Updates and Uncertainty in CP-nets

Cristina Cornelio[1], Judy Goldsmith[2], Nicholas Mattei[3], Francesca Rossi[1], and K. Brent Venable[4]

[1] University of Padova, Italy, {cornelio,frossi}@math.unipd.it.
[2] University of Kentucky, USA, goldsmit@cs.uky.edu.
[3] NICTA and UNSW, Australia, nicholas.mattei@nicta.com.au.
[4] Tulane University and IHMC, USA, kvenabl@tulane.edu.

Abstract. In this paper we present a two-fold generalization of conditional preference networks (CP-nets) that incorporates uncertainty. CP-nets are a formal tool to model qualitative conditional statements (cp-statements) about preferences over a set of objects. They are inherently 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, expressed in probabilistic terms, over the structure of the dependency links and over the individual preference relations.

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

1 Introduction

CP-nets are used to model conditional information about preferences [2]. Preferences play a key role in automated decision making [9] and there is some experimental evidence suggesting qualitative preferences are more accurate than quantitative preferences elicited from individuals in uncertain information settings [19]. CP-nets are compact, arguably quite natural, intuitive in many circumstances, and widely used in many applications in computer science such as recommendation engines [8].

Real life scenarios are often dynamic. A user can change his mind over time or the system under consideration can change its laws. Preferences may change over time. Thus, we need a structure that can respond to change through updates, without the need to completely rebuild the structure. Additionally, we often meet situations characterized by some form of uncertainty. We may be uncertain about our preferences or on what features our preferences depend. In order to model this, we need a structure that includes probabilistic information. The need for encoding uncertain, qualitative information has seen some work in the recommendation engine area [7,16] and is a motivating example.

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. There are two people in this house and let us say that one prefers drama movies to action movies 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 [17], among other nice properties [12]. Using this idea, we add a probabilistic dependency between our belief about who is in front of the television and what we should recommend. We may want to update the probability associated with this belief based on the browsing or other real-time observable habits of the user. To do this we need a updateable and changeable structure that allows us to encode uncertainty.

We propose and study the complexity of reasoning with *PCP-nets*, for Probabilistic CP-nets, which allow for uncertainty and online modification of the dependency structure and preferences. PCP-nets provide a way to express probabilities over dependency links and probability distributions over preference orderings in conditional preference statements. Given a PCP-net, we show how to find the most probable optimal outcome. Additionally, since a PCP-net defines a probability distribution over a set of CP-nets, we also show how to find the most probable induced CP-net.

2 Background and Related Work

Probabilistic reasoning has received a lot of attention in Computer Science [8] and other areas [12]. Elicitation and modeling of preferences has also been considered in probabilistic domains such as POMDPs [3]. Recently, another generalization of CP-nets to include probabilities was introduced by Bigot et al. [1]. The model proposed by Bigot et al. restricts probabilities to be defined on orderings. We allow for probabilities on edges but, as we will show, this is a somewhat redundant specification that is useful for elicitation. Moreover, Bigot et al. focus on optimization and dominance testing in the special tractable case of tree-structured networks, we base our algorithmic approach on a more general connection with Bayesian networks. Reconciling these two models is an important direction for future work.

2.1 CP-nets

CP-nets are a graphical model for compactly representing conditional and qualitative preference relations [2]. They exploit conditional preferential independence by decomposing an agent's preferences via the *ceteris paribus* (cp) assumption (all other things being equal). CP-nets bear some similarity to Bayesian networks (see 2.2). Both use directed graphs where each node stands for a domain variable, and assume a set of features $F = \{X_1, \dots, X_n\}$ with finite domains $\mathscr{D}(X_1), \dots, \mathscr{D}(X_n)$. 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 $\mathscr{D}(X)$ [2].

Note that the number of complete assignments over a set of variables is exponential in the size of the set. Throughout this paper, we assume there is an implicit constant that specifies the maximum number of parent features, $|Pa(X)|$, that any feature may have. With this restriction, and an implicit bound on $|\mathscr{D}(X)|$, we can and do treat the size of the conditional preference representation for any X as a constant.

An *acyclic* CP-net is one in which the dependency graph is acyclic. A CP-net need not be acyclic. For example, my preference for the entree may depend on the choice of the main course, and my preference for the main course may depend on the choice of the entree. However in this paper we focus on acyclic CP-nets.

The semantics of CP-nets depends on the notion of a *worsening flip*. A worsening flip 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 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 preorder over the outcomes.

In general, finding optimal outcomes and testing for optimality in this ordering is NP-hard. However, in acyclic CP-nets, there is only one optimal outcome and this can be found in as many steps as the number of features via a *sweep forward procedure* [2]. We sweep through the CP-net, following the arrows in the dependency graph and assigning at each step the most preferred value in the preference table. Each step in the sweep forward procedure is exponential in the number of parents of the current feature, and there are as many steps as features. In this paper we assume the number of parents is bounded, so this algorithm takes time polynomial in the size of the CP-net.

Determining if one outcome is better than another according to this ordering (called a dominance query) is NP-hard even for acyclic CP-nets [6,10]. Whilst tractable special cases exist, there are also acyclic CP-nets in which there are exponentially long chains of worsening flips between two outcomes.

2.2 Bayesian Networks

Bayesian networks (BNs) allow for a compact representation of uncertain knowledge and for a rigorous way of reasoning with this knowledge [15]. A BN is a directed graph where each node corresponds to a random variable; the set of nodes is denoted by V; a set of directed edges connects pairs of nodes (if there is an edge from node X to node Y, X is said to be a *parent* of Y); the graph has no directed cycles and hence is a directed acyclic graph (DAG); each node X_i has a conditional probability distribution $\mathbb{P}(X_i|Parents(X_i))$ that quantifies the effect of the parents on the node. If the nodes are discrete variables, each X_i has a *conditional probability table (CPT)* that contains the conditional probability distribution, $\mathbb{P}(X_i|Parents(X_i))$. Each CPT row must therefore have probabilities that sum to 1.

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; finding the *most probable explanation (MPE)*, that is, the most probable assignment for all the variables given evidence; and finding the *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.

The 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 [4, 5]. Given an ordering of the variables of a BN, these algorithms have a number of steps linear in the number of variables, and each step is exponential in the number of variables preceding the current one in the ordering and connected to it in the BN graph. The largest of these numbers is the induced width of the graph of the BN. Different variable orderings give steps with different complexity. Finding a good variable ordering is a difficult problem. If we assume the induced width is bounded, the overall algorithm is polynomial, and if $|Pa(X)|$ is bounded by a constant, then the induced width is also bounded.

3 Probabilistic CP-nets

We define a generalization of traditional CP-nets with probabilities on individual cp-statements as well as on the dependency structure.We assume that the probabilities expressed over the dependency structure is consistent with the probabilities expressed over the variable orderings themselves. A model defined in this way allows us to use algorithms and techniques from BNs to efficiently compute outputs for common queries when the size of the dependency graph is bounded.

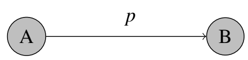
A PCP-net (for Probabilistic CP-net) is a CP-net where: **(1)** each dependency link is associated with a probability of existence consistent with the given variable ordering; and **(2)** for each feature A, instead of giving a preference ordering over the domain of A, we give a probability distribution 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 to a probability distribution over the set of total orderings over the domain of A.

Probabilities expressed on the dependency links and the corresponding PCP-tables are not independent. If we consider all the possible ways in which we can obtain a CP-table from PCP-table by choosing specific orderings we see that we can divide the CP-tables into two classes: those representing a “true” dependency and those representing independence of the child feature. Each induced CP-table is associated to the joint probability of the orderings it contains. The probability of activation or non-activation of a dependency *must* coincide with the sum of probabilities associated to the CP-tables where the dependency is activated or not activated. Otherwise, the probability of the dependency and the probability of the ordering are not reconcilable and the structure itself expresses an impossible relationship.

Example 1. Consider the PCP-net $\mathscr{C}$ shown with two features, A and B, with domains $\mathscr{D}_A = \{a_1, a_2\}$ and $\mathscr{D}_B = \{b_1, b_2\}$. The preferences on B depend on the assignment to A with probability p. Given the probability assignment to the orderings of B given A we have that $p = q_1 \cdot (1 - q_2) + (1 - q_1) \cdot q_2$.

Structure:



Feature A:

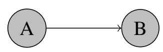
A orderings	$\mathbb{P}$
$a_1 > a_2$	r
$a_2 > a_1$	$1-r$

Feature B:

A values	B orderings	$\mathbb{P}$
a_1	$b_1 > b_2$	q_1
	$b_2 > b_1$	$1-q_1$
a_2	$b_1 > b_2$	q_2
	$b_2 > b_1$	$1-q_2$

The induced CP-net with probability $\mathbb{P} = (1-r)\cdot(1-q_1)\cdot q_2$ is shown below.

Structure:



Feature A:

A orderings
$a_2 > a_1$

Feature B:

A values	B orderings
a_1	$b_2 > b_1$
a_2	$b_1 > b_2$

Given a PCP-net $\mathscr{C}$, a *CP-net induced by* $\mathscr{C}$ has the same features and domains as $\mathscr{C}$. 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, therefore, 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. Similarly, 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 probabilities of the deterministic orderings chosen in the CP-net.

One may note that the probabilities on edges are redundant whenever the probabilities in the PCP-tables are completely specified. However, we have chosen the presented formalism as it may be useful for elicitation purposes. Consider a settings where we are attempting to determine the strength of a relationship between two variables, such as the relationship between time of day and type of movie desired. It may be easier for people to describe this relationship directly rather than express the underlying joint probability distribution as humans are generally poor at estimating and working with probability directly [20]. Using this elicitation method we could then assume some underlying distribution for the variable ordering (skewed one way or another based on evidence). We leave an exploration of this topic for future work and focus on the base case, where PCP-nets are consistent, for the current work.

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*. Given a PCP-net and an outcome (that is, a value for each feature), the probability of such an outcome being optimal corresponds to the sum of the probabilities of the CP-nets that have that outcome as optimal.

4 Reasoning with PCP-nets

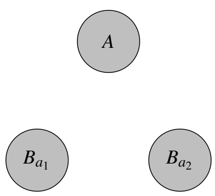
Given a PCP-net we study mainly two tasks: finding the most probable induced CP-net and finding the most probable optimal outcome. These two reasoning tasks have slightly different semantics and may be of use to different groups in the preference reasoning community. The most probable induced CP-net is analogous, in our Netflix example from earlier, to the CP-net that most likely maps onto a viewer in the household. Whereas, the most probable optimal outcome would be what a recommendation engine should suggest to maximize the probability of a correct recommendation. One is an aggregated model, that still retains usefulness for prediction and sampling while the other is an aggregated outcome, that maximizes the probability of being correct.

4.1 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.

Given a PCP-net $\mathscr{C}$, we define the BN called *general network*, or *G-net($\mathscr{C}$)*, associated with $\mathscr{C}$, as follows. We create a variable for each independent feature A of the PCP-net, with domain equal to the set of all possible total orderings over the domain of A. The probability distribution over the orderings is given by the PCP-table of A. For each dependent feature B of the PCP-net, we add as many variables to the G-net as there are combinations of value assignments to the parents. Each of these variables B_1 to B_n will have the same domain: the set of total orderings over the domain of B.

Consider the PCP-net with two features, $\mathscr{C}$, from Example 1 whose corresponding G-net is shown below. The variables have domains $\mathscr{D}_A = \{a_1 > a_2, a_2 > a_1\}$, $\mathscr{D}_{B_{a_1}} = \{b_1 > b_2, b_2 > b_1\}$, and $\mathscr{D}_{B_{a_2}} = \{b_1 > b_2, b_2 > b_1\}$.



Variable A:

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

Variable B_{a_1}:

B_{a_1}	$\mathbb{P}$
$b_1 > b_2$	q_1
$b_2 > b_1$	$1-q_1$

Variable B_{a_2}:

B_{a_2}	$\mathbb{P}$
$b_1 > b_2$	q_2
$b_2 > b_1$	$1-q_2$

Theorem 1. *Given a PCP-net $\mathscr{C}$ and the corresponding G-net N, there is a one-to-one correspondence between the assignments of N and the induced CP-nets of $\mathscr{C}$.*

Theorem 2. *Given a PCP-net $\mathscr{C}$, the probability of realizing one of its induced CP-nets $\mathscr{C}_i$, is the joint probability of the corresponding assignment in the G-net for $\mathscr{C}$.*

Proof. There is a one-to-one correspondence between rows in the PCP-tables and nodes in the G-net. Additionally, choosing a particular ordering in a PCP-net row corresponds to an assignment to a variable in the G-net. $\square$

Theorem 3. *The probabilities over the induced CP-nets of a certain PCP-net form a probability distribution.*

Proof. The probability defined in Theorem 2 is computed as a product of non-negative factors, thus it is non-negative. Moreover, the sum of the probabilities of all the CP-nets in the set of the induced CP-nets is equal to 1, because there's a 1-1 correspondence between the assignments of the G-net with positive probability and the induced CP-nets, and the sum of the probabilities of all the assignments of a BN is equal to 1. $\square$

Theorem 4. *Given a PCP-net $\mathscr{C}$ and its induced CP-nets, the most probable of the induced CP-nets is the variable assignment with maximal joint probability in the G-net for $\mathscr{C}$.*

4.2 The Most Probable Optimal Outcome

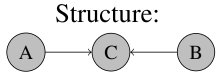
The most probable optimal outcome is the outcome that occurs with the greatest probability as the optimal in the set of induced CP-nets. The probability that an outcome o is optimal corresponds to the sum of the probabilities of the CP-nets that have o as the optimal outcome. Observe that the most probable optimal outcome may not be the optimal outcome of the most probable CP-net. Consider a PCP-net with only one feature A with domain $\mathscr{D}_A = \{a_1, a_2, a_3\}$ and let $a_1 > a_2 > a_3 = 0.3$, $a_1 > a_3 > a_2 = 0.3$, and $a_3 > a_2 > a_1 = 0.4$. The most probable CP-net is the one corresponding to the third ordering and it has the optimal outcome a_3. The other CP-nets have a_1 as optimal, so $\mathbb{P}(a_1) = 0.6$ and $\mathbb{P}(a_3) = 0.4$. The most probable optimal outcome is therefore a_1 but the optimal outcome of the most probable CP-net is a_3.

To find the most probable optimal outcome, we cannot find the most probable induced CP-net by the G-net procedure described above and then find its optimal outcome; we must make use of another BN which we call the *optimal network*.

Given a PCP-net $\mathscr{C}$, the *optimal network (Opt-net)* for $\mathscr{C}$ is a BN with the same dependencies graph as $\mathscr{C}$. Thus, the Opt-net has a variable for each of the PCP-net's features. The domains of the variables in the Opt-net are the values of the corresponding features that are ranked first in at least one ordering with non-zero probability. The conditional probability tables of the Opt-net are obtained from the corresponding PCP-tables as follows: for each assignment of the parent variables, we consider the corresponding probability distribution over the values of the dependent variable defined in the PCP-table. The probability of a value for the dependent variable is the sum of the probabilities of the orderings that have that particular value as most preferred according to that distribution. Notice that our construction applies even when there are cyclic dependences in the corresponding PCP-net.

Example 2. Consider the PCP-net $\mathscr{C}$ with three features A, B and C with domains $\mathscr{D}_A = \{a_1, a_2\}$, $\mathscr{D}_B = \{b_1, b_2\}$ and $\mathscr{D}_C = \{c_1, c_2, c_3\}$.The Opt-net has the same dependency graph as $\mathscr{C}$, with three variables A, B and C with domains: $\mathscr{D}_A = \{a_1, a_2\}$, $\mathscr{D}_B = \{b_1, b_2\}$ and $\mathscr{D}_C = \{c_1, c_2\}$, and two edges AC and BC. The domain of variable C in the Opt-net does not contain value c_3 because it never appears as most preferred in any ordering. Therefore, the Opt-net has a table for entry $a_1 b_2$ where c_1 appears with probability 0.2 and c_2 appears with probability 0.8.

Structure:



Variable A:

ordering for A	$\mathbb{P}$
$a_1 > a_2$	0.8
$a_2 > a_1$	0.2

Variable B:

ordering for B	$\mathbb{P}$
$b_1 > b_2$	0.7
$b_2 > b_1$	0.3

Variable C:

A & B	ordering for C	$\mathbb{P}$
$a_1\ b_1$	$c_1 > c_2 > c_3$	0.3
	$c_2 > c_1 > c_3$	0.7
$a_1\ b_2$	$c_1 > c_2 > c_3$	0.2
	$c_2 > c_1 > c_3$	0.4
	$c_2 > c_3 > c_1$	0.4
$a_2\ b_1$	$c_1 > c_3 > c_2$	0.4
	$c_2 > c_3 > c_1$	0.6
$a_2\ b_2$	$c_1 > c_2 > c_3$	0.1
	$c_2 > c_1 > c_3$	0.9

Theorem 5. *Given a PCP-net $\mathscr{C}$ and its Opt-net, 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 $\mathscr{C}$.*

Theorem 6. *Given a PCP-net $\mathscr{C}$, the probability that an outcome is optimal is the joint probability of the corresponding assignment in the optimal network. If no such corresponding assignment exists, then the probability of being optimal is 0.*

Proof. By construction, the set of assignments of the Opt-net of $\mathscr{C}$ is a subset of those of $\mathscr{C}$. By the definition of the Opt-net, if an assignment of $\mathscr{C}$ is not an assignment of the Opt-net, then it cannot be optimal in any induced CP-net.

Let us now focus on the assignments of $\mathscr{C}$ that have a corresponding assignment in the Opt-net. Let $x = (x_1, x_2, ..., x_n)$ be one of these assignments. We denote by $P_{opt}(x)$ the joint probability of x, $\mathbb{P}(X_1 = x_1, ..., X_n = x_n)$ in the Opt-net. We recall that the probability that x is optimal in the PCP-net is the sum of the probabilities of the induced CP-nets that have assignment x as optimal. We call this probability $P_{cp}(x)$. We must prove that $P_{opt}(x) = P_{cp}(x)$.

Let us consider A_x, the set of induced CP-nets that have x as their optimal assignment; giving $P_{cp}(x) = \sum_{\mathscr{C} \in A_x} \mathbb{P}(\mathscr{C})$. When we compute the optimal value for a CP-net, we sweep forward, starting from the independent features, assigning features their most preferred value. This means that only one subset of the rows of the CP-tables is considered when computing the optimal outcome. We can thus split a CP-net $\mathscr{C}$ into two parts, one affecting the choice of the optimal outcome (denoted with $\mathscr{C}_*$) and one not involved in it (denoted with $\mathscr{C}_{-*}$). If we consider the probability that that CP-net is induced by the PCP-net, we see that these two parts are independent. Thus we have $P_{cp}(x) = \sum_{\mathscr{C} \in A_x} \mathbb{P}(\mathscr{C}) = \sum_{\mathscr{C} \in A_x} \mathbb{P}(\mathscr{C}_*)\mathbb{P}(\mathscr{C}_{-*})$.

Regarding $\mathscr{C}_*$, observe that the optimal outcome x can be produced in many different ways, as there can be many different orderings that produce the same result. For example the orderings $a_1 > a_2 > a_3$ and $a_1 > a_3 > a_4$ produce both the optimal value a_1 for variable X_1. So we can do a disjoint partition of the set A_x into k subsets $A_{x_1}, ..., A_{x_k}$ for some k.

Two CP-nets $\mathscr{C}$ and $\mathscr{D}$ that belong to the same A_{x_i} are equal in the part that actively affects the choice of the optimal value and different in the other parts: $\mathscr{C}_* = \mathscr{D}_*$ and $\mathscr{C}_{-*} \neq \mathscr{D}_{-*}$.

Let $\mathscr{C}_*^i$ be the part that is equal for all the members of A_{x_i}. The probability becomes: $P_{cp}(x) = \sum_{i=1}^k \mathbb{P}(\mathscr{C}_*^i) \sum_{\mathscr{C} \in A_{x_i}} \mathbb{P}(\mathscr{C}_{-*})$. We note that $\sum_{\mathscr{C} \in A_{x_i}} \mathbb{P}(\mathscr{C}_{-*}) = 1 \ \forall i = 1, ..., k$, since we are summing the probability of all possible cases regarding $\mathscr{C}_{-*}$. Thus the probability becomes $P_{cp}(x) = \sum_{i=1}^k \mathbb{P}(\mathscr{C}_*^i)$. However, we have $\mathbb{P}(X_1 = x_1, ..., X_n = x_n) = \sum_{i=1}^k \mathbb{P}(\mathscr{C}_*^i)$ and, thus, $P_{cp}(x) = P_{opt}(x)$. since we built the rows of the probability tables for the variables $X_1, ..., X_n$ by summing the probability of the orderings that have the same head. This is the same as summing the probabilities over the subset A_{x_i}. $\square$

Theorem 7. *To find the most probable optimal outcome for a PCP-net $\mathscr{C}$, it is sufficient to compute the assignment with the maximal joint probability of its optimal network.*

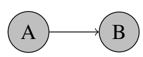
5 PCP-nets and Induced CP-nets

A PCP-net defines a probability distribution over a set of induced CP-nets. However, this step is not always reversible: below we show that, 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. Therefore, if there is a PCP-net which induces a set of CP-nets, we can find it quickly. This observation may be an interesting starting point for future work. We may be able to use CP-nets elicited from individuals to generate a PCP-net with which to "hot start" and create highly probable configurations for a recommendation system that is responsible for suggesting configurations for products to new customers [7, 16].

Theorem 8. *Given a probability distribution over a set of CP-nets (even if they have the same dependency graph), there may exist no PCP-net inducing it.*

Proof. Consider the following four CP-nets ($\mathscr{C}_1$, $\mathscr{C}_2$, $\mathscr{C}_3$ and $\mathscr{C}_4$) defined on the same variables: A and B. The two features have domains $\mathscr{D}_A = \{a_1, a_2\}$ and $\mathscr{D}_B = \{b_1, b_2\}$. The probability distribution on the four CP-nets is defined as follows:

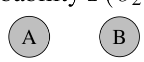
- $\mathscr{C}_1$ has probability $\mathbb{P}(\mathscr{C}_1) = 0.3$ and CP-tables:



ordering for A
$a_1 > a_2$

A	ordering for B
a_1	$b_1 > b_2$
a_2	$b_2 > b_1$

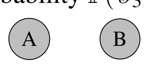
- $\mathscr{C}_2$ has probability $\mathbb{P}(\mathscr{C}_2) = 0.2$ and CP-tables:



ordering for A
$a_1 > a_2$

ordering for B
$b_1 > b_2$

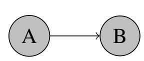
- $\mathscr{C}_3$ has probability $\mathbb{P}(\mathscr{C}_3) = 0.1$ and CP-tables:



ordering for A
$a_1 > a_2$

ordering for B
$b_2 > b_1$

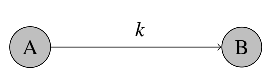
- $\mathscr{C}_4$ has probability $\mathbb{P}(\mathscr{C}_4) = 0.4$ and CP-tables:



ordering for A
$a_1 > a_2$

A	ordering for B
a_1	$b_2 > b_1$
a_2	$b_1 > b_2$

If $\mathscr{C}_1, \mathscr{C}_2, \mathscr{C}_3$ and $\mathscr{C}_4$ were all the induced CP-nets of a PCP-net, this PCP-net would have the dependency graph (on the features A and B with the relationship having probability k of occurring):



and the following PCP-tables:

ordering for A
$a_1 > a_2$

A	ordering for B	probability
a_1	$b_1 > b_2$	p_1
	$b_2 > b_1$	p_2
a_2	$b_1 > b_2$	p_3
	$b_2 > b_1$	p_4

where the values p_1, p_2, p_3 and p_4 need to be solutions of the following system of equations:

$$\begin{cases} p_1 p_3 = 0.2 \\ p_2 p_3 = 0.4 \\ p_1 p_4 = 0.3 \\ p_2 p_4 = 0.1 \end{cases} \qquad \begin{cases} 0 \leq p_1 \leq 1 \\ 0 \leq p_2 \leq 1 \\ 0 \leq p_3 \leq 1 \\ 0 \leq p_4 \leq 1 \end{cases} \qquad \begin{cases} p_1 + p_2 = 1 \\ p_3 + p_4 = 1 \end{cases}$$

However, such a system has no solution. □

Theorem 9. *Given a probability distribution over a set of CP-nets, we can compute a PCP-net to fit this distribution, if it exists.*

6 Updating Probabilistic CP-nets

We now turn our attention to modifications to the structure of a PCP-net. 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. Modifying the structure of the PCP-net is similar to entering evidence in a BN framework. By changing an arc or setting an ordering for a variable we can fix parts of the probability distribution and compute the outcomes of the resulting structure.

To add or delete a dependency or feature we just update the respective probability tables. This may involve deleting redundancy when we delete a feature. 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 $\mathscr{C}$ we also need to modify the probability tables in the associated G-net. This can change the most probable induced CP-net and therefore we need to recompute the outcome of the G-net.

To add or delete a dependency or feature, independent or dependent, we need to add or delete (or both) a number of nodes in the G-net which is exponential in the maximum number of parents which we assume to be bounded. The same can be said with respect to updating a probability table with either evidence or changing the distribution.

When we modify a PCP-net $\mathscr{C}$, the changes affect its Opt-net. Consider the dependency of feature B on feature A. When we add or delete this dependency, or when we change its probability, we only need to recompute the probability table of B in the Opt-net. When computing the most probable optimal outcome, we note that, in the worst case, we must recompute the whole maximal joint probability of the Opt-net. The same can be said when we delete a feature, as this amounts to the deletion of a set of dependencies, or when we modify the probability distribution over the orderings on B for a specific assignment to all of its parents. When we add a feature A to $\mathscr{C}$, we must add the corresponding node in the Opt-net and generate the corresponding probability table. This new node is independent. Thus, revising the current most probable optimal outcome is easy: the new optimal is the current one extended with the optimal value of the new feature.

7 Conclusions and Future Work

We have defined and shown how to reason with a generalized version of CP-nets, called PCP-nets, which can model probabilistic uncertainty and be updated without recomputing their entire structure. We have studied how to reason with these new structures in terms of optimality. PCP-nets can be seen as a way to bring together BNs and CP-nets, thus allowing to model preference and probability information in one unified structure.

We plan to study dominance queries and optimality tests in PCP-nets, as well as to study appropriate eliciting methods for both preferences and probabilities. Bigot et al. [1] have begun this line of inquiry on their model and show that, for PCP-nets that have a tree structure, dominance testing is tractable. We would also like to further explore, as Bigot et al., how our results related to the notion of local Condorcet winners in CP-net aggregation [21] as well as other issues in CP-net aggregation such as bribery [13, 14] and joint decision making [11]. Additionally, we have made several assumptions to bound the reasoning complexity of PCP-nets; we would like to relax these bounds or obtain results about approximability when these assumptions are lifted. We also plan to consider the use of PCP-nets in a multi-agent setting, where classical CP-nets have already been considered [18]. In this setting, PCP-nets can be used to represent probabilistic information on the preferences of a population.

Acknowledgments

This work is supported by the National Science Foundation, under grants CCF-1049360 and IIS-1107011. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

NICTA is funded by the Australian Government through the Department of Broadband, Communications and the Digital Economy and the Australian Research Council (ARC) through the ICT Centre of Excellence program. This research is also funded by AOARD grant 124056.

References

1. Bigot, D., Fargier, H., Mengin, J., Zanuttini, B.: Probabilistic conditional preference networks. In: Proc. 29th Conf. on Uncertainty in Artificial Intelligence (UAI) (2013)
2. Boutilier, C., Brafman, R., Domshlak, C., Hoos, H., Poole, D.: CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. Journal of Artificial Intelligence Research 21, 135–191 (2004)
3. Boutilier, C.: A POMDP formulation of preference elicitation problems. In: Proc. 18th AAAI Conference on Artificial Intelligence. pp. 239–246 (2002)

4. D'Ambrosio, B.: Inference in Bayesian Networks. AI Magazine 20(2), 21 (1999)
5. Dechter, R.: Bucket elimination: A unifying framework for reasoning. Artificial Intelligence 113(1-2), 41–85 (1999)
6. Domshlak, C., Brafman, R.: CP-nets: Reasoning and consistency testing. In: Proc. 8th Intl. Conf. on Principles and Knowledge Representation and Reasoning (KRR) (2002)
7. Faltings, B., Torrens, M., Pu, P.: Solution generation with qualitative models of preferences. Computational Intelligence 20(2), 246–263 (2004)
8. Fürnkranz, J., Hüllermeier, E.: Preference Learning: An Introduction. Springer (2010)
9. Goldsmith, J., Junker, U.: Preference handling for artificial intelligence. AI Magazine 29(4) (2009)
10. Goldsmith, J., Lang, J., Truszczynski, M., Wilson, N.: The computational complexity of dominance and consistency in CP-nets. Journal of Artificial Intelligence Research 33(1), 403–432 (2008)
11. Maran, A., Maudet, N., Pini, M.S., Rossi, F., Venable, K.B.: A framework for aggregating influenced CP-nets and its resistance to bribery. In: Proc. 27th AAAI Conference on Artificial Intelligence (2013)
12. Marden, J.I.: Analyzing and Modeling Rank Data. CRC Press (1995)
13. Mattei, N., Pini, M.S., Rossi, F., Venable, K.B.: Bribery in voting over combinatorial domains is easy. In: Proc. 11th Intl. Joint Conf. on Autonomous Agents and Multiagent Systems (AAMAS) (2012)
14. Mattei, N., Pini, M.S., Rossi, F., Venable, K.B.: Bribery in voting with CP-nets. Annals of Mathematics and Artificial Intelligence (2013)
15. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann (1988)
16. Price, R., Messinger, P.R.: Optimal recommendation sets: Covering uncertainty over user preferences. In: Proc. 20th AAAI Conference on Artificial Intelligence. pp. 541–548 (2005)
17. Regenwetter, M., Dana, J., Davis-Stober, C.P.: Transitivity of preferences. Psychological Review 118(1) (2011)
18. Rossi, F., Venable, K., Walsh, T.: mCP nets: representing and reasoning with preferences of multiple agents. In: Proc. 19th. AAAI Conference on Artificial Intelligence. pp. 729–734 (2004)
19. Roth, A.E., Kagel, J.H.: The handbook of experimental economics, vol. 1. Princeton University Press Princeton (1995)
20. Tversky, A., Kahneman, D.: Judgement under uncertainty: Heuristics and biases. Science 185, 1124–1131 (1974)
21. Xia, L., Conitzer, V., Lang, J.: Voting on multiattribute domains with cyclic preferential dependencies. In: Proc. 23rd AAAI Conference on Artificial Intelligence. pp. 202–207 (2008)