Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17)

Score Aggregation via Spectral Method*

Mingyu Xiao and **Yuqing Wang**
School of Computer Science and Engineering,
University of Electronic Science and Technology of China, China
myxiao@gmail.com, yyqqwang@126.com

Abstract

The score aggregation problem is to find an aggregate scoring over all candidates given individual scores provided by different agents. This is a fundamental problem with a broad range of applications in social choice and many other areas. The simple and commonly used method is to sum up all scores of each candidate, which is called the sum-up method. In this paper, we give good algebraic and geometric explanations for score aggregation, and develop a spectral method for it. If we view the original scores as 'noise data', our method can find an 'optimal' aggregate scoring by minimizing the 'noise information'. We also suggest a signal-to-noise indicator to evaluate the validity of the aggregation or the consistency of the agents.

1 Introduction

In a *score aggregation* system, several agents give scores to a set of candidates independently based on its own personal criteria, and we are going to combine the individual scores to get a final score for each candidate. It is a fundamental process that is integral in everything and intrinsic in our daily life. For example, in schools, each student has scores on different courses and we want to use a single score, such as GPA, to evaluate the overall performance of the student. It has direct application in scoring-based rank aggregation problems. Score aggregation not only plays an important role in social choice theory [Moulin *et al.*, 2016], but also receives a wide range of applications in multi-agent system [Ephrati and Rosenschein, 1991], spam detection [Dwork *et al.*, 2001], bioinformatics, graph drawing [Jackson *et al.*, 2008], database [Fagin *et al.*, 2003] and so on.

In a score aggregation system, agents' criteria may be different and the score profile of each candidates may also differ under different criteria. Score aggregation methods try to make a tradeoff among these different attitudes. In score aggregation problems, usually we investigate methods and mechanisms to get reasonable scores from agents and then take the simple sum (or average) value as the aggregate score. There are several methods developing rules for scoring candidates based on preference rankings [Brams and Fishburn, 2002] [Caragiannis *et al.*, 2017]. In a positional scoring rule, a scoring vector $\boldsymbol{s} = (s_1, s_2, \cdots, s_m)$ of m candidates is given, and a candidate receives a score of s_k from an agent if it is ranked at position k in the agent's preference ranking. The well-known Borda method [Borda, 1781] [Arrow *et al.*, 2010] uses the positional scoring rule by letting the scoring vector be $s = (m-1, m-2, \cdots, 0)$. There are also many other models, such as plurality rule [Feddersen and Wright, 1990], k-approval rule [Brams and Fishburn, 2007], veto [Spitzer and Fisher, 1988] and so on. The main difference among them boils down to the way the agent gives the score. For some cases where the scores are given, people may study methods to modify the scores to make them of the same commensurability by using normalization or weighted methods. The weighted sum-up method is to set a weight to each agent and use the weighted sum scores as the final result. The weights of the agents should reflect the attitudes or differences of the agents. All above investigations are focusing on how to give or adjust the scores to the same measure. After these, the second step is to simply sum up (or average) the adjusted scores.

Actually, the second step is the real operation of 'aggregation'. However, except the simple sum-up (average) method, we are not aware of any other score aggregation methods in social choices. In this paper, we consider the real 'aggregation' operation and introduce a nontrivial mechanism for score aggregations. In our models, we always assume that the scores (data) from agents are of the same measure, or we have done something (such as normalizations) to make them of the same measure, or there is no way to know whether or not the scores from different agents mean the same thing. Under this assumption, we may also be able to regard the model as a 'noise model': there is a ground truth scoring of the candidates, and each agent's score is a noisy estimate of the ground truth. In real life, there are many factors contribute to the noise, such as the bias from each agent, different criteria of each agent, the fluctuation of the candidate's performance, and so on. The Central Limit Theorem says that the sum of many independent random variables will tend towards the Gaussian distribution. If all the effects causing noises are independent, we can believe that the sum of them approxi-

*The work was supported by the National Natural Science Foundation of China (61370071), and the Fundamental Research Funds for the Central Universities (ZYGX2015J057).

mately obeys the Gaussian distribution by the Central Limit Theorem. In the noise models, it is nature to employ a score aggregation method that finds a result most likely to coincide with the ground truth. The simple sum-up or average method may not always return an optimal result. This also motivates the study of this paper.

In social choices, the noise models have been studied in rank aggregations, especially for rules under pairwise comparisons [Procaccia *et al.*, 2012][Shah *et al.*, 2015][Xia *et al.*, 2010][Caragiannis *et al.*, 2014][Conitzer *et al.*, 2009][Mao *et al.*, 2013]. For example, in a noise model for the rank aggregation problem, there is a correct ranking of the candidates and every voter has a noisy perception of this correct ranking. Roughly speaking, each voter ranks each pair of alternatives in the correct order with probability $p > 1/2$, and in the wrong order with probability $1 - p$. In this model, the Kemeny rule can be interpreted as a maximum likelihood estimator of the correct ranking. Many other rules have been also studied in the literature.

In this paper, we focus on score aggregations and try to find an aggregate result most likely to coincide with the ground truth. To do this, we define the 'noise information' as the total squared distances between the original score and the corresponding aggregate score for each candidate, and use a spectral method to find a solution that minimizes the noise information. Note that when the noises obey the Gaussian distribution, minimizing the noise information is equal to maximizing the likelihood function. Our spectral method first conducts an eigenvalue decomposition to the judgment matrix of the score data and then uses the eigenvector corresponding to the largest eigenvalue as the synthesized preferences of the agents. The linear expression of the score matrix on this eigenvector will be used as the final scores of the candidates. Furthermore, it is easy to evaluate the validity of the aggregate scoring or the consistency of the agents. In order to describe it in a clear way, we will give both algebraic and geometric explanations for score aggregation and begin with the simple sum-up method.

2 The Problem

We consider settings with a set of n agents A and a set of m candidates C. Each agent gives a score within the range of the scoring standard to each candidate according to the performance of the candidate and the preference of the agent. A score is a real nonnegative number. Our goal is to get a reasonable consensus scoring over all the candidates as close to the score profile given by the agents as possible.

The score profile can be presented by a matrix, named the *score matrix*, which is formally defined below.

Definition 1. (Score Matrix) *A score matrix of n agents on m candidates is an $m \times n$ matrix*

$$X_{m\times n} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix},$$

where x_{ij} is the score of candidate i obtained from agent j.

The score matrix is also written as follows:

$$S_{m\times n} = \begin{bmatrix} \boldsymbol{c}_1 \\ \boldsymbol{c}_2 \\ \vdots \\ \boldsymbol{c}_m \end{bmatrix} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_n],$$

where the row vector $\boldsymbol{c}_i = (x_{i1}, x_{i2}, \cdots, x_{in})$ is the scores of candidate i obtained from all the agents and the column vector $\boldsymbol{a}_j^T = (x_{1j}, x_{2j}, \cdots, x_{mj})$ is the score of all candidates given by agent j.

The score aggregation problem is to find an $m \times 1$ vector $\boldsymbol{v}$ to represent the score matrix $X_{m\times n}$, where the ith element in $\boldsymbol{v}$ is the final score of the ith candidate. What kinds of vectors $\boldsymbol{v}$ are good? There is no formal definition. Different problem models give different optimal objects and yield different optimal algorithms. We aim at constructing a reasonable problem model with a clear objective and design an algorithm that can find an optimal solution under our model.

Firstly, we view the score aggregation problem from a geometric perspective. We consider the vector space $\mathcal{A}$ with the basis vectors being $\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_n$, representing the n agents. Each candidate is presented by a point $\boldsymbol{c}_i = (x_{i1}, x_{i2}, \cdots, x_{in})$ in the vector space such that the scalar x_{ij} on axis $\boldsymbol{a}_j$ is the score of candidate i given by agent j. Figure 1 shows a two-dimensional vector space with 2 agents $\boldsymbol{a}_1$ and $\boldsymbol{a}_2$ and 5 candidates $\boldsymbol{c}_i$. The score matrix of the candidates is

$$X^T = \begin{bmatrix} 1 & 3 & 3 & 6 & 6 \\ 3 & 2 & 4 & 4 & 2 \end{bmatrix}.$$

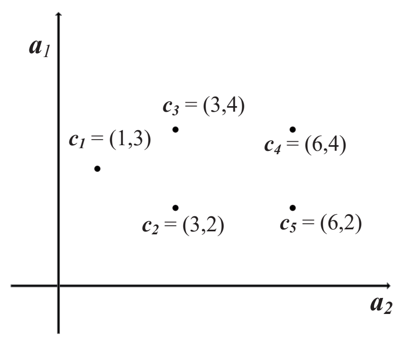

Figure 1: A geometric model

The score aggregation problem is to map the m points (candidates) in the n-dimensional vector space $\mathcal{A}$ into m points in a one-dimensional vector space $\boldsymbol{e}$, called the *object vector space*. There are two things we need to do: the first is to decide which one-dimensional vector $\boldsymbol{e}$ to be mapped onto (we will only be interested in the direction of the vector and consider the unit vector $\boldsymbol{e}$ so that $\boldsymbol{e}\boldsymbol{e}^T = 1$ by ignoring the magnitude of it); and the second is to decide the rules of how to map a point in $\mathcal{A}$ into a point in $\boldsymbol{e}$. Next, we show how the two steps work on the sum-up and average methods.

3 The Sum-up and Average Methods

The sum-up and average methods are simple and widely used in practice. The sum-up method is to use the sum of all scores of a candidate as the final score profile of the candidate, and it returns the vector

$$\boldsymbol{a}_{sum} = (\sum_{i=1}^{n} x_{1i}, \sum_{i=1}^{n} x_{2i}, \cdots, \sum_{i=1}^{n} x_{mi})^T.$$

The average method, a special case of the weighted sum-up method, is to use average score of a candidate as its final score profile, and it returns the vector

$$\boldsymbol{a}_{avg} = (\frac{1}{n}\sum_{i=1}^{n} x_{1i}, \frac{1}{n}\sum_{i=1}^{n} x_{2i}, \cdots, \frac{1}{n}\sum_{i=1}^{n} x_{mi})^T.$$

The results of the two methods can be transferred to each other by dividing or multiplying by a factor n. So these two methods are essentially the same.

Without loss of generality, we look at the average method. The output of this method is obtained by multiplying a vector $\boldsymbol{e} = (1/n, 1/n, \cdots, 1/n)^T$ to the score matrix $X_{m\times n}$, i.e.,

$$\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix} \times \begin{bmatrix} 1/n \\ 1/n \\ \vdots \\ 1/n \end{bmatrix} = \begin{pmatrix} \frac{\sum_{i=1}^{n} x_{1i}}{n} \\ \frac{\sum_{i=1}^{n} x_{2i}}{n} \\ \vdots \\ \frac{\sum_{i=1}^{n} x_{mi}}{n} \end{pmatrix}.$$

The geometric explanation of the above formula is to project the m points $\boldsymbol{c}_i = (x_{i1}, x_{i2}, \cdots, x_{in})$ for $i = 1, \cdots, m$ in the vector space $\mathcal{A}$ onto the object vector space $\boldsymbol{e}^T = (1/n, 1/n, \cdots, 1/n)^T$. For the purpose of presentation, we always consider the vector $\boldsymbol{e}$ to be mapped onto as a unit vector so that $\boldsymbol{e}\boldsymbol{e}^T = 1$. Then we assume that we project the m points onto the unit vector $\boldsymbol{e}' = (1/\sqrt{n}, 1/\sqrt{n}, \cdots, 1/\sqrt{n})$ and will multiple a scalar $||\boldsymbol{e}|| = 1/\sqrt{n}$ in the final result. We use $\hat{\boldsymbol{c}}_i$ to denote the projection point of the original point $\boldsymbol{c}_i$ onto the unit vector $\boldsymbol{e}'$. Figure 2 illustrates the projection in a two-dimensional space with $n = 2$. We will show that the score of each candidate i obtained by the average method is the coordinate of $\hat{\boldsymbol{c}}_i$ in the vector space $\boldsymbol{e}'$ multiplying the scalar $||\boldsymbol{e}|| = 1/\sqrt{n}$.

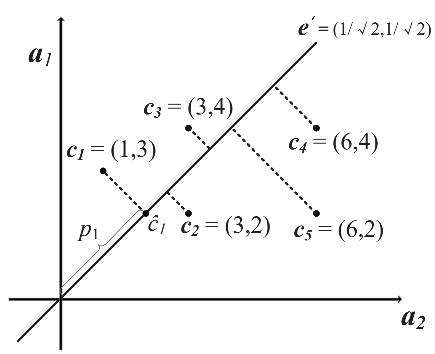

Figure 2: The projection

We prove a general result. Let $\boldsymbol{e}$ be an object vector and $\boldsymbol{a} = X_{m\times n} \cdot \boldsymbol{e}$ be the aggregate score. Let $\boldsymbol{e} = ||e|| \cdot \boldsymbol{e}'$, where $\boldsymbol{e}'$ is a unit vector. Let $\hat{\boldsymbol{c}}_i$ be the projection of $\boldsymbol{c}_i$ on the vector $\boldsymbol{e}$, and p_i be the projection length of $\boldsymbol{c}_i$.

Theorem 1. *Let $\boldsymbol{a}(i)$ denote the ith element in $\boldsymbol{a}$, i.e., the aggregate score of candidate i. It holds that*

$$\boldsymbol{a}(i) = ||\boldsymbol{e}|| \cdot p_i.$$

Proof. We consider candidate i and the point $\boldsymbol{c}_i = (x_{i1}, x_{i2}, \cdots, x_{in})$ in the vector space $\mathcal{A}$. Let θ be the angle between $\boldsymbol{c}_i$ (taking it as a vector) and the vector $\boldsymbol{e}$. It holds that

$$\cos\theta = \frac{\boldsymbol{c}_i \cdot \boldsymbol{e}^T}{||\boldsymbol{c}_i|| \cdot ||\boldsymbol{e}||}.$$

The projection length of the point $\boldsymbol{c}_i$ onto the vector $\boldsymbol{e}$ is

$$\begin{aligned} p_i = ||\hat{\boldsymbol{c}}_i|| &= ||\boldsymbol{c}_i|| \cdot \cos\theta \\ &= \frac{\boldsymbol{c}_i \cdot \boldsymbol{e}^T}{||\boldsymbol{e}||} \\ &= \frac{1}{||\boldsymbol{e}||} \cdot \boldsymbol{a}(i). \end{aligned}$$

The theorem holds. □

For the sum-up method, we multiply a a vector $\boldsymbol{e}'' = (1, 1, \cdots, 1)$ by the score matrix $X_{m\times n}$. The object vector is $\boldsymbol{e}'' = (1, 1, \cdots, 1) = \sqrt{n} \cdot \boldsymbol{e}'$, where $\boldsymbol{e}'$ is the same unit vector as that in the average method. So the score of each candidate i obtained by the sum-up method is the projection length of the point $\boldsymbol{c}_i$ multiplying the scalar $||\boldsymbol{e}''|| = \sqrt{n}$.

The sum-up and average methods are essentially the same if we ignore the stretching factor. They select the unit vector $\boldsymbol{e}' = (1/\sqrt{n}, 1/\sqrt{n}, \cdots, 1/\sqrt{n})$ (after ignoring the stretching factor) as the object vector to be mapped onto; and then map each original point $\boldsymbol{c}_i$ to a point $\hat{\boldsymbol{c}}_i$ on the object vector such that the total *noise information* E is minimized, which is defined as the total least squared distance between the original data points and the mapped points, i.e.,

$$E = \sum_{i=1}^{m} ||\hat{\boldsymbol{c}}_i - \boldsymbol{c}_i||^2.$$

Since $\hat{\boldsymbol{c}}_i$ is the projection of $\boldsymbol{c}_i$ on $\boldsymbol{e}'$, we know that $\hat{\boldsymbol{c}}_i$ is the point on $\boldsymbol{e}'$ such that $||\hat{\boldsymbol{c}}_i - \boldsymbol{c}_i||$ is minimum. However, we prefer to use the squared distance as the object to be optimized since the squared distance has more good properties and is commonly used in the literature. Furthermore, when the noises obey Gaussian distribution, to minimize the squared distance is equal to maximize the likelihood function.

4 The Spectral Method

The sum-up and average methods fix the object vector $\boldsymbol{e}' = (1/\sqrt{n}, 1/\sqrt{n}, \cdots, 1/\sqrt{n})$ and then map the points onto it to minimize the total least squared distance. The second step is reasonable, since the total least squared distance is a standard measure widely used. However, why always choose $\boldsymbol{e}' = (1/\sqrt{n}, 1/\sqrt{n}, \cdots, 1/\sqrt{n})$ as the object vector to be projected? This method seems to always treat all agents as equally important without clear reasons. In Figure 3, if we select the object vector as $\boldsymbol{e}^* = (2/\sqrt{5}, 1/\sqrt{5})$ instead of $\boldsymbol{e}' = (1/\sqrt{2}, 1/\sqrt{2})$ to be projected, we can get projection points with a smaller noise information E. We can see that the total squared distance by selecting $\boldsymbol{e}^* = (2/\sqrt{5}, 1/\sqrt{5})$ as the object vector is less than that of selecting $\boldsymbol{e}' = (1/\sqrt{2}, 1/\sqrt{2})$. See Figure 3 for an illustration.

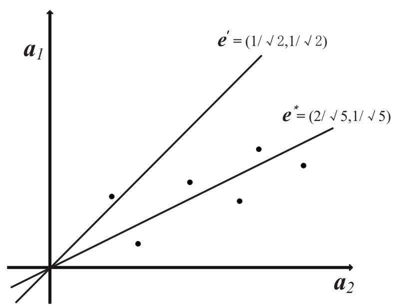

Figure 3: Two projection directions $\boldsymbol{e}'$ and $\boldsymbol{e}^*$

Here arises a question how to find the optimal object vector $\boldsymbol{e}^*$ to minimize the total least square distance? We will answer this question by giving a simple algorithm that calculates the aggregate scores by minimizing the total squared distance. In our algorithm, we do not solve lots of functions to find the optimal object vector and to compute the solution. Instead, we use a spectral method, which is simple to implement and easy to analyze. It is surprising that the spectral method matches all these well.

Our algorithm computes the judgement matrix (defined below) and then uses the eigenvector corresponding to the largest eigenvalue of the judgement matrix as our object vector.

Definition 2. (Judgement Matrix) *Given a score matrix $X_{m\times n}$, a judgement matrix of $X_{m\times n}$ is an $n \times n$ matrix*

$$Z_{n\times n} = (X_{m\times n})^{\mathrm{T}} \times X_{m\times n} = (z_{ij}).$$

The main steps of our algorithm are listed below.

Algorithm SPECTRAL
Input: A score matrix X
Output: A vector corresponding to the aggregate scores of candidates

1. Compute the judgement matrix $Z = X^{\mathrm{T}} \times X$;
2. Conduct eigenvalue decomposition to Z, let λ_{max} be the largest eigenvalue and $\boldsymbol{e}$ be the eigenvector corresponding to it;
3. Return $\boldsymbol{v} = X \times \boldsymbol{e}$.

By Theorem 1, we know that the aggregate score of candidate i by our algorithm is the projection length of the point $\boldsymbol{c}_i$ multiplied by the scalar $||\boldsymbol{e}||$. Next, we prove the following theorem to show the optimality of our algorithm.

Theorem 2. *The eigenvector $\mathbf{e}$ corresponding to the largest eigenvalue of the judgement matrix Z is a vector to be projected onto such that the total squared distance E reaches the minimum value.*

Proof. Assume $\boldsymbol{e}$ is the optimal object vector that minimizes the total squared distance E. Let $\hat{\boldsymbol{c}}_i = p_i\boldsymbol{e}$ denote the mapped points on $\boldsymbol{e}$, where p_i is its coordinate. The total squared distance between the original points and the projection points is

$$\begin{aligned} E &= \sum_{i=1}^{m} ||\hat{\boldsymbol{c}}_i - \boldsymbol{c}_i||^2 \\ &= \sum_{i=1}^{m} ||p_i\boldsymbol{e} - \boldsymbol{c}_i||^2 \\ &= \sum_{i=1}^{m} (p_i^2 - 2p_i\boldsymbol{c}_i\boldsymbol{e}^T + \boldsymbol{c}_i\boldsymbol{c}_i^T). \end{aligned} \tag{1}$$

Consider the minimization with respect to p_i, setting the partial derivative of E with respect to p_i to zero, we obtain

$$p_i = \boldsymbol{c}_i\boldsymbol{e}^T. \tag{2}$$

The above equivalent implies that the scalar p_i is the projection onto the direction $\boldsymbol{e}$. If we substitute for p_i in (1) by using (2), we obtain

$$E = -\sum_{i=1}^{m} \boldsymbol{e}\boldsymbol{c}_i^T\boldsymbol{c}_i\boldsymbol{e}^T + \sum_{i=1}^{m} ||\boldsymbol{c}_i||^2. \tag{3}$$

Note that $\sum_{i=1}^{m} ||\boldsymbol{c}_i||^2$ is a constant and

$$\sum_{i=1}^{m} \boldsymbol{c}_i^T\boldsymbol{c}_i = Z$$

is exactly the judgement matrix. To minimize E is then to maximize

$$J = \boldsymbol{e}Z\boldsymbol{e}^T.$$

We will add a constraint $\boldsymbol{e}\boldsymbol{e}^T = 1$ to maximize J, which requires the vector $\boldsymbol{e}$ being an unit vector and then constrains the solution.

We use a Lagrange multiplier λ to enforce the constraint, and then make an unconstrained maximization of

$$J = \boldsymbol{e}Z\boldsymbol{e}^T + \lambda(\boldsymbol{e}\boldsymbol{e}^T - 1).$$

Setting the derivation with respect to $\boldsymbol{e}$ to zero, we see that this quantity will have a stationary point when

$$Z\boldsymbol{e}^T = \lambda\boldsymbol{e}^T,$$

which says that $\boldsymbol{e}^T$ must be an eigenvector of Z. To minimize E or maximize J, the eigenvector that corresponds to the maximum eigenvalue should be chosen. □

5 Consistency Analysis

The spectral method can always output an aggregate scoring to candidates. However, how much credit we can put on to this scoring or how consistent this scoring is with the agents? We want to use a measure to evaluate the unconsistency of the input data, which can also be recognized as the noise. The total squared distance E defined above is a nice indicator. However, there exists no absolute scale for the total squared noises, but rather all these are quantified relati to the original scores. These is a famous indicator, called the *signal-to-noise ratio* (SNR), frequently used in industry and real life, which is defined as the follows

$$SNR = \frac{extracted\ information}{error}.$$

A high SNR ($\gg 1$) indicates a good extraction of the data, while a low SNR indicates a poor measurement. In our model, the extracted information is the total squared projection distance and the error is the total squared distance between the original data points and the projection data points. Thus, in our problem,

$$SNR = \frac{\sum_{i=1}^{m} p_i^2}{E},$$

where p_i is the projection distance of point $\boldsymbol{c}_i$. We suggest to use SNR as the *consistency indicator*. For some data, our SNR is not large, which means that the scores given by the agents are not consistent. For this case, if the score aggregation is used to rank the candidates, then we know that the input data is not enough to get any reliable order of the candidates due to the very different attitudes of the agents. How to set the threshold of SNR is worthy of deep studying, which should differ for different types of data and applications.

Another advantage of adopting SNR is that there is a simple way to compute the value of SNR in our method. It is enough to calculate the eigenvalues of the judgement matrix.

Theorem 3. *Let $\lambda_1, \lambda_2, \cdots, \lambda_n$ denote the eigenvalues of the judgement matrix Z, where $\lambda_i \geq \lambda_j$ for $i > j$. The consistency indicator of the scoring obtained by the spectral method is*

$$SNR = \frac{\lambda_1}{\sum_{i=2}^{n} \lambda_i}.$$

Proof. Let $\boldsymbol{e}$ denote the eigenvector corresponding to the largest eigenvalue of the judgement matrix Z. Note that $\boldsymbol{c}_i\boldsymbol{e}^T = \boldsymbol{e}\boldsymbol{c}_i^T = p_i$ is the projection distance of point $\boldsymbol{c}_i$ on the direction $\boldsymbol{e}$. We have that

$$\boldsymbol{e}Z\boldsymbol{e}^T = \sum_{i=1}^{m} \boldsymbol{e}\boldsymbol{c}_i^T\boldsymbol{c}_i\boldsymbol{e}^T = \sum_{i=1}^{m} p_i^2.$$

On the other hand, since $\boldsymbol{e}$ is the eigenvector corresponding to the largest eigenvalue of Z, it holds that

$$\boldsymbol{e}Z\boldsymbol{e}^T = \lambda_{max} = \lambda_1.$$

Thus, we have

$$\sum_{i=1}^{m} p_i^2 = \lambda_1.$$

Next, we only need to prove that

$$E = \sum_{i=2}^{n} \lambda_i.$$

Note that for a judgement matrix Z, it holds that

$$\sum_{i=1}^{n} \lambda_i = tr(Z) = \sum_{i=1}^{n} z_{ii},$$

where $\sum_{i=1}^{n} z_{ii} = \sum_{i=1}^{n}\sum_{j=1}^{m} x_{ji}^2 = \sum_{j=1}^{m}\sum_{i=1}^{n} x_{ji}^2 = \sum_{i=1}^{m} ||\boldsymbol{c}_i||^2$.

Since $\hat{\boldsymbol{c}}_i$ is the projection of $\boldsymbol{c}_i$ onto $\boldsymbol{e}$, it holds that

$$||\boldsymbol{c}_i - \hat{\boldsymbol{c}}_i||^2 = ||\boldsymbol{c}_i||^2 - ||\hat{\boldsymbol{c}}_i||^2 = ||\boldsymbol{c}_i||^2 - p_i^2.$$

Then,

$$\begin{aligned} E = \sum_{i=1}^{m} ||\boldsymbol{c}_i - \hat{\boldsymbol{c}}_i||^2 &= \sum_{i=1}^{m} ||\boldsymbol{c}_i||^2 - \sum_{i=1}^{m} p_i^2 \\ &= \sum_{i=1}^{n} \lambda_i - \sum_{i=1}^{m} p_i^2 \\ &= \sum_{i=1}^{n} \lambda_i - \lambda_1 = \sum_{i=2}^{n} \lambda_i. \end{aligned}$$

The theorem holds. □

6 Relations with The Weighted Sum-up Method

The weighted sum-up method is to set a weight w_j to agent j and use the weighted score $\sum_{j=1}^{n} w_j x_{ij}$ as the final score for candidate i. Therefore, the weighted sum-up method outputs

$$\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix} \times \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = \begin{pmatrix} \sum_{i=1}^{n} w_i x_{1i} \\ \sum_{i=1}^{n} w_i x_{2i} \\ \vdots \\ \sum_{i=1}^{n} w_i x_{mi} \end{pmatrix}.$$

It is to project the points onto the object vector $(w_1, w_2, \cdots, w_n)$. For the weighted sum-up method, the crucial part is to set the weights w_i to agents. Traditionally, we set the weights according to experiences or some other given information. The purpose is to make the agents of the same commensurability.

Our spectral method, however, assumes that all the agents are almost of the same commensurability, uses the eigenvector $\boldsymbol{e}$ corresponding to the largest eigenvalue of the judgement matrix Z as the object vector, and sets the weights of agents as the eigenvector $\boldsymbol{e}$. Our method can be regarded as a self-adapted step to set the weights of agents to minimize the noise information. Our method can combine with the traditional weighted sum-up method well: we first set a weight to agents artificially to make the agents of almost the same commensurability and then apply our spectral method on the adjusted scores to get the second weight. The second weight can also be used to adjust the first weight.

7 Computational Examples and Applications

Here we give computational results on some real word data as examples. The first example is about scoring movies. Our data is from GroupLens Research (https://grouplens.org/datasets/movielens/), which collects the rating data for movies from the web site MovieLens (http://movielens.org) [Harper and Konstan, 2016]. The database we used is *ml-latest-small*, which describes 5-star rating for movies by users from movieLens. Ratings are made on a 5-star scale, with half-star increments (0.5 stars - 5.0 stars). Users are represented by an ID without other information. Here we give a simple example with a small number of movies and users. We randomly choose six famous movies and six users who had seen more than 500 movies (including all the six movies). The movies are *Forrest Gump (1994)*, *Star Wars: Episode IV - A New Hope (1977)*, *Toy Story (1995)*, *Dead Men Don't Wear Plaid (1982)*, *The Fugitive, (1993)*, *Seven (1995)*, and the userID are 15, 23, 30, 73, 212, 213. We obtain the following score matrix

$$X = \begin{bmatrix} 2 & 3 & 4 & 5 & 3 & 3 \\ 5 & 4.5 & 4 & 5 & 3.5 & 2.5 \\ 1 & 4.5 & 5 & 5 & 4 & 2 \\ 3 & 3.5 & 5 & 4 & 5 & 2.5 \\ 5 & 3.5 & 5 & 4 & 3 & 4 \\ 5 & 4.5 & 4 & 4.5 & 4 & 5 \end{bmatrix}.$$

We run our algorithm on this score matrix, the final scores of these six movies are $\boldsymbol{v_1} = \{3.40, 4.12, 3.71, 3.88, 4.07, 4.44\}$[1], while the final scores of the average method is $\boldsymbol{v_2} = \{3.33, 4.08, 3.58, 3.83, 4.08, 4.50\}$. The signal-to-noise ratio SNR of our result is 21.7, which is greater than 18.8 for the average method. This is a computational example on a small number of samples to show the computational steps of our algorithm and compare the results. Next, we give an example on a big date set.

The next example is about ranking universities according to the publications. Our data is from the web site of CWTS Leiden Ranking (http://www.leidenranking.com/). The CWTS Leiden Ranking 2016 offers key insights into the scientific performance over 800 major universities worldwide. A sophisticated set of bibliometric indicators provide statistics on the scientific impact of universities and on universities involvement in scientific collaboration. We focus on the indicator called 'top 10% P', which is the number of a university's publications belong to the top 10% most frequently cited in the year. CWTS Leiden Ranking 2016 collected the data from the following five main fields of science: Biomedical and health sciences (Bio), Life and earth sciences (Life), Mathematics and computer science (M&C), Physical sciences and engineering (P&E), and Social sciences and humanities (Soc). The default way is to use the sum of the top 10% cited publications in all the five fields as the indicator of the university, i.e., the sum-up method. We use our method to obtain a result. The first 15 universities under our method are listed in Table 1.

In this table, the column 'ours' is the results of our method, comparing the column 'total' of the results of the sum-up method. In these two columns, the number in the bracket is the final score obtained by the corresponding method and the number out of the bracket is the position ranked by using these scores. The next five columns denote the numbers of top 10% cited publications of these universities on the five main fields.

From the table we can see that some universities with a smaller number of total top 10% cited publications are ranked higher in our method, for examples, Johns Hopkins Univ. and UC-San Francisco. These universities have a large number of top 10% cited publications in some fields say Bio. and a small top cited publication number in other fields. Our method rank these universities higher because the major fields of these universities are strong, although they may not be strong in every field. We think this is reasonable and even can be regarded as an advantage of our method. We should encourage universities (agents) to develop some strong fields, instead of all fields with a middle level.

[1] In fact, the scores were divided by $\sqrt{n} = \sqrt{6}$ to make them of the same stretching factor as the average method.

Table 1: The top 15 universities of our method.

Univ	Ours	Total	Bio	Life	M&C	P&E	Soc
Harvard Univ	1(4949)	1 (7060)	4947	467	136	834	676
Univ Toronto	2(1926)	3 (2956)	1853	201	150	462	289
Johns Hop Univ	3(1917)	7 (2539)	1976	108	51	261	144
Stanford Univ	4(1744)	2 (3223)	1429	300	215	936	342
UC-SF	5(1641)	17(1990)	1775	75	10	46	85
Univ Michigan	6(1602)	4 (2756)	1417	188	169	627	355
Univ Penn	7(1493)	11(2253)	1456	106	80	309	301
UC-Los Angeles	8(1391)	8 (2424)	1231	225	157	516	296
Univ Coll Lon.	9(1381)	13(2162)	1328	118	91	296	329
UW-Seattle	10(1374)	10(2321)	1267	399	89	340	226
Columbia Univ	11(1290)	14(2129)	1217	181	124	286	321
Univ Oxford	12(1286)	9 (2414)	1084	306	130	527	367
UC-SD	13(1270)	15(2126)	1164	253	153	390	166
Yale Univ	14(1199)	16(2001)	1117	239	60	274	312
MIT	15(1184)	6 (2606)	748	254	246	1164	194

We also note that MIT is not ranked very high in our method, although it has the largest numbers of top 10% cited publications in two fields M&C and P&E. The reason is that the total numbers of top 10% cited publications in these two fields are not large. If we normalize all fields to reduce the differences among them, the situation will be better. Now we apply our method directly without any normalization, which indicates that we assume all fields are of the same commensurability. Under this assumption, small total numbers of top 10% cited publications in the two fields M&C and P&E will indicate small influences of these two fields. It is reasonable.

8 Conclusion

This paper suggests a new method for score aggregation, which uses spectral analysis. Our method assumes that all the agents are of the same commensurability and finds an 'optimal' aggregate scoring minimizing the 'noise' part. The new method may open up a new dimension along which scoring rules could be defined and evaluated. It also gives rise to many interesting problems in social choice for further study.

For the method itself, there are also some important issues worthy of further study. For example, this paper only considers projecting scores on vectors which begin with the origin. Could we get better results if we consider other vectors as well? It is possible. However, we may get strange results for some extreme cases. Here is an example of two candidates with scores $(1, 3)$ and $(3, 1)$ from two agents, the global optimal vector of which will pass through these two points. For this case, it is hard to explain the meaning of the projection. It will also be interesting to study more properties and types of vectors to be projected on.

References

[Arrow *et al.*, 2010] Kenneth J. Arrow, Amartya Sen, and Kotaro Suzumura. *Handbook of social choice and welfare*, volume 2. Elsevier, 2010.

[Borda, 1781] J. Borda. *Memoire sur les Elections au Scrutin*. Histoire de l'Academie Royale des Sciences, 1781.

[Brams and Fishburn, 2002] Steven J. Brams and Peter C. Fishburn. Voting procedures. *Handbook of social choice and welfare*, 1:173–236, 2002.

[Brams and Fishburn, 2007] Steven Brams and Peter C. Fishburn. *Approval voting*. Springer Science & Business Media, 2007.

[Caragiannis *et al.*, 2014] Ioannis Caragiannis, Ariel D. Procaccia, and Nisarg Shah. Modal ranking: A uniquely robust voting rule. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence (AAAI 2014)*, pages 616–622, 2014.

[Caragiannis *et al.*, 2017] Ioannis Caragiannis, Xenophon Chatzigeorgiou, George A. Krimpas, and Alexandros A. Voudouris. Optimizing positional scoring rules for rank aggregation. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI 2017)*, pages 430–436, 2017.

[Conitzer *et al.*, 2009] Vincent Conitzer, Matthew Rognlie, and Lirong Xia. Preference functions that score rankings and maximum likelihood estimation. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI 2009)*, pages 109–115, 2009.

[Dwork *et al.*, 2001] Cynthia Dwork, Ravi Kumar, Moni Naor, and Dandapani Sivakumar. Rank aggregation methods for the web. In *Proceedings of the 10th international conference on World Wide Web (WWW 2001)*, pages 613–622. ACM, 2001.

[Ephrati and Rosenschein, 1991] Eithan Ephrati and Jeffrey S. Rosenschein. The clarke tax as a consensus mechanism among automated agents. In *Proceedings of the 9th National Conference on Artificial Intelligence (AAAI 1991)*, pages 173–178, 1991.

[Fagin *et al.*, 2003] Ronald Fagin, Ravi Kumar, and Dandapani Sivakumar. Efficient similarity search and classification via rank aggregation. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD 2003)*, pages 301–312, 2003.

[Feddersen and Wright, 1990] Timothy J. Feddersen and Stephen G. Wright. Rational voting and candidate entry under plurality rule. *American Journal of Political Science*, 34(4):1005–1016, 1990.

[Harper and Konstan, 2016] F. Maxwell Harper and Joseph A. Konstan. The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems*, 5(4):19, 2016.

[Jackson *et al.*, 2008] Benjamin N. Jackson, Patrick S. Schnable, and Srinivas Aluru. Consensus genetic maps as median orders from inconsistent sources. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 5(2):161–171, 2008.

[Mao *et al.*, 2013] Andrew Mao, Ariel D. Procaccia, and Yiling Chen. Better human computation through principled voting. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence (AAAI 2013)*, pages 1142–1148, 2013.

[Moulin *et al.*, 2016] Hervé Moulin, Felix Brandt, Vincent Conitzer, Ulle Endriss, Ariel D. Procaccia, and Jérôme Lang. *Handbook of Computational Social Choice*. Cambridge University Press, 2016.

[Procaccia *et al.*, 2012] Ariel D. Procaccia, Sashank J. Reddi, and Nisarg Shah. A maximum likelihood approach for selecting sets of alternatives. *arXiv preprint arXiv:1210.4882*, 2012.

[Shah *et al.*, 2015] Nisarg Shah, Ariel D. Procaccia, and Yair Zick. Voting rules as error correcting codes. *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI 2015)*, pages 1000–1006, 2015.

[Spitzer and Fisher, 1988] Robert J. Spitzer and Louis Fisher. The presidential veto : touchstone of the american presidency. *American Political Science Review*, 1988.

[Xia *et al.*, 2010] Lirong Xia, Vincent Conitzer, and Jérôme Lang. Aggregating preferences in multi-issue domains by using maximum likelihood estimators. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, pages 399–408, 2010.