Probabilistic Possible Winner Determination

Nadja Betzler[1]

joint work with

Yoram Bachrach[2] and Piotr Faliszewski[3]

[1]Friedrich-Schiller-Universität Jena, Germany
[2]Microsoft Research Ltd., Cambridge, UK
[3]AGH University of Science and Technology, Kraków, Poland

Twenty-Fourth AAAI Conference on Artificial Intelligence
July 2010

Motivation

Voting scenarios arise in many situations:

- political elections
- committees: decisions about job applicants
- web site rankings
- recommender systems
- choice of restaurant
- ...

Motivation

Typical voting scenario for joint decision making:

Voters give preferences over a set of candidates as linear orders.

Example: candidates: $C = \{a, b, c, d\}$

profile:	vote 1:	a	>	b	>	c	>	d
	vote 2:	a	>	d	>	c	>	b
	vote 3:	b	>	d	>	c	>	a

Aggregate preferences according to a voting rule

Voting rules mainly considered in this work: **Scoring rules**

Scoring rules

Examples:

- plurality: $(1, 0, \ldots, 0)$
- 2-approval: $(1, 1, 0, \ldots, 0)$
- veto: $(1, \ldots, 1, 0)$
- Borda: $(m-1, m-2, \ldots, 0)$
- Formula 1 scoring: $(25, 18, 15, 12, 10, 8, 6, 4, 2, 1, 0, \ldots, 0)$

For m candidates, a scoring vector $(\alpha_1, \alpha_2, \ldots, \alpha_m)$ provides a scoring value for every position.

The scoring values of every candidate are summed up and the candidate with the highest score wins.

Partial information

In the typical model, votes need to be presented as linear orders.

Realistic settings: voters may only provide partial information.

For example:

- not all voters have given their preferences yet
- new candidates are introduced
- a voter cannot compare several candidates because of lack of information

How to deal with partial information?

How to deal with partial information?

Previous work [Konczak and Lang 2005], [Walsh, AAAI 2007], [Xia and Conitzer, AAAI 2008], [Betzler, Hemman, and Niedermeier, IJCAI 2009], ...

- Possible Winner problem:
  Is there an extension in which a designated candidate wins?
- Necessary Winner problem:
  Does a designated candidate win in **every** extension?

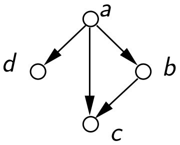
This work

In how many extensions does a designated candidate win?

Partial vote

A **partial vote** is a transitive and antisymmetric relation.

Example: $C = \{a, b, c, d\}$
partial vote: $a \succ b \succ c, a \succ d$



possible **extensions**:

1. $a > d > b > c$
2. $a > b > d > c$
3. $a > b > c > d$

An extension of a profile of partial votes extends every partial vote.

Problems

Setting: unweighted votes

#Possible Winner

Given: Voting rule r, a partial profile on a set of candidates, a designated candidate.
Question: In how many extensions of the partial profile does the designated candidate win according to r?

Special case: #MANIPULATION
Input consists of a set of linear and a set of empty votes.

Overview of Main Results

Exact solutions

- #Possible-Winner:
  #P-hardness (or NP-hardness) for all scoring rules.
- #Manipulation:
  Polynomial-time dynamic programming algorithm for a class of scoring rules.

Randomized approximation

Polynomial-time sampling algorithm approximating the proportion of the number of extensions in which the designated candidate wins.

(In contrast: other works investigate how likely manipulation is "in general" for a specific voting system [Xia and Conitzer, EC 2008], [Friedgut et al., FOCS 2008], ...)

#Possible Winner

- Necessary Winner can be solved in polynomial time for all scoring rules. [Xia and Conitzer, AAAI 2008]
- Possible Winner can be solved in polynomial time for plurality and veto and is NP-hard for every other reasonable scoring rule. [Baumeister and Rothe, ECAI 2010], [Betzler and Dorn, JCSS], [Xia and Conitzer, AAAI 2008]

New results for the counting version:

Theorem

#Possible Winner is #P-hard for plurality and veto even if
there is only one partial vote, or
at most one undetermined pair per vote.

#Manipulation

Manipulation under scoring rules for unweighted votes and unbounded number of candidates:

- NP-hardness for a specific scoring rule [Xia et al., EC 2010]
- some easy-to-see polynomial-time solvability results
- open cases such as Borda

Many results for other voting systems and other scenarios.

#Manipulation

Manipulation under scoring rules for unweighted votes and unbounded number of candidates:

- NP-hardness for a specific scoring rule [Xia et al., EC 2010]
- some easy-to-see polynomial-time solvability results
- open cases such as Borda

Many results for other voting systems and other scenarios.

Theorem

For constant k, #Manipulation can be solved in polynomial time for k-valued scoring rules.

k-valued scoring rule: all but k candidates get the same number of points in a vote

Examples: k-approval, $(25, 18, 15, 12, 10, 8, 6, 4, 2, 1, 0, \ldots, 0)$

Randomized approximation

Input:

- polynomial-time computable voting rule
- a partial profile and a designated candidate
- positive rational numbers δ (error probability) and ϵ (approximation guarantee)

Output:
a value α such that the proportion of extensions in which the designated candidate wins is within $[\alpha - \epsilon, \alpha + \epsilon]$ with probability at least $1 - \delta$.

Running time:
polynomial in "size of partial profile", $1/\epsilon$, and $\log 1/\delta$

Sampling algorithm

Input: Partial profile, a designated candidate, δ, ϵ
Output: "randomized approximation" of the propertion of "winning extensions"

$x := 0$
for $i = 1$ **to** $r := \left\lceil \frac{\ln \frac{2}{\delta}}{2\,\epsilon^2} \right\rceil$
  choose a random linear extension of the partial profile
  if the designated candidate wins
    $x = x + 1$
Return $\frac{x}{r}$

uniformly sampling linear votes can be done in $O(n^3 \log n)$ time
[Huber, Disrete Mathematics 2006]
performance guarantee: follows from Hoeffding's inequality

Overview Results

#Possible-Winner:

- #P-hardness for plurality and veto
- Polynomial-time sampling algorithm approximating the proportion of the number of extensions in which the designated candidate wins.

#Manipulation:

- Polynomial-time dynamic programming algorithm for *k*-valued scoring rules.
- (Polynomial-time solvability results for #Manipulation under plurality with run-off, cup voting with a fixed agenda in general and scoring rules in case of coalition size one.)

Open questions

dynamic programming algorithm for k-valued scoring rules has a running time of $O(s^{f(k)})$ for a profile of size s:
Can this be improved to $s^{O(1)} \cdot f(k)$?

Study of #Possible Winner for other scoring rules such as Borda, Bucklin, or Copeland

The sampling algorithm assumes that all extensions appear with the same probability. What about more realistic models?

Thank You