Judgment Aggregation
in Dynamic Logic of Propositional Assignments

Arianna Novaro, Umberto Grandi, Andreas Herzig
CNRS-IRIT, University of Toulouse

EXPLORE-2017, São Paulo

Motivation

Expressing a (social choice) framework in a formal language allows us to use automated reasoning tools, to *find* or to *check* results.

Social choice functions	$\dashrightarrow$	propositional logic	$\rightarrow$	SAT-solvers
Ranking sets of objects	$\dashrightarrow$	propositional logic	$\rightarrow$	SAT-solvers
Judgment aggregation	$\rightarrow$	JA logic	$\rightarrow$	?

Judgment aggregation $\rightarrow$ DL-PA $\dashrightarrow$ propositional logic $\rightarrow$ SAT-solvers

Papers by Ågotnes, Endriss, Geist, van der Hoek, Lin, Tang, Wooldridge, ...

Talk outline

1. Recap of Judgment Aggregation (in Binary Aggregation)
2. Introduction to Dynamic Logic of Propositional Assignments
3. Translating aggregation rules, axioms and agenda safety
4. A last concluding slide

Binary Aggregation with Integrity Constraints

We have a set of n *agents* and a set of m *issues*.
An *integrity constraint* IC models logical dependencies among issues.

Example of IC: "$\neg$(Issue 1 $\wedge$ Issue 2 $\wedge$ Issue 3)"

	Issue 1	Issue 2	Issue 3
Agent 1	0	1	1
Agent 2	1	0	1
Agent 3	1	1	0
Majority	1	1	1

i's *individual ballot* $B_i \in \{0,1\}^m$
profile $\boldsymbol{B} = (B_1, \ldots, B_n)$
aggregation rule F : Mod(IC)$^n \to \mathcal{P}(\{0,1\}^m) \setminus \{\emptyset\}$

Dynamic Logic of Propositional Assignments

Propositional Dynamic Logic models abstractly computer programs.
Dynamic Logic of Propositional Assignments is an instance of PDL.

The language of DL-PA has two types of expressions:

$$
\begin{array}{rl}
\textbf{formulas} & \varphi \ ::= \ p \mid \top \mid \bot \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle\pi\rangle\varphi \\
\textbf{programs} & \pi \ ::= \ +p \mid -p \mid \pi \,;\, \pi \mid \pi \cup \pi \mid \varphi?
\end{array}
$$

- p ranges over a countable set of propositional variables
- possible to define the other connectives $(\wedge, \rightarrow, \dots)$
- possible to define abbreviations for common programs

$$(p? \,;\, +q) \cup (\neg p? \,;\, -r)$$

$\Rightarrow$ if p then $+\,q$ else $-\,r$

How to translate JA into DL-PA?

The *basic ideas*:

- A profile → a valuation over a set of variables
- An aggregation rule → a DL-PA program
- The outcome → a valuation over another set of variables

	1	2
Agent 1	0	1
Agent 2	1	0
Agent 3	1	1
Majority	1	1

profile

$\mathbb{B}^{3,2} = \{p_{11}, p_{12}, p_{21}, \dots\}$, with p_{11} and p_{22} false

majority

a DL-PA program "maj"

outcome

$\mathbb{O}^2 = \{p_1, p_2\}$, with both p_1 and p_2 true

Translating aggregation rules

All aggregation rules are expressible as DL-PA programs.

Proof idea.

1. Identify a profile $\boldsymbol{B}$ by a formula $\varphi_{\boldsymbol{B}}$
2. Build program $\pi_{F(\boldsymbol{B})}$ setting the outcome as in $F(\boldsymbol{B})$
3. Write a long sequence of "if $\varphi_{\boldsymbol{B}}$ do $\pi_{F(\boldsymbol{B})}$" programs

$\Rightarrow$ Interested in more compact programs for aggregation rules.

Translating Slater rule

Binary Aggregation

$$\mathsf{Slater}_{\mathsf{IC}}(\boldsymbol{B}) \;=\; \operatorname*{argmin}_{B \models \mathsf{IC}} H(B, \mathsf{Maj}(\boldsymbol{B}))$$

DL-PA

change d times the truth value of a variable in $\mathbb{O}^m$

computing the majority rule

$$\mathsf{slater}_{\mathsf{IC}}(\mathbb{B}^{n,m}) \;=\; \boxed{\mathsf{maj}(\mathbb{B}^{n,m})}\,;\, \bigcup_{0 \le d \le m} \left(\boxed{\mathsf{H}(\mathsf{IC}, \mathbb{O}^m, \ge d)?}\,;\,\boxed{\mathsf{flip}^1(\mathbb{O}^m)^d}\right)\,;\,\boxed{\mathsf{IC}?}$$

IC holds?

minimal Hamming distance d to an IC-valuation

We prove that our translations are correct.

Translating axioms

- **Single-profile** axioms *(unanimity, issue-neutrality, . . .)*
  - outcome linked to the structure of a single profile
  - $\Rightarrow$ we use propositional logic

- **Multi-profile** axioms (*independence, monotonicity, anonimity)*
  - outcomes linked to structures of multiple profiles
  - $\Rightarrow$ we use DL-PA

We prove also here that our translations are correct.

Translating monotonicity

Binary Aggregation

Let $(\boldsymbol{B}_{-i}, B'_i) = (B_1, \ldots, B'_i, \ldots, B_n)$ for a profile $\boldsymbol{B}$:

For any issue j, agent i, profiles $\boldsymbol{B} = (B_1, \ldots, B_n)$ and $\boldsymbol{B}' = (\boldsymbol{B}_{-i}, B'_i)$, $\boxed{\text{if}}$ $b_{ij} = 0$ and $b'_{ij} = 1$ $\boxed{\text{then}}$ $F(\boldsymbol{B})_j = 1$ implies $F(\boldsymbol{B}')_j = 1$.

DL-PA

$$\boxed{\bigwedge_{j \in \mathcal{I}} \big(p_j \to \bigwedge_{i \in \mathcal{N}} [+p_{ij} \,; \mathsf{prof}_{\mathsf{IC}}(\mathbb{B}^{n,m}, \mathbb{O}^m) \,; \mathsf{f}(\mathbb{B}^{n,m})] p_j\big)}$$

Translating agenda safety

The *structure* of IC ensures classes of aggregation rules (defined by the axioms they satisfy) to return an outcome satisfying IC.

- median property
- k-median property
- simplified median property

Turned as DL-PA formulas, using the concept of *prime implicants*.

$$\mathsf{PI}(P, \varphi) := [\mathsf{flip}^1(P)]\langle \mathsf{flip}^{\geq 0}(\mathbb{P}_\varphi \setminus P)\rangle \neg\varphi \wedge [\mathsf{flip}^{\geq 0}(\mathbb{P}_\varphi \setminus P)]\varphi.$$

Conclusions

We expressed many different aspects of Judgment Aggregation in Dynamic Logic of Propositional Assignments for the first time.

- Classical aggregation problems (e.g., winner determination) can be expressed in DL-PA.
- Checking whether rules satisfy axioms seems less promising than investigating further the agenda safety problem.
- Implementing examples of automated reasoning.
- Manipulation problem could also be translated.