Automated Reasoning for Social Choice Theory

Ulle Endriss
Institute for Logic, Language and Computation
University of Amsterdam

[Guest Lecture for "Logic, Language and Computation"
Master of Logic, 5 December 2022]

Social Choice Theory

Social choice theory deals with preference aggregation. Example:

Voter 1: $\triangle \succ \bigcirc \succ \square$

Voter 2: $\bigcirc \succ \square \succ \triangle$

Voter 3: $\square \succ \triangle \succ \bigcirc$

Voter 4: $\square \succ \triangle \succ \bigcirc$

Voter 5: $\bigcirc \succ \square \succ \triangle$

?

Outline

- Examples for Voting Rules
- The Problem of Strategic Manipulation
- Formal Analysis of the Problem
- Classical Result: The Gibbard-Satterthwaite Theorem
- Modern Technique: SAT Solving for Theorem Proving in SCT

Three Voting Rules

Suppose n *voters* choose from a set of m *alternatives* by stating their individual preferences as *linear orders* over the alternatives.

How do we decide which alternative to select as the collective choice?

Here are three *voting rules* we might use:

- *Plurality:* elect the alternative ranked first most often
- *Plurality with runoff:* run a plurality election and retain the two front-runners; then run a majority contest between them
- *Borda:* award $m-k$ points to an alternative for every voter who ranks it in the kth position (highest score wins)

Exercise: *Do you know real-world elections where these rules are used?*

Example: Choosing a Beverage for Lunch

Consider this scenario, with nine *voters* having to choose from three *alternatives* (namely what beverage to order for a common lunch):

2 Germans:	Beer	$\succ$	Wine	$\succ$	Milk
3 French people:	Wine	$\succ$	Beer	$\succ$	Milk
4 Dutch people:	Milk	$\succ$	Beer	$\succ$	Wine

Recall that we saw three different voting rules:

- Plurality
- Plurality with runoff
- Borda

Exercise: *For each of the rules, which beverage wins the election?*

More Voting Rules

There are many more voting rules. Examples:

- *Positional scoring rules* other than Borda, with scoring vectors other than $(m-1, m-2, \ldots, 0)$
- *Single transferable vote* (also known as *instant runoff*), where we eliminate plurality losers until there is a majority winner
- *Slater rule*, where we elect the top alternative in the linear order "closest" to the graph obtained via the pairwise majority rule

W.S. Zwicker. Introduction to the Theory of Voting. In F. Brandt *et al.* (eds.), *Handbook of Computational Social Choice*. Cambridge University Press, 2016.

Strategic Manipulation

Recall that under the *plurality rule* (used in most political elections) the candidate ranked first most often wins the election.

Assume the preferences of the people in, say, Florida are as follows:

49%:	Bush $\succ$ Gore $\succ$ Nader
20%:	Gore $\succ$ Nader $\succ$ Bush
20%:	Gore $\succ$ Bush $\succ$ Nader
11%:	Nader $\succ$ Gore $\succ$ Bush

So even if nobody is cheating, Bush will win this election.

> It would have been in the interest of the Nader supporters to *manipulate*, i.e., to misrepresent their preferences.

Exercise: *Is there a better voting rule that avoids this problem?*

The Gibbard-Satterthwaite Theorem

Answer to the previous question: *No!* — surprisingly, not only the plurality rule, but *all* "reasonable" rules have this problem.

Gibbard-Satterthwaite Theorem. *All resolute and surjective voting rules for $\geqslant 3$ alternatives are either manipulable or dictatorial.*

Meaning of the terms mentioned in the theorem:

- *resolute* = the rule always returns a single winner (no ties)
- *surjective* = each alternative can win for *some* way of voting
- *dictatorial* = the top alternative of some fixed voter always wins

Exercise: *How many different rules are there that are dictatorial?*

A. Gibbard. Manipulation of Voting Schemes. *Econometrica*, 1973.

M.A. Satterthwaite. Strategy-proofness and Arrow's Conditions. *Journal of Economic Theory*, 1975.

Formal Model of Voting

Fix a finite set $X = \{a, b, c, \ldots\}$ of *alternatives*, with $|X| = m \geqslant 2$.

Let $\mathcal{L}(X)$ denote the set of all strict linear orders R on X. We use elements of $\mathcal{L}(X)$ to model (true) *preferences* and (declared) *ballots*.

Each member i of a finite set $N = \{1, \ldots, n\}$ of *voters* supplies us with a ballot R_i, giving rise to a *profile* $\boldsymbol{R} = (R_1, \ldots, R_n) \in \mathcal{L}(X)^n$.

Today we restrict attention to *voting rules* that are *resolute* (no ties):

$$F : \mathcal{L}(X)^n \to X$$

Exercise: *How to adapt this definition for arbitrary voting rules?*

Axioms

Axioms are normatively desirable properties of voting rules. Examples:

- F is *anonymous* if $F(R_1, \ldots, R_n) = F(R_{\sigma(1)}, \ldots, R_{\sigma(n)})$ for every profile $\boldsymbol{R} = (R_1, \ldots, R_n)$ and permutation $\sigma : N \to N$.
- F satisfies the *Pareto Principle* if all voters ranking alternative x above alternative y in profile $\boldsymbol{R}$ implies $F(\boldsymbol{R}) \neq y$.
- F is *surjective* if for every alternative $x \in X$ there is a profile $\boldsymbol{R}$ such that $F(\boldsymbol{R}) = x$. So no x is excluded from winning *a priori*.
- F is *strategyproof* (or: immune to manipulation) if for no $i \in N$ there are a profile $\boldsymbol{R}$ (including the "truthful preference" R_i of i) and a ranking R_i' (an "untruthful" ballot of i) such that:

  $$F(R_i', \boldsymbol{R}_{-i}) \text{ is ranked above } F(\boldsymbol{R}) \text{ according to } R_i$$

  Here $(R_i', \boldsymbol{R}_{-i})$ is what we get when in $\boldsymbol{R}$ we replace R_i by R_i'.

The Gibbard-Satterthwaite Theorem

Once more, now stated more clearly as an *impossibility theorem:*

Gibbard-Satterthwaite Theorem. *There exists no resolute rule for $\geqslant 3$ alternatives that is surjective, strategyproof, and nondictatorial.*

Exercise: *Show that the theorem does not hold for $m = 2$ alternatives.*

Exercise: *Show that the theorem is trivially true for $n = 1$ voter.*

A. Gibbard. Manipulation of Voting Schemes. *Econometrica*, 1973.

M.A. Satterthwaite. Strategy-proofness and Arrow's Conditions. *Journal of Economic Theory*, 1975.

Proving G-S

G-S is a deep result that long proved elusive:

- People tried and failed to design strategyproof rules for centuries.
- After Arrow's seminal impossibility theorem (for different axioms) a result *à la* G-S seemed to be "in the air".
- It still took two decades to find the right formulation and prove it.
- The original proofs are hard to digest (the original proof of Arrow's impossibility even was wrong—though not the theorem itself).

Today the proof of G-S is well understood (see expository paper below). But new results of this kind are still hard to identify and then prove.

K.J. Arrow. *Social Choice and Individual Values*. John Wiley and Sons, 2nd edition, 1963. First edition published in 1951.

U. Endriss. Logic and Social Choice Theory. In A. Gupta and J. van Benthem (eds.), *Logic and Philosophy Today*. College Publications, 2011.

Automated Reasoning for Social Choice Theory

Thus: *Need much better methodology to reason about social choice!*

Maybe *automated reasoning*, as studied in AI, can help? *Yes!*

In particular, *SAT solvers* have been used successfully to prove a wide range of (impossibility) theorems in SCT (and related areas):

- automated *verification* of classical results
- automated *proofs* of new theorems
- automated *discovery* of new theorems

A. Biere, M. Heule, H. van Maaren, and T. Walsh (eds.). *Handbook of Satisfiability*. IOS Press, 2nd edition, 2021.

P. Tang and F. Lin. Computer-aided Proofs of Arrow's and other Impossibility Theorems. *Artificial Intelligence*, 2009.

C. Geist and D. Peters. Computer-Aided Methods for Social Choice Theory. In U. Endriss (ed.), *Trends in Computational Social Choice*. AI Access, 2017.

Base Case

The smallest nontrivial case of G-S is that of $n = 2$ voters and $m = 3$ alternatives. If we can prove it, larger cases will be unsurprising.

Idea: Go through all resolute rules for $n = 2$ and $m = 3$ and check.

Exercise: *How many such rules are there?*

Outline of the Approach

There are $3^{3! \times 3!} = 150094635296999121 \approx 1.5 \times 10^{17}$ resolute rules.
Too many to check. Need a better approach. *Logic to the rescue!*

To prove the G-S Theorem for the "base case" of $n = 2$ and $m = 3$:

- express the requirements on F in logic
- show that the resulting formula is not satisfiable

If we can express our requirements in *propositional (boolean) logic*, then we can use (very efficient!) *SAT solvers* for the second step.

Describing Voting Rules in Logic

Consider the propositional (boolean) language with this set of variables:

$$\{\, p_{\boldsymbol{R},x} \mid \boldsymbol{R} \in \mathcal{L}(X)^n \text{ and } x \in X \,\}$$

Intuition: Variable $p_{\boldsymbol{R},x}$ is *true* iff we elect alternative x in profile $\boldsymbol{R}$.

Exercise: *How many variables for $n = 2$ voters and $m = 3$ alternatives?*

Now assignments of truth values to variables correspond to voting rules.

Exercise: *This is almost true, but not quite. What is the problem?*

Voting Rules as Truth Assignments

Not every possible truth assignment corresponds to a voting rule.

We need to ensure at *least one alternative* is elected in each profile:

$$p_{\boldsymbol{R},a_1} \vee p_{\boldsymbol{R},a_2} \vee \cdots \vee p_{\boldsymbol{R},a_m} \quad \text{(for all profiles } \boldsymbol{R}\text{)}$$

We also need to ensure *at most one alternative* is elected:

$$\neg(p_{\boldsymbol{R},x} \wedge p_{\boldsymbol{R},y}) \quad \text{(for all profiles } \boldsymbol{R} \text{ and alternatives } x \neq y\text{)}$$

If φ_{rule} is the conjunction of all of these formulas, then there is a direct correspondence between models of φ_{rule} and resolute voting rules.

Axioms as Formulas

We now can add to our requirements by expressing axioms as formulas.

Here is the formula for *strategyproofness*:

$$\varphi_{\mathsf{sp}} = \bigwedge_{i \in N} \left(\bigwedge_{\boldsymbol{R} \in \mathcal{L}(X)^n} \left(\bigwedge_{\substack{\boldsymbol{R}' \in \mathcal{L}(X)^n \\ \text{s.t. } \boldsymbol{R} =_{-i} \boldsymbol{R}'}} \left(\bigwedge_{x \in X} \left(\bigwedge_{\substack{y \in X \text{ s.t.} \\ x \succ y \text{ in } R_i}} \neg \left(p_{\boldsymbol{R},y} \wedge p_{\boldsymbol{R}',x} \right) \right) \right) \right) \right)$$

<u>Exercise:</u> *Understand the encoding! (Hint: $\boldsymbol{R}$ is the truthful profile.)*

Script to Generate the Master Formula

We need to determine whether the "*master formula*" is satisfiable:

$$\varphi \;=\; \varphi_{\textsf{rule}} \wedge \varphi_{\textsf{sp}} \wedge \varphi_{\textsf{sur}} \wedge \varphi_{\textsf{nd}}$$

<u>Aside:</u> φ is a conjunction of 1,445 clauses (over 108 variables).

Using the so-called *DIMACS format*, we can represent any given formula in CNF on the computer as a list of lists of integers.

<u>Example:</u> `[[1,-2,3],[-1,4]]` represents $(p \vee \neg q \vee r) \wedge (\neg p \vee s)$.

We omit all details, but it is clear that *writing a script* (say, in Python) to generate this representation of our master formula is possible.

Running the SAT Solver

We now can run the SAT solver on our master formula φ ...

Through a Python interface, it will looks something like this:

```
>>> cnf = cnfRule() + cnfSP() + cnfSur() + cnfND()
>>> solve(cnf)
'UNSAT'
```

So φ really is unsatisfiable! Thus, G-S for $n = 2$ and $m = 3$ is true! ✓

Discussion: *Does this count? Do we believe in computer proofs?*

Missing Pieces

We can proof-read our *Python script* just like we would proof-read a mathematical proof. And we can use multiple *SAT solvers* and check they agree. So we can have some confidence in the result.

But some pieces are still missing:

- *Why does the theorem hold?* This proof does not tell us.

  But SAT technology can help here as well: *MUS extraction*

- *Does the theorem generalise to arbitrary $n \geqslant 2$ and $m \geqslant 3$?*

  Intuitively almost obvious, though technically not that easy.
  Basic idea: *induction* over both n and m

Summary and Opportunities

We saw that no good voting rule can avoid strategic behaviour, and we saw how to prove this impossibility theorem using SAT solvers.

Nice example for research that integrates ideas from *economics* (here: axiomatic method) and *computer science* (here: automated reasoning).

Well represented in the MoL programme. Ideal trajectory:

Economics

Game Theory

Computer Science

Knowledge Representation

Computational Complexity

Algorithmic Game Theory

Computational Social Choice

Also: Computational Social Choice Seminar