Computational Social Choice: The First Four Centuries

Making the right decision.

By Ariel D. Procaccia

DOI: 10.1145/2043236.2043249

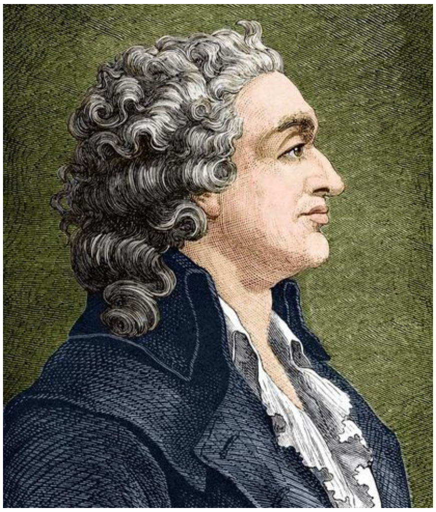
Social choice theory is an area of economics that studies the foundations of collective decision making. We frequently participate in collective decisions in our day-to-day lives when we vote in an election, select students for admission to a graduate program, or even (taking a more nostalgic point of view) share a cake with our siblings. Social choice theory provides mathematical models that capture these situations and others, as well formal guidelines for making the "right" choices.

In the last two decades computer scientists have become interested in social choice, leading to the rise of a new field called computational social choice. This article is meant to serve as an (extremely biased) introduction to the field. It would be a shame though to tell the story of computational social choice without recounting some of the delightful history of social choice theory itself, which spans several centuries (some would say millennia). In fact, as we shall see, some of the prominent figures of social choice theory were very colorful indeed!

THE 18TH CENTURY

Marie Jean Antoine Nicolas de Caritat, Marquis de Condorcet, is sometimes referred to as the founder of social choice theory, at least in its modern form. He was born in 1743 as the only heir to a family of French nobility. Condorcet was an exceptional mathematician who published ground-breaking work in fields such as integral calculus; one of his books was praised by the illustrious mathematician Lagrange as "filled with sublime ideas." At the recommendation of another famous French mathematician, D'Alembert, Condorcet was elected to the French Academy of Sciences, and later became its president.

Nicolas de Caritat's "Condorcet Method" is designed to simulate pairwise elections between all candidates.



In addition to being a first-rate mathematician, Condorcet was also an accomplished writer, and in general was considered one of France's most prominent intellectuals. He supported many a liberal cause, including economic freedom, abolition of slavery, equality, and women's rights.

Because Condorcet was a nobleman, some people mistakenly believe that he was executed during the French revolution. In fact, he became one of the leaders of the revolution. However, a few years later he was declared a traitor due to a disagreement over a draft of the constitution. Fleeing the law, Condorcet remained in hiding for several months. When he was finally forced to leave his hiding place, he took refuge in an inn, disguised as a commoner. Sadly, his deception was uncovered when he asked for an omelet with an "aristocratic amount of eggs, believed to have been 12" [1]. He was thrown in jail, but was found mysteriously dead two days later. One theory asserts that he took his own life with a vial of poison once given to him by a friend in case he ever faced an encounter with the dreaded guillotine.

Today Condorcet's greatest legacy is perhaps his 1785 book titled *Essay on the Application of Probability Analysis to Majority Decisions*. Among other important contributions, the book presents a discovery about the very nature of majority decisions, which at the time was nothing less than shocking.

Suppose that the preferences of voters can be represented (and expressed) as a ranking of the candidates. If the majority of voters rank candidate x above candidate y, that is, prefer x to y, then x is clearly a better candidate than y; I say that x beats y in a pairwise comparison. Now consider the following scenario. Voter 1 prefers a to b to c (I denote this ranking as $a > b > c$); voter 2 has the ranking $b > c > a$; and the preferences of voter 3 are represented by the ranking $c > a > b$. In this setting, a beats b in a pairwise comparison, b beats c, and c beats a—there is a cycle in the preferences of the majority! This troubling phenomenon is known today as the "Condorcet Paradox."

Nevertheless, some fortuitous voting scenarios give rise to a candidate who beats every other candidate in a pairwise comparison. Condorcet argued that any voting rule, which selects a candidate given the preferences of all voters, must select such a strong candidate if it is given preferences where one exists. Therefore, a candidate who beats every other candidate in a pairwise comparison is known as a Condorcet winner. A voting rule that satisfies Condorcet's criterion is said to be Condorcet consistent. Condorcet consistency is still one of the main yardsticks by which social choice theorists judge different voting rules. As an exercise, the reader is invited to verify that plurality, the voting rule most of us use in political elections—where each voter awards one point to his top-ranked candidate and the candidate with most points wins—is not Condorcet consistent.

So far computer science has not played a role in our story, but as we shall see much of the work described within can be traced (directly or indirectly) back to Condorcet.

THE 19TH CENTURY

Our next protagonist is Charles Lutwidge Dodgson. Dodgson was born in 1832 as the third of 11 children. He grew up to be a professor of mathematics at Oxford University. He is actually not considered a great mathematician, but he was a Renaissance man with a passion for photography (which was just beginning to emerge) and writing. As a writer he preferred to use a pseudonym, which he obtained by deleting his last name, swapping his first and middle names, and then altering them slightly. That is how Dodgson came up with the name Lewis Carroll, readily recognized as the author of *Alice's Adventures in Wonderland*.

In his first pamphlet on voting—"A Discussion of the Various Methods of Procedure in Conducting Elections"—Dodgson seemingly plagiarized many of the ideas of his predecessors, including Condorcet. Fortunately, there is evidence that indicates Dodgson independently came up with these ideas. In particular, Condorcet's book was not in the collection of the library of the College of Christ's Church, the Oxford college that employed Dodgson. That book was actually available in Oxford's main library, but some of the pages in the relevant section were uncut (at the time books were printed and bound by printing several pages on a large sheet of paper that was then folded and bound into the book, therefore it was necessary to cut pages in order to read). It is unlikely that Dodgson obtained the book anywhere else, and therefore we can conclude that he did not read it.

Lewis Carroll is less known for his work on voting methods.

Two pamphlets later, Dodgson was already innovating on top of his predecessors' work. A third pamphlet, written in 1876, introduced a new voting rule that still draws interest today. Dodgson, like Condorcet, argued in favor of selecting what is now known as a Condorcet winner when one exists in the given preferences. The question is what to do when given preferences do not give rise to a Condorcet winner. Dodgson suggested a simple solution: Choose a candidate that is "closest" to being a Condorcet winner. In fact, he implicitly supplied a concrete notion of distance to being a Condorcet winner. Imagine that we are allowed to swap adjacent candidates in the given preferences of the voters. For example, if a voter has the ranking $a > b > c$, we may swap b and c to obtain the ranking $a > c > b$. The distance of a candidate from being a Condorcet winner is then the minimum number of swaps one must perform in the given preferences to make that candidate a Condorcet winner.

As another example, say that the preferences of voter 1 are represented by the ranking $a > b > c$, those of voter 2 are $b > a > c$, and voter 3 has the ranking $a > c > b$. Here a is a Condorcet winner and so his distance according to Dodgson is 0. A single swap between a and b in the preferences of voter 1 is sufficient to make b a Condorcet winner. Candidate c is weaker and requires three swaps, e.g., to push him to the top of the preferences of voters 1 and 3.

It is finally time to put on my computer science hat. Does Dodgson's rule sound complicated? It turns out that it is provably so. In an important paper written more than a century after Dodgson's third pamphlet, Bartholdi et al. showed the problem of deciding whether a given candidate would be selected by Dodgson's rule in a given voting situation is computationally hard [2]. This means to resolve some elections governed by Dodgson's voting rule we can expect to wait a long time indeed. Naturally, computational hardness is a huge barrier against employing a voting rule; no one wants to wait a billion years to find out who won the presidency! Dodgson's rule is by no means the only well-known voting rule to hold the dubious distinction of being computationally hard (although it certainly has the most interesting history); others include rules by Young, Kemeny, and Slater.

But not all hope is lost for Dodgson and his friends, as computer scientists have long successfully tackled computationally hard problems. One common approach is using heuristics. The work of Conitzer et al. provides a nice example [3]. They focused on computing Kemeny's rule, which, in brief, ranks the candidates in an order that minimizes the total number of disagreements with voters about the relative ranking of pairs of candidates. Although the problem is computationally hard, they gave practical heuristics for solving it. So practical, in fact, that Conitzer's current employer, Duke University's computer science department, uses Kemeny's rule to rank hundreds of Ph.D. applicants.

Alternatively, one can take a theoretical approach. I will just mention that one possible approach relies on parameterized complexity [4]. Another common theoretical approach for dealing with hard combinatorial optimization problems turns to the notion of approximation. In the context of minimization problems, the approximation ratio given by the algorithm is the worst-case ratio between the cost of the algorithm's solution and the cost of the optimal solution. The problem of computing a candidate's Dodgson distance is a minimization problem that is amenable to this type of analysis. So, a natural suggestion is to approximate the Dodgson distance in polynomial time, and select a candidate closest to being a Condorcet winner according to the approximation algorithm. In this sense, one can view approximation algorithms as new voting rules, and seek approximation algorithms that are desirable from a social choice point of view [5]. This approach hybridizes computer science and social choice in an attempt to rethink, and ultimately influence, the methods that we use for collective decision making.

THE 20TH CENTURY

During the first half of the 20th century there was no major progress in social choice theory. It wasn't until 1949 when Kenneth Arrow proved his famous impossibility theorem and thereby reshaped the field. Taking an approach that in a sense can be traced back to Condorcet, Arrow suggested several basic properties that one would want to see in a voting rule. His shocking theorem established that there are no voting rules that satisfy these properties. The ingenuity of Arrow's theorem lies in providing an axiomatic framework that captures all voting rules simultaneously, rather than studying voting rules one by one (to determine, e.g., whether they are Condorcet consistent). For the last 62 years, social choice theorists have largely followed Arrow's lead. Arrow received a Nobel Memorial Prize in Economic Sciences in 1972.

It is becoming apparent that computer scientists have a good shot at being the ones to ultimately eliminate manipulation in elections.

In parallel, the mid 20th century has seen the rise of the field of game theory, through the work of von Neumann and Morgenstern, and then John Nash. Very generally speaking, game theory attempts to model and analyze rational interaction between different entities. The subfield of mechanism design, which started out in the '60s, is concerned with designing the rules of interaction so that rational entities will behave in a desirable way (the designer is often interested in incentivizing these entities to truthfully report their private information).

By the 1970s social choice theory and game theory were mature enough to give rise to the Gibbard-Satterthwaite Theorem; in fact the timing was so right that Gibbard and Satterthwaite proved the theorem independently! The theorem is concerned with the phenomenon of manipulation in elections. To see an example, consider a setting with three voters and four candidates. We will use the voting rule suggested by the chevalier de Borda, a contemporary of Condorcet. In our setting, under the so-called Borda count each voter gives three points to his top candidate, two points to the second place, one point to the third, and zero to the least preferred candidate; the candidate with most points, summed across voters, wins. Assume that voter 1 holds the ranking $b > a > c > d$, voter 2 holds the same ranking, and the preferences of voter 3 are $a > b > c > d$. Borda count awards eight points to b and seven points to a, and therefore (noting that other candidates are clearly weaker) b is the winner. What happens though if voter 3 reports the preferences $a > c > d > b$? In that case a still has seven points, but b receives only six, and a becomes the winner of the election. Because a is preferred to b according to the original preferences of voter 3, this voter can obtain a better outcome by lying about his preferences.

When asked about this phenomenon, Borda famously replied that his scheme is "intended only for honest men." Most of us are not as optimistic as Borda though. One may also argue voters usually vote using secret ballots, and the lack of information would prevent potential manipulators from lying about their preferences. However, many of us do vote strategically in political elections by not supporting favorite candidates who lose in the polls. It is therefore natural to take a worst-case approach by asking whether there are voting rules that are immune to strategic manipulation, even if the manipulator has full information about others' votes; this property is known as "strategyproofness."

The Gibbard-Satterthwaite Theorem answers this question in the negative, by adopting Arrow's approach. The theorem asserts that any strategyproof voting rule where at least three candidates have some chance of being selected must be a dictatorship, in the sense that the rule always selects the most preferred candidate of some fixed voter (known as the dictator), regardless of what other voters want. Note that a dictatorship is itself strategyproof: The dictator cannot gain from lying because he always gets what he wants, whereas other voters cannot gain from lying because their preferences are in any case ignored by the voting rule. (This can be interpreted as implying that dictatorship is the only good form of government!)

Two decades later, in another seminal 1989 paper, Bartholdi et al. suggested that computational complexity can serve as an obstacle against manipulation in elections [6]. Indeed, they argued, the Gibbard-Satterthwaite Theorem tells us that there are voting situations in which voters can gain from lying about their preferences. Nevertheless, perhaps it is possible to find voting rules that would make it difficult for a voter to compute such a beneficial lie. Interestingly, while in the previous section computational hardness was bad (because it made it difficult to use Dodgson's rule), here it is actually good (serving as a shield against an undesirable phenomenon).

As it turns out, several prominent voting rules are computationally hard to manipulate. One well-known example is single transferable vote (STV), which works as follows. The election proceeds in rounds, where in the first round each voter casts a vote for his most preferred candidate. The candidate with the least votes is then eliminated. In the second round, each voter casts a vote for the most preferred candidate among the surviving candidates (hence, the voters who liked the eliminated candidate best in the first round transfer their vote to their second choice). STV continues eliminating candidates, until only one is left standing. This voting rule is actually used in elections for parliament in Australia and in municipal elections in Cambridge, MA. In addition, most prominent voting rules are known to be hard to manipulate when there is a coalition of manipulators, and there is much research about related questions, such as the complexity of successfully bribing voters [7, 8].

As its founders themselves acknowledged, this approach suffers from a serious flaw. The standard notion of computational hardness is worst case, and in particular it is possible that only a small fraction of the instances of a hard problem are hard. In the context of manipulation, this means that even though a voting rule is hard to manipulate in the worst case, it may be that voters are almost always able to efficiently compute beneficial manipulations. What we would really want is a voting rule that is hard to manipulate in an average, perhaps cryptographic, sense [9]. It is not yet known which voting rules, if any, have this property. Recent results suggest that such a voting rule would have to be randomized, but the jury is still out [10].

At the same time, computer scientists are proposing additional innovative methods of circumventing the Gibbard-Satterthwaite Theorem. An especially intriguing recent paper by Birrell and Pass suggests common voting rules can be well-approximated by randomized voting rules that are almost strategyproof [11]. As we explore existing approaches and invent new ones, it is becoming apparent that computer scientists have a good shot at being the ones to ultimately eliminate manipulation in elections.

THE 21ST CENTURY

So what does the future hold in store for computational social choice? I see two main ways in which the field will continue to grow and ultimately have a great impact on both computer science and social choice.

First, much of the literature on social choice is negative in nature. Following Arrow, Gibbard, and Satterthwaite, social choice theorists often suggest a set of desirable properties and then prove that no voting rule satisfies these properties simultaneously. In contrast, computational social choice is constructive in nature, in that researchers use computational thinking to overcome barriers (such as the Gibbard-Sattherthwaite Theorem) and obtain positive results.

Second, the principles of social choice theory are difficult to test in political elections, because changing the voting rule is almost impossible. The recent United Kingdom alternative vote referendum, held on May 5, 2011, is a fine example. Although most experts agreed the alternative voting rule is superior, it was perceived as helping Nick Clegg, the unpopular leader of the Liberal Democrats, and the proposal to replace the voting rule was rejected by 68 percent of the voters. (One may wonder whether the voting system used to decide whether to change the voting system should be changed, but in this case there were only two alternatives so majority voting is essentially perfect.)

In contrast, some computer science environments make it easy to change the voting rule. There is much talk among artificial intelligence researchers about employing social choice for aggregating preferences of autonomous software agents interacting in a multi-agent system (e.g., agents acting on behalf of suppliers, manufacturers, and retailers in a supply chain). In such a system the designer is free to experiment with different voting rules. Moreover, human computation systems—where a computer outsources some computational steps to humans—like EteRNA (http://eterna.cmu.edu) use voting today in order to aggregate noisy information, but this is done in an unprincipled way. Once again, the designer is at liberty to employ any voting rule his heart desires. A great challenge for the future is leveraging social choice theory to improve the way voting is used in human computation systems, and at the same time rethinking some of the foundations of social choice theory to better understand these systems.

Acknowledgment

This work is supported by a gift from Google, and by a Yahoo! Academic Career Enhancement Award.

Biography

Ariel Procaccia is an assistant professor in the Computer Science Department at Carnegie Mellon University. He received his Ph.D. from the Hebrew University of Jerusalem and was subsequently a postdoc at Microsoft and Harvard University.

References

[1] Szpiro, G.G. *Numbers Rule: The Vexing Mathematics of Democracy, from Plato to the Present*. Princeton University Press, Princeton, NJ, 2010.

[2] Bartholdi, J., Tovey, C. A., and Trick, M. A. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare* 6,2 (1989), 157-165.

[3] Conitzer, V., Davenport, A, and Kalagnanam, H. Improved bounds for computing Kemeny rankings. In *Proceedings of the 21st AAAI Conference on Artificial Intelligence* (Boston, MA, July 2006), 620-626.

[4] Betzler, N., Guo, J., and Niedermeier, R. Parameterized computational complexity of Dodgson and Young elections. *Information and Computation* 208,2 (2010), 165-177.

[5] Caragiannis, I., Kaklamanis, C., Karanikolas, N. and Procaccia, A. D. Socially desirable approximations for Dodgson's voting rule. In *Proceedings of the 11th ACM Conference on Electronic Commerce* (Cambridge, MA, June 2010), 253-262.

[6] Bartholdi, J., Tovey, C. A., and Trick, M. A. The computational difficulty of manipulating an election. *Social Choice and Welfare* 6,2 (1989), 227-241.

[7] Conitzer, V., Sandholm, T., and Lang, J. When are elections with few candidates hard to manipulate? *Journal of the ACM* 54,3 (2007), 1-33.

[8] Faliszewski, P., Hemaspaandra, E., and Hemaspaandra, L. How hard is bribery in elections? *Journal of Artificial Intelligence Research* 35,1 (2009), 485-532.

[9] Procaccia, A. D. and Rosenschein, J. S. Junta distributions and the average-case complexity of manipulating elections. *Journal of Artificial Intelligence Research* 28 (Jan-April 2007), 157-181.

[10] Isaksson, M., Kindler, G., and Mossel, E. The geometry of manipulation-a quantitative proof of the Gibbard Satterthwaite Theorem. In *Proceedings of the 51st Symposium on Foundations of Computer Science* (Las Vegas, NV, Oct. 2010), 319-328.

[11] Birrell, E. and Pass, R. Approximately strategy-proof voting. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence* (Barcelona, Spain, July 2011), 67-72.