Voting: Manipulation, Control, and Bribery

Ivan Petrov

Manipulation

Voters declare preferences different from their true ones to get a better outcome.

Example - plurality: Voter 3’s true preferences are c > b > a , ties broken by alphabetical order

1	a	b	c
2	b	c	a
3			

Will voter 3 vote truthfully?

Brandt, F., Conitzer, V., Endriss, U., Lang, J., & Procaccia, A. D. (Eds.). (2016). *Handbook of computational social choice*. Cambridge University Press.

Gibbard-Satterthwaite Theorem

Recall: any voting rule with 3 or more alternatives must be at least one of the following:

1. **Dictatorial** - there exists a voter whose preferred alternative is chosen for every profile
2. **Imposing** - at least one alternative doesn't win under any profile
3. **Manipulable** - not strategyproof

Properties 1 and 2 are unacceptable in almost all voting situations. So, we are stuck with voting rules that are manipulable.

Brandt, F., Conitzer, V., Endriss, U., Lang, J., & Procaccia, A. D. (Eds.). (2016). *Handbook of computational social choice*. Cambridge University Press.

Downsides to Manipulation

- Bad equilibria
- Lack of information
- Disenfranchisement of less knowledgeable voters
- Wasted effort

Brandt, F., Conitzer, V., Endriss, U., Lang, J., & Procaccia, A. D. (Eds.). (2016). *Handbook of computational social choice*. Cambridge University Press.

Revelation Principle

For any mechanism that results in a good equilibrium (in a game-theoretic sense), there exists another mechanism that results in the same outcomes, but in which agents report their preferences directly and they have no incentive to misreport them.

Problem: This assumes agents are computationally unbounded.

Potential solution: Make computation hardness the barrier to manipulation

Brandt, F., Conitzer, V., Endriss, U., Lang, J., & Procaccia, A. D. (Eds.). (2016). *Handbook of computational social choice*. Cambridge University Press.

Single-Peaked Preferences

- Assumes there exists an ordering of the alternatives (like on the political spectrum, for example)
- Voter selects their most-preferred alternative
  - If a is most-preferred, and $a < b < c$ or $c < b < a$, then the voter prefers b to c.
- Use median voter rule
  - Order voters by their most preferred alternatives and choose the median
- Strategyproof and always chooses a Condorcet winner
- **Problem:** cannot force voter preferences to be single-peaked when they are not.

Brandt, F., Conitzer, V., Endriss, U., Lang, J., & Procaccia, A. D. (Eds.). (2016). *Handbook of computational social choice*. Cambridge University Press.

Randomized Rules

- Map profile to a probability distribution
- Example:
  - Voter preferences are $a < b < c$
  - Should the voter prefer b or a 50-50 lottery between a and c?
  - It depends on voter utilities for the alternatives - if they are 3,2, and 0 then choosing b has higher expected utility, but if they are 3,1, and 0 then a 50-50 lottery between a and c gives higher expected utility
- Strategyproof iff for every utility function that is consistent with the voter preferences, the voter maximizes her utility by reporting those true preferences.
- Gibbard also showed that any strategyproof randomized rule is either a dictatorship or not onto.

Brandt, F., Conitzer, V., Endriss, U., Lang, J., & Procaccia, A. D. (Eds.). (2016). *Handbook of computational social choice*. Cambridge University Press.

Computation Hardness as a Barrier to Manipulation

- Manipulation problem: given a profile of votes Π cast by everyone but the manipulator and a preferred alternative a, is there a vote the manipulator can cast to ensure that a wins?
  - If the voting rule is polynomial to execute, then it takes polynomial time to check a witness
- Problem is in NP
  - It has been shown to be in P for a number of rules, like plurality, cup, Copeland, Borda
  - NP-complete for STV, ranked pairs
- Assumptions:
  - Manipulator knows how everyone else voted
  - No one else is manipulating

Brandt, F., Conitzer, V., Endriss, U., Lang, J., & Procaccia, A. D. (Eds.). (2016). *Handbook of computational social choice*. Cambridge University Press.

Computation Hardness as a Barrier to Manipulation

A voting rule satisfies **BTT conditions** if:

1. It can be run in polynomial time
2. For every profile Π and every alternative a, the rule assigns a score $S(\Pi, a)$ to a
3. For every profile Π, the alternative with the maximum score wins
4. Monotonicity condition: for any Π, Π' and any alternative a, if for each voter i we have that $\{b : a >_i b\} \subseteq \{b : a >'_i b\}$, then $S(\Pi, a) \leq S(\Pi', a)$.

The manipulation problem can be solved in polynomial time for any rule satisfying BTT conditions.

Brandt, F., Conitzer, V., Endriss, U., Lang, J., & Procaccia, A. D. (Eds.). (2016). *Handbook of computational social choice*. Cambridge University Press.

Coalitions of Manipulators

- Same problem, but with a set of manipulators
- Allows for more situations to become manipulable
- Drawback: complexity
  - Even coordinating the coalition of manipulators can be NP-hard (Borda, Copeland) with only 2 manipulators

Brandt, F., Conitzer, V., Endriss, U., Lang, J., & Procaccia, A. D. (Eds.). (2016). *Handbook of computational social choice*. Cambridge University Press.

Coalitions of Manipulators

	unweighted votes constructive manipulation		weighted votes constructive				destructive		
# alternatives			2	3	4	$\geqslant 5$	2	3	$\geqslant 4$
# manipulators	1	$\geqslant 2$							
plurality	P	P	P	P	P	P	P	P	P
plurality with runoff	P	P	P	NP-c	NP-c	NP-c	P	NP-c	NP-c
veto	P	P	P	NP-c	NP-c	NP-c	P	P	P
cup	P	P	P	P	P	P	P	P	P
Copeland	P	P	P	P	NP-c	NP-c	P	P	P
Borda	P	NP-c	P	NP-c	NP-c	NP-c	P	P	P
Nanson	NP-c	NP-c	P	P	NP-c	NP-c	P	P	NP-c
Baldwin	NP-c	NP-c	P	NP-c	NP-c	NP-c	P	NP-c	NP-c
Black	P	NP-c	P	NP-c	NP-c	NP-c	P	P	P
STV	NP-c	NP-c	P	NP-c	NP-c	NP-c	P	NP-c	NP-c
maximin	P	NP-c	P	P	NP-c	NP-c	P	P	P
Bucklin	P	P	P	NP-c	NP-c	NP-c	P	P	P
fallback	P	P	P	P	P	P	P	P	P
ranked pairs	NP-c	NP-c	P	P	P	NP-c	P	P	?
Schulze	P	P	P	P	P	P	P	P	P

Brandt, F., Conitzer, V., Endriss, U., Lang, J., & Procaccia, A. D. (Eds.). (2016). *Handbook of computational social choice*. Cambridge University Press.

Weighted Votes

- Complexity encountered even for a small number of alternatives
- Example: manipulate the veto (anti-plurality method) for 3 alternatives
  - After counting non-manipulated votes, *b* and *c* are tied with the fewest vetoes, and manipulators want *a* to win.
  - Manipulators need to divide their votes precisely so that *b* and *c* get (at least) one more veto each than *a*.
  - This reduces to partitioning a set of integers into two subsets of equal weight, which is an NP-complete problem.

Brandt, F., Conitzer, V., Endriss, U., Lang, J., & Procaccia, A. D. (Eds.). (2016). *Handbook of computational social choice*. Cambridge University Press.

Additional Factors

- Tie-breaking
  - Can change the complexity - Copeland with straightforward tiebreaking is polynomial time to manipulate, but NP-hard with second-order Copeland tiebreaking rule
  - Some rules become poly time if you assume manipulators have utility over all alternatives and seek to maximize expected utility
- Incomplete information
  - Manipulators usually don’t know everyone’s exact vote
- Building in hardness
  - Hybrid rules
  - Randomize/hide which rule is used

Brandt, F., Conitzer, V., Endriss, U., Lang, J., & Procaccia, A. D. (Eds.). (2016). *Handbook of computational social choice*. Cambridge University Press.

Is Manipulation Hard In Practice?

- A manipulation algorithm for a voting rule that is NP-hard to manipulate can hope to either:
  - Always succeed, and usually take polynomial time
  - Usually succeed, and always take polynomial time
- Empirical and heuristic algorithms often succeed in practice. Example:
  - Usually manipulators can only make 2 alternatives win. How to find them?
  - Find one possible winner *a'*, and for each other alternative *a*, choose a voting profile for the manipulators where everyone ranks *a* first and *a'* last.
- Approximation methods
  - Optimization problem: minimize the number of manipulators needed to make *a* win
  - Example with Borda rule: put *a* first, order the rest in increasing order of score. Requires at most one more additional manipulator than the optimal solution.

Brandt, F., Conitzer, V., Endriss, U., Lang, J., & Procaccia, A. D. (Eds.). (2016). *Handbook of computational social choice*. Cambridge University Press.

Frequency of Manipulability

- Given an instance of the coalitional manipulation problem, is it easy to tell whether manipulators are likely to succeed based on the relative size of the voter set and manipulators set?
- Suppose the number of manipulators is in $O(n^p)$, where n is the number of voters. What is the probability that a random profile is manipulable?
  - If $p < ½$, the probability goes to 0
  - If $p > ½$, the probability goes to 1

Brandt, F., Conitzer, V., Endriss, U., Lang, J., & Procaccia, A. D. (Eds.). (2016). *Handbook of computational social choice*. Cambridge University Press.

Game-Theoretic Models

- What if we model other voters as strategic agents instead of choosing random preferences?
- **Problem:** lots of equilibria
- Solution: have voters vote sequentially instead of simultaneously.
    - There is a unique alternative that wins in a subgame-perfect Nash equilibrium
- Does this produce a good outcome?
    - “Unambiguously bad” outcomes for most voting rules.
- Can this be efficiently computed?
    - Use dynamic programming corresponding to backwards induction - for each voter, compute what to do for every ‘situation’ they are in
    - Runtime depends on how you define situation
    - Exact complexity not known

Brandt, F., Conitzer, V., Endriss, U., Lang, J., & Procaccia, A. D. (Eds.). (2016). *Handbook of computational social choice*. Cambridge University Press.

Control - Adding Candidates

- Given a set of A qualified candidates and B spoiler candidates, can we choose a subset B’ of B such that |B’| <= k and alternative *p* is the winner?
- Example:

points:	3	2	1	0
voter 1:	a	c	b	d
voter 2:	b	a	c	d
voter 3:	c	d	b	a
voter 4:	d	b	c	a
voter 5:	d	c	b	a
winner:	c (score 9)			

→

points:	4	3	2	1	0
voter 1:	a	c	b	f	d
voter 2:	b	a	f	c	d
voter 3:	c	d	b	a	f
voter 4:	d	b	f	c	a
voter 5:	d	c	b	f	a
winner:	b (score 13)				

Brandt, F., Conitzer, V., Endriss, U., Lang, J., & Procaccia, A. D. (Eds.). (2016). *Handbook of computational social choice*. Cambridge University Press.

Control - Deleting Candidates

- Given a set of A qualified candidates and B spoiler candidates, can we delete at most k candidates to make alternative *p* the winner?
- Example:

points:	5	4	3	2	1	0
voter 1:	a	c	b	f	e	d
voter 2:	b	a	f	c	e	d
voter 3:	c	d	b	a	f	e
voter 4:	e	d	b	f	c	a
voter 5:	e	d	c	b	f	a
winner:	b with score 16					

→

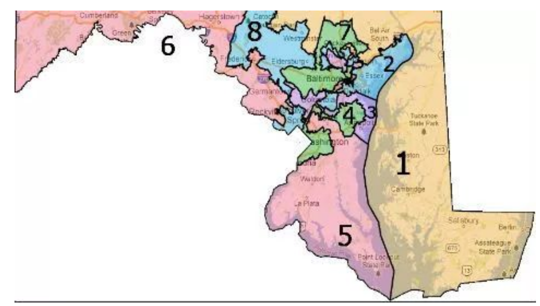
points:	4	3	2	1	0
voter 1:	a	c	b	e	d
voter 2:	b	a	c	e	d
voter 3:	c	d	b	a	e
voter 4:	e	d	b	c	a
voter 5:	e	d	c	b	a
winner:	c (score 12)				

Brandt, F., Conitzer, V., Endriss, U., Lang, J., & Procaccia, A. D. (Eds.). (2016). *Handbook of computational social choice*. Cambridge University Press.

Control - Adding Voters

- Given a list R of registered voters, and a list S of unregistered voters, can we choose a sublist S’ of S with size at most k such that p is the unique winner of the election after adding S’ to the voter list?
- Example:

points:	5	4	3	2	1	0
voter 1:	*a*	*c*	*b*	*f*	*e*	*d*
voter 2:	*b*	*a*	*f*	*c*	*e*	*d*
voter 3:	*c*	*d*	*b*	*a*	*f*	*e*
voter 4:	*e*	*d*	*b*	*f*	*c*	*a*
voter 5:	*e*	*d*	*c*	*b*	*f*	*a*
winner:	*b* with score 16					



Brandt, F., Conitzer, V., Endriss, U., Lang, J., & Procaccia, A. D. (Eds.). (2016). *Handbook of computational social choice*. Cambridge University Press.

Control - Removing Voters

- Given a list R of registered voters, can we make *p* the winner b deleting no more than k votes?
- Example:

points:	5	4	3	2	1	0
voter 1:	a	c	b	f	e	d
voter 2:	b	a	f	c	e	d
voter 3:	c	d	b	a	f	e
voter 4:	e	d	b	f	c	a
voter 5:	e	d	c	b	f	a
winner:	b with score 16					

→

points:	5	4	3	2	1	0
voter 1:	a	c	b	f	e	d
~~voter 2:~~	~~b~~	~~a~~	~~f~~	~~c~~	~~e~~	~~d~~
voter 3:	c	d	b	a	f	e
voter 4:	e	d	b	f	c	a
voter 5:	e	d	c	b	f	a

c wins with score 13

Brandt, F., Conitzer, V., Endriss, U., Lang, J., & Procaccia, A. D. (Eds.). (2016). *Handbook of computational social choice*. Cambridge University Press.

Control - Immunity, Resistance, Vulnerability

- A voting rule is **immune** to a type of control if there is no action the chair can take to change the result
- Otherwise it is **susceptible** to this type of control
- The rule is **vulnerable** if the corresponding problem is in P
- The rule is **resistant** if the corresponding problem is NP-hard
- Immunity is rare, and also undesirable for some problems (adding voters, for instance)

Brandt, F., Conitzer, V., Endriss, U., Lang, J., & Procaccia, A. D. (Eds.). (2016). *Handbook of computational social choice*. Cambridge University Press.

Control - Immunity, Resistance, Vulnerability Examples

- For Condorcet, approval, and plurality
- Adding candidates
  - Condorcet and approval are immune, plurality is resistant
- Removing candidates
  - Condorcet and approval are vulnerable, plurality is resistant
- Adding/Deleting voters
  - Condorcet and approval are resistant, plurality is vulnerable

Brandt, F., Conitzer, V., Endriss, U., Lang, J., & Procaccia, A. D. (Eds.). (2016). *Handbook of computational social choice*. Cambridge University Press.

Partitioning Cases and Destructive Control

- Conduct a two-stage election by partitioning the set of alternatives such that some candidate wins both pre-elections and the main elections.
- Example (Borda, ties resolved alphabetically):

	3	2	1	0
v1	b	c	a	d
v2	d	a	b	c
v3	c	a	d	b
v4	b	c	a	d

b,c = 7 **a = 6** **d = 4**

b wins

v1	a	d
v2	d	a
v3	a	d
v4	a	d

v1	b	c
v2	b	c
v3	c	b
v4	b	c

v1	b	a
v2	a	b
v3	a	b
v4	b	a

a wins

Brandt, F., Conitzer, V., Endriss, U., Lang, J., & Procaccia, A. D. (Eds.). (2016). *Handbook of computational social choice*. Cambridge University Press.

Partitioning Cases and Destructive Control

- Conduct a two-stage election by partitioning the set of voters such that some candidate wins both pre-elections and the main elections.
- Example (Borda):

points:	5	4	3	2	1	0
voter 1:	a	c	b	f	e	d
voter 2:	b	a	f	c	e	d
voter 3:	c	d	b	a	f	e
voter 4:	e	d	b	f	c	a
voter 5:	e	d	c	b	f	a
winner:	b with score 16					

voter 3:	c	d	b	a	f	e

voter 1:	a	c	b	f	e	d
voter 2:	b	a	f	c	e	d
voter 4:	e	d	b	f	c	a
voter 5:	e	d	c	b	f	a

b wins with score of 13

c beats b by 3 to 2

Brandt, F., Conitzer, V., Endriss, U., Lang, J., & Procaccia, A. D. (Eds.). (2016). *Handbook of computational social choice*. Cambridge University Press.

Bribery

Given

(a) an election (A,R) where the set of voters is N=$\{1, ..., n\}$ and R contains a preference order $\succ_i$ for each i in N,
(b) a preferred alternative $p \in A$,
(c) a budget $B \in \mathbb{N}$, and
(d) a collection of price functions $\Pi = (\pi_1, ..., \pi_n)$. For each $i, 1 \leq i \leq n$, and each preference order $\succ$ over A, $\pi_i(\succ)$ is the cost of convincing the ith voter to cast vote $\succ$.

Does there exist a preference profile $R' = (\succ'_1, ..., \succ'_n)$ such that p is the winner of election (A, R') and $\Sigma_{i=1}^{n} \pi_i(\succ'_i) \leq B$?

Brandt, F., Conitzer, V., Endriss, U., Lang, J., & Procaccia, A. D. (Eds.). (2016). *Handbook of computational social choice*. Cambridge University Press.

Bribery

- Problem: what price functions to use?
- We can’t list all |A|! argument-value pairs for the function
- Limit discussion to three families of functions:
  - **discrete** - changing a vote costs 1
  - **$discrete** - changing a vote costs c_i (different for each voter)
  - **Swap-bribery** - define cost for swapping each two alternatives and sum up those costs
- Associated problems: *Bribery, $Bribery, Weighted-Bribery, Weighted-$Bribery, Swap-Bribery*

Brandt, F., Conitzer, V., Endriss, U., Lang, J., & Procaccia, A. D. (Eds.). (2016). *Handbook of computational social choice*. Cambridge University Press.

Bribery

f	f-Bribery
plurality	P
veto	P
2-approval	P
k-veto, $k \in \{2, 3\}$	P
k-approval, $k \geqslant 3$	NP-complete
k-veto, $k \geqslant 4$	NP-complete
Borda	NP-complete
STV	NP-complete
Bucklin	NP-complete
fallback	NP-complete
maximin	NP-complete
Copeland	NP-complete
Schulze	NP-complete
ranked pairs	NP-complete
approval	NP-complete
range voting	NP-complete

- For all voting rules, *Bribery* reduces to *$Bribery* and *Weighted-Bribery*
- Even more hardness results for *Swap-Bribery*
- Limited research on how to overcome hardness of bribery problems

Brandt, F., Conitzer, V., Endriss, U., Lang, J., & Procaccia, A. D. (Eds.). (2016). *Handbook of computational social choice*. Cambridge University Press.

Questions?

Brandt, F., Conitzer, V., Endriss, U., Lang, J., & Procaccia, A. D. (Eds.). (2016). *Handbook of computational social choice*. Cambridge University Press.