Learning Conditional Preference Networks: an Approach Based on the Minimum Description Length Principle

Anonymous authors

Abstract

CP-nets are a very expressive graphical model for representing preferences over combinatorial spaces. They are particularly well suited for settings where an important task is to compute the optimal completion of some partially specified alternative; this is, for instance, the case of interactive configurators, where preferences can be used at every step of the interaction to guide the decision maker towards a satisfactory configuration. Learning CP-nets is challenging when the input data has the form of pairwise comparisons between alternatives. Furthermore, this type of preference data is not commonly stored: it can be elicited but this puts an additional burden on the decision maker. In this article, we propose a new method for learning CP-nets from sales history, a kind of data readily available in many e-commerce applications. The approach is based on the minimum description length (MDL) principle. We show some theoretical properties of this learning task, namely its sample complexity and its NP-completeness, and we experiment with this learning algorithm in a recommendation setting with real sales history from a car maker.

1 Introduction

Online shopping services, like video-on-demand streaming services and product configurators for computers, cars, or kitchens, rely on recommendation and customization of the user experience to boost sales [Zhang, 2014]. Recommendations are essential in large, combinatorial product spaces, where the number of alternatives can lead to over-choice confusion [Huffman and Kahn, 1998]. In such a case, a user is overwhelmed by the possibilities and cannot choose. A common tool in configurators is optimal completion, where the configurator automatically completes a partially configured product by maximizing the product's utility for the user. Recommendation, and optimal completion, in particular, are typically based on a modeling of user preferences. However, except when the number of attributes is very small, it is intractable to represent a linear order over the space of all possible alternatives in extension. Several compact, graphical representations of preferences have been studied in the literature. Combinatorial preferences can be modeled with numerical models, such as GAI-nets [Gonzales and Perny, 2004] and ensemble ranking function [Freund et al., 2003], or by ordinal graphical models, such as lexicographic preferences trees (LP-trees) [Fraser, 1993] and conditional preferences networks (CP-nets) [Boutilier et al., 2004].

CP-nets, in particular, are well-suited for interactive configuration: they can represent (albeit partially) any preference relation, and the optimal completion query can be answered in polynomial time with acyclic CP-nets with the Forward Sweep algorithm [Boutilier et al., 2004].

Learning CP-nets has often been studied in settings where the input data has the form of pairwise comparisons between alternatives [Dimopoulos et al., 2009, Lang and Mengin, 2009, Chevaleyre et al., 2011, Allen et al., 2017, Alanazi et al., 2016, Labernia et al., 2017, Liu and Liu, 2019, Alanazi et al., 2020]. However, this type of preference data is not commonly stored. It can be elicited [Koriche and Zanuttini, 2010] but this puts an additional burden on the decision maker. In this article, we investigate a new method for learning CP-nets from sales history, a kind of data readily available in many e-commerce applications. The approach is based on the minimum description length (MDL) principle, which has been successfully applied in the unsupervised learning of many classes of models, such as Bayesian networks [Suzuki, 1993, Lam and Bacchus, 1994], causal networks [Mian et al., 2021] and formal grammars [Garofalakis et al., 2003].

More precisely, the contributions of this article are:

- the sample complexity of unsupervised CP-nets learning;
- the NP-completeness of unsupervised CP-nets learning;
- an unsupervised CP-nets learning algorithm;
- an experimental evaluation on a real-world recommendation task.

2 Related work

We review here some works on learning preferences from a list of observed optimal alternatives, not pairwise comparisons.

Bayesian networks [Pearl, 1985, 2009] are graphical probabilistic models that can represent any probability distribution. Because it associates a numerical value to every alternative (its probability), they can be used to order these alternatives. Syntactically, a Bayesian network is a directed acyclic graph where each node is associated to a conditional probability table, making them syntactically similar to CP-nets. Learning the probability tables when the structure is known amounts to some simple frequencies computation, but learning the optimal, graphical structure of a Bayesian network is an NP-hard problem. Numerous methods for learning Bayesian networks have been proposed (see, e.g., a recent survey by Kitson et al. [2023]). Learning methods typically use local *conditional independence* tests to decide the presence or absence of some arrows in the graph, and/or some global scoring function to explore the space of possible structures. In particular, the *Bayesian Information Criterion (BIC)* is at the core of approaches based on the MDL principle. Note that the computation of the optimal, most probable completion of a partial instantiation is an NP-complete task for Bayesian networks [Kwisthout, 2011]. Our experiments in a configuration settings, described in Section 7, show that the recommendation time is significantly lower when using a CP-net than with a Bayesian network.

Bigot et al. [2014], Khoshkangini et al. [2018] propose methods to transform a learnt Bayesian network into a CP-net: local probability tables are converted to local preference orders, with most probable value being preferred. However, the probabilistic dependencies in a Bayesian network are not directed, whereas preferential dependencies in a CP-net are directed: Bigot et al. [2014] propose to identify the correct direction of the dependencies in a CP-net in an active learning settings, using pairwise comparison queries, and [Khoshkangini et al., 2018] assume that the structure is partially known at the start.

Fargier et al. [2018] propose a greedy algorithm to learn preferences represented by lexicographic preference trees from sales history. The aim is to find a model that attributes low ranks to alternatives that appear often in the history. The preference tree is computed in a top-down fashion, based on statistics calculated from the sales history on the attributes and their values. They prove that the algorithm computes the optimal structure when restricted to linear structures (lexicographic preference lists). In Fargier et al. [2022], the authors prove that the sample complexity for this preference learning method is logarithmic in the number of attributes. However, experiments described by [Fargier et al., 2020] indicate that lexicographic preference models may be too restrictive to represent well preferences from a group of agents (even if they are known to be a good model to represent preferences of one agent). Besides, their approach is based on the rank of an alternative and, therefore, cannot be applied to learn languages that represent partial orders, such as CP-nets.

3 Background

Combinatorial Domain We consider a combinatorial domain over a finite set $\mathcal{X}$ of discrete attributes that characterise the possible alternatives. Each attribute $X \in \mathcal{X}$ has a finite set of possible values $\underline{X}$. $\underline{\mathcal{X}}$ denotes the Cartesian product of the domains of the attributes in $\mathcal{X}$, its elements are called *alternatives*. We often use the symbols o, o', o_1, o_2, $\ldots$ to denote alternatives. In the following, n is the number of attributes in $\mathcal{X}$, and d is a bound on the size of the domains of the attributes: for every $X \in \mathcal{X}$, $2 \leqslant |\underline{X}| \leqslant d$.

For a subset U of $\mathcal{X}$, we will denote by $\underline{U}$ the Cartesian product of the domains of the attributes in U, every $u \in \underline{U}$ is an instantiation of U, or partial instantiation (of $\mathcal{X}$). If v is an instantiation of some $V \subseteq \mathcal{X}$, $v[U]$ denotes the restriction of v to the attributes in $V \cap U$.

Preference relations In this paper, we consider that a preference relations is a linear order over $\underline{\mathcal{X}}$, i.e., a total, transitive, irreflexive binary relations over $\underline{\mathcal{X}}$, often denoted with curly symbol $\succ$. For alternatives $o, o' \in \underline{\mathcal{X}}$, $o \succ o'$ indicates that o is strictly more preferred to o'. Given a partial instantiation u, consider the set of alternatives that extend u: this set is finite and the projection of linear order $\succ$ over this set is a linear order too, so it has a unique "most preferred" element. We denote this element by $\mathrm{opt}(u, \succ)$.

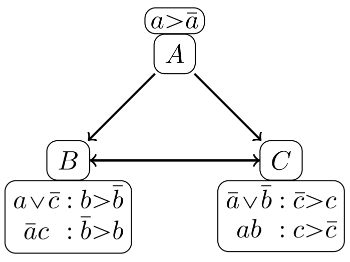
CP-nets Given the exponential size of $\underline{\mathcal{X}}$, it is not tractable to represent preference relations in extension. So, we use preferences models that rely on the assumption that the preference relations of interest exhibit some structure. We focus on one family of graphical models: *Conditional Preference Networks* (CP-nets). CP-nets have been introduced in [Boutilier et al., 2004] as a tool to make explicit a particular kind of structure, called preferential (in)dependence.

Figure 1a depicts a CP-net φ_0. More generally, a CP-net is a triple $\varphi = (\mathcal{X}, Pa, CPT)$, where:

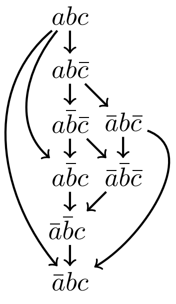
- Pa associates to every attribute $X \in \mathcal{X}$, a subset $Pa(X)$ of $\mathcal{X} \backslash \{X\}$. Thus Pa defines a directed graph over $\mathcal{X}$, where there is an edge (X, Y) if and only if $X \in Pa(Y)$. $Pa(Y)$ is the set of *parents* of Y. In this article, we only consider acyclic CP-nets.
- CPT is a set of *conditional preference tables*. One table $CPT(X)$ for every attribute X: $CPT(X)$ contains, for every instantiation u of $Pa(X)$, a rule $u :>$, where $>$ is a linear order over $\underline{X}$.

Let us call *swap* any pair of alternatives that have identical values for every attribute except one. A CP-net φ orders every swap $\{o, o'\}$ as follows: let X be the only attribute such that $o[X] \neq o'[X]$, let $u = o[Pa(X)] = o'[Pa(X)]$, let $u :>$ be the corresponding rule in $CPT(X)$, then (o, o') is a *worsening swap* (w.r.t. φ) if and only if $o[X] > o'[X]$. The transitive closure of all the worsening swaps sanctioned by φ is, by definition, transitive, and we denote it by $\succ_\varphi$. It is not necessarily irreflexive, and not complete in general. Acyclic CP-nets are guaranteed to be consistent, i.e., there always exists a total order that extends the order defined by a CP-net.

Example 1. *Figure 1a depicts a CP-net φ_0 over a combinatorial domain with three boolean attributes A, B and C, with respective domains $\{a, \bar{a}\}$, $\{b, \bar{b}\}$ and $\{c, \bar{c}\}$: $Pa(A) = \{\}$ and $CPT(A) = \{a > \bar{a}\}$, $Pa(B) = \{A, C\}$ and $CPT(B) = \{a \vee \bar{c} : b > \bar{b}, \bar{a}c : \bar{b} > b\}$. Figure 1b depicts $\succ_{\varphi_0}$: edges $o \longrightarrow o'$ represent the worsening swaps sanctioned by φ_0. Some of them are redundant since implied, by transitivity of $\succ_{\varphi_0}$: for instance the fact that $abc \succ_{\varphi_0} \bar{a}bc$ is implied by the worsening swaps $(abc, ab\bar{c})$, $(\bar{a}b\bar{c}, ab\bar{c})$, $(\bar{a}b\bar{c}, \bar{a}bc)$.*



(a) A CP-net φ_0.



(b) Swaps of φ_0.

Figure 1: A CP-net (left) and a representation of its order (right).

Boutilier et al. [2004] proved that, when a CP-net ϕ is acyclic, given a partial instantiation u, $\mathrm{opt}(u, \succ_\phi)$ exists and is unique, and it can be computed in time linear in the number of attributes with the so-called *Forward Sweep* procedure.

Although CP-nets cannot in general represent a total preference relation, they capture enough information to answer the opt query: more precisely, any total preference relation $\succ$ induces a CP-net $\phi_\succ$ such that $\mathrm{opt}(u, \succ) = \mathrm{opt}(u, \succ_{\phi_\succ})$ when $\phi_\succ$ is acyclic [Gimenez and Mengin, 2023].

Minimal Description Length In Machine Learning, the Minimal Description Length (MDL) principle is akin to Occam's razor principle, which states that the simplest hypothesis should be privileged. MDL enforces this idea through compression: given some data D and a class of possible models that may explain D, one should choose the model H that enables the lossless compression of D with minimum size [Grünwald, 2000]. Formally, if $L(D|H)$ denotes the length of the representation of D knowing H, one can define the minimum description length for D given a class of models $\mathcal{H}$ as $\min_{H\in\mathcal{H}} \big(L(H) + L(D|H)\big)$.

4 Applying the MDL principle to CP-nets learning

We now recall the learning problem that is the focus of this paper, and the MDL-based approach that is proposed.

Following Fargier et al. [2018], the input of the learning process is a sales history, i.e. a multiset $D \subseteq \underline{\mathcal{X}}$. Its elements are alternatives corresponding to products that may have been, for instance, configured by users of some configurator and bought. Each user has a preference relation $\check{\succ}$ among products, and is free to configure the products according to her preference. However, for some reasons like the influence of advertisements, special offers or unavailability, she may end up with a product which is not her most preferred one. Yet, the higher an outcome is ranked in the user's preference, the greater the probability that she ends up with it. Given this input, we want to compute a CP-net that represents a preference relations that explains this dataset D, and can be used to guide future users throughout the interaction with the configurator, towards an alternative that best suits their needs.

Gimenez and Mengin [2023] propose, given an acyclic CP-net ϕ, a lossless compression of such a dataset D as follows: for alternative o, $\mathrm{opt}(o, \phi)$ is the smallest partial instantiation u such that $\mathrm{opt}(u, \succ_\phi) = o$; they prove that this u is uniquely defined, and can be computed in polynomial time. Using this function opt to compress D given ϕ, we have:

$$L(D \mid \phi) = \sum_{o\in D} \Big[\, L_{\mathbb{N}}(|\,\mathrm{opt}(H,o)|) + \log_2 \tbinom{|\mathcal{X}|}{|\,\mathrm{opt}(o,H)|} + \sum_{X\in \mathrm{opt}(H,o)} \log_2(|\underline{X}| - 1) \,\Big] \quad (1)$$

where, for each outcome $o \in D$, the terms encode, in order: the length of the minimal code of o, the set of variables that are assigned in this code, and the value for each attribute; and where $L_{\mathbb{N}}$ is the length of the Rissanen universal integer encoding [Rissanen, 1983]. Moreover, the size of the representation of a given CP-net $\phi = (\mathcal{X}, Pa, CPT)$ is :

$$L(\phi) = L_{\mathbb{N}}(|\mathcal{X}|) + \sum_{N\in\mathcal{X}} L_{\mathbb{N}}(|Pa(N)|) + \log_2 \tbinom{|\mathcal{X}|-1}{|Pa(N)|} + |\underline{Pa(N)}| \log_2 |\underline{N}| \quad (2)$$

where, slightly abusing notation, $|\underline{N}|$ denotes the domain size of the attribute labelling N. These terms encode, in order: the total number of nodes, and for each node, the number of its parents, its set of parents and the optimal value for each value of its parents.

Therefore, the learning problem that we study in the remainder of the paper is the following one: given a set of attributes $\mathcal{X}$, and a sales history D, find the CP-net ϕ that minimizes $L(\phi) + L(D|\phi)$ as defined above. In the next section, we prove two complexity results about this problem. Section 6 proposes a simple local search algorithm to explore the set of CP-nets guided by this MDL-inspired cost function, and section 7 describe some experiments to compare this new approach to previous approaches to learning preferences in this settings.

5 Theoretical results

In this section, we propose a formalization of the MDL score that is more suited for theoretical analysis, and we present two new results on the CP-nets unsupervised learning: an upper bound on its sample complexity and the NP-completeness.

5.1 Normalized mean code length

Cost equations Eq. (2) and (1) are complex because, according to the MDL principle, the compressed data must contain all the information required for a lossless decompression, making these equations unsuited for theoretical analysis. We propose simplifying the alternative's cost by defining it to be equal to its *code length*. In the learning setting that assumes that the alternatives are drawn according to a probability distribution p, we propose to use the *normalized mean code length* (NMCL for short) as the loss of $\succ$, where n is the number of attributes:

$$NMCL_p(\succ) = \frac{1}{n} E_p[|code(\succ, \cdot)|] \quad (3)$$

This metric is between 0 and 1. Algorithm 4 proposed in [Gimenez and Mengin, 2023] shows that, for acyclic CP-nets:

$$|code(\succ, o)| = |\{N \mid o[N] \neq Pref(N, o[Pa(N)])\}|$$

where $Pref(N, o[Pa(N)])$ is the preferred value in the rule associated to $o[Pa(N)]$ in the CPT of N. Remark that we can further simplify the expression of $NMCL_p$ by introducing $p_{err}(N, v)$, the probability that a randomly drawn alternative is instantiated to v for the parents of N and does not include the most preferred value of N: $p_{err}(N, v) = p(v \wedge \neg Pref(N, v))$. Then:

$$\begin{aligned} NMCL_p(\succ) &= \frac{1}{n}\sum_N \sum_o p(o)[o[N] \neq Pref(N, o[Pa(N)])] \\ &= \frac{1}{n}\sum_N \sum_{v \in \underline{Pa(N)}} \sum_{k \in \underline{Var(N)}} [k \neq Pref(N, v)] p(vk) \\ &= \frac{1}{n}\sum_N \sum_{v \in \underline{Pa(N)}} p_{err}(N, v) \qquad (4) \end{aligned}$$

5.2 Upper bound on the sample complexity

Our analysis of the sample complexity is done within the PAC learning theory proposed by Valiant [Valiant, 1984], where we are interested in the number $N(\delta, \epsilon)$ such that, if $N(\delta, \epsilon)$ examples are available at training, then there is a high probability (bigger than $1 - \delta$) that the distance between the CP-net that minimizes the empirical normalized mean code length and the target unknown CP-net is small (lower than ϵ). In our case, this distance is the difference of their normalized mean code length. In this section, we show that the sample complexity is polynomial for fixed bounds on the size of the domains of the attributes and the number of parents.

Proposition 1. *For the family of CP-nets with n nodes and whose nodes have at most k parents, $N(\delta, \epsilon) = O\big(\frac{d^{2k}}{\epsilon^2}(\ln\frac{1}{\delta} + k(\ln d + \ln(n+1)))\big)$*

Proof. Denote CP-net^k the set of CP-nets whose nodes have at most k parents. Denote $\breve{\phi}$ a CP-net in CP-net^k that represents the preference relation $\breve{\succ}$. p is a probability distribution that is non-increasing w.r.t. $\breve{\succ}$, i.e. $o \breve{\succ} o' \Rightarrow p(o) \geqslant p(o')$. Let us denote ϕ^* the CP-net in CP-net^k that minimizes the empirical normalized mean code length with respect to some sample $\mathcal{S}$. Then:

$$\begin{aligned} loss(\phi^*, \breve{\phi}) &= NMCL_p(\phi^*) - NMCL_p(\breve{\phi}) \\ &= \frac{1}{n}\Big(E_p[|code(\phi^*, \cdot)|] - E_p[|code(\breve{\phi}, \cdot)|]\Big) \\ &\leqslant \frac{1}{n}\Big(\quad |E_p[|code(\phi^*, \cdot)|] - E_{p_\mathcal{S}}[|code(\phi^*, \cdot)|]| \\ &\qquad + \ E_{p_\mathcal{S}}[|code(\phi^*, \cdot)|] - E_{p_\mathcal{S}}[|code(\breve{\phi}, \cdot)|] \\ &\qquad + \ |E_{p_\mathcal{S}}[|code(\breve{\phi}, \cdot)|] - E_p[|code(\breve{\phi}, \cdot)|]|\Big) \\ &\leqslant \frac{2}{n} \max_{\phi \in \text{CP-net}^k} |E_p[|code(\phi, \cdot)|] - E_{p_\mathcal{S}}[|code(\phi, \cdot)|]| \end{aligned}$$

because $E_{p_\mathcal{S}}[|code(\phi^*, \cdot)|] - E_{p_\mathcal{S}}[|code(\breve{\phi}, \cdot)|] \leqslant 0$ by definition of ϕ^*. Now, for any $\phi \in$ CP-net^k :

$$\begin{aligned} &\frac{1}{n}|E_p[|code(\phi, \cdot)|] - E_{p_\mathcal{S}}[|code(\phi, \cdot)|]| \\ &\leqslant \frac{1}{n}\sum_N \sum_{v \in \underline{Pa(N)}} |p_{err}(N, v) - p_{\mathcal{S},err}(N, v)| \leqslant 2M \times d^k \end{aligned}$$

where d is a bound on the domain size of the attributes, and M is an upper bound on $|p_{err}(N, v) - p_{\mathcal{S},err}(N, v)|$ for every $N \in nodes(\phi)$, every $v \in \underline{Par(N)}$.

Thus, if $\frac{1}{n}|E_p[|code(\phi, \cdot)|] - E_{p_\mathcal{S}}[|code(\phi, \cdot)|]| \geqslant \epsilon$, there must be some $V \subseteq \mathcal{X}$ and $v \in \underline{V}$ such that $|p_{err}(N, v) - p_{\mathcal{S},err}(N, v)| \geqslant \epsilon/(2d^k)$, which implies that:

$$\begin{aligned} &Pr(loss(\phi^*, \breve{\phi}) \geqslant \epsilon) \\ &\leqslant Pr\Big(\bigcup_{\substack{V \subseteq \mathcal{X} \\ v \in \underline{V}}} |p_{err}(N, v) - p_{\mathcal{S},err}(N, v)| \geqslant \epsilon/(2d^k)\Big) \\ &\leqslant \sum_{\substack{V \subseteq \mathcal{X} \\ v \in \underline{V}}} Pr\left(|p_{err}(N, v) - p_{\mathcal{S},err}(N, v)| \geqslant \epsilon/(2d^k)\right) \end{aligned}$$

For every $V \subseteq \mathcal{X}$ and every $v \in \underline{V}$, $p_\mathcal{S}(v)$ is an estimate, from sample $\mathcal{S}$, of the ground probability $p(v)$ of drawing an alternative o such that $o[V] = v$. Hoeffding's inequality states that for every $\alpha > 0$:

$$Pr(|p_{err}(N, v) - p_{\mathcal{S},err}(N, v)| \geqslant \alpha) \leqslant 2e^{-2|\mathcal{S}|\alpha^2}$$

For every $i \in \{1, \ldots, k\}$, there are $\binom{n}{i}$ ways of choosing a subset V of $\mathcal{X}$ of cardinality i, then $|\underline{V}| \leqslant d^i$; therefore:

$$\begin{aligned} Pr(loss(\phi^*, \breve{\phi}) \geqslant \epsilon) &\leqslant \left(\sum_{i=1}^{k} \binom{n}{i} d^i\right) 2e^{-2|\mathcal{S}|\epsilon^2/(4d^{2k})} \\ &\leqslant 2d^k(1+n)^k e^{-|\mathcal{S}|\epsilon^2/(4d^{2k})} \end{aligned}$$

Therefore, in order to have $Pr(loss(\phi^*, \breve{\phi}) \leqslant \epsilon) \geqslant 1 - \delta$, it is sufficient to have $1 - 2d^k(1+n)^k \exp(-|\mathcal{S}|\epsilon^2/(4d^{2k})) \geqslant 1 - \delta$, which is equivalent to $|\mathcal{S}| \geqslant \frac{2d^{2k}}{\epsilon^2}(\ln\frac{1}{\delta} + k(\ln d + \ln(n+1)) - \ln 2)$. □

This low sample complexity shows that CP-nets, like other ordinal models, require a low number of examples to be learnt accurately. This is a very positive result, since CP-nets are still very expressive in the preferences they can express.

5.3 CP-net unsupervised learning is NP-complete

While CP-nets require few examples to be learned accurately, they rely on a graph, a data structure for which many problems are NP-complete. In fact, the following proposition shows that learning a CP-net that minimizes the empirical normalized mean code length is indeed NP-complete.

Proposition 2. *Let D be a dataset. The problem of finding the minimal acyclic CP-net that minimizes the empirical normalized mean code length over D is NP-complete.*

Proof sketch. The full proof is available in the appendix.

To show that the problem is NP-hard, we reduce the minimum feedback arc set (FAS) problem, known to be NP-complete [Karp, 2010], to it. Let $G = (V, E)$ be an instance of FAS, whose n nodes are denoted $V_1, V_2, \ldots, V_n$. The problem is to find a minimal set of edges E' such that $G = (V, E \backslash E')$ is acyclic. To that end, we construct a dataset such that the CP-net that minimizes the empirical normalized mean code length allows us to solve FAS easily. The combinatorial space of the dataset is defined over $|V| + |E|$ attributes: $\mathcal{X} = V \cup \{V_{i,j} \mid i \to j \in E\}$. Each attribute is binary and its domain is $\{\top, \bot\}$. We initialize the dataset D with $3|E| + 1$ alternatives that all have value $\bot$ for every attribute. This set of alternatives will ensure that for every attribute in $\mathcal{X}$, the most frequent value in D is $\bot$. Then, for each edge $V_i \to V_j$ in E, we add three alternatives to D: o_1 with $o_1[V_i] = o_1[V_{i,j}] = o_1[V_j] = \top$, and $o_1[X] = \bot$ for every other attribute; o_2 with $o_2[V_i] = o_2[V_j] = \top$, and $o_2[X] = \bot$ for every other attribute; and o_3, with $o_3[V_{i,j}] = o_3[V_j] = \top$, and $o_3[X] = \bot$ for every other attribute X. So, for every alternative $o \in D$, $o[V_j] = o[V_i] \vee o[V_{i,j}]$. In the following, we will refer to $\{(V_i, V_{i,j}, V_j) \mid V_i \to V_j \in E\}$ as "triplets".

The optimal and minimal CP-net ϕ minimizes the empirical score $NMCL_{p_D}(\succ) = \frac{1}{n} \sum_N \sum_{v \in \underline{Pa(N)}} p_{D,err}(N, v)$. Let X be any node of ϕ and Y be a node that never appears in the same triplet as X. We can show by contradiction that there is no edge (Y, X) in ϕ by constructing a CP-net without this edge and showing that its score is not greater than ϕ's score. So, for each node N, its set of parents in ϕ is a subset of the triplets N is a part of.

Then, because ϕ minimizes the empirical normalized mean code length, we show that the subgraph for each triplets can only be of two sorts: structure A or B, as depicted below:

A:
$$\begin{tikzcd}
\boxed{\bot}\ \boxed{V_i} \arrow[dr] & & \\
& \boxed{V_j} & \boxed{\begin{array}{c} V_i = \bot \wedge V_{i,j} = \bot : \bot \\ V_i = \top \vee V_{i,j} = \top : \top \end{array}} \\
\boxed{\bot}\ \boxed{V_{i,j}} \arrow[ur] & &
\end{tikzcd}$$

B:
$$\begin{tikzcd}
\boxed{\bot}\ \boxed{V_i} & & \\
& \boxed{V_j} & \boxed{\begin{array}{c} V_{i,j} = \bot : \bot \\ V_{i,j} = \top : \top \end{array}} \\
\boxed{\bot}\ \boxed{V_{i,j}} \arrow[ur] & &
\end{tikzcd}$$

$V_{i,j}$ cannot appear in multiple triplets. For this reason, structure B cannot introduce any cycle in the CP-net, while it can be the case for structure A. Therefore the optimal acyclic CP-net has a maximum number of occurrences of structure A, and will use structure B in all other cases.

Given an optimal CP-net, we can provide a solution to the initial FAS problem as follows: for every triplet $(V_i, V_{i,j}, V_j)$, if V_i is not the parent of V_j, then $V_i \to V_j$ is added to the feedback arc set. The size of this set is the number of structure B, which is minimized by the optimal CP-net. For this reason, the constructed set is a minimal feedback arc set. This reduction is polynomial because the algorithms to construct the dataset and the minimal feedback arc set are in $O(|V| + |E|)$.

Therefore, the problem of finding the CP-net that minimizes the normalized mean code length is NP-hard. It is in NP because the empirical normalized mean code length can be computed in polynomial time. □

6 CP-net learning algorithm

The learning algorithm we propose has two parts: the structure learning and the conditional preference tables fitting.

6.1 Structure learning with hill climbing

Since CP-nets learning is NP-complete, we propose an approximate learning algorithm. We propose to use a hill-climbing approach based on a series of greedy improvements of the score by local modifications of the model. This approach is akin to what is used in Bayesian networks learning, with the difference that we rely on the cost defined by Eq. (2) and (1) rather than the typical metrics based on the log-likelihood. This method iteratively modifies the current best model with random transformations and checks whether one altered model has a lower cost. In that case, the algorithm selects this model and restarts the modifications. The operations we propose to use for CP-nets learning are the classical transformations for Bayesian network learning: 1) edge addition, 2) edge deletion, and 3) edge inversion. The method is summarized in Algorithm 1. This algorithm takes as parameter an initial CP-net ϕ'. It can be, for example, a separable CP-net, i.e., a CP-net whose underlying directed acyclic graph has no edge. In our implementation, we also added a tabu list.

Algorithm 1: Hill climbing search for CP-net learning

```
Data: a dataset $D$, an initial CP-net $\phi'$
Result: a CP-net $\phi$
1  $score \leftarrow L(\phi') + L(D|\phi')$; $previous\_score \leftarrow +\infty$
2  while $score < previous\_score$ do
3      $\phi \leftarrow \phi'$
4      $neighbors \leftarrow transformations(\phi)$
5      remove non-acyclic graphs from $neighbors$
6      fit CPTs of $neighbors$ from $D$
7      $\phi' \leftarrow \arg\min_{\phi'' \in neighbors} L(\phi'') + L(D|\phi'')$
8      $previous\_score \leftarrow score$
9      $score \leftarrow L(\phi') + L(D|\phi')$
10 return $\phi$
```

6.2 Conditional preference tables fitting

Let N be a node labeled by attribute X. To fit the CPT of N, one can order, for each value v of $\underline{Pa(N)}$, the values of N in decreasing order of their number of occurrences given v, so the preferred value for X given v is $\arg\max_{x \in \underline{X}} p_D(vx)$. This method minimizes Eq. (4) for a fixed structure.

7 Experiments

We assess CP-net learning with a practical application to recommendation systems by applying an experimental protocol similar to that of Fargier et al. [2018]. In these experiments, we seek to answer the following questions:

- What is the best model selection heuristic when using clustering?
- Is converting a Bayesian network to a CP-net an effective learning tool?
- Does the use of clusters increase the recommendation accuracy of CP-nets?
- How does our method compare with the k-LP-tree approach proposed in [Fargier et al., 2018]?
- What recommendation methods are compatible with online applications?

The source code, the learned models and the datasets are available online[1].

7.1 Experimental protocol

The experiments use three datasets provided by Renault, the French car manufacturer. They are three anonymized sales history: "small" (48 attributes, 27,088 vectors), "medium" (44 attributes, 14,786 vectors) and "big" (87 attributes, 17,715 vectors). Each dataset is split into a training set (80%) and a test set (20%). To assess the learnt models, we simulate an online configuration as described in algorithm 2: for each vector o in the test set, we draw uniformly an order over the attributes, and, one attribute V at a time, the model provides a recommended value r for V given the current partial instantiation u; r is then compared to $o[V]$ and u is updated with $o[V]$. To reduce the variance of the results, we draw 100 orders per vector.

Algorithm 2: Recommendation in configure experimental protocol

```
Data: a test set $D_{test}$, a model $M$
Result: the mean recommendation accuracy of $M$
1  $success \leftarrow 0$
2  for $o \in D_{test}$ do
3      for $i$ from 1 to 100 do
4          $order \leftarrow$ random order over $\mathcal{X}$
5          $u \leftarrow$ empty vector
6          for $V \in order$ do
7              $r \leftarrow$ M.recommend($V \mid u$)
8              if $o[V] = r$ then $success \leftarrow success + 1$;
9              $u[V] \leftarrow o[V]$
10 return $success/(100 \times n \times |D_{test}|)$
```

7.2 Clustering

Acyclic CP-nets have a limited expressivity: there exist orders that can only be represented by cyclic CP-nets. Fargier et al. [2018], the authors faced a similar issue with LP-trees and proposed to use clustering to alleviate this reduced expressivity. They start by clustering the dataset with the k-means algorithm using the Hamming distance and then learning one model for each cluster. The algorithm then maintains one model for every cluster. To compute the recommended valeur $opt(u)$, they use the model whose cluster's centroid is the closest (according to the Hamming distance) to u.

Since this approach was very effective for their experimental settings, we propose to adapt it to CP-nets as well. We also propose a new heuristic for selecting the model for preferential optimization. Instead of checking the distance between u and the centroids of the clusters, this heuristic selects the model most suited for the partially defined alternative u: to compute $opt(u)$, we use the model that minimizes $s_u(\succ) = |code(opt(u, \succ))|$, that we name "Minimal code".

Table 1 shows the accuracy of recommendations by CP-nets with clustering with various model selection heuristics. In this experiment, the clusters and the models are fixed, so only the model selection heuristics impact the results. Random selects a random model and is presented as a baseline. "Closest centroid" is the method proposed in [Fargier et al., 2018]. Choosing a random model yields poor results, motivating the need for a more sophisticated method. The "Minimal code" heuristic yields generally the best accuracy, although the difference with "Closest centroid" is low. So, since it has better recommendation accuracy and has interesting properties, we conclude that "Minimal code" is a more compelling heuristic. In the following, we will only use this heuristic.

Dataset	Random	Closest centroid	Minimal code
Small	61.61%	87.52%	**88.12%**
Medium	67.25%	86.97%	**88.68%**
Big	80.46%	**92.41%**	92.35%

Table 1: Accuracy of a CP-net with clustering for three different model selection heuristics

7.3 Learning CP-nets from Bayesian network

Khoshkangini et al. [2018] propose to learn CP-nets by learning a Bayesian network and converting it into a CP-net. We now compare this approach with our method. Besides, we propose a hybrid approach, where a Bayesian network is used to initialize a CP-net that is then optimized with our MDL-score guided hill climbing learning method. We use an open-source Python implementation *pgmpy* to learn the Bayesian network. The BN learning algorithm is hill climbing, with the BIC score and a K2 prior. Table 2 contains the recommendation accuracy of CP-nets learned using these three strategies. First, remark that the difference in accuracy is small, indicating that the method proposed by Khoshkangini et al. [2018] is indeed effective, even though there is no apparent connection between NMCL (cf. Eq. (4)) and log-likelihood-based scores. However, our approach generally yields better results. Finally, converting a Bayesian network into a CP-net and then optimizing it with hill climbing can be a good trade-off between learning time (Bayesian network learning libraries are very optimized) and accuracy.

Dataset	HC	BN	BN+HC
Small	84.90%	84.75%	**85.61%**
Medium	**87.43%**	87.12%	87.12%
Big	**91.46%**	90.11%	90.92%

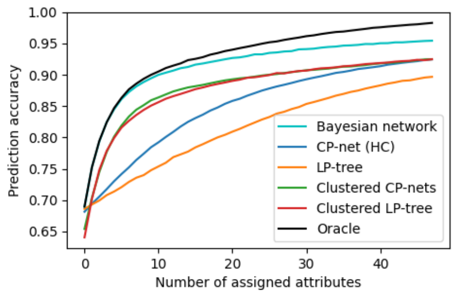
Table 2: Accuracy of a CP-net learnt with hill climbing guided by the MDL score (HC), derived from a Bayesian network (BN) or derived from a Bayesian network and optimized with hill climbing guided by the MDL score (BN+HC)

7.4 Recommendation accuracy and time

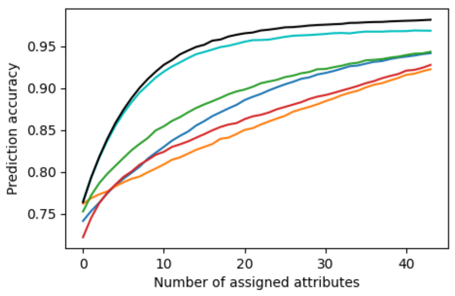
To assess the recommendation accuracy and time, we propose to compare the following models:

- the "Oracle" (proposed in [Fargier et al., 2018]), which gives an upper bound on the recommendation accuracy;

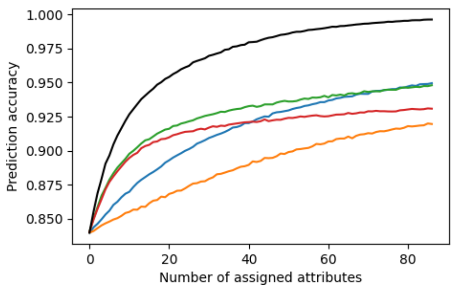
[1]removed for anonymous submission



(a) Accuracy over the “small” dataset

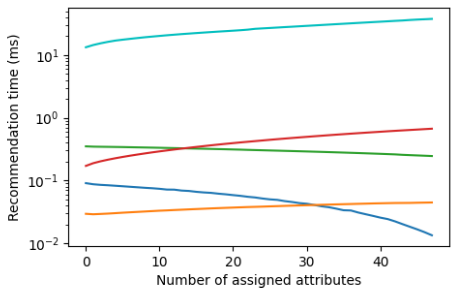


(b) Accuracy over the “medium” dataset

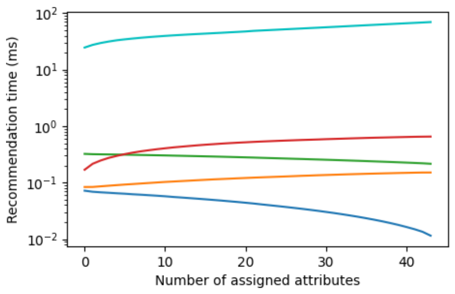


(c) Accuracy over the “big” dataset

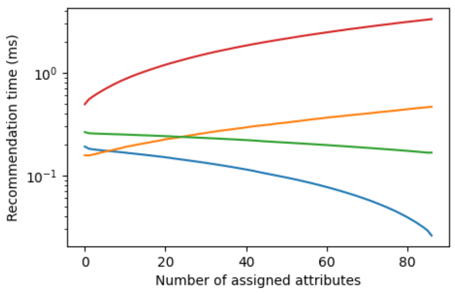
Figure 2: Recommendation accuracy over the three datasets



(a) Time over the “small” dataset



(b) Time over the “medium” dataset



(c) Time over the “big” dataset

Figure 3: Recommendation time (in ms) over the three datasets

- Bayesian network, used with the exact inference algorithm “variable elimination”;
- our implementation of k-LP-tree learning (with $k = 3$ and $\tau = 20$), as well as a version with 3 clusters;
- CP-nets learnt with our MDL approach (Algo. 1), as well as a version with 3 clusters.

Recommendation accuracy are presented in Fig. 2a, 2b and 2c. As expected, the “oracle” (in black) always has the highest accuracy since it is an upper bound. The Bayesian network (in cyan) always has the highest accuracy and is close to this upper bound, except on the “big” dataset where it did not finish the recommendations within our 72-hour timeout. The two models using clusters (in red and green) always outperform their base models. Among these models, the clustered CP-nets have the highest accuracy. Finally, LP-trees have the lowest accuracy of all tested models. This can be most notably explained by their strong assumption that the target order is lexicographic, while such an hypothesis is not necessary for CP-net learning. This can be observed for higher numbers of assigned attributes: even non-clustered CP-nets have generally higher accuracy than clustered LP-trees.

Recommendation time are presented in Fig. 3a, 3b and 3c. Bayesian networks are the slowest model by up to two orders of magnitude. Indeed, the MAP query used for recommendation is NP-hard, while it is polytime for LP-trees and CP-nets. In fact, this complexity can explain why it did not finish on the “big” dataset. We suppose this is mostly due to the Python library we used, which only proposes the “variable elimination” algorithm, even though more effective inference algorithms exist [Pourret et al., 2008]. The recommendation times for CP-nets and LP-trees are similar, even though CP-nets are generally faster. The use of clusters makes the recommendation longer. However, even for the biggest dataset, the recommendation time is always within 3 ms, making it usable on terminals with limited computational power, like smartphones or embedded systems. Besides, we expect implementations in a compiled language (like C or Rust) to be one or two orders of magnitude faster.

8 Conclusion and perspectives

This paper gives the first theoretical analysis of unsupervised learning of CP-nets. It introduces the NMCL metric and shows that the sample complexity is polynomial and that learning the optimal CP-nets is NP-complete. Finally, we demonstrated the usefulness of this approach by comparing it with various state-of-the-art methods on a real-world recommendation task. More generally, the NMCL can be used to learn any model with an efficient preferential optimization algorithm, so this approach is not limited to CP-nets.

In future works, we plan to study the connection between Bayesian networks and CP-nets to support theoretically the method of transforming Bayesian networks into CP-nets. We conjecture that the difference of recommendation accuracy between Bayesian networks and CP-nets is due to the limited expressivity of acyclic CP-nets, and so we plan to study the learning of (subclasses of) cyclic CP-nets.

References

Eisa Alanazi, Malek Mouhoub, and Sandra Zilles. The Complexity of Learning Acyclic CP-Nets. In Subbarao Kambhampati, editor, *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI 2016)*, pages 1361–1367. IJCAI/AAAI Press, 2016. ISBN 978-1-57735-770-4. URL http://www.ijcai.org/Abstract/16/196.

Eisa Alanazi, Malek Mouhoub, and Sandra Zilles. The complexity of exact learning of acyclic conditional preference networks from swap examples. *Artifical Intelligence*, 278, 2020. doi: 10.1016/j.artint.2019.103182.

Thomas E. Allen, Cory Siler, and Judy Goldsmith. Learning tree-structured cp-nets with local search. In Vasile Rus and Zdravko Markov, editors, *Proceedings of the Thirtieth International Florida Artificial Intelligence Research Society Conference (FLAIRS 2017)*, pages 8–13. AAAI Press, 2017. URL https://aaai.org/ocs/index.php/FLAIRS/FLAIRS17/paper/view/15467.

Damien Bigot, Jérôme Mengin, and Bruno Zanuttini. Learning probabilistic cp-nets from observations of optimal items. In *STAIRS*, pages 81–90, 2014.

Craig Boutilier, editor. *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI'09)*, 2009.

Craig Boutilier, Romen I. Brafman, Carmel Domshlak, Holger H. Hoos, and David Poole. CP-nets: a tool for representing and reasoning with conditional ceteris paribus preference statements. *Journal of Artificial Intelligence Research*, 21:135–191, 2004.

Yann Chevaleyre, Frédéric Koriche, Jérôme Lang, Jérôme Mengin, and Bruno Zanuttini. Learning Ordinal Preferences on Multiattribute Domains: the Case of CP-nets. In Johannes Fürnkranz and Heike Hüllermeier, editors, *Preference learning*, pages 273–296. Springer, 2011.

Yannis Dimopoulos, Loizos Michael, and Fani Athienitou. Ceteris paribus preference elicitation with predictive guarantees. In Boutilier [2009], pages 1890–1895.

Hélène Fargier, Pierre-François Gimenez, and Jérôme Mengin. Experimental evaluation of three value recommendation methods in interactive configuration. *The Journal of Universal Computer Science*, 26(3):318–342, 2020. URL http://www.jucs.org/jucs_26_3/experimental_evaluation_of_three.

Hélène Fargier, Pierre-François Gimenez, Jérôme Mengin, and Bao Ngoc Le Nguyen. The complexity of unsupervised learning of lexicographic preferences. In Meltem Öztürk, Paolo Viappiani, Christophe Labreuche, and Sébastien Destercke, editors, *Proceedings of the 13th Multidisciplinary Workshop on Advances in Preference Handling*, 2022.

Hélène Fargier, Pierre Francois Gimenez, and Jérôme Mengin. Learning lexicographic preference trees from positive examples. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI 2018)*, page 2959–2966. AAAI Press, 2018. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17272/16610.

Niall M Fraser. Applications of preference trees. In *Proceedings of SMC'93*, pages 132–136, 1993.

Yoav Freund, Raj D. Iyer, Robert E. Schapire, and Yoram Singer. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4:933–969, 2003.

Minos Garofalakis, Aristides Gionis, Rajeev Rastogi, Sridhar Seshadri, and Kyuseok Shim. Xtract: learning document type descriptors from xml document collections. *Data mining and knowledge discovery*, 7:23–56, 2003.

Pierre-François Gimenez and Jérôme Mengin. Conditionally acyclic co-networks for efficient preferential optimization. In *26th European Conference on Artificial Intelligence (ECAI 2023)*, 2023.

Christophe Gonzales and Patrice Perny. GAI networks for utility elicitation. In *Proceedings of KR'04*, pages 224–234, 2004.

Peter Grünwald. Model selection based on minimum description length. *Journal of mathematical psychology*, 44(1): 133–152, 2000.

Cynthia Huffman and Barbara E Kahn. Variety for sale: mass customization or mass confusion? *Journal of retailing*, 74 (4):491–513, 1998.

Richard M Karp. *Reducibility among combinatorial problems*. Springer, 2010.

Reza Khoshkangini, Maria Silvia Pini, Francesca Rossi, and Dinh Tran-Van. Constructing cp-nets from users past behaviors. In *CIKM Workshops*, 2018.

Neville Kenneth Kitson, Anthony C Constantinou, Zhigao Guo, Yang Liu, and Kiattikun Chobtham. A survey of Bayesian network structure learning. *Artificial Intelligence Review*, pages 1–94, 2023.

Frédéric Koriche and Bruno Zanuttini. Learning conditional preference networks. *Artificial Intelligence*, 174(11):685–703, 2010. doi: 10.1016/j.artint.2010.04.019.

Johan Kwisthout. Most probable explanations in bayesian networks: Complexity and tractability. *International Journal of Approximate Reasoning*, 52(9):1452–1469, 2011.

Fabien Labernia, Bruno Zanuttini, Brice Mayag, Florian Yger, and Jamal Atif. Online learning of acyclic conditional preference networks from noisy data. In Vijay Raghavan, Srinivas Aluru, George Karypis, Lucio Miele, and Xindong Wu, editors, *IEEE International Conference on Data Mining (ICDM 2017)*, pages 247–256. IEEE Computer Society, 2017. doi: 10.1109/ICDM.2017.34.

Wai Lam and Fahiem Bacchus. Learning bayesian belief networks: An approach based on the mdl principle. *Computational intelligence*, 10(3):269–293, 1994.

Jérôme Lang and Jérôme Mengin. The complexity of learning separable ceteris paribus preferences. In Boutilier [2009], pages 848–853.

Su Liu and Jinglei Liu. Cp-nets structure learning based on mrmcr principle. *IEEE Access*, 7:121482–121492, 2019.

Osman A Mian, Alexander Marx, and Jilles Vreeken. Discovering fully oriented causal networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8975–8982, 2021.

Judea Pearl. Bayesian netwcrks: A model cf self-activated memory for evidential reasoning. In *Proceedings of the 7th conference of the Cognitive Science Society, University of California, Irvine, CA, USA*, pages 15–17, 1985.

Judea Pearl. *Causality*. Cambridge university press, 2009.

Olivier Pourret, Patrick Na, Bruce Marcot, et al. *Bayesian networks: a practical guide to applications*. John Wiley & Sons, 2008.

Jorma Rissanen. A universal prior for integers and estimation by minimum description length. *The Annals of statistics*, 11(2):416–431, 1983.

Joe Suzuki. A construction of bayesian networks from databases based on an mdl principle. In *Uncertainty in Artificial Intelligence*, pages 266–273. Elsevier, 1993.

Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

Linda L Zhang. Product configuration: a review of the state-of-the-art and future research. *International Journal of Production Research*, 52(21):6381–6398, 2014.