Collective Decision Making in Combinatorial Domains

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

Introduction to *computational social choice*, with some examples:

- *Logical modelling* of social choice problems
- *Computational complexity* of strategic behaviour in elections
- Choosing from huge numbers of alternatives (*combinatorial domains*)
Expert 1: $\bigcirc \succ \bigcirc \succ \bigcirc$
Expert 2: $\bigcirc \succ \bigcirc \succ \bigcirc$
Expert 3: $\bigcirc \succ \bigcirc \succ \bigcirc$
Expert 4: $\bigcirc \succ \bigcirc \succ \bigcirc$
Expert 5: $\bigcirc \succ \bigcirc \succ \bigcirc$

?
Social Choice and the Condorcet Paradox

*Social Choice Theory* asks: how should we aggregate the preferences of the members of a group to obtain a “social preference”?

Expert 1: $\bigcirc \succ \bigcirc \succ \bigcirc$
Expert 2: $\bigcirc \succ \bigcirc \succ \bigcirc$
Expert 3: $\bigcirc \succ \bigcirc \succ \bigcirc$
Expert 4: $\bigcirc \succ \bigcirc \succ \bigcirc$
Expert 5: $\bigcirc \succ \bigcirc \succ \bigcirc$

Marie Jean Antoine Nicolas de Caritat (1743–1794), better known as the **Marquis de Condorcet**: Highly influential Mathematician, Philosopher, Political Scientist, Political Activist. Observed that the *majority rule* may produce inconsistent outcomes (“Condorcet Paradox”).
A Classic: Arrow’s Impossibility Theorem

In 1951, K.J. Arrow published his famous *Impossibility Theorem*:

Any preference aggregation mechanism for *three* or more alternatives that satisfies the axioms of *unanimity* and *IIA* must be *dictatorial*.

- **Unanimity**: if everyone says $A \succ B$, then so should society.
- **Independence of Irrelevant Alternatives (IIA)**: if society says $A \succ B$ and someone changes their ranking of $C$, then society should still say $A \succ B$.

Social Choice and Computer Science

Social choice theory has natural applications in computer science:

- **Search Engines**: to determine the most important sites based on links ("votes") + to aggregate the output of several search engines
- **Recommender Systems**: to recommend a product to a user based on earlier ratings by other users
- **Multiagent Systems**: to aggregate the beliefs + to coordinate the actions of groups of autonomous software agents

*Vice versa*, techniques from computer science are useful for advancing the state of the art in social choice theory . . .

Logical Modelling

What kind of features do we need in a logic to be able to reason about problems in social choice?

Example for a result:

**Theorem 1** *The first-order theory $T_{\text{ARROW}}$ has no finite model.*

Also of interest:

- use of automated theorem provers to confirm results
- automated search for new results with variants of axioms
- model checking to assess concrete algorithms for voting rules

Example: Strategic Manipulation

Remember Florida 2000 (simplified):

<table>
<thead>
<tr>
<th>Percentage</th>
<th>Preferences</th>
</tr>
</thead>
<tbody>
<tr>
<td>49%</td>
<td>Bush ≻ Gore ≻ Nader</td>
</tr>
<tr>
<td>20%</td>
<td>Gore ≻ Nader ≻ Bush</td>
</tr>
<tr>
<td>20%</td>
<td>Gore ≻ Bush ≻ Nader</td>
</tr>
<tr>
<td>11%</td>
<td>Nader ≻ Gore ≻ Bush</td>
</tr>
</tbody>
</table>

Questions:

- Who wins?
- What would your advice to the Nader-supporters have been?
Complexity as a Barrier against Manipulation

By the classical *Gibbard-Satterthwaite Theorem*, any voting rule for \( \geq 3 \) candidates can be manipulated (unless it is dictatorial).

**Idea:** So it’s always *possible* to manipulate, but maybe it’s *difficult*!

Tools from *complexity theory* can be used to make this idea precise.

- For *some* procedures this does *not* work: if I know all other ballots and want \( X \) to win, it is *easy* to compute my best strategy.
- But for *others* it does work: manipulation is *NP-complete*.

Recent work in COMSOC has expanded on this idea:

- NP is a worst-case notion. What about average complexity?
- Also: complexity of winner determination, control, bribery, . . .


Multi-issue Elections

Suppose 13 voters are asked to each vote yes or no on three issues; and we use the simple majority rule for each issue independently:

- 3 voters each vote for YNN, NYN, NNY.
- 1 voter each votes for YYY, YYN, YNY, NYY.
- No voter votes for NNN.

But then NNN wins! (on each issue, 7 out of 13 vote no)

What to do instead? The number of candidates is exponential in the number of issues (e.g., $2^3 = 8$), so even just representing the voters’ preferences is a challenge (knowledge representation).


Paradox?
<table>
<thead>
<tr>
<th>Judge 1:</th>
<th>p</th>
<th>p → q</th>
<th>q</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>True</td>
<td>True</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Judge 2:</th>
<th>p</th>
<th>p → q</th>
<th>q</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>False</td>
<td>False</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Judge 3:</th>
<th>p</th>
<th>p → q</th>
<th>q</th>
</tr>
</thead>
<tbody>
<tr>
<td>False</td>
<td>True</td>
<td>False</td>
<td></td>
</tr>
<tr>
<td>Voter</td>
<td>fund museum?</td>
<td>fund school?</td>
<td>fund metro?</td>
</tr>
<tr>
<td>-------</td>
<td>--------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Voter 1:</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Voter 2:</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Voter 3:</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

? 

[Constraint: we have money for \textit{at most two projects}]

Ulle Endriss
General Perspective

We can view many of our problems as problems of *binary aggregation*:

- *Do you rank option ○ above option □?* Yes/No
- *Do you believe formula “p → q” is true?* Yes/No
- *Do you want the new school to get funded?* Yes/No

Each problem domain comes with its own *integrity constraints*:

- *Rankings should be transitive and not have any cycles.*
- *The accepted set of formulas should be logically consistent.*
- *We should fund at most two projects.*

The *paradoxes* we have seen show that the *majority rule* does not *lift* our integrity constraints from the *individual* to the *collective* level.
Characterisation Results

So: Which aggregation rules lift which integrity constraints?

Example for a result:

**Theorem 2** An aggregator $F$ will lift all integrity constraints that can be expressed as a conjunction of literals if and only if $F$ is unanimous.

Can we avoid all paradoxes?

That is: Are the aggregators that lift all integrity constraints? Yes!

**Theorem 3** An aggregator $F$ will lift all integrity constraints if and only if $F$ is a generalised dictatorship (that is, if $F$ is defined by a function $g$ from profiles to agents via $F(B_1, \ldots, B_n) = B_{g(B_1, \ldots, B_n)}$).

This includes some pretty bad aggregators:

- proper (Arrovian) **dictatorships**: $g \equiv i$ (dictator fixed in advance)

And some that look at least interesting:

- return the individual vector closest to the **majority** vector
- return the individual vector closest to the **average** vector

Voting as Choosing the Most Representative Voter

Somewhat surprisingly, this *majority-voter rule* and *average-voter rule* have excellent properties:

- *no paradoxes* (outcomes are always consistent)
- *low complexity* (MVP slightly lower than AVP)
- *2-approximations* of the (intractable) distance-based rule returning the consistent vector closest to the profile (AVP slightly better)
- satisfaction of *choice-theoretic axioms* (except for independence): anonymity, neutrality, unanimity (MVP also reinforcement)

That is, our method of seeking to characterise aggregators via the IC’s they lift has helped to identify useful practical methods . . .

Last Slide

I have tried to offer a glimpse at computational social choice.

Examples discussed:

- *logical modelling* in social choice (Arrow’s Theorem in FOL)
- *computational hardness* as a barrier against strategic behaviour
- *choice-theoretic* and *algorithmic* challenges in multi-issue elections
- *characterisation* of aggregation rules in terms of the IC’s lifted

COMSOC is a booming field of research with lots of opportunities (and links to your favourite topic in computation yet to be discovered).

To find out more about the field, you could have a look at this website (biannual workshop series, PhD theses, mailing list):

http://www.illc.uva.nl/COMSOC/