Fixed-Parameter Algorithms for Kemeny Scores

Nadja Betzler

joint work with

Michael R. Fellows, Jiong Guo, Rolf Niedermeier, and Frances A. Rosamond

Friedrich-Schiller-Universität Jena, Germany
University of Newcastle, Australia

AAIM
June 24, 2008

Kemeny Consensus

Election

Set of votes V, set of candidates C.
A vote is 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"?

How to measure the 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:

$$v : a > b > c$$

$$w : c > a > b$$

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

Kemeny Consensus

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

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

Kemeny score of a ranking r

sum of KT-distances between r and all votes

Kemeny Consensus

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

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

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 + Motivation

Kemeny Score

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

Applications:

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

Known results

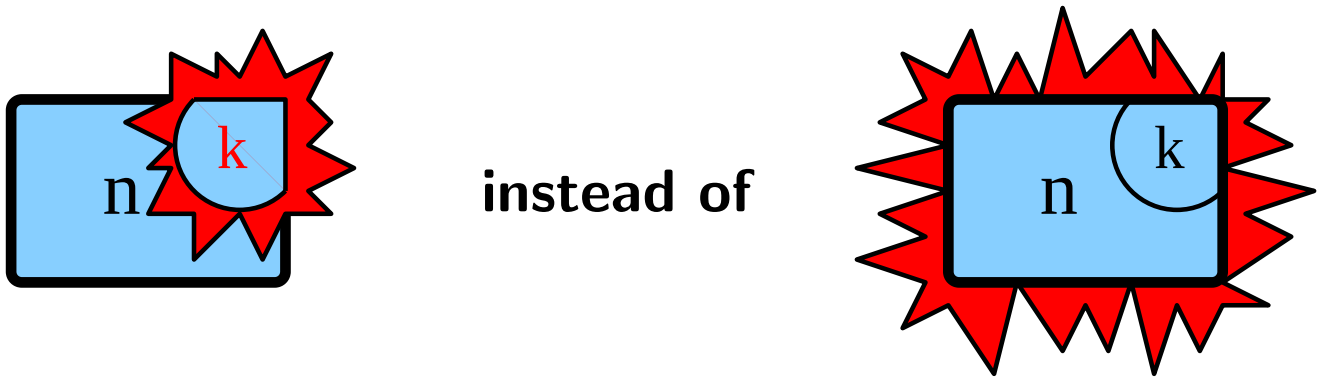
- Kemeny Score is NP-complete (even for 4 votes)
  [Dwork and al., WWW 2001]
- Kemeny Winner is $P_{\|}^{NP}$-complete
  [E. Hemaspaandra and al., TCS 2005]

Algorithms:

- randomized factor 11/7-approximation
  [Ailon and al., STOC 2005]
- factor 8/5-approximation
  [van Zuylen and Williamson, WAOA 2007]
- PTAS [Kenyon-Mathieu and Schudy, STOC 2007]
- Heuristics; greedy, branch and bound
  [Davenport and Kalagnanam AAAI 2004],
  [Conitzer and al. AAAI, 2006]

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.

Results

	Kemeny Score
Kemeny score k	kernelization $O^*(1.53^k)$
Maximum pairwise KT-distance d	$O^*((3d+1)!)$
Number of candidates m	$O^*(2^m)$
Number of votes n [Dwork & al. WWW 2001]	NP-c for $n = 4$

Generalization to

- votes with ties and
- incomplete votes.

Definitions and Observations

Lemma [Truchon, Technical Report 1998]

Let a and b be two candidates in C. If $a > b$ in all votes $v \in V$, then every Kemeny consensus has $a > b$.

Dirty pair

Two candidates a and b form a *dirty pair* if in V there is one vote with $a > b$ and a second vote with $b > a$.

Example:

$$\begin{array}{lccccccc} v_1: & a & > & b & > & c & > & d \\ v_2: & a & > & d & > & b & > & c \\ v_3: & b & > & c & > & a & > & d \end{array}$$

In optimal consensus: "$a > d$" and "$b > c$"
Dirty pairs: $\{a, b\}$, $\{a, c\}$, $\{b, d\}$, and $\{c, d\}$

Search tree I

Trivial search tree:
A dirty pair "increases" the Kemeny score at least by 1.
Branching into dirty pairs results in a search tree of size $O(2^k)$.

Improvement by branching into "dirty triples":

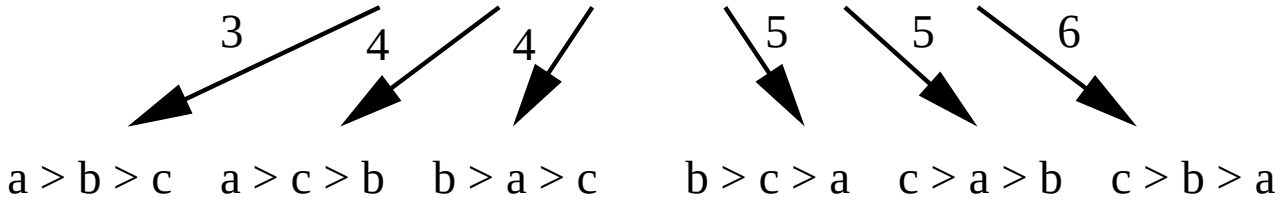
Definition

Three candidates a, b, c form a *dirty triple* if they occur in at least two dirty pairs.

Search tree II

Case 1: dirty triple with 3 dirty pairs

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



branching vector: $(3, 4, 4, 5, 5, 6) \Rightarrow$ branching number: 1.52

Case 2: dirty triple with 2 dirty pairs

branching vector: $(3, 3, 2) \Rightarrow$ branching number: 1.53

Case 3: no more dirty triples

for every dirty pair: make a "majority decision"
(and decrease the parameter accordingly)

Search tree III

Combining search tree algorithm and kernelization:

Theorem

KEMENY SCORE can be solved in $O(1.53^k + m^2 n)$ time, where k denotes the Kemeny score of the given election.

Parameter Maximum KT-distance

Recall:

KT-distance (between two votes v and w)

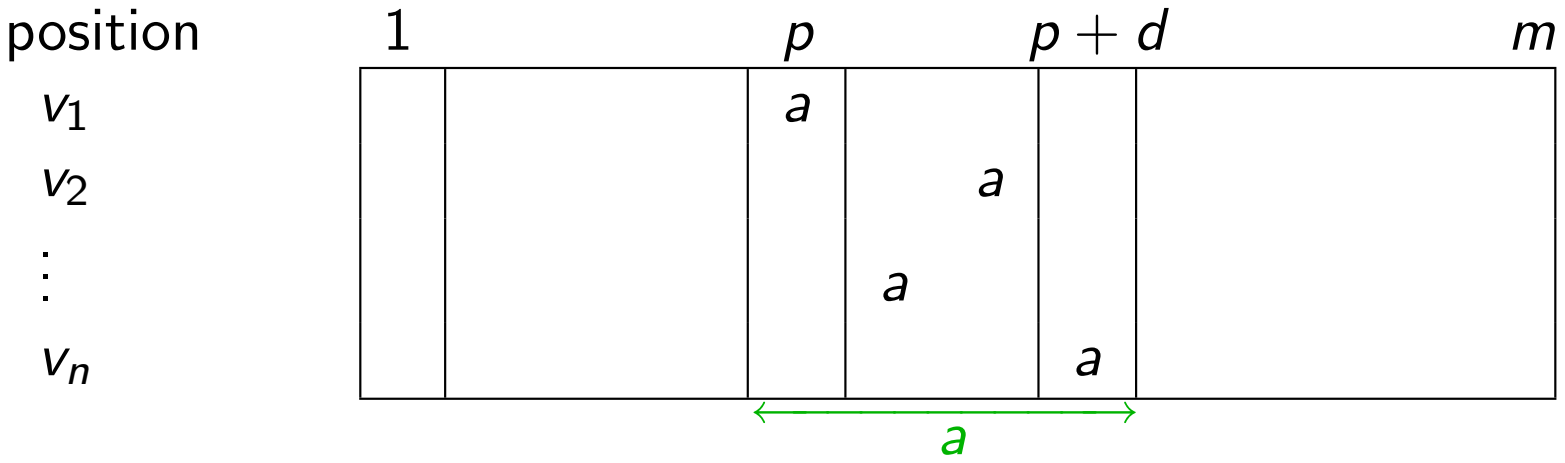
is the number of dirty pairs.

Parameter d is maximum pairwise KT-distance:

$$d := \max_{v,w \in V} \text{KT-dist}(v, w)$$

Basic idea of DP-algorithm

Parameter d: maximum KT-distance between two input votes.

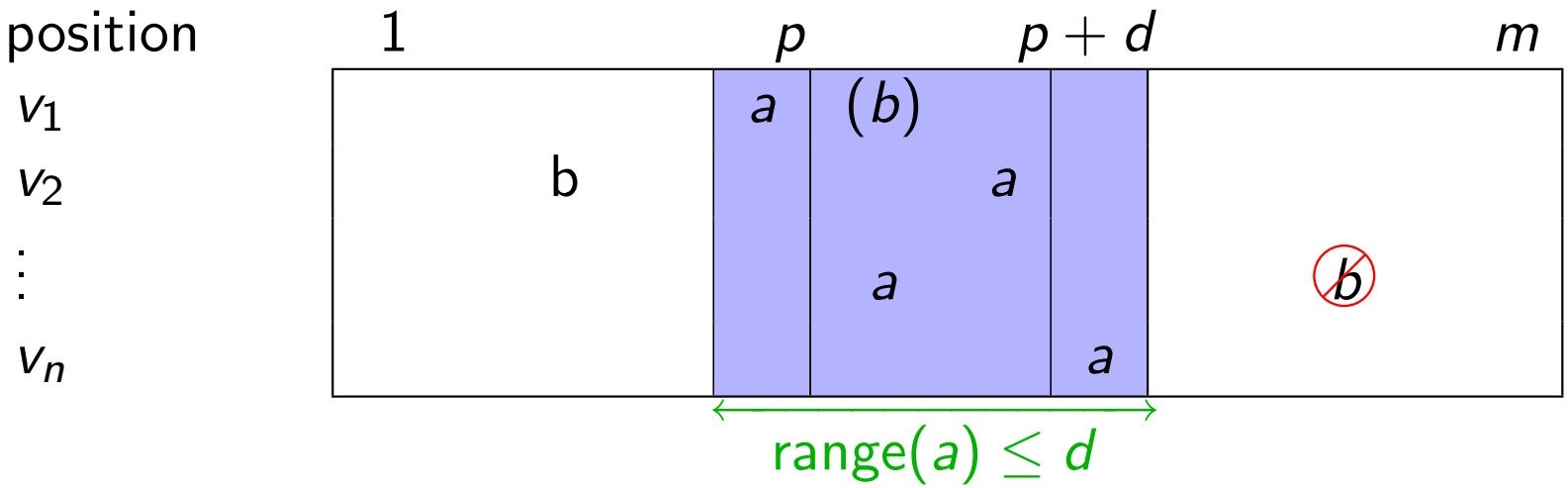


Two fundamental observations:

1. Range of positions of every candidate is bounded by d

Basic idea of DP-algorithm

Parameter d: maximum KT-distance between two input votes.

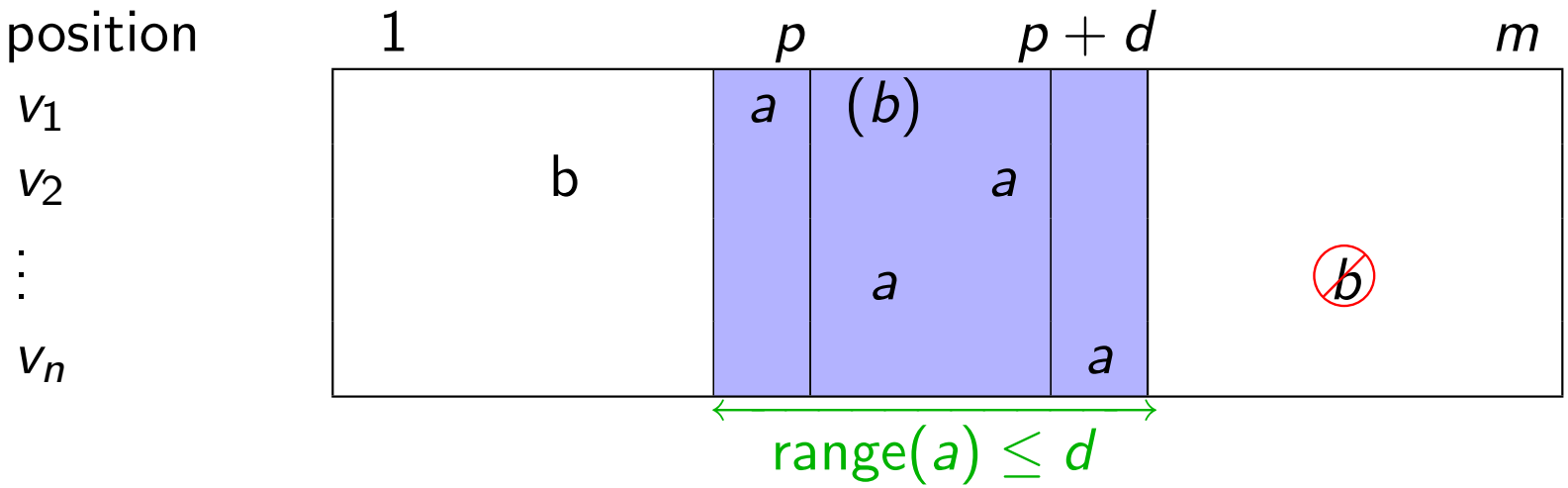


Two fundamental observations:

1. Range of positions of every candidate is bounded by d
   $\Rightarrow$ a block of size $d+1$ separates input votes

Basic idea of DP-algorithm

Parameter d: maximum KT-distance between two input votes.



Two fundamental observations:

1. Range of positions of every candidate is bounded by d
   $\Rightarrow$ a block of size $d + 1$ separates input votes
2. The number of candidates in a block is bounded by $3d + 1$.

Algorithm computes partial local solution for every block and combines them into a Kemeny consensus.

Observation 1

Lemma

In an input instance with maximum KT-instance d, the positions of a candidate in two votes differ by at most d.

Proof by contradiction:

position	1	2	...	p	...	$p+d+1$	...	m
v:				c	x			
w:					x	c		

$> d$

In v : $p-1$ candidates that are better than c

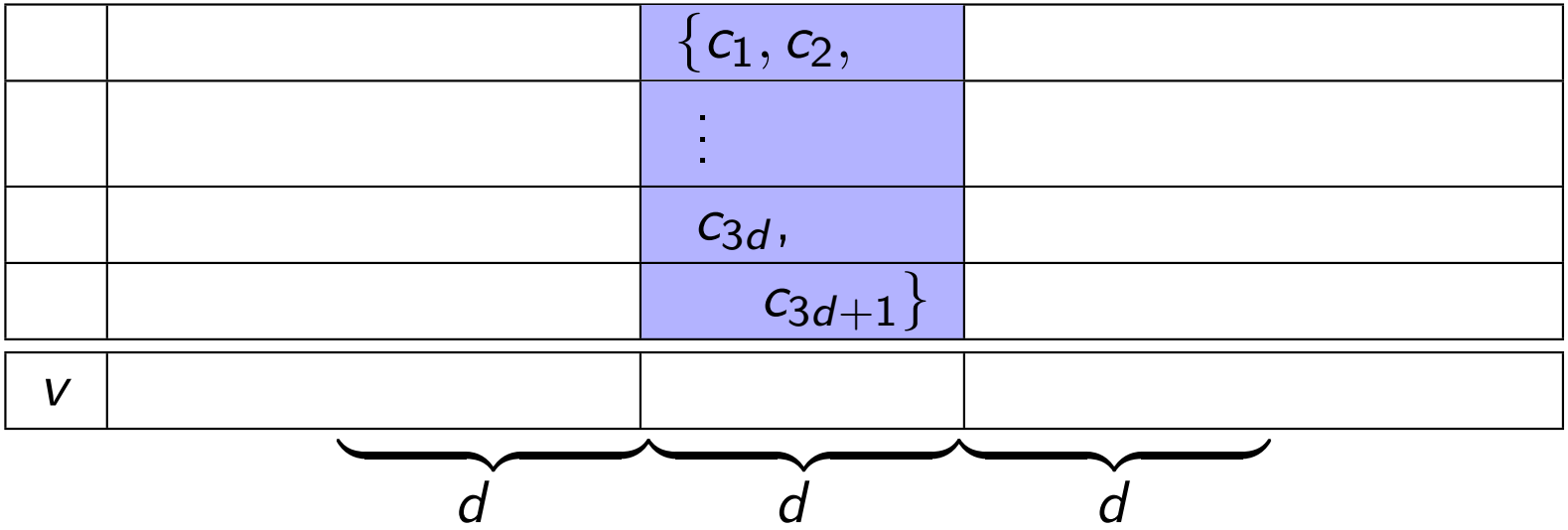
In w : $p+d$ candidates that are better than c

$\Rightarrow$ More than d candidates x with $c < x$ in v and $x < c$ in w

$\Rightarrow$ KT-distance between v and w is more than d $\square$

Observation 2

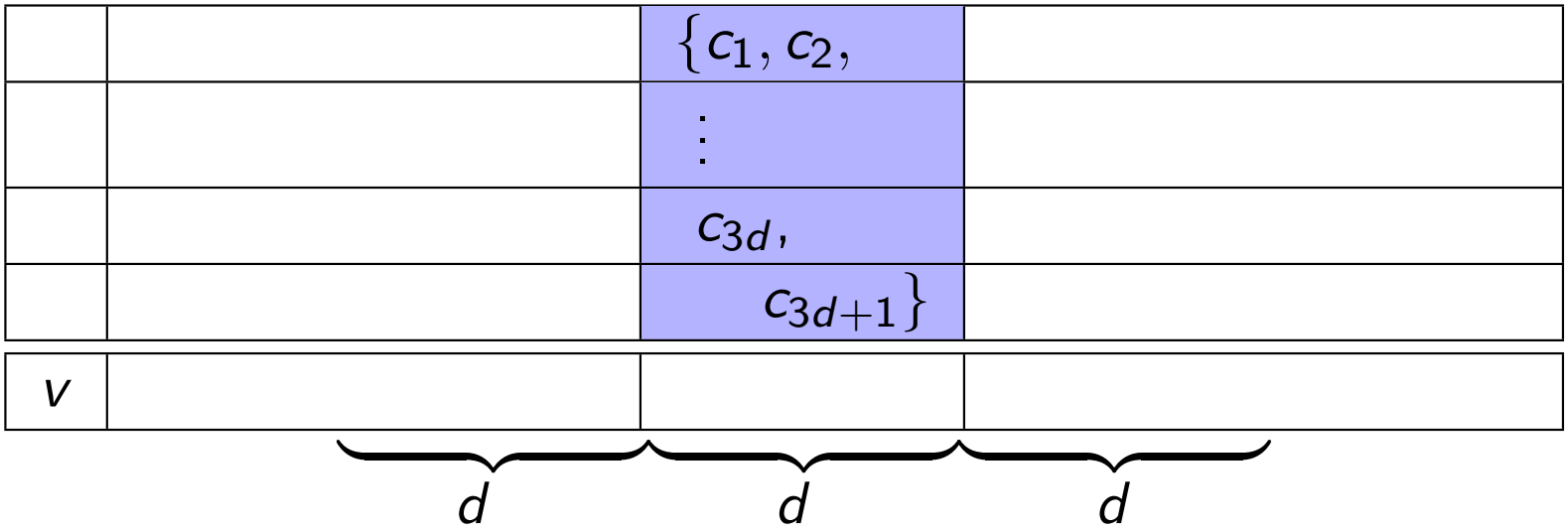
Bound the number of candidates in a block of width d:



Proof: By contradiction, using Observation 1

Observation 2

Bound the number of candidates in a block of width d:



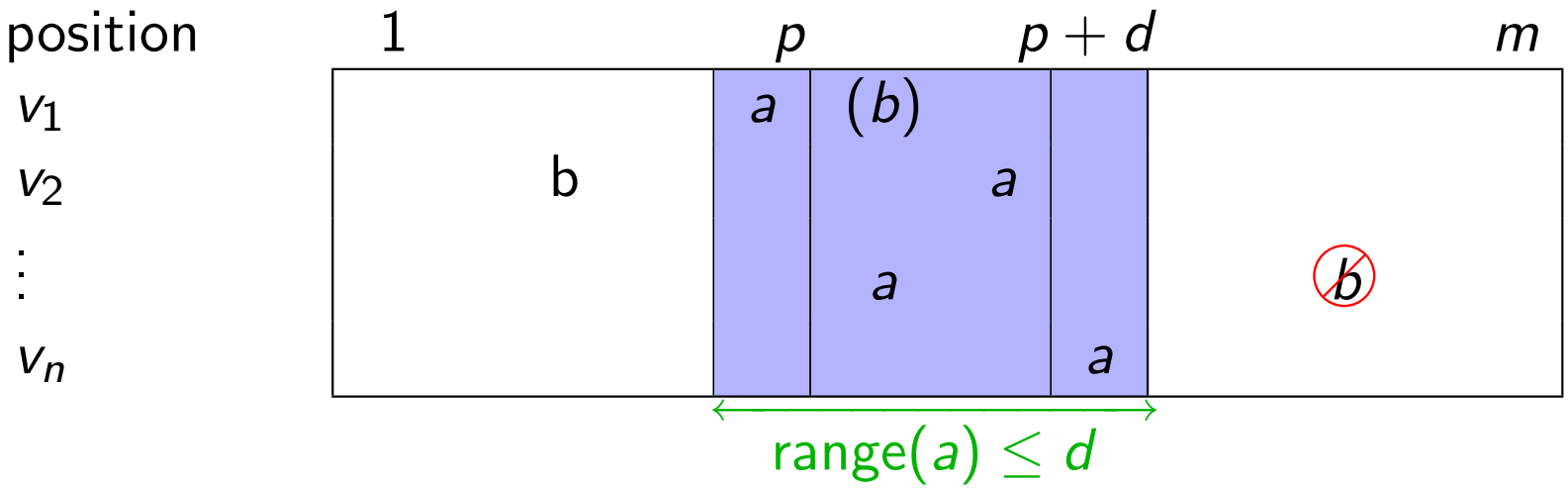
Proof: By contradiction, using Observation 1

Lemma

In an input instance with maximum KT-instance d, every block of width $d+1$ contains at most $3d+1$ candidates.

Basic idea of DP-algorithm

Parameter d: maximum KT-distance between two input votes.



Two fundamental observations:

1. Range of positions of every candidate is bounded by d
   $\Rightarrow$ a block of size $d + 1$ separates input votes
2. The number of candidates in a block is bounded by $3d + 1$.

Algorithm computes partial local solution for every block and combines them into a Kemeny consensus.

Example

v_1:	b	c	a	e	d				
v_2:	b	a	d	c	..	e		...	.
v_3:	b	a	c	d	e				

assume "range of positions" ≤ 3 for every candidate

Table: $T($"partial order", block$)$ = minimum "partial score"

Initialization:

$$4! \begin{cases} T((a > b > c > d), 1) = 2 + 2 + 1 = 5 \\ T((b > a > c > d), 1) = \ldots \\ \vdots \\ T((d > c > b > a), 1) = \ldots \end{cases}$$

Example

v_1:	b	c	a	e	d				
v_2:	b	a	d	c	..	e		...	.
v_3:	b	a	c	d	e				

Update:

$$T((a > c > d > e), 2) = \min \begin{cases} T((a > c > d > \mathbf{b}), 1) + \text{score}(e) \\ T((a > c > \mathbf{b} > d), 1) + \text{score}(e) \\ T((a > \mathbf{b} > c > d), 1) + \text{score}(e) \\ T((\mathbf{b} > a > c > d), 1) + \text{score}(e) \end{cases}$$

Example

v_1:	b	c	a	e	d			
v_2:	b	a	d	c	..	e	...	.
v_3:	b	a	c	d	e			

Update:

$$T((a > c > d > e), 2) = \min \begin{cases} T((a > c > d > \mathbf{b}), 1) + \text{score}(e) \\ T((a > c > \mathbf{b} > d), 1) + \text{score}(e) \\ T((a > \mathbf{b} > c > d), 1) + \text{score}(e) \\ T((\mathbf{b} > a > c > d), 1) + \text{score}(e) \end{cases}$$

Kemeny score is the minimum entry of the last block.

Dynamic Programming

Algorithm:
Iterate from left to right over the blocks of the votes,
try all possible orders of the candidates within a block.

Block size is bounded by $3d + 1$, there are $(3d + 1)!$ possible orders of a block.

Theorem

Kemeny Score can be solved in $O((3d + 1)! \cdot d \cdot \log d \cdot m \cdot n)$ time with d being the maximum KT-distance between two input votes.

Overview

	Kemeny Score	with Ties	Incomplete Votes
k	kernelization $O^*(1.53^k)$	kernelization $O^*(1.76^k)$	$O^*(k! \cdot 4^k)$
d	$O^*((3d+1)!)$	$O^*((6d+2)! \cdot 2^{6d+2})$	NP-c for $d = 0$
m	$O^*(2^m)$	$O^*(2^m)$	$O^*(2^m)$
n	NP-c for $n = 4$	NP-c for $n = 4$	NP-c for $n = 4$

Kemeny score k
Maximum KT-distance d
Number of candidates m
Number of votes n

Further tasks

- Study further (structural) parameterizations
  very recently: FPT-results for average distance as parameter
- Implementation and experimental studies
- NP-hard for 3 votes?