A Computational Perspective on Judgment Aggregation

Umberto Grandi

Department of Mathematics
University of Padova

8 July 2013

Computational Aspects of Collective Decision Making

Different problems of social choice studied:

- Choosing a winner given individual preferences over candidates.
- Allocate resources to users in an optimal way.
- Finding a stable matching of students to schools.
- Merge the results of several search engines.

Different computational techniques used:

- Algorithm design to implement complex mechanisms.
- Complexity theory to understand limitations.
- Knowledge representation techniques to compactly model problems.
- Simulations and real-world data analysis.

Research areas: Computational Social Choice, Algorithmic Decision Theory, Algorithmic Game Theory, Computational Economics.

Y. Chevaleyre, U. Endriss, J. Lang, and N. Maudet. A Short Introduction to Computational Social Choice. Proceedings of SOFSEM-2007.

Examples: Computational Complexity and Combinatorial Voting

Example 1: Elections can always be manipulated (see the Gibbard Satterthwaite Theorem), but how difficult is it to manipulate an election?

For the majority rule it is easy (=polynomial) for other cases like Australian elections it is computationally hard (NP-complete)

> Computational complexity can thus be a barrier to manipulation.

Example 2: When the set of alternatives has a combinatorial structure, representation of preferences may be exponential! Examples:

- Electing a committee (3 members out of 8 candidates: 112 combinations)
- Multiple variables to describe features for recommender systems

> Computer Science provides tools for compact representation of preferences and studies how to reason with such structures.

Overview

1. Binary aggregation with integrity constraints:
   - A general framework for the aggregation of individual expressions
   - Characterisation of paradoxes in Social Choice Theory and computational complexity of safe aggregation
2. Iteration of strategic voting:
   - Experimental simulation of iterated response to polls
   - Surprising increase in Condorcet efficiency

Binary Aggregation with Integrity Constraints

Everything Starts From Paradoxical Situations

Suppose three agents in a multi-agent system need to decide whether to perform a collective decision A. The decision is performed if two parameters T_1 and T_2 exceed a given threshold. Consider the following situation:

	T_1	T_2	A
Agent 1	Yes	Yes	Yes
Agent 2	No	Yes	No
Agent 3	Yes	No	No
Majority	Yes	Yes	No

Should the agents perform action A or not?

- A majority of agents think the first parameter exceeds the threshold.
- A majority of agents think the second parameter exceeds the threshold.
- But: a majority of agents think action A should not be performed!!

Binary Aggregation

Ingredients:

- A finite set N of individuals
- A finite set $\mathcal{I} = \{1, \dots, m\}$ of issues
- A boolean *combinatorial domain*: $\mathcal{D} = D_1 \times \cdots \times D_m$ with $|D_i| = 2$

Definition

An aggregation procedure is a function $F : \mathcal{D}^N \to \mathcal{D}$ mapping each profile of ballots $\mathbf{B} = (\mathbf{B}_1, \dots, \mathbf{B}_n)$ to an element of the domain $\mathcal{D}$.

Example: Three agents with sensors

- $\mathcal{N} = \{1, 2, 3\}$
- $\mathcal{I} = \{T_1, T_2, A\}$
- Individuals submit ballots in $\mathcal{D} = \{0, 1\}^3$

$B_1 = (0, 1, 0)$ the first individual think the action should not be performed.

Wilson (1975), Rubinstein and Fishburn (1986), Dokow and Holzman (2005)

Integrity Constraints

A propositional language $\mathcal{L}$ to express integrity constraints:

- One propositional symbol for every issue: $PS = \{p_1, \dots, p_m\}$
- $\mathcal{L}_{PS}$ closing under connectives $\wedge$, $\vee$,$\neg$, $\rightarrow$ the set of atoms PS

IC $\in \mathcal{L}$ expresses properties of ballots $B \in \mathcal{D} = \{0,1\}^m$
Given an integrity constraint IC $\in \mathcal{L}_{PS}$, a rational ballot is $B \in \mathrm{Mod}(\mathrm{IC})$

Example: Three agents with sensors

Perform action A if both parameters exceed the thresholds.
Propositional constraint: IC $= (p_{T_1} \wedge p_{T_2}) \rightarrow p_A$

Individual 1 submits $B_1 = (1,1,1)$: B_1 satisfies IC ✓
Individual 2 submits $B_2 = (0,1,0)$: $B_2 \models$ IC ✓
Individual 3 submits $B_3 = (1,0,0)$: $B_3 \models$ IC ✓

Majority aggregation outputs $(1,1,0)$: IC not satisfied.

Condorcet Paradox

Preference aggregation (Condorcet, 1785, Arrow, 1963) studies how to obtain a collective ranking of alternatives from individual preferences. Used in voting, political theory, and CS (e.g. aggregate rankings of search engines).

In 1785 le Marquis de Condorcet pointed out that:

$$\begin{array}{c} \triangle \succ_1 \bigcirc \succ_1 \square \\ \square \succ_2 \triangle \succ_2 \bigcirc \\ \bigcirc \succ_3 \square \succ_3 \triangle \\ \hline \triangle \succ \bigcirc \succ \square \succ \triangle \end{array}$$

The collective ranking is a cycle!

Preference Aggregation

$IC_<$ encodes the rationality assumption for decision theory:

Irreflexivity: $\neg p_{aa}$ for all $a \in \mathcal{X}$

Completeness: $p_{ab} \vee p_{ba}$ for all $a \neq b \in \mathcal{X}$

Transitivity: $p_{ab} \wedge p_{bc} \rightarrow p_{ac}$ for $a, b, c \in \mathcal{X}$ pairwise distinct

And translate the paradox in binary aggregation:

	△○	○□	△□
Agent 1	1	1	1
Agent 2	0	1	0
Agent 3	1	0	0
Maj	1	1	0

The majority rule generates a paradox with respect to $IC_<$

The Common Structure of Paradoxes

Proposition

The majority rule does not generate a paradox with respect to IC if and only if IC is equivalent to a conjunction of clauses of size $\leqslant 2$.

$$\mathcal{IC}(Maj) = \text{2-CNF}$$

Common feature of all paradoxes:
clauses of size 3 are not lifted by majority

Problem

How (computationally) hard is it to check whether the IC is paradoxical?
Very hard: most likely it is Σ_2^p-hard.

U. Endriss, U. Grandi, D. Porello. Complexity of Judgment Aggregation. JAIR, 2012.

Restricted Iterative Voting

Individuals have preferences $\leqslant_i$ over candidates C and a voting rule associates a winning candidate in C with a profile $(\leqslant_1, \ldots, \leqslant_n)$.

Example: give s_j points to candidates in position j in individual preferences, and elect the candidates with maximal score (positional scoring rules). Plurality has vector of scores $(1, 0, \ldots, 0)$.

Manipulation occurs whenever a voter changes her ballot in her favour:

3 voters	$a \succ b \succ c$
2 voters	$b \succ c \succ a$
2 voters	$c \succ b \succ a$
Plurality	a

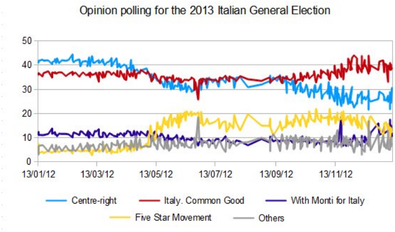
$\longrightarrow$

3 voters	$a \succ b \succ c$
2 voters	$b \succ c \succ a$
2 voters	$b \succ c \succ a$
Plurality	b

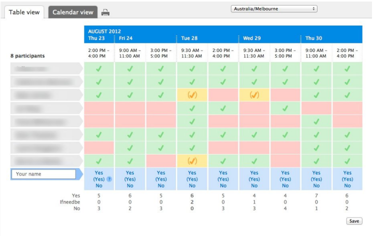
We allow voters to iterate in turns using a restricted set of manipulation moves

Practical Examples

In practice, iterative manipulation do occur:



Iterative response
to repeated polls



Approval voting with
iterative manipulation

Image source: Wikipedia, Doodle.com

Theoretical and Experimental Analysis

With our restricted manipulation moves we can prove/observe convergence of iteration. Axiomatic properties (e.g., unanimity) are preserved after iteration.

Experimental results

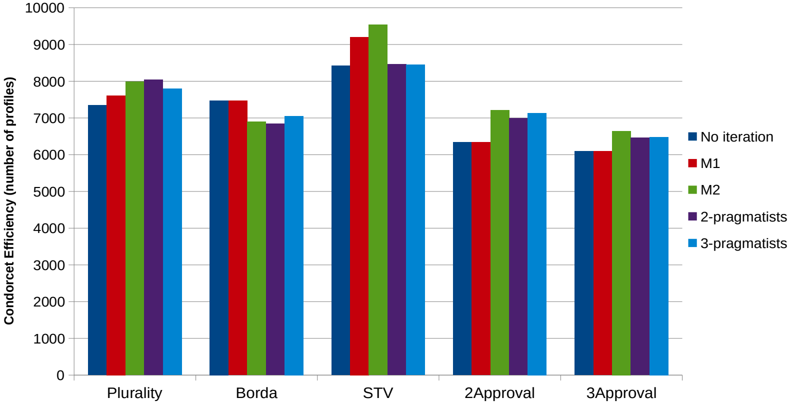
We generated 10.000 profiles using the urn model to increase correlation between individual preferences.

We tested the Condorcet efficiency: how many times a voting rule elects a Condorcet winner when there exists one (a Condorcet winner is a candidate which beats every other voter in pairwise comparison).

We observed an increase in the Condorcet efficiency of most iterated voting rules!

Results: Condorcet Efficiency - Urn Model

Good results for low number of voters and high number of candidates
Modelling a classic Doodle poll (25 time slots, 10 candidates)



Conclusions

Computer Science and Social Choice Theory model agents in a similar way:

- Rationality assumptions (e.g., transitive preferences, consistent judgments...) can be modelled using a simple logical language
- Problems of interest are related to implementation and reasoning: computational complexity, compactness, algorithms...
- Possibility of performing experimental simulations (e.g., the case of restricted iterative manipulation)

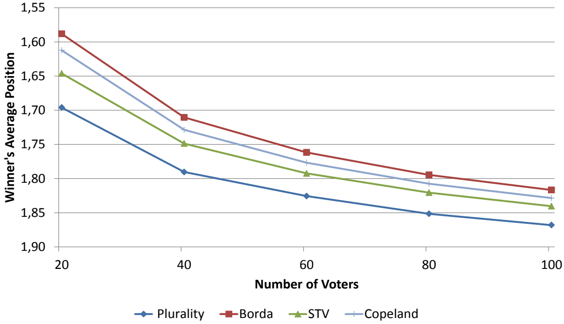
Thank you for your attention!

U. Grandi and U. Endriss, Lifting Integrity Constraints in Binary Aggregation. *Artificial Intelligence* 199-200: 45-66, 2013.

U. Grandi, A. Loreggia, F. Rossi, K. B. Venable and T. Walsh. Restricted Manipulation in Iterative Voting: Condorcet Efficiency and Borda Score. To appear in *Proceedings of the 3rd International Conference on Algorithmic Decision Theory (ADT-2013)*, 2013.

Average Position of the Winner (aka Borda score)

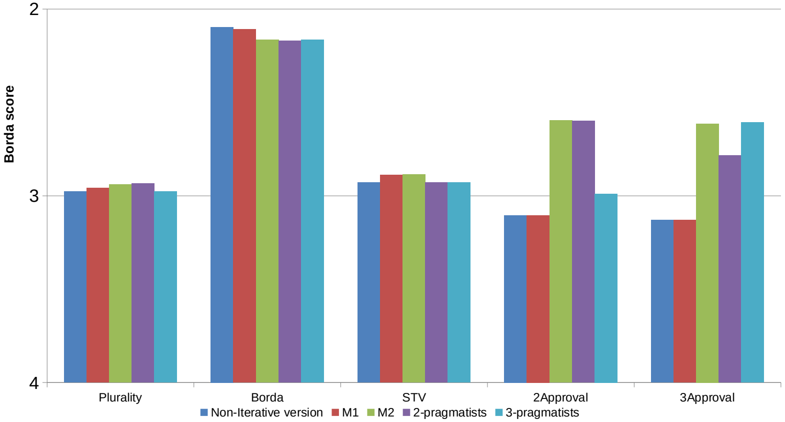
How much preferred is the winner in average?



Recall that Borda elects the candidate with the highest "average position"

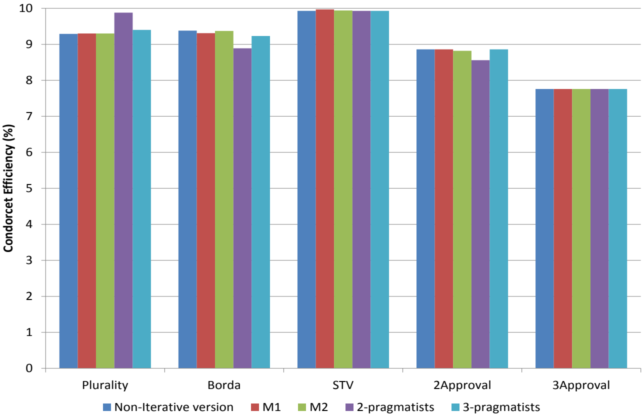
Average Position of the Winner: Urn Model

For all voting rules (except for Borda) the position of the winner shows an increase by allowing iterated restricted manipulation:



Real-world data: the Netflix Dataset

Data constructed from www.preflib.org, Netflix dataset
10 candidates, 100 voters: too little iteration!
Profiles are too often unanimous, iteration takes place in $< 0.1\%$ of profiles!



Same experiment with Skate dataset: too much correlation, too little iteration