Data Reduction and Problem Kernels for Voting Problems

Nadja Betzler

Friedrich-Schiller-Universität Jena

Dagstuhl Seminar
Computational Foundations of Social Choice
March 2010

Data reduction rule

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 **NP-hard problems**.

First example: KEMENY SCORE (or RANK AGGREGATION)

Kemeny Score

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:

$$v : a > b > c$$
$$w : c > a > b$$

$$\begin{aligned} \text{KT-dist}(v, w) &= d_{v,w}(a, b) + d_{v,w}(a, c) + d_{v,w}(b, c) \\ &= 0 + 1 + 1 \\ &= 2 \end{aligned}$$

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 score at most k?

- Kemeny Score is NP-complete (even for 4 votes)
  [Bartholdi et al., SCW 1989], [Dwork et al., WWW 2001]
- many algorithmic results and applications...

Simple reduction rules

Condorcet property: A candidate c beating every other candidate in more than half of the votes, that is,

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

takes the first position in every Kemeny consensus.

Reduction Rule

If there is a Condorcet winner in an election provided by a Kemeny Score instance, delete this candidate.

Simple reduction rules

Condorcet property: A candidate c beating every other candidate in more than half of the votes, that is,

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

takes the first position in every Kemeny consensus.

Reduction Rule

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

Reduction Rule

If there is a subset $C' \subset C$ of candidates with $c' >_{1/2} c$ 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' must be found in polynomial time.

Reduction rules using "dirty candidates"

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

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.

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

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

Reduction rules using "dirty candidates"

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

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"...

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]

Theoretical Analysis

Central question

How to theoretically measure the performance of data reduction rules?

For which kind of instances do the reduction rules perform well?

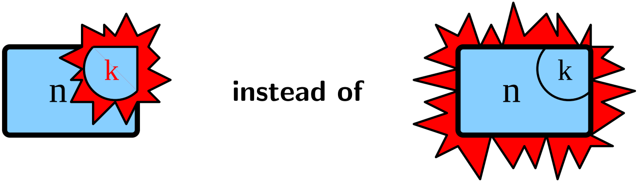
Recall: NP-hard problems and polynomial-time data reduction rules

Classical complexity theory:
Reduction rules cannot always be applied, else P=NP.

$\Rightarrow$ No provable performance guarantees for data reduction.

Parameterized Complexity

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



Definition

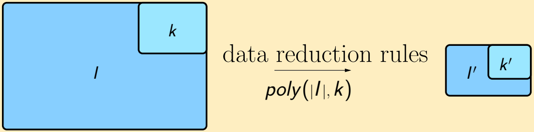
A problem of size n is called *fixed-parameter tractable* with respect to a parameter k if it can be solved exactly in $f(k) \cdot n^{O(1)}$ time.

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

Problem kernel

Let $L \subseteq \Sigma^* \times N$ be a parameterized problem. An instance of L is denoted by (I, k).

Kernelization



- $(I, k) \in L \iff (I', k') \in L$
- $k' \leq k$
- $|I'| \leq g(k)$

If g is a polynomial, we say L admits a *polynomial problem kernel*.

Known fact: A problem is fixed-parameter tractable if and only if it admits problem kernel.

Partial problem kernel for KEMENY SCORE

Parameter: average KT-distance between the input votes

$$d_a := \frac{1}{n(n-1)} \cdot \sum_{u,v \in V, u \neq v} \text{KT-dist}(u, v).$$

Known FPT-results:

- dynamic programming with running time $O^*(16^{d_a})$
  [BETZLER, FELLOWS, GUO, NIEDERMEIER, AND ROSAMOND, AAMAS 2009]
- branching algorithm with running time $O^*(5.83^{d_a})$
  [SIMJOUR, IWPEC 2009]

Partial problem kernel for KEMENY SCORE

Parameter: average KT-distance between the input votes

Theorem

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

Partial problem kernel for Kemeny Score

Parameter: average KT-distance between the input votes

Theorem

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

Partial problem kernel for Kemeny Score

Parameter: average KT-distance between the input votes

Theorem

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

Theoretical challenge: "Reduce the number of votes"

Example for a problem kernel

Possible Winner Problem

Input: Given a set of partial orders (votes) on a set of candidates and a distinguished candidate c.
Question: Can the partial orders be extended to linear ones such that c becomes a winner?

k-approval voting: In every vote, the first k candidates get one point and the remaining candidates zero points.

Example for a problem kernel

POSSIBLE WINNER Problem

Input: Given a set of partial orders (votes) on a set of candidates and a distinguished candidate c.
Question: Can the partial orders be extended to linear ones such that c becomes a winner?

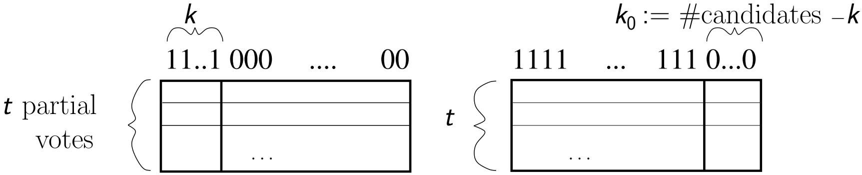
k-approval voting: In every vote, the first k candidates get one point and the remaining candidates zero points.

"Relevant" known results:

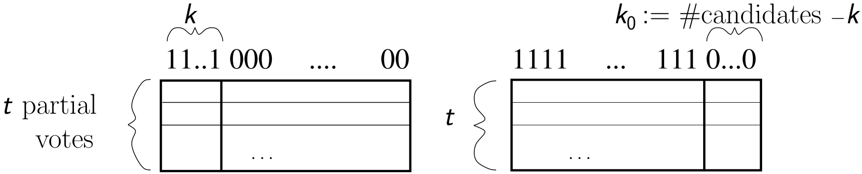
- NP-hard for every fixed $k \in \{2, \ldots, |C| - 2\}$
  [BETZLER AND DORN, MFCS 2009]
- NP-hard for two (or more) partial votes
  [BETZLER ET AL., IJCAI 2009]

Consider the **combined parameter** "number of partial votes" and "number of candidates receiving zero/one point per vote".

Possible Winner for k-approval



Possible Winner for k-approval



Theorem

For k-approval, Possible Winner admits a polynomial-size problem kernel with respect to the parameter (t, k_0).
More precisely, an instance can be reduced to an instance with $O(t \cdot k_0^2)$ candidates and $O(t^2 \cdot k_0^2)$ votes.

Theorem

For k-approval, Possible Winner parameterized by (t, k) is fixed-parameter tractable but does not admit a polynomial-size problem kernel unless coNP $\subseteq$ NP/poly.

Conclusion

In practice:

Data reduction should be applied whenever possible!

In theory:

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

Hardly any results on problem kernels for problems in the voting context, many open questions..

Literature

Talk is based on

- Exact Rank Aggregation Based on Effective Data Reduction [N. Betzler, R. Bredereck, and R. Niedermeier, manuscript]
- On Problem Kernels for Possible Winner Determination Under the k-Approval Protocol [N. Betzler, manuscript]

General literature on problem kernels and parameterized algorithms

- Invitation to data reduction and problem kernelization [J. Guo and R. Niedermeier, ACM SIGACT News 2007]
- Kernelization: New upper and lower bound techniques [H. Bodlaender, IWPEC 2009]
- R. G. Downey and M. R. Fellows, Parameterized Complexity, Springer, 1999
- R. Niedermeier, Invitation to Fixed-Parameter Algorithms, Oxford University Press, 2006