstor.org/stable/20026529.

M. G. Kendall and B. B. Smith. On the method of paired comparison. *Biometrika*, 31(3-4):324–345, 1940. doi: 10.1093/biomet/31.3-4.324.

R. H. Keshavan and S. Oh. A gradient descent algorithm on the grassman manifold for matrix completion. *arXiv*, 2009. http://arxiv.org/abs/0910.5260.

K. Lee and Y. Bresler. Admira: Atomic decomposition for minimum rank approximation. *arXiv*, 2009. http://arxiv.org/abs/0905.0044.

H. Li, T.-Y. Liu, and C. Zhai, editors. *SIGIR '08 Workshop: Learning to Rank for Information Retrieval*. 2008. http://research.microsoft.com/en-us/um/beijing/events/lr4ir-2008/PROCEEDINGS-LR4IR%202008.PDF.

K. Massey. Statistical models applied to the rating of sports teams. Master's thesis, Bluefield College, 1997.

R. Mazumder, T. Hastie, and R. Tibshirani. Regularization methods for learning incomplete matrices. *arXiv*, 2009. http://arxiv.org/abs/0906.2034v1.

G. A. Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychol. Rev.*, 101(2):343–352, 1956. doi: 10.1037/h0043158.

F. D. Murnaghan and A. Wintner. A canonical form for real matrices under orthogonal transformations. *PNAS*, 17 (7):417–420, 1931. http://www.pnas.org/content/17/7/417.full.pdf+html.

B. Recht, M. Fazel, and P. A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*, 52(3):471–501, 2010. doi: 10.1137/070697835.

S. Rendle, L. Balby Marinho, A. Nanopoulos, and L. Schmidt-Thieme. Learning optimal ranking with tensor factorization for tag recommendation. In *KDD '09*, pages 727–736. ACM, 2009. doi: 10.1145/1557019.1557100.

T. L. Saaty. Rank according to Perron: A new insight. *Math. Mag*, 60(4):211–213, 1987. http://jstor.org/stable/2689340.

P.-N. Tan and R. Jin. Ordering patterns by combining opinions from multiple sources. In *KDD '04*, pages 695–700. ACM, 2004. doi: 10.1145/1014052.1014142.

R. Tibshirani. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 58(1):267–288, 1996. http://www.jstor.org/stable/2346178.

K.-C. Toh and S. Yun. An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. *Opt. Online*, 2009. http://www.optimization-online.org/DB_FILE/2009/03/2268.pdf.

L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Rev.*, 38(1):49–95, 1996. doi: 10.1137/1038003.