New Complexity Results on Aggregating Lexicographic Preference Trees Using Positional Scoring Rules

Xudong Liu
School of Computing
University of North Florida

Miroslaw Truszczynski
Department of Computer Science
University of Kentucky

6th International Conference on Algorithmic Decision Theory
Saturday, 10/26/2019

The Problems

1. In social choice theory, given a set $\mathcal{X}$ of alternatives, a profile P of votes over $\mathcal{X}$, and a score-based voting rule R, one may ask the following preference aggregation questions:
    - How do we compute the score of an alternative $o \in \mathcal{X}$ in P w.r.t R? (The **score** problem)
    - Who is the winning alternative in P w.r.t R? (The **winner** problem)
    - Given a threshold value h, is there an alternative whose score w.r.t P and R is at least h? (The **evaluation** problem)
2. In knowledge representation and reasoning, there have been proposed numerial often-compact preference models over combinatorial domains of alternatives, such as:
    - Answer set optimization programs (ASO-programs)
    - Ceteris paribus networks (CP-nets)
    - **Lexicographic preference trees (LP-trees)**

Our Contributions

1. We showed that both the winner and the evaluation problems can be solved in polynomial time, when votes are specified as lexicographic preference trees, and the voting rule is $(2^{p-1} \pm f(p))$-Approval.
2. We then showed that, however, these two problems are NP-hard, when the voting rule is b-Borda, a generalized Borda rule.

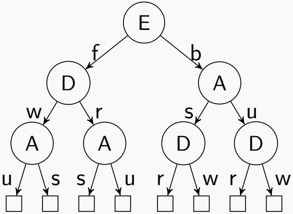
Lexicographic Preference Trees

1. An *LP-tree* over a set $\mathcal{A}$ of p binary attributes $X_1, \ldots, X_p$ is a complete *binary tree*.
2. Each non-leaf node is labeled by an attribute from $\mathcal{A}$.
3. Every non-leaf node has two outgoing edges, each labeled by a distinct value in the domain of the labeling attribute.
4. Each attribute appears **exactly once** on each path from the root to a leaf.
5. Every leaf node is drawn as a box, not labeled.

A Combinatorial Domain: Dinner

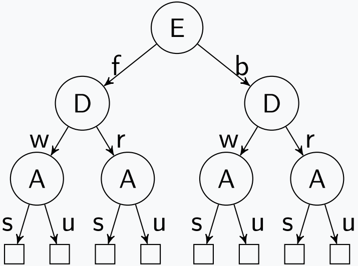
A *combinatorial domain* is given by a set of binary attributes $\mathcal{A}$. The domain implicitly is the Cartesian product of the binary attributes.

1. Appetizer: salad (s) and soup (u)
2. Entree: beef (b) and fish (f)
3. Drink: beer (r) and wine (w)

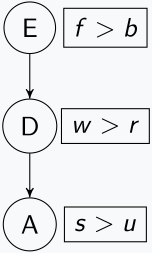


- Domance testing is computationally easy: e.g., sbr$\succ$ubw, decided by Appetizer on the right subtree.
- The LP-trees represent total orders, orders of the leaves
- Computing the rank of a given alternative is easy.
- Computing the alternative at a given rank is easy too.

Unconditional Importance and Unconditional Preference

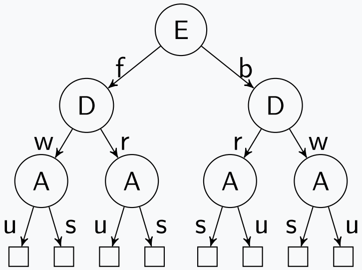


(a) Full

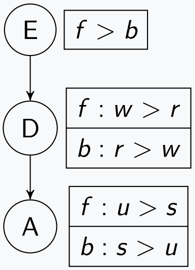


(b) UI-UP

Unconditional Importance and Conditional Preference

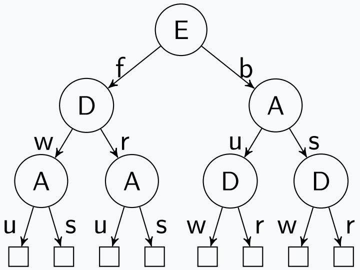


(a) Full

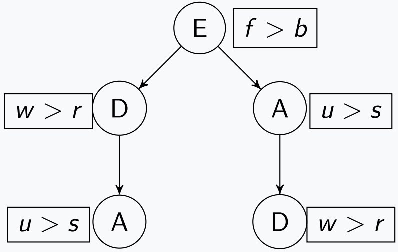


(b) UI-CP

Conditional Importance and Unconditional Preference

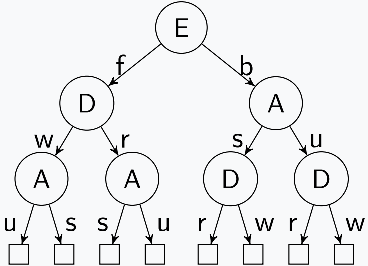


(a) Full

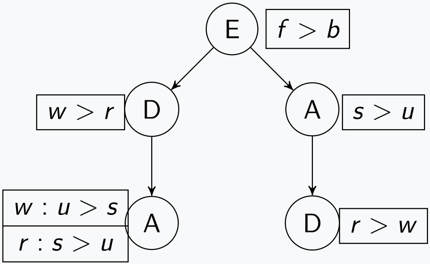


(b) CI-UP

Conditional Importance and Conditional Preference

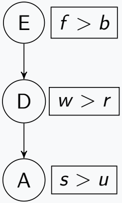


(a) Full

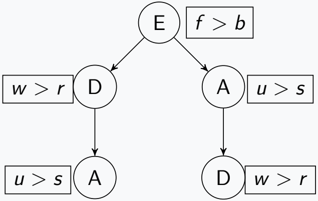


(b) CI-CP

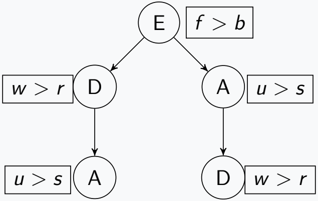
Compactness



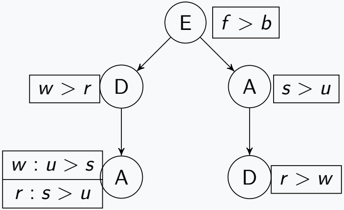
(a) UI-UP



(b) CI-UP



(c) CI-UP



(d) CI-CP

- Domance testing, computing the rank of an alternative, and computing the alternative of a rank remain computationally easy.

Positional Scoring Rules

1. Traditionally, a positional scoring rule is given by a *scoring vector* $w = (w_0, \ldots, w_{n-1})$ such that $w_0 \geq w_1 \geq \ldots w_{n-1}$ and $w_0 > w_{n-1}$.
2. Two such rules we consider:
   - k-Approval: (1,. . . ,1,0,. . . ,0) with k 1's.
   - b-Borda: $(2^{p-b} - 1, 2^{p-b} - 2, \ldots, 1, 0, \ldots, 0)$ where $0 \leq b < p$.
   - Due to the exponential size of these vectors, they are not the input to our problems of study.
3. The score: $s_w(o, P) = \sum_{v \in P} s_w(o, v) = \sum_{v \in P} w_{r(o,v)}$. The winner is the alternative with the maximum score.
   - k-Approval: $s_{kA}(o, v) = 1$, if $r(o, v) < k$; 0, otherwise.
   - b-Borda: $s_{bB}(o, v) = \max\{2^{p-b} - 1 - r(o, v), 0\}$.

The Problems We Studied

- We fix $\mathcal{C} \in \{UI, CI\} - \{UP, CP\}$, and R a positional scoring rule.
  1. Given a profile P of class $\mathcal{C}$ LP-trees, the winner problem asks to compute $\arg\max_{o \in \mathcal{X}} s_R(o, P)$.
  2. Given a profile P of class $\mathcal{C}$ LP-trees and a positive integer threshold h, the evaluation problem asks to decide whether there exists an alternative such that $s_R(o, P) \geq h$.

Results for k-Approval, Where $k = 2^{p-1} + f(p)$

Theorem 1

Let $f(p)$ be a polynomial in p such that $0 < f(p) < 2^{p-1}$ for all $p \geq 1$. The winner problem under k-approval, where $k = 2^{p-1} + f(p)$, for any profile of LP-trees of any class in {UI,CI}-{UP,CP}, can be solved in time polynomial in the size of the profile.

- Clearly, this problem for k-Approval, where k is a constant, is in P.
- This problem for 2^{p-1}-Approval is in P. (Lang, Mengin and Xia, AIJ, 2018)
- This problem for k-Approval is in NP-complete, when $k = \alpha \cdot 2^p$, where α is a rational number of form $a/2^p$ for any integer $1 \leq a < 2^p$, k is not a constant, and $\alpha \neq 1/2$. (Lang, Mengin and Xia, AIJ, 2018)
  - E.g., NP-hard when $k = \frac{3}{8}2^p$ or $\frac{5}{8}2^p$.
  - So where between $\frac{3}{8}2^p$ and $\frac{1}{2}2^p$ does the complexity change?
  - Similarly, where between $\frac{1}{2}2^p$ and $\frac{5}{8}2^p$ does the complexity change?

Results for k-Approval, Where $k = 2^{p-1} + f(p)$

The algorithm to solve the winner problem for $(2^{p-1} + f(p))$-Approval:

1. We write $s_K(o)$ and $s_H(o)$ for the scores of $o \in \mathcal{X}$ for any profile P according to the $(2^{p-1} + f(p))$-approval and 2^{p-1}-approval, respectively.
2. Compute set S of all alternatives o s.t. $s_K(o) > s_H(o)$. (Poly time by taking the union $\bigcup_{T \in P} \{o \in \mathcal{X} : 2^{p-1} < r(o, T) \leq 2^{p-1} + f(p)\}$.)
3. Set $b_{i,0}$ ($b_{i,1}$) to be the number of trees in P with X_i being the root with $0 > 1$ ($1 > 0$, resp.) preference.
4. Compute tuple $(x_1, \ldots, x_p)$, each $x_i = 0$ if $b_{i,0} > b_{i,1}$, $x_i = 1$ if $b_{i,1} > b_{i,0}$, and $x_i = *$, o/w.
5. Pick $\alpha = \arg\max_{o \in S} s_K(o)$.
6. Pick any β that instantiates tuple $(x_1, \ldots, x_p)$.
7. Return $\arg\max_{o \in \{\alpha, \beta\}} s_K(o)$.

Results for k-Approval, Where $k = 2^{p-1} - f(p)$

Theorem 2

Let $f(p)$ be a polynomial in p such that $0 < f(p) < 2^{p-1}$ for all $p \geq 1$. The winner problem under k-approval, where $k = 2^{p-1} - f(p)$, for any profile of LP-trees of any class in $\{$UI,CI$\}$-$\{$UP,CP$\}$, can be solved in time polynomial in the size of the profile.

Results for k-Approval, Where $k = 2^{p-1} - f(p)$

The algorithm to solve the winner problem for $(2^{p-1} - f(p))$-Approval:

1. As before, we write $s_K(o)$ and $s_H(o)$ for the scores of $o \in \mathcal{X}$ for any profile P according to the $(2^{p-1} - f(p))$-approval and 2^{p-1}-approval, respectively.
2. Compute set A of all alternatives o s.t. $s_K(o) < s_H(o)$. (Poly time by taking the union $\bigcup_{T \in P} \{o \in \mathcal{X} : 2^{p-1} - f(p) < r(o, T) \leq 2^{p-1}\}$.)
3. If $|A| = 2^p$, return $\arg\max_{o \in A} s_K(o)$.
4. If $|A| < 2^p$, compute set B of the top $|A| + 1$ alternatives w.r.t their s_H scores, and return $\arg\max_{o \in A \cup B} s_K(o)$.
    - We showed B can be computed by a recursive procedure in poly time.

Results for b-Borda

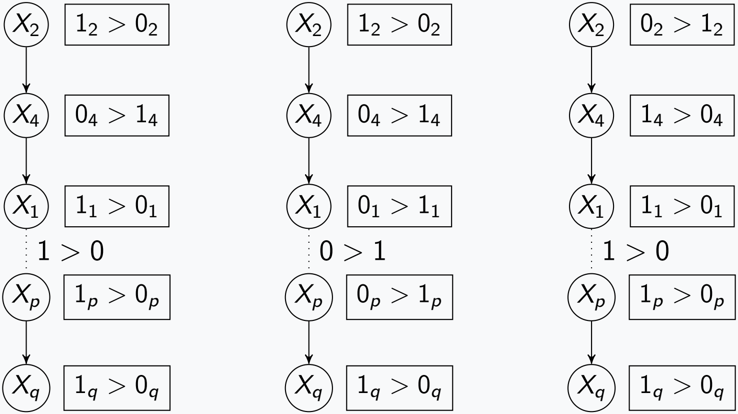
Theorem 3

The evaluation problem under 1-Borda for the class of UI-UP profiles over $p > 1$ binary attributes is NP-complete.

- Note that this problem for 0-Borda, the regular Borda rule, is in P.
- The hardness proof results from a poly time reduction from the NP-complete problem MIN-2SAT: Given a set Φ of 2-clauses $\{C_1, \ldots, C_m\}$ over a set of propositional variables $\{X_1, \ldots, X_p\}$, and a positive integer l ($l \leq n$), decide whether there is a truth assignment that satisfies at most l clauses in Φ.

The Reduction in Proof of Theorem 3

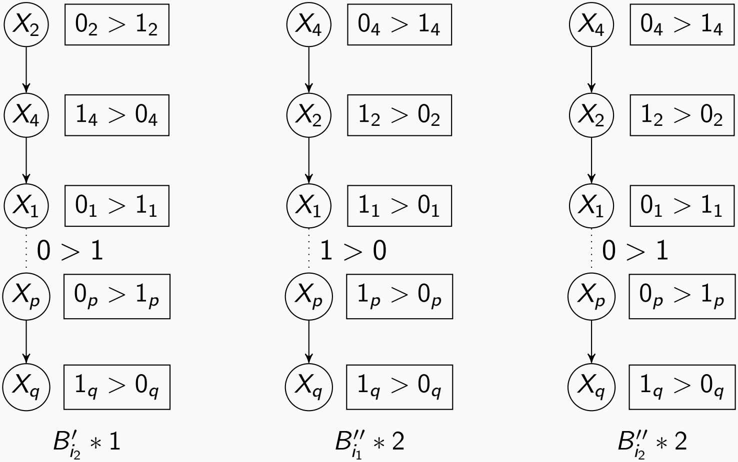
For a 2-clause, e.g., $C_i = \neg X_2 \vee X_4$, we build the profile:



$B_{i_1} * 3$ $B_{i_2} * 3$ $B'_{i_1} * 1$

The Reduction in Proof of Theorem 3

For a 2-clause, e.g., $C_i = \neg X_2 \vee X_4$, we build the profile:



Proof Sketch for Theorem 3

- The six trees are of 3 complementary pairs. E.g.,
  - X_q is a dummy attribute, evaluated to 1 always.
  - If $o \models X_q \wedge \neg C_i$, that is, $o \models X_q \wedge X_2 \wedge \neg X_4$, we have $S_{1B}(o, \{B_{i_1}, B'_{i_1}\}) = 2^p - 1 + 2^{p-1} + 1$.
- The key thing is that all alternatives satisfying C_i score $3 \cdot 2^{p-1}$, and that all alternatives falsifying C_i score $15 \cdot 2^{p-1}$.
- We showed that there exists an outcome over $\mathcal{A}$ with 1-Borda score at least $R = 15 \cdot 2^{p-1} \cdot (m - l) + 3 \cdot 2^{p-1} \cdot l$ if and only if there exists an assignment over I that satisfies at most l clauses in Φ.
- Therefore, MIN-2SAT $\preceq^{poly}$ our problem.

Results for b-Borda

Theorem 4

Let b be an arbitrary integer such that $b > 1$. The evaluation problem under b-Borda for the class of UI-UP profiles over $p > b$ binary attrbutes is NP-complete.

- Hardness is reduced from the problem in Theorem 4.

Results for b-Borda

Theorem 5

Let b be an arbitrary integer such that $b \geq 1$. The evaluation problem under b-Borda for the class of UI-CP (CI-UP and CI-CP, respectively) profiles over $p > b$ binary attributes is NP-complete.

- Hardness is reduced from the same problem for 0-Borda.

Results Summary

Table: Complexity Results

	UP	CP
UI	P (Thms 1&2)	P (Thms 1&2)
CI	P (Thms 1&2)	P (Thms 1&2)

(a) $(2^{p-1} \pm f(p))$-Approval for $0 < f(p) < 2^{p-1}$

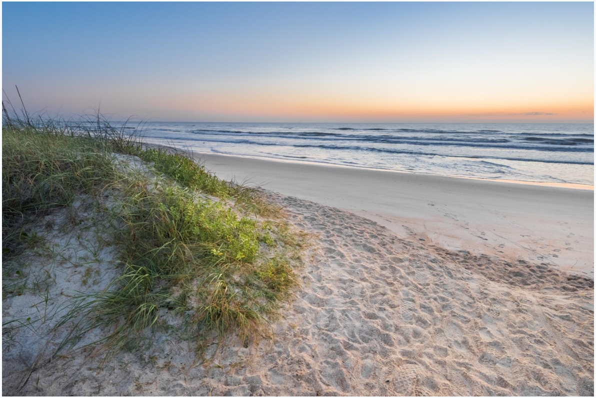
	UP	CP
UI	NPC (Thms 3&4)	NPC (Thm 5)
CI	NPC (Thm 5)	NPC (Thm 5)

(b) b-Borda for $b > 0$

- If the winner problem is in P, so is evaluation.
- If the evaluation problem is NP-complete, winner is NP-hard.

Thank you!

- Questions?
- (We are hiring tenure-track assistant professors to start Fall 2020 at School of Computing, UNF, Jacksonville, FL. Mild weather year round, beautiful beaches, and affordable living and housing.)



St Augustine Beach