

Preference Trees over Combinatorial Domains

Xudong Liu

Ph.D. Candidate

Advisor: Dr. Mirosław Truszczyński

Department of Computer Science

College of Engineering

University of Kentucky

Lexington, KY, USA

Tuesday, 1/19/2016

Preferences Are Ubiquitous

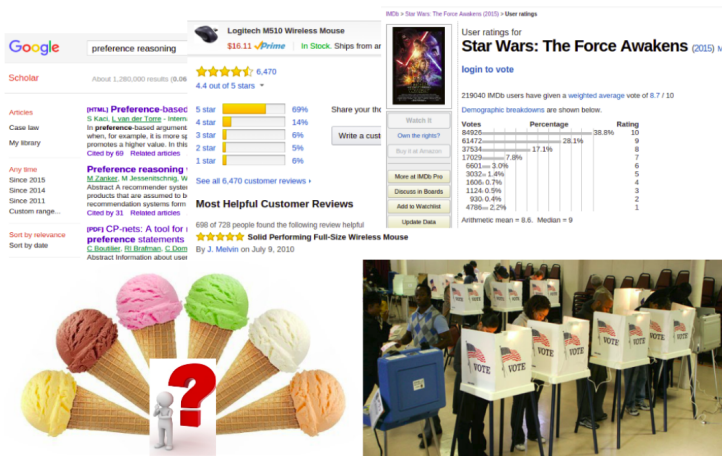


Figure : Preferences of different forms

Describing Preferences

Car1



<mvan, 7m, gray, big, honda, med, med>

Car2



<sedan, 5, blue, med, vw, med, med>



Figure : How to express preferences?

① How will I rate cars?

- For BodyType, I will assign 7 points to minivans, 5 to sedans, ...
- For Color, I will assign 8 points to blue, 4 to gray, ...

② What are the desired properties I see in cars?

- I prefer minivans to sedans, ...
- If minivan, I prefer gray to blue; if sedan, I prefer blue to gray; ...

Describing Preferences

Car1



<mvan, 7m, gray, big, honda, med, med>

Car2



<sedan, 5, blue, med, vw, med, med>



Figure : How to express preferences?

① How will I rate cars? (**Quantitative**)

- For BodyType, I will assign 7 points to minivans, 5 to sedans, ...
- For Color, I will assign 8 points to blue, 4 to gray, ...

② What are the desired properties I see in cars? (**Qualitative**)

- I prefer minivans to sedans, ...
- If minivan, I prefer gray to blue; if sedan, I prefer blue to gray; ...

Binary Relations

Let O be a set of objects. A *binary relation* R over O is a collection of ordered pairs of objects in O ; that is,

$$R \subseteq O \times O.$$

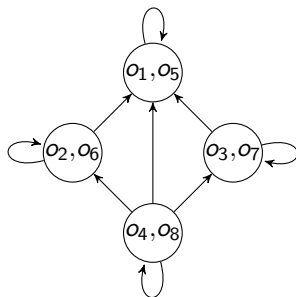
Properties of binary relations related to preferences:

- ➊ Reflexivity: $\forall o \in O, (o, o) \in R$.
- ➋ Irreflexivity: $\forall o \in O, (o, o) \notin R$.
- ➌ Totality: $\forall o_1, o_2, (o_1, o_2) \in R$ or $(o_2, o_1) \in R$.
- ➍ Transitivity: $\forall o_1, o_2, o_3$, if $(o_1, o_2) \in R$ and $(o_2, o_3) \in R$, then $(o_1, o_3) \in R$.
- ➎ Symmetry: $\forall o_1, o_2$, if $(o_1, o_2) \in R$, then $(o_2, o_1) \in R$.
- ➏ Antisymmetry: $\forall o_1, o_2$, if $(o_1, o_2) \in R$ and $(o_2, o_1) \in R$, then $o_1 = o_2$.

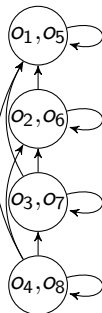
Relations and Orderings

Orderings

\succeq is a *partial preorder* if it is reflexive and transitive, a *total preorder* if it is a partial preorder and total, a *partial order* if it is a partial preorder and antisymmetric, and a *total order* if it is a partial order and total.



(a) partial preorder

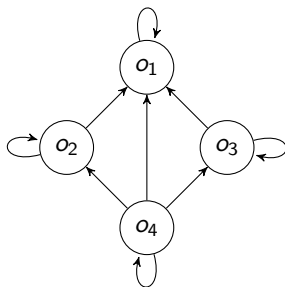


(b) total preorder

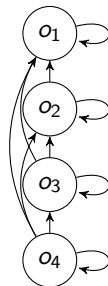
Relations and Orderings

Orderings

\succeq is a *partial preorder* if it is reflexive and transitive, a *total preorder* if it is a partial preorder and total, a *partial order* if it is a partial preorder and antisymmetric, and a *total order* if it is a partial order and total.



(a) partial order

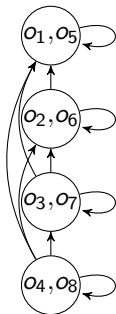


(b) total order

Relations and Orderings

Preference Relations

Let \succeq be a preference relation that is a total preorder over O . We say that o_1 is *weakly preferred* to o_2 if $o_1 \succeq o_2$, that o_1 is *strictly preferred* (\succ) to o_2 if $o_1 \succeq o_2$ and $o_2 \not\succeq o_1$, and that o_1 is *equivalent* (\approx) with o_2 if $o_1 \succeq o_2$ and $o_2 \succeq o_1$.



(a) total preorder

$$o_1 \succ o_5,$$

$$o_2 \succ o_4,$$

$$o_4 \approx o_8,$$

(b) preferences

Combinatorial Domains

Let \mathcal{I} be a finite set of attributes $\{X_1, \dots, X_p\}$, associated with a set of finite domains $\{Dom(X_1), \dots, Dom(X_p)\}$ for each attribute X_i . A *combinatorial domain* $CD(\mathcal{I})$ is a set of *outcomes* described by combinations of values from $Dom(X_i)$:

$$CD(\mathcal{I}) = \prod_{X_i \in \mathcal{I}} Dom(X_i).$$

Combinatorial Domains: Example

Domain of cars over set \mathcal{I} of p binary attributes:

① **BodyType**: {mvan, sedan}.

② **Capacity**: {5, 7m}.

③ **Color**: {blue, gray}.

⋮

$$CD(\mathcal{I}) = \underbrace{\{\langle \text{sedan}, 5, \text{blue}, \dots \rangle, \langle \text{mvan}, 7\text{m}, \text{gray}, \dots \rangle, \dots\}}_{2^p \text{ outcomes, too many!}}.$$

Combinatorial Domains: Example

Domain of cars:

- ① **BodyType**: {mvan, sedan, sport, suv}.
- ② **Capacity**: {2, 5, 7m}.
- ③ **Color**: {black, blue, gray, red, white}.
- ④ **LuggageSize**: {big, med, small}.
- ⑤ **Make**: {bmw, ford, honda, vw}.
- ⑥ **Price**: {low, med, high, vhigh}.
- ⑦ **Safety**: {low, med, high}.

Single Agent

Car1



<mvan, 7m, gray, big, honda, med, med>

Car2



<sedan, 5, blue, med, vw, med, med>



Figure : Dominance and Optimization



Figure : Social Choice and Welfare

Research Problems of Interest

- ① Preference representation formalisms to compactly model qualitative preferences over combinatorial domains.
- ② Preference elicitation and learning methods to cast preferences of agents as a theory in a preference formalism.
- ③ Preference reasoning tasks:
 - Dominance and optimization
 - Manipulation: better off by misreporting preferences.

Q: How do we compactly represent qualitative preferences over combinatorial domains?

- ① Preference Trees (P-trees)^{1,13}
- ② Partial Lexicographic Preference Trees (PLP-trees)⁸
- ③ Lexicographic Preference Trees (LP-trees)^{4,12}

¹Niall M Fraser. "Ordinal preference representations". In: Theory and Decision (1994)

²Xudong Liu and Mirosław Trzuszczynski. "Preference Trees: A Language for Representing and Reasoning about Qualitative Preferences". In: Proceedings of the 8th Multidisciplinary Workshop on Advances in Preference Handling (MPREF). 2014

³Xudong Liu and Mirosław Trzuszczynski. "Learning Partial Lexicographic Preference Trees over Combinatorial Domains". In: Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI). 2015

⁴Richard Booth et al. "Learning conditionally lexicographic preference relations". In: ECAI. 2010

⁵Xudong Liu and Mirosław Trzuszczynski. "Aggregating Conditionally Lexicographic Preferences Using Answer Set Programming Solvers". In: Proceedings of the 3rd International Conference on Algorithmic Decision Theory (ADT). 2013

Preference Learning

Q: How do we learn predictive qualitative preference models over combinatorial domains?

- ① Partial Lexicographic Preference Trees (PLP-trees)^{6,7,8}
 - Active and passive learning
 - Compute a (possibly small) PLP-tree consistent with all the data
 - Compute a PLP-tree that agrees with the data as much as possible
- ② Preference Forests⁹
- ③ Preference Approximation¹⁰

⁶Michael Schmitt and Laura Martignon. "On the complexity of learning lexicographic strategies". In: The Journal of Machine Learning Research (2006)

⁷József Dombi, Csanád Imreh, and Nándor Vincze. "Learning lexicographic orders". In: European Journal of Operational Research (2007)

⁸Xudong Liu and Mirosław Truszczyński. "Learning Partial Lexicographic Preference Trees over Combinatorial Domains". In: Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI). 2015

⁹Xudong Liu and Mirosław Truszczyński. "Learning Preference Trees and Forests". In: Preparation

¹⁰Xudong Liu and Mirosław Truszczyński. "Approximating Conditional Preference Networks Using Lexicographic Preference Trees". In: Preparation

Q: How do we reason about preferences over combinatorial domains?

① Preference Optimization^{11,12,13,14}:

- Dominance testing: $o_1 \succeq_P o_2$?
- Optimality testing: $o_1 \succeq_P o_2$ for all $o_2 \neq o_1$?
- Optimality computing: what is the optimal outcome wrt P ?
- Preference aggregation: which candidate wins the election?

② Preference Misrepresentation^{15,16}:

- Manipulation

¹¹Jérôme Lang, Jérôme Mengin, and Lirong Xia. "Aggregating Conditionally Lexicographic Preferences on Multi-issue Domains". In: CP. 2012

¹²Xudong Liu and Miroslaw Truszczynski. "Aggregating Conditionally Lexicographic Preferences Using Answer Set Programming Solvers". In: Proceedings of the 3rd International Conference on Algorithmic Decision Theory (ADT). 2013

¹³Xudong Liu and Miroslaw Truszczynski. "Preference Trees: A Language for Representing and Reasoning about Qualitative Preferences". In: Proceedings of the 8th Multidisciplinary Workshop on Advances in Preference Handling (MPREF). 2014

¹⁴Xudong Liu and Miroslaw Truszczynski. "Reasoning with Preference Trees over Combinatorial Domains". In: Proceedings of the 4th International Conference on Algorithmic Decision Theory (ADT). 2015

¹⁵Felix Brandt, Vincent Conitzer, and Ulle Endriss. "Computational social choice". In: Multiagent systems (2012)

¹⁶Xudong Liu and Miroslaw Truszczynski. "Complexity of Manipulation in Elections Where Votes Are Lexicographic Preference Trees". In: Preparation

Q: What fields can we apply preferences to?

- ① Role-playing Games:
 - Hedonic games¹⁷
- ② Automated Planning and Scheduling:
 - Trip planning¹⁸
- ③ Data-Driven Decision Making:
 - Predictive models¹⁹

¹⁷Matthew Spradling et al. "Roles and Teams Hedonic Game". In: Proceedings of the 3rd International Conference on Algorithmic Decision Theory (ADT). 2013

¹⁸Xudong Liu et al. "On Personalizability and Extensibility of Multi-Modal Trip Planning". In: PARC Symposium. 2015

¹⁹Xudong Liu and Mirosław Trzuszczynski. "Learning Preference Trees and Forests". In: Preparation

- ① The languages of PLP-trees and LP-trees
- ② Learning preference models in case of PLP-trees
- ③ Reasoning with preferences:
 - Computing winners and “strong” outcomes when votes are LP-trees
 - Application in trip planning
- ④ Future research directions

- 1 The languages of PLP-trees and LP-trees
- Learning preference models in case of PLP-trees
- Reasoning with preferences:
 - Computing winners and “strong” outcomes when votes are LP-trees
 - Application in trip planning
- Future research directions

Example: The Cars Domain

- ① **BodyType(B)**: {mvan, sedan, sport, suv}.
- ② **Capacity(C)**: {2, 5, 7m}.
- ③ **Color(O)**: {black, blue, gray, red, white}.
- ④ **LuggageSize(L)**: {big, med, small}.
- ⑤ **Make(M)**: {bmw, ford, honda, vw}.
- ⑥ **Price(P)**: {low, med, high, vhigh}.
- ⑦ **Safety(S)**: {low, med, high}.

Partial Lexicographic Preference Trees (PLP-Trees)

A *PLP-tree* over $CD(\mathcal{I})$ is a tree, where

- 1 every non-leaf node t is labeled with an attribute $Attr(t)$ in \mathcal{I} ,
- 2 every non-leaf node t has $|Dom(Attr(t))|$ outgoing edges labeled with a value of $Attr(t)$, and
- 3 every attribute appears *at most* once on every branch.

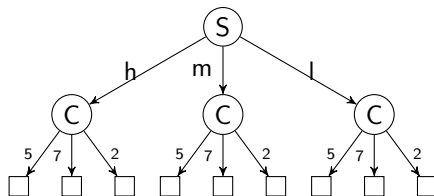


Figure : A PLP-tree over cars

Partial Lexicographic Preference Trees (PLP-Trees)

Car1



<mvan, 7m, gray, big, honda, med, med>

Car2



<sedan, 5, blue, med, vw, med, med>

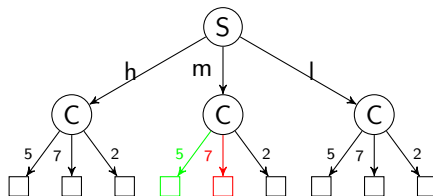
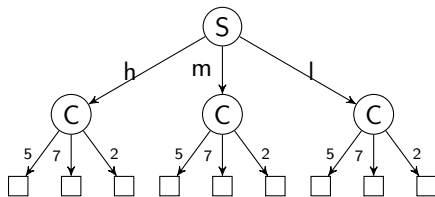


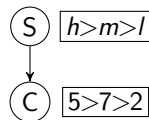
Figure : A PLP-tree over cars

Car2 \succ Car1

Compact Representations of PLP-Tree



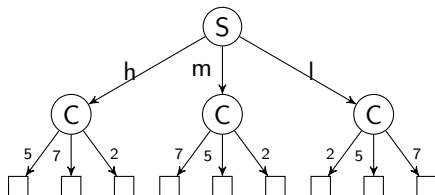
(a) Full



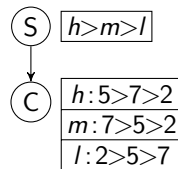
(b) Compact

Figure : Unconditional Importance & Unconditional Preference (UIUP)

Compact Representations of PLP-Tree



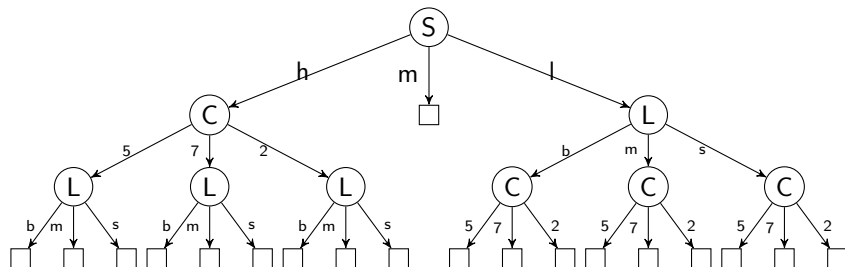
(a) Full



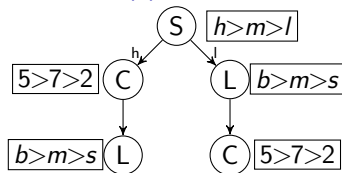
(b) Compact

Figure : Unconditional Importance & Conditional Preference (UICP)

Compact Representations of PLP-Tree



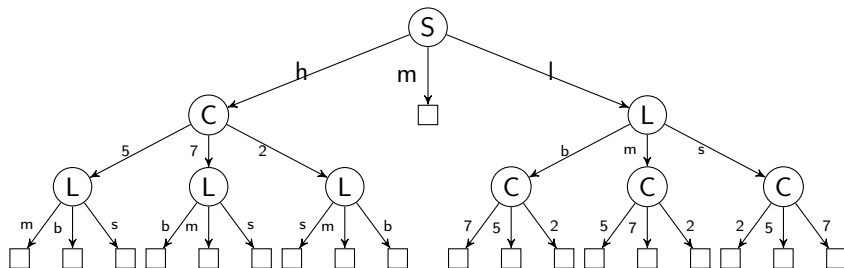
(a) Full



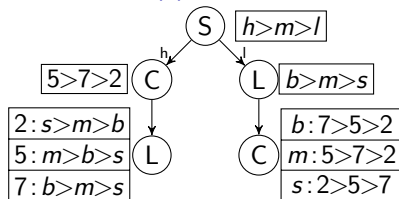
(b) Compact

Figure : Conditional Importance & Unconditional Preference (CIUP)

Compact Representations of PLP-Tree



(a) Full



(b) Compact

Figure : Conditional Importance & Conditional Preference (CICP)

Lexicographic Preference Trees (LP-Trees)

- ① An *LP-tree* \mathcal{L} over $CD(\mathcal{I})$ is a PLP-tree, where
- each attribute appears **exactly once** on every path from the root to a leaf.
 - Unlike PLP-trees, an LP-tree induces a total order.

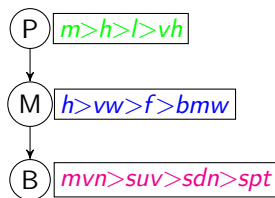
- The languages of PLP-trees and LP-trees
- **2 Learning preference models in case of PLP-trees**
- Reasoning with preferences:
 - Computing winners and “strong” outcomes when votes are LP-trees
 - Application in trip planning
- Future research directions

Learning PLP-trees

Consistent Learning (CONSLearn)

Given an example set \mathcal{E} , decide whether there exists a PLP-tree T (of a particular type) such that T is consistent with \mathcal{E} .

$(\langle \text{sdn}, 5, \text{blk}, m, h, m, m \rangle, \langle \text{suv}, 7m, \text{wht}, b, f, m, m \rangle)$
 $(\langle \text{spt}, 2, \text{wht}, s, \text{bmw}, h, h \rangle, \langle \text{spt}, 2, \text{wht}, s, \text{bmw}, \text{vh}, h \rangle)$
 $(\langle \text{mvn}, 7m, \text{gry}, b, f, m, m \rangle, \langle \text{sdn}, 5, \text{bl}, m, f, m, m \rangle)$



UIUP tree

Learning PLP-trees

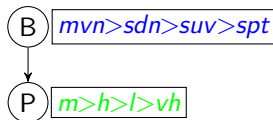
Small Learning (SMALLLEARN)

Given an example set \mathcal{E} and a positive integer l ($l \leq |\mathcal{E}|$), decide whether there exists a PLP-tree T (of a particular type) such that T is consistent with \mathcal{E} and $|T| \leq l$.

($\langle \text{sdn}, 5, \text{blk}, m, h, m, m \rangle, \langle \text{suv}, 7m, \text{wht}, b, f, m, m \rangle$)

($\langle \text{spt}, 2, \text{wht}, s, \text{bmw}, h, h \rangle, \langle \text{spt}, 2, \text{wht}, s, \text{bmw}, \text{vh}, h \rangle$)

($\langle \text{mvn}, 7m, \text{gry}, b, f, m, m \rangle, \langle \text{sdn}, 5, \text{bl}, m, f, m, m \rangle$)



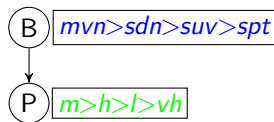
UIUP tree

Learning PLP-trees

Maximal Learning (MAXLEARN)

Given an example set \mathcal{E} and a positive integer k ($k \leq m$), decide whether there exists a PLP-tree T (of a particular type) such that T satisfies at least k examples in \mathcal{E} .

($\langle \text{sdn}, 5, \text{blk}, m, h, m, m \rangle, \langle \text{suv}, 7m, \text{wht}, b, f, m, m \rangle$)
($\langle \text{spt}, 2, \text{wht}, s, \text{bmw}, h, h \rangle, \langle \text{spt}, 2, \text{wht}, s, \text{bmw}, v, h \rangle$)
($\langle \text{mvn}, 7m, \text{gry}, b, f, m, m \rangle, \langle \text{sdn}, 5, \text{bl}, m, f, m, m \rangle$)
($\langle \text{suv}, 7m, \text{gry}, b, v, v, h, m \rangle, \langle \text{suv}, 7m, \text{gry}, b, v, v, h, m \rangle$)

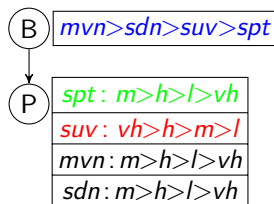


UIUP tree

Consistent Learning (CONSLearn)

Given an example set \mathcal{E} , decide whether there exists a PLP-tree T (of a particular type) such that T is consistent with \mathcal{E} .

($\langle \text{sdn}, 5, \text{blk}, \text{m}, \text{h}, \text{m}, \text{m} \rangle, \langle \text{suv}, 7\text{m}, \text{wht}, \text{b}, \text{f}, \text{m}, \text{m} \rangle$)
($\langle \text{spt}, 2, \text{wht}, \text{s}, \text{bmw}, \text{h}, \text{h} \rangle, \langle \text{spt}, 2, \text{wht}, \text{s}, \text{bmw}, \text{vh}, \text{h} \rangle$)
($\langle \text{mvn}, 7\text{m}, \text{gry}, \text{b}, \text{f}, \text{m}, \text{m} \rangle, \langle \text{sdn}, 5, \text{bl}, \text{m}, \text{f}, \text{m}, \text{m} \rangle$)
($\langle \text{suv}, 7\text{m}, \text{gry}, \text{b}, \text{vw}, \text{vh}, \text{m} \rangle, \langle \text{suv}, 7\text{m}, \text{gry}, \text{b}, \text{vw}, \text{h}, \text{m} \rangle$)



UICP tree

Computational Complexity

- ① P , NP , $coNP$: We typically believe that $P \subset NP$ and $P \subset coNP$.
- ② Δ_2^P : P^{NP} , Σ_2^P : NP^{NP} , and Π_2^P : $coNP^{NP}$.
- ③ C -complete: hardest decision problems in class C .

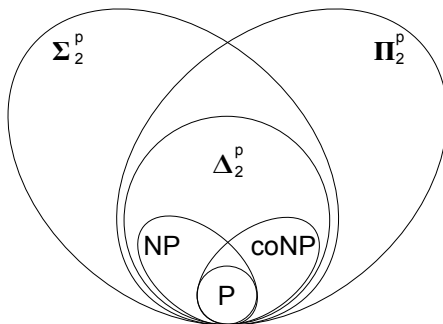


Figure : Computational complexity diagram

Complexity Results on PLP-trees

	UP	CP
UI	P	P
CI	NPC ²⁰	P

(a) CONSLearn

	UP	CP
UI	NPC	NPC
CI	NPC	NPC

(b) SMALLLearn

	UP	CP
UI	NPC ²¹	NPC
CI	NPC	NPC

(c) MAXLearn

Figure : Complexity results for learning PLP-trees

²⁰Booth et al., *Learning Conditionally Lexicographic Preference Relations*, 2010.

²¹Schmitt and Martignon, *On the Complexity of Learning Lexicographic Strategies*, 2006.

Experimentation

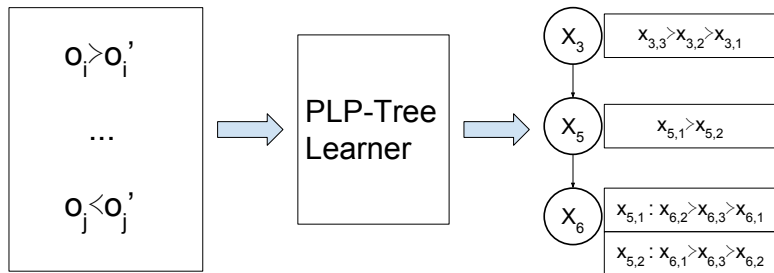


Figure : PLP-tree learning system

Datasets

Dataset	#Attributes	#Outcomes	#Examples
BreastCancerWisconsin	9	270	9009
CarEvaluation	6	1728	682721
CreditApproval	10	520	66079
GermanCredit	10	914	172368
Ionosphere	10	118	3472
MammographicMass	5	62	792
Mushroom	10	184	8448
Nursery	8	1266	548064
SPECTHeart	10	115	3196
TicTacToe	9	958	207832
Vehicle	10	455	76713
Wine	10	177	10322

Figure : Preference Learning Library²²

²²<http://www.cs.uky.edu/~liu/preflearnlib.php>

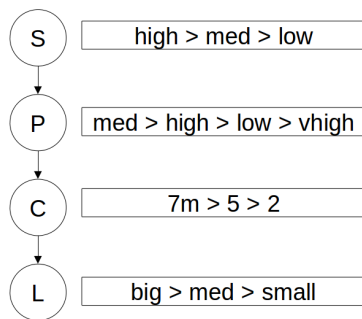


Figure : Unconditional Importance & Unconditional Preference (UIUP)

PLP-Trees To Learn

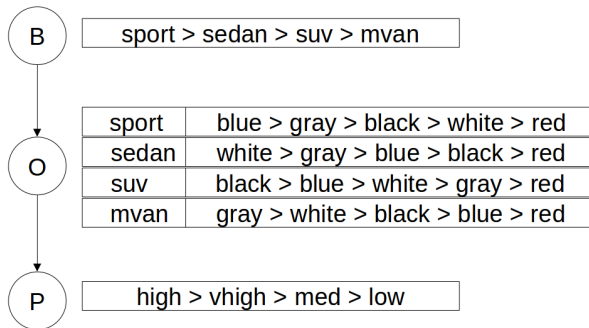


Figure : UICP with at most 1 parent (UICP-1)

PLP-Trees To Learn

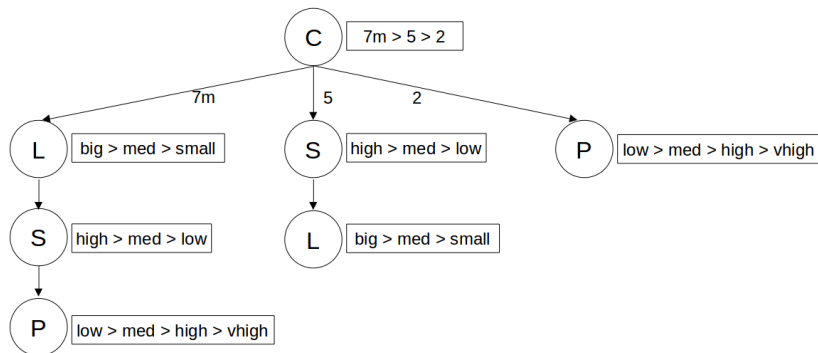


Figure : CIUP with 1 split at the root (CIUP-1)

PLP-Trees To Learn

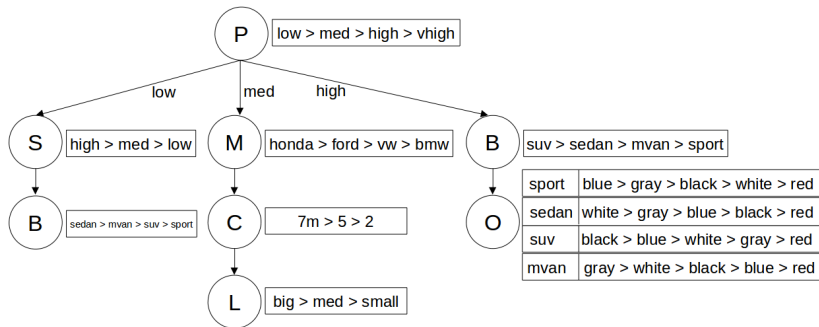
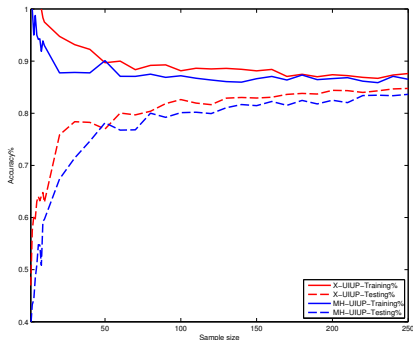


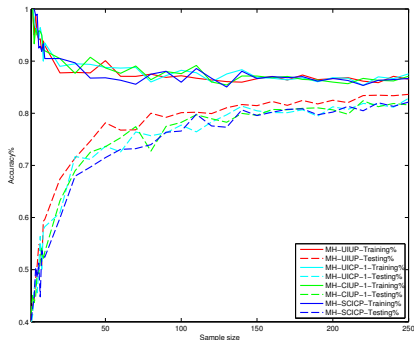
Figure : Simple CICP (SCICP)

Experimental Results: CarEvaluation²³

#attributes:6, #outcomes:1728, #examples:682721



(a) Compare exact & greedy heuristic



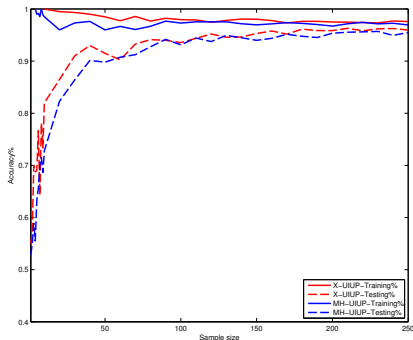
(b) Greedy heuristic

Figure : Learning curves solving MAXLEARN

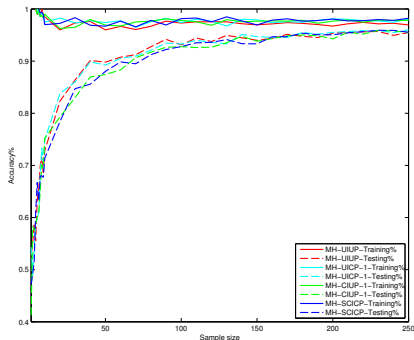
²³<http://www.cs.uky.edu/~liu/preflearnlib.php>

Experimental Results: Wine²⁴

#attributes:10, #outcomes:177, #examples:10322



(a) Compare exact & greedy heuristic



(b) Greedy heuristic

Figure : Learning curves solving MAXLEARN

²⁴<http://www.cs.uky.edu/~liu/preflearnlib.php>

- The languages of PLP-trees and LP-trees
- Learning preference models in case of PLP-trees
- ③ Reasoning with preferences:
 - Computing winners and “strong” outcomes when votes are LP-trees
 - Application in trip planning
- Future research directions

Positional Scoring Rules

- k -approval: $(1, \dots, 1, 0, \dots, 0)$ with k being the number of 1's and $m - k$ the number of 0's where $m = 2^p$.
- (k, l) -approval: $(a, \dots, a, b, \dots, b, 0, \dots, 0)$, where a and b are constants ($a > b$) and the numbers of a 's and b 's equal to k and l , respectively.
- b -Borda: $(b, b - 1, \dots, 0)$, where if $b > m - 1$, b -Borda is reduced to the regular Borda rule with $(m - 1, m - 2, \dots, 1, 0)$.

The Evaluation and Winner Problems

The Evaluation Problem

Let r be a positional scoring rule with a scoring vector w , \mathcal{C} a class of LP-trees. Given a \mathcal{C} -profile P of n LP-trees over p attributes and a positive integer R , the *evaluation* problem is to decide whether there exists an alternative $o \in \mathcal{X}$ such that $s_w(o, P) \geq R$.

The Winner Problem

Let r be a positional scoring rule with a scoring vector w , \mathcal{C} a class of LP-trees. Given a \mathcal{C} -profile P of n LP-trees over p attributes, the *winner* problem is to compute an alternative $o \in \mathcal{X}$ with the maximum score $s_w(o, P)$.

Complexity of the Evaluation Problem: k -Approval

	UP	CP
UI	P	P
CI	P	P

(a) $k = 2^{p-1} \pm f(p)$, $f(p)$ is a poly

	UP	CP
UI	NPC	NPC
CI	NPC	NPC

(b) $k = 2^{p-c}$, $c > 1$ is a const

Figure : k -Approval

Complexity of the Evaluation Problem: (k, l) -Approval

	UP	CP
UI	P	P
CI	P	P

(a) $k = l = 2^{p-1}$

	UP	CP
UI	NPC	NPC
CI	NPC	NPC

(b) $k = l = 2^{p-c}$, $c > 1$ is a const

Figure : (k, l) -Approval

Complexity of the Evaluation Problem: b -Borda

	UP	CP
UI	P	NPC
CI	NPC	NPC

(a) $b = 2^p - 1$

	UP	CP
UI	NPC	NPC
CI	NPC	NPC

(b) $b = 2^{p-c} - 1$, $c \geq 1$ is a const

Figure : b -Borda

Modeling the Problems in ASP

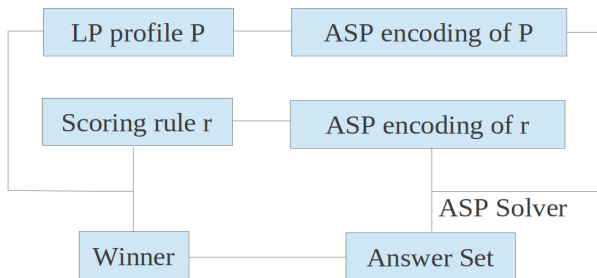


Figure : The winner problem

- Solvers: *clingo*²⁵, *clingcon*²⁶

²⁵M. Gebser et al. "Potassco: The Potsdam Answer Set Solving Collection". In: AI Communications (2011)

²⁶Max Ostrowski and Torsten Schaub. "ASP modulo CSP: The clingcon system". In: TPLP (2012)

Modeling the Problems in W-MAXSAT

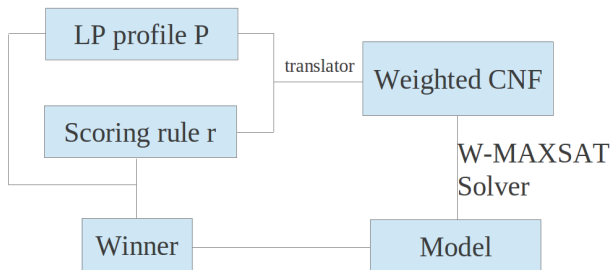


Figure : The winner problem

- Solver: *toulbar*²⁷

²⁷M Sanchez et al. "Max-CSP competition 2008: toulbar2 solver description". In: the Third International CSP Solver Competition (2008)

Random LP Profiles

- To experiment with LP profiles, we developed methods to randomly generate *encodings* of a special type of CI-CP LP-tree of size linear in the number of attributes

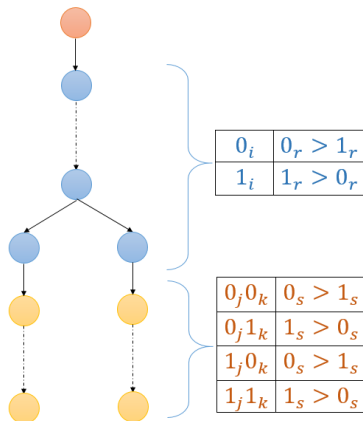
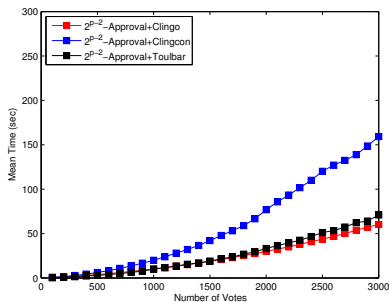
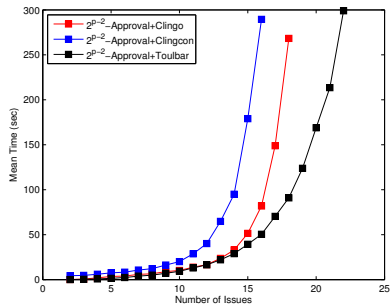


Figure : Random LP-tree

Varying p and n : 2^{p-2} -approval



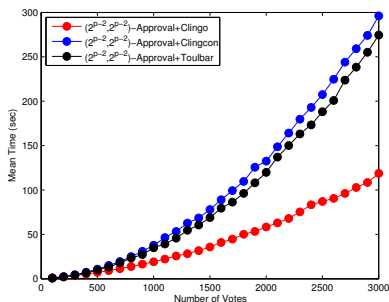
(a) Fixed #attributes (10)



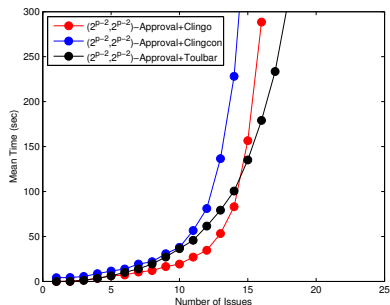
(b) Fixed #votes (1000)

Figure : Solving the winner problem

Varying p and n : $(2^{p-2}, 2^{p-2})$ -approval²⁸



(a) Fixed #attributes (10)

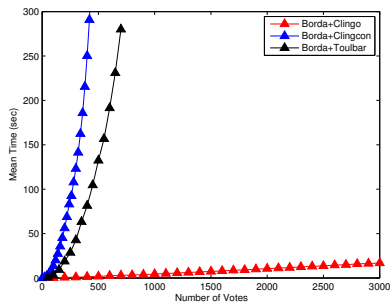


(b) Fixed #votes (1000)

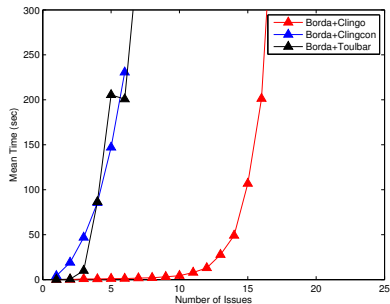
Figure : Solving the winner problem

²⁸ scoring vector: $(2, \dots, 2, 1, \dots, 1, 0, \dots, 0)$ with the numbers of 2's and 1's equal to 2^{p-2}

Varying p and n : Borda



(a) Fixed #attributes (10)



(b) Fixed #votes (1000)

Figure : Solving the winner problem

- The languages of PLP-trees and LP-trees
- Learning preference models in case of PLP-trees
- ③ Reasoning with preferences:
 - Computing winners and “strong” outcomes when votes are LP-trees
 - Application in trip planning
- Future research directions

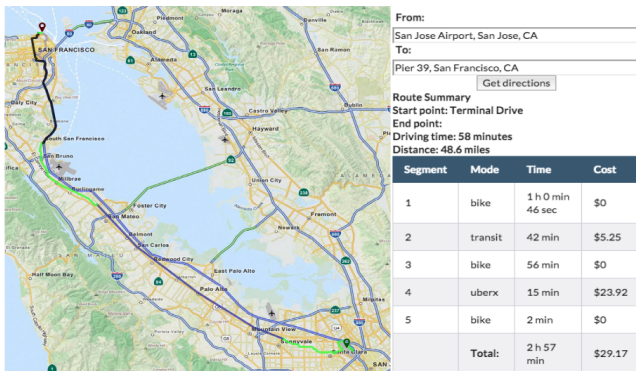
Personalization in Trip Planning

- 1 Important to incorporate user constraints and preferences into trip planning systems.
- 2 Collaboration with experts (in AI, planning, optimization, multi-agent systems) at PARC.
- 3 Developed a hipergraph-based trip planner that accommodates constraints specified as *linear temporal logic* and preferences expressed as *preferential cost function* to compute optimal routes using A*²⁹.
- 4 Available later for trip planning in the Bay Area, LA, and Denver.

²⁹Xudong Liu et al. "On Personalizability and Extensibility of Multi-Modal Trip Planning". In: PARC Symposium. 2015

Personalization in Trip Planning

- 1 From SJC, to Pier 39, Monday, 9am.
- 2 Constraints: never drive a car, and bike for 1 to 2 hours.
- 3 Preferences: bike = public (0.25) > wait(2) > walk(3), and 30\$/hr.



- The languages of PLP-trees and LP-trees
- Learning preference models in case of PLP-trees
- Reasoning with preferences:
 - Computing winners and “strong” outcomes when votes are LP-trees
 - Application in trip planning
- ④ Future research directions

Data-Driven Preference Engineering

Data-Driven Preference Learning:

- ① Recommender Systems³⁰:
 - Collaborative
 - Content-based
 - Hybrid
- ② Machine Learning (fitting function):
 - Supervised learning (e.g., decision trees, random forests)
 - Label ranking³¹
- ③ Preference Learning (learning interpretable decision models):
 - Preference Elicitation (Human-in-the-Loop)
 - Conditional Preference Networks, Preference Trees
 - Stochastic Models (e.g., Choquet integral³², TOPSIS-like models³³)

³⁰Gediminas Adomavicius and Alexander Tuzhilin. "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions". In: Knowledge and Data Engineering, IEEE Transactions on (2005)

³¹Eyke Hüllermeier et al. "Label ranking by learning pairwise preferences". In: Artificial Intelligence (2008)

³²Ali Fallah Tehrani, Weiwei Cheng, and Eyke Hüllermeier. "Choquistic Regression: Generalizing Logistic Regression using the Choquet Integral." In: EUSFLAT. 2011

³³Manish Agarwal, Ali Fallah Tehrani, and Eyke Hüllermeier. "Preference-based Learning of Ideal Solutions in TOPSIS-like Decision Models". In: Journal of Multi-Criteria Decision Analysis (2014)

Preference Reasoning and Applications:

- ① Social Choice and Welfare³⁴:
 - Voting
 - Fair division
 - Strategyproof Social Choice
- ② Automated Planning and Scheduling:
 - Travel scheduling
 - Manufacturing
 - Traffic control
- ③ Computer Vision and Image Processing:
 - Image retrieval
 - Image and video understanding

³⁴Kenneth J Arrow, Amartya Sen, and Kotaro Suzumura. Handbook of Social Choice and Welfare. Vol. 1 & 2. 2010

Computational and Data-Driven Research on Preferences

- ① Submit proposals to NSF (esp. CISE/IIS), DUE, DARPA, ARPA-E, among other possibilities.
- ② Data: binary vs. graded, absolute vs. relative, explicit vs. implicit, structured vs. unstructured, and single vs. multiple users.
 - Construct datasets using data mining and filtering techniques.
 - Tools: Bash, Python, and Matlab.
- ③ Methods/Models: recommender systems, supervised learning, conditional preference networks, and preference trees.
 - Familiarize with related literature, design and implement new preference representations.
 - Collaboration within the department and with the Department of Mathematics.
 - Tools: C/C++, logic programming solvers, and Matlab.

- ④ Behavioral research:
 - Design and implement research involving human participants through CofC IRB.
 - Collaboration with the School of Humanities and Social Sciences.
- ⑤ Reasoning tasks: dominance and optimization, and preference aggregation.
 - Implement decision support systems.
 - Tools: C/C++, logic programming solvers, SAT solvers, and Matlab.
- ⑥ Applications: fair division, strategyproof social choice, and planning and scheduling.
 - Prove computational complexities.
 - Design and implement approximation algorithms.
 - Tools: C/C++, logic programming solvers, and SAT solvers.

- ① Xudong Liu. “Modeling, Learning and Reasoning with Qualitative Preferences”. Algorithmic Decision Theory, 2015.
- ② Xudong Liu and Mirosław Truszczyński. “Reasoning with Preference Trees over Combinatorial Domains”. Algorithmic Decision Theory, 2015.
- ③ Xudong Liu and Mirosław Truszczyński. “Learning Partial Lexicographic Preference Trees over Combinatorial Domains”. AAAI Conference on Artificial Intelligence, 2015.
- ④ Xudong Liu and Mirosław Truszczyński. “Preference Trees: A Language for Representing and Reasoning about Qualitative Preferences”. Multidisciplinary Workshop on Advances in Preference Handling, 2014.

- 5 Matthew Spradling, Judy Goldsmith, Xudong Liu, Chandrima Dadi, and Zhiyu Li. “Roles and Teams Hedonic Game”. Algorithmic Decision Theory, 2013.
- 6 Xudong Liu and Mirosław Truszczynski. “Aggregating Conditionally Lexicographic Preferences Using Answer Set Programming Solvers”. Algorithmic Decision Theory, 2013.
- 7 Xudong Liu. “Aggregating Lexicographic Preference Trees Using Answer Set Programming: Extended Abstract”. International Joint Conference on Artificial Intelligence Doctoral Consortium, 2013.
- 8 Xudong Liu and Mirosław Truszczynski. “Learning Preference Trees and Forests”. (In Preparation).

- 9 Xudong Liu and Mirosław Truszczyński. “Approximating Conditional Preference Networks Using Lexicographic Preference Trees”. (In Preparation).
- 10 Xudong Liu and Mirosław Truszczyński. “Complexity of Manipulation in Elections Where Votes Are Lexicographic Preference Trees”. (In Preparation).
- 11 Xudong Liu and Mirosław Truszczyński. “Reasoning About Lexicographic Preferences Over Combinatorial Domains”. (In Preparation).
- 12 Xudong Liu and Christian Fritz. “On Personalizability and Extensibility of Multi-Modal Trip Planning”. (In Preparation).

① Quantitative:

- Utility/Cost Functions³⁵
- Possibilistic Logic³⁶
- Fuzzy Preference Relations³⁷
- Penalty Logic³⁸

② Qualitative:

- Answer-Set Optimization Theories³⁹
- Ceteris Paribus Networks (e.g., CP-nets⁴⁰, TCP-nets⁴¹, CI-nets⁴²)
- Conditional Preference Theories⁴³

³⁵Souhila Kaci. Working with Preferences: Less Is More: Less Is More. Springer Science & Business Media, 2011

³⁶Didier Dubois, Jérôme Lang, and Henri Prade. "A Brief Overview of Possibilistic Logic". In: ECSQARU. 1991

³⁷SA Orlovsky. "Decision-making with a fuzzy preference relation". In: Fuzzy sets and systems (1978)

³⁸Gadi Pinkas. Propositional non-monotonic reasoning and inconsistency in symmetric neural networks. 1991

³⁹Gerhard Brewka, Ilkka Niemelä, and Mirosław Truszczyński. "Answer Set Optimization". In: IJCAI. 2003

⁴⁰C. Boutilier et al. "CP-nets: A Tool for Representing and Reasoning with Conditional Ceteris Paribus Preference Statements". In: Journal of Artificial Intelligence Research (2004)

⁴¹Ronen I. Brafman and Carmel Domshlak. "Introducing Variable Importance Tradeoffs into CP-Nets". In: UAI. 2002

⁴²Sylvain Bouveret, Ulle Endriss, and Jérôme Lang. "Conditional importance networks: A graphical language for representing ordinal, monotonic preferences over sets of goods". In: (2009)

⁴³Nic Wilson. "Extending CP-Nets with Stronger Conditional Preference Statements". In: AAAI-04. 2004

Questions?

Thank you!