Binary Aggregation with Integrity Constraints

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

[ joint work with Umberto Grandi ]
Preference Aggregation

Individual 1: $\triangle \succ o \succ \square$

Individual 2: $\square \succ \triangle \succ o$

Individual 3: $o \succ \square \succ \triangle$

?
Judgment Aggregation

<table>
<thead>
<tr>
<th></th>
<th>$p$</th>
<th>$p \rightarrow q$</th>
<th>$q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Judge 1:</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>Judge 2:</td>
<td>True</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>Judge 3:</td>
<td>False</td>
<td>True</td>
<td>False</td>
</tr>
</tbody>
</table>

?
### Multiple Referenda

<table>
<thead>
<tr>
<th></th>
<th>Fund Museum?</th>
<th>Fund School?</th>
<th>Fund Metro?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Voter 1:</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><strong>Voter 2:</strong></td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Voter 3:</strong></td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

? 

- Constraint: we have money for *at most two projects*
General Perspective

The last example is arguably the clearest. We can rephrase many aggregation 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.
Talk Outline

• Framework: binary aggregation with integrity constraints
• Focus on language used to express IC (\(\sim\) feasible outcomes)
• Idea: characterise aggregators via language of IC’s it can lift
• Applications of that idea
The Model

Basic terminology and notation:

- Finite set of issues $I = \{1, \ldots, m\}$, defining a boolean combinatorial domain $D = D_1 \times \cdots \times D_m$, with $D_i = \{0, 1\}$.

- Each of a finite set of individuals $N = \{1, \ldots, n\}$ votes by supplying a ballot $B_i \in D$. Profile $B = (B_1, \ldots, B_n) \in D^N$.

- A binary aggregator is a function $F : D^N \rightarrow D$.

We can define axioms in the usual manner, possibly restricting their scope to some (feasible) subdomain $X \subseteq D$. Example:

- $F$ is unanimous on $X \subseteq D$, if for any $(B_1, \ldots, B_n) \in X^N$ and any $x \in \{0, 1\}$, if $b_{i,j} = x$ for all $i \in N$, then $F(B_1, \ldots, B_n)_j = x$. 
Integrity Constraints

Rather than defining the subdomain $X \subseteq D$ extensionally, we want to give an intentional characterisation, by means of integrity constraints.

- Introduce a *propositional variable* $p_i$ for each issue $i \in I$ and consider the *propositional language* $\mathcal{L}_{PS}$ over $PS = \{p_1, \ldots, p_m\}$ (closed under $\neg, \wedge, \lor, \rightarrow, \leftrightarrow$).

- Any given *integrity constraint* (formula) $IC \in \mathcal{L}_{PS}$ defines a domain of aggregation $X = \text{Mod}(IC) := \{B \in D \mid B \models IC\}$.

- Ballots are models (truth assignments) for formulas in $\mathcal{L}_{PS}$. Call ballot $B \in D$ *rational* wrt. $IC \in \mathcal{L}_{PS}$ if $B \models IC$.

Recall the three-project example:

\[
IC = \neg(p_1 \wedge p_2 \wedge p_3) = \text{“we cannot afford all three projects”}
\]

Voter 1: $B_1 = (1, 1, 0) \leadsto B_1 \models IC$ (rational)

Majority: $M = (1, 1, 1) \leadsto M \not\models IC$ (irrational)
What’s a paradox?

As a first application, we can give a generic definition of “paradox”:

A paradox is a triple \((F, B, IC)\) consisting of an aggregator \(F\), a profile \(B\), and an integrity constraint \(IC\) such that \(B_i \models IC\) for all \(i \in \mathcal{N}\) but \(F(B) \not\models IC\).

Examples:

• Preference aggregation:
  - \(p_{ab} \leftrightarrow \neg p_{ba}\) for all \(a \neq b\) and \(\neg p_{aa}\) for all \(a\)
  - \(p_{ab} \land p_{bc} \rightarrow p_{ac}\) for all \(a, b, c\)

• Judgment aggregation:
  - \(p_{\varphi} \lor p_{\bar{\varphi}}\) for all complementary \(\varphi, \bar{\varphi}\)
  - \(\neg \bigwedge_{\varphi \in S} p_{\varphi}\) for all minimally inconsistent sets \(S \subseteq \text{AGENDA}\)
Collective Rationality wrt. a Language

Collective rationality wrt. an integrity constraint:

- An aggregator $F$ is \textit{collectively rational} wrt. $IC \in \mathcal{L}_{PS}$ if $B_i \models IC$ for all $i \in \mathcal{N}$ implies $F(B_1, \ldots, B_n) \models IC$ ($F$ can “lift” $IC$).
- Thus: $F$ is CR wrt. $IC \iff \not\exists B$ s.t. $(F, B, IC)$ is a paradox.

Now consider a \textit{language} $\mathcal{L} \subseteq \mathcal{L}_{PS}$ of integrity constraints, e.g.,

- the language of \textit{cubes} (conjunctions of literals),
- the language of \textit{clauses} of length $\leq 2$, etc.

Collective rationality wrt. a language:

- An aggregator $F$ is \textit{collectively rational} wrt. $\mathcal{L} \subseteq \mathcal{L}_{PS}$ if $F$ is collectively rational wrt. every $IC \in \mathcal{L}$.
Template for Results

Two ways of defining classes of aggregators:

- The class of aggregators that lift all integrity constraints in $\mathcal{L}$:

$$CR[\mathcal{L}] := \{ F : \mathcal{D}^N \to \mathcal{D} \mid F \text{ is collectively rational wrt. } \mathcal{L} \}$$

- The class of aggregators defined by a given list of axioms $AX$:

$$\mathcal{F}_\mathcal{L}[AX] := \{ F : \mathcal{D}^N \to \mathcal{D} \mid F \text{ satisfies } AX \text{ on all } \mathcal{L}\text{-domains} \}$$

What we want:

$$CR[\mathcal{L}] = \mathcal{F}_\mathcal{L}[AX]$$
Example for a Characterisation Result

Cubes (\(=\) conjunctions of literals) are lifted by an aggregator \(iff\) that aggregator satisfies \textit{unanimity}:

\[
\mathcal{CR}[\text{cubes}] = \mathcal{F}_{\text{cubes}}[\text{Unanimity}]
\]
More Results

Characterisation results:

- $\mathcal{CR}[p \leftrightarrow q] = \mathcal{F}_\leftrightarrow[\text{Issue-Neutrality}]$
- $\mathcal{CR}[p \text{ XOR } q] = \mathcal{F}_\text{XOR}[\text{Domain-Neutrality}]$

Negative results:

- there exists no language $\mathcal{L}$ such that $\mathcal{CR}[\mathcal{L}] = \mathcal{F}[\text{Anonymity}]$
- there exists no language $\mathcal{L}$ such that $\mathcal{CR}[\mathcal{L}] = \mathcal{F}[\text{Independence}]$

Characterisation within a noncharacterisable class:

- $\mathcal{CR}[k\text{-pclauses}] \cap \mathcal{QR} = \mathcal{QR}[\sum q_i < n + k] \cup \mathcal{QR}[\prod q_i = 0]$

  quoata rules
Application: Preference Aggregation

Call a preference aggregator *imposed* if there exist $x$ and $y$ such that $x$ is collectively preferred to $y$ in every profile. **A theorem:**

*Any anonymous, independent and monotonic preference aggregator for $\geq 3$ alternatives and $\geq 2$ individuals is imposed.*

Proof:

- Adapt Dietrich-List result on quota rules in JA to show that any A-I-M aggregator must be a *quota rule*.

- IC’s for preference aggregation entail two *3-clauses*:

$$p_{ba} \lor p_{cb} \lor p_{ac} \quad p_{ab} \lor p_{bc} \lor p_{ca}$$

- Apply our *lifting theorem* to derive a constraint on the quotas:

$$\sum q_i < n + 3 \quad \text{or} \quad \prod q_i = 0 \ [\Leftrightarrow \text{imposed}]$$

- Rewriting of LHS (and $p_{xy} + p_{yx} = n + 1$) yields contradiction. ✓
Application: Good Binary Aggregators

Is there an aggregator that will lift every integrity constraint? Yes!

\( F \) will lift every IC \( \in \mathcal{L}_{PS} \) iff \( F \) is a generalised dictatorship, i.e., iff there exists a function \( g : \mathcal{D}^{\mathcal{N}} \rightarrow \mathcal{N} \) such that always

\[
F(B_1, \ldots, B_n) = B_{g(B_1, \ldots, B_n)}.
\]

The class of generalised dictatorships includes:

- proper dictatorships \( F_i : (B_1, \ldots, B_n) \mapsto B_i \) for each \( i \in \mathcal{N} \)
- distance-based generalised dictatorships mapping \((B_1, \ldots, B_n)\) to that \(B_i\) that minimises the sum of the Hamming distances to the others (\(+\) tie-breaking). An attractive procedure!
Last Slide

Binary aggregation with integrity constraints:

- *language* to express *rationality assumptions* in binary aggregation
- concept of *collective rationality* with respect to a language
- characterisation results, relating *axioms* and *languages*
- *applications*: preference + judgment aggregation, good procedures

---
