(67686) Mathematical Foundations of AI June 25, 2008

Lecture 7

Lecturer: Ariel D. Procaccia *Scribe: Amit Goldstein, Roy Schwartz and Yoav Wilf*

1 Introduction and Reminders

1.1 The Voting Model

- Let $N = \{1 \dots n\}$ a set of voters and $A = \{a, b, c, \dots\}$ set of candidates such that $|A| = m$.
- For each voter $i \in N$ there exists $\prec_i$: i's *Preference over A*: a ranking of all the various candidates of the voter i.
- Let $L(A)$ the set of all possible preferences of A (by one voter).
- Let $\prec^N = (\prec_1, \dots \prec_n) \in L^n$ the *preference profile*: the preference of every voter in N.
- A **Voting Rule** is a function $L^n \to A$, selecting the winning candidate according to the preference profile of the voters.

1.2 Definitions and Theorems

1. A voting rule f is **strategy-proof** if:

$$\forall \prec \in L^n, i \in N, \prec_i^{'} \in L : f(\prec) \succeq_i f(\prec_i^{'}, \prec_{-i}).$$

   given a preference profile, a single player cannot benefit from lying about his preference (providing a different preference $\prec_i^{'}$ that does not reflect his original preference $\prec_i$)

2. A voting rule f is **dictatorial** if there exists $i \in N$ such that i's most preferred candidate always wins.
3. **The Gibbard-Satterthwaite theorem**: let f be a strategy-proof voting rule **onto** A (every candidate in A might win), if $m > 2$ then f is **dictatorial**.

1.3 Introduction

We've seen that for more than two candidates, every non dictatorial voting rule is manipulable. Because this theorem dictates our reality, our goal is to make the manipulation harder to perform (instead of trying to prevent it). In order to create a voting rule that will be as fair as possible, we'll try to create a voting rule that has voters that might benefit from lying, but will be required to perform tough computations in order to find the desired lie. By doing this we will bypass the problem and find a fair voting rule.

2 Lecture Body

2.1 Definition of the computational problem

MANIPULATION$_\mathbf{f}$: given a set of candidates A, a group of voters N, a voting rule f, a manipulator $i \in N$, a preference profile of all voters except $i \prec_{-i}^{N} = (\prec_1, \dots \prec_{i-1}, \prec_{i+1}, \dots \prec_n)$, and a specific candidate $p \in A$: Answer whether the manipulator can provide a preference such that p will be chosen by f.

Notice the following observations:

- The problem is a decision problem. We would like to know if there exists such a preference.
- The problem is defined more strongly than the simple definition of manipulation. The question is not whether the manipulator can guarantee a victory of a more preferred candidate, but whether he can guarantee the victory of a specific candidate.
- The Gibbard-Satterthwaite theorem specifies that one of the voters can manipulate the results. We don't know if this is true under the given circumstances, where i and p were given.

2.2 A greedy algorithm deciding the problem under specific voting rules

This algorithm generates the preference of the voter i such that the candidate p will be chosen, or returns that this is not possible. The algorithm does the following:

1. Rank p in 1^{st} place.
2. While there are candidates that were not ranked yet:
   - If there exists a candidate that can be ranked in the next spot without preventing p from winning, rank that candidate in the next spot.
   - otherwise - declare that the desired preference does not exist.

Example

In this example m=4, n=3, and the **Borda** voting rule (where every voter gives m-1 points to his top candidate, m-2 to the next, etc.) will be used. The preference of voters 1 and 2 is a, p, b, c and the result of the algorithm for the 3^{rd} voter is:

1. Rank p in 1^{st} place, and find that with the two other voters p has 7 points.
2. Notice that a can't be ranked next, since he'll end with 8 points. Rank b in 2^{nd} place, leaving him with 4 points.
3. Notice that a can't be ranked next, since he'll end with 7 points. Rank c in 3rd place, leaving him with 1 point.
4. Rank a in 4^{th} place, leaving him with 6 points.

According to f, p wins with 7 points (while a, b, and c have 6, 4, and 1 points respectively).

2.3 Generalization

Theorem 1 *Let f be a voting rule such that there exists a function $s : L \times A \rightarrow R$ denoted $s(\prec, a)$ (representing a function that given a preference of the manipulator and a candidate, returns the score of the candidate), and suppose that:*

1. *For every $\prec \in L$, f chooses $\arg\max_{x \in A}(s(\prec, x))$, the candidate maximizing $s(\prec, x)$.*
2. *If $\{x \in A : x \prec a\} \subseteq \{x \in A : x \prec^{'} a\}$, meaning if the set of candidates that are below a in the preference is contained in the set of candidates that are below a in the preference $\prec^{'}$, then $s(\prec, a) \leq s(\prec^{'}, a)$*

Then the greedy algorithm always decides MANIPULATION$_f$ correctly.

Proof It's obvious that if the algorithm found a preference resulting in p winning, the decision that the manipulator can provide a preference such that p wins is correct. Let's examine the case where the algorithm could not find such a preference.

Preparations

Suppose that the algorithm stopped ranking when the subset $U \subseteq A$ were not yet ranked, and assume that there exists a preference $\prec^{'}$ for the manipulator that causes p to win. Denote $u \in U$ the top candidate of the subset U according to $\prec^{'}$. Complete the partial preference created by the algorithm to a preference $\prec$ that u is ranked in the 1^{st} place that was not already ranked by the algorithm.

Example

Let $A = \{a, b, c, d, p\}$, suppose that the algorithm generated the partial preference (p, c, d) and assume that $\prec^{'} = (b, c, d, p, a)$. Then $U = \{a, b\}$ and $u = b$.

Body

Notice the following:

- $\prec$ ranks p 1^{st}, and thus surely $\{x \in A : x \prec^{'} p\} \subseteq \{x \in A : x \prec p\}$. $\Rightarrow$ according to condition 2, $s(\prec, p) \geq s(\prec^{'}, p)$.
- Furthermore, p wins according to $\prec^{'}$, and thus, by condition 1, $\forall x : s(\prec^{'}, p) > s(\prec^{'}, x)$. $\Rightarrow$ specifically, $s(\prec^{'}, p) > s(\prec^{'}, u)$.
- Finally, in $\prec$, u is ranked only above all the other candidates in U (and no other candidates). In $\prec^{'}$, u is also ranked above all the other candidates in U (and maybe some other candidates as well). $\Rightarrow$ by condition 2, $s(\prec^{'}, u) \geq s(\prec, u)$.

Combine the three deductions to:

$$s(\prec, p) \geq s(\prec^{'}, p) > s(\prec^{'}, u) \geq s(\prec, u)$$

$$\Rightarrow s(\prec, p) > s(\prec, u)$$

but when running the algorithm we were not able to rank any candidate in the next spot in $\prec$, because every preference resulted in a winner other than p. This is a contradiction to the conclusion above, and therefore our assumption was false! ■

$\Rightarrow$ if s can be computed in polynomial time, then the algorithm will run in polynomial time, and $MANIPULATION_f \in P$.

Conclusion

Manipulation in polynomial time is possible for the following voting rules:

- Plurality: where every voter gives one point to his top candidate.
- Borda: where every voter gives $m - k$ points to the kth candidate in his preference.
- Copeland: where the score of each candidate is the number of candidates that he beats in pairwise elections.
- Maximin: where if $p(a, b) = |\{i \in N : b \prec_i a\}|$ then the score for candidate a is $\min_{x \in A} p(a, x)$.

2.4 A voting rule where the manipulation is NP-Hard

Definition: Plurality with Preround (PwP)

A voting rule which consists of two rounds:

1. The candidates are divided into pairs. The loser of pairwise elections in each pair is removed.
2. After we are left with half of the original candidates, perform **Plurality**. Meaning, every voter gives one point to his top candidate who survived the 1^{st} round.

Theorem 2 $\mathbf{MANIPULATION_{PwP}} \in NP - Complete$

Proof Obviously, $\mathbf{MANIPULATION_{PwP}} \in NP$, since given $\prec'$, we can decide in polynomial time if p was elected or not. We'll prove that is it NP-Hard using a reduction from the SAT problem[1].

Preparations

Let $\varphi = (x_1 \vee \neg x_2) \wedge (x_2 \vee \neg x_7 \vee x_8) \wedge \ldots$ be a CNF formula.

- Denote $V = \{v : v \text{ variable in the formula}\}$ a set of variables.
- Denote $L = \{v, \neg v : v \in V\}$ set of literals.

[1]See http://en.wikipedia.org/wiki/Boolean_satisfiability_problem for details

- Denote $C = \{c : c \text{ clause in the formula }\}$ set of clauses.
- Define D to be a set of dummy candidates such that $|D| = |C| + 1$.
- Define $A = \{p\} \cup L \cup C \cup D$ the set of candidates.

We'll create an instance of $\mathbf{MANIPULATION_{PwP}}$ in the following way:
Define V with the 3 following separate and covering subsets:

- Subset 1: consists of $4|C| + 2$ voters, such that p is the top candidate in their preference, and the candidates from D are last.
- Subset 2: consists of $4|C|$ voters for every $c \in C$, such that c is the top candidate in their preference, and the candidates from D are last. In total, there are $4|C|^2$ voters in this subset.
- Subset 3: consists of 4 voters for every $c \in C$, such that in their preference the literals that appear in c are ranked 1^{st}, c is ranked after them, and all the candidates from D are last. In total, there are $4|C|$ voters in this subset.

Additionally, the voters will rank the Literals in L such that for every $v \in V$, the literal v is in tie with the literal $\neg v$ (this is possible since we did not define the rankings of the majority of the voters on the literals, and there is an even number of voters).
In the preround, we'll divide the candidates to pairs such that for every $v \in V$, v is paired with $\neg v$, and every candidate in $\{p\} \cup C$ is paired with a candidate in D.

Reduction

All the candidates in $\{p\} \cup C$ survive the 1^{st} round because all the voters rank the candidates in D last (no matter what preference the manipulator provided). Additionally, since every v and his negation are in tie, the manipulator can decide for every $v \in V$ which of the two will proceed to the 2^{nd} round.

- $\Rightarrow$ Suppose that there exists a satisfying assignment to the given CNF formula. If the manipulator's preference is such that all the literals that receive true value in the assignment will proceed to the 2^{nd} round, he'll cause p to be elected: p has at least $4|C| + 2$ points in the 2^{nd} round (the 1^{st} group of voters voted for him). For each $c \in C$, at least one $l \in c$ survived the 1^{st} round (because for every clause, the manipulator passed at least one literal - the one that was satisfied in the clause). Therefore, every voter of the 3^{rd} group has at least one literal ahead of c, leaving c with only the $4|C|$ points he received from the 2^{nd} group of voters.
  Additionally, every literal $l \in L$ receives points only from the 3^{rd} group of voters, leaving him with up to $4|C|$ points (the size of the 3^{rd} group).
  We've examined all the candidates that survived the 1^{st} round, and found that even if the manipulator chooses to vote to a candidate other than p, that candidate will have up to $4|C| + 1$ points, not enough to surpass p and his $4|C| + 2$ points. We can conclude that if there is a satisfying assignment to the given CNF formula, there is a preference by the manipulator that can cause the election of p.

- $\Leftarrow$ Suppose there is no satisfying assignment to the given formula. Because in the 1^{st} round for each variable only one of his literals proceeds to the 2^{nd} round, for every preference of the manipulator (representing an assignment) there exists a clause $c^* \in C$ such that every one of its literals were false, and therefore didn't proceed to the 2^{nd} round.
  Therefore, c^* receives $4|C|$ points from the 2^{nd} group of voters, and 4 more points from the 4 voters in the 3^{rd} group that ranked the literals of c^* 1^{st}, and c^* after them (since these literals didn't proceed to the 2^{nd} round). We conclude that the candidate c^* has at least $4|C| + 4$ points while p can have at most $4|C| + 3$ points (one point from the manipulator and the rest from the 1^{st} group of voters).
  We proved that if there is no satisfying assignment, p will lose no matter the manipulator's preference, as required.

In conclusion, we proved that $\mathbf{MANIPULATION_{PwP}} \in$ NP-Hard ■

2.5 Another voting rule where the manipulations is NP-Hard

Single Transferable Vote

The voting rule consists of $m - 1$ rounds. In each round every voter gives a point to the top candidate in his preference that is still eligible (survived all previous rounds). The candidate that received the fewest points in the round is removed. In the end, we are left with a single candidate, and he is the winner.

Example

We'll use the example shown in previous lectures, reminding that, as we saw last week, in this example under four different voting rules four different candidates were elected (A, B, C and E). Using the STV voting rule the 5^{th} candidate is elected (D). The table below demonstrates this example. Each column represents a single voting order, where the 1^{st} cell is the number of people who voted in that order, and the following is the order of the vote.

33	16	3	8	18	22
A	B	C	C	D	E
B	D	D	E	E	C
C	C	B	B	C	B
D	E	A	D	B	D
E	A	E	A	A	A

1. The results of the 1^{st} round are $A - 33, B - 16, C - 11, D - 18, E - 22 \Rightarrow C$ is removed.
2. The results of the 2^{nd} round are $A - 33, B - 16, D - 21, E - 30 \Rightarrow B$ is removed.
3. The results of the 3^{rd} round are $A - 33, D - 37, E - 30 \Rightarrow E$ is removed.
4. The results of the 4^{th} round are $A - 33, D - 67 \Rightarrow A$ is removed

and the winner according to the STV voting rule is D.

Theorem 3 *(without proof) The Single Transferable Vote voting rule is NP-Hard to manipulate.*

2.6 Note

We found rules that are hard to manipulate. The problem is that in the two rules that were shown (and in many other hard to manipulate rules) the manipulation will be easy to calculate in most cases (since the hardness is with respect to the worst case and not the average case). Apparently, it seems that every reasonable voting rule can usually be manipulated under typical distributions.

References

[1] J. J. Bartholdi, III, C. A. Tovey, and M. A. Trick, 1989. The computational difficulty of manipulating an election. *Social Choice and Welfare*, $6(3) : 227 - 241$.

[2] V. Conitzer , and T. Sandholm, 2003. Universal voting protocol tweaks to make manipulation hard. *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI).*