Computational Social Choice 2022

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

[http://www.illc.uva.nl/~ulle/teaching/comsoc/2022/]

Plan for Today

Our main goal for today will be to illustrate the power of JA as a modelling tool for a wide range of collective decision making scenarios by showing how to *embed preference aggregation* into JA.

But first we require some more basic machinery:

- more examples for *aggregation rules*
- changing the model: *integrity constraints*
- enriching the model: *rationality* and *feasibility constraints*

Then we'll be ready to get into the main topic:

- the *preference agenda*
- modelling common *voting rules*

Voting Rules

For preferences, lots of voting rules have been suggested. We saw:

- Positional scoring rules: Borda, plurality, veto, k-approval
- Staged rules: plurality with runoff, STV
- Dodgson: elect the alternative closest to being a Condorcet winner

Here are a few more:

- *Copeland:* maximise majority contests won minus those lost
- *Slater:* build majority graph; find closest linear order; elect top
- *Kemeny:* same but now using the weighted majority graph
- *Ranked Pairs:* build linear order by accepting ranked pairs in order of support strength, skipping those that'd lead to cycles; elect top

In contrast, in JA far fewer aggregation rules have been proposed.
Next: some more rules (and later: understanding *why* there are fewer)

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

The Kemeny Rule for Judgment Aggregation

For a profile $\boldsymbol{J} = (J_1, \ldots, J_n)$ and judgment set J, think of $J \cap J_i$ as the agreement between J and agent i, and thus of $|J \cap J_i|$ as her satisfaction with J. The *Kemeny rule* maximise *total satisfaction:*

$$F_{\text{kem}}(\boldsymbol{J}) \;\in\; \operatorname*{argmax}_{J \in \mathcal{J}(\Phi)} \sum_{i \in N} |J \cap J_i| \;=\; \operatorname*{argmax}_{J \in \mathcal{J}(\Phi)} \sum_{\varphi \in J} |N_\varphi^{\boldsymbol{J}}|$$

Note that F_{kem} is *irresolute*. Alternative readings of above definition:

- maximise total *agreement* with the formulas selected
- maximise *average* satisfaction (or agreement)
- minimise cumulative (Hamming) *distance* to the profile

Also known under the name of *distance-based rule* (amongst others).

Exercise: *How would you go about implementing this rule?*

The Slater Rule for Judgment Aggregation

The *Slater rule* maximses the number of *majority-supported* formulas:

$$F_{\mathrm{sla}}(\boldsymbol{J}) \;\in\; \operatorname*{argmax}_{J \in \mathcal{J}(\Phi)} \sum_{\varphi \in J} \mathbb{1}_{|N_\varphi^{\boldsymbol{J}}| > \frac{n}{2}}$$

This rule also is irresolute. So *tie-braking* is required in practice.

Binary Aggregation with Integrity Constraints

Today we work with a simple variant of the formula-based JA model:

The *agenda* Φ includes *only literals*, but it now comes paired with a so-called *integrity constraint* Γ (a propositional formula).

The notion of consistency is refined to Γ-*consistency:*

judgment set $J \subseteq \Phi$ is Γ-consistent if $J \cup \{\Gamma\}$ is consistent

Let $\mathcal{J}(\Phi, \Gamma)$ be the set of all complete and Γ-consistent judgment sets.

We now are interested in aggregation rules $F : \mathcal{J}(\Phi, \Gamma)^n \to 2^\Phi$.

Remark: This model more closely corresponds to how we had first introduced the *doctrinal paradox* (legal doctrine $= \Gamma$).

Exercise: *How to define the Slater and Kemeny rules for this model?*

Exercise: *And how about the majority rule and other quota rules?*

Embedding Formula-Based Judgment Aggregation

Any formula-based JA scenario with agenda Φ can be translated into the new model (using twice the number of agenda items):

- For every $\varphi \in \Phi$ create a positive literal p_φ and its negation.
- Construct an IC as the conjunction of these formulas:
  - encoding completeness: $p_\varphi \vee p_{\neg\varphi}$ for all positive $\varphi \in \Phi$
  - encoding consistency: $\neg \bigwedge_{\varphi \in X} p_\varphi$ for all mi-sets $X \subseteq \Phi$

Exercise: *Explain how this achieves a direct correspondence.*

Discussion: The IC obtained may be very large (exponential in $|\Phi|$). So, though possible, translation is not always a good idea in practice.

Translation in the other direction is also possible (but less obvious).

U. Endriss, U. Grandi, R. de Haan, and J. Lang. Succinctness of Languages for Judgment Aggregation. KR-2016.

Example: Rationality and Feasibility Constraints

So far, we always imposed the same conditions on both individual judgments and collective judgments. *Too restrictive?*

Suppose the five members of a local government council have to decide on whether to approve funding for three community initiatives ...

	School?	**Theatre?**	**Parking?**
Alice	No	No	Yes
Bob	Yes	Yes	Yes
Chris	Yes	No	Yes
Dana	Yes	Yes	No
Eve	No	Yes	No
Majority	Yes	Yes	Yes

Rationality Constraint = "I should support at least one initiative"

Feasibility Constraint = "We cannot afford paying for all initiatives"

Enriching the Model

We enrich the model of binary aggregation with integrity constraints by using two (possibly distinct) constraints:

- *rationality constraint* Γ_{in}: assume $J_i \in \mathcal{J}(\Phi, \Gamma_{\text{in}})$ for all $i \in N$
- *feasibility constraint* Γ_{out}: hope for $F(\boldsymbol{J}) \in \mathcal{J}(\Phi, \Gamma_{\text{out}})$

Exercise: *How does this affect the definition of our rules?*

Discussion: Γ_{out} could be more demanding than Γ_{in}, it could be less demanding, they could be the same, or they could be incomparable.

Embedding Preference Aggregation

In *preference aggregation*, agents express preferences (linear orders) over a set of alternatives X and we need to find a collective preference.

Construct the *preference agenda* $\Phi^X_{\succcurlyeq} = \{p_{x \succcurlyeq y}, \neg p_{x \succcurlyeq y} \mid x, y \in X\}$.

Then construct an *integrity constraint* Γ as the conjunction of:

- Completeness: $p_{x \succcurlyeq y} \vee p_{y \succcurlyeq x}$ for all $x, y \in X$
- Antisymmetry: $\neg(p_{x \succcurlyeq y} \wedge p_{y \succcurlyeq x})$ for all $x, y \in X$ with $x \neq y$
- Transitivity: $p_{x \succcurlyeq y} \wedge p_{y \succcurlyeq z} \rightarrow p_{x \succcurlyeq z}$ for all $x, y, z \in X$

Now we can simulate the *Condorcet paradox:*

	$p_{x \succcurlyeq y}$	$p_{x \succcurlyeq z}$	$p_{y \succcurlyeq z}$	*corresponding order*
Agent 1	Yes	Yes	Yes	$x \succ y \succ z$
Agent 2	No	No	Yes	$y \succ z \succ x$
Agent 3	Yes	No	No	$z \succ x \succ y$
Majority	Yes	No	Yes	*not a linear order*

Challenge: Simulating Voting Rules

Using the preference agenda, simulating a preference aggregation rule that returns a social preference order is fairly straightforward.

The reason is that input and output are of the same type (rankings).

Challenge: Simulate voting rules (where outputs are alternatives).
Idea: Use rationality and feasibility constraints.

J. Lang and M. Slavkovik. Judgment Aggreg. Rules and Voting Rules. ADT-2013.

U. Endriss. JA with Rationality and Feasibility Constraints. AAMAS-2018.

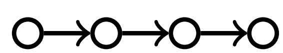
Useful Constraints

Use constraints to describe properties of binary relations:

$$\begin{aligned}
\textsc{Complete} &= \bigwedge_{x,y} (p_{x \succcurlyeq y} \vee p_{y \succcurlyeq x}) \\
\textsc{AntiSym} &= \bigwedge_{x \neq y} \neg (p_{x \succcurlyeq y} \wedge p_{y \succcurlyeq x}) \\
\textsc{Transitive} &= \bigwedge_{x,y,z} (p_{x \succcurlyeq y} \wedge p_{y \succcurlyeq z} \rightarrow p_{x \succcurlyeq z})
\end{aligned}$$

$$\begin{aligned}
\textsc{WeakOrder} &= \textsc{Complete} \wedge \textsc{Transitive} \\
\textsc{Ranking} &= \textsc{WeakOrder} \wedge \textsc{AntiSym}
\end{aligned}$$

Judgment sets satisfying $\textsc{Ranking}$ correspond to relations like this:



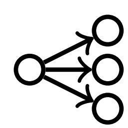
Further Useful Constraints

Let's use $p_{x \succ y}$ as a shorthand for $p_{x \succcurlyeq y} \wedge \neg p_{y \succcurlyeq x}$. More constraints:

$$\begin{aligned}
\textsc{NoChain} &= \bigwedge_{x,y,z} \neg(p_{x \succ y} \wedge p_{y \succ z}) \\
\textsc{Rooted} &= \bigvee_{x} \bigwedge_{y \neq x} p_{x \succ y}
\end{aligned}$$

$$\begin{aligned}
\textsc{Dichotomous} &= \textsc{WeakOrder} \wedge \textsc{NoChain} \\
\textsc{Winner} &= \textsc{Rooted} \wedge \textsc{Dichotomous}
\end{aligned}$$

Judgment sets satisfying $\textsc{Winner}$ correspond to relations like this:



Simulating Voting Rules

Fix a set of alternatives X. Let F be an aggregation rule for $\Phi^X_{\succsim}$.

Fix the rationality constraint $\Gamma_{\text{in}} = \textsc{Ranking}$. Thus: profiles of judgment sets correspond to preference profiles.

For the feasibility constraint Γ_{out} we require $\Gamma_{\text{out}} \models \textsc{Rooted}$. Thus: from any Γ_{out}-consistent output we can extract a winning alternative.

For F guaranteeing Γ_{out}-consistent outputs on Γ_{in}-consistent profiles, we say that F *simulates* voting rule F' if this always works:

- translate the preference profile into a profile of judgment sets
- apply (the possibly irresolute) F to that profile of judgment sets
- extract the "root alternative" from each outcome set
- the set of alternatives extracted should be the winners under F'

Note: $\textsc{Winner} \models \textsc{Rooted}$ but also $\textsc{Ranking} \models \textsc{Rooted}$.

Simulation Results

Getting the proof details right is tricky (and to date has only been done for odd n), but understanding the intuitions is not too difficult:

Rationality	Feasibility	Slater$^{\text{JA}}$	Kemeny$^{\text{JA}}$
$\Gamma_{\text{in}} = \textsc{Ranking}$	$\Gamma_{\text{out}} = \textsc{Ranking}$	Slater	Kemeny
$\Gamma_{\text{in}} = \textsc{Ranking}$	$\Gamma_{\text{out}} = \textsc{Winner}$	Copeland	Borda

Discussion: Maybe this explains the sparsity of rules for JA?

U. Endriss. JA with Rationality and Feasibility Constraints. AAMAS-2018.

Computational Considerations

The judgment aggregation rules discussed in previous lectures were all computationally easy (e.g., counting up to a given quota).

Today's "optimisation-based" rules are different:

- Need to search through *all feasible models* to find the "best" one. There might be *exponentially* many of them.
- The case is similar for the formula-based model: we need to check (possibly) exponentially many *consistent judgment sets*.

  And checking consistency is itself a hard problem.

Summary

We saw a richer model for JA: *rationality* and *feasibility constraints* (however, with agenda items being restricted to literals).

We saw that some specific judgment aggregation rules each can *simulate several voting rules*, if we switch feasibility constraints:

- Slater (for JA) simulates Slater (for preferences) and Copeland
- Kemeny (for JA) simulates Kemeny (for preferences) and Borda

This clarifies connections between different areas of social choice theory and offers deep insights into the nature of aggregation.

What next? Computational complexity of judgment aggregation rules.