Making CP-Nets (More) Useful

Thomas E. Allen
University of Kentucky
Department of Computer Science
329 Rose Street
Lexington, Kentucky 40506-0633
www.cs.uky.edu/∼teal223/
thomas.allen@uky.edu

Preferences have been studied in philosophy, economics, psychology, and computer science and have a wide range of applications, such as e-commerce, recommender systems, decision support systems, and control of automated systems. A variety of methods have been proposed for modeling preferences. The one that I consider here is that of *conditional preference networks* (CP-nets). First studied by Boutilier et al. (2004), CP-nets exploit the power of conditional *ceteris paribus* preference rules to enable (in many cases) a compact representation of human preferences.

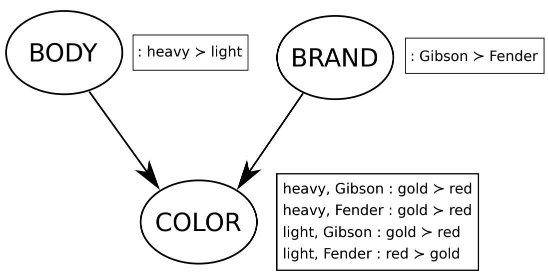

Figure 1: A Strict, Complete, Binary, Acyclic CP-net

Definition 1. *A* CP-net $\mathcal{N}$ *is a directed graph. Each node represents a variable* $X_i \in \mathbf{V}$ *and is annotated with a* conditional preference table (CPT) *describing the subject's preferences over the domain of* X_i *given its dependencies. An edge* (X_h, X_i) *indicates that the preferences over* X_i *depend directly on the value of* X_h.

Various problems involving CP-nets have been studied, including 1. finding the *optimal* and k-best *outcomes*, 2. using the model to *reason* which if either of a pair of outcomes the subject will prefer, 3. *learning* a CP-net from outcome comparison data or by eliciting queries from the subject, and 4. *aggregating* the CP-nets of two or more subjects.

CP-nets have much appeal. However, the study of CP-nets has not advanced sufficiently for their widespread use in complex, real-world engineering applications. These limitations include: 1. Many studies of CP-nets are restricted to preferences that are strict, binary, and complete. 2. The dependency graph is usually assumed to be acyclic. 3. It is generally assumed that all features on which an outcome depends are known *a priori*. 4. Most learning algorithms do not allow for noise or apparently inconsistent preferences. 5. While various algorithms have been proposed, in most cases these have not been compared experimentally. Moreover, such experiments depend on the availability of factored preference data from human subjects and better algorithms for randomly generating CP-nets.

It is easy to find simple, real-life examples of preferences that cannot be modeled under such restrictive assumptions as these. Binary valued features do arise in everyday life ("Do you prefer tea or coffee?"), but multivalued features are far more common ("We offer a selection of fine teas"). A child playing with toys in the nursery may prefer only those that are a favorite color, but otherwise be indifferent; her preferences are thus neither strict nor complete. Cycles, despite the potential complexity issues they present (Goldsmith et al., 2008), arise naturally in situations in which a subject prefers to coordinate features (see Fig. 2). My present research represents a step toward addressing these limitations to make CP-nets more useful and perhaps facilitate their adoption for complex engineering applications.

At the Algorithmic Decision Theory conference (Guerin, Allen, and Goldsmith, 2013), we presented a heuristic algorithm, earlier proposed by Guerin (2012), for learning CP-nets from user queries. Our algorithm differs from other approaches in that: 1. It learns through elicitation rather than passive observation. 2. It can employ a richer set of queries, such as outcome comparisons that can differ in any number

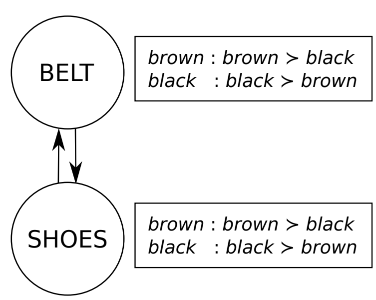

Figure 2: Cyclic CP-net Arising from Coordination

of variables and attribute comparisons that offer a heuristic for faster learning. 3. The algorithm is robust: it will always output a CP-net, never failure. 4. While the algorithm does not always recover the original CP-net, the learned CP-net that it outputs is guaranteed to be consistent with the original on all queries encountered in the learning process. 5. The algorithm is efficient, assuming a bound on in-degree; that is, if a preference order can be represented as a CP-net, the algorithm learns a CP-net in time $\mathcal{O}(n^p)$, where n is the number of nodes (features) and p is a bound on the number of parents a node may have. 6. We employed an experimental approach to evaluate our algorithm, simulating the query process using randomly generated CP-nets. For the conference paper, I planned and performed additional experiments, including a series that used statistical sampling when exhaustive analysis was infeasible. I also modified the algorithm that randomly generated CP-nets to exclude the possibility of degenerate CPTs that could conflict with the dependency graph. The resulting set of experiments demonstrated that the learned CP-nets agree with the originals on a high percent of non-training preference comparisons.

Most research on CP-nets assumes strict, complete preferences over binary variables. In an invited paper at the Allerton Conference (Allen, 2013), I relaxed these assumptions to explore the problems of learning and reasoning with CP-nets that can model preferences over multi-valued variables over which the subject may be indifferent. I presented the case that such CP-nets are necessary for many simple, real-world problems. I then showed how to leverage the power of SAT solvers to learn and reason with such CP-nets. The problem of reasoning with CP-nets is generally framed as one of searching for a transitive sequence from one outcome to the other that is consistent with the rules in the CPTs. The planning algorithm that I proposed outputs such a *flipping sequence*, provided at least one exists and that its length can be bounded by a constant. However, Boutilier et al. (2004) proved that for multivalued, incomplete CP-nets, the sequence may be exponentially long in the number of variables in the worst case. Thus, the problem of reasoning with CP-nets is not in NP: even if we could somehow guess a correct flipping sequence, the problem of verifying the sequence could be intractable. Nevertheless, there is reason to believe that this worst-case scenario is rare. In a series of experiments, I showed that most flipping sequences are likely to be short and that a longest flipping sequence (the diameter of the CP-net) is also likely to be short, at least in the case of randomly generated CP-nets, I further showed that, even if very long flipping sequences do occasionally exist in learned or elicited CP-nets, they are unlikely to provide useful information about the actual preferences of a human subject if there is some small probability ϵ that the rules in the CPTs are noisy.

In our ADT paper (Guerin, Allen, and Goldsmith, 2013) we limited our study to learning strict, binary CP-nets. For the journal paper, I am extending the algorithm to allow for generally multivalued variables. Moreover, the subject would be permitted to answer with indifference or to decline to answer altogether, as well as well as expressing a strict preference over outcomes or features. Additionally, since the algorithm attempts to approximate the subject's preferences as closely as possible rather than recovering the original CP-net exactly, I conjecture that it will prove resilient in the presence of noise or occasional inconsistent responses. I am planning new experiments to test this assumption.

My next project involves extending my earlier work (Allen, 2013) on the expected length of flipping sequences and the diameter of a CP-net. For that I plan to perform a further series of experiments involving 1. randomly generated networks with features that vary in the number of values (rather than domains of uniform size), 2. networks with particular structures in which long flipping sequences may be more common (e.g., chain-shaped CP-nets), and 3. CP-nets learned from data as well as those that are generated randomly. From these experiments I hope to obtain a formal statistical model of the expected length of a flipping sequence and diameter of a CP-net parameterized by its number of variables, the size of their respective domains, the number of rules in the CPTs, and constraints on the structure of the network. I hope to have much of this work completed prior to the AAAI conference.

Following the conference, I plan to study the problem of learning CP-nets in which the subject prefers to coordinate the values of two or more features, as in the example in Fig. 2. To my knowledge, all present CP-net learning algorithms fail to represent correctly the subject's preferences when features are coordinated in this way. Present algorithms either misinterpret the preferences or fail to output a CP-net at all. I hope instead to detect such preferences directly and resolve the cyclic relationship among the features by 1. merging the coordinated variables, 2. inferring the presence of a latent variable on which the nodes that are involved in the cycle mutually depend, or 3. learning and outputting a consistent CP-net that contains an actual cycle in the dependency graph. I also plan to study the complexity of cyclic CP-nets in which the length of the longest cycle is bounded by a small constant.

References

Allen, T. E. 2013. CP-nets with indifference. In *Communication, Control, and Computing (Allerton), 2013 51st Annual Allerton Conference on*, 1488–1495. IEEE.

Boutilier, C.; Brafman, R. I.; Domshlak, C.; Hoos, H. 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.

Goldsmith, J.; Lang, J.; Truszczyński, M.; and Wilson, N. 2008. The computational complexity of dominance and consistency in CP-nets. *Journal of Artificial Intelligence Research* 33:403–432.

Guerin, J. T.; Allen, T. E.; and Goldsmith, J. 2013. Learning CP-net preferences online from user queries. In *Proc. ADT*, LNAI. Springer.

Guerin, J. T. 2012. *Graphical Models for Decision Support in Academic Advising*. Ph.D. Dissertation, University of Kentucky.