Dynamic and Probabilistic CP-nets

Cristina Cornelio (PhD student) and Francesca Rossi (Supervisor)

University of Padua - Department of Mathematics
Via Trieste, 63 - 35131, Padua, Italy
E-mail: {cornelio,frossi}@math.unipd.it

Abstract. In this paper we present a generalization of conditional preference networks (CP-nets) that incorporates uncertainty. CP-nets are a formal tool to model qualitative conditional preference statements about preferences over a set of objects. They are static structures, both in their ability to capture dependencies between objects and in their expression of preferences over features of a particular object. Moreover, CP-nets do not provide the ability to express uncertainty over the preference statements. We present and study a generalization of CP-nets which supports changes and allows for encoding uncertainty over the structure of the dependency links and over the individual preference relations.

Keywords: CP-nets, Preferences, Graphical Models, Probabilistic Reasoning

1 Introduction

CP-nets are used to model conditional information about preferences [1]. Preferences play a key role in automated decision making [6] and there is some experimental evidence suggesting qualitative preferences are more accurate than quantitative preferences elicited from individuals in uncertain information settings [10]. CP-nets are compact, arguably quite natural, intuitive in many circumstances, and widely used in many applications in computer science such as recommendation engines [5].

Real life scenarios are often dynamic. A user can change his preferences over time or the system under consideration can change its laws. Thus, we need a dynamic structure that can respond to change, without the need to completely rebuild the structure. We often meet situations characterized by some form of uncertainty. We may have some missing informations or we may be uncertain about our preferences or on what features our preferences depend. Thus, we need a structure that includes probabilistic information. The need for encoding uncertain, qualitative information has seen some work in the recommendation engine area [4, 7].

Consider a household of two people and their Netflix account. The recommendation engine only observes what movies are actually watched, what time they are watched, and their final rating. Let us say that one person prefers drama movies to action ones while the other has the opposite preference. When making a recommendation about what type of movie to watch, the engine may have several solid facts: comedies may always be watched in the evening, so we can put a deterministic, causal link between time of day and type of movie. However, we cannot observe which user is sitting in front of the television at a given time.

There is strong evidence from the behavioral social sciences showing that adding uncertainty to preference frameworks may be a way to reconcile transitivity when eliciting input from users [8]. Using this idea, we add a probabilistic dependency between our belief about who is in front of the television and what we should recommend. Thus we need a dynamic structure that encode uncertainty. We propose and study the *Dynamic PCP-nets* (Dynamic Probabilistic CP-nets) structure which allow for uncertainty and online modification of the dependency structure and preferences.

2 Background

In this section we are going to introduce the main background concepts concerning CP-nets (the starting point of our work) and Bayesian Networks, that we use as a tool to reason about PCP-nets.

2.1 CP-nets

CP-nets are a graphical model for compactly representing conditional and qualitative preference relations [1]. They exploit conditional preferential independence by decomposing an agent's preferences via the *ceteris paribus* assumption (all other things being equal). CP-nets bear some similarity to Bayesian Networks (see Section 2.2). Both use directed graphs where each node stands for a domain variable, and assume a set of features $F = \{X_1, \ldots, X_n\}$ with finite domains $\mathcal{D}(X_i)$. For each feature X_i, each user specifies a set of *parent* features $Pa(X_i)$ that can affect her preferences over the values of X_i. This defines a *dependency graph* in which each node X_i has $Pa(X_i)$ as its immediate predecessors. Given this structural information, the user explicitly specifies her preference over the values of X_i for *each complete assignment* on $Pa(X_i)$. This preference is a total or partial order over $\mathcal{D}(X_i)$ [1]. Each node X_i has a *conditional preference table (CP-table)* that contains the conditional preference statements.

Note that the number of complete assignments over a set of variables is exponential in the size of the set, but we assume that there are bounds on $|Pa(X)|$ and on $|\mathcal{D}(X)|$.

In this paper we focus on *acyclic* CP-nets (in which the dependency graph is acyclic). The semantics of CP-nets depends on the notion of a *worsening flip* that is a change in the value of a variable to a value which is less preferred by the cp-statement for that variable. We say that one outcome (an assignment of all the variables) α is *better* than another outcome β (written $\alpha > \beta$) if and only if there is a chain of worsening flips from α to β. This definition induces a pre-order over the outcomes.

In general, finding optimal outcomes and testing for optimality in this ordering is NP-hard but, in acyclic CP-nets, the unique optimal outcome can be found via a *sweep forward procedure* [1] and takes polynomial time in the size of the CP-net (recall that the number of parents is bounded).

2.2 Bayesian Networks

A Bayesian network (BN) is a directed acyclic graph (DAG) where each node $v \in V$ corresponds to a random variable. If there is a directed edge from node X to node Y, X is said to be a *parent* of Y. Each node X_i has a conditional probability distribution $\mathbb{P}(X_i|Parents(X_i))$ that, in the case of discrete variables, is stored in the *conditional probability table (CPT)* corresponding to that variable.

Inference in a BN corresponds to calculating $\mathbb{P}(X|E)$ where both X and E are sets of variables of the BN, or to finding the most probable assignment for X given E. The variables in E are called *evidence*.

There are three standard inference tasks in BNs: *belief updating*, which is finding the probability of a variable or set of variables, possibly given evidence; *most probable explanation (MPE)*, that is the most probable assignment for all the variables given evidence; and *maximum a-posteriori hypothesis (MAP)*, where we are interested in a subset of m variables $A_1, \cdots, A_m$ and we want to compute the most probable assignment of $\{A_1, \cdots, A_m\}$ by summing over the values of all combinations of $V \setminus \{A_1, \cdots, A_m\} \cup E$, where E is a (possibly empty) set of evidence variables.These inference tasks are computationally hard. However, they can be solved in polynomial time if we impose some restrictions on the topology of the BNs such as bounding the induced width [2, 3].

3 Probabilistic CP-nets (PCP-nets)

We define a generalization of traditional CP-nets with probabilities on individual cp-statements as well as on the dependency structure. In general, we assume these probabilities are independent. This allows us to use algorithms and techniques from BNs to obtain to efficiently compute restricted dependency structures.

A PCP-net is a CP-net where each dependency link is associated with a probability of existence; and for each feature A, instead of giving a preference ordering over the domain of A, we give a probability distribution over the set of all preference orderings for A. More precisely, given a feature A in a PCP-net, its *PCP-table* is a table associating each combination of the values of the parent features of A (including a null value to account for the possible non-existence of the dependency link) to a probability distribution over the set of total orderings over the domain of A.

Example 1. Consider the PCP-net $\mathcal{C}$ with three features, A, B and C, with domains $\mathcal{D}_F = \{f_1, f_2\}$ with $F \in \{A, B, C\}$. The preferences on C depend on the assignment to A with probability p. The structure and PCP-tables are defined as follows:

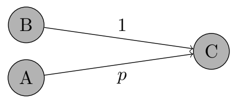


Fig. 1. PCP-net $\mathcal{C}$

A orderings	$\mathbb{P}$
$a_1 > a_2$	r_A
$a_2 > a_1$	$1 - r_A$

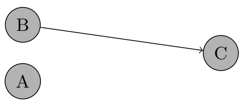
B orderings	$\mathbb{P}$
$b_1 > b_2$	r_B
$b_2 > b_1$	$1 - r_B$

A & B	C orderings	$\mathbb{P}$
$a_1 b_1$	$c_1 > c_2$ $c_2 > c_1$	q_1 $1 - q_1$
$a_2 b_1$	$c_1 > c_2$ $c_2 > c_1$	c_2 $1 - q_2$
$null_A b_1$	$c_1 > c_2$ $c_2 > c_1$	q_3 $1 - q_3$

A values	B orderings	$\mathbb{P}$
$a_1 b_2$	$c_1 > c_2$ $c_2 > c_1$	q_4 $1 - q_4$
$a_2 b_2$	$c_1 > c_2$ $c_2 > c_1$	q_5 $1 - q_5$
$null_A b_2$	$c_1 > c_2$ $c_2 > c_1$	q_6 $1 - q_6$

Given a PCP-net $\mathcal{C}$, a *CP-net induced by* $\mathcal{C}$ has the same features and domains as $\mathcal{C}$. The dependency edges of the induced CP-net are a subset of the edges in the PCP-net which must contain all edges with probability 1. CP-nets induced by the same PCP-net may have different dependency graphs. Moreover, the CP-tables are generated accordingly for the chosen edges. For each independent feature, one ordering over its domain (i.e., a row in its PCP-table) is selected, and for dependent features, an ordering is selected for each combination of the values of parent features. Each induced CP-net has an associated probability obtained from the PCP-net by taking the product of the probability of the chosen edges (obtained by multiplying the probabilities p_i for each active dependencies and $1 - p_i$ for each of the inactive ones, where p is the probability of the existence of an edge), and the probability of the chosen orderings.

Example 2. Given the PCP-net of Example 1, an induced CP-net is:



A orderings
$a_2 > a_1$

B orderings
$b_1 > b_2$

B values	C orderings
b_1	$c_2 > c_1$
b_2	$c_1 > c_2$

Fig.2. Induced CP-net

This induced CP-net has probability: $\mathbb{P} = (1 - p) \cdot (1 - r_A) \cdot r_B \cdot (1 - q_3) \cdot q_6$.

Since we have a probability distribution on the set of all induced CP-nets, it is important to be able to find the *most probable induced CP-net*. We are also interested in finding the *most probable optimal outcome*.

3.1 The most probable induced CP-net

We reduce the problem of finding the most probable induced CP-net to that of finding an assignment with maximal joint probability of an appropriately defined BN (*General-network or G-net*, of which we omit the description for lack of space). We prove that *given a PCP-net $\mathcal{C}$ and the corresponding G-net N, there is a one-to-one correspondence between the assignments of N and the induced CP-nets of $\mathcal{C}$ and the probability of realizing one of its induced CP-nets is the joint probability of the corresponding assignment in N*. Thus, computing the assignment with maximal joint probability of the G-net, we find the most probable induced CP-net.

3.2 The most probable optimal outcome

To find the most probable optimal outcome (the outcome that occurs with the greatest probability as the optimal one in the set of induced CP-nets), we must make use of another BN (*Optimal network*). First we transform the given PCP-net $\mathcal{C}$ into a PCP-net $\mathcal{C}^T$ with probabilities only in the PCP-tables (not on edges), and then we build the *Opt-net* to find the most probable optimal outcome associated with $\mathcal{C}^T$ and thus with the PCP-net $\mathcal{C}$. We obtain that *given a PCP-net $\mathcal{C}$ and the Opt-net of $\mathcal{C}^T$, there is a one-to-one correspondence between the assignments (with non-zero probability) of the Opt-net and the outcomes that are optimal in at least one induced CP-net of $\mathcal{C}$. Given a PCP-net $\mathcal{C}$, the probability that an outcome is optimal is the joint probability of the corresponding assignment in the optimal network of $\mathcal{C}^T$. If no such corresponding assignment exists, then the probability of being optimal is 0.* To find the most probable optimal outcome for a PCP-net $\mathcal{C}$, it is sufficient to compute the assignment with the maximal joint probability on the optimal network for $\mathcal{C}^T$.

4 Dynamic Probabilistic CP-nets

We now turn our attention to dynamic modifications to the structure of a PCP-net. One can think of modifying the structure of the PCP-net by adding or removing an arc or a feature or setting an ordering for a variable. These changes can be implemented in an efficient way and their effects on computing the most probable optimal outcome and the most probable induced CP-net are minimal, in terms of complexity.

To add or delete a dependency or a feature we just update the respective probability tables, including $null_i$ values. This may involve deleting redundancy when we delete a feature. Additionally, due to the independence assumptions, we can modify probabilities over ordering and features at a local level, with no need to recompute the entire structure when new information is added.

When we modify a PCP-net $\mathcal{C}$ we also need to modify the associated BN: G-net and Opt-net, but all the updating steps take constant time in the size of the connected component of the change. When we update the most probable induced CP-net, we must, in the worst case, recompute the whole maximal joint probability of the G-net. Similarly, the updating of the most probable optimal outcome, in the worst case, could involve the recomputing of the whole maximal joint probability of the Opt-net.

5 PCP-nets and Induced CP-nets

A PCP-net defines a probability distribution over a set of induced CP-nets. However, this step is not always reversible: given a probability distribution over a set of CP-nets, all with the same features and domains, there may be no PCP-net such that the given CP-nets are its induced CP-nets. However, the function that maps a PCP-net to its set of induced CP-nets is injective. These considerations give us interesting clues about the practicality of eliciting and aggregating CP-nets. We have thus considered how to use PCP-nets in a multi-agent setting, where classical CP-nets have already been considered [9]. We see the PCP-nets as a compact way to model the preferences of a population. In some instances, we can derive an exact PCP-net from a population of users, each one with his elicited CP-net. This process is possible in the case that a function that maps a PCP-net to this set of induced CP-nets exists. If this set of induced CP-nets has no a inverse image then we can only obtain an approximated PCP-net, the one, for example, that generates a set of induced CP-nets that has the minimum distance from the starting set (for a given notion of distance).

In this context the most probable induced CP-net represent the most common profile of user in the population (the user that represent better the population) and the most probable optimal outcome is the the will of the community, the common preferences.

6 Current and Future Work

Currently we are interested in development of the aggregation approach briefly introduced before (see Section 5) and in learning PCP-nets. For this latter purpose we started from the known procedures for learning CP-net [11] and we have tried to generalize this processes (we have some results only on the case of separable PCP-nets). We plan to study dominance queries and optimality tests in dynamic PCP-nets, as well as to study appropriate eliciting methods for both preferences and probabilities. Additionally, we have made several assumptions to bound the complexity of PCP-nets; we would like to relax these bounds or obtain results about approximability when these assumptions are lifted.

References

1. Boutilier C., Brafman R., Domshlak C., Hoos H., Poole D.: *CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements.* J. Artif. Intell. Res. (JAIR) 21, 135191 (2004)
2. D'Ambrosio B.: *Inference in bayesian networks.* AI Magazine 20(2), 21 (1999)
3. Dechter R.: Bucket elimination: *A unifying framework for reasoning.* Artificial Intelligence 113(1-2), 4185 (1999)
4. Faltings B., Torrens M., Pu P.: *Solution generation with qualitative models of preferences.* Computational Intelligence 20(2), 246263 (2004)
5. Fürnkranz J., Hüllermeier E.: *Preference learning: An introduction.* Springer (2010)
6. Goldsmith J., Junker U.: *Preference handling for artificial intelligence.* AI Magazine 29(4), 9 (2009)
7. Price R., Messinger P.R.: *Optimal recommendation sets: Covering uncertainty over user preferences.* Proceedings of the 20th National Conference on Artificial Intelligence (AAAI-05). vol. 20, pp. 541548 (2005)
8. Regenwetter M., Dana J., Davis-Stober C.P.: *Transitivity of preferences.* Psychological Review 118(1), 42 (2011)
9. Rossi F., Venable K., Walsh T.: *mCP-nets: representing and reasoning with preferences of multiple agents.* Proceedings of the 19th National Conference on Artificial intelligence (AAAI-04). pp. 729734. AAAI Press (2004)
10. Roth A.E., Kagel J.H.: *The handbook of experimental economics*, vol.1. Princeton University Press Princeton (1995)
11. Chevaleyre Y., Koriche F., Lang J., Mengin J., Zanuttini B.: *Learning Ordinal Preferences on Multiattribute Domains: the Case od CP-nets* in Preference Learning pp. 273-296 (2010)