Ranking from Crowdsourced Pairwise Comparisons via Smoothed Matrix Manifold Optimization

Jialin Dong, Kai Yang, and Yuanming Shi

School of Information Science and Technology, ShanghaiTech University, Shanghai, China
E-mail: {dongjl, yangkai, shiym}@shanghaitech.edu.cn

***Abstract*—As the blooming development of data mining in social computing systems (e.g., crowdsourcing system), statistical inference from crowdsourced data severs as a powerful tool to provide diversified services. To support critical applications (e.g., recommendation), in this paper, we shall focus on the collaborative ranking problems and construct a system of which the input is crowdsourced pairwise comparisons and the output is individual rankings. Under the Bradley-Terry-Luce (BTL) parametric model assumption, we present a maximum likelihood estimation (MLE) based on low-rank approach to estimate the underlying weight/score matrix, thereby predicting the individual ranking for each user. To address the unique challenge of the coupled non-convex low-rank constraint and the non-smooth elementwise infinity norm constraint in the resulting MLE problem, we propose a novel regularized formulation with the smoothed surrogate of elementwise infinity norm. By further exploiting the geometry of quotient manifolds of fixed-rank matrices, we solve the resulting smoothed rank-constrained optimization problem via developing the Riemannian trust-region algorithm which converges to an *approximate local minimum* from *arbitrary initial points*. Numerical results demonstrate the extraordinary effectiveness of the proposed method compared with the state-of-art algorithms.**

I. Introduction

Crowdsourcing is one of the most ubiquitous computing systems empowering crowded users to involve in social interaction, share their creativity, devote their ingeniousness, distribute information and collectively handle complicated issues [1]. Over the past decade, it has emerged as a powerful and low-cost tool to deal with large-scale data for data mining [1]. In particular, there is rich literature on statistical inference from crowdsourced data, such as evaluating the machine learning models, clustering data and scening recognition [1].

In recent years, there is a growing body of works on information recovery based on pairwise measurements which spans various fields, such as pairwise difference for community detection [2] and pairwise distance for localization [3]. In particular, pairwise comparison has been exploited for ranking prediction [4] with applications such as user preference prediction in recommendation systems [5]. Compared with the conventional numerical measurement in ranking problems, pairwise comparison measurement has advantages in statistics [6] and manageability. In this paper, we are particularly interested in the problem of ranking prediction from pairwise comparisons in crowdsourcing system. To further reduce the overhead of data collection, only partial pairwise comparisons are required.

After collecting pairwise measurements in crowdsourcing system, underlying information needs to be revealed. For ranking problem, aggregate ranking [7] and collaborative ranking [5] are two typical problems by assuming the existence of the underlying preference weight/score vector and matrix respectively. A line of works focuses on estimating matrix or vector from parametric models, such as Bradley-Terry-Luce (BTL) model [7], [8], strong stochastic transitivity (SST) [4] model and mixture model [4]. In this paper, to support efficient algorithm design, to allow the heterogeneity of the workers in social computing system, as well as to exploit diversity of preferences among items, we assume that the pairwise comparison measurements follow the BTL model with weight/score matrix as the parameters. Furthermore, the underlying preference weight matrix in the BTL model is assumed to be low-rank based on the fact that preferences are only affected by a few factors [9]. This property improves the possibility of recovering the exact ranking lists only based on partial pairwise measurements.

In this paper, we present the maximum likelihood estimation (MLE) approach to estimate the underlying weight/ score matrix under the BTL model, followed by the individual rankings recovery based on the estimated weight/score matrix. Therein, elementwise infinity norm constraint is introduced to avoid the excessive "spikiness" of the score matrix. To address this non-convex and highly intractable low-rank optimization problem, inspired by the recent works [10], [11], we propose a smoothed Riemannian trust-region algorithm to solve the low-rank optimization problem. The proposed Riemannian trust-region algorithm converges to an *approximate local minimum* $\boldsymbol{x}^*$ [10], which satisfies $\|\nabla f(\boldsymbol{x}^*)\| \leq \varepsilon$ and $\nabla^2 f(\boldsymbol{x}^*) \succeq -\sqrt{\varepsilon}\boldsymbol{I}$ with a sufficiently small ε [12], from *arbitrary initial points*. The proposed algorithm enjoys the faster convergence rate and better performance than state-of-art algorithms, such as spectral project-gradient (SPG) algorithm [13] and Bi-factor gradient descent (BFGD) [14].

We summarize the major contributions to the ranking problem from crowdsourced pairwise comparisons as follow:

- We present a low-rank optimization approach with pairwise measurements for ranking problem. To address the unique challenge of coupled non-convex fixed-rank

2375-9259/17 $31.00 © 2017 IEEE
DOI 10.1109/ICDMW.2017.130

constraint and non-smooth elementwise infinity norm constraint, we recast the original problem as a rank-constrained smoothed regularized optimization problem via smoothing the elementwise infinity norm function.
- To adopt the versatile framework of Riemannian optimization, we generalize the classical geometric concepts (e.g., the notion of length, set of directional derivatives, motion along geodesics, gradient and Hessian) in the Euclidean space to the geometric concepts (e.g., the notion of Riemannian metric, tangent space, retraction, Riemannian gradient and Riemannian Hessian) on the quotient manifold of fixed-rank matrices.
- To reduce the computational cost and achieve good performance, we develop the smoothed Riemannian trust-region algorithm to solve the rank-constrained smoothed optimization problem. It turns out that the proposed smoothed Riemannian optimization algorithm converges to an *approximate local minimum* from *arbitrary initial points*.

Simulation results will show the proposed smoothed regularized approach supported by Riemannian trust-region algorithm outperforms the state-of-art algorithms in terms of both performance and computational efficiency.

II. Related work

A growing body of works has clear and strong theoretical guarantees on the semidefinite programming approach (SDP) via nuclear norm relaxation for low-rank matrix optimization [5], [8], [13]. However, the computational and memory requirement for solving an SDP problem is prohibitive from being moderated to high-dimensional data problem. This motivates the development of matrix factorization $\boldsymbol{X} \in \mathbb{R}^{m\times n}$ as $\boldsymbol{U}\boldsymbol{V}^T$ where $\boldsymbol{U} \in \mathbb{R}^{m\times r}$ and $\boldsymbol{V} \in \mathbb{R}^{n\times r}$ to reduce the computational and storage cost. Several papers e.g.,[11], have provided various algorithms to address this non-convex optimization problem with respect to $\boldsymbol{U}$ and $\boldsymbol{V}$.

Specifically, a line of works on low-rank matrix optimization (e.g., [15], [16], [17]) showed that based on a good enough initial point, basic local search algorithms, including gradient descent (GD), stochastic gradient descent (SGD), alternating minimization (AltMin) and block coordinate descent (BCD) methods, enjoy fast local convergence to global minimum. In addition, the work [14] proposed well-initialized Bi-factor gradient descent (BFGD) algorithm converging to the rank-r approximation to the underlying matrix. However, works like [10], [18], [19], [20], [21] eschew the need for careful initialization procedures while still achieve theoretical guarantees.

In particular, the recent work [11] showed Riemmanian trust-region (RTR) algorithm globally converges to an *approximate local minimum* $\boldsymbol{x}^*$ on manifolds from *arbitrary initial points*. In this paper, motivated by the benefits of computational efficiency, remarkable convergence results and initialization robustness, we thus exploit the Riemannian trust-region algorithm in this paper.

III. System Model And Problem Formulation

In this section, we present the data model and problem formulation under the framework of crowdsourcing system.

A. Data model

Consider a crowdsourcing system where the organizer distributes the task of providing preferences among n items to m crowd users. Instead of the generic numerical measurement, we collect pairwise comparisons $\{Y_{ijk} \in \{1, -1\} : (i, j, k) \in \Omega\}$ with $\Omega \subseteq [m] \times [n] \times [n]$ as the observation set, where $[n]$ represents the set $\{1, 2, \cdots, n\}$. Here, $Y_{ijk} = 1$ denotes that the user i prefers item j to item k, otherwise $Y_{ijk} = -1$. The main purpose for employing pairwise comparisons is to address the inconsistencies among various users. Due to the lack of standardization of numerical rating system, it is arduous to normalize data provided by distinct workers, which induces uncertainties [6].

We analyze the observation variables under the well-known BTL model associated with the logistic distribution [22]. The logistic function is given by $f(z) = \frac{1}{1+\exp(\frac{-z}{\sigma})}$, where the parameter $\sigma > 0$. Specifically, let $\boldsymbol{X} \in \mathbb{R}^{m\times n}$ be the underlying preference score/weight matrix, then the pairwise comparison outcome between item j and k provided by user i is given by [13]

$$Y_{ijk} = \begin{cases} +1 & \text{w.p.} \;\; f(\Delta_{ijk}) \\ -1 & \text{w.p.} \;\; 1 - f(\Delta_{ijk}) \end{cases} \quad \forall\ (i, j, k) \in \Omega, \tag{1}$$

where $\Delta_{ijk} = X_{ij} - X_{ik}$; "w.p." is short for "with probability" and Ω is the index set of the obtained pairwise comparison measurements. Note that the observations are assumed to be independent with each other.

In this paper, we are interested in the individual rankings recovery problem, for which, we introduce an associated score $\tau_j^{(i)}$ for each user $i \in [m]$ over the item $j \in [n]$, which is defined as [4]

$$\tau_j^{(i)} := \frac{1}{n} \sum_{k=1}^{n} f(\Delta_{ijk}). \tag{2}$$

That is to say, the score $\tau_j^{(i)}$ associated with user i represents the probability that item j is preferred to an item chosen uniformly at random among all n items. Assume that the scores $\tau_j^{(i)}$ are strictly distinct from each other for each user with high probability. Then a ranking list for user i over a set of n items is given by a mapping $\pi : [n] \to [n]$ such that

$$\tau_{\pi(1)}^{(i)} > \tau_{\pi(2)}^{(i)} > \cdots > \tau_{\pi(n)}^{(i)}, \tag{3}$$

where $\pi(k)$ denotes the k-th ranked item according to the scores derived from (2).

Our goal is to recover the individual rankings (3) for all users. This is achieved by estimating the weight matrix $\boldsymbol{X}$ from pairwise comparisons [5]. The weight matrix is further assumed to be low-rank, which is based on the fact that only a small number of factors affect the preference [23].

B. Maximum-Likelihood Estimation of Weight Matrix

We adopt the maximum-likelihood estimation (MLE) method to estimate the weight matrix $\boldsymbol{X}$ with partial pairwise measurements. Based on the BTL model for the pairwise comparisons (1), the negative log-likelihood function is given by [13]

$$\mathcal{L}_{\Omega,\boldsymbol{Y}}(\boldsymbol{X}) = -\sum_{(i,j,k)\in\Omega} \big\{\mathbb{I}_{(Y_{ijk}=1)}\log(f(\Delta_{ijk})) + \mathbb{I}_{(Y_{ijk}=-1)}\log(1-f(\Delta_{ijk}))\big\}, \quad (4)$$

where $\boldsymbol{Y} \in \{1,-1\}^{m\times n\times n}$ represents observed pairwise comparisons and $\mathbb{I}_\mu$ denotes the indicator function, i.e., $\mathbb{I}_\mu = 1$ when the event μ is true, otherwise, $\mathbb{I}_\mu = 0$. To recover the low-rank weight matrix $\boldsymbol{X}$, we minimize the negative log-likelihood function under the exact rank constraint as follows:

$$\begin{aligned} &\underset{\boldsymbol{X}\in\mathbb{R}^{m\times n}}{\text{minimize}} \quad \mathcal{L}_{\Omega,\boldsymbol{Y}}(\boldsymbol{X}) \\ &\text{subject to} \quad \text{rank}(\boldsymbol{X}) = r, \end{aligned} \quad (5)$$

where $r \ll \min\{m,n\}$ is the prior information denoting the rank of weight matrix. Note that $\mathcal{L}_{\Omega,\boldsymbol{Y}}(\boldsymbol{X})$ can be further written as $\mathcal{L}_{\Omega,\boldsymbol{Y}}(\boldsymbol{X}) = -\sum_{(i,j,k)\in\Omega}\log(f(Y_{ijk}(X_{ij}-X_{ik})))$.

Furthermore, to avoid the excessive "spikiness" of the matrix and ill-posedness of problem (5), we impose the elementwise infinity norm constraint to bound the magnitude of each element in matrix $\boldsymbol{X}$ [8]. The estimation problem (5) thus can be rewritten as

$$\begin{aligned} &\underset{\boldsymbol{X}\in\mathbb{R}^{m\times n}}{\text{minimize}} \quad \mathcal{L}_{\Omega,\boldsymbol{Y}}(\boldsymbol{X}) \\ &\text{subject to} \quad \text{rank}(\boldsymbol{X}) = r, \quad \|\boldsymbol{X}\|_\infty \le \alpha, \end{aligned} \quad (6)$$

where $\alpha > 0$ is an arbitrary reasonable parameter and $\|\boldsymbol{X}\|_\infty = \max_{i,j}|X_{ij}|$ is the elementwise infinity norm. However, problem (6) is non-convex due to the fixed-rank constraint. In this paper, we aim at providing efficient algorithms to solve this non-convex estimation problem with near-optimal performance.

C. Problem Analysis

The original problem (6) is NP-hard due to the fixed-rank constraint. In this subsection, we present the existing methods for solving the low-rank optimization problems and analyze their limitations.

Convex relaxation approach. Nuclear norm was adopted in [5], [8], [13] to serve as a convex relaxation for both rank constraint and elementwise infinity norm constraint, yielding the following formulation:

$$\begin{aligned} &\underset{\boldsymbol{X}\in\mathbb{R}^{m\times n}}{\text{minimize}} \quad \mathcal{L}_{\Omega,\boldsymbol{Y}}(\boldsymbol{X}) \\ &\text{subject to} \quad \|\boldsymbol{X}\|_* \le \alpha\sqrt{rmn}, \end{aligned} \quad (7)$$

where $\|\boldsymbol{X}\|_*$ denotes the nuclear norm of $\boldsymbol{X}$. Note that the estimated matrix $\boldsymbol{X}$ is required to be scaled to $\|\boldsymbol{X}\|_\infty = \alpha$ to ensure the elementwise infinity norm constraint in (6). Though problem (7) can be formulated as a semidefinite programming (SDP), the computational cost of solving SDP often limits applicability to large-dimensional data set. This challenge motivates the development of scalable computational methods, such as non-monotone spectral projected-gradient (SPG) method [13] based on the projected gradient descent method, Newton-ADMM method [24] and splitting conic solver (SCS) [25], both of which is based on projections onto positive semidefinite cone with respect to problem (7). However, calculating projections via singular value decomposition is too computationally expensive at each iteration. The extension of such convex paradigms to large-dimensional data is still inapplicable.

Non-convex Optimization Paradigms. Non-convex optimization algorithms are ubiquitous in a line of recent works [14], [11] for practical applications due to the low computational complexity via matrix factorization (i.e., factoring $\boldsymbol{X}$ as $\boldsymbol{U}\boldsymbol{V}^T$, where $\boldsymbol{U} \in \mathbb{R}^{m\times r}$ and $\boldsymbol{V} \in \mathbb{R}^{n\times r}$). The common method guaranteeing the elementwise infinity norm constraint is to utilize log-barrier penalty function [26], [27, Section 11.2]. Thus, the original problem (6) can be reformulated to the following non-convex optimization problem [28]:

$$\underset{\boldsymbol{U}\in\mathbb{R}^{m\times r},\boldsymbol{V}\in\mathbb{R}^{n\times r}}{\text{minimize}} \mathcal{L}_{\Omega,\boldsymbol{Y}}(\boldsymbol{U}\boldsymbol{V}^T) - \frac{1}{\tau}R(\boldsymbol{U},\boldsymbol{V}), \quad (8)$$

where $R(\boldsymbol{U},\boldsymbol{V}) = \sum_{a,b}\log(1-(\boldsymbol{U}_{a,\cdot}\boldsymbol{V}_{b,\cdot}/\alpha)^2)$ and $\boldsymbol{U}_{a,\cdot}$ denotes the a-th row of $\boldsymbol{U}$ and $\boldsymbol{V}_{b,\cdot}$ denotes the b-th row of $\boldsymbol{V}$. The parameter τ determines the tightness of approximation of elementwise infinity norm constraint via the log-barrier function. This problem can be resolved by the log-barrier method [27] via solving a sequence of convex problems with the gradient descent algorithm [28]. This can be achieved by the Bi-factor gradient descent algorithm (BFGD) via updating factorizations simultaneously [14]. Note that the estimated matrix $\boldsymbol{X}$ is required to scale to $\|\boldsymbol{X}\| = \alpha$. However, the outer iteration of log-barrier method increases the computational complexity. The first-order method, BFGD, also yields slow convergence rate.

In this paper, we exploit the quotient manifold of fixed-rank matrices to remove the indeterminacy for matrix factorization. Then we shall develop the Riemannian trust-region algorithm under the Riemannian optimization framework satisfying the Lipschitz-type assumptions. The algorithm guarantees to globally return an *approximate local minimum* [12]. However, challenge arises due to the additional non-smooth elementwise infinity norm constraint when pursuing to satisfy Lipschitz-type assumptions provided in [10]. The procedure of addressing this issue will be discussed in next section.

IV. Regularized Smoothed MLE for Score Matrix Estimation via Riemannian Optimization

The commonly used log-barrier penalty regularization method [27], [28] fails to satisfy the requirement for implementing Riemannian trust-region algorithms (i.e., Lipschitz gradient and Lipschitz Hessian) due to infiniteness near the boundary of feasible set. Thus, we recast the low-rank MLE problem (6) to the smoothed regularized version in order to develop matrix manifold optimization [29] in this section.

A. Computational Opportunities via Smoothing Methods

Based on Theorem 4.2 in [30], we choose $\log \sum_{ij} e^{X_{ij}^2}$ as the smoothed surrogate of $\|\boldsymbol{X}\|_\infty^2$ to assure the constraint $\|\boldsymbol{X}\|_\infty^2 \leq \alpha^2$ in problem (6). As a result, the regularized smoothed version of problem (6) can be written as

$$\mathscr{P}: \underset{\boldsymbol{X}\in\mathcal{M}}{\text{minimize}} \quad F(\boldsymbol{X}) := \mathcal{L}_{\Omega,\boldsymbol{Y}}(\boldsymbol{X}) + \lambda \log N(\boldsymbol{X}), \quad (9)$$

where $N(\boldsymbol{X}) = \sum_{i,j} e^{X_{ij}^2}$ and $\lambda = r^2\sqrt{K}\log K$ is a constant regularized parameter to well approximate problem (6) [31]. Note that the estimated matrix $\boldsymbol{X}$ needs to be scaled to $\|\boldsymbol{X}\|_\infty = \alpha$. The complicated structure of the objective function (9) yields unique challenges in generalizing geometric concepts in the Euclidean space to the geometric concepts on the quotient manifold of fixed-rank matrices. The effort that we make to compute matrix manifold optimization related ingredients will be presented in Section V-B.

Moreover, the objective function $F(\boldsymbol{X})$ is smooth and convex over the compact convex set in the Euclidean space $\mathbb{R}^{m\times n}$, inducing the *Lipschitz continuous gradient* and *Lipschitz continuous Hessian* in manifold $\mathcal{M}$ [10]. The smooth convex structure of problem (9) lays the foundation to develop sophisticated algorithms on the manifold $\mathcal{M}$ which will be explained in the next section.

B. Quotient Manifold Space

The main idea of Riemannian optimization for rank-constrained optimization is based on matrix factorization. Specifically, the balanced factorization, i.e.,

$$\boldsymbol{X} = (\boldsymbol{U}\boldsymbol{\Sigma}^{\frac{1}{2}})(\boldsymbol{\Sigma}^{\frac{1}{2}}\boldsymbol{V}^T) = \boldsymbol{L}\boldsymbol{R}^T, \quad (10)$$

takes advantages of lower-dimensional search space [32] over the other general forms of matrix factorization (e.g.,the subspace-projection factorization and the polar factorization). However, the balanced factorization is not unique as the transport operation $(\boldsymbol{L},\boldsymbol{R}) \mapsto (\boldsymbol{L}\boldsymbol{M}^{-1}, \boldsymbol{R}\boldsymbol{M}^T)$ makes the original matrix $\boldsymbol{X} = \boldsymbol{L}\boldsymbol{M}^{-1}(\boldsymbol{R}\boldsymbol{M}^T)^T = \boldsymbol{L}\boldsymbol{R}^T$ unchanged, where $\boldsymbol{M} \in \mathrm{GL}(r) = \{\boldsymbol{M} \in \mathbb{R}^{r\times r} : \det(M) \neq 0\}$ is the *Lie group* containing all $r\times r$ invertible matrices [33], [34]. Therefore, to address this issue, the search space for problem $\mathscr{P}$ should be identified with the *quotient space* $\mathcal{M}/\sim := (\mathbb{R}_*^{m\times r}\times\mathbb{R}_*^{n\times r})/\mathrm{GL}(r)$, where $\mathcal{M} := \mathbb{R}_*^{m\times r}\times\mathbb{R}_*^{n\times r}$ is the *computational space*, $\mathrm{GL}(r)$ is the fiber space and $\sim$ represents the equivalence relation. The dimension of $\mathcal{M}/\sim$ is $(m+n-r)r$ and this quotient space describes the set of equivalence classes

$$[(\boldsymbol{L},\boldsymbol{R})] = \{(\boldsymbol{L}\boldsymbol{M}^{-1}, \boldsymbol{R}\boldsymbol{M}^T) : \ M \in \mathrm{GL}(r)\}. \quad (11)$$

Since the quotient manifold $\mathcal{M}/\sim$ is an abstract space, to design algorithms, the corresponding matrix representations of geometric objects in $\mathcal{M}/\sim$ are needed. Based on the theory of *Riemannian submersion* [33, Section 3.6.2], the matrix representations can be obtained in the computational space.

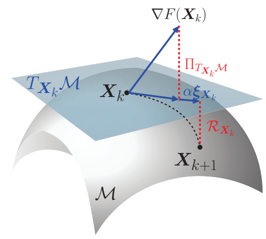

Fig. 1. Graphical representation of the concept of matrix manifold optimization.

V. Matrix Optimization over Quotient Manifolds

In this section, we develop the matrix optimization algorithm over the quotient manifold space endowed with fixed-rank matrices.

A. The Framework of Riemannian Optimization

A *Riemannian metric* in the computational space is required to assure the structure of quotient space on which optimization algorithms are developed. In particular, the Riemannian metric $g_{\boldsymbol{X}} : T_{\boldsymbol{X}}\mathcal{M}\times T_{\boldsymbol{X}}\mathcal{M} \to \mathbb{R}$ is an inner product between the tangent vectors on the tangent space $T_{\boldsymbol{X}}\mathcal{M}$, which is invariable along the set of equivalence classes (11). According to [33, Example 3.6.4], we choose the natural metric for the space $R_*^{m\times r}\times R_*^{n\times r}$, given by [29]

$$g_{\boldsymbol{X}}(\boldsymbol{\zeta}_{\boldsymbol{X}},\boldsymbol{\xi}_{\boldsymbol{X}}) = \mathrm{Tr}((\boldsymbol{L}^{\boldsymbol{T}}\boldsymbol{L})^{-1}\boldsymbol{\zeta}_L^T\boldsymbol{\xi}_L) + \mathrm{Tr}((\boldsymbol{R}^{\boldsymbol{T}}\boldsymbol{R})^{-1}\boldsymbol{\zeta}_R^T\boldsymbol{\xi}_R), \quad (12)$$

where $\boldsymbol{\zeta}_{\boldsymbol{X}} = (\boldsymbol{\zeta}_L,\boldsymbol{\zeta}_R)$, $\boldsymbol{\xi}_{\boldsymbol{X}} = (\boldsymbol{\xi}_L,\boldsymbol{\xi}_R)$ and $\boldsymbol{X} = (\boldsymbol{L},\boldsymbol{R})$.

With respect to the metric, the tangent space $T_{\boldsymbol{X}}\mathcal{M}$ at the point $\boldsymbol{X}$ can be represented as the sum of two complementary spaces:

$$T_{\boldsymbol{X}}\mathcal{M} = \mathcal{V}_{\boldsymbol{X}}\mathcal{M} \oplus \mathcal{H}_{\boldsymbol{X}}\mathcal{M}, \quad (13)$$

where $\mathcal{V}_{\boldsymbol{X}}\mathcal{M}$ is the *vertical space* and $\mathcal{H}_{\boldsymbol{X}}\mathcal{M}$ is the *horizontal space*. Specifically, directions of vectors in the vertical space $\mathcal{V}_{\boldsymbol{X}}\mathcal{M}$ are tangent to the set of equivalence classes (11) and directions of vectors in the horizontal space $\mathcal{H}_{\boldsymbol{X}}\mathcal{M}$ are orthogonal to the set of equivalence classes $[\boldsymbol{X}]$ (11). Thus, vectors $\xi_{\boldsymbol{X}} \in \mathcal{H}_{\boldsymbol{X}}\mathcal{M}$ are invariant along the equivalence class $[\boldsymbol{X}]$ (11). Let $T_{[\boldsymbol{X}]}(\mathcal{M}/\sim)$ denote the tangent space at point $[\boldsymbol{X}]$ on the quotient space $\mathcal{M}/\sim$, then there exists unique element $\xi_{\boldsymbol{X}} \in \mathcal{H}_{\boldsymbol{X}}\mathcal{M}$ being the matrix representation of $\xi_{[\boldsymbol{X}]} \in T_{[\boldsymbol{X}]}(\mathcal{M}/\sim)$, called the *horizontal lift* of $\xi_{[\boldsymbol{X}]}$ at $\boldsymbol{X}$ [33, Section 3.5.8].

Based on the above discussion, the general process of Riemannian optimization framework in the computational space $\mathcal{M}$ can be briefly described as searching the descent direction $\boldsymbol{\xi}_{\boldsymbol{X}}$ with steepest decrease in $F(\boldsymbol{X})$ on the horizontal space $\mathcal{H}_{\boldsymbol{X}}\mathcal{M}$. Then retract the vector $\boldsymbol{\xi}_{\boldsymbol{X}}$ onto the manifold via the operation $\mathcal{R}_{\boldsymbol{X}} : \mathcal{H}_{\boldsymbol{X}}\mathcal{M} \to \mathcal{M}$ called *retraction*. Based on notions above, a generic matrix manifold optimization algorithm is presented in Algorithm 1. In addition, the graphical representation of Algorithm 1 is illustrated in Fig.1.

Algorithm 1: Matrix Manifold Optimization

Given: Riemannian manifold $\mathcal{M}$ with Riemannian metric g; retraction mapping $\mathcal{R}$; objective function F and the stepsize α.

Output: $\boldsymbol{X}_k$

1: **Initialize:** initial point $\boldsymbol{X}_0$, $k=0$
2: **while** not converged **do**
3: Compute a descent direction $\boldsymbol{\xi}_k$. (e.g., via implementing trust-region method)
4: Update $\boldsymbol{X}_{k+1} = \mathcal{R}_{\boldsymbol{X}_k}(\alpha \boldsymbol{\xi}_k)$
5: $k = k+1$.
6: **end while**

B. Optimization Related Ingredients

In this subsection, we present the matrix representations of abstract geometrics objects on the quotient manifold in detail to develop Riemannian optimization algorithms.

Riemannian gradient. The Riemannian gradient $\mathrm{grad}_{\boldsymbol{X}} f \in T_{\boldsymbol{X}}\mathcal{M}$ satisfies [33]

$$g_{\boldsymbol{X}}(\boldsymbol{\xi}_{\boldsymbol{X}}, \mathrm{grad}_{\boldsymbol{X}} f) = Df(\boldsymbol{X})[\boldsymbol{\xi}_{\boldsymbol{X}}] \tag{14}$$

where $Df(\boldsymbol{X})[\boldsymbol{\xi}_{\boldsymbol{X}}]$ denotes the Euclidean directional derivative of the objective function $f(\boldsymbol{X})$ in the direction of $\boldsymbol{\xi}_{\boldsymbol{X}} \in T_{\boldsymbol{X}}\mathcal{M}$. Thus, the Riemannian gradient is deduced from the Euclidean derivative of $f(\boldsymbol{X})$ computed in the sequel.

Let $M_{ijk}(\boldsymbol{X})$ denote $Y_{ijk}(X_{ij} - X_{ik})$, which is a linear function of which the derivative with respect to the matrix $\boldsymbol{X}$ is

$$M'_{ijk}(\boldsymbol{X}) = Y_{ijk}\boldsymbol{\delta}_{ij} - Y_{ijk}\boldsymbol{\delta}_{ik}, \tag{15}$$

where $\boldsymbol{\delta}_{ij}$ is a $m \times n$ matrix with $[\boldsymbol{\delta}_{ij}]_{ij} = 1$ and the rest being zeros. Let $\mathcal{L}_{ijk}(\boldsymbol{X})$ denote $\log(f(Y_{ijk}(X_{ij} - X_{ik})))$ which is the separated element of the summation $\mathcal{L}_{\Omega,\boldsymbol{Y}}(\boldsymbol{X})$. Additionally, the derivative of $N(\boldsymbol{X})$ is derived as $N'(\boldsymbol{X}) = 2\boldsymbol{X} \circ \exp(\boldsymbol{X}^{\circ 2})$, where '$\circ$' denotes the Hadamard product operation, i.e., elementwise product/power. Hence, the Euclidean derivative of $f(\boldsymbol{X})$ is given as

$$\mathrm{grad}(\boldsymbol{X}) = \nabla \mathcal{L}_{\Omega,\boldsymbol{Y}}(\boldsymbol{X}) + \lambda\, Q(\boldsymbol{X}), \tag{16}$$

where $\nabla \mathcal{L}_{\Omega,\boldsymbol{Y}}(\boldsymbol{X}) = \sum_{(i,j,k)\in\Omega} \mathcal{L}'_{ijk}(\boldsymbol{X}) M'_{ijk}(\boldsymbol{X})$ and $Q(\boldsymbol{X}) = \frac{N'(\boldsymbol{X})}{N(\boldsymbol{X})}$.

The Riemannian Hessian is given by [29]

$$\mathrm{Hess}_{\boldsymbol{X}} f[\boldsymbol{\xi}_{\boldsymbol{X}}] = \Pi_{\mathcal{H}_{\boldsymbol{X}}\mathcal{M}}(\nabla_{\boldsymbol{\xi}_{\boldsymbol{X}}} \mathrm{grad}_{\boldsymbol{X}} f), \tag{17}$$

where $\mathrm{grad}_{\boldsymbol{X}} f$ denotes the Riemannian gradient; $\Pi_{\mathcal{H}_{\boldsymbol{X}}\mathcal{M}}$ is a projection operator mapping a tangent vector $\boldsymbol{\eta}_{\boldsymbol{X}} \in T_{\boldsymbol{X}}\mathcal{M}$ onto *the horizontal space* $\mathcal{H}_{\boldsymbol{X}}\mathcal{M}$ [33] given in [29] and $\nabla_{\boldsymbol{\xi}_{\boldsymbol{X}}} \mathrm{grad}_{\boldsymbol{X}} f$ is the Riemannian connection revealed in the following.

Riemannian Hessian. The *Riemannian connection* of $\boldsymbol{\eta}_{\boldsymbol{X}} \in T_{\boldsymbol{X}}\mathcal{M}$ in the direction of $\boldsymbol{\xi}_{\boldsymbol{X}} \in T_{\boldsymbol{X}}\mathcal{M}$ is defined as

$$\nabla_{\boldsymbol{\xi}_{\boldsymbol{X}}} \boldsymbol{\eta}_{\boldsymbol{X}} = D\boldsymbol{\eta}_{\boldsymbol{X}}[\boldsymbol{\xi}_{\boldsymbol{X}}] + (\boldsymbol{A}_L, \boldsymbol{A}_R), \tag{18}$$

where $(\boldsymbol{A}_L, \boldsymbol{A}_R)$ is given in [29]. Thus, the Riemannian connection $\nabla_{\boldsymbol{\xi}_{\boldsymbol{X}}} \mathrm{grad}_{\boldsymbol{X}} f$ can be deduced from the second-order partial derivatives of $f(\boldsymbol{X})$. Therein, the partial directional derivative with respect to $\boldsymbol{L}$ in the direction of $\boldsymbol{\xi}_{\boldsymbol{X}} := (\boldsymbol{\xi}_L, \boldsymbol{\xi}_R) \in T_{\boldsymbol{X}}\mathcal{M}$ is written as

$$\nabla^2_L f(\boldsymbol{X})[\boldsymbol{\xi}_{\boldsymbol{X}}] = \nabla \mathrm{grad}(\boldsymbol{X})[\boldsymbol{\xi}_{\boldsymbol{X}}] \cdot \boldsymbol{R} + \mathrm{grad}(\boldsymbol{X}) \cdot \boldsymbol{\xi}_R \tag{19}$$

where $\mathrm{grad}(\boldsymbol{X})$ is the Euclidean derivative of the objective function $f(\boldsymbol{X})$. While the second-order partial derivative with respect to $\boldsymbol{R}$ is similar as that in (19). Furthermore, the directional derivative of Euclidean gradient in (19) in the direction of $\boldsymbol{\xi}_{\boldsymbol{X}}$ is derived as

$$\nabla \mathrm{grad}(\boldsymbol{X})[\boldsymbol{\xi}_{\boldsymbol{X}}] = \nabla^2 \mathcal{L}_{\Omega,\boldsymbol{Y}}(\boldsymbol{X}) + \lambda\, DQ(\boldsymbol{X})[\boldsymbol{\xi}_{\boldsymbol{X}}], \tag{20}$$

where

$$\nabla^2 \mathcal{L}_{\Omega,\boldsymbol{Y}}(\boldsymbol{X}) = \sum_{(i,j,k)\in\Omega} \left[M_{ijk}(\boldsymbol{\xi}_L \boldsymbol{R}^T + \boldsymbol{L}\boldsymbol{\xi}_R^T)\right] \cdot \mathcal{L}''_{ijk}(\boldsymbol{X}) M'_{ijk}(\boldsymbol{X}) \tag{21}$$

and the directional derivative of $Q(\boldsymbol{X})$ in the direction of $\boldsymbol{\xi}_{\boldsymbol{X}}$ is written as

$$\begin{aligned} DQ(\boldsymbol{X})[\boldsymbol{\xi}_{\boldsymbol{X}}] = \tfrac{1}{N^2(\boldsymbol{X})} \big[& N(\boldsymbol{X})(2e^{\boldsymbol{X}^{\circ 2}} + 2\boldsymbol{X} \circ \boldsymbol{N}'(\boldsymbol{X})) \circ \boldsymbol{K} \\ & - \boldsymbol{N}'(\boldsymbol{X}) \cdot \textstyle\sum_{ij} (\boldsymbol{N}'(\boldsymbol{X}) \circ \boldsymbol{K})_{ij} \big] \end{aligned} \tag{22}$$

where $\boldsymbol{K} = \boldsymbol{\xi}_L \boldsymbol{R}^T + \boldsymbol{L}\boldsymbol{\xi}_R^T$. To sum up, the optimization-related ingredients for problem $\mathscr{P}$ are presented in Table I. More details about fixed-rank Rimannian optimization can be further referred to [29].

C. Trust Region Algorithm

Based on the aforementioned matrix manifold optimization framework and the matrix representations, we present the trust-region algorithm. Consider a sequence of iterates $\boldsymbol{X}_0, \boldsymbol{X}_1, \cdots$ and assume the current iterate $\boldsymbol{X}_t \in \mathcal{M}$. The trust-region subproblem is formulated as

$$\begin{aligned} \underset{\boldsymbol{\xi}_{\boldsymbol{X}_t} \in \mathcal{H}_{\boldsymbol{X}_t}\mathcal{M}}{\text{minimize}} \quad & m(\boldsymbol{\xi}_{\boldsymbol{X}_t}) \\ \text{subject to} \quad & g_{\boldsymbol{X}_t}(\boldsymbol{\xi}_{\boldsymbol{X}_t}, \boldsymbol{\xi}_{\boldsymbol{X}_t}) \le \delta_t^2, \end{aligned} \tag{23}$$

where δ_t is the trust-region radius in t-th iteration and the cost function in the problem (23) is written as

$$\begin{aligned} m(\boldsymbol{\xi}_{\boldsymbol{X}_t}) = F(\boldsymbol{X}_t) + & g_{\boldsymbol{X}_t}(\boldsymbol{\xi}_{\boldsymbol{X}_t}, \mathrm{grad}_{\boldsymbol{X}_t} f) \\ & + \frac{1}{2} g_{\boldsymbol{X}_t}(\boldsymbol{\xi}_{\boldsymbol{X}_t}, \mathrm{Hess}_{\boldsymbol{X}_t} f\,[\boldsymbol{\xi}_{\boldsymbol{X}_t}]), \end{aligned} \tag{24}$$

where $\mathrm{grad}_{\boldsymbol{X}_t} f$ and $\mathrm{Hess}_{\boldsymbol{X}_t} f\,[\boldsymbol{\xi}_{\boldsymbol{X}_t}]$ are the matrix representation of the Riemannian gradient and Riemannian Hessian on the quotient manifold, respectively.

The trust-region radius δ_t is adjusted according to current iterate. The new iterate is updated according to

$$\boldsymbol{X}_{t+1} = \mathcal{R}_{\boldsymbol{X}_t}(\boldsymbol{\xi}_{\boldsymbol{X}_t}), \tag{25}$$

where the retraction mapping operator $\mathcal{R}_{\boldsymbol{X}_t} : \mathcal{H}_{\boldsymbol{X}}\mathcal{M} \to \mathcal{M}$

TABLE I
OPTIMIZATION-RELATED INGREDIENTS FOR PROBLEM $\mathscr{P}$

	$\mathscr{P}$: minimize $\mathcal{L}_{\Omega,\boldsymbol{Y}}(\boldsymbol{L}\boldsymbol{R}^T) + \lambda \log N(\boldsymbol{L}\boldsymbol{R}^T)$
Matrix representation of an element $\boldsymbol{X} \in \mathcal{M}$	$\boldsymbol{X} = (\boldsymbol{L}, \boldsymbol{R})$
Computational space $\mathcal{M}$	$\mathbb{R}_*^{m\times r} \times \mathbb{R}_*^{n\times r}$
Quotient space	$\mathcal{M}/\sim := (\mathbb{R}_*^{m\times r} \times \mathbb{R}_*^{n\times r})/\mathrm{GL}(r)$
Metric $g_{\boldsymbol{X}}(\boldsymbol{\zeta}_{\boldsymbol{X}}, \boldsymbol{\xi}_{\boldsymbol{X}})$ for $\boldsymbol{\zeta}_{\boldsymbol{X}}, \boldsymbol{\xi}_{\boldsymbol{X}} \in T_{\boldsymbol{X}}\mathcal{M}$	$g_{\boldsymbol{X}}(\boldsymbol{\zeta}_{\boldsymbol{X}}, \boldsymbol{\xi}_{\boldsymbol{X}}) = \mathrm{Tr}((\boldsymbol{L}^T\boldsymbol{L})^{-1}\zeta_L^T\boldsymbol{\xi}_L) + \mathrm{Tr}((\boldsymbol{R}^T\boldsymbol{R})^{-1}\zeta_R^T\boldsymbol{\xi}_R)$
Riemannian gradient $\mathrm{grad}_{\boldsymbol{X}} f$	$\mathrm{grad}_{\boldsymbol{X}} f = (\mathrm{grad}_{\boldsymbol{L}} f, \mathrm{grad}_{\boldsymbol{R}} f) = (\nabla_{\boldsymbol{L}} F(\boldsymbol{X})\boldsymbol{L}^T\boldsymbol{L}, \nabla_{\boldsymbol{R}} F(\boldsymbol{X})\boldsymbol{R}^T\boldsymbol{R})$
Riemannian Hessian $\mathrm{Hess}_{\boldsymbol{X}} f[\boldsymbol{\xi}_{\boldsymbol{X}}]$	$\mathrm{Hess}_{\boldsymbol{X}} f[\boldsymbol{\xi}_{\boldsymbol{X}}] = \Pi_{\mathcal{H}_{\boldsymbol{X}}\mathcal{M}}(\nabla_{\boldsymbol{\xi}_{\boldsymbol{X}}} \mathrm{grad}_{\boldsymbol{X}} f)$
Retraction $\mathcal{R}_{\boldsymbol{X}} : T_{\boldsymbol{X}}\mathcal{M} \to \mathcal{M}$	$\mathcal{R}_{\boldsymbol{X}}(\boldsymbol{\xi}_{\boldsymbol{X}}) = (\boldsymbol{L} + \boldsymbol{\xi}_L, \boldsymbol{R} + \boldsymbol{\xi}_R)$

in each iteration is given by

$$\mathcal{R}_{\boldsymbol{X}}(\boldsymbol{\xi}_{\boldsymbol{X}}) = (\boldsymbol{L} + \boldsymbol{\xi}_L, \boldsymbol{R} + \boldsymbol{\xi}_R), \tag{26}$$

where $\boldsymbol{\xi}_{\boldsymbol{X}} := (\boldsymbol{\xi}_L, \boldsymbol{\xi}_R) \in \mathcal{H}_{\boldsymbol{X}}\mathcal{M}$ [29]. More details on the trust region method can be found in [33].

VI. EXPERIMENTS

In this section, we conduct experiments on synthetic data to validate the effectiveness of the proposed smoothed matrix manifold optimization algorithm for individual rankings recovery from pairwise comparisons.

A. Synthetic Data and Performance Metric

We run our simulations with synthetic data and results are evaluated with well-defined performance metric. The simulation settings are given as follows:

1) Weight matrix $\boldsymbol{X}^*$: We generate the weight/score matrix $\boldsymbol{X}^* = \boldsymbol{U}\boldsymbol{V}^T$, where $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{R}^{K\times r}$ have i.i.d. entries uniformly chosen from $[-0.5, 0.5]$. Matrix $\boldsymbol{X}^*$ are then scaled so that $\|\boldsymbol{X}^*\|_\infty = 1$.
2) Pairwise comparisons Y_{ijk}: The pairwise comparisons are derived from the BTL model with the underlying weight matrix $\boldsymbol{X}^*$ and $\sigma = 0.18$ [13].
3) Observation/sampling set Ω: Given the sample size of Ω as $|\Omega|$, we choose $|\Omega|$ independent observations uniformly at random.
4) Performance metric: After scaling $\boldsymbol{X}$ such that $\|\boldsymbol{X}\|_\infty = 1$, we adopt the relative mean square error (MSE) to evaluate the performance of weight matrix estimation [5]

$$\mathrm{err}(\boldsymbol{X}) = \|\boldsymbol{X} - \boldsymbol{X}^*\|_F^2 / \|\boldsymbol{X}^*\|_F^2. \tag{27}$$

In particular, we compare three algorithms described as:

- **Proposed Riemannian trust-region algorithm solving log-sum-exp regularized problem (PRTRS)**: The algorithm uses *Manopt* [36] to solve the problem (9).
- **Bi-factor gradient descent solving log-barrier regularized problem (BFGDB)**: This algorithm [14] solves the problem (8). The regularization term coefficient τ_t is set to $\mu \cdot \tau_{t-1}$ during t-th outer iteration of the algorithm [27], [28]. Meanwhile, Bi-factor gradient descent with the constant stepsize being $s := \frac{2}{187}\{\frac{1}{\|\boldsymbol{U}_0\|_F^2}, \frac{1}{\|\boldsymbol{V}_0\|_F^2}\}$ [14] is implemented in the inner iteration.

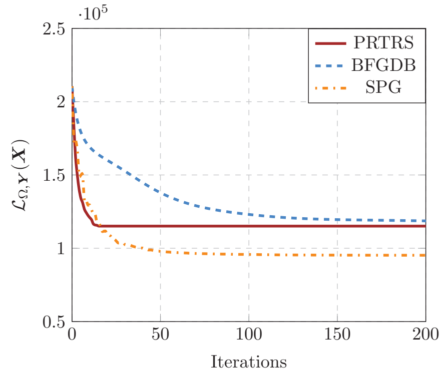

Fig. 2. Rate of convergence of different algorithms.

- **Spectral projected-gradient (SPG)**: In this algorithm [13], we adopt codes provided in [13] to solve the problem (7) with setting the elementwise infinity norm constraint coefficient α equal to $\|\boldsymbol{X}^*\|_\infty$ [5].

We set $K = m = n$ for all experiments. All algorithms are initialized with $\boldsymbol{U}_0, \boldsymbol{V}_0$ whose entries are i.i.d. and drawn form the standard normal distribution, which are scaled to $\|\boldsymbol{X}_0\|_\infty = 0.95$ with $\boldsymbol{X}_0 = \boldsymbol{U}_0\boldsymbol{V}_0^T$. In PRTRS algorithm, it is terminated either the norm of Riemannian gradient $\|\mathrm{grad}_{\boldsymbol{X}_t} f\| < 10^{-6}$ or the number of iterations exceeding 500. The stopping criterion for inner iteration of BFGDB is the same as [14] and it is also ended when $\|\boldsymbol{X}\|_\infty \geq 1$. As for the outer iteration of BFGDB, the regularization term coefficient τ_1 is set to $mn/\mathcal{L}_{\Omega,\boldsymbol{Y}}(\boldsymbol{X}_0)$ and the number of outer iterations is

$$\left\lceil \frac{\log mn - \log \eta - \log \tau_1}{\log \mu} \right\rceil \tag{28}$$

where $\eta = 10^{-3}$ and $\mu = 2$ [27], [28]. The setting for SPG algorithm is the same as [13].

B. Evaluation

Convergence Rate Consider the circumstance with $K = 200$ and the sampling size being $(drK\log K)$, where $d = 15$ and $r = 15$ denote the rescaled sample size and the rank of weight matrix respectively [8], Fig. 2 demonstrates the convergence rate of different algorithms.

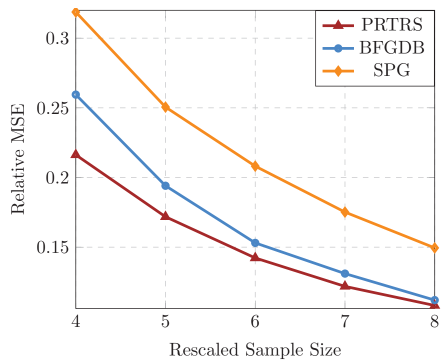

Fig. 3. Relative MSE with different sample sizes d.

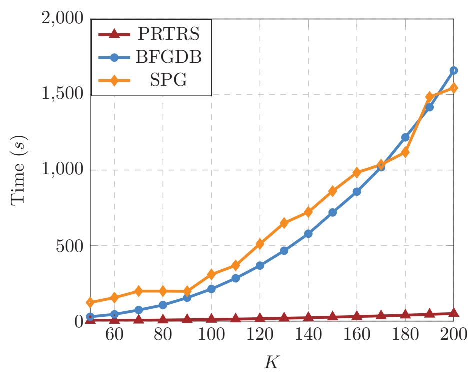

Fig. 4. Computation time with different sizes.

Relative MSE with Different Sample Sizes Under the circumstance of $K = 200$ and $r = 10$, we simulate with different sample sizes $(drK \log K)$ [8]. We conduct numerical experiments averaged over 100 realizations to compare these three algorithms. Fig. 3 demonstrates the Relative MSE corresponding to different rescaled sample sizes d.

Computation Time with Different Problem Sizes With fixed parameters as $r = 10$ and $d = 5$, we conduct the numerical experiments averaged over 100 realizations to simulate three algorithms with different sizes K. Fig. 4 demonstrates the computational time with different sizes K.

In summary, simulations demonstrate that the proposed Riemannian trust-region algorithm significantly outperforms the BFGD and SPG algorithm in terms of speedups (i.e., rate of convergence and computational time) and performance (i.e., MSE). In particular, the effectiveness of the proposed Riemannian trust-region algorithm is achieved by exploiting the structures of fixed-rank matrices and smoothed objective function.

VII. Conclusions

In this paper, we develop a low-rank approach based on maximum likelihood estimation (MLE) with coupled rank constraint and elementwise infinity norm constraint to recover individual rankings from pairwise comparisons in social computing system. We further proposed a smoothed surrogate of elementwise infinity norm in the adopted objective function. In addition, a versatile framework of Riemannian optimization is represented by generalizing the classical geometric concepts [33] to geometric concepts on the quotient manifolds of fixed-rank matrices. Under the framework, the Rimannian trust-region algorithm is developed to return an *approximate local minimum* from *arbitrary initial points*. Simulation results demonstrate that the proposed regularized smoothed approach supported by the Riemannian trust-region algorithm significantly outperforms the state-of-art algorithms in terms of performance (i.e., relative MSE), computational cost and convergence rate.

References

[1] G. Xintong, W. Hongzhi, Y. Song, and G. Hong, “Brief survey of crowdsourcing for data mining,” *Expert Syst. Appl.*, vol. 41, no. 17, pp. 7987–7994, 2014.

[2] Y. Chen, C. Suh, and A. J. Goldsmith, “Information recovery from pairwise measurements,” *IEEE Trans. Inf. Theory*, vol. 62, pp. 5881–5905, Oct. 2016.

[3] I. Dokmanic, R. Parhizkar, J. Ranieri, and M. Vetterli, “Euclidean distance matrices: Essential theory, algorithms, and applications,” *IEEE Commun. Mag.*, vol. 32, pp. 12–30, Nov 2015.

[4] N. B. Shah and M. J. Wainwright, “Simple, robust and optimal ranking from pairwise comparisons,” *arXiv preprint arXiv:1512.08949*, 2015.

[5] D. Park, J. Neeman, J. Zhang, S. Sanghavi, and I. S. Dhillon, “Preference completion: large-scale collaborative ranking from pairwise comparisons,” in *Proc. Int. Conf. Mach. Learn. (ICML)*, pp. 1907–1916, 2015.

[6] J. Zhu, C. Zhang, and M. Y. Ma, “Multi-aspect rating inference with aspect-based segmentation,” *IEEE Trans. Affect. Comput.*, vol. 3, no. 4, pp. 469–481, 2012.

[7] X. Chen, P. N. Bennett, K. Collins-Thompson, and E. Horvitz, “Pairwise ranking aggregation in a crowdsourced setting,” in *Proc. 6th ACM Int. Conf. on WSDM*, pp. 193–202, ACM, 2013.

[8] Y. Lu and S. N. Negahban, “Individualized rank aggregation using nuclear norm regularization,” in *Proc. 53rd Annu. Allerton Conf. Commun., Control, Comput.*, pp. 1473–1479, IEEE, 2015.

[9] J. D. Rennie and N. Srebro, “Fast maximum margin matrix factorization for collaborative prediction,” in *Proc. Int. Conf. Mach. Learn. (ICML), 22nd*, pp. 713–719, ACM, 2005.

[10] N. Boumal, P.-A. Absil, and C. Cartis, “Global rates of convergence for nonconvex optimization on manifolds,” *arXiv preprint arXiv:1605.08101*, 2016.

[11] A. S. Bandeira, N. Boumal, and V. Voroninski, “On the low-rank approach for semidefinite programs arising in synchronization and community detection,” in *29th Annu. Proc. Conf. Learn. Theory (COLT)*, pp. 23–26, 2016.

[12] N. Agarwal, Z. Allen-Zhu, B. Bullins, E. Hazan, and T. Ma, “Finding approximate local minima for nonconvex optimization in linear time,” in *Proc. ACM Symp. Theory of Computing (STOC)*, 2017.

[13] M. A. Davenport, Y. Plan, E. van den Berg, and M. Wootters, “1-bit matrix completion,” *Inf. Inference*, vol. 3, no. 3, pp. 189–223, 2014.

[14] D. Park, A. Kyrillidis, C. Caramanis, and S. Sanghavi, “Finding low-rank solutions via non-convex matrix factorization, efficiently and provably,” *arXiv preprint arXiv:1606.03168*, 2016.

[15] R. Sun and Z.-Q. Luo, “Guaranteed matrix completion via non-convex factorization,” *IEEE Trans. Inf. Theory*, vol. 62, no. 11, pp. 6535–6579, 2016.

[16] Y. Chen and M. J. Wainwright, “Fast low-rank estimation by projected gradient descent: General statistical and algorithmic guarantees,” *arXiv preprint arXiv:1509.03025*, 2015.

[17] P. Jain, P. Netrapalli, and S. Sanghavi, “Low-rank matrix completion using alternating minimization,” in *Proc. ACM Symp. Theory Comput.*, pp. 665–674, ACM, 2013.

[18] J. D. Lee, M. Simchowitz, M. I. Jordan, and B. Recht, “Gradient descent only converges to minimizers,” in *29th Annu. Proc. Conf. Learn. Theory (COLT)*, pp. 1246–1257, 2016.

[19] S. Bhojanapalli, B. Neyshabur, and N. Srebro, "Global optimality of local search for low rank matrix recovery," in *Adv. Neural. Inf. Process. Syst. (NIPS)*, pp. 3873–3881, 2016.

[20] R. Ge, F. Huang, C. Jin, and Y. Yuan, "Escaping from saddle points-online stochastic gradient for tensor decomposition.," in *28th Annu. Proc. Conf. Learn. Theory (COLT)*, pp. 797–842, 2015.

[21] J. Sun, Q. Qu, and J. Wright, "When are nonconvex problems not scary?," *NIPS Workshop on Non-convex Optimization for Machine Learning: Theory and Practice*, 2015.

[22] R. A. Bradley and M. E. Terry, "Rank analysis of incomplete block designs: I. The method of paired comparisons," *Biometrika*, vol. 39, no. 3-4, pp. 324–345, 1952.

[23] A. Agarwal, S. Negahban, and M. J. Wainwright, "Fast global convergence of gradient methods for high-dimensional statistical recovery," *Annals of Statistics*, vol. 40, no. 5, pp. 2452–2482, 2012.

[24] A. Ali, E. Wong, and J. Z. Kolter, "A semismooth newton method for fast, generic convex programming," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2017.

[25] B. ODonoghue, E. Chu, N. Parikh, and S. Boyd, "Conic optimization via operator splitting and homogeneous self-dual embedding," *J. Optim. Theory. Appl.*, vol. 169, no. 3, pp. 1042–1068, 2016.

[26] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[27] E. Candes and J. Romberg, "l1-magic: Recovery of sparse signals via convex programming," *California Inst. Technol., Pasadena, CA, Tech. Rep.*, 2005.

[28] S. A. Bhaskar and A. Javanmard, "1-bit matrix completion under exact low-rank constraint," in *In Proc. IEEE Conf. Inform. Science and Systems (CISS)*, pp. 1–6, IEEE, 2015.

[29] B. Mishra, G. Meyer, S. Bonnabel, and R. Sepulchre, "Fixed-rank matrix factorizations and Riemannian low-rank optimization," *Comput. Statist.*, vol. 29, no. 3-4, pp. 591–621, 2013.

[30] A. Beck and M. Teboulle, "Smoothing and first order methods: A unified framework," *SIAM J. Optim.*, vol. 22, no. 2, pp. 557–580, 2012.

[31] A. M. Thompson, J. C. Brown, J. W. Kay, and D. M. Titterington, "A study of methods of choosing the smoothing parameter in image restoration by regularization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 4, pp. 326–339, 1991.

[32] N. Boumal and P.-a. Absil, "RTRMC: A Riemannian trust-region method for low-rank matrix completion," in *Adv. Neural. Inf. Process. Syst. (NIPS)*, pp. 406–414, 2011.

[33] P. A. Absil, R. Mahony, and R. Sepulchre, *Optimization algorithms on matrix manifolds*. Princeton University Press, 2008.

[34] G. Meyer, S. Bonnabel, and R. Sepulchre, "Linear regression under fixed-rank constraints: a Riemannian approach," in *Proc. Int. Conf. Mach. Learn. (ICML), 28th*, 2011.

[35] Y. Jiao, A. Korba, and E. Sibony, "Controlling the distance to a kemeny consensus without computing it," in *Proc. Int. Conf. Mach. Learn. (ICML)*, pp. 2971–2980, 2016.

[36] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre, "Manopt, a Matlab toolbox for optimization on manifolds," *J. Mach. Learn. Res.*, vol. 15, pp. 1455–1459, 2014.