Social Choice Theory -- Voting Methods in Grant Panel Review Setting

Mentee: Mia (Minghe) Zhang, Terry Yuan
Mentor: Michael Pearce, Elena Erosheva, Conor Mayo-Wilson

Under the circumstance where certain number of individuals want to reach to an agreement or to make a decision, a way for them to express their opinions and to aggregate their preferences is obviously needed, and voting is a popular means. The core of social choice theory is about the analysis of preference aggregation—to make individuals' preferences into a single and collective one over the alternatives.

Throughout the quarter, we are reading papers and doing research in the realm of social choice theory and various voting. Specifically, in our project, we would like to focus on aggregating individuals' preferences under the panel review setting. Considering the possibility of missing data in the panel review situation due to its specialty, we've selected a few voting methods that could work with some modifications and to implement modified algorithms for the selected voting methods in R. They are the score method, the Condorcet method, and the majority judgement.

Here are the modifications that are the results that we've reached throughout the discussions for the quarter.

1. Modified Condorcet method, with adjustment for NAs

We've developed a modified Condorcet method considering the addressed problems: inconsistent ballots and missing data. The traditional Condorcet method would make comparisons between every two candidates in the election. In each comparison, we would count the number of times where one candidate's rank is higher than the other across all the valid votes from voters, and the candidate with a higher count would win the competition. If a candidate wins one comparison, its total count of winning the comparison would be added with 1. Among all the candidates, the one with the highest total count, meaning the one who wins the highest number of comparisons, would be the winner. The candidate winning the second highest number of competitions would be the second winner, and so forth.

Compared with the traditional Condorcet method, this implementation would only consider the pairwise comparisons between two valid scores, meaning that there is no missing data in the comparison. It first gives rank based on the score, with tie-breaking method to be "minimum". For example, if we have a set of scores of {1.1, 1.1, 1.2, 1.2, 1.2, 1.3} , where 1.1 is the best score, we may assign the rank as {1, 1, 3, 3, 3, 6}. Then it will calculate the times of pairwise comparisons between two candidates where one's rank is higher than the other. The one who wins more pairwise comparisons would receive a score of 1, while the loser would receive a 0, indicating its failure.

The corresponding scores would be recorded in a Condorcet comparison matrix with both the number of columns and the number of rows being equal to the number of candidates. An entry (i, j) means that comparison between the ith candidate and the jth candidate. For example, if (2, 3) has a score of 1, it means that the 2^{nd} candidate wins over the 3^{rd} candidate. The (3, 2) entry would be 0 correspondingly, indicating that the 3^{rd} candidate lose the comparison with the 2^{nd} candidate.

The final winner would be calculated based on the sum of the scores of each row in the Condorcet comparison matrix, where each row represents each candidate and the numbers in every entry on the same row represents whether this candidate wins another. The one with the highest sum of scores will win.

The modified Condorcet matrix can also tell us whether there is a Condorcet winner or a Condorcet loser. It will calculate whether there is a candidate who wins every pairwise comparison and whether there is a candidate who loses every comparison. The method will provide a complete ordering of the candidates from the best to the worst. For example, if the total sums of scores for each row are: {c1 = 6, c2 = 3, c3 = 4, c4 = 5, c5 = 1}, where each number represents the number of times the candidate wins over the other in pairwise comparisons, then the method will calculate the complete ordering from the best to the worst as: {c1, c4, c3, c2, c5}. If user wants to select three winners, or to know the candidates with the first three top ranking, he/she could look at the complete ordering to select the top three.

There are, however, several issues concerning the ties, that need to be solved. Suppose we have already identified the n-1 best proposals and would like to select n winners in total. There is only one remaining spot but we have a tie for the next-ranking candidates—for example, two candidates are both in the next-ranking. How could we make a decision? A possible solution might be that we look back at the original scores of the two candidates and that we compare the average scores each candidate receives. Another possible solution might be that we look back at the rank of the two candidates ranking the same, and compare the ranks between the two.

Another potential problem arises when considering the extreme case where, for example, voter 1 gives out 1.5 for 1st, 2nd, and 3rd candidates. Voter 2 gives out 1.5 for the 1st candidate and does not give any score for the 2nd or the 3rd candidate. In this case, how could solve the ties?

Overall, this adjustment has considered the case of having missing data or inconsistent ballets, and we do not need to make any assumptions about the missing data in order to perform the Condorcet method.

2. Modified Condorcet method, proportion version, with adjustment for NAs

In the proportion version, the method will also calculate the times of pairwise comparisons between two candidates where one's rank is higher than the other, but with a little variation as explained below.

It will calculate the proportion of the number of times where one's rank is over the other in the comparison and will record the proportion of the two candidates in the comparison respectively in the Condorcet comparison matrix, instead of assigning 1 or 0 to the two candidates. The corresponding number would be recorded in the Condorcet comparison matrix with both the number of columns and the number of rows being equal to the number of candidates. An entry (i, j) records that proportion of the number of times where the ith candidate's rank is over the jth candidate. For example, if (2, 3) has a proportion of 0.6, it means that the 60% of 2nd candidate's ranks are over the 3rd candidate's. The (3, 2) entry would be 0.4 correspondingly, indicating that 40% of 3rd candidate's ranks are over the 2nd candidate's.

The final winner would be calculated based on the sum of the scores of each row, where each row represents each candidate and the numbers on every entry on the same row represents the proportion of the number of times where the candidate's rank is higher than the other in pairwise comparisons. The one with the highest score will win.

The modified Condorcet method with proportion version also tells us whether there is a Condorcet winner or a Condorcet loser and it will provide a complete ordering of the candidates from the best to the worst too.

The potential problem also arises here when considering the extreme case where, for example, voter 1 gives out 1.5 for 1^{st}, 2^{nd}, and 3^{rd} candidates. Voter 2 gives out 1.5 for the 1^{st} candidate and does not give any score for the 2^{nd} or the 3^{rd} candidate. In this case, how could solve the ties?

Meanwhile, regarding the tie-breaking of total sum of proportions, we talked about the necessity of examining such problem. Meanwhile, we came up with a new question: how can we check whether there is a cycle in this new Condorcet method and how can we check the sufficiency and other properties? This would be the future implementation to work on. We would also try to give answers to the questions in the future discussions: whether the modified method is a good method and how good it is

Overall, the proportion version may be preferred over the first version in the way such that we can hardly encounter ties with proportions with decimals, compared to the integers. Another reason is that since we are not considering the comparison with any missing data, the total number of times we compare the ranks of two candidates in comparisons might differ, and we can fix such problem with the idea of using proportions.

3. Modified score method

In this modification, we would calculate the average scores of all the valid votes for one candidate and compare the average scores. The one with the smallest score wins. It will provide a complete ordering of the candidates from the best to the worst.

There are, however, potential problems in the extreme case related to missing data. For example, if a proposal receives only one perfect rating, should we treat it as better than another proposal with many pretty good ratings but just slightly lower than the one perfect rating? Moreover, if we have the same average scores, how do we break the ties? A possible solution to it would be that we apply the Majority Judgement method to the candidates with tie to select one.

4. Majority Judgement method

The first thing we have to do is to sort the scores of each candidate in the descending order in the original data set and get to know how many seats there are for candidates or how many winners we need to choose. Then, we will repeat finding the top one candidate each time, put it into the elected candidate list and remove it from the candidate list until the remaining seat is zero. Within the repeating process, we will first find the median of each candidate by counting only those scores that are not missing data (valid scores). If the number of valid scores is odd, then choose the middle one as the median. Otherwise, choose the one below the middle as the median*. Then, sort the medians of candidates and get the lowest score(s). If there is only one candidate that has that score, the candidate will be elected and added to the elected list. If there is more than one candidate that has the same lowest median, the method will pick out all the data for these candidates and continue the competition. In the second-round competition, we will remove the value at the median place of every candidate in the competition each time (eliminating process), calculate the new median of the new data of each candidate and compare the current median until only one candidate stands out and gets the best score. If the remaining number scores of any candidate after eliminating is 1 without getting the best elected one (i.e. everyone's medians are still the same), we will pick up the first candidate as the elected one in alphabetical order**. After getting the winner in each round, the data of the elected candidate will be removed from the candidate list to prevent selecting it again in the next round.

However, the Majority Judgement method, like the other methods, is not the perfect one for a "absolutely" fair result. The most serious problem is how to deal with the (any kinds of and any degrees of) tie. For example, when. there are two candidates, one has three 1.5-scores and 4 missing data, the other have four1.5-scores, one 1.2-score, one 4-score and 1 missing data, we cannot fairly choose one winner within them by using Majority Judgement (Table 1). More strictly, if one has three 1.5-scores and 4 missing data, the other have seven 1.5-scores, we cannot choose one winner at all by MJ (Table 2).

	Voter1	Voter2	Voter3	Voter4	Voter5	Voter6	Voter7
Candidate1	1.5	1.5	1.5	NA	NA	NA	NA
Candidate2	1.2	1.5	1.5	1.5	1.5	4	NA

Table 1: Who is the winner?

	Voter1	Voter2	Voter3	Voter4	Voter5	Voter6	Voter7
Candidate1	1.5	1.5	1.5	NA	NA	NA	NA
Candidate2	1.5	1.5	1.5	1.5	1.5	1.5	1.5

Table 2: Who is the winner?

Also, there is another question. When a new voter joins the voting process and think that all candidates are at the same level (give the same score to all the candidates or give the same rank to all candidates), the winner picked by MJ algorithm should not change because for the new voter, it has no preference. However, the winner does change by using MJ algorithm (Table 3 and Table 4).

	Voter1	Voter2	Voter3	Voter4	Voter5	Voter6
Candidate1	1	1.5	2	3.5	4	3
Candidate2	1.8	1.9	2.1	2.7	3.9	3

Table 3: New Data With Adding A Voter With No Preference

Candidate1	1	1.5	2	3	3.5	4
Candidate2	1.8	1.9	2.1	2.7	3	3.9

Table 4: Sorted New Data

We can see from the two tables above that when the new Voter6 does not join the voting process, our winner between candidate1 and candidate2 is candidate1 because its median is 2 which is smaller than the median of another candidate which is 2.1. When Voter6 who has the same preference for two candidates (or no preference between two candidates) joins, the winner will be candidate2 because its median is 2.7 now which is smaller than the median of another one which is 3 (When using the lower one of the two values in the middle; We can easily find another example if using the upper one of the two values in the middle; If using the mean of the two values in the middle as the median, the winner will also change).

* Here is a question that why we choose the lower one in the two values in the middle. It seems that sometimes choosing the lower one will be better and sometimes upper will be better. The reason we choose the lower one is more than half of voters will have better scores or rankings than the one at median place. But note that we cannot use the mean of the two values in the middle as the median, just like the way we calculate median in mathematical problems, because we have to remove the median if there exist a tie.

** There might be a better way to select a winner in that situation instead of choosing in alphabetical order. 1. Calculating the average of remaining scores of each candidate and the one with the best average is the winner; 2. The one with the most valid scores in the remaining data wins; 3. Resample the data for each candidate with the sample size same as the candidate with the smallest number of valid scores and do the second-round competition.

5. Result and Conclusion

	1st winner	2nd winner	3rd winner	4th winner	5th winner
Score	17	19	25	21	4
Condorcet	17	25	19	21	4
MJ	17	19	25	21	4

It is surprising to see from the form that most of the results yielded by three different methods are the same despite there is a big difference between the algorithm of each method. For the score method and Majority Judgement, the reason they have the same results is that our data set does not contain outliers so that the mean and median agree with each other. For the Condorcet method, the second winner and third winner are slightly different from the other two methods. But it does matter a lot if we only choose two winners which shows small changes could be meaningful for the whole decision-making process.