Monte Carlo Simulation for Elections - Voting Systems

Gaurav Dharra, Prof. John Weible
Programming for Analytics and Data Processing, School of Library and Information Sciences, University of Illinois at Urbana-Champaign

INTRODUCTION

Various voting systems are being studied to determine an effective system for selecting the most eligible and suitable candidate. The different systems include plurality, runoff, Borda, Condorcet, and score voting. The below description provides a brief understanding of the various systems stated:

- **Plurality**: The simplest system wherein each person casts one vote for their favorite candidate and the candidate with the most votes wins.
- **Runoff**: It is similar to plurality except that, after the first round of voting, the candidate with the least votes is eliminated and the election is repeated with the remaining candidates.
- **Borda**: In this system, voters don't vote for a single candidate, rather they rank the candidates and each candidate receives points respective to their rank.
- **Condocert**: It runs pairwise popular elections between every possible pair of candidates. If a single candidate beats all others in these pairwise elections, that candidate is the winner.
- **Score Voting**: A system in which each voter gives each candidate a score on some scale and the candidate with the highest average score wins. One noteworthy difference of this system is that it does not require any ranking of the candidates since a voter can give multiple candidates the same score.

AIM

This project aims to simulate these voting systems to determine which of these electoral system
- works the best in picking the most suitable candidate and
- is least prone to strategic manipulation.

The hypothesis that I would like to come up with based on the definitions of each of the voting system is –

- The Condorcet system of voting may work the best in selecting a suitable candidate since every candidate is ranked with every other pair of candidates and the most favorable candidate gets to win.
- The Score Voting is vulnerable to strategic manipulation since the score is very subjective and quantitative.

METHOD

Monte Carlo Simulation is applied for simulation of voting for elections

- **Candidates** – Three candidates are chosen A, B and C that will be contesting for the election. Each candidate is assigned a random fame score that signifies popularity.
- **Voters** – A random set of 10000 voters are generated each having a unique voter id and a randomly generated preferential score. This randomly generated preferential score is compared with each candidate fame score to generate preference value for each candidate.
- **Expected Winner** – This is computed purely based on summation of the preferential scores of all the voters.
- **Generation of votes** –
    - **Honest Voters:** Votes are generated based on the preferential score provided by the candidate
    - **Strategic Manipulation in voting:** The concept of strategic manipulation is introduced by changing votes of random voters to a famous candidate instead of their preferred candidate.
- A method exist per voting system to generate the winning candidate
- The next step involves simulating the whole process 100 times to
    - compute probability of each voting system generating the expected winner
    - Identify effect of strategic manipulation on each voting system
- **Parallel Processing** was used to simulate the results using a multiprocessing module of python library to make an async call to the run_simulation() method.
- Based on the number of times a voting system produced an accurate result, the **probability of generating the expected results was computed** and is tabulated as shown in the results section.

RESULTS

Election results for various voting systems are as shown below:

Voting System	Probability of accurate results (without strategy)	Probability of accurate results with strategic manipulation
Plurality	0.42	0.42
Runoff	0.7	0.7
Borda	0.0	
Condocert	0.58	0.42
Score Voting	1.0	0.27

SNAPSHOT OF INPUTS

	voter_id	preferential_score_candidate_A	preferential_score_candidate_B	preferential_score_candidate_C
0	0	0.433692	0.894958	0.768422
1	1	0.732417	0.596233	0.932653
2	2	0.266194	0.947544	0.600924
3	3	0.885332	0.433318	0.779938
4	4	0.422586	0.896064	0.757316
5	5	0.409996	0.908654	0.744726
6	6	0.666737	0.651913	0.998533
7	7	0.189525	0.870875	0.524255
8	8	0.268450	0.949800	0.603180
9	9	0.185476	0.866827	0.520206
10	10	0.451713	0.866937	0.786443
11	11	0.399428	0.919222	0.734158

List of voters generated with preferential scores.

List of voters generated with ranking for each candidate

	rank_a	rank_b	rank_c
0	3	1	2
1	1	3	2
2	1	3	2
3	3	1	2
4	1	3	2
5	3	1	2
6	1	3	2
7	1	3	2
8	1	3	2
9	1	3	2
10	3	2	1
11	3	1	2
12	3	1	2

Expected Winner

```
Expected Winner =  C
```

CONCLUSIONS

The following are the three important conclusions:
- On repeatedly executing the simulation, the results show that Score voting gives the most accurate results.
- As seen in the table above, the score voting simulation results are tweaked by strategic manipulation.
- Borda results have been consistently inconsistent with the expected results.

FUTURE WORK

- The voters and candidates can be generated by simulating the statistics from a real election data.
- Additional ways of strategically manipulating votes can be implemented to study their impact on the results.

ILLINOIS