Tactical Voting Analyst Lab

Group 11

Sai Koneru, Zhaolin Li, Chinmay Rao, Tom Scholer

ABSTRACT

A Tactical Voting Analyst (TVA) is a software program that, given a voting situation, analyzes all possible ways each voter can manipulate the voting outcome to their benefit by faking their preferences. This work deals with implementing a TVA program and analyzing various voting situations using it. Four different voting schemes are compared based on the strategic voting risk associated with them. Additionally, we study the effect of strategic voting on global happiness and based on the assumption that using tactics always decreases global happiness, we ponder over the question "*Is there a possibility of an increase in global happiness after voting tactically*?" Furthermore, a number of more complex scenarios such as voter collusion, counter-strategic voting and presence of imperfect information are briefly discussed with regards to their complexity. Finally, a new voter happiness formula is proposed that can reduce the risk of strategic voting in a certain restricted set of cases.

KEYWORDS

Multi-agent System; Tactical Voting; Voting scheme; Voter Happiness

1 INTRODUCTION

In elections, tactical voting occurs when a voter supports an alternative more strongly other than the sincere preference in order to achieve a desired voting outcome. The application of tactical voting brings the risk of influencing the result through deliberate manipulation, which is undesired for the honest election. In this assignment, we developed and implemented a software to analyze the risk of tactical voting taking place given a preference matrix and a voting scheme.

This assignment aims for one-round voting. Possible voting schemes are Plurality voting, Voting for two, Anti-plurality voting and Borda voting. The voting vector for each scheme is in Table 1.

Voting scheme	Voting vector
Plurality voting	1, 0, ..., 0
Voting for two	1, 1, , 0, ..., 0
Anti-plurality voting	1, 1, ..., 1, 0
Borda voting	m-1, m-2, ..., 1, 0

Table 1: Voting vector of each voting scheme

Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (eds.), May 2020, Auckland, New Zealand

https://doi.org/doi

Three tactical voting methods are selected: compromising, burying and bullet voting. Compromising is ranking an alternative insincerely higher. Burying means ranking an alternative insincerely lower. Bullet voting is voting for just one alternative, despite having the option to vote for more than one. Push-over method is not select because it is for run-off voting.

This assignment has three simplifications in experiments. All experiments only use single-voter manipulation, voter collusion is not considered. Experiments doesn't consider the counter-strategic voting, which means performing the tactical voting in response to an earlier strategic vote. Besides, Tactical voting analyst knows true preferences of all voters.

In this assignment, we investigated whether there is voting scheme that always has a better risk than others, the relationship between the risk and the voting preference matrix, and the relation between the global happiness and tactical voting. Besides, we performed an analysis on the difficulty and complexity of extending the simplifications.

2 MATERIAL AND METHODS

2.1 Implementation and code structure

For implementing the TVA, we choose Python 3 as our platform since it provides high-level data manipulation and numerical computation libraries like Pandas and Numpy. The *TVA.py* script contains all the following functions:

(1) *TVA()*: This is the main function that implements the TVA at a high level. Given a matrix of true preferences and a voting scheme, it outputs the non-strategic outcome *O*, the non-strategic overall happiness *H*, a dictionary of strategic options *S* and the risk of strategic voting *R*.
(2) *getOutcome*(): This is a Scheme selector function that channels the parameters given to it towards one of the Scheme functions - *getPluralOutcome()*, *getVotingForTwoOutcome()*, *getAntiPluralOutcome()*, *getBordaOutocme()*. This function mainly takes the preference matrix as the input and returns the outcome *O* based on the user's scheme of choice.
(3) *getHappiness()*: Given an outcome list and the true preference matrix, this function computes the overall happiness *H*. For each voter, the function *getVoterHappiness()* is used for calculating individual happiness value.
(4) *applyBestTactic()*: Given a voter i and the true preference matrix, this function applies the best tactic for the voter. Three tactics are considered here - compromise, bury and bullet. The lower level functions *_comproTactic()*, *_buryTactic()* and *_bulletTactic()* implement the corresponding tactics by brute-forcing over all possibilities and choosing the best one.

Figure 1 illustrates the relationship between each of the above mentioned functions. The arrows represent function calls. Note that although the compromise and the bury tactics are to be considered equivalent in this case, the code implements them separately merely for the sake of clarity, though at a cost of some redundancy.

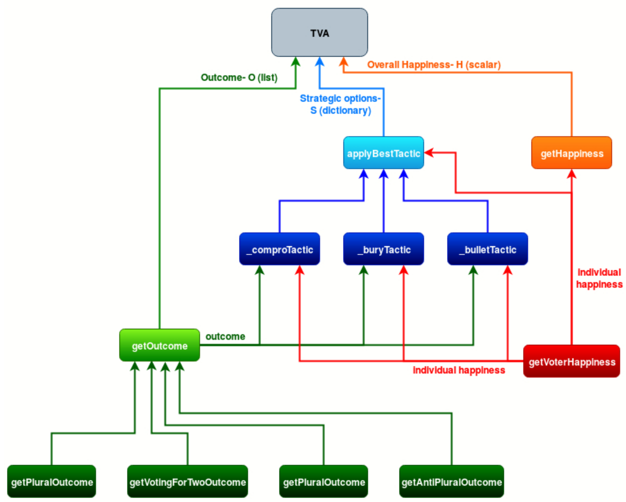

Figure 1: Relationship between functions

The pseudo-code for TVA is given as follows:

```
def TVA(pref_matrix):
    scheme = get_scheme_from_user
    O = getOutcome(pref_matrix)
    H = getHappiness(O, pref_matrix)
    S={}
    for each voter i:
        s_i = applyBestTactic(i, pref_matrix)
        If voter i's happiness can increase, then
            S[i] = s_i
    R = len(S)/n

    print(O,H,S,R)
```

2.2 Implementation complexity

In this subsection, we discuss the complexity of tactical voting based on our implementation. Consider a general case of n voters and m alternatives per voter.

2.2.1 Computing the outcome: For computing the Plurality and Voting-for-Two outcomes, each voter has to be visited once making the complexity $O(n)$. In case of Anti-plurality, each alternative (except the last) of each voter has to be visited which makes its complexity $O(n(m-1))$. And in the case of Borda scheme, it is $O(nm)$.

2.2.2 Computing the happiness: While computing the individual happiness, all the alternatives of a voter are once visited. The complexity is thus $O(m)$. While calculating the Overall Happiness H individual happiness for each voter is computed which makes its time complexity $O(nm)$.

2.2.3 Applying tactics: The *applyTactics()* function is invoked for all n voters. For each voter, bullet tactic and all possibilities of compromise and bury are attempted. The number of possibilities for compromise and bury are $m(m-1)$ each. Moreover, while testing each possibility, individual voter happiness and outcome are computed. Assuming the Borda scheme, the total time complexity of the function *applyTactics()* is in the order of $O(nm^3)$.

2.3 Voting preference matrix and scheme

Plurality voting is a popular scheme and widely used in election. while it has a shortcoming that it ignores most of a voter's preference order. Besides plurality voting, this assignment tested other three voting schemes: Voting-for-two (select first two alternatives with count 1), Anti-plurality Voting (select all alternatives except the last with count 1) and Borda voting (select all alternatives with descending counts).

These four schemes produce different outcome and therefore have different risks. In the next section, we try to show that whether there is no voting scheme that is always better than others. Table 2 shows the six designed voting example with different numbers of preference and voter. Each example was randomly initialized.

Data example	Structure (preference*voters)
Data 1	3 * 4
Data 2	10 * 10
Data 3	4 * 5
Data 4	3 * 10
Data 5	10 * 3
Data 6	10 * 6

Table 2: Different preference matrices

2.4 Global happiness vs Tactical Voting

Global happiness is given by the sum of happiness for each individual voter. Tactical voting is carried out by a voter if his strategic preference increases his happiness with respect to his true preferences. In this assignment, we investigated the trend in global happiness in relation to tactical voting for each voting scheme. If global happiness decreases with respect to tactical voting we can say that it is not good in consideration to true preference of all voters. If it increases we need to start analysing in depth to see if it is always the case or in some circumstances.

3 RESULTS

3.1 Voting preference matrix and scheme

To analyse the relationship between voting scheme and preference matrix, we designed six different preference matrices (see Table 2), and calculated the risk of each scheme.

As shown in Table 3, there is no voting scheme that is always better than others. Borda voting scheme always has a high risk.

Besides the structure of preference matrix, we think the initialization of preference matrix may have influence of the risk as well. We change the preference of the first, fifth and sixth voter in data 6,

Data example	Plurality	Vote-for-two	Anti-plurality	Borda
Data 1	0	0,50	0,50	0,50
Data 2	0,40	0,50	0,50	0,90
Data 3	0,80	0	0	0,60
Data 4	0,30	0	0	0,20
Data 5	0,33	1	1	1
Data 6	0,33	0,50	0,67	1

Table 3: Risks for different preference matrix and schemes

which in more likely to vote tactically, by swapping the preferences of A and B, and swapping the preferences of D and I. The alternatives A and B, and D and I, are selected randomly. As shown in Table 3 and Table 4, different initialization influences the possible tactical voters and risks of each scheme. For example, after changing the initial preference of voter 5, the risk changes from 0.33, 0.5, 0.67 and 1 to 0.33, 0.67, 0.83, and 1 respectively. The possible of tactical voter changes with the initialization as well.

Data example	Plurality	Vote for two	Anti-plurality	Borda
Data 6	1, 6	1, 5, 6	1, 2, 3, 5	1, 2, 3, 4, 5, 6
Data 6-v1	1, 3	1, 5, 6	1, 2, 3, 6	1, 2, 3, 4, 5, 6
Data 6-v5	1, 6	1, 6	1, 2, 3, 5	1, 2, 3, 4, 5, 6
Data 6-v6	3, 6	1, 3, 5, 6	1, 2, 3, 5, 6	1, 2, 3, 4, 5, 6

Table 4: Possible tactical voters for different preference matrix and schemes

3.2 Global happiness vs Tactical voting

	1^{st} voter	2^{nd} voter	3^{rd} voter	4^{th} voter	5^{th} voter
1^{st} preference	C	B	C	B	B
2^{nd} preference	A	D	D	D	A
3^{rd} preference	D	C	A	C	D
4^{th} preference	B	A	B	A	C

Table 5: Global Happiness Increase Scenario

Let us consider a case where the global happiness increases with respect to tactical voting. Consider the following scenario where we have 5 voters and each of them have 4 preferences as shown in Table 5.

True Preference	Outcome	Strategic Preference	Strategic Outcome
B	B	B	B
D	C	D	D
C	D	A	C
A	A	C	A

Table 6: Effect on Global Happiness After Tactical Voting

Let us assume that we are using Borda voting scheme. Second voter can tactically vote here to improve his happiness as shown in Table 6. If he changes his preference and swaps his third and fourth preference insincerely he can make the voting outcome exactly as his true preference list. Initially before any tactical voting true outcome will be B, C, D, A and global happiness according to this outcome is 1.47. After changing his preferences to B, D, A, C from B, D, C, A the outcome will change to B, D, C, A and global happiness will increase to 2.47.

We can construct other examples where global happiness decreases from tactical voting. If the global happiness of the voters increases this may indicate that we need a better voting scheme. But, for any voting scheme we can easily construct an example where the global happiness increases after tactical voting . No matter what the scheme is we can easily construct examples where global happiness increases or decreases and every scheme is flawed for some specific examples.

We can conclude that there is no voting scheme that is not susceptible to tactical voting and also that decreases global happiness after tactical voting.

4 EXTENSIONS

4.1 Voter collusion

In voter collusion, two or more voters choose together which candidates to vote for in order to bury candidates that pose a threat to their true preference and compromise candidates that have little chance to pose a threat to their true preference. With perfect information a group of voters can impact the outcome very easily. Also the more voters collude together the better they can adjust the votes to get their desired outcome. Voters want to collude together if they have the same or similar true preference of candidates. For example for a set of 10 candidates, voters would want to collude in plurality voting if they have the same top preference. However for other voting schemes they might only want to collude if they have the same top 3 preferences.

To extend our TVA to be able to take collusion into account would be a complex issue since we would have to determine who would want to collude with whom. If we have a number of n voters, we would have to check for every voter a collusion with a variable set of every other voter. This variable set could be a set of 1 to n-1 voters. Then for all the different possible collusions we would have to determine what every voter votes to increase the happiness of the collective and also increase its own happiness since different voters might not have the exact same preferences. After this TVA would also have to determine how many groups of collusion should be formed and what voters would be in what group.

Collusion becomes very complex very quickly and depending on the rules can be very hard to implement. It would be a fairly easy implementation if only one collusion of two voters should be considered for a set of 10 voters. However, with a increasing set of voters and candidates, machine learning would have to be applied to categorize voters into collusion groups.

4.2 Counter-strategic voting

Counter-strategic voting is normally applied when voters suspect other voters of strategic-voting. For example if some voters suspect other voters of compromising a candidate so that their personal happiness increases, the other voters could then counter this by compromising other candidates so that their personal happiness increases. With perfect information counter-strategic voting can be successfully applied and would have the same effect on the voter's happiness as if plain strategic-voting is used to compromise an election where everybody votes his true preference.

To extend our TVA to be able to apply counter-strategic voting would not be difficult or complex. In our TVA we calculate the strategic options for every voter that can increase its happiness by applying a strategic-vote. We could then consider every option as a new voting input and apply strategic voting on it again. This would check and compute the options for every voter that could apply a strategic vote to increase its happiness. This would basically correspond to a second round of strategic-voting.

4.3 Both voter collusion & counter-strategic voting

Implementing both voter collusion and counter-strategic voting would be as hard as just implementing voter collusion (as discussed in 4.1) since counter-strategic voting is fairly incomplex. After having determined a strategic vote for a collusion another strategic vote would be applied to determine the outcome of a counter-strategic collusion vote.

4.4 Imperfect information

If we are considering voters not to have perfect information on what every voter is going to vote, it becomes very difficult to form collusions. If none of the preferences of every voter are visible, forming collusions is impossible since nobody would know who to collude with. If for example only the first preference of every voter is known, the voters with the same first preference could form a collusion. However, there are a couple of factors that come into play depending on : the number of winners of an election (only first wins or two first win) and the voting scheme that is applied. For example if only the first preference is known but the two first candidates of an outcome win, some voters might not want to collude with other voters even though they have the same first preference but a different second preference.

Counter-strategic voting is a very risky process if voters do not have perfect information on what all the voters are actually going to vote for. The satisfaction of a voter that applies counter-strategic voting without having perfect information can lead to either an increase or decrease, the outcome is not predictable.

5 DISCUSSION

5.1 Voting preference matrix and scheme

From experiments, we conclude that there is no voting scheme that is always better than others. Borda voting scheme always produce higher risk, and it makes sense because Borda takes whole preference order into account and is more likely to be influenced by performing tactical voting. Besides, we conclude that the initialization of the preference matrix influences the risks and possible tactical voters of all voting schemes except the Borda voting. As our initialization method is simple and random, and has not very large preference matrix, it is insufficient to conclude a accurate relationship between the matrix initialization and each voting scheme.

5.2 Happiness with respect to winner vs outcome

Happiness with respect to outcome has been used in this project. This formula with respect to outcome is useful when the whole outcome of the voting scheme is important in the situation. For example, voters are rating movies in a order from top to bottom. Then, we are concerned with the whole outcome and tactical voting should be done if he can increase the happiness with respect to outcome.

Lets consider cases where we use plurality and we only care about the winner of the election. For example, where we are electing a politician we are much more concerned with who wins the election than who comes second or third. In this situation if we use happiness with respect to outcome to analyze the preference matrix we would not get the outcome we desire. We need to calculate the happiness of individual voter with respect to winner than the whole outcome. Hence we propose a formula that is calculated with respect to the winner.

$$H_i = 1/(1 + d_i) \tag{1}$$

This equation1 gives us the happiness of each voter where d_i is the distance from the winner in the outcome to where the winner is in your true preference list. If a voter's first preference is the winner then $d_i = 0$ and happiness of that voter will be equal to 1. Now the global happiness of the voter is calculated using sum of happiness for each individual voter.

$$GlobalHappiness = \sum_{n=1}^{N} H_i \tag{2}$$

With this formula we can prevent a voter from voting tactically when its first preference has already won. In some cases such as presidential elections, if the first preference has won there is no need to tactically vote but there may be possibilities of counter strategic voting. However, in cases such as movie ranking, order of the outcome is of significance and so in this case, the proposed formula cannot completely capture the results. Therefore, depending on the situation we are analyzing we need to define the happiness function accordingly.

Implementation of happiness with respect to winner in schemes like plurality and anti-plurality can be found in the software uploaded. Computation is much easier as we are not worried on the outcome. In plurality a voter can only tactically vote if second winner in the outcome has a difference of maximum of one assuming that there is no voter collusion. This will reduce the computation then checking by brute force.

6 CONCLUSION

Tactical voting analyst can be used to calculate risk in different voting schemes with happiness calculated by the outcome. Complexity when extending to different situations such as voter collusion are discussed but not implemented. Risk for every voting scheme with different data is shown and can be inferred that no voting scheme is the best. Analysis on global happiness that is if it increases or decreases after tactical voting is discussed. Finally a different formula for happiness is shown which can be useful depending on the situation. Future work would be to analyze the situation when extended to different situations and also the effect when happiness formula is used with respect to the outcome.