Preference Trees over Combinatorial Domains

Xudong Liu

Ph.D. Candidate
Advisor: Dr. Miroslaw Truszczynski

Department of Computer Science
College of Engineering
University of Kentucky
Lexington, KY, USA
Sunday, 4/17/2016

Preferences Are Ubiquitous

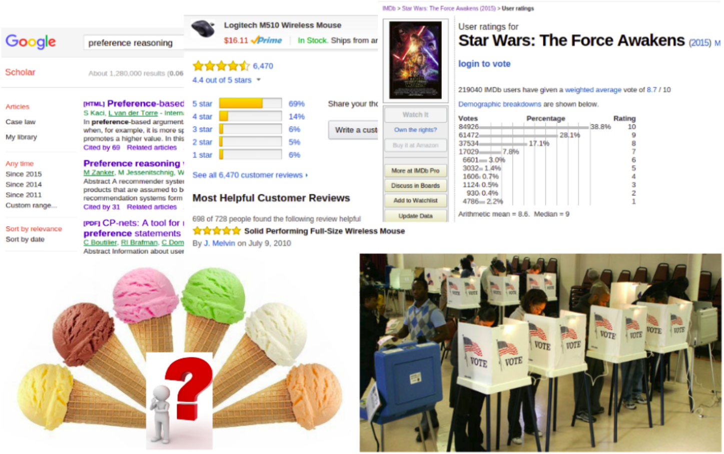


Figure : Preferences of different forms

Describing Preferences

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

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; ...

Relations and Orderings

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,

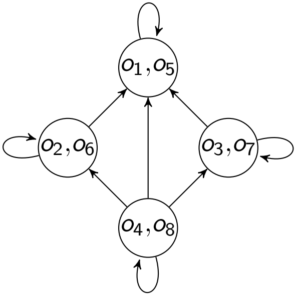
$$R \subseteq O \times O.$$

Properties of binary relations related to preferences:

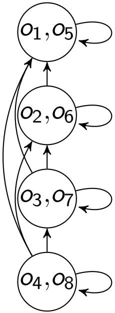
1. Reflexivity: $\forall o \in O$, $(o, o) \in R$.
2. Irreflexivity: $\forall o \in O$, $(o, o) \notin R$.
3. Totality: $\forall o_1, o_2$, $(o_1, o_2) \in R$ or $(o_2, o_1) \in R$.
4. Transitivity: $\forall o_1, o_2, o_3$, if $(o_1, o_2) \in R$ and $(o_2, o_3) \in R$, then $(o_1, o_3) \in R$.
5. Symmetry: $\forall o_1, o_2$, if $(o_1, o_2) \in R$, then $(o_2, o_1) \in R$.
6. Antisymmetry: $\forall o_1, o_2$, if $(o_1, o_2) \in R$ and $(o_2, o_1) \in R$, then $o_1 = o_2$.

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

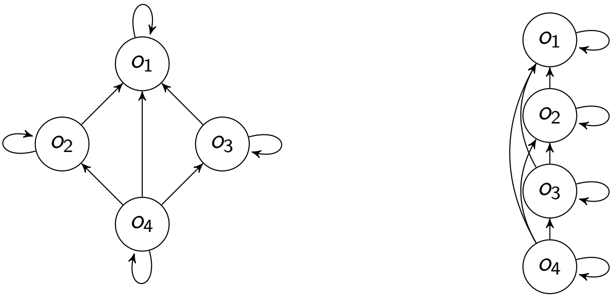


(b) total preorder

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.

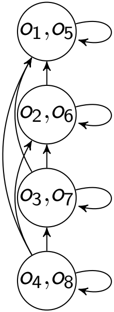


(a) partial order (b) total order

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

$$
\begin{aligned}
o_1 &\succeq o_5, \\
o_2 &\succ o_4, \\
o_4 &\approx o_8,
\end{aligned}
$$

(b) preferences

Combinatorial Domains

Combinatorial Domains

Let $\mathcal{I}$ be a finite set of attributes $\{X_1, \ldots, X_p\}$, associated with a set of finite domains $\{Dom(X_1), \ldots, Dom(X_p)\}$ for each attribute X_i. A *combinatorial domain* $CD(\mathcal{I})$ is a set of *objects* 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}.

⋮

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

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}.

Car1 Car2

<mvan, 7m, gray, big, honda, med, med> <sedan, 5, blue, med, vw, med, med>

Figure : Dominance and Optimization

Multi-Agent

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]

---

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

[2] Xudong 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

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

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

[5] Xudong 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

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 Forests[9]
3. Preference Approximation[10]

---

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

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

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

[9] Xudong Liu and Miroslaw Truszczynski. "Learning Preference Trees and Forests". In: Preparation

[10] Xudong Liu and Miroslaw Truszczynski. "Approximating Conditional Preference Networks Using Lexicographic Preference Trees". In: Preparation

Preference Reasoning

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 object wrt P?
    - Preference aggregation: which candidate wins the election?
2. Preference Misrepresentation[15,16]:
    - Manipulation

---

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

[12] Xudong 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

[13] Xudong 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

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

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

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

Preference Applications

Q: What fields can we apply preferences to?

1. Role-playing Games:
   - Hedonic games[17]
2. Automated Planning and Scheduling:
   - Trip planning[18]
3. Data-Driven Decision Making:
   - Predictive models[19]

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

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

[19] Xudong Liu and Miroslaw Truszczynski. "Learning Preference Trees and Forests". In: Preparation

Outline

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” candidates when votes are LP-trees
    - Application in trip planning
4. Future research directions

Outline

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" candidates when votes are LP-trees
    - Application in trip planning
4. Future research directions

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.

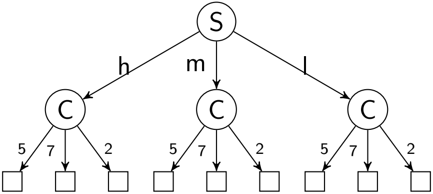


Figure : A PLP-tree over cars

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.

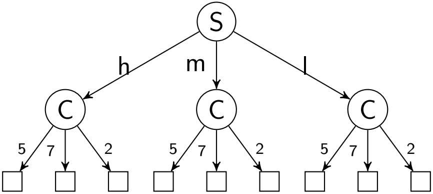


Figure : A PLP-tree over cars

Total preorder

Partial Lexicographic Preference Trees (PLP-Trees)

Car1

Car2

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

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

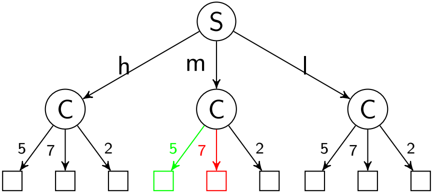
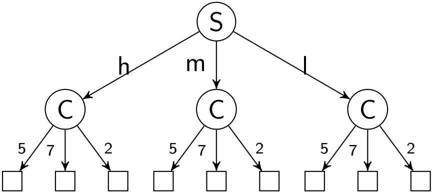


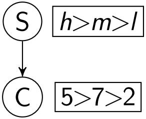
Figure : A PLP-tree over cars

$Car2 \succ Car1$

Compact Representations of PLP-Tree



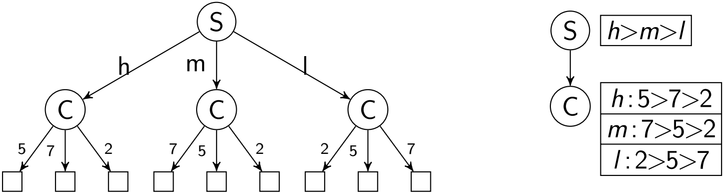
(a) Full



(b) Compact

Figure : Unconditional Importance & Unconditional Preference (UIUP)

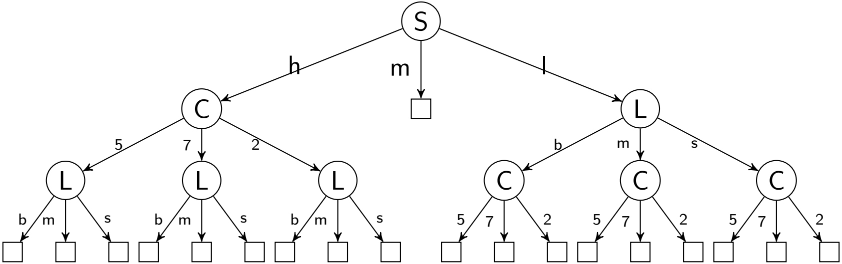
Compact Representations of PLP-Tree



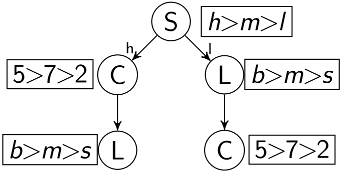
(a) Full (b) Compact

Figure : Unconditional Importance & Conditional Preference (UICP)

Compact Representations of PLP-Tree



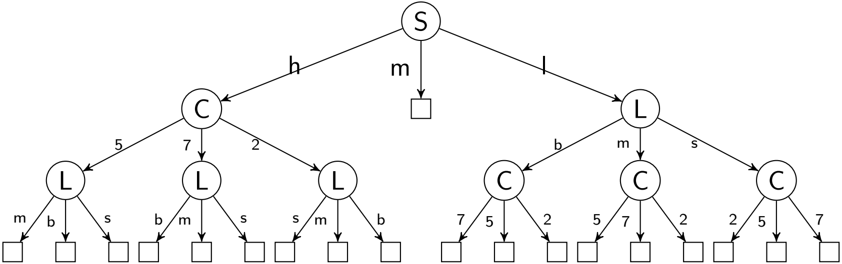
(a) Full



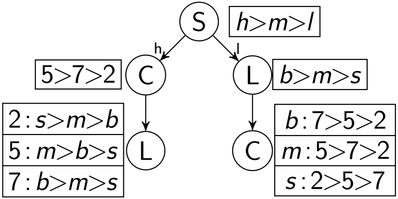
(b) Compact

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.

Outline

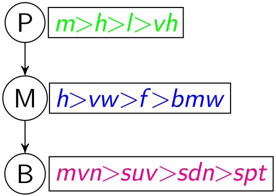
Consistent Learning (CONSLEARN)

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

(<sdn,5,blk,m,h,m,m>,<suv,7m,wht,b,f,m,m>)

(<spt,2,wht,s,bmw,h,h>,<spt,2,wht,s,bmw,vh,h>)

(<mvn,7m,gry,b,f,m,m>,<sdn,5,bl,m,f,m,m>)



UIUP tree

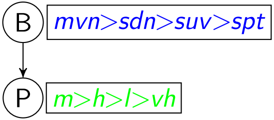
Small Learning (SmallLearn)

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 $|T| \leq l$.

(<sdn,5,blk,m,h,m,m>,<suv,7m,wht,b,f,m,m>)

(<spt,2,wht,s,bmw,h,h>,<spt,2,wht,s,bmw,vh,h>)

(<mvn,7m,gry,b,f,m,m>,<sdn,5,bl,m,f,m,m>)



UIUP tree

Maixmal Learning (MaxLearn)

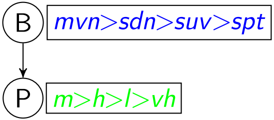
Given an example set $\mathcal{E}$ and a positive integer k ($k \leq m$), decide whether there exists a PLP-tree T (of a particular type) such that T satisfies at least k examples in $\mathcal{E}$.

(<sdn,5,blk,m,h,m,m>,<suv,7m,wht,b,f,m,m>)

(<spt,2,wht,s,bmw,h,h>,<spt,2,wht,s,bmw,vh,h>)

(<mvn,7m,gry,b,f,m,m>,<sdn,5,bl,m,f,m,m>)

(<suv,7m,gry,b,vw,vh,m>,<suv,7m,gry,b,vw,h,m>)



UIUP tree

Learning PLP-trees

Consistent Learning (ConsLearn)

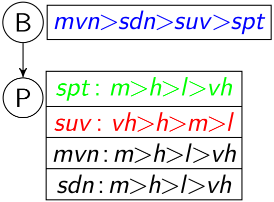
Given an example set $\mathcal{E}$, decide whether there exists a PLP-tree T (of a particular type) such that T is consistent with $\mathcal{E}$.

(<sdn,5,blk,m,h,m,m>,<suv,7m,wht,b,f,m,m>)

(<spt,2,wht,s,bmw,h,h>,<spt,2,wht,s,bmw,vh,h>)

(<mvn,7m,gry,b,f,m,m>,<sdn,5,bl,m,f,m,m>)

(<suv,7m,gry,b,vw,vh,m>,<suv,7m,gry,b,vw,h,m>)



UICP tree

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.

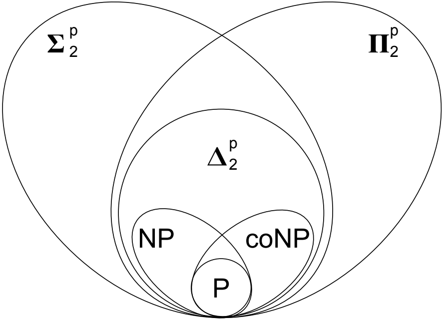


Figure : Computational complexity diagram

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

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

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

Experimentation

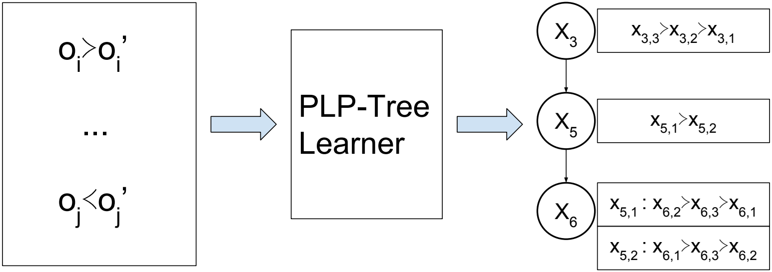


Figure : PLP-tree learning system

Datasets

Dataset	#Attributes	#Objects	#Examples
BreastCancerWisconsin	9	270	9,009
CarEvaluation	6	1,728	682,721
CreditApproval	10	520	66,079
GermanCredit	10	914	172,368
Ionosphere	10	118	3,472
MammographicMass	5	62	792
Mushroom	10	184	8,448
Nursery	8	1,266	548,064
SPECTHeart	10	115	3,196
TicTacToe	9	958	207,832
Vehicle	10	455	76,713
Wine	10	177	10,322

Figure : Preference Learning Library[22]

[22] http://www.cs.uky.edu/~liu/preflearnlib.php

PLP-Trees To Learn

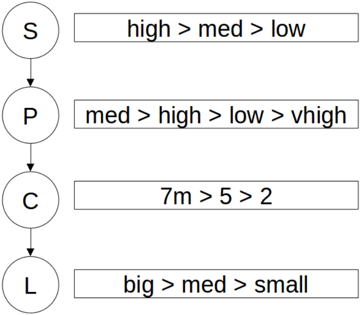


Figure : Unconditional Importance & Unconditional Preference (UIUP)

PLP-Trees To Learn

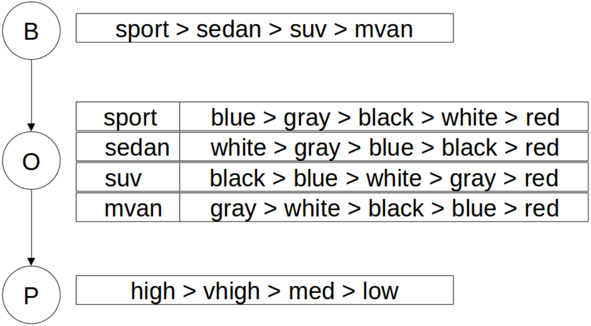


Figure : UICP with at most 1 parent (UICP-1)

PLP-Trees To Learn

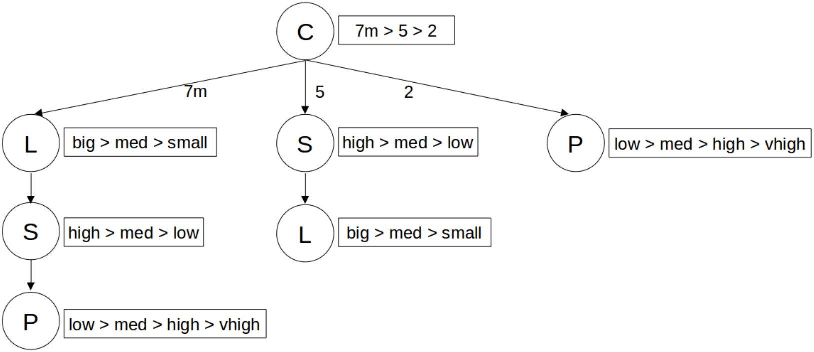


Figure : CIUP with 1 split at the root (CIUP-1)

PLP-Trees To Learn

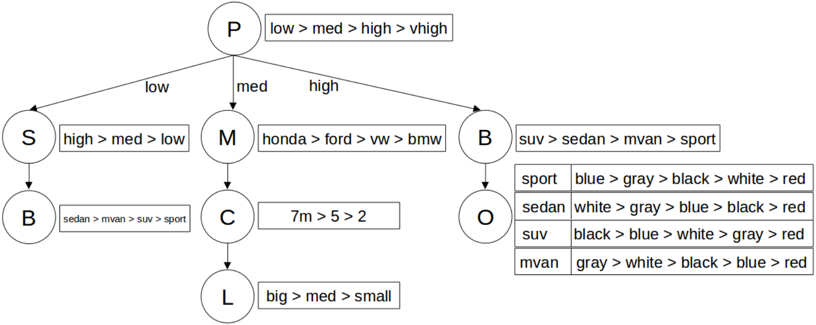
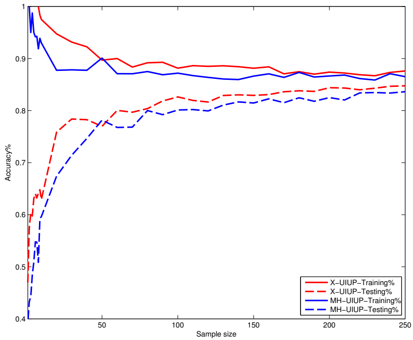
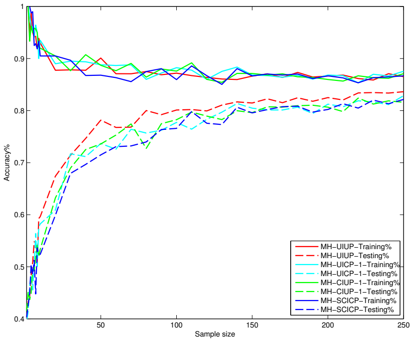


Figure : Simple CICP (SCICP)

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



(a) Compare exact & greedy heuristic



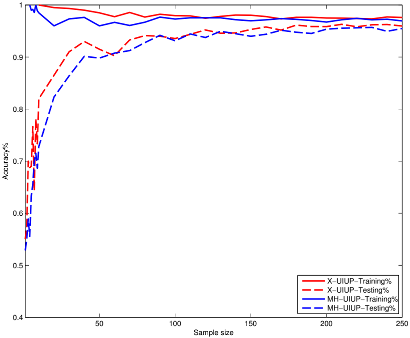
(b) Greedy heuristic

Figure : Learning curves solving MaxLearn

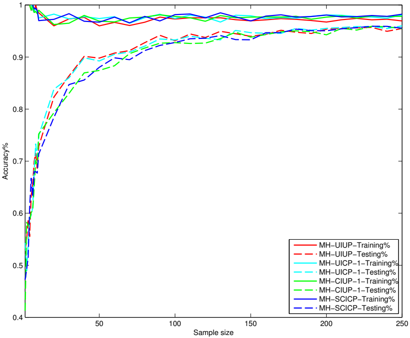
[23] http://www.cs.uky.edu/~liu/preflearnlib.php

Experimental Results: Wine[24]

#attributes:10, #objects: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

Outline

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” candidates when votes are LP-trees
   - Application in trip planning
4. Future research directions

Positional Scoring Rules

- k-approval: $(1, \dots, 1, 0, \dots, 0)$ with k being the number of 1's.
- (k, l)-approval: $(c, \dots, c, d, \dots, d, 0 \dots, 0)$, where c and d are constants ($c > d$), and the numbers of c's and d's equal to k and l.
- b-Borda: $(b, b-1, \dots, b-m+1)$, where b is a constant and m is the number of candidates.

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

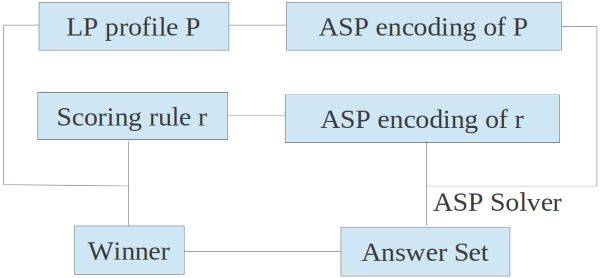


Figure : The winner problem

- Solvers: *clingo*[25], *clingcon*[26]

[25] M. Gebser et al. "Potassco: The Potsdam Answer Set Solving Collection". In: AI Communications (2011)
[26] Max Ostrowski and Torsten Schaub. "ASP modulo CSP: The clingcon system". In: TPLP (2012)

Modeling the Problems in W-MAXSAT

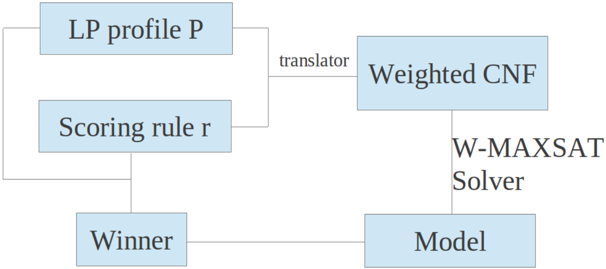
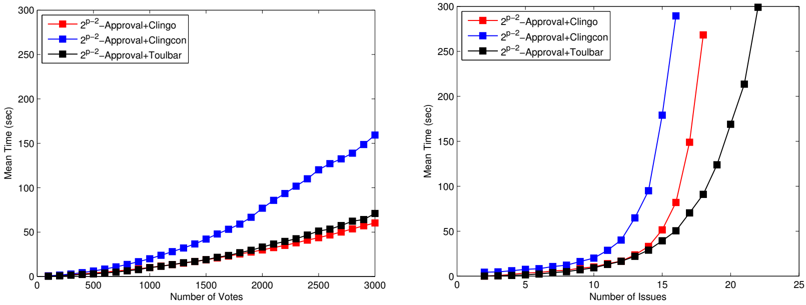


Figure : The winner problem

- Solver: *toulbar*[27]

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

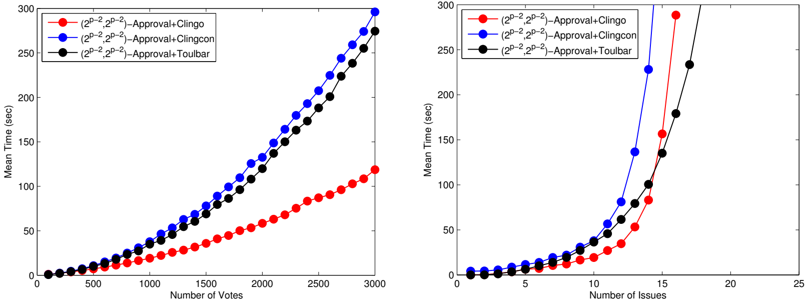
Varying p and n: 2^{p-2}-approval



(a) Fixed #attributes (10) (b) Fixed #votes (1000)

Figure : Solving the winner problem

Varying p and n: $(2^{p-2}, 2^{p-2})$-approval [28]

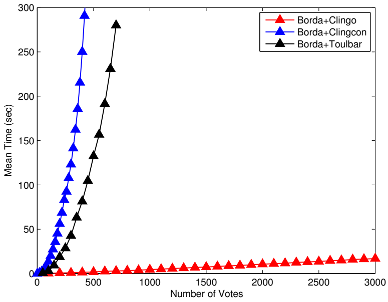


(a) Fixed #attributes (10)   (b) Fixed #votes (1000)

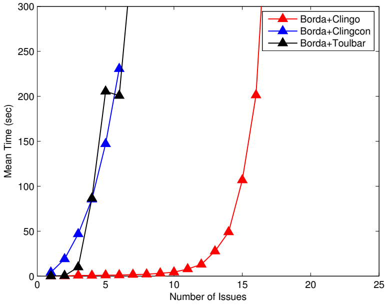
Figure : Solving the winner problem

[28] scoring vector: $(2, \ldots, 2, 1, \ldots, 1, 0, \ldots, 0)$ 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

Outline

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" candidates when votes are LP-trees
   - Application in trip planning
4. Future research directions

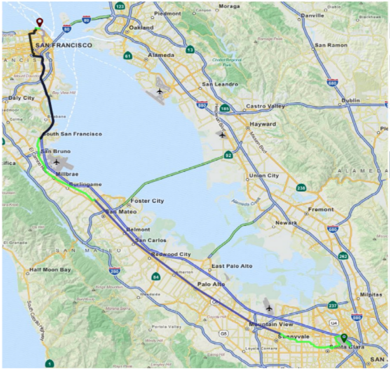
Personalization in Trip Planning

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.

[29] Xudong 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.



From:

San Jose Airport, San Jose, CA

To:

Pier 39, San Francisco, CA

Get directions

Route Summary
Start point: Terminal Drive
End point:
Driving time: 58 minutes
Distance: 48.6 miles

Segment	Mode	Time	Cost
1	bike	1 h 0 min 46 sec	$0
2	transit	42 min	$5.25
3	bike	56 min	$0
4	uberx	15 min	$23.92
5	bike	2 min	$0
	Total:	2 h 57 min	$29.17

Outline

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” candidates when votes are LP-trees
   - Application in trip planning
4. Future research directions

Data-Driven Preference Engineering

Data-Driven Preference Learning:

1. Recommender Systems[30]:
   - Collaborative
   - Content-based
   - Hybrid
2. Machine Learning (fitting function):
   - Supervised learning (e.g., decision trees, random forests)
   - Label ranking[31]
3. Model-based Learning (learning interpretable decision models):
   - Preference Elicitation (Human-in-the-Loop)
   - Conditional Preference Networks, Preference Trees
   - Stochastic Models (e.g., Choquet integral[32], TOPSIS-like models[33])

[30] Gediminas 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)

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

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

[33] Manish 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 Welfare[34,35]:
   - Voting
   - Fair division
   - Strategyproof Social Choice
2. Automated Planning and Scheduling[36,37,38]:
   - Travel scheduling
   - Manufacturing
   - Traffic control

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

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

[36] Tran Cao Son and Enrico Pontelli. "Planning with preferences using logic programming". In: Theory and Practice of Logic Programming (2006)

[37] Meghyn Bienvenu, Christian Fritz, and Sheila A McIlraith. "Specifying and computing preferred plans". In: Artificial Intelligence (2011)

[38] Hannah Bast et al. "Route planning in transportation networks". In: arXiv preprint (2015)

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, and single vs. multiple users.
    - Construct datasets using data mining and filtering techiques.
3. Methods/Models: recommender systems, supervised learning, conditional preference networks, and preference trees.
    - Familiarize with related literature, and design new preference representations.
    - Collaboration within CS Department and with the Department of Mathematics & Statistics.

Computational and Data-Driven Research on Preferences

4. Behavioral Research:
   - Design and implement experiments involving human participants through TCNJ IRB.
   - Collaboration with the School of Humanities and Social Sciences.
5. Preference Reasoning and Applications:
   - Prove computational complexities.
   - Design and implement decision support systems.
   - Design and implement approximation algorithms.

Publications

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

Publications

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 Miroslaw 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 Miroslaw Truszczynski. "Learning Preference Trees and Forests". (In Preparation).

Publications

9. Xudong Liu and Miroslaw Truszczynski. “Approximating Conditional Preference Networks Using Lexicographic Preference Trees”. (In Preparation).
10. Xudong Liu and Miroslaw Truszczynski. “Complexity of Manipulation in Elections Where Votes Are Lexicographic Preference Trees”. (In Preparation).
11. Xudong Liu and Miroslaw Truszczynski. “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).

Related Work

1. Quantitative:
   - Utility/Cost Functions[39]
   - Possibilistic Logic[40]
   - Fuzzy Preference Relations[41]
   - Penalty Logic[42]
2. Qualitative:
   - Answer-Set Optimization Theories[43]
   - Ceteris Paribus Networks (e.g., CP-nets[44], TCP-nets[45], CI-nets[46])
   - Conditional Preference Theories[47]

---

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

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

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

[42] Gadi Pinkas. Propositional non-monotonic reasoning and inconsistency in symmetric neural networks. 1991

[43] Gerhard Brewka, Ilkka Niemelä, and Miroslaw Truszczynski. "Answer Set Optimization". In: IJCAI. 2003

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

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

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

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

Questions?

Thank you!