Computational Social Choice: Spring 2015

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

Plan for Today

Our goal for today is to introduce more *specific aggregation rules* for judgment aggregation. We will do this by taking inspiration from (the much older field of) voting theory.

- Definition of an alternative framework for aggregation theory: *binary aggregation with integrity constraints* (easier to work with)
- *Translation* between frameworks: preferences and judgments in BA
- Some *voting rules* and their interpretation in BA

The main references for today, cited below, define the BA framework and introduce the methodology of interpreting voting rules.

U. Grandi and U. Endriss. Binary Aggregation with Integrity Constraints. Proc. 22nd International Joint Conference on Artificial Intelligence (IJCAI-2011).

J. Lang and M. Slavkovik. Judgment Aggregation Rules and Voting Rules. Proc. 3rd International Conference on Algorithmic Decision Theory (ADT-2013).

Binary Aggregation with Integrity Constraints

We will now introduce an alternative framework for aggregation, called *binary aggregation with integrity constraints* (BA).

This may be considered a more direct implementation of the original doctrinal paradox (with the IC taking the role of the doctrine).

Broadly speaking, (formula-based) JA and BA with IC's are equivalent. It depends both on the domain of aggregation to be modelled and the technical result sought, which of the two is the more convenient framework to work in. We will switch between them freely.

U. Grandi and U. Endriss. Binary Aggregation with Integrity Constraints. Proc. 22nd International Joint Conference on Artificial Intelligence (IJCAI-2011).

Formal Framework

Let $\mathcal{I} = \{1, \ldots, m\}$ be a finite set of *issues* on which to take a decision and let $\mathcal{D} := \{0, 1\}^m$ be the binary combinatorial domain defined by $\mathcal{I}$.

A *ballot* is a vector $B = (b_1, \ldots, b_m) \in \mathcal{D}$, indicating for each issue $j \in \mathcal{I}$ whether it is *accepted* $(b_j = 1)$ or *rejected* $(b_j = 0)$.

Associate a set $PS = \{p_1, \ldots, p_m\}$ of *propositional symbols* with $\mathcal{I}$ and let $\mathcal{L}_{PS}$ be the *language* of propositional logic over PS.
Note that *models* for formulas in $\mathcal{L}_{PS}$ are isomorphic to ballots.

An *integrity constraint* (IC) is a formula $\Gamma \in \mathcal{L}_{PS}$. For a given Γ, we say that ballot B is *rational* if $B \in \mathrm{Mod}(\Gamma)$ (that is, if $B \models \Gamma$).

Now a finite set $\mathcal{N} = \{1, \ldots, n\}$ of *individuals*, with $n \geqslant 2$, each report a rational ballot, producing a *profile* $\boldsymbol{B} = (B_1, \ldots, B_n)$.

A (resolute) *aggregation rule* for n individuals, issues $\mathcal{I}$, and IC Γ is a function $F : \mathrm{Mod}(\Gamma)^n \to \mathcal{D}$, mapping profiles to single ballots/models.

Example: Defining the Majority Rule

For an *odd* number n of agents (and any Γ), the strict *majority rule* now is defined as follows:

$$\begin{aligned} F_{\mathsf{maj}} &: \mathrm{Mod}(\Gamma)^n \to \mathcal{D} \\ F_{\mathsf{maj}} &: \boldsymbol{B} \mapsto (\mathbf{1}_{|N^{\boldsymbol{B}}_{1:1}|>\frac{n}{2}}, \ldots, \mathbf{1}_{|N^{\boldsymbol{B}}_{m:1}|>\frac{n}{2}}) \end{aligned}$$

Here $N^{\boldsymbol{B}}_{j:v} := \{i \in \mathcal{N} \mid b_{ij} = v\}$ for $v \in \{0,1\}$ is the set of agents who in profile $\boldsymbol{B}$ assign value v to issue j. (Note: b_{ij} is the jth item in B_i.)

And $\mathbf{1}_E$ is the *characteristic function* for the boolean expression E: $\mathbf{1}_E = 1$ if E is true, and $\mathbf{1}_E = 0$ if E is false.

Discussion: For even n, F_{maj} is biased towards value 0 (not neutral!). And: we get "complement-freeness" and "completeness" for free here. But: cannot define (resolute) weak majority rule for even n.

Embedding Formula-Based Judgment Aggregation

Any formula-based JA problem with an *agenda* Φ of *size* m can be translated into a problem of BA with m *issues*.

Use $\mathcal{I} := \Phi$ (slight abuse of notation). We just need to define the right *integrity constraint* Γ, namely as the conjunction of all these formulas:

- Completeness: $p_\varphi \vee p_{\sim\varphi}$ for all $\varphi \in \Phi$
- Consistency: $\neg \bigwedge_{\varphi \in X} p_\varphi$ for all minimally inconsistent sets $X \subseteq \Phi$

Now $\mathrm{Mod}(\Gamma)$ corresponds directly to $\mathcal{J}(\Phi)$.

<u>Discussion:</u> The IC obtained may he huge. So, though always possible, translation is not always a good idea in practice.

<u>Discussion:</u> An alternative translation would use only $\frac{m}{2}$ issues (corresponding to the positive agenda formulas) and simply force completeness and complement-freeness (w.r.t. Φ) by design.

Aside: Translating Back and Forth

We can also translate back: to see how to translate from BA into formula-based JA (not obvious!), consult Dokow and Holzman (2009).

Discussion: Translation is useful to port results and clarify hidden connections, but arguing which model is "most general" is not fruitful. Use the one that fits your problem most naturally!

E. Dokow and R. Holzman. Aggregation of Binary Evaluations for Truth-functional Agendas. *Social Choice and Welfare*, 32(2):221–241, 2009.

Embedding Preference Aggregation

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

To translate into BA, make every *ordered pair of alternatives* an issue. Write $p_{x \succ y}$ for the prop. symbol corresponding to $(x, y) \in \mathcal{X}^2 = \mathcal{I}$.

Build an *integrity constraint* Γ as the conjunction of:

- Irreflexivity: $\neg p_{x \succ x}$ for all $x \in \mathcal{X}$
- Completeness: $p_{x \succ y} \vee p_{y \succ x}$ for all $x \neq y \in \mathcal{X}$
- Transitivity: $p_{x \succ y} \wedge p_{y \succ z} \rightarrow p_{x \succ z}$ for all $x, y, z \in \mathcal{X}$

Now the *Condorcet Paradox* can be modelled in BA:

	(x, y)	(x, z)	(y, 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*

New Aggregation Rules

Idea: take inspiration from voting and preference aggregation rules!

To ease presentation, for the rest of this lecture, assume the number n of agents is *odd*. Thus: *no ties* for *pairwise majority contests*.

Some Voting Rules

Each voter $i \in \mathcal{N}$ provides a linear order $\succ_i$ over the alternatives $\mathcal{X}$.

We have seen examples for voting rules already (Plurality, Borda, ...). Here are three voting rules we will work with today:

- *Borda:* winner = maximises sum of numbers of opponents dominated
- *Copeland:* winner = alternative with most majority wins
- *Maximin:* winner = alternative with weakest worst majority defeat

Remark: All of these rules permit *ties* (there could be several winners).

Voting Rules from Preference Aggregation Rules

Some voting rules are most naturally defined as a 2-step process:

(1) aggregate profile of orders into a *collective order*
(2) pick the *top element* of collective order.

For step (1), each of the following rules picks a linear order over the alternatives that maximises a particular objective function:

- *Slater Rule:* maximise number of times the majority agrees with the chosen ranking for a pair of alternatives
- *Kemeny Rule:* maximise number of times an individual voter agrees with the chosen ranking for a pair of alternatives
- *Tideman's Ranked-Pairs Rule:* order pairs by majority strength, lock in pairwise rankings in that order, but avoid cycles

Remark: Again, there may be *ties* (several optimal collective orders).

Interpretation in Binary Aggregation

The following results are due to Lang and Slavkovik (2013).

However, they originally were stated for a more complex aggregation framework, mixing agenda formulas and integrity constraints. Also, various other details in our presentation here are different, leading to somewhat different statements of results.

We will make heavy use of the concept of the *majority strength* for assigning value $v \in \{0, 1\}$ to issue j in profile $\boldsymbol{B}$:

$$\mathrm{ms}^{\boldsymbol{B}}_{j:v} \quad := \quad |N^{\boldsymbol{B}}_{j:v}| - |N^{\boldsymbol{B}}_{j:\overline{v}}|$$

J. Lang and M. Slavkovik. Judgment Aggregation Rules and Voting Rules. Proc. 3rd International Conference on Algorithmic Decision Theory (ADT-2013).

Preference Aggregation Rules in Binary Aggregation

Recall our encoding of preference aggregation into BA:

$$\Gamma^{\text{pref}} = \underbrace{\bigwedge_{x} (\neg p_{x \succ x})}_{\text{Irreflexivity}} \wedge \underbrace{\bigwedge_{x \neq y} (p_{x \succ y} \vee p_{y \succ x})}_{\text{Completeness}} \wedge \underbrace{\bigwedge_{x,y,z} (p_{x \succ y} \wedge p_{y \succ z} \rightarrow p_{x \succ z})}_{\text{Transitivity}}$$

What aggregation rule (for BA) do we need to use to recover the orders returned by our three preference aggregation rules?

- *Slater* is recovered for Γ^{pref} by the *max-number rule:* compute rational outcome satisfying the most majorities
- *Kemeny* is recovered for Γ^{pref} by the *max-sum rule:* compute rational outcome maximising sum of majority strengths
- *Tideman* is recovered for Γ^{pref} by the *greedy-max rule:* compute rational outcome, accepting issues by majority strength

Proof of correctness of the translation: by inspection. ✓

Remark: completeness constraint is redundant (due to maximisation).

Changing the IC

The results regarding the recovery of Slater / Kemeny / Tideman have been folklore for a few years (though they were rarely stated clearly). The novel idea of Lang and Slavkovik was to look for voting rules we can recover by considering a *different IC*.

For voting rules not defined by going via a full collective linear order, imposing Γ^{pref} (particularly transitivity) is too demanding!

Two possible IC's to instead just say that there is *one winner*:

$$\Gamma^{\text{win}} \quad := \quad \bigvee_x \left[\bigwedge_{y \neq x} (p_{x \succ y} \wedge \neg p_{y \succ x}) \right]$$

$$\Gamma^{\text{win}}_{\sim} \quad := \quad \bigvee_x \left[\underbrace{\bigwedge_{y \neq x} (p_{x \succ y} \wedge \neg p_{y \succ x})}_{x \text{ dominates all}} \wedge \underbrace{\bigwedge_{y,z \neq x} (\neg p_{y \succ z} \wedge \neg p_{z \succ y})}_{\text{indifference between all others}} \right]$$

Results

Careful inspection of the rules induced by combining our three optimisation rules with our three IC's yields this picture:

	max-number	**max-sum**	**greedy-max**
Γ^{pref}	Slater	Kemeny	Tideman
Γ^{win}	Copeland	?	Maximin
$\Gamma^{\text{win}}_{\sim}$	Copeland	Borda	*Maximax*

The combination Γ^{win}/max-sum is a (useful?) variant of Borda (defined by Lang and Slavkovik). *Maximax* is an *ad-hoc* voting rule electing the alternative with the strongest single majority win.

<u>Discussion:</u> Clarifying these parallels between known rules is insightful.

<u>Discussion:</u> So maybe should use these (new) BA rules also in general.

Computational Considerations

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

Today's optimisation rules are different:

- Need to search through *all rational models* to find the "best" one. There might be (*exponentially*) many models that satisfy the IC.
- For the corresponding formula-based *JA* rules, the case is similar: we need to check (exponentially) many *consistent judgment sets*.

  And checking consistency is itself a hard problem.

Summary

Introduced a second framework (after *formula-based JA*) for modelling aggregation of judgments: *binary aggregation with integrity constraints*.

- BA: logical structure imposed "from outside" by *IC*
- JA: logical structure imposed "from within" via *consistency* of sets

Which is better depends what you want to do with it.

Showed how to translate *preference aggregation* into BA. And:

- Natural *generalisation* of *Slater* / *Kemeny* / *Tideman* rules for preference aggregation lead to interesting BA rules.

- Change of IC (dominance *vs.* transitivity) recovers other known voting rules from same BA rules: *Copeland* / *Borda* / *Maximin*.

Remark: All of these rules are *irresolute* (may return several outcomes).

What next?

Next, we want to look into the *computational complexity* of various problems arising in judgment aggregation, including the most basic one: *winner determination* (= computing the outcome).

We will start with a refresher on complexity theory:

- Basics: decision problems, Big-O notation, ...
- Basic complexity classes, such as P and NP
- Notion of completeness w.r.t. a class (as in NP-completeness)
- Polynomial-time reductions to establish NP-hardness results
- Some higher complexity classes of special interest to COMSOC