Human-in-the-Loop Learning of Qualitative Preference Models

Joseph Allen
School of Computing
University of North Florida
Jacksonville, FL
n01045721@unf.edu

Ahmed Moussa
School of Computing
University of North Florida
Jacksonville, FL
nagar@aucegypt.edu

Xudong Liu
School of Computing
University of North Florida
Jacksonville, FL
xudong.liu@unf.edu

Abstract

In this work, we present a novel human-in-the-loop framework to help the human user understand the decision making process that involves choosing preferred options. We focus on qualitative preference models over alternatives from combinatorial domains. This framework is *interactive*: the user provides her behavioral data to the framework, and the framework explains the learned model to the user. It is *iterative*: the framework collects feedback on the learned model from the user and tries to improve it accordingly till the user terminates the iteration. In order to communicate the learned preference model to the user, we develop visualization of intuitive and *explainable* graphic models, such as lexicographic preference trees and forests, and conditional preference networks. To this end, we discuss key aspects of our framework for lexicographic preference models.

Introduction

Preferences are an essential component to decision making and have been extensively studied in research communities such as decision theory, computational social choice, recommender systems, and knowledge representation. Various preference models have been proposed in the literature to represent preferences of different types including two major ones: quantitative models and qualitative models. Quantitative preference models integrate into the models numeric values used to define the preference relation of objects. These models include fuzzy constraint satisfaction models (Schiex 1992), penalty logic (De Saint-Cyr, Lang, and Schiex 1994), and possibilistic logic (Dubois, Lang, and Prade 1994). On the other hand, qualitative preference models describe, either directly or indirectly, the relative ordinal relation of objects. Such models include lexicographic preference trees (LP-trees) (Booth et al. 2010; Liu and Truszczynski 2013; 2015b), lexicographic preference forests (LP-forests) (Liu and Truszczynski 2016; 2018), conditional preference network (CP-nets) (Boutilier et al. 2004), and answer set optimization (Brewka, Niemelä, and Truszczynski 2003). These models draw our focus because they are proven to be intuitive, cognitively plausible, and predictive with high accuracy (Allen et al. 2015; Liu and Truszczynski 2018).

In this paper, we focus on the learning problem of qualitative preference models, in particular, graphical models that are intuitive and often compact in size, such as LP-trees, LP-forests and CP-nets. Recently, active and passive learning of these graphical models have been studied, both theoretically and empirically, in the community (Liu and Truszczynski 2015a; 2018; Koriche and Zanuttini 2010; Alanazi, Mouhoub, and Zilles 2016; Allen, Siler, and Goldsmith 2017). However, these traditional preference learning works do not leverage the intuitivity and explainability of the models to *interact* with the decision maker in the learning process. Models explainable to human users are desirable when decision makers in various applications are to understand or even trust the resulting models formulated by intelligent machine partners (Gunning 2017).

To this end, we propose a novel framework that learns qualitative preference models. This framework is interactive and iterative: from the decision making user it obtains behavioral data, which is then preprocessed before sent to a preference learner that computes models (i.e., an LP-tree, LP-forest or CP-net) to be visualized and presented back to the user for feedback in order to improve the models in the following iterations. This learning process terminates when the user is satisfied with the learned models. In this report, we show our design and implementation of the framework, so far for learning LP-trees and LP-forests, that is a web application using Django with Python and C++ as the programming languages on the server.

In the next section, we define and exemplify the two models: LP-trees and LP-forests. Then, we present our human-in-the-loop preference learning framework, and demonstrate it by showing our prototype that learns LP models. Finally, we conclude, pointing to possible future research directions.

Lexcigraphic Preference Trees and Forests

The preference models we consider in this work are over alternatives from combinatorial domains of multi-valued attributes. Now we define combinatorial domains and preference models we focus in our paper including lexicographic preference trees and forests. Let $\mathcal{A} = \{X_1, \ldots, X_p\}$ be a set of categorical attributes, each X_i with a finite domain D_i, where $|D_i|$ is bounded by a constant. The *combinatorial domain $CD(\mathcal{A})$* over $\mathcal{A}$ is the Cartesian product $D_1 \times \ldots \times D_p$. Elements of combinatorial domains are called *alternatives*.
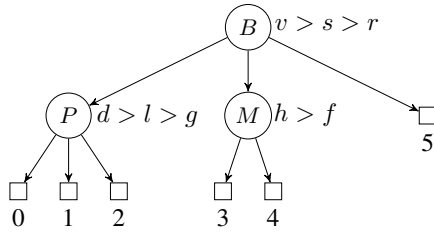
Figure 1: A preference tree T over the car domain

Figure 2: UIUP and UICP LP-trees

A lexicographic preference tree over $CD(\mathcal{A})$ is an ordered labeled tree, where (1) every non-leaf node is labeled by an attribute $X_i \in \mathcal{A}$, and by a *local preference* $>_i$, which is a total order over D_i; (2) every non-leaf node labeled by an attribute X_i has $|D_i|$ outgoing edges, ordered from left to right according to $>_i$; (3) every leaf node is denoted by $\square$; and (4) on every path from the root to a leaf each attribute appears *at most once* as a label. Each tree induces a *total preorder* that precisely is defined by the order of the leafs.

To illustrate, let us consider the domain of cars described by four attributes: *BodyType* (B) with values: *minivan* (v), *sedan* (s), and *sport* (r); *Make* (M) with values *Honda* (h) and *Ford* (f); *Price* (P) with values *low* (l), *medium* (d), and *high* (g); and *Transmission* (T) with *automatic* (a) and *manual* (m). An user's preference order on cars from this space could be expressed by a tree T in Figure 1.

Tree T informs us that the most important attribute is *BodyType* with the user preferring minivans the most, then sedans and sports the least. Among minivans, the most important attribute is $Price$ with medium preferred to low to high. Other non-leaf nodes in the tree are interpreted similarly. Leaf nodes, however, represent sets of cars with the instantiations of the attributes along their paths.

Given an alternative o, we can traverse the tree and find its leaf. To compare two alternatives, we say that they are *equivalent* if they have same leaf. If they have different leaf nodes, the alternative in the preceding leaf is the preferred one. For instance, a Honda sedan is better than a Ford sedan, because the former ends up in leaf 3, preceding leaf 4, the leaf the latter car has.

Types of LP-Trees

LP-trees, in general, can be of size exponential in the size of the combinatorial domain. However, trees with special structures can be collapsed to achieve compact representation. When the labeling attributes on all paths of the tree are exactly the same and the local preference orderings are the same on same attributes, this tree can be collapsed to a list of nodes labeled by attributes and unconditional preference orders. We call this type of LP-trees *unconditional importance and unconditional preference* LP-trees (*UIUP LP-trees*). Keeping this tree structure, if the local preferences are different on same attributes, these trees can also be collapsed, but to a list of nodes labeled by attributes and tables of conditional preference orders. We call this type of LP-trees *unconditional importance and conditional preference* LP-trees (*UICP LP-trees*). All the other LP-trees that are
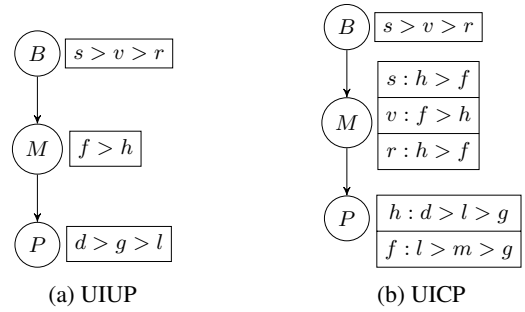
uncollapsible are called *conditional importance and conditional preference* LP-trees (*CICP LP-trees*), for the importance order of nodes depends on how their ancestors are instantiated in the tree. We show examples of these compact representations in Figure 2a and Figure 2b. One example of a CICP tree is the one in Figure 1.

LP-Forests

An LP-forest is a finite ensemble of LP-trees over combinatorial domains. To compare alternatives using an LP-forest, researchers have proposed to apply a plethora of voting rules (such as Borda's and Copeland's rules) to aggregate the decisions of member trees (Lang, Mengin, and Xia 2018; Liu and Truszczynski 2018). We propose to study the visualization problem of big forests of trees and how to effectively present the forest to the user. Clearly, visualizing and presenting the whole forest is infeasible. In this paper, our approach is to only present representative trees in the forest according to some distance measure, for which we consider Kendall's τ distance. This measurement calculates the number of pairwise disagreements between two orderings. Clearly, it directly applies to computing distances between LP-trees, for LP-trees represent total orders, after possible equivalent alternatives are broken alphabetically. However, if we compute $\tau(T_1, T_2)$ via computing their total orders first, the process may take time exponential in the size of the two trees if they are compactly represented of type UIUP or UICP. To alleviate this, we resort to polynomial algorithms proposed by Li and Kazimipour (Li and Kazimipour 2018). We implemented these algorithms to compute the τ distances for all pairs of trees in the forest. Then, the trees are clustered based on these distance values.

Framework

We now introduce our framework, shown in Figure 3, for interactive learning of qualitative preference models in the following. We call our framework ILPref for short. The goal of ILPref is to learn, and help the user to understand, her preferential decision making process over complex domains of options.

User

The user is the central decision maker, whom the framework tries to help understand her decision making process. The
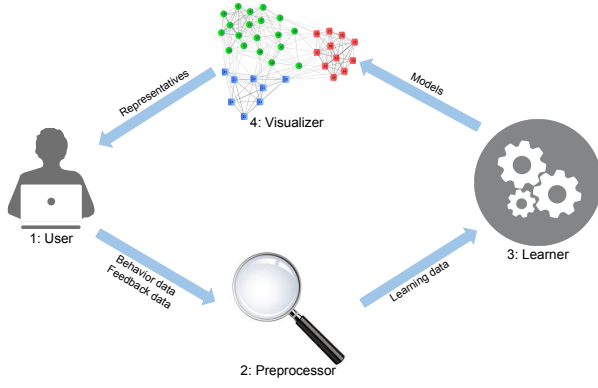
Figure 3: ILPref

Figure 4: Preferential data collection

Figure 5: Visualization and feedback for UIUP trees

user provides *behavioral data* that can be either explicit or implicit. Explicit data are such as query answers and scaled ratings, whereas implicit data can be time or clicking distribution over a set of options reviewed by her. These are the source data our framework is learning the decision model from. Our implementation as is, shown in Figure 4, elicits behavioral data via binary queries asking the user to select the optional car the she likes more than the other.

Another type of data provided by the user is the *feedback data*, which are critiques based on the user's input – visually explained model. Clearly, feedback data are not provided by the decision maker in the initial iteration, for no model is learned yet. In general, feedback data are for the learning algorithm to adjust the learned model accordingly. In our current implementation for UIUP LP-tree models, the user can describe her feedback on the order of some attributes and the order of some attributes' values. For instance, as shown in Figure 5, we see the learned UIUP tree presented. Based on it, the user provides the feedback that BuyingPrice should be more important than Persons. Also, she actually prefers medium to low on BuyingPrice, and big to medium on Luggage.

Preprocessor and Learner

The preprocessor takes the behavioral and feedback data and applies text mining techniques to formalize the data to be ready for the learning module. The learner then takes the domain description and examples and learns a model. Currently, we implemented the greedy heuristic for learning
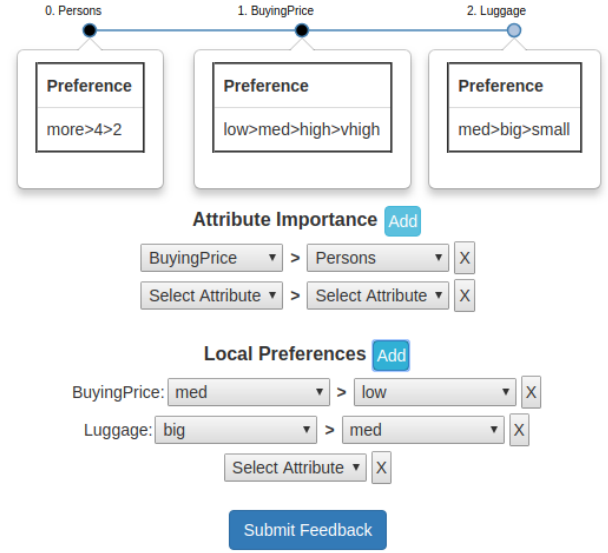
UIUP, UICP and CICP trees and forests of these trees (Liu and Truszczynski 2016). This implementation is augmented to handle feedback data from the user, in such a way that these data are treated as hard constraints.

Visualizer

To present the learned model to the user, the visualizer draws the model and provides annotated description of it. Our prototype implements this module for visualizing LP-trees and LP-forests (cf. Figure 5 for a UIUP tree model). An LP-tree is drawn with expandable nodes to show or hide subtrees. It is up to only a few levels of the tree, as deeper attributes are less important in the model. When a forest of trees are learned, the framework only visualizes a very small number of representatives, selected by some clustering algorithm. Our prototype applies a single-link clustering algorithm SLINK, by Sibson (Sibson 1973), based on pairwise Kendall's τ distances between individual trees. Taking as input the $\tau(T, T')$ distances between all pairs of trees, SLINK starts with clusters of single trees. It merges two clusters with the minimum distance between them, where the distance between clusters are defined as the average of distances between all pairs of trees in them. For example, we see the dendrogram of 13 UIUP trees in Figure 6, where the y-axis are the threshold values that are the numbers of disagreed examples between clusters of trees. To select the representatives using this dendrogram, we simply use a cut-off threshold value to partition the trees into buckets and select one model from each bucket.

Conclusion and Future Work

In this paper, we presented a novel human-in-the-loop framework to help the human user understand the decision making process of choosing from alternatives. We focused on alternatives from combinatorial domains and qualitative
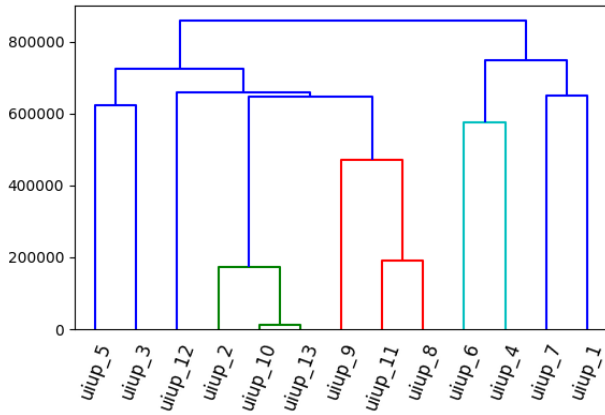
Figure 6: Clusters of UIUP trees

preference models that are intuitive and explainable, such as LP-trees, LP-forests, and CP-nets. This framework, which we call ILPref, is an interactive and iterative system. It visualizes the learned model to the user for feedback, which is taken to improve the model in the following iterations. To this end, we discussed the key aspects of our prototype system for learning the LP models.

For future work, we are interested in extending our prototype to enclose more preference models, e.g., CP-nets. We also plan to perform a thorough user case study with human subjects to evaluate our system, and our selected decision models.

References

Alanazi, E.; Mouhoub, M.; and Zilles, S. 2016. The complexity of learning acyclic cp-nets. In *IJCAI*, 1361–1367.

Allen, T. E.; Chen, M.; Goldsmith, J.; Mattei, N.; Popova, A.; Regenwetter, M.; Rossi, F.; and Zwilling, C. 2015. Beyond theory and data in preference modeling: Bringing humans into the loop. In *International Conference on Algorithmic DecisionTheory*, 3–18. Springer.

Allen, T. E.; Siler, C.; and Goldsmith, J. 2017. Learning tree-structured cp-nets with local search. In *International Florida Artificial Intelligence Research Society Conference*.

Booth, R.; Chevaleyre, Y.; Lang, J.; Mengin, J.; and Sombattheera, C. 2010. Learning conditionally lexicographic preference relations. In *ECAI*, 269–274.

Boutilier, C.; Brafman, R.; Domshlak, C.; Hoos, H.; and Poole, D. 2004. CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *Journal of Artificial Intelligence Research* 21:135–191.

Brewka, G.; Niemelä, I.; and Truszczynski, M. 2003. Answer set optimization. In *IJCAI*, volume 3, 867–872.

De Saint-Cyr, F. D.; Lang, J.; and Schiex, T. 1994. Penalty logic and its link with dempster-shafer theory. In *Uncertainty Proceedings*. Elsevier. 204–211.

Dubois, D.; Lang, J.; and Prade, H. 1994. Possibilistic logic 1.

Gunning, D. 2017. Explainable artificial intelligence (xai). *Defense Advanced Research Projects Agency (DARPA), nd Web*.

Koriche, F., and Zanuttini, B. 2010. Learning conditional preference networks. *Artificial Intelligence* 685–703.

Lang, J.; Mengin, J.; and Xia, L. 2018. Voting on multi-issue domains with conditionally lexicographic preferences. *Artificial Intelligence*.

Li, M., and Kazimipour, B. 2018. An efficient algorithm to compute distance between lexicographic preference trees. In *IJCAI*, 1898–1904.

Liu, X., and Truszczynski, M. 2013. Aggregating conditionally lexicographic preferences using answer set programming solvers. In *International Conference on Algorithmic Decision Theory*, 244–258. Springer.

Liu, X., and Truszczynski, M. 2015a. Learning partial lexicographic preference trees over combinatorial domains. In *AAAI Conference on Artificial Intelligence (AAAI)*, 1539–1545. AAAI Press.

Liu, X., and Truszczynski, M. 2015b. Reasoning with preference trees over combinatorial domains. In *International Conference on Algorithmic Decision Theory (ADT)*, 19–34. Springer.

Liu, X., and Truszczynski, M. 2016. Learning partial lexicographic preference trees and forests over multi-valued attributes. In *Global Conference on Artificial Intelligence (GCAI)*, volume 41 of *EPiC Series in Computing*, 314–328. EasyChair.

Liu, X., and Truszczynski, M. 2018. Preference learning and optimization for partial lexicographic preference forests over combinatorial domains. In *International Symposium on Foundations of Information and Knowledge Systems (FoIKS)*. Springer.

Schiex, T. 1992. Possibilistic constraint satisfaction problems or "how to handle soft constraints?". In *Uncertainty in Artificial Intelligence, 1992*. Elsevier. 268–275.

Sibson, R. 1973. Slink: an optimally efficient algorithm for the single-link cluster method. *The computer journal* 16(1):30–34.