Exact Rank Aggregation with Parameterized Algorithms

Robert Bredereck

Friedrich-Schiller-Universität Jena, Germany

Cost IC0602 International Doctoral School
Algorithmic Decision Theory: Computational Social Choice
April 12, 2010

Rank Aggregation

Election

Set of votes V, set of candidates C.
A vote is a ranking (total order) over all candidates.

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

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

How to aggregate the votes into a "consensus ranking"?

Kemeny score: KT-distance

KT-distance (between two votes v and w)

$$\text{KT-dist}(v, w) = \sum_{\{c,d\} \subseteq C} d_{v,w}(c, d),$$

where $d_{v,w}(c, d)$ is 0 if v and w rank c and d in the same order, 1 otherwise.

Example:

v_1: a > b > c
v_2: a > c > b
v_3: b > c > a

$$\begin{aligned}\text{KT-dist}(v_1, v_2) &= d_{v,w}(a, b) + d_{v,w}(a, c) + d_{v,w}(b, c)\\ &= 0 + 0 + 1\\ &= 1\end{aligned}$$

Kemeny Consensus

Kemeny score of a ranking r:

Sum of KT-distances between r and all votes

Kemeny consensus r_{con}:

A ranking that minimizes the Kemeny score

v_1 :	$a > b > c$	$\text{KT-dist}(r_{con}, v_1) = 0$
v_2 :	$a > c > b$	$\text{KT-dist}(r_{con}, v_2) = 1$ because of $\{b, c\}$
v_3 :	$b > c > a$	$\text{KT-dist}(r_{con}, v_3) = 2$ because of $\{a, b\}$ and $\{a, c\}$
r_{con} :	$\mathbf{a} > \mathbf{b} > \mathbf{c}$	Kemeny score: $0 + 1 + 2 = 3$

Decision problem

Kemeny Score

Input: An election (V, C) and a positive integer k.
Question: Is there a Kemeny consensus of (V, C) with Kemeny score at most k?

Decision problem

Kemeny Score

Input: An election (V, C) and a positive integer k.
Question: Is there a Kemeny consensus of (V, C) with Kemeny score at most k?

Applications:

- Ranking of web sites (meta search engine)
- Sport competitions
- Databases
- Voting systems

Known results

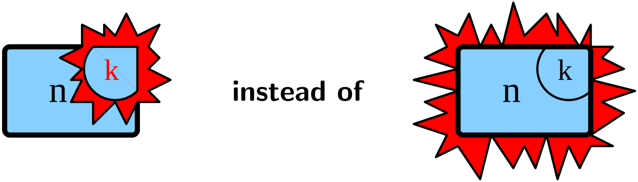
- KEMENY SCORE is NP-complete (even for 4 votes)
  [BARTHOLDI ET AL., SCW 1989], [DWORK ET AL., WWW 2001]

Algorithms:

- factor 8/5-approximation, randomized: factor 11/7
  [VAN ZUYLEN AND WILLIAMSON, WAOA 2007],
  [AILON ET AL., JACM 2008]
- PTAS [KENYON-MATHIEU AND SCHUDY, STOC 2007]
- Heuristics; greedy, branch and bound (experimental)
  [DAVENPORT AND KALAGNANAM, AAAI 2004],
  [V. CONITZER, A. DAVENPORT, AND J. KALAGNANAM, AAAI 2006],
  [F. SCHALEKAMP AND A. VAN ZUYLEN, ALENEX 2009]

Parameterized Complexity

Given an NP-hard problem with input size n and a parameter k
Basic idea: Confine the combinatorial explosion to k



Definition

A problem of size n is called **fixed-parameter tractable** with respect to a parameter k if it can be solved exactly in $f(k) \cdot n^{O(1)}$ time.

Parameters: # votes, # candidates, **average KT-distance**, ...

Data reduction rule

You can see data reduction rules as preprocessing step to solve a problem:

Basic idea

A data reduction rule shrinks an instance of a problem to an "equivalent" instance by cutting away easy parts of the original instance.

We focus on **polynomial-time** data reduction rules for **Kemeny Score**.

Simple reduction rules

Condorcet property: (weak) A candidate c beating every other candidate at least than half of the votes, that is,

$$c \geq_{1/2} c' \text{ for every candidate } c' \neq c,$$

takes the first position in at least one Kemeny consensus.

Reduction Rule

If there is (weak) Condorcet winner in an election provided by a KEMENY SCORE instance, then delete this candidate.

Simple reduction rules

Condorcet property: (weak) A candidate c beating every other candidate at least than half of the votes, that is,

$$c \geq_{1/2} c' \text{ for every candidate } c' \neq c,$$

takes the first position in at least one Kemeny consensus.

Reduction Rule

If there is (weak) Condorcet winner in an election provided by a KEMENY SCORE instance, then delete this candidate.

Reduction Rule

If there is a subset $C' \subset C$ of candidates with $c' \geq_{1/2} c$ for every $c' \in C'$ and every $c \in C \setminus C'$, then replace the original instance by the two subinstances "induced" by C' and $C \setminus C'$.

Note: A subset C' can be found in polynomial time.

Back to our initial example

v_1:	a	>	b	>	c
v_2:	a	>	c	>	b
v_3:	b	>	c	>	a

Back to our initial example

v_1:	a	$>$	b	$>$	c
v_2:	a	$>$	c	$>$	b
v_3:	b	$>$	c	$>$	a

The candidate a is a condorcet winner. The set $\{b, c\}$ is a condorcet looser set.

Reduction rules using "dirty candidates"

A candidate c is **non-dirty** if for every other candidate c' either $c' \geq_{3/4} c$ or $c \geq_{3/4} c'$. Otherwise c is **dirty**.

Lemma

For a non-dirty candidate c and candidate $c' \in C \setminus \{c\}$:
If $c \geq_{3/4} c'$, then $c > \cdots > c'$ in every Kemeny consensus.
If $c' \geq_{3/4} c$, then $c' > \cdots > c$ in every Kemeny consensus.

Reduction Rule

If there is a non-dirty candidate, then delete it and partition the instance into two subinstances accordingly.

Reduction rules using “dirty candidates”

A candidate c is **non-dirty** if for every other candidate c' either $c' \geq_{3/4} c$ or $c \geq_{3/4} c'$. Otherwise c is **dirty**.

Lemma

For a non-dirty candidate c and candidate $c' \in C \setminus \{c\}$:
If $c \geq_{3/4} c'$, then $c > \cdots > c'$ in every Kemeny consensus.
If $c' \geq_{3/4} c$, then $c' > \cdots > c$ in every Kemeny consensus.

Reduction Rule

If there is a non-dirty candidate, then delete it and partition the instance into two subinstances accordingly.

Further rule: an “extended” reduction rule based on “sets of non-dirty candidates”…

Reduction rules using "dirty candidates"

$a_1 > a_2 > a_3 > c > b_1 > b_2$
$a_3 > a_2 > c > a_1 > b_2 > b_1$
$a_1 > c > a_2 > b_2 > b_1 > a_3$
$a_2 > a_3 > a_1 > b_1 > b_2 > c$

$a_i \geq_{3/4} c$ and $c \geq_{3/4} b_i$
$\Rightarrow$
in every Kemeny consensus:
$\{a_1, a_2, a_3\} > c > \{b_1, b_2\}$

A candidate c is **non-dirty** if for every other candidate c' either $c' \geq_{3/4} c$ or $c \geq_{3/4} c'$. Otherwise c is **dirty**.

Lemma

For a non-dirty candidate c and candidate $c' \in C \setminus \{c\}$:
If $c \geq_{3/4} c'$, then $c > \cdots > c'$ in every Kemeny consensus.
If $c' \geq_{3/4} c$, then $c' > \cdots > c$ in every Kemeny consensus.

Reduction Rule

If there is a non-dirty candidate, then delete it and partition the instance into two subinstances accordingly.

Average KT-distance as parameter for Kemeny Score

Parameter: average KT-distance between the input votes

$$d := \frac{2}{n(n-1)} \cdot \sum_{\{u,v\} \subseteq V} \text{KT-dist}(u, v).$$

Known fixed-parameter tractability results:

- dynamic programming with running time $O(16^d \cdot \text{poly}(n))$
  [Betzler, Fellows, Guo, Niedermeier, and Rosamond, AAMAS 2009]
- branching algorithm with running time $O(5.83^d \cdot \text{poly}(n))$
  [Simjour, IWPEC 2009]

Average KT-distance as parameter for Kemeny Score

Theorem

A Kemeny Score instance with average KT-distance d can be reduced in polynomial time to an "equivalent" instance with less than $11 \cdot d$ candidates.

Average KT-distance as parameter for Kemeny Score

Theorem

A KEMENY SCORE instance with average KT-distance d can be reduced in polynomial time to an "equivalent" instance with less than $11 \cdot d$ candidates.

Idea of proof:

- Recall: Reduction rule which deletes all non-dirty candidates
- Every dirty candidate must be involved in at least one candidate pair that is not ordered according to the "$\geq_{3/4}$-majority"
- For an instance with n votes, every such pair contributes with at least $n/4 \cdot 3n/4$ to the average KT-distance.

Experimental results: Meta search engines

Four votes: Google, Lycos, MSN Live Search, and Yahoo!

top 1000 hits each, candidates that appear in all four rankings

search term	#cand.	time [s]	structure of reduced instance solved/unsolved		
affirmative action	127	0.41	[27]	> **41** >	[59]
alcoholism	115	0.21	[115]		
architecture	122	0.47	[36]	> **12** > [30] > **17** >	[27]
blues	112	0.16	[74]	> **9** >	[29]
cheese	142	0.39	[94]	> **6** >	[42]
classical guitar	115	1.12	[6]	> **7** > [50] > **35** >	[17]
Death Valley	110	0.25	[15]	> **7** > [30] > **8** >	[50]
field hockey	102	0.21	[37]	> **26** > [20] > **4** >	[15]
gardening	106	0.19	[54]	> **20** > [2] > **9** > [8] > **4** >	[9]
HIV	115	0.26	[62]	> **5** > [7] > **20** >	[21]
lyme disease	153	2.61	[25]	> **97** >	[31]
mutual funds	128	3.33	[9]	> **45** > [9] > **5** > [1] > **49** >	[10]
rock climbing	102	0.12	[102]		
Shakespeare	163	0.68	[100]	> **10** > [25] > **6** >	[22]
telecommuting	131	2.28	[9]	> **109** >	[13]

Conclusion

In practice:

Data reduction should be applied whenever possible. There are many real-world instances that are only (exactly) solvable with data reduction rules.

In theory:

Parameterized algorithmics offer a framework to analyze the effectiveness of data reduction rules.

Still open:

- more (structural) parameters
- bound also number of votes
- more data reduction rules

Literature

Talk is based on

- Exact Rank Aggregation Based on Effective Data Reduction
  [N. Betzler, R. Bredereck, and R. Niedermeier, manuscript]

General literature on parameterized algorithms

- R. G. Downey and M. R. Fellows, Parameterized Complexity, Springer, 1999
- R. Niedermeier, Invitation to Fixed-Parameter Algorithms, Oxford University Press, 2006