Arguing about Voting Rules

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

[ joint work with Olivier Cailloux ]
Talk Outline

Paper and talk focus on the problem of *justifying an election outcome* by means of a sequence of simple arguments:

- example of what a future system might be able to do
- logic for expressing arbitrary arguments about voting rules
- algorithm for justifying Borda outcomes
Example

Not always obvious who should win. For example, for the profile below the *Veto* rule recommends $b$, while the *Borda* rule recommends $a$:

| Voter 1: $a \succ b \succ c$ |
| Voter 2: $a \succ b \succ c$ |
| Voter 3: $c \succ b \succ a$ |

Suppose you want to convince a user that $a$ should win . . .
Voter 1: $a \succ b \succ c$
Voter 2: $a \succ b \succ c$
Voter 3: $c \succ b \succ a$

**System:** Take the *red subprofile*. Here, *a should win*, right? [unanimity]

**User:** Obviously!

**System:** Now consider the *green subprofile*. For symmetry reasons, there should be a *three-way tie*, right? [cancellation]

**User:** Sounds reasonable.

**System:** So, as there was a three-way tie for the green part, the red part should decide the overall winner, right? [reinforcement]

**User:** Yes.

**System:** To summarise, you agree that *a should win*. }

*Ulle Endriss*
Voting Theory for Variable Electorates

Basic ingredients:

- $\mathcal{A}$: finite set of *alternatives*
- $\mathcal{L}(\mathcal{A})$: linear orders (*preferences*) on $\mathcal{A}$
- $\mathcal{N}$: infinite set of potential *voters*

A *profile* is a partial function $\mathcal{R} : \mathcal{N} \rightarrow \mathcal{L}(\mathcal{A})$ (pref’s of some voters).

A *voting rule* $f$ maps any given profile $\mathcal{R}$ to a nonempty set $\mathcal{A} \subseteq \mathcal{A}$. 
The Logic

Propositional language over atoms \([R \mapsto A]\), one for each profile \(R\) and each nonempty set \(A\) of alternatives, interpreted on voting rules \(f\):

\[ f \models [R \mapsto A] \iff f(R) = A \]

Can express anything about voting rules, albeit in a brute force fashion.

For example, the reinforcement axiom can be written as the set of all the following formulas with \(\text{dom}(R) \cap \text{dom}(R') = \emptyset\) and \(A \cap A' \neq \emptyset\):

\[ [R \mapsto A] \land [R' \mapsto A'] \rightarrow [R \oplus R' \mapsto A \cap A'] \]
Justifying Election Outcomes

Write $\Delta \models \varphi$ to say that every voting rule $f$ that satisfies all the formulas in $\Delta$ also satisfies $\varphi$. For example:

- $\Delta$ might be a set of intuitively appealing properties (axioms)
- $\varphi$ might be a claim about a specific outcome, such as $[R \mapsto f(R)]$

**Theorem 1 (Completeness)** $\Delta \models \varphi$ in our logic iff $\Delta \cup \text{FUNC} \vdash \varphi$ in classical propositional logic, where:

$$\text{FUNC} = \bigcup_{R} \left\{ \bigvee_{A} [R \mapsto A] \right\} \cup \bigcup_{R} \bigcup_{A \neq A'} \left\{ [R \mapsto A] \land [R \mapsto A'] \rightarrow \bot \right\}$$

Thus, we can prove claims $\varphi$ about voting rules given assumptions $\Delta$ using, say, natural deduction. At least in theory.

In practice, $\Delta$ will usually be huge and deciding $\vdash$ is coNP-complete.
Justifying Borda Outcomes in Practice

Main technical contribution of the paper is an algorithm to compute, for any profile $R$, a proof for $[R \mapsto \text{Borda}(R)]$ from some axioms.

Main axioms used are:

- **Reinforcement**: $[R \mapsto A] \land [R' \mapsto A'] \rightarrow [R \oplus R' \mapsto A \cap A']$
- **Cancellation**: if all majority contests are tied, everyone wins

Main trick is to build a profile $R'$ with (i) “obvious” winners $f(R)$ and (ii) same weighted majority graph as $kR$. Claim then follows:

$$kR \oplus \overline{kR} \oplus R'$$

Profile $R'$ is built using Reinforcement on basic profiles such as:

\[
\begin{align*}
\left[ \begin{array}{c}
a \succ b \succ c \succ d \\
b \succ a \succ d \succ c
\end{array} \right] & \mapsto \{a, b\} \\
\left[ \begin{array}{c}
a \succ b \succ c \succ d \\
d \succ a \succ b \succ c \\
c \succ d \succ a \succ b \\
b \succ c \succ d \succ a
\end{array} \right] & \mapsto \{a, b, c, d\}
\end{align*}
\]
Last Slide

We have seen:

- logic for describing example-based properties of voting rules
- can be used to justify outcomes (in theory very general)
- concrete algorithm to compute short justifications for Borda

Long-term agenda: arguing about voting rules, beyond justification