Spectral Ranking with Covariates

Siu Lun Chau, Mihai Cucuringu, and Dino Sejdinovic

Department of Statistics, University of Oxford, 29 St Giles' Oxford OX1, UK
{siu.chau, mihai.cucuringu, dino.sejdinovic}@stats.ox.ac.uk

Abstract. We consider spectral approaches to the problem of ranking n players given their incomplete and noisy pairwise comparisons, but revisit this classical problem in light of player covariate information. We propose three spectral ranking methods that incorporate player covariates and are based on *seriation*, *low-rank structure* assumption and *canonical correlation*, respectively. Extensive numerical simulations on both synthetic and real-world data sets demonstrated that our proposed methods compare favorably to existing state-of-the-art covariate-based ranking algorithms.

Keywords: ranking, spectral methods, kernel methods

1   Introduction

We consider the classical problem of ranking n players, given a set of pairwise comparisons, but revisiting this problem in light of available player covariate information. We assume ranking is represented by a vector $\mathbf{r} \in \mathbb{R}^n$, which can be interpreted as a one-dimensional representation of players' underlying skills. In practice, such pairwise comparisons are often incomplete and inconsistent with respect to the underlying skill vector $\mathbf{r}$; therefore, the objective of a ranking problem is often to recover a total ordering of the players that is as consistent as possible with the available noisy and sparse information.

There exists a rich literature on ranking that can be dated back as early as the seminal work of Kendall and Smith [18] in the 1940s. Classical examples such as the Bradley-Terry-Luce model [2], Random Utility model [30], Mallows-Condorcet model [20], and Thurstone-Mosteller model [21] have inspired numerous developments of modern ranking algorithms [4]. These methods take a probabilistic approach, in which they treat ranking as the output generated from a probabilistic model, where maximum likelihood estimators can be built subsequently.

Apart from probabilistic approaches, spectral methods that apply the theory of linear maps (in particular, eigenvalues and eigenvectors) to the pairwise comparison matrix or derivations of it, in order to extract rankings, is also a century-old [31] idea that was made famous by Google's PageRank [25] algorithm. There are several other popular spectral ranking methods proposed in recent years as well. For example, Fogel et al. [12] connected the problem of seriation [1] to ranking and proposed SERIALRANK algorithm. Cucuringu et al.[11], on the other hand, proposed SVD-RANK, a simple SVD-based approach to recover rankings

when cardinal comparisons are observed. Cucuringu also proposed SYNCRANK [8] based on group synchronization using semidefinite programming and spectral relaxations.

In many instances, it is of interest to investigate whether some player covariates affect the results of the comparisons. It is also natural to believe that incorporating informative covariates will lead to better ranking estimations. Besides improving estimation, we believe there are at least two more reasons why this might be useful in practice:

1. Covariate-free ranking algorithms cannot take into account instances where new players are added to the comparison sets, without first observing new matches with existing players. This can be overcome by building a model of the ranking as a function of player covariates.
2. Traditional ranking methods often break down if the sample complexity of the comparison graph does not scale as $O(n \log n)$, meaning the graph is disconnected with high probability [11]. By utilising the smoothness of rankings across covariate information, one is able to overcome this obstacle and infer meaningful rankings even in the cases of very sparse comparisons that render the graph disconnected.

A number of extensions to incorporate covariates have been proposed for probabilistic approaches [7,24,28]. In contrast, spectral ranking with covariates received much lesser attention. To the best of our knowledge, the recent work of Jain et al. [17] was the first to incorporate covariates into spectral ranking. However, unlike our approaches, their approach cannot be used to predict rankings on new players.

To this end, we introduce a suite of spectral ranking algorithms incorporating covariate information: C-SERIALRANK, SVDCOVRANK and CCRANK. Both C-SERIALRANK and SVDCOVRANK extend on existing spectral ranking techniques, while CCRANK is a new spectral ranking approach motivated by analysing the canonical correlation between covariate information and match outcomes. All proposed methods appeal to a class of expressive non-parametric models based on reproducing kernel Hilbert spaces. Our contributions are summarised as follows:

1. We extend SERIALRANK and SVDRANK to incorporate player covariate information and propose a new spectral ranking algorithm CCRANK. This method extracts rankings based on the canonical correlation between covariate and match information.
2. We provide an extensive set of numerical experiments on simulated and real-world data, showcasing the utility of including covariate information as well as the competitiveness of the proposed methods with state-of-the-art covariate-based ranking algorithms.

Outline. The rest of the paper is organised as follows. Section 2 covers related work in the spectral ranking literature. Our proposed methodologies will be presented in Section 3. Sections 4 and 5 cover experiments and concluding remarks.

Notation. Scalars are denoted by lower case letters while vectors and matrices are denoted by bold lower and upper case letters, respectively. In general, there are n players to rank, and we denote the $n \times n$ pairwise comparison matrix by $\mathbf{C}$. We use the notation $\succ$ to denote the direction of preference. For ordinal comparisons, $C_{i,j} = 1$ if $i \succ j$, and -1 otherwise. For simplicity, we assume no ties between players. For *cardinal* comparisons, $C_{i,j} \in \mathbb{R}$ is considered as a proxy or noisy evaluation of the skill offset $r_i - r_j$ as in [8,11]. We denote the player covariate information matrix as $\mathbf{X} \in \mathbb{R}^{n \times d}$ and its subsequent kernel matrix as $\mathbf{K} \in \mathbb{R}^{n \times n}$ with $K_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$, where $\mathbf{x}_i$ is the ith row of $\mathbf{X}$ and k the kernel. Given a skill vector $\mathbf{r}$, we define upsets as a pair of items for which the player with lower skill is preferred over the player with higher skill, i.e. there is an upset if the recovered ranking indicates $r_i > r_j$ but the input data has $C_{i,j} < 0$.. Finally, the ranking is inferred by sorting the entries of r, choosing either increasing or decreasing order to *minimise* the number of upsets with respect to the observed $\mathbf{C}$.

2   Background

We give a brief review of both probabilistic and spectral ranking algorithms in this section.

2.1   Probabilistic Ranking.

Probabilistic ranking methods typically use a generative model of how the pairwise comparison is conducted: given players' skill vector $\mathbf{r}$, also denoted as *worths* or *weights parameters*, one can model the match outcome probability based on the difference between the skills of the two players, i.e. $\mathbb{P}(i \succ j) = F(r_i - r_j)$, where F is some cumulative distribution function (CDF). The Thurstone [30] model uses a normal CDF for F while the Bradley-Terry-Luce (BT) model uses a logistic CDF. This simple setup has led to various extensions. For example, Huang et al. [16] proposed a generalisation of Bradley-Terry models to tackle paired team comparisons instead of individuals comparisons. Chen et al. [6] studied intransitivity in comparison data by extending ranking from a one-dimensional vector to a multi-dimensional representation. Bayesian approaches taking into account prior beliefs over ranking are developed in [7,3].

Extending BT models to incorporate player covariates has been studied thoroughly in the past decades as well. Springall [28] proposes to describe the skill vector $\mathbf{r}$ as the linear combination of player covariates $\mathbf{X}$, i.e. $\mathbf{r} = \mathbf{X}\boldsymbol{\beta}$ for some regression coefficients $\boldsymbol{\beta} \in \mathbb{R}^d$. Chu et al. [7] later extended this formulation to a more flexible Gaussian Processes [32] regression model as well. We will use these two extensions and the BT model as our baselines in our numerical experiments.

2.2   Spectral Ranking.

Spectral ranking is a family of fundamentally different approaches and the core object of interest are the matrices that represent pairwise relationships between

the players. It often involves computing the leading eigenvector of the sparse comparison matrix, or similar matrix operators derived from it, which can be computed efficiently in contrast to the optimisation procedures inherent in the MLE [17]. The computational advantage stems from the fact that typical eigen-solvers can compute the leading eigenvectors of a sparse matrix in running time linear in the number of nonzero entries in the matrix (ie. edges of the comparison graph). We refer the reader to Vigna [31] for a comprehensive review of the early history of spectral ranking. We now present two specific spectral ranking algorithms that our proposed methods are based on, and also discuss Jain et al. [17]'s spectral method that utilises covariates as a baseline in our numerical experiments.

SERIALRANK. Fogel et al. [12,13] proposed a seriation approach to ranking, building on [1], where they defined similarities between players based on their pairwise comparisons. Given an ordinal pairwise comparison matrix $\mathbf{C}$, we define the similarity $S_{i,j}$ between player i, j as the number of agreeing comparisons between i, j with other players, i.e.

$$S_{i,j} = \sum_{k=1}^{n} \left(\frac{1 + C_{i,k} C_{j,k}}{2} \right), \tag{1}$$

which can be expressed as $\mathbf{S} = \frac{1}{2}(n\mathbf{1}\mathbf{1}^\top + \mathbf{C}\mathbf{C}^\top)$, where $\mathbf{1} \in \mathbb{R}^n$ denotes the all-one vector. The optimal ordering in this seriation process is then recovered by extracting the *Fiedler vector* of $\mathbf{S}$, which is the eigenvector corresponding to the second smallest eigenvalue of the graph Laplacian matrix $\mathcal{L}(\mathbf{S}) = \text{diag}(\mathbf{S}\mathbf{1}) - \mathbf{S}$. This corresponds to the following optimisation

$$\mathbf{r}_* = \arg\min_{\mathbf{r}} \left(\mathbf{r}^\top \mathcal{L}(\mathbf{S})\mathbf{r} \right) \quad \text{s.t} \quad \mathbf{r}^\top \mathbf{r} = 1, \quad \mathbf{r}^\top \mathbf{1} = 0. \tag{2}$$

$\mathbf{r}_*$ can be interpreted as the smoothest non-trivial graph signal with respect to the similarity graph $\mathbf{S}$ [26].

SVDRANK. Cucuringu et al. [8,11] considered a simple alternative spectral method, arising from models wherein the pairwise comparisons are modelled as noisy entries of a rank 2 skew-symmetric matrix $\mathbf{C} = \mathbf{r}\mathbf{1}^\top - \mathbf{1}\mathbf{r}^\top$, or a proxy of it. This leads to the following optimisation

$$\mathbf{r}_* = \arg\max_{\mathbf{r}} \left(\mathbf{r}^\top \mathbf{C}\mathbf{C}^\top \mathbf{r} \right), \quad \text{s.t.} \quad \mathbf{r}^\top \mathbf{r} = 1, \quad \mathbf{r}^\top \mathbf{1} = 0. \tag{3}$$

One computes the top two singular vectors of $\mathbf{C}$ and extracts the ranking induced by ordering their entries to minimise upsets.

RANK-CENTRALITY. Negahban et al. [23] proposed to estimate player rankings from the stationary distribution of a certain random walk on the graph of players, where edges encode the probabilities of pairwise comparison outcomes. This model was first designed to combine **multiple** *ordinal* pairwise comparisons

made on the same pairs of players under the context of *Rank Aggregation* [19]. Cucuringu [27] later combined this approach with SERIALRANK to handle the single observation setting. Given the similarity matrix $\mathbf{S}$ computed using Eq (1), the key idea behind RANK-CENTRALITY is to construct a transition probability matrix $\mathbf{A}$ where $A_{i,j} = 1 - \frac{S_{i,j}}{2n}$ if $C_{i,j} > 0$, $\frac{S_{i,j}}{2n}$ if $C_{i,j} < 0$, and 0 otherwise. One then extracts the leading top left-eigenvector from $\mathbf{A}$ as the ranking. Jain et al. [17] later extended this to incorporate covariates by setting $\hat{\mathbf{A}} = \mathbf{A}\mathbf{V}$ where $\mathbf{V}$ is some row-stochastic player similarity matrix, i.e., a scaled kernel matrix. Jain et al. called this method the REGULARISED RANK-CENTRALITY algorithm. We note that this method cannot predict the ranking of unseen players because no explicit functional relationships between $\mathbf{r}$ and player covariates $\mathbf{X}$ are modelled.

3   Proposed methods

In this section, we introduce three lines of spectral ranking algorithms, each having a different flavour in combining the two sources of data: pairwise comparisons and covariate information. C-SERIALRANK controls the information covariates contribute via a simple hyperparameter (that can be selected based on cross-validation), and therefore can recover a robust ranking even when features are noisy. On the other hand, SVDCOVRANK connects $\mathbf{r}$ with covariates via a functional model, allowing us to predict the rank of previously unseen players while no comparison with existing players is needed. Finally, CCRANK considers the optimal embedding of both match outcomes and covariate information that maximises their canonical correlation and extracts ranking from those embeddings.

3.1   C-SERIALRANK

Recall in SERIALRANK, ranking is extracted from the Fiedler vector of the graph Laplacian matrix $\mathcal{L}(\mathbf{S})$, where $\mathbf{S} = \frac{1}{2}(n\mathbf{1}\mathbf{1}^{\top} + \mathbf{C}\mathbf{C}^{\top})$ is the player similarity matrix induced by their match outcomes. However, this is not the only source of player similarity. If we also have access to the covariate matrix $\mathbf{X}$, we can define another source of similarity through the Gram matrix $\mathbf{K}$, with $K_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$ for a given kernel k. We propose to merge this extra source of similarity with S via a simple linear combination, i.e. $\hat{\mathbf{S}} = \mathbf{S} + \lambda\mathbf{K}$ with λ controlling the tradeoff between the two sources of data. We call this method C-SERIALRANK, where C stands for covariates. The ranking can then be recovered by computing the Fiedler vector from the new graph Laplacian matrix $\mathcal{L}(\hat{\mathbf{S}})$

$$\mathbf{r}_* = \arg\min_{\mathbf{r}} \mathbf{r}^T\big(\mathcal{L}(\mathbf{S} + \lambda\mathbf{K})\big)\mathbf{r} \quad \text{s.t} \quad \mathbf{r}^{\top}\mathbf{r} = 1 \quad \mathbf{r}^{\top}\mathbf{1} = 0. \tag{4}$$

We summarise the method in Algorithm 1. The tuning parameter λ controls the contribution of the covariates – intuitively, it should be high for covariates that are informative about match outcomes, and low for less informative covariates.

Algorithm 1: C-SERIALRANK

Require: A set of pairwise comparisons $\mathbf{C}_{i,j} \in \{-1, 0, +1\}$ or $\mathbb{R}$, kernel matrix $\mathbf{K}$, regularisation parameter λ.

1 Compute similarity matrix $\hat{\mathbf{S}} = \frac{1}{2}(n\mathbf{1}\mathbf{1}^T + \mathbf{C}\mathbf{C}^T) + \lambda\mathbf{K}$.

2 Compute the associated graph Laplacian matrix $\mathcal{L}(\hat{\mathbf{S}}) = \text{diag}(\hat{\mathbf{S}}\mathbf{1}) - \hat{\mathbf{S}}$.

3 Compute the Fiedler vector of $\mathbf{S}$ by extracting the eigenvector corresponding to the second smallest eigenvalue of $\mathcal{L}(\hat{\mathbf{S}})$.

4 Output the ranking induced by sorting the Fiedler vector of $\hat{\mathbf{S}}$, with the global ordering chosen to minimise upsets with respect to $\mathbf{C}$.

3.2   SVDCovRank

C-SERIALRANK provides a simple method to incorporate player covariates into ranking. However, similar to the REGULARISED RANK-CENTRALITY algorithm, the downside of C-SERIALRANK is that one cannot perform predictions on unseen players based on available covariate information. To this end, we introduce SVDCovRANK, an SVD based ranking algorithm that connects covariates to $\mathbf{r}$ using a functional model.

Recall that when solving for SVDRANK, one converts Eq (3) to a standard eigenvector problem by projecting $\mathbf{r}$ onto $\{\mathbf{1}\}^\perp$. This can be done by setting $\mathbf{r} = \mathbf{Z}\mathbf{q}$, where $\mathbf{Z} \in \mathbb{R}^{n \times (n-1)}$ is a matrix whose columns form an orthonormal basis of $\{\mathbf{1}\}^\perp$. As a result, Eq (3) can be rewritten as

$$\mathbf{q}_* = \arg\max_{\mathbf{q}} \left(\mathbf{q}^\top \mathbf{Z}^\top \mathbf{C}\mathbf{C}^\top \mathbf{Z}\mathbf{q}\right), \quad \text{s.t. } \mathbf{q}^\top \mathbf{q} = 1. \tag{5}$$

By construction, $\mathbf{r}_* = \mathbf{Z}\mathbf{q}_*$ is orthogonal to $\mathbf{1}$ where $\mathbf{q}_*$ is the top eigenvector of $\mathbf{Z}^\top \mathbf{C}\mathbf{C}^\top \mathbf{Z}$. Now consider the scenario where we also have access to player covariates $\mathbf{X}$, and we assume the final ranking implied by $\mathbf{r}$ varies smoothly as a linear function of these covariates, e.g. $\mathbf{r} = \mathbf{X}\boldsymbol{\beta}$. Following the same line of thought as above, one could set $\mathbf{X}\boldsymbol{\beta} = \mathbf{Z}\mathbf{q}$, and substitute $\mathbf{q} = \mathbf{Z}^\top \mathbf{X}\boldsymbol{\beta}$ into Eq (5) to arrive at

$$\boldsymbol{\beta}_* = \arg\max_{\boldsymbol{\beta}} \left(\boldsymbol{\beta}^\top \mathbf{X}^\top \mathbf{H}\mathbf{C}\mathbf{C}^\top \mathbf{H}\mathbf{X}\boldsymbol{\beta}\right), \quad \text{s.t. } \boldsymbol{\beta}^\top \mathbf{X}\mathbf{H}\mathbf{X}\boldsymbol{\beta} = 1, \tag{6}$$

where $\mathbf{H} = \mathbf{Z}\mathbf{Z}^\top$ is the centering matrix. To solve for $\boldsymbol{\beta}$, we further decompose our feature covariance matrix $\mathbf{X}^\top \mathbf{H}\mathbf{X}$ into $\mathbf{L}\mathbf{L}^\top$ via a Choleskey decomposition or SVD. By setting $\boldsymbol{\beta} = \mathbf{L}^{-\top}\boldsymbol{\gamma}$, we again recover the standard eigenvector problem

$$\boldsymbol{\gamma}_* = \arg\max_{\boldsymbol{\gamma}} \left(\boldsymbol{\gamma}^\top \boldsymbol{\Psi}\boldsymbol{\gamma}\right), \quad \text{s.t. } \boldsymbol{\gamma}^\top \boldsymbol{\gamma} = 1, \tag{7}$$

with $\boldsymbol{\Psi} = \mathbf{L}^{-1}\mathbf{X}^\top \mathbf{H}\mathbf{C}\mathbf{C}^\top \mathbf{H}\mathbf{X}\mathbf{L}^{-\top}$. Finally, the optimal ranking $\mathbf{r}$ is recovered by setting $\mathbf{r}_* = \mathbf{H}\mathbf{X}\mathbf{L}^{-\top}\boldsymbol{\gamma}_*$. We summarise the procedure in Algorithm 2.

Furthermore, if there is reason to believe the skill vector $\mathbf{r}$ relates to the covariates non-linearly, it is straightforward to kernelise the procedure by setting

---

Algorithm 2: SVDCovRank

Require: A set of pairwise comparisons $\mathbf{C}_{i,j} \in \{-1, 0, 1\}$ or $\mathbb{R}$, covariate matrix $\mathbf{X}$ and kernel function k.

1 if *kernelise* **then**
2 $\quad\lfloor$ Compute the kernel matrix $\mathbf{K}$ with $K_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$ and set $\mathbf{B} = \mathbf{K}$
3 else
4 $\quad\lfloor$ Set $\mathbf{B} = \mathbf{X}$.
5 Compute $\boldsymbol{\Psi} = \mathbf{L}^{-1}\mathbf{B}^\top\mathbf{H}\mathbf{C}\mathbf{C}^\top\mathbf{H}\mathbf{B}\mathbf{L}^{-\top}$, where $\mathbf{L}$ is the Cholesky decomposition of $\mathbf{B}^\top\mathbf{H}\mathbf{B}$.
6 Compute the top eigenvector $\boldsymbol{\gamma}$ of $\boldsymbol{\Psi}$ and set $\mathbf{r} = \mathbf{H}\mathbf{B}\mathbf{L}^{-\top}\boldsymbol{\gamma}$.
7 Sort $\mathbf{r}$ to minimise the number of upsets with respect to $\mathbf{C}$.
8 if *kernelise* **then**
9 $\quad\lfloor$ Return the skill vector $\mathbf{r}$ and skill function $r(\mathbf{x}) = k(\mathbf{x}, \mathbf{X})\mathbf{L}^\top\boldsymbol{\gamma}$ where $k(\mathbf{x}, \mathbf{X}) = [k(\mathbf{x}, \mathbf{x}_1), ..., k(\mathbf{x}, \mathbf{x}_n)]$.
10 else
11 $\quad\lfloor$ Return the skill vector $\mathbf{r}$ and skill function $r(\mathbf{x}) = \mathbf{x}^\top\mathbf{L}^\top\boldsymbol{\gamma}$.

$\mathbf{r} = \mathbf{K}\boldsymbol{\alpha}$, where $\mathbf{K}$ is the kernel matrix of player covariates and $\boldsymbol{\alpha} \in \mathbb{R}^n$ some dual weights, and obtain a non-parametric variation of SVDCovRank. Similarly by setting $\mathbf{K}\boldsymbol{\alpha} = \mathbf{Z}^\top\mathbf{q}$, we can express Eq (5) as

$$\boldsymbol{\alpha}_* = \arg\max_{\boldsymbol{\alpha}} \left(\boldsymbol{\alpha}^\top\mathbf{K}\mathbf{H}\mathbf{C}\mathbf{C}^\top\mathbf{H}\mathbf{K}\boldsymbol{\alpha}\right), \quad \text{s.t. } \boldsymbol{\alpha}^\top\mathbf{K}\mathbf{H}\mathbf{K}\boldsymbol{\alpha} = 1. \tag{8}$$

The rest follows in analogy to the linear case and the complete method is given in Algorithm 2.

3.3   CCRank

Besides aggregating match and covariate information via a linear combination or by imposing a regression structure between $\mathbf{r}$ and $\mathbf{X}$, we propose to study the Canonical Correlation between match and covariate information to derive the ranking. We call the method CCRank. Consider the similarity matrix $\mathbf{S} = \frac{1}{2}(n\mathbf{1}\mathbf{1}^\top + \mathbf{C}\mathbf{C}^\top)$ from Section 2.2 and player covariate matrix $\mathbf{X}$. As each row of $\mathbf{S}$ and $\mathbf{X}$ describe the same player with two distinct sources of information, it is natural to define ranking as the "most agreeing" projections between $\mathbf{S}$ and $\mathbf{X}$. This coincides with the classical Canonical Correlation Analysis (CCA) problem.

CCA is a classical technique developed by H. Hotelling [15] to study the linear relationship between two multidimensional sets of variables. Assume both matrix $\mathbf{S}$ and $\mathbf{X}$ are full rank, CCA can be defined as the problem of finding the optimal projection $\mathbf{z}_S = \mathbf{S}\mathbf{a}$ and $\mathbf{z}_X = \mathbf{X}\mathbf{b}$, with $\mathbf{a} \in \mathbb{R}^n$ and $\mathbf{b} \in \mathbb{R}^d$, such that $\mathbf{z}_S$ and $\mathbf{z}_X$ are maximally correlated, i.e.

$$\hat{\mathbf{a}}, \hat{\mathbf{b}} = \arg\max_{\mathbf{a},\mathbf{b}} \frac{\mathbf{z}_S^\top\mathbf{z}_X}{||\mathbf{z}_S||||\mathbf{z}_X||} = \arg\max_{\mathbf{a},\mathbf{b}} \frac{\mathbf{a}^\top\boldsymbol{\Sigma}_{SX}\mathbf{b}}{\sqrt{\mathbf{a}^\top\boldsymbol{\Sigma}_{SS}\mathbf{a}\mathbf{b}^\top\boldsymbol{\Sigma}_{XX}\mathbf{b}}}, \tag{9}$$

where $\boldsymbol{\Sigma}_{SX} = \mathbf{SX}$ is an estimate of the cross-covariance matrix between $\mathbf{S}$ and $\mathbf{X}$. Eq (9) is equivalent to the following constrained optimisation problem

$$\hat{\mathbf{a}}, \hat{\mathbf{b}} = \underset{\mathbf{a},\mathbf{b}}{\arg\max} = \mathbf{a}^{\top} \boldsymbol{\Sigma}_{SX} \mathbf{b} \quad \text{s.t.} \quad \begin{cases} \mathbf{a}^{\top} \boldsymbol{\Sigma}_{SS} \mathbf{a} = 1 \\ \mathbf{b}^{\top} \boldsymbol{\Sigma}_{XX} \mathbf{b} = 1. \end{cases} \tag{10}$$

The solution to this problem can then be obtained by solving a generalised eigenvector problem as detailed in Algorithm 3. We recover ranking by setting $\mathbf{r}$ to be $\mathbf{z}_S$ or $\mathbf{z}_X$ depending on which embedding better minimises the upsets with respect to $\mathbf{C}$. However, note that $\mathbf{z}_S = \mathbf{Sa}$ cannot be used to predict rankings on unseen players by construction.

Similar to SVDCovRank, if we believe the covariates relate to the optimal ranking non-linearly, we could kernelise the procedure and seek projections belonging to RKHSs. Denote ℓ as the kernel on $\mathbf{S}$ and $\mathcal{H}_{\ell}$ the corresponding RKHS. The objective of Kernel CCA [14] can be expressed using cross-covariance operators, which for simplicity, we use the same notation as the corresponding cross-covariance matrices

$$\hat{f}, \hat{g} = \underset{f \in \mathcal{H}_k, g \in \mathcal{H}_{\ell}}{\arg\max} \langle f, \boldsymbol{\Sigma}_{XS} g \rangle_{\mathcal{H}_{\ell}} \quad \text{s.t} \quad \begin{cases} \langle f, \boldsymbol{\Sigma}_{XX} f \rangle_{\mathcal{H}_k} = 1 \\ \langle g, \boldsymbol{\Sigma}_{SS} g \rangle_{\mathcal{H}_{\ell}} = 1. \end{cases} \tag{11}$$

The rest of the algorithm is outlined in Algorithm 3. Different from SVDCovRank which postulates a low-rank structure on $\mathbf{C}$, CCRank does not pose any structure over the match matrix, and thus is more robust to model misspecification. This is illustrated later in our experiments.

4   Experiments

In this section, we compare the performance of C-SerialRank (C-Serial from now on for brevity), SVDCovRank (SVDC), kernelised SVDCovRank (SVDK), CCRank (CC), kernelised CCRank (KCC) with that of a number of other benchmark algorithms from the literature. We compare against the spectral ranking algorithms SerialRank (Serial), SVDRank (SVDN), Rank-Centrality (RC), Regularised Rank-Centrality (RRC) and against the probabilistic ranking algorithms Bradley-Terry-Luce model (BT) as well as its two extension based on logistic regression (BT-LR) and Gaussian processes [7] (BT-GP). For BT-GP we will simply utilise its posterior mean for rank prediction. We note that Bradley-Terry-Luce and Rank Centrality based methods cannot handle *cardinal* comparisons, therefore we first transform each entry of the pairwise comparison matrix to $\text{sign}(C_{i,j})$. See Table 1 for a summary of our methods and benchmarks. All code and implementations are made publicly available.[1]

Throughout our experiments, we will use the Radial Basis Function (RBF) kernel $k(x, x') = \exp\left(\frac{||x - x'||^2}{2\ell_k}\right)$ where the lengthscale ℓ_k is chosen via the median

---

Algorithm 3: CCRANK

Require: A set of pairwise comparisons $C_{i,j} \in \{-1, 0, 1\}$ or $\mathbb{R}$, covariate matrix $\mathbf{X}$, kernel functions k, g, centering matrix $\mathbf{H}$, identity matrix $\mathbf{I}_n$ and small perturbation ϵ

1 Compute $\mathbf{S} = \frac{1}{2}(n\mathbf{1}\mathbf{1}^\top + \mathbf{C}\mathbf{C}^\top)$

2 **if** *kernelise* **then**

3   Compute the kernel matrix $\mathbf{K}$ with $K_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$ and $\mathbf{G}$ with $G_{i,j} = g(\mathbf{s}_i, \mathbf{s}_j)$ where $\mathbf{s}_i, \mathbf{s}_j$ corresponds to row i, j of $\mathbf{S}$.

4   Compute $\tilde{\mathbf{K}} = \mathbf{HKH}$ and $\tilde{\mathbf{G}} = \mathbf{HGH}$

5   Compute $\boldsymbol{\Sigma}_{XS} = n^{-1}\tilde{\mathbf{K}}\tilde{\mathbf{G}}, \boldsymbol{\Sigma}_{XX} = n^{-1}\tilde{\mathbf{K}}(\tilde{\mathbf{K}} + \epsilon\mathbf{I}_n)$ and $\boldsymbol{\Sigma}_{SS} = n^{-1}\tilde{\mathbf{G}}(\tilde{\mathbf{G}} + \epsilon\mathbf{I}_n)$

6 **else**

7   Compute $\boldsymbol{\Sigma}_{SX} = \mathbf{SX}, \boldsymbol{\Sigma}_{XX} = \mathbf{X}^\top\mathbf{X}$ and $\boldsymbol{\Sigma}_{SS} = \mathbf{S}^\top\mathbf{S}$.

8 Compute the top generalised eigenvector of the problem

$$\begin{bmatrix} 0 & \boldsymbol{\Sigma}_{XS} \\ \boldsymbol{\Sigma}_{XS}^\top & 0 \end{bmatrix}\begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix} = \rho\begin{bmatrix} \boldsymbol{\Sigma}_{XX} & 0 \\ 0 & \boldsymbol{\Sigma}_{SS} \end{bmatrix}\begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix} \tag{12}$$

    if *kernelise* **then**

9   Return the embeddings $\mathbf{z}_S = \mathbf{GH}\boldsymbol{\beta}$ and $\mathbf{z}_X = \mathbf{KH}\boldsymbol{\alpha}$ that minimises upsets with respect to $\mathbf{C}$ and skill function $r(\mathbf{x}) = k(\mathbf{x}, \mathbf{X})\mathbf{H}\boldsymbol{\alpha}$

10 **else**

11   Return the embeddings $\mathbf{z}_S = \mathbf{S}\boldsymbol{\beta}$ and $\mathbf{z}_X = \mathbf{X}\boldsymbol{\alpha}$ that minimises upsets with respect to $\mathbf{C}$ and skill function $r(\mathbf{x}) = \mathbf{x}^\top\boldsymbol{\alpha}$.

---

heuristic method. Hyperparameters such as λ in C-SERIAL are tuned using 10-fold cross-validation on withheld match outcomes. In practice, we should select the kernel that reflects the nature of the non-linearity in data. We assess the performance of all algorithms via the following two tasks:

Task 1: Rank Inference. Given n players with a sparse and potentially noisy pairwise comparison matrix $\mathbf{C}$, we pick **70% of the observed matches** as training data $\mathbf{C}_{\text{train}}^{(1)}$, and the remaining observed matches as testing data $\mathbf{C}_{\text{test}}^{(1)}$. Our goal is to train the algorithms on $\mathbf{C}_{\text{train}}^{(1)}$ to recover the rankings, and then measure their performances by reporting the accuracy scores in predicting matches from $\mathbf{C}_{\text{test}}^{(1)}$ based on the estimated rankings.

Task 2: Rank Prediction. Given n players with comparison matrix $\mathbf{C}$, withhold **70% of the players** and observed the induced comparison matrix $\mathbf{C}_{\text{train}}^{(2)} \in \mathbb{R}^{n' \times n'}$ where $n' = \lfloor n \times 0.7 \rfloor$. Set up the testing data $\mathbf{C}_{\text{test}}^{(2)}$ based on the induced comparison matrix of the remaining players. We train on $\mathbf{C}_{\text{train}}^{(2)}$ and report the accuracy in predicting matches from $\mathbf{C}_{\text{test}}^{(2)}$ based on the predicted rankings on unseen players.

Table 1: A summary of our proposed methods (shaded grey) and benchmarks.

| Algorithm	Model type	Comparisons	Covariates	Unseen player rank prediction
BT [2]	Probabilistic	Ordinal	✗	✗
BT-LR \ BT-GP [7]	Probabilistic	Ordinal	✓	✓
RC [22]	Spectral	Ordinal	✗	✗
RRC [17]	Spectral	Ordinal	✓	✗
SERIAL [13]	Spectral	Ordinal + Cardinal	✗	✗
C-SERIAL (Ours)	Spectral	Ordinal + Cardinal	✓	✗
SVD [10]	Spectral	Ordinal + Cardinal	✗	✗
SVDC \ SVDK (Ours)	Spectral	Ordinal + Cardinal	✓	✓
CC \ KCC (Ours)	Spectral	Ordinal + Cardinal	✓	✓

4.1   Simulations on synthetic data

We test the performances of the 12 ranking algorithms on both rank inference and prediction problems using synthetic data. We report the Kendall Tau score on test data as our performance metric, as we have access to the ground truth ranking. We simulate $n = 1000$ players with covariates $\mathbf{X}$ drawn from $N(0,1)$. We set the skill function as $r(x) = \sin(3\pi x) - 1.5x^2 + \epsilon N(0,1)$ with the noise level $\epsilon \in \{0, 0.05, 0.2\}$. The corresponding comparison matrix $\mathbf{C}$ is given by $C_{i,j} = r(x_i) - r(x_j)$. Besides feature noise, we also consider adding noises to the matches. We follow the *FLIP* noise model deployed in [9] where we flip each match result with probability $p \in \{0, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5\}$. With probability $\phi \in \{0.7, 0.9\}$, we remove an observed match in the comparison matrix $\mathbf{C}$ to yield sparse comparisons matrices as in most practical situations. Experiments are repeated over 10 seeds.

Fig. 1a and Fig. 1b demonstrate the performances of the algorithms under different sparsity levels, feature noises and model error rates for the *inference* and *prediction* tasks. For the *inference* tasks, we see a performance drop in the covariate-based ranking algorithms as the feature noise increases. This coincides with our usual "garbage covariate-in, garbage out" intuition in modelling. We also note that as the skill vector non-linearly relates to the covariates by design, all kernelised algorithms outperformed their linear counterparts. We also observe that as the flip error rate p increases, the performance of covariate-free algorithms decreases quickly, while covariate-based methods performed relatively stable until the extreme error rate. For *prediction* tasks, similar trends to the *inference* tasks in terms of algorithmic performances can be seen with respect to model error rates, sparsity and feature noise levels.

Specifically, we observe that the kernelised spectral ranking algorithm SVDK performed competitively to probabilistic ranking BT-GP until a high flip error rate in both tasks. KCC performed slightly worse than the two, but still much better than the covariate-based spectral method baseline RRC in general. We note that the linear methods do not appear to be sensitive to model error rate because the ranking extracted is dominated by the information in the covariates.
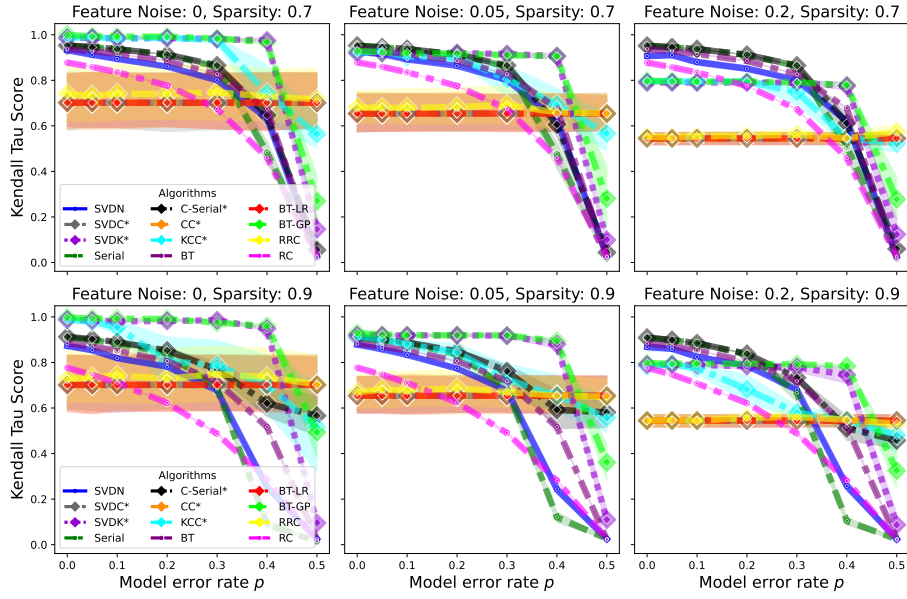
(a) Rank Inference
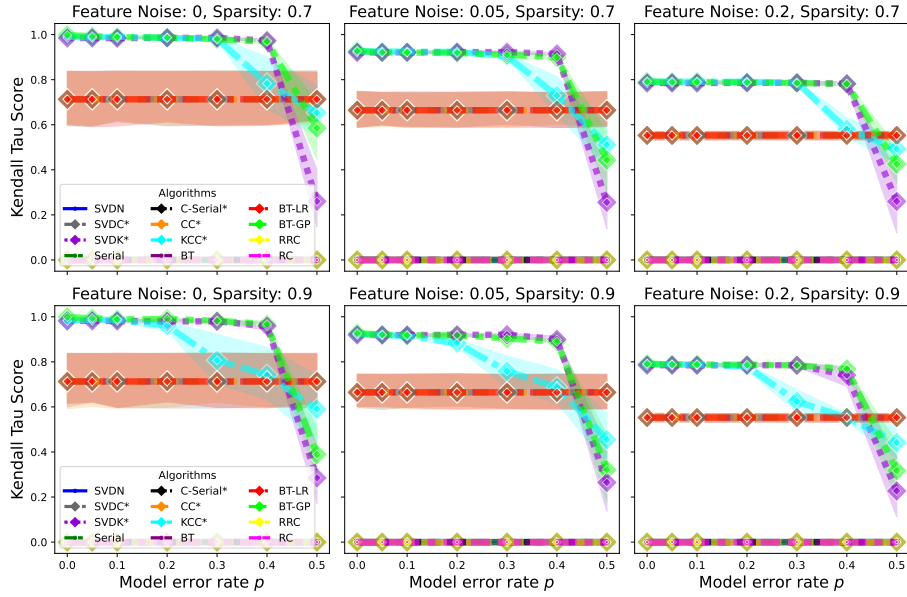
(b) Rank Prediction

Fig. 1: Performance in terms of Kendall's tau score (the closer to 1, the better) under different synthetic noise models with 1000 players. Results are averaged over 10 iterations and 95% confidence intervals are reported. Methods with diamond markers use covariates, and ∗ indicates our proposed methods.

Table 2: Accuracy of match outcome predictions based on recovered ranking on each data set. Scores are averaged over 10 seeds, and 1 s.d is reported. † indicates algorithm using covariate information. We shaded our proposed methods in grey. The best performing method is shown in bold red, and second best in bold blue.

Algorithm	Chameleon		FlatLizard		Pokemon		NFL 2000-18	
	Task 1	Task 2	Task 1	Task 2	Task 1	Task 2	Task 1	Task 2
BT	$0.83_{\pm0.03}$	-	$0.86_{\pm0.06}$	-	$0.86_{\pm0.01}$	-	$0.63_{\pm0.04}$	-
BT-LR†	$0.71_{\pm0.03}$	$0.82_{\pm0.14}$	$0.84_{\pm0.04}$	$0.77_{\pm0.12}$	$0.88_{\pm0.01}$	$0.89_{\pm0.01}$	$0.68_{\pm0.04}$	$0.62_{\pm0.09}$
BT-GP†	$0.75_{\pm0.04}$	$0.74_{\pm0.13}$	$0.80_{\pm0.05}$	$0.74_{\pm0.12}$	$0.72_{\pm0.01}$	$0.72_{\pm0.02}$	$0.58_{\pm0.04}$	$0.61_{\pm0.08}$
RC	$0.61_{\pm0.06}$	-	$0.66_{\pm0.05}$	-	$0.81_{\pm0.01}$	-	$0.58_{\pm0.04}$	-
RRC†	$0.61_{\pm0.03}$	-	$0.66_{\pm0.01}$	-	$0.85_{\pm0.01}$	-	$0.62_{\pm0.04}$	-
SVD	$0.72_{\pm0.08}$	-	$0.69_{\pm0.05}$	-	$0.84_{\pm0.02}$	-	$0.57_{\pm0.04}$	-
SVDC†	$0.65_{\pm0.06}$	$0.74_{\pm0.12}$	$0.81_{\pm0.05}$	$0.73_{\pm0.10}$	$0.89_{\pm0.01}$	$0.90_{\pm0.01}$	$0.66_{\pm0.04}$	$0.62_{\pm0.09}$
SVDK†	$0.76_{\pm0.06}$	$0.79_{\pm0.06}$	$0.68_{\pm0.05}$	$0.65_{\pm0.09}$	$0.85_{\pm0.01}$	$0.72_{\pm0.09}$	$0.59_{\pm0.04}$	$0.62_{\pm0.09}$
SERIAL	$0.79_{\pm0.04}$	-	$0.70_{\pm0.05}$	-	$0.88_{\pm0.01}$	-	$0.58_{\pm0.04}$	-
C-SERIAL†	$0.80_{\pm0.03}$	-	$0.88_{\pm0.01}$	-	$0.88_{\pm0.01}$	-	$0.59_{\pm0.04}$	-
CC†	$0.66_{\pm0.10}$	$0.95_{\pm0.10}$	$0.78_{\pm0.08}$	$0.92_{\pm0.15}$	$0.78_{\pm0.08}$	$0.79_{\pm0.08}$	$0.61_{\pm0.05}$	$0.66_{\pm0.10}$
KCC†	$0.71_{\pm0.06}$	$0.80_{\pm0.10}$	$0.78_{\pm0.03}$	$0.71_{\pm0.11}$	$0.81_{\pm0.01}$	$0.72_{\pm0.09}$	$0.67_{\pm0.05}$	$0.62_{\pm0.08}$

4.2   Experiments on real-world data

In this section, we apply all ranking algorithms to a variety of real-world data sets (22 in total) to perform rank *inferences* and *predictions*. We average the accuracy scores over 10 seeds and report the results in Table 2. 5% significant level one-sided paired Wilcoxon tests for all algorithm pairs are conducted for each data set, with results reported in Fig. 2.

Male Cape Dwarf Chameleons Contest. The first data set is used in the study by Stuart-Fox et al. [29]. Physical measurements are made on 35 male Cape dwarf chameleons, and the results of 104 contests are measured and encoded as ordinal comparison data. From Table 2 and Fig. 2 we see that BT outperformed the rest in rank *inferences* (closely followed by C-SERIAL), while CC performed best compared with other covariate based methods in rank *prediction*.

Flatlizard Competition. The data is collected at Augrabies Falls National Park (South Africa) in September-October 2002 WM09, on the contest performance and background attributes of 77 male flat lizards (*Platysaurus Broadleyi*). The results of 100 contests were recorded, along with 18 physical measurements made on each lizard, such as *weight* and *head size*. From Fig 2 we see that C-SERIAL and CC statistically significantly outperformed the rest for inference and prediction, respectively.

Pokémon Battle. Our next data set comes from a Kaggle competition[2]. Approximately $45,000$ battles among 800 Pokémon are recorded in ordinal format. 25 characteristic of each Pokémon, such as their *type*, *attack* and *defence*, are

---

[2] data collected from `https://www.kaggle.com/c/intelygenz-pokemon-challenge`

(a) Chameleon‡

(b) Flatlizard‡

(c) Pokemon‡

(d) NFL ‡

(e) Chameleon¶

(f) Flatlizard¶

(g) Pokemon¶

(h) NFL¶

Fig. 2: Statistical Significance plots for our real-world experiments. We shade an entry if the ith row algorithm is statistically significantly better than the jth column algorithm on a one-sided paired Wilcoxon test with a 5% significance level. We shade our algorithms in red and the benchmarks in black. ‡ and ¶ indicate rank inference and prediction tasks respectively.

available. These battles are generated by a custom algorithm written by the Kaggle host that closely reflects the game mechanics. Table 2 shows that linear covariate-based models, in general, outperformed their kernelised counterparts. This is not surprising as the "strength" of a pokemon is linearly related to the covariates provided (e.g. *attack* and *defence)*. Given these highly informative features, covariate-based methods, in general, outperformed covariate-free methods as well. We note that SVDC, closely followed by BT-LR, was the top performer for both inference and prediction tasks.

NFL Football 2000-2018. The last data set we use contains the outcome of National Football League (NFL) matches during the regular season, for the years $2000 - 2018$[3]. In addition, 256 matches per year between 32 teams, along with 18 performance metrics, such as *yards per game* and *number of fumbles* are recorded. We used their score differences at each match as the cardinal comparisons. We run our algorithms on each year's comparison graphs separately and average the results. We see for rank *inference*, our covariate-based algorithms outperformed their covariate-free counterparts, with BT-LR the top performer, followed closely by SVDC, KCC and RRC. For rank *prediction*, CC was significantly better than all other methods.

5 Conclusions and Future directions

We proposed three spectral ranking algorithms to tackle the problem of ranking given noisy and incomplete pairwise comparisons when player covariates are available. We demonstrated the efficacy, strengths and weaknesses of the proposed methods in comparison to baselines through an extensive set of numerical experiments on synthetic and 22 real-world problems.

C-SERIALRANK extends SERIALRANK by incorporating covariate-induced player similarities into the seriation process through a linear combination. Although simple, through a straight forward cross-validation scheme on the regularisation parameter, we can ensure the ranking recovered in this way is at least as good as the ranking recovered from SERIALRANK, as demonstrated in our experiments. The downside of C-SERIALRANK, similar to REGULARISED RANK CENTRALITY, is that it cannot be used on rank prediction. However, the results of our experiments show that the former outperformed the latter in the majority of the experiments. Our second contribution SVDCOVRANK extends SVDRANK by relating covariates to the eigenvector of the pairwise comparison matrix $\mathbf{C}$ in a regression fashion. This allows us to predict the rankings of unseen players, which previous spectral ranking with covariates cannot handle. As showed in our experiments, SVDCOVRANK performed comparably to covariate-based probabilistic methods. At last, CCRANK approaches the ranking problem using the canonical correlation analysis between match outcomes and player covariates. CCRANK places fewer assumptions on the underlying data generating process and is thus more robust in practice. This is demonstrated by its competitive performance against other methods when predicting the rank of unseen players.

---

[3] data collected from `nfl.com`

Each proposed method will find its application in different practical scenarios. When rank prediction on unseen players is not a concern, we recommend practitioners to use the C-SERIALRANK algorithm because of its simplicity in balancing covariate and match information. On the other hand, if one believes the pairwise comparisons are caused by a strong underlying skill difference, then SVDCovRANK algorithms should be considered. At last, if one seeks to place fewer assumptions on the data, then the CCRANK algorithm may give a more robust ranking than the approaches that place specific structures on the match outcome matrix $\mathbf{C}$.

There are several avenues for future work. An interesting direction would be to incorporate more general types of side information into the ranking problem, e.g., prior information on partially observed player relationships. Yet another relevant direction is the extraction of partial rankings using the side covariates. In many real-world scenarios, aiming for a global ordering of the players is unrealistic, and one is often interested in uncovering partial orderings that reflect accurately various subsets/clusters of a less homogeneous population of players. Leveraging features for the discovery of latent structures behind cyclical and inconsistent preferences, as in [5] but using spectral methods, is also an interesting direction to explore.

References

1. Atkins, J.E., Boman, E.G., Hendrickson, B.: A spectral algorithm for seriation and the consecutive ones problem. SIAM Journal on Computing **28**(1), 297–310 (1998)
2. Bradley, R.A., Terry, M.E.: Rank analysis of incomplete block designs: I. the method of paired comparisons. Biometrika **39**(3/4), 324–345 (1952)
3. Caron, F., Doucet, A.: Efficient bayesian inference for generalized bradley–terry models. Journal of Computational and Graphical Statistics **21**(1), 174–196 (2012)
4. Cattelan, M.: Models for paired comparison data: A review with emphasis on dependent data. Statistical Science **27**(3), 412–433 (2012)
5. Chau, S., González, J., Sejdinovic, D.: Learning inconsistent preferences with gaussian processes (2022)
6. Chen, S., Joachims, T.: Modeling intransitivity in matchup and comparison data. In: Proceedings of the Ninth ACM International Conference on Web Search and Data Mining. p. 227–236 (2016)
7. Chu, W., Ghahramani, Z.: Preference learning with Gaussian processes. In: Proceedings of the 22nd International Conference on Machine Learning. pp. 137–144 (2005)
8. Cucuringu, M.: Sync-Rank: Robust Ranking, Constrained Ranking and Rank Aggregation via Eigenvector and Semidefinite Programming Synchronization. IEEE Transactions on Network Science and Engineering **3**(1), 58–79 (2016)
9. Cucuringu, M.: Sync-rank: Robust ranking, constrained ranking and rank aggregation via eigenvector and sdp synchronization. IEEE Transactions on Network Science and Engineering **3**(1), 58–79 (2016)
10. Cucuringu, M., Koutis, I., Chawla, S., Miller, G., Peng, R.: Simple and scalable constrained clustering: a generalized spectral method. In: Artificial Intelligence and Statistics. pp. 445–454 (2016)

11. d'Aspremont, A., Cucuringu, M., Hemant, T.: Ranking and synchronization from pairwise measurements via SVD. arXiv:1906.02746 (2019)
12. Fogel, F., d'Aspremont, A., Vojnovic, M.: Serialrank: Spectral ranking using seriation. In: Advances in Neural Information Processing Systems 27. pp. 900–908 (2014)
13. Fogel, F., d'Aspremont, A., Vojnovic, M.: Spectral ranking using seriation. Journal of Machine Learning Research **17**(88), 1–45 (2016)
14. Fukumizu, K., Bach, F.R., Gretton, A.: Statistical consistency of kernel canonical correlation analysis. Journal of Machine Learning Research **8**(Feb), 361–383 (2007)
15. Hotelling, H.: Relations between two sets of variates. In: Breakthroughs in statistics, pp. 162–190. Springer (1992)
16. Huang, T.K., Weng, R.C., Lin, C.J., Ridgeway, G.: Generalized bradley-terry models and multi-class probability estimates. Journal of Machine Learning Research **7**(1) (2006)
17. Jain, L., Gilbert, A., Varma, U.: Spectral methods for ranking with scarce data. In: Conference on Uncertainty in Artificial Intelligence. pp. 609–618. PMLR (2020)
18. Kendall, M.G., Smith, B.B.: On the method of paired comparisons. Biometrika **31**(3-4), 324–345 (1940)
19. Li, X., Wang, X., Xiao, G.: A comparative study of rank aggregation methods for partial and top ranked lists in genomic applications. Briefings in bioinformatics **20**(1), 178–189 (2019)
20. Mallows, C.: Non null ranking models I. Biometrika **44**, 114–130 (1957)
21. Mosteller, F., Nogee, P.: An experimental measurement of utility. Journal of Political Economy **59**(5), 371–404 (1951)
22. Negahban, S., Oh, S., Shah, D.: Iterative ranking from pair-wise comparisons. In: Advances in Neural Information Processing Systems 25. pp. 2474–2482 (2012)
23. Negahban, S., Oh, S., Shah, D.: Rank centrality: Ranking from pairwise comparisons. Operations Research **65**(1), 266–287 (2017). `https://doi.org/10.1287/opre.2016.1534`
24. Niranjan, U., Rajkumar, A.: Inductive pairwise ranking: going beyond the n log (n) barrier. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)
25. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank citation ranking: Bringing order to the Web. In: Proceedings of the 7th International World Wide Web Conference. pp. 161–172 (1998)
26. Shuman, D.I., Narang, S.K., Frossard, P., Ortega, A., Vandergheynst, P.: The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. IEEE signal processing magazine **30**(3), 83–98 (2013)
27. Singer, A.: Angular synchronization by eigenvectors and semidefinite programming. Appl. Comput. Harmon. Anal. **30**(1), 20–36 (2011)
28. Springall, A.: Response surface fitting using a generalization of the bradley-terry paired comparison model. Journal of the Royal Statistical Society: Series C (Applied Statistics) **22**(1), 59–68 (1973)
29. Stuart-Fox, D.M., Firth, D., Moussalli, A., Whiting, M.J.: Multiple signals in chameleon contests: designing and analysing animal contests as a tournament. Animal Behaviour **71**(6), 1263–1271 (2006)
30. Thurstone, L.: A law of comparative judgement. Psychological Review **34**, 278–286 (1927)
31. Vigna, S.: Spectral ranking. Network Science **4**(4), 433–445 (2016)
32. Williams, C.K., Rasmussen, C.E.: Gaussian Processes for Machine Learning. MIT Press (2006)