CS269I: Incentives in Computer Science
Lecture #3: Strategic Voting*

Tim Roughgarden†

October 3, 2016

1 Voting in Computer Science

This lecture is all about voting. Voting provides us with another example of a familiar system where it's interesting to think about incentives. But also, these days voting is naturally coming up in several computer science contexts.

1. *Rank aggregation.* Here, voters correspond to different ranking algorithms. Consider a prediction problem where the goal is to come up with a ranked list, such as the Web pages most relevant to a search query. Imagine you have several different heuristics for producing such ranked lists. Maybe one heuristic is based on anchor text, another on link structure, and another on the page content.[1] It's natural to try to combine these different and likely conflicting rankings into a "consensus ranking." This is exactly a voting problem. The hope is that the consensus ranking is somehow "better" or "more accurate" than the individual rankings. For example, maybe the best pages are high on all of the lists, whereas a spam page is high on only a subset of the lists, and hence is ranked low in the consensus ranking.

2. *Crowdsourcing.* Here, voters correspond to "workers" (in crowdsourcing terminology). For example, the voters could be workers on Mechanical Turk, or peer graders in a MOOC. Suppose each worker produces a ranking — maybe you showed them three different candidate user interfaces and asked about their most and least favorite options. A natural goal is to summarize the workers' results with a single ranking. Again, this is exactly a voting problem.

3. *Participatory democracy.* Here, voters are, um, voters. The goal in participatory democracy is to get more people involved in government decisions, especially at the

†Department of Computer Science, Stanford University, 474 Gates Building, 353 Serra Mall, Stanford, CA 94305. Email: tim@cs.stanford.edu.
[1]For some queries, the major search engines give surprisingly different lists of results.

local level.[2] Work-to-date has focused mostly on budgeting decisions, like which capital expenditures to prioritize. For example, residents might be asked whether they'd rather see money spent on improving parks, schools, or public housing. This forces voters to grapple with the types of trade-offs faced by the government. Participatory budgeting is getting increasingly popular—currently, 31 of New York City's 51 local districts use it every year.

Technology plays at least two important roles in enabling participatory democracy. The more obvious one is in making voting easy (e.g., voting on your smartphone rather than at a polling place).

More interestingly, technology also enables new types of voting. The systems currently in place typically use "k-approval voting" — each voter is told the overall budget (e.g., $1 million) and a list of project descriptions with costs, and the voter picks their k (e.g., 5) favorite projects, with no ordering between them. Projects are then sorted in decreasing order of the number of votes received, and funded in this order until the budget runs out. A drawback with this system is that voters need not take into account projects' costs (e.g., the k projects' combined costs are allowed to exceed the budget), which results in more expensive projects being overrepresented. One current prototype for a replacement is "knapsack voting," where a voter is allowed to approve any number of projects, as long as their total cost is at most the budget. This forces voters to account for project costs, in that voting for more expensive projects decreases the number of projects that you can vote for. It's hard to imagine implementing knapsack voting with a paper ballot at a polling station — a computerized platform for large-scale voting is essential for its viability.

2 Examples of Voting Rules

2.1 Formalism

The point of a voting mechanism is to make a decision as a function of what people want (i.e., their preferences). So we need a model of people's preferences (this will be a recurring theme in the course), and we'll use the same model of ranked lists that we were working with last week.

Precisely, let A denote a set of alternatives (different user interfaces, or Web pages, or candidates in a political election). Each voter i has her own ranked list over the alternatives in A, and declares some ranked list L_i to a voting mechanism. (L_i may or may not be voter i's true preferences—the mechanism has no way of knowing.)

A voting rule can then have two different forms, depending on whether its output is a full ranked list or just a single alternative (the "winner"). So it's either a map of the form

$$L_1, L_2, \ldots, L_n \mapsto L^*,$$

[2] For cool work being done by Ashish Goel's group here at Stanford, check out `http://pbstanford.org`.

where L^* is a ranking of A, or a map

$$L_1, L_2, \dots, L_n \mapsto a^*,$$

where $a^* \in A$.[3] The former type of map is what you want in the rank aggregation application, and perhaps also in the crowdsourcing application. The latter map suffices for political elections (of a single candidate).

Last week's discussion of the Draw and of stable matching also involved participants with ranked lists over the possible outcomes. How is the voting problem different? The voting formalism is general enough to capture the applications last week as special cases, with the set A of alternatives corresponding to all possible assignments (of students to rooms, or residents to hospitals). But there was additional structure to everyone's preferences in the Draw and stable matching, namely that every participant only cared about what their own allocation was, and not those of the other participants. In an election, the outcome of the voting rule can affect everybody.

2.2 Majority Rule

Suppose there are only two alternatives ($|A| = 2$). One obvious voting rule is *majority vote*: elect that alternative that appears first in the largest number of voters' lists. (If there is an even number of voters and a tie, break the tie in some canonical way, perhaps randomly.) It's hard to think of any other voting rule that you might plausibly use when there are two alternatives.

Observe that the majority vote rule is strategyproof. If you submit a false list (i.e., reverse the order of the two alternatives), the only possible effect is to flip the winner from the alternative that you prefer to the other alternative. So maybe designing voting rules isn't all that difficult?

2.3 Plurality Rule

Suppose now that $|A| \geq 3$, as is the case in many applications. What would be the analog of the majority rule? (Suppose we only care about electing a single winner.) If some candidate appears first in more than 50% of the voters' ballots, then it's intuitively clear that she should be the winner. But in many real elections with 3 or more candidates, this does not happen. (E.g., with 3 candidates, you might have a 40/35/25 split.) So then what?

In most countries (including the U.S.), the most common solution is the *plurality rule*, which elects the candidate with the most first-place votes. (With this result, a voter does not need to submit her full ranked list; just her top choice suffices.) Is this rule strategyproof?

You don't have to look that far back in history to understand the incentive issues with the plurality rule. In the 2000 U.S. presidential election, the outcome of the election came

[3] Some voting theory jargon: the former type of map is called a *social welfare rule*, the latter a *social choice rule*.

down to the state of Florida.[4] Here were the final vote tallies in Florida:

Candidate	Party	Vote Total
Bush	Republican	2,912,790
Gore	Democrat	2,912,253
Nader	Green	97,488

There were additional candidates, but the fourth-place finisher was well behind Nader. This was obviously a very close election—there was about a 500-vote difference between Bush and Gore, a margin of less than .01%.

It is generally assumed that most voters who viewed Nader as their first choice also viewed Gore as their second choice. (Nader's political stance was substantially closer to that of Gore than that of Bush.) Thus Nader was a "spoiler" candidate, in the sense that his presence flipped the election's outcome (from Gore to Bush), despite the fact that he was nowhere close to winning the election.

This example also shows why the plurality rule is not strategyproof. If your true preferences were to rank Nader over Gore over Bush, and if you knew that Nader had no chance of winning, then it makes sense to vote for Gore instead of Nader—this could cause your second-favorite candidate to get elected over your third-favorite candidate.[5]

More generally, the plurality rule tends to be biased toward "extreme" candidates. To make the point clear, imagine there there are 10 "mainstream" candidates, all very similar to each other, and 1 "extreme" candidate. Imagine that 90% of the population prefers any of the mainstream candidates to the extreme, with the other 10% preferring the extreme candidate. If the mainstream candidates manage to split the 90% of the vote equally, then each mainstream candidate receives only 9% of the first-place votes, and the extreme candidate winds up being the winner. This is one of the reasons why voting theorists tend to have a dim view of the plurality rule (despite its prevalence in practice).

2.4 Ranked-Choice Voting

We've seen that the plurality rule has some serious issues. So what else could you do? Another rule you hear about frequently in the news is *ranked-choice voting*, also known as *single transferable vote (STV)* or *instant-runoff voting*. For example, not too long ago Oakland switched their local elections from plurality vote to ranked-choice voting.

For ranked-choice voting, voters submit a full ranked list (not just their first choice). First, if there is some alternative a^* that receives more than 50% of the first-choice votes, then a^* is the winner. Otherwise, the alternative with the fewest first-choice votes is deleted and the winner is computed recursively on the remaining alternatives. Eventually, there will be only two alternatives remaining at which point the rule is the same as the majority rule.

[4] Let's leave aside incentive issues stemming from the idiosyncratic electoral college—that would make for a whole other lecture.

[5] Such voting is sometimes deemed "dishonest"—but can you really blame a voter who understands the incentive issues for acting according to her preferences?

Note that in the example at the end of the previous section, ranked-choice voting will elect one of the mainstream candidates.

For example, consider the following five votes over the set of alternatives $A = \{a, b, c, d\}$:

	Voters #1,2	Voters #3,4	Voter #5
1st Choice	a	b	c
2nd choice	d	a	d
3rd choice	c	d	b
4th choice	b	c	a

Each of a, b, c has at least one first-choice vote, while d does not have any. So we eliminate d and recurse on the remaining alternatives, leaving us with:

	Voters #1,2	Voters #3,4	Voter #5
1st Choice	a	b	c
2nd choice	c	a	b
3rd choice	b	c	a

Since a, b have two first-choice votes each, while c only has one, we eliminate c and repeat:

	Voters #1,2	Voters #3,4,5
1st Choice	a	b
2nd choice	b	a

In the last round we apply the majority rule, so b wins.

Should we be happy with this outcome? Not necessarily. For example, looking back at the original preferences, note that a majority of the voters prefer c to b, and also a majority prefer d to b. So voters might well wonder how b got elected over both c and d.

Ranked-choice voting is not strategyproof (see Exercise Set #2). The intuition is that there can be an incentive to influence who gets eliminated early on, so that your preferred candidate gets more favored matchups in later rounds. Compared to the plurality rule, however, it seems trickier for a voter to reason about how to game the system in ranked-choice voting.[6] This is one of the reasons why most voting theorists prefer ranked-choice voting to plurality voting.

For a given set of voter preferences, changing the voting system can change the election outcome. This fact tends to politicize debates over which system should be used. For example, in 2011 the U.K. held a referendum on whether or not to switch from plurality voting to ranked-choice voting. The proposal was soundly defeated, and it seems that one reason was a belief that a switch would increase the probability of the re-election of a specific candidate who was strongly disliked in certain circles. It's important to remember, however, that no voting system is intrinsically more "left-wing" or "right-wing" than another; differences between voting systems become a political issue only once you have specific candidates and voter preferences in mind.

[6] For example, even if you know everyone else's votes, the problem of checking for a profitable manipulation is NP-hard (in the worst case, as the number of alternatives grows large) [2].

2.5 The Borda Count

Another well-known voting rule is the *Borda count*, which uses a point system. Voters submit their full ranked lists. An alternative gets $|A|$ points for each first-choice vote, $|A|-1$ points for each second-choice vote, and so on, with 1 point for each last-choice vote. For example, with voter preferences as in Section 2.4, the point totals would be 15 for a, 12 for b, 10 for c, and 13 for d. Thus the Borda count would elect a (in contrast to the outcome of ranked-choice voting). The Borda count has been used many times—often in sports, for example. The Eurovision contest also uses a rule similar to the Borda count.

More generally, one can use a different choice of how many points gets accrued per jth-choice vote. Voting rules based on such point systems are called *positional* or *scoring* rules. For example, plurality is a scoring rule, with 1 point per first-choice vote and 0 points for all other votes. Is ranked-choice voting a scoring rule?

The Borda count rule also fails to be strategyproof. For example, you might want to rank the closest competitor to your preferred alternative last.

3 The Gibbard-Satterthwaite and Arrow Impossibility Theorems

All of the voting rules we've looked at (save majority, in the $|A|=2$ case) have failed to be strategyproof. Are we just not being smart enough, or is there a more fundamental issue?

The *Gibbard-Satterthwaite theorem* states that it's not our fault that we haven't thought of any strategyproof voting rules with three or more alternatives—non-trivial such voting rules simply *do not exist*.

To make a precise statement, we need to define a "non-trivial voting rule." Here are two "trivial" rules:

- **Dictator:** a dictator rule has a dictator voter i, and always elects i's first choice.
- **Duple:** a duple rule chooses a pair $a, b \in A$ of alternatives (independent of voters' preferences), and runs majority vote between a and b.

Theorem 3.1 (Gibbard-Satterthwaite Theorem [4, 8]) *Every strategyproof voting rule that can produce at least three different outcomes is a dictator.*

Ouch! So the Gibbard-Satterthwaite theorem basically says that strategyproofness is hopeless in general for voting.[7]

The Gibbard-Satterthwaite theorem is closely related to, and can be derived from, an even more famous impossibility result: *Arrow's impossibility theorem*.[8] We state the theorem and then provide the missing definitions.

[7] The Gibbard-Satterthwaite theorem is about deterministic voting rules, but there are analogous impossibility results also for randomized voting rules [5].

[8] Arrow received one of the earliest Nobel Prizes in economics (in 1972), in part for this result. He has been a professor at Stanford for many years.

Theorem 3.2 (Arrow's Impossibility Theorem [1]) *With three or more alternatives, no voting rule satisfies the following three properties:*

1. *Non-dictatorship.*
2. *Unanimity.*
3. *Independent of irrelevant alternatives (IIA).*

Non-dictatorship just means that there is no dictator (a voter i such that the output of the voting rule is always the same as i's ranked list). Unanimity means that if every voter ranks a over b, then the voting rule should also rank a over b. IIA means that, for every pair a, b of alternatives, the relative order of a, b in the produced ranking should be a function only of the relative order of a, b in each voter's list, and not depend on the position of any "irrelevant" alternative c in anyone's preferences. Most natural voting rules satisfy the first two conditions; IIA is the one that does most of the work (and is the most debatable as a desirable property).

For example, the plurality rule does not satisfy IIA; the outcome of the 2000 U.S. Presidential Election was certainly not independent of the position of Nader in voters' preference lists. More generally, a voting rule that allows spoiler candidates cannot satisfy IIA.

Many proofs of Theorems 3.1 and 3.2 are known (e.g. [3, 9]). If you're mathematically inclined, you're encouraged to check them out—the proofs are accessible, if a little tedious at points.[9]

4 A Tractable Special Case: Single-Peaked Preferences

The Gibbard-Satterthwaite theorem dashes any hopes of having a general and useful strategyproof voting rule. So should we give up? Well, that's not exactly an option—elections with three or more candidates aren't going away, and some voting rule must be chosen. So just as computational intractability results (like NP-hardness results) forces us to compromise our goals (resorting to approximation algorithms, restricting to computationally tractable special cases, etc.), impossibility theorems in voting theory imply that compromises must be made. For example, one can relax the strategyproof condition in various ways. We conclude this lecture with a different type of result, which identifies an interesting restricted class of preferences that admits a natural strategyproof voting rule [6].

Suppose the set of alternatives is the unit interval $[0, 1]$. For example, the interval could represent the political spectrum (from radical to reactionary). A voter i has *single-peaked*

[9]One can also deduce the Gibbard-Satterthwaite theorem from Arrow's theorem, using a type of "reduction." That is, given an allegedly non-trivial and strategyproof voting rule, one can extract a voting rule that satisfies the three conditions in Arrow's theorem. Put differently, failures of IIA give rise to opportunities for strategic manipulation. Most of the details of the reduction resolve the "typechecking error" between the two theorems—Arrow's theorem is a result about social welfare functions (voting rules that produce a full ranked list) while the Gibbard-Satterthwaite theorem holds even for social choice functions (voting rules that merely elect a winner). See e.g. [7] for more details.

preferences if there is a "peak" $p_i \in [0,1]$ such that whenever z is farther from p_i than y (meaning $z < y < p_i$ or $z > y > p_i$), the voter prefers y to z. See Figure 1.

Figure 1: Single-peaked preference. Voter i has peak p_i; since y is closer to p_i than z, the voter prefers y to z.

One can imagine scenarios where single-peaked preferences are a reasonable first-cut approximation of voters' preferences (electing a politician, locating a school, etc.).

Suppose each voter votes by providing a reported peak $x_i \in [0,1]$. Which alternative should we choose? One idea is to choose the average, $\frac{1}{n}\sum_{i=1}^{n} x_i$. But note that this is not strategyproof: a voter might be able to pull the chosen outcome closer to her peak by reporting an overly extreme peak. Can you think of real-life examples of this phenomenon?

A second and better idea is to choose the median of the reported peaks. (For simplicity, assume there is an odd number of voters.) The median voting rule *is* strategyproof. The only way a voter can manipulate the median is to report a peak on the opposite side of the median from her true peak (Figure 2). But this can only pull the median farther away from her true peak, a worse outcome for her.

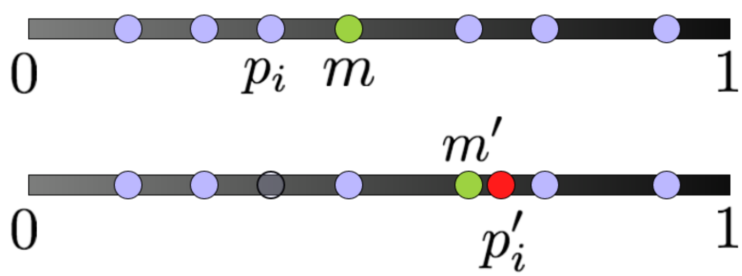

Figure 2: Median m of reported peaks (with $n = 7$). A voter i with peak p_i cannot change the median unless she overshoots to the other side by misreporting p_i'; this would move the median m' even further from i's true preference p_i.

References

[1] K. J. Arrow. *Social choice and individual values.* Wiley, 1951.

[2] J. J. Bartholdi and J. B. Orlin. Single transferable vote resists strategic voting. *Social Choice and Welfare*, 8(4):341–354, 1991.

[3] F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia, editors. *Handbook of Computational Social Choice*. Cambridge University Press, 2016.

[4] A. Gibbard. Manipulation of voting schemes: A general result. *Econometrica*, 41(4):587–601, 1973.

[5] A. Gibbard. Manipulation of schemes that mix voting with chance. *Econometrica*, 45(3):665–681, 1977.

[6] H. Moulin. On strategy-proofness and single peakedness. *Public Choice*, 35(4):437–455, 1980.

[7] N. Nisan. Introduction to mechanism design (for computer scientists). In N. Nisan, T. Roughgarden, É. Tardos, and V. Vazirani, editors, *Algorithmic Game Theory*, chapter 9, pages 209–241. Cambridge University Press, 2007.

[8] M. A. Satterthwaite. Strategy-proofness and Arrows conditions: existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10(2):187–217, 1975.

[9] Y. Shoham and K. Leyton-Brown. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, 2009.