A Notion of Distance Between CP-nets

Andrea Loreggia
University of Padova
andrea.loreggia@gmail.com

Nicholas Mattei
IBM Research AI
n.mattei@ibm.com

Francesca Rossi
IBM Research AI
University of Padova
francesca.rossi2@ibm.com

K. Brent Venable
Tulane University
kvenabl@tulane.edu

Abstract

In many scenarios including multi-agent systems and recommender systems, user preference play a key role in driving the decisions the system makes. Thus it is important to have preference modeling frameworks that allow for expressive and compact representations, effective elicitation techniques, and efficient reasoning and aggregation. CP-nets offer convenient tradeoffs among all these desiderata. It is often useful to be able to measure the distance between the preferences of two individuals; between a group and an individual; between the preferences of the same individual at different times; or between subjective preferences and exogenous priorities, e.g., ethical principles, feasibility constraints, or business values. To this end we define a notion of distance between CP-nets that is tractable to compute and has useful theoretical and experimental properties when compared to the Kendall-tau distance between the partial orders generated by the CP-nets.

Introduction

Preferences are ubiquitous in real-life, and are central to decision making, whether the decision is made by a single individual or by a group. The study of preferences in computer science has been central to AI for a number of years with important theoretical and practical results (Domshlak et al. 2011; Pigozzi, Tsoukiàs, and Viappiani 2015). CP-nets provide an effective compact way to qualitatively model preferences over outcomes with a combinatorial structure (Boutilier et al. 2004). CP-nets are also easy to elicit and provide efficient optimisation reasoning. Moreover, CP-nets provide a way to model not only subjective preferences, but also priorities and optimisation criteria, thus allowing for a homogeneous modelling and reasoning framework where a seamless integration of several optimisation and preference reasoning modalities are supported.

Besides modelling, learning, reasoning with, and aggregating preferences, often it is useful to be able to measure the distance between the preferences of two individuals, or between a group and an individual, in order to measure the amount of disagreement and possibly get closer to a consensus. A notion of distance can also be useful in the presence of exogenous priorities, besides subjective preferences. Such priorities can be derived from ethical principles, feasibility constraints, or business values. Being able to measure the distance between preferences and external priorities provides a way to alert about deviations from feasibility or ethical constraints, and possibly suggest more compliant decisions.

In this paper we define a notion of distance (formally a distance function or metric) between CP-nets. CP-nets are a compact representation of a partial order over outcomes, so the ideal notion of distance would be a distance between the underlying partial orders of the CP-net. However, the size of the induced orders is exponential to the size of the CP-net, and we conjecture that computing a distance between such partial order is computationally intractable because of this possibly exponential expansion.

As a baseline for comparison we generalize the classic Kendall's τ (KT) distance (Kendall 1938), which counts the number of inverted pairs between two complete, strict linear orders. We add a penalty parameter p defined for partial rankings as in (Fagin et al. 2006), and use this distance we call KTD to compare partial orders. In this measure the contribution of pairs of outcomes that are ordered in opposite ways is 1 and that of those that are ordered in one partial order and incomparable in the other one is p. We show that we must have $0.5 \leq p < 1$ to be a distance.

To achieve tractability, we define a normalised distance between CP-nets, called CPD, that does not look at the partial orders but rather analyses the dependency structure of the CP-nets and their CP-tables in order to compute the distance. This notion of distance is an approximation of the KTD distance between partial orders. However, when KTD is 0 then CPD is also 0. This happens when the two CP-nets have the same dependency structure and CP-tables. The pairs of outcomes for which CPD could give an incorrect contribution to the distance are those that are either incomparable in both CP-nets, in this case CPD could generate an error of $+p$ or $-p$, or that are incomparable in a CP-net and ordered in the other, in this case the CPD error can be +1.

In order to give upper and lower bounds to the error that CPD can make, we study the incomparabilities present in a CP-net, proving that it is polynomial to compute the number of incomparable pairs of outcomes in a separable CP-net. Non-separable CP-nets have fewer incomparable pairs of outcomes, since each dependency link eliminates at least one incomparable pair. However, these theoretical bounds are very loose. For this reason, we also perform an experimental analysis of the relationship between CPD and KTD, which shows that the average error is never more than 10%.

Contribution. We define a distance function between CP-nets that generalizes the Kendall τ distance between the underlying partial orders. We conjecture that this distance is hard to compute and define an approximation of this distance which can be computed in polynomial time. We provide bounds on this approximation based on the number of incomparable pairs in a CP-net and perform empirical experiments to show that our approximation is never more than 10% away from the true distance.

Background: CP-nets

CP-nets (Boutilier et al. 2004) (for Conditional Preference networks) are a graphical model for compactly representing conditional and qualitative preference relations. They are sets of *ceteris paribus* preference statements (cp-statements). For instance, the cp-statement *"I prefer red wine to white wine if meat is served."* asserts that, given two meals that differ *only* in the kind of wine served *and* both containing meat, the meal with red wine is preferable to the meal with white wine. Formally, a CP-net has a set of features $F = \{x_1, \ldots, x_n\}$ with finite domains $\mathcal{D}(x_1), \ldots, \mathcal{D}(x_n)$. For each feature x_i, we are given a set of *parent* features $Pa(x_i)$ that can affect the 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. An *acyclic* CP-net is one in which the dependency graph is acyclic. Given this structural information, one needs to specify the preference over the values of each variable x for *each complete assignment* on $Pa(x)$. This preference is assumed to take the form of a total or partial order over $\mathcal{D}(x)$. A cp-statement has the general form $x_1 = v_1, \ldots, x_n = v_n : x = a_1 \succ \ldots \succ x = a_m$, where $Pa(x) = \{x_1, \ldots, x_n\}$, $D(x) = \{a_1, \ldots, a_m\}$, and $\succ$ is a total order over such a domain. The set of cp-statements regarding a certain variable X is called the cp-table for X.

Consider a CP-net whose features are A, B, C, and D, with binary domains containing f and $\overline{f}$ if F is the name of the feature, and with the cp-statements as follows: $a \succ \overline{a}$, $b \succ \overline{b}$, $(a \wedge b) : c \succ \overline{c}$, $(\overline{a} \wedge \overline{b}) : c \succ \overline{c}$, $(a \wedge \overline{b}) : \overline{c} \succ c$, $(\overline{a} \wedge b) : \overline{c} \succ c$, $c : d \succ \overline{d}$, $\overline{c} : \overline{d} \succ d$. Here, statement $a \succ \overline{a}$ represents the unconditional preference for $A = a$ over $A = \overline{a}$, while statement $c : d \succ \overline{d}$ states that $D = d$ is preferred to $D = \overline{d}$, given that $C = c$.

A *worsening flip* is a change in the value of a variable to a less preferred value according to the cp-statement for that variable. For example, in the CP-net above, passing from $abcd$ to $ab\overline{c}d$ is a worsening flip since c is better than $\overline{c}$ given a and b. One outcome α is *better* than another outcome β (written $\alpha \succ \beta$) if and only if there is a chain of worsening flips from α to β. This definition induces a preorder over the outcomes, which is a partial order if the CP-net is acyclic.

Finding the optimal outcome of a CP-net is NP-hard (Boutilier et al. 2004). However, in acyclic CP-nets, there is only one optimal outcome and this can be found in linear time by sweeping through the CP-net, assigning the most preferred values in the cp-tables. For instance, in the CP-net above, we would choose $A = a$ and $B = b$, then $C = c$, and then $D = d$. In the general case, the optimal outcomes coincide with the solutions of a set of constraints obtained replacing each cp-statement with a constraint (Brafman and Dimopoulos 2004): from the cp-statement $x_1 = v_1, \ldots, x_n = v_n : x = a_1 \succ \ldots \succ x = a_m$ we get the constraint $v_1, \ldots, v_n \Rightarrow a_1$. For example, the following cp-statement (of the example above) $(a \wedge b) : c \succ \overline{c}$ would be replaced by the constraint $(a \wedge b) \Rightarrow c$.

In this paper we want to compare CP-nets while leveraging the compactness of the representation. To do this, we consider profile (P, O), where *P* is a collection of n CP-nets (whose graph is a directed acyclic graph (DAG)) over m common variables with binary domains and *O* is a total order over these variables. We require that the profile is O-legal (Lang and Xia 2009), which means that in each CP-net, each variable is independent to all the others following in the ordering O. Given a variable X_i the function $flw(X_i)$ returns the number of variables following X_i in the ordering O.

Since every acyclic CP-net is satisfiable (Boutilier et al. 2004), we could compute a distance among two CP-nets by comparing a linearization of the partial orders induced by the two CP-nets. In this paper, we consider the linearization generated using the algorithm described in the proof of Theorem 1 of (Boutilier et al. 2004). This algorithm works as follows: Given an acyclic CP-net A over n variables and a ordering O to which the A is O-legal, then we know there is at least one variable with no parents. If more than one variable has no parents, then choose the one that comes first in the ordering O; let X be such a variable. Let $x_1 \succ x_2$ be the ordering over $Dom(X)$ dictated by the cp-table of X. For each $x_i \in Dom(X)$, construct a CP-net, N_i, with the $n-1$ variables $V - X$ by removing X from the initial CP-net, and for each variable Y that is a child of X, revising its CPT by restricting each row to $X = x_i$. We can construct a preference ordering $\succ_i$ for each of the reduced CP-nets N_i. For each N_i recursively identify the variable X_i with no parents and construct a CP-net for each value in $Dom(X_i)$ following the same algorithm until a CP-net have variables. We can now construct a preference ordering for the original network A by ranking every outcome with $X = x_i$ as preferred to any outcome with $X = x_j$ if $x_i \succ x_j$ in CPT(X). This linearization, which we denote with $LexO(A)$, assures that ordered pairs in the induced partial order are ordered the same in the linearization and that incomparable pairs are linearized using the cp-tables.

A CP-net Distance Function

In what follows we will assume that all CP-nets are acyclic and in minimal (non-degenerate) form, i.e., all arcs in the dependency graph have a real dependency expressed in the cp-statements, see the extended discussion in (Allen et al. 2016; 2017).

The following definition is an extension of the Kendall's τ (KT) distance (Kendall 1938) with a penalty parameter p defined for partial rankings in (Fagin et al. 2006).

Definition 1. *Given two CP-nets A and B inducing partial orders P and Q over the same set of outcomes U:*

$$KTD(A,B) = KT(P,Q) = \sum_{\forall i,j \in U, i \neq j} K^p_{i,j}(P,Q) \quad (1)$$

where i and j are two outcomes with $i \neq j$, we have:

1. *$K^p_{i,j}(P,Q) = 0$ if i,j are ordered in the same way or they are incomparable in both P and Q;*
2. *$K^p_{i,j}(P,Q) = 1$ if i,j are ordered inversely in P and Q;*
3. *$K^p_{i,j}(P,Q) = p$, $0.5 \leq p < 1$ if i,j are ordered in P (resp. Q) and incomparable in Q (resp. P).*

In the previous definition we choose $p \geq 0.5$ to make $KTD(A,B)$ a distance function, indeed if $p < 0.5$ the distance does not satisfy the triangle inequality. We also exclude $p = 1$ so that there is a penalty for two outcomes being considered incomparable in one and ordered in another CP-net. This allows us, assuming O-legality, to define for each CP-net a unique most distant CP-net.

Proposition 1. *Given two acyclic CP-nets A and B that are not O-legal, deciding if $KTD(A,B) = 0$ cannot be computed in polynomial time unless $P = NP$.*

The NP-complete problem of checking for equivalence for two arbitrary CP-nets (Santhanam, Basu, and Honavar 2013), i.e., deciding if two CP-nets induce the same ordering, can be reduced to the problem of checking if their KTD distance is 0. That is, if we had a polynomial time algorithm for deciding if $KTD(A,B) = 0$ then we could decide the equivalence problem for acyclic CP-nets. Hence, it seems reasonable to conjecture that the same result holds also for O-legal CP-nets.

Due to this likely intractability we will define another distance for CP-nets which can be computed efficiently directly from the CP-nets without having to explicitly compute the induced partial orders. This new distance is defined as the Kendal Tau distance of the two $LexO$ linearizations of the partial orders (see Section).

Definition 2. *Given two O-legal CP-nets A and B, with m features, we define:*

$$CPD(A,B) = KT(LexO(A), LexO(B)) \quad (2)$$

We show that CPD is a distance over O-legal CP-nets.

Theorem 1. *Function CPD(A,B) satisfies the following properties:*

1. *$CPD(A,B) \geq 0$;*
2. *$CPD(A,B) = CPD(B,A)$;*
3. *$CPD(A,B) \leq CPD(A,C) + CPD(C,B)$.*
4. *$CPD(A,B) = 0$ if and only if $A = B$;*

Proof. Properties 1-3 are directly derived from the fact that KTD is a distance function over total orders. Let us now focus on property 4. In our context, $A = B$ if and only if they induce the same partial order. It is, thus, obvious that if $A = B$ then $CPD(A,B) = 0$ since $LexO(A) = LexO(B)$. Let us now assume that $A \neq B$. Thus A and B induce different partial orders. In principle, what could happen is that one partial order is a subset of the other. In such a case they would have the same $LexO$ linearizations and it would be the case that $CPD(A,B) = 0$, despite them being different. We need to show that this cannot be the case if A and B are O-legal. Let us first assume that A and B have the same dependency graph but that they differ in at least one ordering in one CP-table. It is easy to see that in such a case there is at least one pair of outcomes that are ordered in the opposite way in the two induced partial orders. Assume that A and B have a different dependency graph. Due to O-legality it must be that there is a least an edge which is present, say, in A and missing B. In this case by adding a non-redundant dependency we are reversing the order of at least two outcomes. □

We will now show how $CPD(A,B)$ can be directly computed from CP-nets A and B, without having to compute the linearizations. The computation comprises of two steps. The first step, which we call, normalization, modifies A and B so that each feature will have the same set of parents in both CP-nets. This means that each feature will have in both normalized CP-nets a CP-table with exactly the same number of rows corresponding each to the same assignment to its parents. The second step, broadly speaking, computes the contribution to the distance of each difference in the CP-table entries. We describe each step in turn.

Step 1: Normalization. Consider two CP-nets, A and B over m variables $V = \{X_1, \ldots, X_m\}$ each with binary domains. We assume the two CP-nets are O-legal with respect to a total order $O = X_1 < X_2 < \cdots < X_{m-1} < X_m$. We note that O-legality implies that the X_i can only depend on a subset of $\{X_1, \ldots, X_{i-1}\}$

Each variable X_i has a set of parents $Pa_A(X_i)$ (resp. $Pa_B(X_i)$) in A (resp. in B), and is annotated with a conditional preference table in each CP-net, denoted $CPT_A(X_i)$ and $CPT_B(X_i)$.

We note that, in general we will have that $Pa_A(X_i) \neq Pa_B(X_i)$. However, it is easy to extend the two CP-nets so that in both X_i will have the same set of parents $Pa_A(X_i) \cup Pa_B(X_i)$. This is done by adding redundant information to the CP-tables, which does not alter the induced ordering.

For example, let us consider $CPT_A(X_i)$, then we will add $2^{Pa_A(X_i) \cup Pa_B(X_i)} - 2^{Pa_A(X_i)}$ copies of each original row to $CPT_A(X_i)$, that is, one for each assignment to the variables on which X_i depends in B but not in A. After this process is applied to all the features in both CP-nets, each feature will have the same parents in both CP-nets and its CP-tables will have the same number of rows in both CP-nets. We denote with A' and B' the resulting CP-nets.

We note that normalization can be seen as the reverse process of CP-net reduction (Apt, Rossi, and Venable 2008) which eliminates redundant dependencies in a CP-net.

Step 2: Distance calculation Given two normalized CP-nets A and B, let $diff(A,B)$ represent the set of CP-table entries of B which are different in A and let $var(i) = j$ if CP-table entry i refers to variable X_j. Moreover, let $m =$

$|V|$ and $flw(X)$ denote the number of features following X in order O. Let us define the two following quantities:

$$nSwap(A,B) = \sum_{j \in diff(A,B)} 2^{flw(var(j))+(m-1)-|Pa_B(var(j))|} \tag{3}$$

which counts the number of inversions that are caused by each different table entry and sums them up.

Theorem 2. *Given two normalized CP-nets A and B, we have:*

$$CPD(A,B) = nSwap(A,B) \tag{4}$$

We provide an example that gives an intuition of how a difference in a CP-table entry affects the $LexO$ linearization.

Example 0.1. *Consider a CP-net with three binary features, A, B, and C, with domains containing f and $\overline{f}$ if F is the name of the feature, and with the cp-statements as follows: $a \succ \overline{a}$, $b \succ \overline{b}$, $c \succ \overline{c}$. A linearization of the partial order induced by this CP-net can be obtained by imposing an order over the variables, say Let variable ordering $O = A \succ B \succ C$. The $LexO(A)$ is as follows:*

$$\overbrace{\overbrace{abc \succ ab\overline{c}}^{B1Zone} \succ \overbrace{a\overline{b}c \succ a\overline{b}\overline{c}}^{B2Zone}}^{A1Zone} \succ \overbrace{\overbrace{\overline{a}bc \succ \overline{a}b\overline{c}}^{B3zone} \succ \overbrace{\overline{a}\overline{b}c \succ \overline{a}\overline{b}\overline{c}}^{B4zone}}^{A2zone}$$

Now, consider changing only the cp-statement regarding A to $\overline{a} \succ a$. Then, the linearization of this new CP-net can be obtained by the previous one by swapping the first outcome in the $A1zone$ with the first outcome in the $A2zone$, the second outcome in the $A1zone$ with the second outcome in the $A2zone$ and so on. Moreover, the number of swaps is directly dependent on the number of variables that come after A in the total order.

From Theorem 2 we can see that $0 \leq CPD(A,B) \leq 2^{m-1}(2^m - 1)$, where m is the number of features. In particular:

- $CPD(A,B) = 0$ when the two CP-nets have the same dependency graph and cp-tables and so they are representing the same preferences;
- $CPD(A,B) = 2^{m-1}(2^m - 1)$ when the two CP-nets have the same dependency graph but cp-tables with reversed entries, so they are representing preferences that are opposite to each other.

Notice that variables with different cp-statements in the representation give more value to the distance if they come first in the total order: the value decreases as the position in the total order increases. For instance it is easy to prove that if the cp-statement of the first variable in the total order differs, than $CPD \geq 2^{m-2}(2^m - 1)$.

Bounding the error of CPD

The reason for introducing CPD is to provide a distance over CP-nets which can be computed directly from their structures and which approximates their KTD.

To understand to what extent CPD can differ from KTD, let us consider two O-legal CP-nets, A and B, with induced partial orders P and Q, and two outcomes o and o'. From Definitions 1 and 2 it follows that:

- If o and o' are ordered in both P and Q then the pair will contribute in the same way (either with 0, if they are ordered in the same way, or 1, if they are ordered in the opposite way) to both $KTD(A,B)$ and $CPD(A,B)$.
- If o and o' are incomparable in both P and Q then the contribution of the pair to $KTD(A,B)$ is 0, while its contribution to $CPD(A,B)$ can be either 0 or 1 depending if the $LexO$ linearization has linearized the pair in the same or opposite way in the two induced orderings.
- If o and o' are ordered in, say, P and incomparable in Q then the contribution of this pair to $KTD(A,B)$ is p while its contribution to $CPD(A,B)$ is either 0 or 1.

Summarizing, for each pair CPD can overestimate of at most 1 and under-estimate of at most p only if the pair is incomparable in at least one of the orderings. Thus, an absolute upperbound to the error that CPD makes can be estimated by counting the maximum number of incomparable pairs in an ordering induced by a CP-net. We will now compute this number.

Let us consider a separable CP-net S, that is, a CP-net over a set of m variables V with binary domains and no dependencies between the variables. Let P be the partial order induced by S over the set of outcomes U. A *chain* is a subset $U' \subseteq U$ such that for each $(x,y) \in U' \times U'$, $x > y$ or $x < y$. We recall that the *height* of a partial order P, denoted $h(P)$, is the number of elements in the longest chain. We call $incomp(P)$ the set of all the incomparable pairs of outcomes in P. In the following section we will call o_b the best outcome and o_w the worst outcome in P.

We start by observing that the height of a partial order induced by an acyclic CP-net corresponds to the length of the longest path from the best outcome to the worst outcome. This is a direct consequence of the fact that a CP-net induces a lattice.

Proposition 2. *The height $h(P)$ of a partial order P induced by an acyclic CP-net coincides with the length of the longest path from o_b to o_w.*

We now observe that the number of incomparable pairs in a partial order is connected to its height. In fact, Mirsky's theorem states that the height of a partial order equals the cardinality of the minimum antichains partition that cover the partial order (Mirsky 1971). This result can be extended to partial orders induced by CP-nets.

Theorem 3. *Given two partial orders P and Q induced by two O-legal acyclic CP-nets defined over the same set of variables V, if $h(P) > h(Q)$ then $incomp(P) < incomp(Q)$.*

Given Proposition 2 and Theorem 3 we can prove that separable CP-nets are indeed inducing partial orders with the maximal set of incomparables with respect to all other O-legal CP-nets.

Theorem 4. *A separable CP-net S induces a partial order P where the number of incomparable pairs of outcomes is maximal with respect of all the possible acyclic O-legal CP-nets over the same set of variables.*

We provide the intuition behind the proof. Let's start by computing the partial order induced by the CP-net, starting from the best outcome o_b. We can now build the next level of outcomes by changing just one assignment for each variable, let's call $var(o_{i,j})$ the subset of variables for which we change the value in the $j-th$ outcome of the $i-th$ level with respect to o_b. We get a subset of outcomes that differ for just one value from o_b. In the induced partial order all these outcomes are worst than o_b by definition. The cardinality of this subset is $\binom{n}{1}$. For each outcome we can now compute the subset of outcomes of the next level by changing the assignment of just one variable except the ones in $var(o_{i,j})$. Each outcome in level $i+1$ derived from $o_{i,j}$ is worst than it. For each level i the number of outcomes is $\binom{n}{i}$. In such a way there does not exist an outcome o' in level i' which is better than another outcome o'' in level i'', with $i' > i''$. Roughly speaking, we have shown that level partitioning is minimal and any other CP-net structure leads to an increment of height of the induced partial order. But due to Theorem 3, a completely separable CP-net has the maximal number of incomparable pairs with respect to other CP-net over the same features.

The reasoning above also allow us to compute such maximum number of incomparables.

Theorem 5. *Then the total number of incomparable pairs in any completely separable CP-net is:*

$$\sum_{i=1}^{m-1} \binom{m}{i} \cdot \left(\frac{1}{2}\left(\binom{m}{i} - 1\right) + \sum_{l=1}^{i} \binom{i}{l} \sum_{j=l+1}^{m-1} \binom{m-1}{j}\right) \quad (5)$$

In fact, starting from the best outcome we can flip the value of a single variable to have a new outcome which is directly comparable with the best outcome. This can be done $\binom{m}{i}$ times (with $i=1$) to have all the possible outcomes that are directly comparable with the best outcome in the induced partial order. All the outcomes computed in such a way are incomparable with respect to each other since the CP-net is separable and since each outcome differs on two values from any other outcome at the same level (let's denote a level with the value of index i). So the number of incomparable pairs is $\binom{m}{i} \cdot (\frac{1}{2}(\binom{m}{i} - 1)$. Iterating this computation by increasing the index i we have the number of incomparable pairs for each level. We now need to compute the number of incomparable pairs due to the transitive closure. At each level, the index i also represents the number of variables with a different values with respect to the best outcome, we call $F \subset V$ the subset of such variables. Let's consider two outcomes o at level i and o' at level j, with $j = i+1$. The two outcomes are incomparable if at least one variable in F and at least one more variables not in F have different values. The proof is straightforward, if all the variables in F for o and o' have the same values, then a single variable is changing from o to o' that makes the two outcomes comparable. We can choose $\binom{i}{l}$ different combinations for the variables in F and $\binom{m-1}{j}$ of different combinations for variables not in F. Iterating this process by increasing the index j allows us to compute the number of outcomes which are incomparable to o.

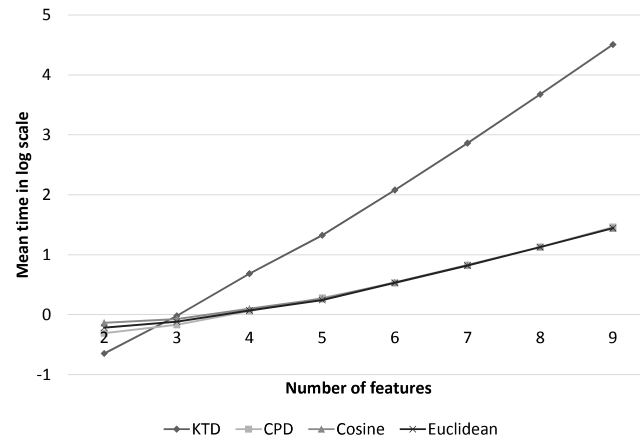

Figure 1: Mean time on log scale.

The number of incomparable pairs can be used to bound the error of CPD. It is easy to see that each non-fake arc added to the dependency graph of a CP-net reverses at least one arc in the induced partial order. This reduces of at least 1 the number of incomparable pairs, since we connect two outcomes of the same level in the induced partial order.

Empirical analysis

To support theoretical results, we run a set of experiments to have empirical evidence of the distance function's soundness. Firstly we generate CP-nets uniformly at random using the software described in (Allen et al. 2016). During this phase we generate CP-nets over n variables with binary domains (with $2 \leq n \leq 9$) and for each n we generate 1000 CP-nets using the default software settings, which means that for $2 \leq n \leq 6$ the maximum number of parents is $n-1$, while for $7 \leq n \leq 9$ the maximum number of parents for a feature is 5. We use these generated CP-nets to test different properties of the distance function. At the very beginning we tested the processing time for the KT distance function and we compared this processing time with CPD (we use $p = 0.5$ in the experiment). We also built a simple vector representation of a CP-net. In this vector representation, given two CP-nets A and B, where A is the referee, we normalized the two CP-nets to have the same number of cp-entries. Each cell of the vector representing A is equal to 1. Each i-th cell in the vector representing B is equal to 1 if the correspondent cp-entry in the two CP-nets are equal,

-1 otherwise. This vector representation is used to compute a cosine similarity and the euclidean distance, allowing us to compare CPDs performance with simple and well-known distance functions often used as metrics for similarity.

Fig. 1 shows the average time used to compute KTD, CPD, cosine similarity, and the euclidean distance. While KTD grows very fast, the other mean times are very similar one each other. This suggests that simple distance functions have good performance on this combinatorial domain. Clearly for small values of n the running time for CPD is higher than that of KTD because we also take into account normalization processing time. However, the mean time for KTD increases and it is almost 4 orders of magnitudes greater than the mean time of CPD, showing that computing an approximation of KTD is desirable also from a time point of view for values of $n \geq 3$.

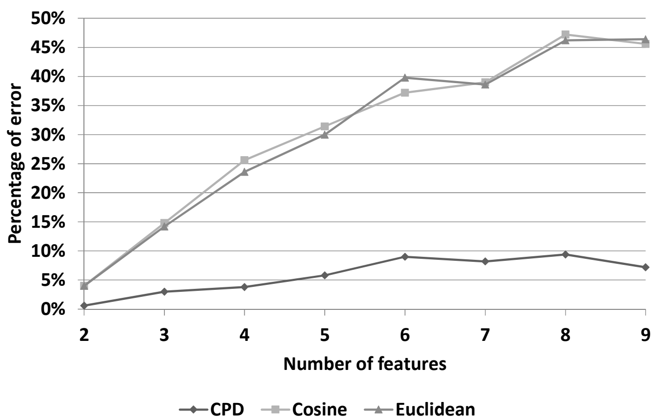

Figure 2: Percentage of errors.

During the first experiment we also checked how many times CPD, cosine similarity and the euclidean distance are incorrect. This means that given a triple of CP-nets (A, B, C) over the same set of variables we compute $kt1 = KTD(A, B)$, $kt2 = KTD(A, C)$, $l1 = CPD(A, B)$ and $l2 = CPD(A, C)$ (respectively the cosine similarity and the euclidean distance) and we count how many times $kt1 >= kt2$ but $l1 < l2$ or vice-versa. In other words we count how many times a CP-net B is closer to a CP-net A with respect to a third CP-net C according to KTD, but CPD, the cosine similarity and the euclidean distance state the opposite.

The number of errors made by the approximation metric, CPD, is shown up in Fig. 2. While the cosine similarity and the euclidean distance seem to behave badly when we increase the number of features, it is interesting to notice that CPD is stable and the total error is less than 9%. This is interesting because with a high value of n the number of incomparable pairs in the induced partial order increases, these incomparable pairs are the reason for the over-estimation and the under-estimation of the KDT value. Hence, CPD shows good resilience to incomparable pairs even when there are many such pairs.

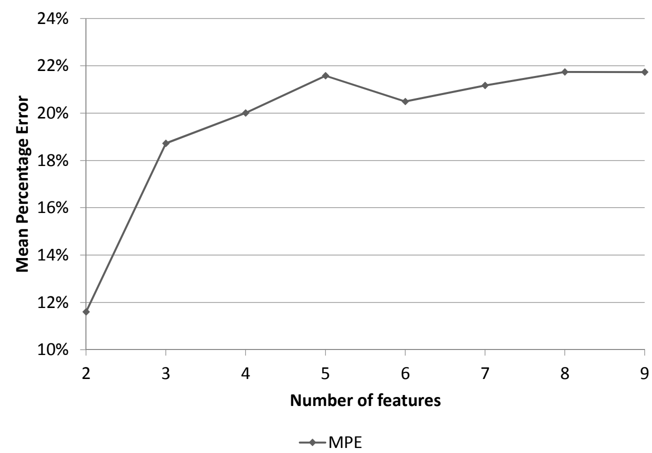

Figure 3: Mean percentage errors.

We take trace of the deviation of CPD with respect to the real (KTD) distance between the two CP-nets. In particular we compute the Mean Percentage Error (MPE) which is reported in Fig. 3. The MPE is a scale invariant measure for the error and gives an idea of how much the approximation is far from the real value. In Figure 3 we see that after we have $n = 3$ features, the error stays relatively constant.

Conclusions

In this paper we proposed a first step towards a useful notion of distance between CP-nets. This is for the best of our knowledge the first attempt to compute efficiently a distance between CP-nets. We give a theoretical study of this new approach and an experimental evaluation of the distance, that show it is both accurate and efficient to compute. Many extensions to our setting can be considered for the future. First, the CP-nets over which we define the CPD distance have the same features with the same domains but can differ in their dependency structure and CP-table. However, it would be useful to also consider CP-nets with different features and domains. Moreover, we consider CP-nets which are O-legal, that is, there is a total order of the CP-nets features that is compatible with the dependency links of both CP-nets. Relaxing this assumption is an important step for future work.

References

Allen, T. E.; Goldsmith, J.; Justice, H. E.; Mattei, N.; and Raines, K. 2016. Generating CP-nets uniformly at random. In Schuurmans, D., and Wellman, M. P., eds., *AAAI*, 872–878. AAAI Press.

Allen, T. E.; Goldsmith, J.; Justice, H. E.; Mattei, N.; and Raines, K. 2017. Uniform random generation and dominance testing for cp-nets. *Journal of Artificial Intelligence Research* 59:771–813.

Apt, K. R.; Rossi, F.; and Venable, K. B. 2008. Comparing the notions of optimality in cp-nets, strategic games and soft constraints. *Ann. Math. Artif. Intell.* 52(1):25–54.

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.

Brafman, R. I., and Dimopoulos, Y. 2004. Extended semantics and optimization algorithms for CP-networks. *Computational Intelligence* 20(2):218–245.

Domshlak, C.; Hüllermeier, E.; Kaci, S.; and Prade, H. 2011. Preferences in AI: An overview. 175(7):1037–1052.

Fagin, R.; Kumar, R.; Mahdian, M.; Sivakumar, D.; and Vee, E. 2006. Comparing partial rankings. *SIAM J. Discret. Math.* 20(3):628–648.

Kendall, M. G. 1938. A new measure of rank correlation. *Biometrika* 30(1/2):81–93.

Lang, J., and Xia, L. 2009. Sequential composition of voting rules in multi-issue domains. *Mathematical Social Sciences* 57(3):304–324.

Mirsky, L. 1971. A dual of Dilworth's decomposition theorem. *The American Mathematical Monthly* 78(8):876–877.

Pigozzi, G.; Tsoukiàs, A.; and Viappiani, P. 2015. Preferences in artificial intelligence. 77:361–401.

Santhanam, G. R.; Basu, S.; and Honavar, V. 2013. Verifying preferential equivalence and subsumption via model checking. In Perny, P.; Pirlot, M.; and Tsoukiàs, A., eds., *Algorithmic Decision Theory - Third International Conference, ADT 2013, Bruxelles, Belgium, November 12-14, 2013, Proceedings*, volume 8176 of *Lecture Notes in Computer Science*, 324–335. Springer.