CS269I: Incentives in Computer Science
Lecture #4: Voting, Machine Learning, and Participatory Democracy*

Tim Roughgarden†

October 5, 2016

1 Preamble

Last lecture was all about strategyproof voting rules (and the lack thereof). Highlights included: the majority rule is strategyproof (for two alternatives); plurality, ranked-choice voting, and the Borda count all fail to be strategyproof; the Gibbard-Sattherwaite theorem states that no non-trivial voting is strategyproof; but sometimes by compromising and restricting the space of possible voter preferences, positive results can be recovered (like the median rule for single-peaked preferences).

Strategyproofness is a very reasonable thing to worry about in political decisions, including the participatory democracy applications discussed in Section 5. But recall the other two computer science applications of voting that we mentioned last lecture — rank aggregation (e.g., of multiple heuristics ranking a set of Web pages by relevance) and crowdsourcing (e.g., aggregating the opinions of many Mechanical Turk workers about several different user interfaces). Is strategyproofness relevant here? Not really, since the voters don't actually care personally about which outcome gets chosen. So in applications like these, what properties of voting rules should we focus on?

2 Voting Rules as Maximum Likelihood Estimators

2.1 Two Interpretations

There are two rather different ways to interpret a voting rule.

*©2016, Tim Roughgarden.

†Department of Computer Science, Stanford University, 474 Gates Building, 353 Serra Mall, Stanford, CA 94305. Email: tim@cs.stanford.edu.

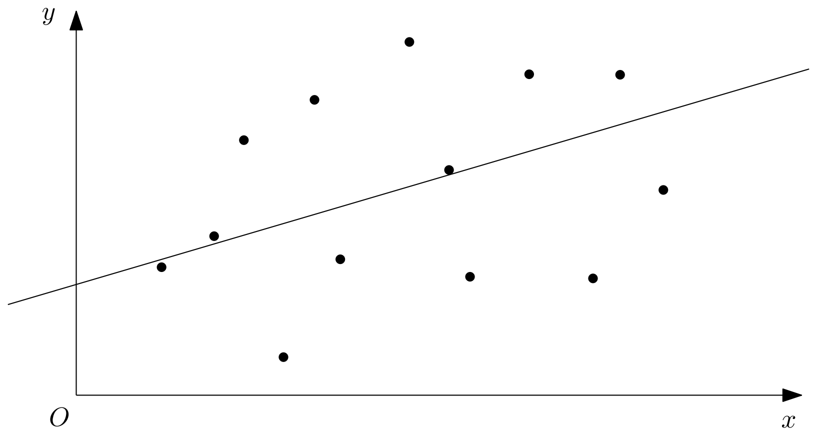

Figure 1: A set of points and the "best-fit line."

1. *As a consensus between different subjective opinions.* Here, voters have genuinely different preferences (as in an election), and a voting rule is a method for aggregating them. With this interpretation, voters are motivated to influence the output of the voting rule, and strategyproofness is a relevant property.

2. *To recover the "ground truth" from noisy, imperfect estimates.* Here, voters are effectively cooperating in a quest for the truth. There is an objectively correct answer—like the true ranking of a set of Web pages by relevance—and the voters do their best to figure it out but inevitably make errors. Under this interpretation, strategyproofness is not a relevant property.

If you've studied machine learning, then the second perspective should remind you of the learning problem of inferring an underlying relationship from noisy data. Could this be a fruitful analogy for the design and analysis of voting rules?

2.2 Digression on Regression

To see how the analogy might work, let's review a canonical machine learning problem, linear regression. The setup is: you are given n data points $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_y)$, where the $\mathbf{x}_i$'s are points in $\mathbb{R}^d$ and the y_i's are real values. (Go ahead and assume that $d = 1$, if you like.) The goal is to compute a "best-fit line" (Figure 1). (Or more generally, for $d > 1$, the best-fit affine subspace.) This is sort of like a voting problem, where each data point is effectively voting for a line that passes through it. Just as there are many voting rules, there are many choices of how to define a "best-fit line." For example, how should the distance between a point and line be measured (perpendicular or vertical?), and how should we aggregate these distances over the data points?

Ordinary least squares is a canonical choice of how to define the "best-fit line," as the one minimizing the total squared vertical distance:

$$\sum_{i=1}^{n} \left(\sum_{j=1}^{d} a_j x_{ij} - y_i \right)^2, \qquad (1)$$

where $(a_1, \ldots, a_d, b)$ are the coefficients specifying the subspace (a line, if $d = 1$).

How can one justify this notion of "best fit" over others? One reason is technical convenience—the objective function (1) is convex and can be minimized by gradient descent (for example), and there is even a closed-form solution. But is there a more fundamental reason why this is the "right" notion of the "best fit?"

One justification uses a probabilistic interpretation. For clarity, assume that $d = 1$ and so we are looking for the "best-fit" line. Suppose there is a "ground truth" line $a^*x + b$—the relationship that we're trying to learn. For data points $x_1, \ldots, x_n$, assume that each label y_i has the form $y_i = a^*x_i + b + \epsilon_i$, where the ϵ_i's are drawn i.i.d. from a Gaussian (i.e., normal, or "bell curve") distribution with mean 0. (The variance turns out not to matter.) Now suppose that, given $(x_1, y_1), \ldots, (x_n, y_n)$, you want to reverse engineer your "best guess" of the unknown ground truth. This is known as the *maximum likelihood* solution—given the data (the (x_i, y_i) pairs), which line maximizes the probability of generating the data? That is, the maximum likelihood solution is the solution to $\max_{a,b} \mathbf{Pr}[\text{data} \mid a, b]$. The punchline is: *ordinary least squares computes the maximum likelihood solution*—the most likely linear relationship between the x_i's and the y_i's, given our assumption about how the data is generated. (This is a somewhat tedious but not overly difficult calculation—see e.g. CS229 for details.)

2.3 Take-Aways

How can you argue that one solution to a problem is better than the others? So far, we've focused on whether or not proposed solutions have certain desirable properties, like strategyproofness or Pareto optimality. The discussion in Section 2.2 suggests another general approach: prove that, in some model, the solution is provably *optimal*. (E.g., maximizes the likelihood objective function with respect to some generative model of data.) The hope is that such a solution continues to be a very good solution even when the assumptions needed for optimality are relaxed significantly.

Is this philosophy relevant for voting? For example, suppose we posit some ground truth ranking of the alternatives, and assume that votes are generated probabilistically, as noisy versions of the ground truth. Given such a model and votes, it makes mathematical sense to talk about the maximum likelihood ground truth ranking—the ranking that is most likely to generate the observed votes. This induces a voting rule (given votes, return the most likely ground truth ranking) — are such voting rules natural and/or interesting?

3 Majority Vote as a Maximum Likelihood Estimator

The Marquis de Condorcet was an interesting guy—you should look him up. He was a French intellectual in the 18th century. He initially focused on math, but his interests soon branched out into philosophy and political science (hence the connection to voting theory). Condorcet played an active role in the French Revolution, but afterwards had a falling out with the new government, became a fugitive, was caught and sent to prison, where he later died under mysterious circumstances.

Condorcet was the first to give a probabilistic justification of a voting rule, specifically for the majority rule when there are only two alternatives ($A = \{a, b\}$). This was quite novel at the time, as probability theory was relatively new on the mathematical scene.

Here's the formal model: assume that one of the two alternatives is the "correct" one (i.e., the ground truth), and that each voter independently votes for the correct alternative with probability p and the incorrect alternative with probability $1 - p$, where $p \in (\frac{1}{2}, 1]$ is a parameter. Thus, a voter is more likely to be correct than incorrect.

Given this model of generating data, what is the maximum likelihood estimator? Suppose the data consists of k votes with a on top, and $n - k$ votes with b on top. (So n voters total, and for simplicity assume that n is odd.) If the ground truth is that a is the better alternative, then the probability of seeing this data is

$$\binom{n}{k} p^k (1-p)^{n-k}, \tag{2}$$

since there are $\binom{n}{k}$ choices for the subset of k correct votes, and given such a choice, all of the allegedly correct votes really should be correct (this has probability p^k) and similarly for the incorrect votes (probability $(1-p)^{n-k}$).

Similarly, if b is the better alternative, then the probability of seeing the given data is

$$\binom{n}{k} p^{n-k} (1-p)^k. \tag{3}$$

Since $p > \frac{1}{2}$, the expression (2) is bigger than (3) if and only if $k > n - k$, or equivalently $k > \frac{n}{2}$. That is, alternative a is the maximum likelihood estimator if and only if it is the winner by majority vote. This argument provides a justification of the majority rule that is equally interesting as and totally different from the strategyproofness considerations discussed last lecture.[1]

4 The Kemeny Voting Rule

In the $|A| = 2$ case, we already knew the voting rule that we cared about (majority), and reverse engineered a natural sense in which the rule is optimal. In the $|A| \geq 3$ case, we have

[1]Condorcet also proved, using similar calculations, that the probability that the majority rule recovers the ground truth tends to 1 as $n \to \infty$ (for any fixed $p > \frac{1}{2}$).

a bewildering number of incomparable options for which voting rule to use. Maybe we can use the MLE approach to help us select among our options, or even to come up with a new rule?

4.1 The Maximum Likelihood Estimator for Independent Pairwise Comparisons

Our model of the data is the following. There is a ground truth ranking π of A. The vote of voter i describes the outcomes of all $\binom{|A|}{2}$ possible pairwise comparisons. (This is a different setup than previously, where bidders submitted rankings.) For convenience, we allow a vote to contain cycles (e.g., a beats b, b beats c, but c beats a). The benefit of allowing cycles is that we can use a simple generative model for votes: independently for each voter i and each pair $a, b \in A$ of alternatives, with probability p i's vote agrees with π on the relative order of a and b, and with probability $1 - p$ its comparison between a and b is the opposite of π. (Again, $p > \frac{1}{2}$ is a parameter of the model.) Note that this model of "noisy votes" can produce cycles (which is why we're allowing them).[2] Now that we've fixed a model of data, we can ask about the maximum likelihood estimator. So what is it?

Fix the votes of all n voters. For a ranking σ of A and outcomes $a, b \in A$, let k^σ_{ab} denote the number of votes that agree with π on the relative order of a and b. An observation: ranging over all rankings σ, k^σ_{ab} can only take on one of two possible values. If $k^\sigma_{ab} = x$, then for every σ' we either have $k^{\sigma'}_{ab} = x$ (if σ' ranks a and b in the same relative order as σ) or $k^{\sigma'}_{ab} = n - x$ (if not).

The maximum likelihood computation now proceeds as for the majority rule, with one term of the form (2) for each pair $a, b \in A$ of distinct outcomes. (This is using the fact that the different pairwise comparisons of a voter are flipped independently.) That is, for a ranking σ of A, the probability that σ would generate the observed votes is the product of all $\binom{|A|}{2}$ terms of the form

$$\binom{n}{k^\sigma_{ab}} p^{k^\sigma_{ab}} (1-p)^{n-k^\sigma_{ab}}. \tag{4}$$

For fixed a, b, since k^σ_{ab} can only take on two possible values (x or $n - x$, for some x), the binomial coefficient in (4) is a *constant*, independent of σ. This means that when identifying the most likely σ, we can ignore these binomial coefficients. Ignoring these terms and multiplying the $\binom{|A|}{2}$ expressions of the form (4), identifying the maximum likelihood solution σ boils to maximizing

$$p^{\sum_{a,b\in A} k^\sigma_{ab}} (1-p)^{\sum_{a,b\in A}(n-k^\sigma_{ab})}. \tag{5}$$

Since $p > \frac{1}{2}$ and the two exponents in (5) always sum to $\binom{|A|}{2}n$ (no matter what σ is), the ranking with the maximum likelihood is the one maximizing

$$\sum_{a,b\in A} k^\sigma_{ab}, \tag{6}$$

[2] There are analogous results for the case where voters submit rankings, see Section 4.2.

which is the number of pairwise agreements with the data. This idea generalizes the majority rule by averaging across all pairs of outcomes.

This voting rule—given pairwise votes, output the ranking σ that maximizes (6)—is called the *Kemeny rule*.[3] This rule is different from all of the other ones that we've seen, but it is quite natural, and has been studied extensively in the context of rank aggregation (e.g. [3]).

4.2 The Mallows Model

Here's a different model of data, where voters submit ranked lists instead of pairwise comparisons, for which the Kemeny rule is again the maximum likelihood estimator. Suppose again that there is a ground truth ranking π. In the *Mallows model*, the probability that a voter submits the vote (i.e., ranked list) σ decays exponentially with the number of disagreements between σ and π (equivalently, the "bubblesort" distance between σ and π). Thus votes σ "far" from the ground truth, in the sense that a large number of swaps between adjacent elements are needed to transform σ into π, are very unlikely. A calculation similar to that in (4)–(6) verifies that Kemeny rule is the maximum likelihood estimator for this model.[4]

4.3 Computational Considerations

To this point, we have ignored the question of how to efficiently evaluate different voting rules. Last lecture this didn't really come up, since the voting rules discussed (plurality, ranked-choice voting, and the Borda count) are all easy to implement as a computationally efficient algorithm. What about the Kemeny rule? Given votes, how do we compute the output of the rule? When there is a small (constant) number of alternatives (and possibly many voters), the problem can be solved by exhaustive search (just try all $|A|!$ possibilities). In general, however, the problem of evaluating the Kemeny rule is NP-hard [1].

All of the usual comments about NP-hardness apply here: it is not a death sentence, and does not mean that the problem cannot be solved satisfactorily on the instances that you care about. NP-hardness does mean, however, that you need to lower your expectations and make one or more compromises. While it's likely that small and even medium-sized problems can be solved in a reasonable amount of time, large problems are likely to be problematic (unless they possess special structure).[5] In a rank aggregation context, where the number of

[3]The rule was proposed by Kemeny [5]; its interpretation as a maximum likelihood estimator is due to Young [6].

[4]There are many other results that justify various voting rules as maximum likelihood estimators with respect to some data model. Not all natural voting rules can be justified in this way, however. See [2] and Exercise Set #2 for further examples.

[5]There's actually a Web site, Pnyx (`pnyx.dss.in.tum.de`), that provides an implementation of the Kemeny rule (in addition to other voting rules). Pnyx is like Doodle on steroids, and offers many different ways of aggregating the results of complex polls. The algorithm is based on integer programming, and should suffice for typical small- and medium-sized problems. (Preliminary tests by your instructor with 11 alternatives took less than one second.)

alternatives can be large, heuristics are often used to approximate the output of the Kemeny rule [3].

5 Participatory Budgeting

Recall that the goal in participatory democracy is to get more people involved in government decisions, especially at the local level. 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.

5.1 k-Approval 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.

k-Approval Voting

1. Each voter votes for at most k projects.
2. Sort the projects in decreasing order of number of votes.
3. Fund the maximal prefix such that the total cost is at most the budget B.[6]

A drawback with k-approval voting is that voters need not take into account projects' costs (e.g., the combined costs of the k projects are allowed to exceed the budget), which results in more expensive projects being overrepresented. For example, suppose that the budget is 1 million USD, that $k = 1$, and that all voters have identical values for the three possible projects:

Project Number	Value (per voter)	Project Cost
1	4	1 million
2	3	500K
3	2	500K

It is clear that the socially optimal thing to do is to fund the second and third projects, garnering value 5 per voter. It's not clear that voters will vote in this way, however. Given

[6] What if there is not enough money left to fund the fifth-most popular project, but there is for the (cheaper) sixth-most popular project? In practice, one typically will fund the latter project also (and any subsequent projects that there is money for).

that a voter can only vote for one project, it is possible (even likely) that she will vote for the single project for which she has the most value. This would result in everybody voting for the first project, a suboptimal result.[7] Thus the outcome of straightforward voting in the k-approval scheme need not be Pareto optimal. Approval voting is also not strategyproof (exercise).

5.2 Knapsack Voting

We next take a glimpse into the current state-of-the-art, and describe one current prototype for a replacement: *knapsack voting* [4].[8] The idea is to allow a voter to approve any number of projects, as long as their total cost is at most the budget.

Knapsack Voting

1. Each voter votes for a subset S_i of projects for which the total cost is at most the budget B.[9]
2. Sort the projects in decreasing order of number of votes.
3. Fund the maximal prefix such that the total cost is at most the budget B.

For the proofs, we also allow the final project considered—the most popular one that can't be fully funded—to be partially funded, so that the entire budget of B is spent. It may or may not be realistic to partially fund projects (e.g., renovate only one floor of a school rather than the entire building), but partial funding and insistence on spending all of one's budget are both common in practice.

The intuition behind knapsack voting is that it forces voters to account for project costs—voting for more expensive projects decreases the number of projects that you can vote for—and hence should result in a better choice of projects. In the three-project example in the preceding section, straightforward knapsack voting results in the optimal choice (with the second and third projects getting funded). But how would we argue this point more generally?

5.3 Properties of Knapsack Voting

Our strategy is to formally prove that knapsack voting has a number of nice properties, under relatively specific assumptions. As always, the hope is that the conclusions reached remain (at least approximately) valid much more generally.

[7] Even with $k = 2$, it seems likely that the outcome would be the same (why?).

[8] See also `http://pbstanford.org`, and especially the "Boston '16" link to see knapsack voting in action.

[9] 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.

The first order of business is to model what voters want. We'll strive for both strategyproofness and Pareto optimality guarantees, and this requires making fairly specific assumptions (otherwise impossibility results kick in).[10] The two assumptions are:

1. Voter i has some set of projects S_i^* that she wants to fund, with the total cost of S_i^* at most the budget B.
2. Voter i wants as much money as possible to be spent on the projects in S_i^*. Thus the utility of voter i is

$$\sum_{j \in S_i^*} [\text{money spent on project } j]. \tag{7}$$

The definition (7) implies that if a project (of S_i^*) is partially funded, then the utility earned is pro-rated accordingly.[11] The most unrealistic aspect of this utility model is the extreme assumption that a voter i has absolutely no value for any project outside its preferred set S_i^*.

Under the assumptions, knapsack voting has several nice properties.

Proposition 5.1 ([4]) *With voter utilities as in* (7)*, knapsack voting is strategyproof, meaning that a player always maximizes her utility by voting for her true set S_i^*.*

We leave the proof to Exercise Set #2. Intuitively, misreporting your preferred set can only transfer funds from projects you do want to projects that you don't want (a bit like with the median mechanism from last lecture).

Proposition 5.2 ([4]) *With voter utilities as in* (7)*, and assuming that voters report their true sets, knapsack voting results in a Pareto optimal choice of projects.*

Again, we leave the proof to Exercise Set #2. Intuitively, this follows from the greedy nature of the way projects get funded, from most popular to least popular.

Our final result ties knapsack voting back into the discussion of the first part of this lecture (Sections 2–4).

Proposition 5.3 ([4]) *Knapsack voting is the maximum likelihood estimator for a semi-natural generative model of votes.*

We won't prove (or even formally state) Proposition 5.3, but the setup is reminiscent of the Mallows model for ranked lists (Section 4.2). The model here is: there is a ground truth set S^* of "correct" projects, and the probably of seeing a given vote S from a voter decays exponentially with the cost of the projects in $S^* \setminus S$, that is, with the amount of funds not spent on S^*. Proposition 5.3 then states that knapsack voting is the maximum likelihood solution for this Mallows-type model.

[10] There is ongoing work focused on relaxing these assumptions.

[11] Also, we assume that no project is ever funded at a level larger than its cost.

References

[1] J. Bartholdi III, C. Tovey, and M. Trick. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare*, 6:157–165, 1989.

[2] V. Conitzer and T. Sandholm. Common voting rules as maximum likelihood estimators. In *Proceedings of the 21st Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 145–152, 2005.

[3] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the Web. In *Proceedings of the 10th International World Wide Web Conference (WWW)*, pages 613–622, 2001.

[4] A. Goel, A. Krishnaswamy, S. Sakshuwong, and T. Aitamurto. Knapsack voting. Working paper, 2016.

[5] John G. Kemeny. Mathematics without numbers. *Daedalus*, 88(4):577–591, 1959.

[6] H. P. Young. Condorcet's theory of voting. *American Political Science Review*, 82:1231–1244, 1988.