Modeling User Preferences via Theory Refinement

Ben Geisler[*], Vu Ha[*]
[*]Decision Systems and Artificial Intelligence Lab
Dept. of EE&CS
University of Wisconsin-Milwaukee
{bgeisler, vu, haddawy}@cs.uwm.edu

Peter Haddawy[†*]
[†]CSIM Program ,School of Advanced Technologies
Asian Institute of Technology
Bangkok, Thailand
haddawy@ait.ac.th

ABSTRACT
We present an approach to elicitation of user preference models in which assumptions can be used to guide but not constrain the elicitation process. We show how to encode assumptions concerning preferential independence and monotonicity in a Knowledge-Based Artificial Neural Network. We quantify the degree to which user preferences violate a set of assumptions. We empirically compare the KBANN network with an unbiased ANN in terms of learning rate and accuracy for preferences consistent and inconsistent with the assumptions. We go on to demonstrate how the technique can be used to learn a fine-grained preference structure from simple binary classification data.

Keywords: User modeling, Personalization, Neural networks, Decision Theory

1. INTRODUCTION
There is increasing interest in the use of user preference information to personalize user interfaces. But accurately eliciting a preference model can be highly time consuming. Thus we are faced with the conflicting goals of eliciting a preference model as accurately as possible and asking as few questions of the user as possible.

Multi-Attribute Utility Theory [3] provides a rich framework for representation and elicitation of preferences. According to MAUT, a set of items is described in terms of a set of attributes with an item being a complete assignment of values to all the attributes. In domains containing no uncertainty, user preferences over a set of items can be represented by a value function that assigns a real number to each item, resulting in a total ordering over the items. But when the number of items is large, eliciting a high-dimensional value function is impractical, so practitioners typically assume attributes to be mutually preferentially independent. This permits a value function to be represented as a linear combination of sub-value functions. As a running example, we will use the domain of airline flight selection, but we emphasize that the presented techniques are domain independent. Suppose we have a user who wishes to find a flight from Milwaukee to the Santa Fe. We might describe flights by three attributes: time (T), cost (C), and whether or not there is a layover involved (L). Time and cost are continuous, while layover is binary. If we assume that the user prefers shorter flights, cheaper flights, and flights without layovers, we need only assess the tradeoff coefficients that indicate how much the decision maker is willing to trade off the level of one attribute for a more desirable level of another. A resulting value function might be $v(t,c,l) = -100t - c - 50l$.

A major drawback of this traditional elicitation approach is that the elicitation process is *constrained* by the independence assumptions. If the assumptions are incorrect, the elicited model will be inaccurate. It would be useful to be able to work more flexibly with simplifying assumptions. In particular, we may wish to make assumptions that at least approximately apply to a large segment of the user population and then correct for inaccuracies in those assumptions when we encounter an individual to whom they do not apply. The problem of starting with an approximately correct domain theory and then correcting and refining it in the light of data has been studied in the area of theory refinement. In this paper we explore how to use Knowledge-Based Artificial Neural Networks (KBANN) [5] for preference elicitation.

The KBANN technique permits a domain theory expressed as an acyclic set of propositional Horn-clauses to be encoded in a back-propagation neural network. The propositions become nodes and the logical relations between them become links. By training the network on data an incomplete domain theory can be supplemented and an inaccurate domain theory can be corrected. Given an approximately correct domain theory, KBANN has been shown to require fewer examples to correctly classify a test set than an unbiased network [5]. When using an unbiased network, the weights are initialized randomly, so that the network starts out in a randomly chosen point in function space. In contrast, initializing the network with an approximately correct domain theory starts it at a point that is close to the target function.

2. REPRESENTING PREFERENCES
In eliciting preferences, we must choose a representation through which the user can communicate preference information. Furthermore, if we wish to use theory refinement, we must also have a representation in which to

express the domain theory. In the literature on preference elicitation, two representations are predominant: numeric ratings and pairwise preferences. Numeric ratings are the most commonly used representation used in work on collaborative filtering. A degenerate case of numeric ratings is the *binary classification*, where items are rated as good or bad, interesting or uninteresting, etc. Numeric ratings have the disadvantage of fixed granularity. Either we choose a coarse granularity and cannot make fine distinctions in preference or we choose a fine granularity and are faced with deciding exactly what rating to assign to an item. Furthermore, it is difficult to imagine how to encode a domain theory in a principled way if we must express it in terms of a rating assigned to an individual item. In contrast, use of pairwise preferences provides us the luxury of variable granularity. Notice that any set of numeric ratings assigned to a set of items can be represented in terms of a set of pairwise comparisons among the items. The domain theory can be written in a natural way, stating under what conditions the user prefers one item to another.

Using pairwise preferences requires the network to have input nodes for the attributes of two items and a binary output node that indicates preference or no preference. If two items are input to the network in one order and then the other and the output indicates no preference for both cases, then the user is considered to be indifferent between the two items.

3. ENCODING THE ASSUMPTIONS AND NETWORK ARCHITECTURE

We can conveniently encode assumptions concerning preferential independence and monotonicity of preference through statements of dominance. For example, we can encode the assumptions that the user prefers cheaper flights to more expensive flights, shorter flights to longer flights, and flights without layover to those with using the following set of Horn-clauses:

$$C_1 < C_2 \wedge T_1 \leq T_2 \wedge L_1 \leq L_2 \rightarrow F_1 \succ F_2$$

$$C_1 \leq C_2 \wedge T_1 < T_2 \wedge L_1 \leq L_2 \rightarrow F_1 \succ F_2$$

$$C_1 \leq C_2 \wedge T_1 \leq T_2 \wedge L_1 < L_2 \rightarrow F_1 \succ F_2,$$

where C_i and T_i are the cost and time of flight i, respectively. L_i is 0 if no layover and 1 otherwise. $F_1 \succ F_2$ indicates that flight 1 is preferred to flight 2.

Figure 1 shows the general network architecture. To accommodate this domain theory we include an extra hidden layer that contains the logic for our comparison operators. The layer is located between the input layer and the initial hidden layer. While every node at each layer is connected to every node at the next with low randomly chosen weights, the Horn-clauses are encoded by setting some weights to high values. The figure shows the links that encode the first Horn-clause in the domain theory above. While it would be possible to compute the comparisons outside the network and

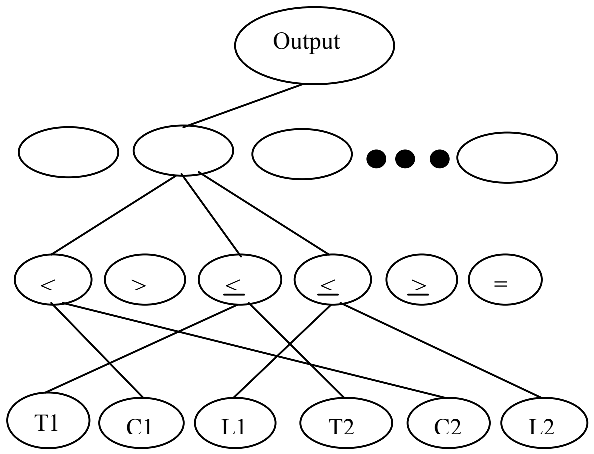

Figure 1: Structure of the elicitation network.

include them as separate inputs, representing them as nodes in the network permits them to be modified during training.

4. EMPIRICAL EVALUATION

We created two networks: one with no domain theory (ANN), and one with the domain theory above (dominance). We tested the three networks by randomly generating examples of pairwise preferences from three different value functions: $v(t,c,l) = -100t - 1c - 50l$, $v(t,c,l) = -50t - 1c - 100l$, and $v(t,c,l) = -1t - 100c - 50l$. Each value function experiment was run 10 times, with early stopping used on each run. The test size was kept constant at 50 items. After each of the 10 runs for each of the TrainSize possibilities, we computed a mean for each value function. Similar results were observed with all three value functions. Figure 2 shows the mean of the experiments

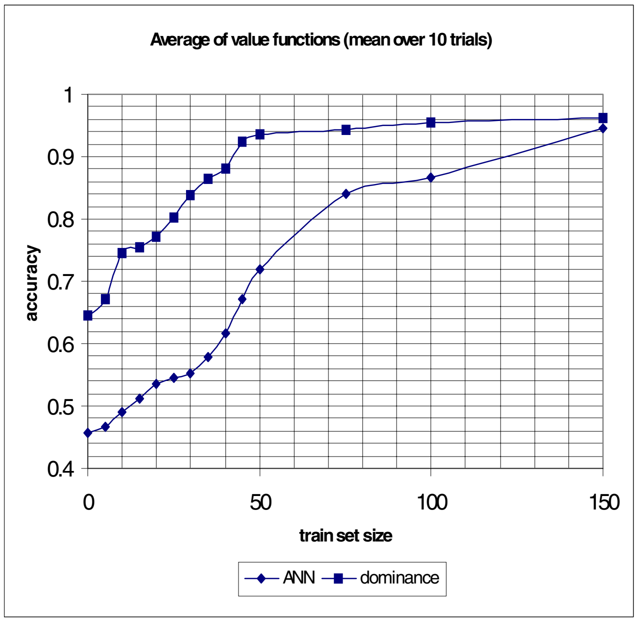

Figure 2: Mean of learning curves for three value functions.

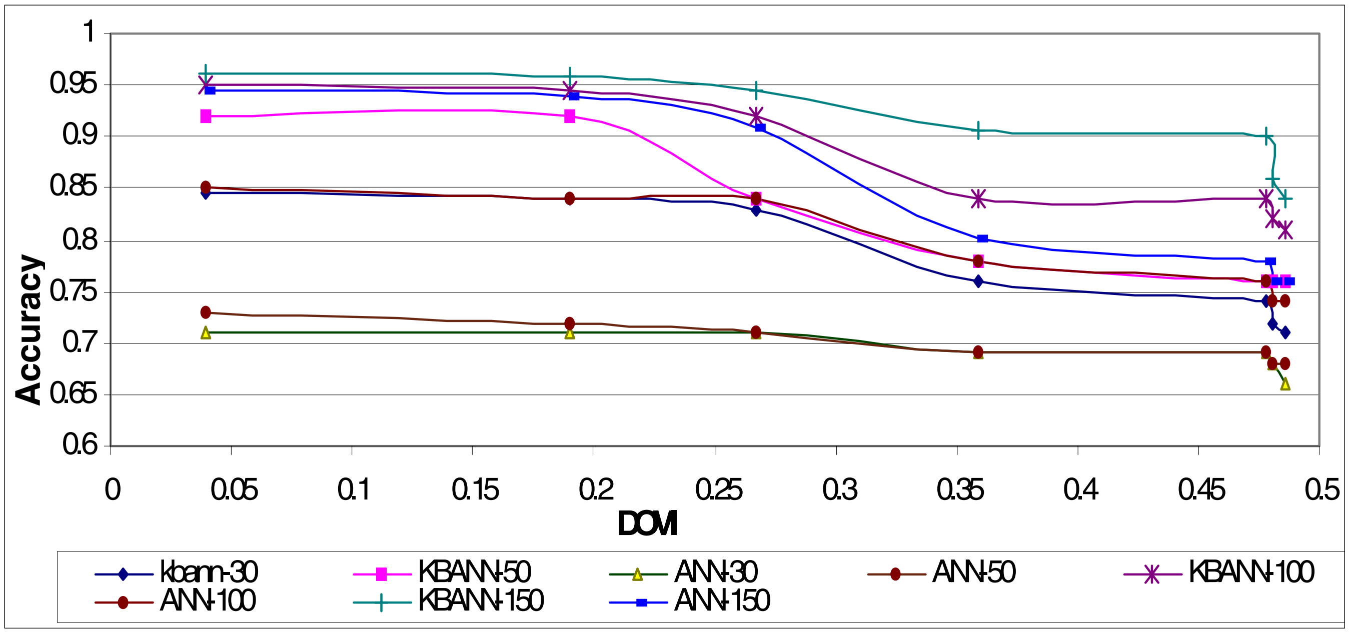

Figure 3: Accuracy of training on examples generated from value functions of varying DOVI.

with the three different value functions. The network trained with the domain theory consistently outperforms the unbiased network.

4.1 Robustness Analysis

We examined the performance of the networks for a number of value functions that violate the independence assumptions to various degrees. It is expected that the more a value function violates the assumptions, the worse the performance. The best we can hope for is that this decrease does not occur in a precipitous manner.

The first issue we have to address in this experiment is to define a measure of domain theory violation. Since we are dealing with domain theory composed of independence assumptions, we will call this measure *degree of violating the independences* or *DOVI*. Given a domain theory D and a value function v, what would be a good measure of v violating D? Note that we can view D as a set of value functions that satisfy D. If we have a measure of distance d between two value functions, then we can define a violation measure as the distance between v and the member of D that is closest to v. To define the distance between two value functions, we use the *probabilistic distance*, proposed by Ha and Haddawy [2]. According to this measure, the distance between two value functions is determined by the probability that they disagree in their relative rankings of two randomly chosen outcomes. The other issue, which is mainly technical, is the computation of the minimum distance. Since there are infinitely many value functions that satisfy the domain theory D, we approximate the violation measure by sampling from D. We achieve this by sampling additive value functions with randomly generated coefficients. Our experiments show that this method of sampling settles to a measure point after about 2000 iterations.

Figure 3 shows the results of our robustness analysis. We measure performance on several preference orders with DOVI varying from 0.04 to 0.5. The DOVI does not exceed 0.5 due to the fact that value functions consistent with the domain theory intuitively “cover” the space of preference orders very well. The performance of the biased network remains essentially the same even for a preference order whose DOVI is 0.19. The performance of the unbiased network is dominated in all cases.

5. REDUCING USER INPUT

One way to easily obtain a large number of examples of pairwise preferences is to have the user indicate preference between *classes* of items rather than just individual items. If we have n items in each class, we then have n^2 pairwise preferences implicitly represented. But it is not clear whether examples generated in this way will have as much information content as those generated randomly since each item will appear in n examples. So we empirically evaluated this approach on one value function.

We simulated a user’s binary categorization of flights into “like” and “dislike” groups. We scored flights according the first value function (-1t-100c-50l) and selected the top one third of our flights as the "like" set and the bottom one third as the “dislike” set. We then used these sets to generate examples of pairwise preferences. The results are shown in Figure 4. Both the biased and unbiased networks are able to achieve significantly higher accuracy using fewer flights than when using randomly generated examples. The KBANN

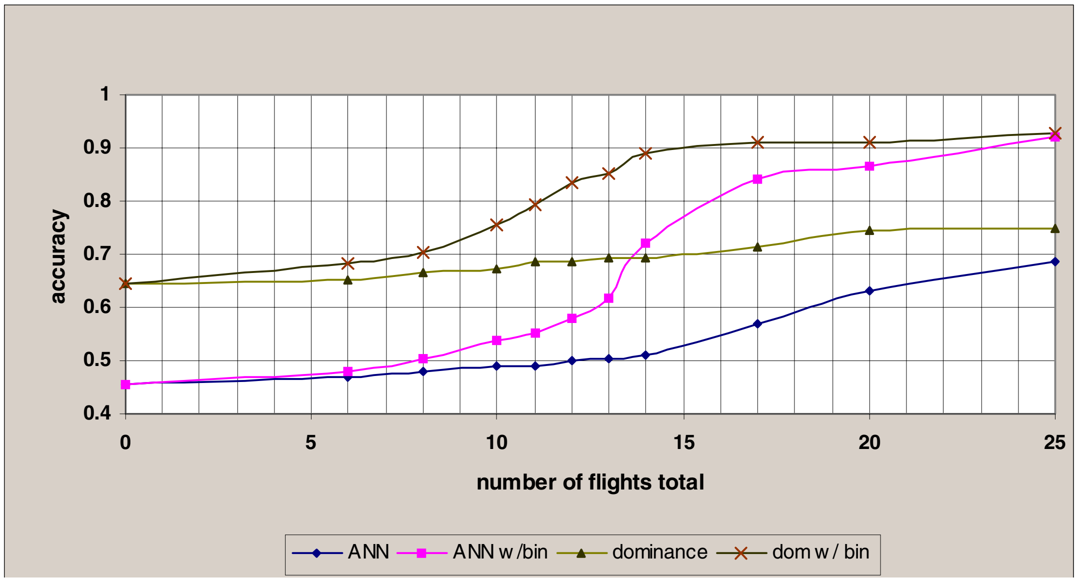

Figure 4: Accuracy of training on examples generated from binary categorization vs those generated randomly.

network achieves an accuracy of 89% using only 14 flights – a reduction of 85% in the number of flights needed.

6. RELATED WORK

The Wisconsin Adaptive Web Assistant (WAWA) [6] is a system that finds and recommends web pages. It uses KBANN to determine if a user would like to visit a given web page. The attribute set is a localized bag-of-words. WAWA can learn from advice given to the system as well as from example classifications. The ScorePage neural network outputs a rating measure for each web page. In WAWA instances are classified as like and dislike. The domain theory is specified in the form of rules providing advice, e.g. "WHEN consercutiveWordsInHypertext (intelligent user interface) STRONGLY SUGGEST FOLLOWING HYPERLINK."

The Decision-Theoretic Interactive Video Advisor (DIVA) [4] uses pairwise preferences to represent user preferences in the domain of movie recommendation. Pairwise preferences are obtained by having the user classify movies into like, ok, and dislike categories. DIVA is a collaborative filtering system that maintains a database of user preference structures. The system makes recommendations by computing similarity between preference structures.

Chajewska, et al [1] present an approach to eliciting preferences under uncertainty that uses prior information in the form of a probability distribution over possible utility values of the user, and updates this based on information elicited from the user. Questions follow the standard gamble pattern in which the user is asked to express preference between some outcome for certain and a gamble with the best and worst outcomes as "prizes." Value of information is used to decide which questions will best help toward making a rational decision. The objective is to make good decisions using a small number of questions.

7. ACKNOWLEDGEMENTS

We thank Jude Shavlik and David Page for helpful discussions.

8. REFERENCES

1. Chajewska, U., Koller, D., Parr, R. Making rational decision using adaptive utility elicitation, in Proc. AAAI-00, Aug 2000, pp 363-369.
2. Ha, V. & Haddawy, P. Toward case-based preference elicitation: Similarity measures on preference structures, in Proc. UAI98 (July 1998), pp 193-201.
3. Keeney, R. L., and Raiffa, H. *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. Wiley, New York, 1976.
4. Nguyen, H. & Haddawy, P. The decision-theoretic interactive video advisor, in Proc. UAI99 (Aug 1999), pp 494-501.
5. Shavlik, J. & Towell, G. An approach to combining explanation-based and neural learning algorithms. *Connection Science*, 1(3): 233-255, 1989.
6. J. Shavlik, S. Calcari, T. Eliassi-Rad, & J. Solock. An Instructable, Adaptive Interface for Discovering and Monitoring Information on the World-Wide Web, in Proc. IUI-99.