Interactive Learning of Acyclic Conditional Preference Networks

Eisa Alanazi[1], Malek Mouhoub[2], and Sandra Zilles[2]

[1]Department of Computer Science
Umm Al-Qura University
Makkah, Saudi Arabia
[2]Department of Computer Science
University of Regina
Regina, SK, Canada S4S 0A2

Abstract

Learning of user preferences, as represented by, for example, Conditional Preference Networks (CP-nets), has become a core issue in AI research. Recent studies investigate learning of CP-nets from randomly chosen examples or from membership and equivalence queries. To assess the optimality of learning algorithms as well as to better understand the combinatorial structure of classes of CP-nets, it is helpful to calculate certain learning-theoretic information complexity parameters. This paper determines bounds on or exact values of some of the most central information complexity parameters, namely the VC dimension, the (recursive) teaching dimension, the self-directed learning complexity, and the optimal mistake bound, for classes of acyclic CP-nets. We further provide an algorithm that learns tree-structured CP-nets from membership queries. Using our results on complexity parameters, we assess the optimality of our algorithm as well as that of another query learning algorithm for acyclic CP-nets presented in the literature. Our algorithm is near-optimal, and can, under certain assumptions be adapted to the case when the membership oracle is faulty.

1 Introduction

Since preference learning is important in many AI applications, there is a need for a strong theoretical underpinning of research on this topic. In recent years, substantial advances have been made in this field, for example in the design of Conditional Preference Networks (CP-nets) [1] and the study of their learnability [2, 3, 4, 5, 6, 7]. A CP-net is a compact preference representation for multi-attribute domains where the preference of one attribute may depend on the values of other attributes.

Koriche and Zanuttini [2] investigated query learning of bounded acyclic CP-nets (i.e., with a bound on the number of attributes on which the preferences for any attribute may depend). Their successful algorithms used both membership and equivalence queries, cf. [8], while they proved that equivalence queries alone are not sufficient for efficient learnability. CP-nets have also been studied in models of passive learning from examples, both for batch learning [3, 4, 5, 7] and for online learning [6].

A fundamental question in assessing the proposed algorithms is how many queries or examples would be needed by the best possible learning algorithm in the given learning model. For several models, lower bounds can be derived from the Vapnik Chervonenkis dimension (VCD, [9]). This central parameter is one of several that, in addition to yielding bounds on the performance of learning algorithms, provide deep insights into the combinatorial structure of the studied concept class. Such insights can in turn help to design new learning algorithms.

Our main contributions are the following:

(a) We provide the first study that exactly calculates the VCD for the class of unbounded acyclic CP-nets, and give a lower bound for any bound k. So far, the only existing studies present a lower bound [2], which we prove incorrect for large values of k, and asymptotic complexities [10]. The latter show that VCD $\in \Theta(2^n)$ for $k = n-1$ and VCD $\in \tilde{\Theta}(n2^k)$ when $k \in o(n)$, in agreement with our result that VCD $= 2^n - 1$ for $k = n-1$, and is at least VCD $\geq 2^k - 1 + (n-k)2^k$ for general values of k. It should be noted that both previous studies assume that CP-nets can be incomplete, i.e., for some variables no preference relations may be given. In our study, we make the (not uncommon) assumption that CP-nets are complete, but our result specifically on VCD also applies to the more general case that includes incomplete CP-nets. Further, our results are more general than existing ones in that they cover also the case of CP-nets with multi-valued variables (as opposed to binary variables).

As a byproduct of our study, we obtain that the VCD of the class of *all* consistent CP-nets (whether acyclic or cyclic)[1] equals that of the class of all

[1] A consistent CP-net is one that does not prefer an outcome o over another outcome $\hat{o}$ while at the same time preferring $\hat{o}$ over o. Acyclic CP-nets are always consistent, but

acyclic CP-nets. Hence, the class of acyclic CP-nets is less expressive than that of all consistent CP-nets, but may (at least in some models) be as hard to learn.

(b) We further provide exact values (or, in some cases, non-trivial bounds) for other important information complexity parameters, namely the teaching dimension [11], the recursive teaching dimension [12], the self-directed learning complexity [13], and the optimal mistake bound [14].

(c) We present a new algorithm that learns tree-structured CP-nets from membership queries and use our results on the teaching dimension to show that our algorithm is close to optimal.

(d) In most real-world scenarios, one would expect some degree of noise in the responses to membership queries, or that sometimes no response at all is obtained. To address this issue, we demonstrate how, under certain assumptions on the noise and the missing responses, our algorithm for learning tree CP-nets can be adapted to handle incomplete or incorrect answers to membership queries.

(e) We re-assess the degree of optimality of Koriche and Zanuttini's algorithm for learning bounded acyclic CP-nets, using our result on the VCD.

This article extends a previous conference paper [15].

2 Background

2.1 Conditional Preference Networks (CP-nets)

We largely follow the notation introduced by Boutilier et al. [1] in their seminal work on CP-nets.

Let $V = \{v_1, v_2, \dots, v_n\}$ be a set of attributes or variables. Each variable $v_i \in V$ has a set of possible values (its domain) $D_{v_i} = \{v^i_1, v^i_2, \dots, v^i_m\}$. We assume that every domain D_{v_i} is of a fixed size $m \geq 2$, independent of i. An assignment x to a set of variables $X \subseteq V$ is a mapping for every variable $v_i \in X$ to a value from D_{v_i}. We denote the set of all assignments of $X \subseteq V$ by $\mathcal{O}_X$ and remove the subscript when $X = V$. A preference is an irreflexive, transitive binary relation $\succ$. For any $o, o' \in \mathcal{O}$, we write $o \succ o'$ (resp. $o \not\succ o'$) to denote the fact that o is strictly preferred (resp. not preferred) to o', where o and o' are incomparable w.r.t. $\succ$ if both $o \not\succ o'$ and $o' \not\succ o$ holds. We use $o[X]$ to denote the projection of o onto $X \subset V$ and write $o[v_i]$ instead of $o[\{v_i\}]$.

cyclic ones are not necessarily so.

The CP-net model captures complex qualitative preference statements in a graphical way. Informally, a CP-net is a set of statements of the form $u : v^i_j \succ v^i_k$ which states that the preference over v_i with $D_{v_i} = \{v^i_1, v^i_2, \ldots, v^i_m\}$ is conditioned upon the assignment of $U \subseteq V \backslash \{v_i\}$ where $v^i_j, v^i_k \in D_{v_i}$. In particular, when U has the value $u \in \mathcal{O}_U$, v^i_j is preferred to v^i_k as a value of v_i *ceteris paribus* (all other things being equal). That is, for any two outcomes $o, o' \in \mathcal{O}$ where $o[v_i] = v^i_j$ and $o'[v_i] = v^i_k$ the preference holds when i) $o[U] = o'[U] = u$ and ii) $o[Z] = o'[Z]$ for all $Z = V \backslash U \cup \{v_i\}$. In such case, we say o is preferred to o' ceteris paribus. Clearly, there could be exponentially many pairs of outcomes (o,o') that are affected by one such statement.

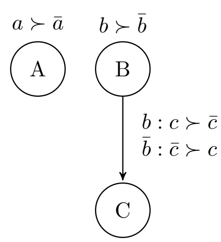
CP-nets provide a compact representation of preferences over $\mathcal{O}$ by providing such statements for every variable. For every $v_i \in V$, the decision maker[2] chooses a set $Pa(v_i) \subseteq V \backslash \{v_i\}$ of parent variables that influence the preference order of v_i. For any $u \in \mathcal{O}_{Pa(v_i)}$, the decision maker specifies an ordering $\succ^{v_i}_u$ over D_{v_i}. We refer to $\succ^{v_i}_u$ as the conditional preference statement of v_i in the context of u. A Conditional Preference Table for v_i, CPT(v_i), is a set of conditional preference statements $\{\succ^{v_i}_{u_1}, \ldots, \succ^{v_i}_{u_k}\}$. CPT($v_i$) is said to be complete, if, for every element in $\mathcal{O}_{Pa(v_i)}$, CPT(v_i) contains a statement that imposes a total order on D_{v_i}; otherwise CPT(v_i) is incomplete.

Definition 1 (CP-net [1])**.** *Given, V, $Pa(v)$, and* CPT(v) *for $v \in V$, a CP-net is a directed graph (V, E), where, for any $v_i, v_j \in V$, $(v_i, v_j) \in E$ iff $v_i \in Pa(v_j)$.*

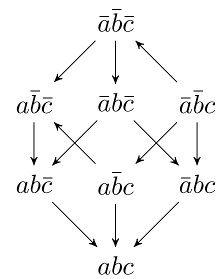
Analogously, a CP-net N is said to be complete if every CPT it poses is complete; otherwise it is incomplete. A CP-net is acyclic if it contains no cycles. It is separable if the edge set of its graph is the empty set and it is a tree if it is acyclic with indegree at most one. Lastly, we assume CP-nets are defined in their minimal form, i.e., there is no *dummy* parent in any CPT that actually does not affect the preference relation.

Example 1. *Figure 1a shows a CP-net over $V = \{A, B, C\}$ with $D_A = \{a, \bar{a}\}$, $D_B = \{b, \bar{b}\}$, $D_C = \{c, \bar{c}\}$. Each variable is annotated with its* CPT. *For variable A, the user prefers a to $\bar{a}$ unconditionally. For C, the preference depends on the values of B, i.e., $Pa(C) = \{B\}$. For instance, in the context of $\bar{b}$, $\bar{c}$ is preferred over c.*

[2] This can be any entity in charge of constructing the preference network, i.e., a computer agent, a person, a group of people, etc.



(a) The CP-net.



(b) The induced preference graph.

Figure 1: An acyclic CP-net (c.f. Def. 1) and its induced preference graph (c.f. Def. 2).

Two outcomes $o, \hat{o} \in \mathcal{O}$ are swap outcomes ('swaps' for short) if they differ in the value of exactly one variable v_i; then v_i is called the swapped variable [1].

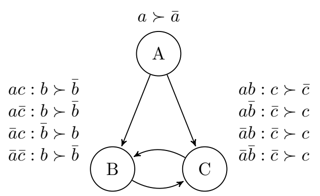
The size of a preference table for a variable v_i, denoted by $size(\mathrm{CPT}(v_i))$, is the number of preference statements it holds which is $m^{|Pa(v_i)|}$ if $\mathrm{CPT}(v_i)$ is complete. The size of a CP-net N is defined as the sum of its tables' sizes.

Example 2. *In Figure 1, $abc, \bar{a}bc$ are swaps over the swapped variable A. The size of the CP-net is* $1 + 1 + 2 = 4$.

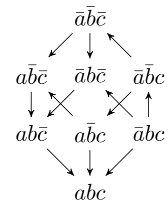
The semantics of CP-nets are described in terms of improving flips. Let $u \in \mathcal{O}_{Pa(v_i)}$ be an assignment of the parents for a variable $v_i \in V$. Let $\succ_u^{v_i} = v_1^i \succ \ldots \succ v_m^i$ be the preference order of v_i in the context of u. Then, all else being equal, going from v_j^i to v_k^i is an improving flip over v_i whenever $k < j \leq m$.

Example 3. *In Figure 1a, $(a\bar{b}c, abc)$ is an improving flip with respect to the variable B.*

The improving flip notion makes every pair $(o, \hat{o})$ of swap outcomes comparable, i.e., either $o \succ \hat{o}$ or $\hat{o} \succ o$ holds [1]. The question "is $o \succ \hat{o}$?" is then a special case of a so-called dominance query and can be answered directly from the preference table of the swapped variable. Let v_i be the swapped variable of a swap $(o, \hat{o})$. Let u be the context of $Pa(v_i)$ in both o and $\hat{o}$. Then, $o \succ \hat{o}$ iff $o[v_i] \succ_u^{v_i} \hat{o}[v_i]$. A general dominance query is of the form: given two outcomes $o, \hat{o} \in \mathcal{O}$, is $o \succ \hat{o}$? The answer is yes, iff o is preferred to



(a) The network.



(b) The induced preference graph.

Figure 2: An example of a *consistent* cyclic CP-net.

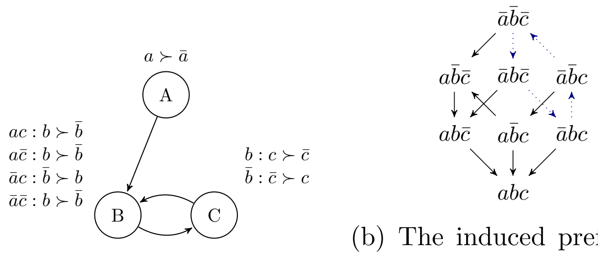
$\hat{o}$, i.e., there is a sequence $(\lambda_1, \ldots, \lambda_n)$ of improving flips from $\hat{o}$ to o, where $\hat{o} = \lambda_1$, $o = \lambda_n$, and $(\lambda_i, \lambda_{i+1})$ is an improving flip for all $i \in \{1, \ldots, n-1\}$ [1].

Example 4. *In Figure 1b, $abc \succ \bar{a}\bar{b}c$, as witnessed by the sequence $\bar{a}\bar{b}c \to a\bar{b}c \to abc$ of improving flips.*

Definition 2 (Induced Preference Graph [1])**.** *The induced preference graph of a CP-net N is a directed graph G where each vertex represents an outcome $o \in \mathcal{O}$. An edge from $\hat{o}$ to o exists iff $(o, \hat{o}) \in \mathcal{O} \times \mathcal{O}$ is a swap w.r.t. some $v_i \in V$ and $o[v_i]$ precedes $\hat{o}[v_i]$ in $\succ^{v_i}_{o[Pa(v_i)]}$.*

Therefore, a CP-net N defines a partial order $\succ$ over $\mathcal{O}$ that is given by the transitive closure of its induced preference graph. If $o \succ \hat{o}$ we say N entails $(o, \hat{o})$. N is consistent if there is no $o \in \mathcal{O}$ with $o \succ o$, i.e., if its induced preference graph is acyclic. Acyclic CP-nets are guaranteed to be consistent while such guarantee does not exist for the cyclic CP-nets and their consistency depends on the actual values of the CPTs [1]. Lastly, the complexity of finding the best outcome in an acyclic CP-net has been shown to be linear [1] while the complexity of answering dominance queries depends on the structure of CP-nets: PSPACE-complete for arbitrary (cyclic and acyclic) consistent CP-nets [16] and linear in case of trees [17].

Example 5. *Figure 2 shows an example of a cyclic CP-net that is consistent while Figure 3 shows an inconsistent one. Note that both share the same* CPT*s except for* CPT(C)*. The dotted edges in the induced preference graph of Figure 3 represent a cycle.*



(a) The network.

(b) The induced preference graph.

Figure 3: An example of an *inconsistent* cyclic CP-net.

2.2 Concept Learning

The first part of our study is concerned with determining—for the case of acyclic CP-nets—the values of information complexity parameters that are typically studied in computational learning theory. By information complexity, we mean the complexity in terms of the amount of information a learning algorithm needs to identify a CP-net. Examples of such complexity notions will be introduced below.

A specific complexity notion corresponds to a specific formal model of machine learning. Each such learning model assumes that there is an information source that supplies the learning algorithm with information about a hidden target concept c^*. The latter is a member of a concept class, which is simply the class of potential target concepts, and, in the context of this paper, also the class of hypotheses that the learning algorithm can formulate in the attempt to identify the target concept c^*.

Formally, one fixes a finite set $\mathcal{X}$, called instance space, which contains all possible instances (i.e., elements of) an underlying domain. A concept c is then defined as a mapping from $\mathcal{X}$ to $\{0, 1\}$. Equivalently, c can be seen as the set $c = \{x \in \mathcal{X} \mid c(x) = 1\}$, i.e., a subset of the instance space. A *concept class* $\mathcal{C}$ is a set of concepts. Within the scope of our study, the information source (sometimes called oracle), supplies the learning algorithm in some way or another with a set of *labelled examples* for the target concept $c^* \in C$. A labelled example for c^* is a pair $(x, b) \in \mathcal{X} \times \{0, 1\}$ where $x \in \mathcal{X}$ and $b = c(x)$. Under the set interpretation of concepts, this means that $b = 1$ if and only if the instance x belongs to the concept c^*. A concept c is consistent with a set $S \subseteq X \times \{0, 1\}$ of labelled examples, if and only if $c(x) = b$ for all $(x, b) \in S$, i.e., if every element of S is an example for c.

In practice, a concept is usually *encoded* by a representation $\sigma(c)$ defined based on a representation class $\mathcal{R}$ [18]. Thus, one usually has some fixed representation class $\mathcal{R}$ in mind, with a one-to-one correspondence between the concept class C and its representation class R. We will assume in what follows that the representation class is chosen in a way that minimizes the worst case size of the representation of any concept in C. Generally, there may be various interpretations of the term "size;" since we will focus on learning CP-nets, we use CP-nets as representations for concepts, and the size of a representation is simply the size of the corresponding CP-net as defined above.

At the onset of a learning process, both the oracle and the learning algorithm (often called *learner* for short) agree on the representation class $\mathcal{R}$ (and thus also on the concept class C), but only the oracle knows the target concept c^*. After some period of communication with the oracle, the learner is required to identify the target concept c^* either exactly or approximately.

Many learning models have been proposed to deal with different learning settings [18, 8, 14, 19]. These models typically differ in the constraints they impose on the oracle and the learning goal. One also distinguishes between learners that actively query the oracle for specific information content and learners that passively receive a set of examples chosen solely by the information source. One of the best known passive learning models is the *Probably Approximately Correct* (PAC) model [19]. The PAC model is concerned with finding, with high probability, a close approximation to the target concept c^* from randomly chosen examples. The examples are assumed to be sampled independently from an unknown distribution. On the other end of the spectrum, a model that requires exact identification of c^* is Angluin's model for learning from queries [8]. In this model, the learner actively poses queries of a certain type to the oracle.

In this paper, we consider specifically two types of queries introduced by Angluin [8], namely membership queries and equivalence queries. A membership query is specified by an element $x \in \mathcal{X}$ of the instance space, and it represents the question whether or not c^* contains x. The oracle supplies the learner with the correct answer, i.e., it provides the label $c^*(x)$ in response to the membership query for x. In an equivalence query, the learner specifies a hypothesis c. If $c = c^*$, the learning process is completed as the learner has then identified the target concept. If $c \neq c^*$, the learner is provided with a labelled example $(x, c^*(x))$ that witnesses $c \neq c^*$. That means, $c^*(x) \neq c(x)$. Note that x in this case can be any element in the symmetric difference of the sets associated with c and c^*.

A class $\mathcal{C} \subseteq \{0,1\}^n$ is learnable from membership and/or equivalence queries via a representation class $\mathcal{R}$ for $\mathcal{C}$, if there is an algorithm $\mathcal{A}$ such that for every concept $c^* \in \mathcal{C}$, $\mathcal{A}$ asks polynomially many adaptive membership and/or equivalence queries and then outputs a hypothesis h that is equivalent to c^*. By adaptivity, we here mean that learning proceeds in rounds; in every round the learner asks a single query and receives an answer from the oracle before deciding on its subsequent query. The number of queries to be polynomial means that it is upper-bounded by a polynomial in n and $size(c^*)$ where $size(c^*)$ is the size of the minimal representation of c^* w.r.t. $\mathcal{R}$.

The above definition is concerned only with the information or query complexity, i.e., the number of queries required to exactly identify any target concept. Moreover, $\mathcal{C}$ is said to be efficiently learnable from membership and/or equivalence queries if there exists an algorithm $\mathcal{A}$ that exactly learns $\mathcal{C}$, in the above sense, and runs in time polynomial in n and $size(c^*)$. Every one of the query strategies we describe in Section 5 gives an obvious polynomial time algorithm in this regard, and thus we will not explicitly mention run-time efficiency of learning algorithms henceforth.

The combinatorial structure of a concept class $\mathcal{C}$ has implications on the complexity of learning $\mathcal{C}$, in particular on the sample complexity (sometimes called information complexity), which refers to the number of labelled examples the learner needs in order to identify any target concept in the class under the constraints of a given learning model. One of the most important complexity parameters studied in machine learning is the Vapnik-Chervonenkis dimension (VCD). In what follows, let $\mathcal{C}$ be a concept class over the (finite) instance space $\mathcal{X}$.

Definition 3. *[9] A subset $Y \subseteq \mathcal{X}$ is shattered by $\mathcal{C}$ if the projection of $\mathcal{C}$ onto Y has $2^{|Y|}$ concepts. The VC dimension of $\mathcal{C}$, denoted by $\mathrm{VCD}(\mathcal{C})$, is the size of the largest subset of $\mathcal{X}$ that is shattered by $\mathcal{C}$.*

The number of randomly chosen examples needed to identify concepts from $\mathcal{C}$ in the PAC-learning model is linear in $\mathrm{VCD}(\mathcal{C})$ [20]. By contrast to learning from random examples, in teaching models, the learner is provided with well-chosen labelled examples.

Definition 4. *[11, 21] A teaching set for a concept $c^* \in \mathcal{C}$ with respect to $\mathcal{C}$ is a set $S = \{(x_1, \ell_1), \ldots, (x_z, \ell_z)\}$ of labelled examples such that c^* is the only concept $c \in \mathcal{C}$ that satisfies $c(x_i) = \ell_i$ for all $i \in \{1, \ldots, z\}$. The teaching dimension of c with respect to $\mathcal{C}$, denoted by $\mathrm{TD}(c, \mathcal{C})$, is the size of the smallest teaching set for c with respect to $\mathcal{C}$. The teaching dimension of $\mathcal{C}$, denoted by $\mathrm{TD}(\mathcal{C})$, is given by $\mathrm{TD}(\mathcal{C}) = \max\{\mathrm{TD}(c, \mathcal{C}) \mid c \in \mathcal{C}\}$.*

Table 1: The class $\mathcal{C}$ of all singletons and the empty concept over a set of t instances, along with the teaching dimension value of each individual concept.

$\mathcal{C}$	x_1	x_2	x_3	x_4	...	x_t	TD
c_0	0	0	0	0	...	0	t
c_1	1	0	0	0	...	0	1
c_2	0	1	0	0	...	0	1
c_3	0	0	1	0	...	0	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
c_t	0	0	0	0	...	1	1

$\mathrm{TD}_{min}(\mathcal{C}) = \min\{\mathrm{TD}(c,\mathcal{C}) \mid c \in \mathcal{C}\}$ denotes the smallest TD of any $c \in \mathcal{C}$. A well-studied variation of teaching is called recursive teaching. Its complexity parameter, the recursive teaching dimension, is defined by recursively removing from $\mathcal{C}$ all the concepts with the smallest TD and then taking the maximum over the smallest TDs encountered in that process. For the corresponding definition of teachers, see [12].

Definition 5. *[12] Let $\mathcal{C}_0 = \mathcal{C}$ and, for all i such that $\mathcal{C}_i \neq \emptyset$, define $\mathcal{C}_{i+1} = \mathcal{C}_i \setminus \{c \in \mathcal{C}_i \mid \mathrm{TD}(c,\mathcal{C}_i) = \mathrm{TD}_{min}(\mathcal{C}_i)\}$. The recursive teaching dimension of $\mathcal{C}$, denoted by $\mathrm{RTD}(\mathcal{C})$, is defined by $\mathrm{RTD}(\mathcal{C}) = \max\{\mathrm{TD}_{min}(\mathcal{C}_i) \mid i \geq 0\}$.*

As an example, consider $\mathcal{C}' = \{c_1, c_2, \ldots, c_t\}$ to be the class of singletons defined over the instance space $\mathcal{X} = \{x_1, x_2, \ldots, x_t\}$ where $c_i = \{x_i\}$ and let $\mathcal{C} = \mathcal{C}' \cup \{c_0\}$, where c_0 is the empty concept, i.e., $c_0(x) = 0\ \forall x \in \mathcal{X}$. Table 1 displays this class along with $\mathrm{TD}(c,\mathcal{C})$ for every $c \in \mathcal{C}$. Since distinguishing the concept c_0 form all other concepts in $\mathcal{C}$ requires t labelled examples, one obtains $\mathrm{TD}(\mathcal{C}) = t$. However, $\mathrm{RTD}(\mathcal{C}) = 1$ as witnessed by $\mathcal{C}_0 = \mathcal{C}'$ (each concept in $\mathcal{C}'$ can be taught with a single example) and $\mathcal{C}_1 = \{c_0\}$ (the remaining concept c_0 has a teaching dimension of 0 with respect to the class containing only c_0).

As opposed to the TD, the RTD exhibits interesting relationships to the VCD. For example, if $\mathcal{C}$ is a maximum class, i.e., its size $|\mathcal{C}|$ meets Sauer's upper bound $\binom{|\mathcal{X}|}{0} + \binom{|\mathcal{X}|}{1} + \ldots + \binom{|\mathcal{X}|}{\mathrm{VCD}(\mathcal{C})}$ [22], and in addition $\mathcal{C}$ can be "corner-peeled"[3], then $\mathcal{C}$ fulfills $\mathrm{RTD}(\mathcal{C}) = \mathrm{VCD}(\mathcal{C})$ [24]. The same equality holds if $\mathcal{C}$ is intersection-closed or has VCD 1 [24].

[3]Corner peeling is a sample compression procedure introduced by Rubinstein and Rubinstein [23]; the actual algorithm or its purpose are not of immediate relevance to our paper.

We will further determine complexity parameters for online prediction, namely the self-directed learning complexity and the optimal mistake bound. A self-directed learner passes a prediction $(x, \ell) \in \mathcal{X} \times \{0, 1\}$ to an oracle, which responds with the information whether or not the target concept c^* fulfills $c^*(x) = \ell$. In case $c^*(x) \neq \ell$, the learner has made a mistake. The self-directed learning complexity SDC($\mathcal{C}$) is the smallest number z for which some self-directed learner exists that makes no more than z mistakes on any concept in $\mathcal{C}$ [13]. In classical online learning [14], the sequence of instances x for which the learner makes label predictions is determined by an adversary. The best worst-case number of mistakes achievable in this model, where again the worst case is taken over all concepts in $\mathcal{C}$, is called the optimal mistake bound of $\mathcal{C}$, denoted by OPT($\mathcal{C}$).

In the example in Table 1, a best possible self-directed learner would guess 0 on any unseen instance and may predict labels for instances in any order. Obviously, such a learner makes at most one mistake, and hence SDC($\mathcal{C}$) = 1. Likewise, OPT($\mathcal{C}$) = 1. Note that also VCD($\mathcal{C}$) = 1, since there is no set of two examples that is shattered by $\mathcal{C}$.

3 The Complexity of Learning CP-nets

Assuming a user's preferences are captured in a target CP-net N^*, an interesting learning problem is to identify N^* from a set of observations representing the user's preferences, i.e., labelled examples, of the form $o \succ o'$ or $o \not\succ o'$ where $\succ$ is the relation induced by N^*.

As CP-net semantics are completely determined by the preference relation over swaps, one may consider the set $\mathcal{X}^*_{swap} = \{(o, o') \in \mathcal{O} \times \mathcal{O} \mid (o, o') \text{ is a swap}\}$ as the instance space.[4] The size of this instance space is $nm^n(m-1)$: every variable has m^{n-1} different assignments of the other variables and fixing each assignment of these we have $m(m-1)$ instances. For complete CP-nets, this, however, has a lot of redundant instances as if $c((o, o')) = 0$ then we know for certain that $c((o', o)) = 1$. However, in the case of incomplete CP-nets, $c((o, o')) = 0$ does not necessarily mean $c((o', o)) = 1$ as there could be no relation between the two outcomes, i.e., o and o' are incomparable. As we are focusing on the complete case, we consider the instance space $\mathcal{X}_{swap}$ where those redundant instances are removed and thus $|\mathcal{X}_{swap}| = nm^{n-1}\binom{m}{2} = \frac{m^n n(m-1)}{2}$. Thus, for any two swap outcomes o, o', exactly one of the two pairs $(o, o'), (o', o)$ is included here.

[4] As explicated in Section 7, it is very common in the literature to use swap examples for learning CP-nets.

$\mathcal{X}_{swap}$	$(abc, \bar{a}bc)$	$(ab\bar{c}, \bar{a}b\bar{c})$	$(a\bar{b}c, \bar{a}\bar{b}c)$	$(a\bar{b}\bar{c}, \bar{a}\bar{b}\bar{c})$	$(abc, a\bar{b}c)$	$(ab\bar{c}, a\bar{b}\bar{c})$	$(\bar{a}bc, \bar{a}\bar{b}c)$	$(\bar{a}b\bar{c}, \bar{a}\bar{b}\bar{c})$	$(abc, ab\bar{c})$	$(a\bar{b}c, a\bar{b}\bar{c})$	$(\bar{a}bc, \bar{a}b\bar{c})$	$(\bar{a}\bar{b}c, \bar{a}\bar{b}\bar{c})$
c_1	1	1	1	1	1	1	1	1	1	0	1	0
c_2	1	1	1	1	1	1	0	1	1	0	0	0

Figure 4: The concepts c_1 and c_2 represent the CP-nets in Figures 1 and 2, respectively, over $\mathcal{X}_{swap}$.

For $x = (o, o') \in \mathcal{X}_{swap}$, let $V(x)$ denote the swapped variable of x. We refer to the first and second outcomes of an example x as $x.1$ and $x.2$, respectively. We use $x[\Gamma]$ to denote the assignments (in both $x.1$ and $x.2$) of $\Gamma \subseteq V \backslash \{V(x)\}$. Note that $x[\Gamma]$ is guaranteed to be the same in $x.1$ and $x.2$, otherwise x will not form a swap instance.

Let $\mathcal{N}_k$ be the set of all complete acyclic CP-nets with indegree at most k. $\mathcal{N}_k$ serves as a representation class for concepts of the form $c : \mathcal{X}_{swap} \rightarrow \{0, 1\}$. A concept c is representable by $\mathcal{N}_k$ if there is a CP-net $N \in \mathcal{N}_k$ such that $c(x) = 1$ if and only if N entails the pair $(x.1, x.2)$. In other words, a CP-net N with induced order $\succ$ represents a concept c if and only if, for every $x \in \mathcal{X}_{swap}$, the following holds:

$$c(x) = \begin{cases} 1 & \text{if } x.1 \succ x.2 \\ 0 & \text{otherwise} \end{cases}$$

The concept class is then defined as $\mathcal{C}_{ac}^k = \{\, c \mid c \text{ is representable by a CP-net } N \in \mathcal{N}_k\}$, that is, the set of all concepts that are representable by $\mathcal{N}_k$. It is not hard to see that, in the complete case, every concept $c \in \mathcal{C}_{ac}^k$ is representable by exactly one CP-net N. Otherwise, there exists two distinct complete CP-nets N and N' with exactly the same set of swap entailments which is impossible. We, therefore, use c and its representation N interchangeably.

Figure 4 shows two concepts c_1 and c_2 that correspond to the CP-nets shown in Figures 1 and 2, respectively, along with one choice of $\mathcal{X}_{swap}$. It is important to restate the fact that $c(x)$ is actually a dominance relation between $x.1$ and $x.2$, i.e., $c(x)$ is mapped to 1 (resp. to 0) if $x.1 \succ x.2$ (resp. $x.2 \succ x.1$) holds. Thus, we sometimes talk about the value of $c(x)$ in terms of the relation between $x.1$ and $x.2$ ($x.1 \succ x.2$ or $x.2 \succ x.1$).

In the remaining part of this work, we fix $k \in \{0, \ldots, n-1\}$ and consider the class $\mathcal{C}_{ac}^k$ of all acyclic CP-nets whose nodes have an indegree of at most k. A concept c contains a swap pair x iff the corresponding CP-net entails $(x.1, x.2)$. By $size(c)$, we refer to the size of the CP-net represented by c. Lastly, $\mathcal{M}_k = \max\{size(c) \mid c \in \mathcal{C}_{ac}^k\}$ is the maximum number of statements in any concept in $\mathcal{C}_{ac}^k$. It can be verified that $\mathcal{M}_k = (n-k)m^k + \sum_{i=0}^{k-1} m^i$ (see AppendixA for constructing a CP-net with $\mathcal{M}_k$ statements in the binary

Table 2: Summary of complexity results. $\mathcal{M}_k = (n-k)m^k + \sum_{i=0}^{k-1} m^i$; $\mathcal{U}_k$ is defined after Theorem 2.

class	VCD	TD	RTD	SDC	OPT
$\mathcal{C}_{ac}^{k}$	$\geq (m-1)\mathcal{M}_k$	$n(m-1)\mathcal{U}_k$	$(m-1)\mathcal{M}_k$	$(m-1)\mathcal{M}_k$	$\geq \lceil \log(m!) \rceil \mathcal{M}_k$
$\mathcal{C}_{ac}^{n-1}$	$m^n - 1$	$n(m-1)m^{n-1}$	$m^n - 1$	$m^n - 1$	$\geq \lceil \log(m!) \rceil \frac{m^n-1}{m-1}$
$\mathcal{C}_{ac}^{0}$	$(m-1)n$	$(m-1)n$	$(m-1)n$	$(m-1)n$	$\geq \lceil \log(m!) \rceil n$

case).

Table 2 summarizes our complexity results for acyclic CP-nets whose nodes have indegrees bounded by k. The two extreme cases are unbounded acyclic CP-nets ($k = n-1$) and separable CP-nets ($k = 0$).

The most striking observation from our results is that VCD, RTD, and SDC are equal for all values of m in $\mathcal{C}_{ac}^{n-1}$. Further, when $m = 2$ (the most studied case in the literature), we have that TD equals the instance space size $n2^{n-1}$ and in general the ratio of the teaching dimension to the instance space size is $\frac{2}{m}$. A close inspection of the case $m = 2$ shows that $\mathcal{X}_{swap}$ has only n instances that are relevant for $\mathcal{C}_{ac}^{0}$, and $\mathcal{C}_{ac}^{0}$ corresponds to the class of all concepts over these n instances. Thus the values of VCD, TD, RTD, SDC, and OPT are trivially equal to n in this special case.

In the case of online prediction, for $m \leq 11$, $\lceil \log(m!) \rceil$ is known to be the minimum number of comparisons needed to sort m elements [25]. However, for most practical applications, $m \leq 11$ is sufficient and thus our results are still useful for judging the optimality of online learning algorithms. The fact that SDC is asymptotically strictly smaller than OPT shows that actively selecting examples strictly decreases the number of mistakes when m is large.

The remainder of this section is dedicated to proving the statements from Table 2.

3.1 VC Dimension

Our first theorem substantially improves on (and corrects) a result by Koriche and Zanuttini (2010), who present a lower bound on VCD($\mathcal{C}_{ac}^{k}$); their bound is in fact incorrect unless $k \ll n$.

Theorem 1. *VCD($\mathcal{C}_{ac}^{n-1}$) $= m^n - 1$, VCD($\mathcal{C}_{ac}^{0}$) $= (m-1)n$ and VCD($\mathcal{C}_{ac}^{k}$) $\geq (m-1)\mathcal{M}_k$.*

The proof of Theorem 1 relies on decomposing $\mathcal{C}_{ac}^{k}$ as a direct product of concept classes over subsets of $\mathcal{X}_{swap}$.

Definition 6. *Let* $\mathcal{C}_i \subseteq 2^{\mathcal{X}_i}$ *and* $\mathcal{C}_j \subseteq 2^{\mathcal{X}_j}$ *be concept classes with* $\mathcal{X}_i \cap \mathcal{X}_j = \emptyset$. *The concept class* $\mathcal{C}_i \times \mathcal{C}_j \subseteq 2^{\mathcal{X}_i \cup \mathcal{X}_j}$ *is defined by* $\mathcal{C}_i \times \mathcal{C}_j = \{c_i \cup c_j \mid c_i \in \mathcal{C}_i \text{ and } c_j \in \mathcal{C}_j\}$. *For concept classes* $\mathcal{C}_1, \ldots, \mathcal{C}_r$*, we define* $\prod_{i=1}^{r} \mathcal{C}_i = \mathcal{C}_1 \times \cdots \times \mathcal{C}_r = (\cdots((\mathcal{C}_1 \times \mathcal{C}_2) \times \mathcal{C}_3) \times \cdots \times \mathcal{C}_r)$.

It is a well-known and obvious fact that $\mathrm{VCD}(\prod_{i=1}^{t} \mathcal{C}_i) = \sum_{i=1}^{t} \mathrm{VCD}(\mathcal{C}_i)$.

For any $v_i \in V$ and any $\Gamma \subseteq V \setminus \{v_i\}$, we define $\mathcal{C}^{\Gamma}_{\mathrm{CPT}(v_i)}$ to be the concept class consisting of all preference relations corresponding to some $\mathrm{CPT}(v_i)$ where $Pa(v_i) = \Gamma$ and $|\Gamma| \leq k$; here the instance space is the set of all swap pairs x with $V(x) = v_i$. Now, if we fix the context of v_i by fixing an assignment $\gamma \in \mathcal{O}_\Gamma$ of all variables in Γ, we obtain a concept class $\mathcal{C}^{\Gamma}_{\succ_\gamma^{v_i}}$, which corresponds to the set of all preference statements concerning the variable v_i conditioned on the context γ. Its instance space is the set of all swaps x with $V(x) = v_i$ and $x[\Gamma] = \gamma$.

Recall that $V = \{v_1, \ldots, v_n\}$. By S_n we denote the class of all permutations of $\{1, \ldots, n\}$.

Proof of Theorem 1.

Lemma 1 (below) states that

$$\mathcal{C}^k_{ac} = \bigcup_{\sigma \in S_n} \prod_{i=1}^{n} \bigcup_{\Gamma \subseteq \{v_{\sigma(1)}, \ldots, v_{\sigma(i-1)}\}, |\Gamma| \leq k} \prod_{\gamma \in \mathcal{O}_\Gamma} \mathcal{C}^{\Gamma}_{\succ_\gamma^{v_{\sigma(i)}}},$$

which yields the bound

$$\mathrm{VCD}(\mathcal{C}^k_{ac}) \geq \max_{\sigma \in S_n} \sum_{i=1}^{n} \max_{\Gamma \subseteq \{v_{\sigma(1)}, \ldots, v_{\sigma(i-1)}\}, |\Gamma| \leq k} \sum_{\gamma \in \mathcal{O}_\Gamma} \mathrm{VCD}(\mathcal{C}^{\Gamma}_{\succ_\gamma^{v_{\sigma(i)}}}).$$

Using $\mathrm{VCD}(\mathcal{C}^{\Gamma}_{\succ_\gamma^{v_{\sigma(i)}}}) = m - 1$ (see Lemma 2), independent of Γ and γ, one obtains, for any $\sigma \in S_n$,

$$\begin{aligned} \mathrm{VCD}(\mathcal{C}^k_{ac}) &\geq (m-1) \sum_{i=1}^{n} \max_{\Gamma \subseteq \{v_{\sigma(1)}, \ldots, v_{\sigma(i-1)}\}, |\Gamma| \leq k} |\mathcal{O}_\Gamma| \\ &= (m-1) \sum_{i=1}^{n} \max_{\Gamma \subseteq \{v_{\sigma(1)}, \ldots, v_{\sigma(i-1)}\}, |\Gamma| \leq k} m^{|\Gamma|} \\ &= (m-1)\mathcal{M}_k. \end{aligned}$$

It remains to verify $\mathrm{VCD}(\mathcal{C}^k_{ac}) \leq (m-1)\mathcal{M}_k$ for $k \in \{0, n-1\}$.

For $k = 0$, we have $\mathcal{M}_k = n$, so let us consider any set Y of size greater than $(m-1)n$ and argue why Y cannot be shattered by $\mathcal{C}^0_{ac}$. Clearly, there exists a variable v_i that is swapped in at least m instances in Y. In order to shatter these $\geq m$ instances with the same swapped variable, a concept class of CP-nets would need to contain CP-nets in which some variables have non-empty parent sets, which is not the case for $\mathcal{C}^0_{ac}$. Thus Y is not shattered, i.e., $\text{VCD}(\mathcal{C}^k_{ac}) \leq (m-1)\mathcal{M}_k$.

For $k = n-1$, the upper bound $\text{VCD}(\mathcal{C}^k_{ac}) \leq m^n - 1 = (m-1)\mathcal{M}_{n-1}$ follows immediately from an observation made by Booth et al. [26] (their Proposition 3,) which states that any concept class corresponding to a set of transitive and irreflexive relations (such as a class of acyclic CP-nets) over $\{0,1\}^n$ has a VC dimension no larger than $2^n - 1$. It is not hard to see that their argument applies to the non-binary case as well, yielding a VC dimension no larger than $m^n - 1$ for any $m \geq 2$. ∎

Lemma 1. $\mathcal{C}^k_{ac} = \bigcup_{\sigma \in S_n} \prod_{i=1}^{n} \bigcup_{\Gamma \subseteq \{v_{\sigma(1)},\ldots,v_{\sigma(i-1)}\}, |\Gamma| \leq k} \prod_{\gamma \in \mathcal{O}_\Gamma} \mathcal{C}^{\Gamma}_{\succ^{v_{\sigma(i)}}_{\gamma}}$.

Proof. By definition, for $v \in V$ and $\Gamma \subseteq V \setminus \{v\}$, the class $\mathcal{C}^{\Gamma}_{\text{CPT}(v)}$ equals $\prod_{\gamma \in \mathcal{O}_\Gamma} \mathcal{C}^{\Gamma}_{\succ^{v}_{\gamma}}$. (Any concept representing a preference table for v with $Pa(v) = \Gamma$ corresponds to a union of concepts each of which represents a preference statement over D_v conditioned on some context $\gamma \in \mathcal{O}_\Gamma$.)

Any concept corresponds to choosing a set Γ_v of parent variables of size at most k for each variable v, which means $\mathcal{C}^k_{ac} \subseteq \prod_{i=1}^{n} \bigcup_{\Gamma \subseteq V \setminus \{v_i\}, |\Gamma| \leq k} \mathcal{C}^{\Gamma}_{\text{CPT}(v_i)}$. By acyclicity, $v_j \in Pa(v_i)$ implies $v_i \notin Pa(v_j)$, so that for each concept $c \in \mathcal{C}^k_{ac}$ some $\sigma \in S_n$ fulfills

$$c \in \prod_{i=1}^{n} \bigcup_{\Gamma \subseteq \{v_{\sigma(1)},\ldots,v_{\sigma(i-1)}\}, |\Gamma| \leq k} \mathcal{C}^{\Gamma}_{\text{CPT}(v_{\sigma(i)})} .$$

Thus, $\mathcal{C}^k_{ac} \subseteq \bigcup_{\sigma \in S_n} \prod_{i=1}^{n} \bigcup_{\Gamma \subseteq \{v_{\sigma(1)},\ldots,v_{\sigma(i-1)}\}, |\Gamma| \leq k} \prod_{\gamma \in \mathcal{O}_\Gamma} \mathcal{C}^{\Gamma}_{\succ^{v_{\sigma(i)}}_{\gamma}}$.

Similarly, one can argue that every concept in the class on the right hand side represents an acyclic CP-net with parent sets of size at most k. With $\mathcal{C}^{\Gamma}_{\text{CPT}(v_i)} = \prod_{\gamma \in \mathcal{O}_\Gamma} \mathcal{C}^{\Gamma}_{\succ^{v_i}_{\gamma}}$, the statement of the lemma follows. ∎

Lemma 2. $\text{VCD}(\mathcal{C}^{\Gamma}_{\succ^{v_i}_{\gamma}}) = m - 1$ *for any* $v_i \in V$, $\Gamma \subseteq V \setminus \{v_i\}$, *and* $\gamma \in \mathcal{O}_\Gamma$.

Proof. Let $v_i \in V$, $\Gamma \subseteq V \setminus \{v_i\}$, and $\gamma \in \mathcal{O}_\Gamma$. We show that $\mathcal{C}^{\Gamma}_{\succ^{v_i}_{\gamma}}$ shatters some set of size $m-1$, but no set of size m. Note first that, by definition, $\mathcal{C}^{\Gamma}_{\succ^{v_i}_{\gamma}}$ is simply the class of all total orders over the domain D_{v_i} of v_i.

To show $(\mathcal{C}^{\Gamma}_{\succ_\gamma^{v_i}}) \geq m-1$, choose any set of $m-1$ swaps over $\Gamma \cup \{v_i\}$ with fixed context γ, in which the pairs of swapped values in v_i are (v_1^i, v_2^i),..., (v_{m-1}^i, v_m^i).

Fix any set $S \subseteq \mathcal{X}_{swap}$ of m swaps over $\Gamma \cup \{v_i\}$ with fixed context γ. To show that S is not shattered, consider the undirected graph G with vertex set D_{v_i} in which an edge between v_r^i and v_s^i exists iff S contains a swap pair flipping v_r^i to v_s^i or vice versa. G has m vertices and m edges and thus contains a cycle. The directed versions of G correspond to the labellings of S; therefore some labelling ℓ of S corresponds to a cyclic directed version of G, which does not induce a total order over D_{v_i}. Hence the labelling ℓ is not realized by $\mathcal{C}^{\Gamma}_{\succ_\gamma^{v_i}}$, so that S is not shattered by $\mathcal{C}^{\Gamma}_{\succ_\gamma^{v_i}}$. ■

It is worth observing that, for the case $k = n-1$, we can generalize our result on the VC dimension to the class of *all consistent* (acyclic or cyclic) CP-nets. As any consistent CP-net (whether acyclic or cyclic) defines an irreflexive, transitive relation, the result from [26] that we use to upper-bound the VC dimension of $\mathcal{C}_{ac}^{n-1}$ by $m^n - 1$ also applies to the larger class of all consistent unbounded CP-nets, even when allowing incomplete CP-nets (and, in the latter case, even when extending the instance space $\mathcal{X'}_{swap} = \mathcal{X}_{swap} \cup \{(o', o) \mid (o, o') \in \mathcal{X}\}$, so that incomplete CP-nets can be represented). Hence the VC dimension of the class of all consistent CP-nets is at most $m^n - 1$. Our Theorem 1 shows that a subclass of this class already has a VC dimension of $m^n - 1$, so that we obtain the following corollary.

Corollary 1. *Over the instance space $\mathcal{X'}_{swap}$, the VC dimension of the class of complete and incomplete consistent CP-nets with unbounded indegree equals $m^n - 1$.*

Sauer's Lemma [22] then bounds the number of consistent CP-nets from above by

$$\sum_{i=0}^{m^n-1} \binom{|\mathcal{X}_{swap}|}{i}.$$

One possible interpretation of Theorem 1 and Corollary 1 is that acyclic CP-nets, while less expressive, are in some sense as hard to learn as all consistent CP-nets.

To conclude our discussion of the VC dimension, we would like to remark that our results contradict a lower bound on the VC dimension of the class of all complete and incomplete acyclic CP-nets with bounded indegree that was presented in [2]. In AppendixA, we argue why the latter bound is incorrect.

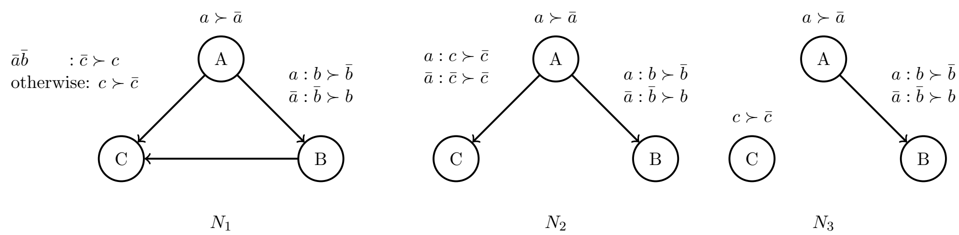


Figure 5: Three networks each of which is subsumed by the ones to its left.

3.2 (Recursive) Teaching Dimension

For studying teaching complexity, it is useful to identify concepts that are "easy to teach." To this end, we use the notion of subsumption [2]: given CP-nets N, N', we say N subsumes N' if for all $v_i \in V$ the following holds: If $y_1 \succ y_2$ is specified in CPT(v_i) in N' for some context γ', then $y_1 \succ y_2$ is specified in CPT(v_i) in N for some context containing γ'. If in addition $N \neq N'$, we say that N strictly subsumes N'.

Now let $\mathcal{C} \subseteq \mathcal{C}^k_{ac}$. A concept $c \in \mathcal{C}$ is maximal in $\mathcal{C}$ if no $c' \in \mathcal{C}$ strictly subsumes c. Note that maximal concepts in $\mathcal{C}^k_{ac}$ are of size $\mathcal{M}_k$.

Example 6. *Let $\mathcal{C}^2_{ac}$ be the class of all unbounded complete acyclic CP-nets over three variables $V = \{A, B, C\}$. Consider the three CP-nets in Figure 5. Clearly, N_3 is subsumed by N_2 and N_1, and N_2 is subsumed by N_1. Further, N_1 is maximal with respect to $\mathcal{C}^2_{ac}$.*

The following two lemmas formalize the intuition that maximal concepts are "easy to teach." Lemma 3 gives an upper bound on the size of a smallest teaching set of any maximal concept, while Lemma 4 implies that a maximal concept is never harder to teach than any concepts it subsumes.

Lemma 3. *For any maximal concept c in a concept class $\mathcal{C} \subseteq \mathcal{C}^k_{ac}$, we have* $\mathrm{TD}(c, \mathcal{C}) \leq (m-1)size(c)$.

Proof. Every statement in the CP-net N represented by c corresponds to an order of m values for some variable v_i under a fixed context γ. For every such order $y_1 \succ^{v_i}_{\gamma} \ldots \succ^{v_i}_{\gamma} y_m$, we include $m-1$ positively labelled swap examples in a set T. For $1 \leq j \leq m-1$, the jth such example labels a pair $x = (x.1, x.2)$ of swap outcomes with $V(x) = v_i$, the projection of x onto $\{v_i\}$ is (y_j, y_{j+1}), and the projection of x onto the remaining variables contains γ. The set T then has cardinality $(m-1)size(c)$ and is obviously consistent with N.

It remains to show that no other CP-net in $\mathcal{C}$ is consistent with T. Suppose some $c' \neq c$ in $\mathcal{C}$ is consistent with T. Since $c' \neq c$, there is some $v_i \in V$, $\gamma \in \mathcal{O}_{V\setminus\{v_i\}}$, and $y, y' \in D_{v_i}$ such that $y \succ_{\gamma}^{v_i} y'$ holds in c' while $y' \succ_{\gamma}^{v_i} y$ holds in c. Thus (i) c' disagrees with some statement in a preference table of c, or (ii) c' has a statement in one of its preference tables that is not contained in c. (i) is impossible since c' is consistent with T, and (ii) is impossible since c is maximal in $\mathcal{C}$. ■

Lemma 4. *Each non-maximal* $c' \in \mathcal{C}_{ac}^k$ *is strictly subsumed by some* $c \in \mathcal{C}_{ac}^k$ *s.t.* $\mathrm{TD}(c', \mathcal{C}_{ac}^k) \geq \mathrm{TD}(c, \mathcal{C}_{ac}^k)$.

Proof. From the graph G' for c', we build a graph G by adding the maximum possible number of edges to a single variable v. As c' is not maximal, it is possible to add at least one edge. The CP-nets corresponding to G and G' differ only in CPT(v). Let c be the concept representing G and z be the size of its CPT for v. A smallest teaching set T' for c' can be modified to a teaching set for c by replacing only those examples that refer to the swapped variable v; $(m-1)z$ examples suffice. To distinguish c' from c, T' must contain at least $(m-1)z$ examples referring to the swapped variable v ($m-1$ for each context in CPT(v) in c). Hence $\mathrm{TD}(c', \mathcal{C}_{ac}^k) \geq \mathrm{TD}(c, \mathcal{C}_{ac}^k)$. ■

Using these lemmas, one can show that $\mathrm{TD}(\mathcal{C}_{ac}^k)$ equals the TD of the class of separable CP-nets within $\mathcal{C}_{ac}^k$ and $\mathrm{RTD}(\mathcal{C}_{ac}^k)$ is the TD of maximal concepts within $\mathcal{C}_{ac}^k$. The latter is at most $(m-1)\mathcal{M}_k$ by Lemma 3, and can be verified to be at least $(m-1)\mathcal{M}_k$ when arguing that a teaching set for a maximal concept must contain $m-1$ examples for each statement in its CPTs, so as to determine the preferences for each context. We thus obtain the following theorem.

Theorem 2. $\mathrm{RTD}(\mathcal{C}_{ac}^k) = (m-1)\mathcal{M}_k$.

To compute $\mathrm{TD}(\mathcal{C}_{ac}^k)$, i.e., the teaching dimension of any separable CP-net with respect to the class $\mathcal{C}_{ac}^k$, consider any unconditional $\mathrm{CPT}(v_i) = \{y_1 \succ \cdots \succ y_m\}$. For every $R \subseteq V\setminus\{v_i\}$, $|R| = k$, we create the dummy CPT(v_i) where $Pa(v_i) = R$ with the same statement $y_1 \succ \cdots \succ y_m$ in every context of $\mathcal{O}_R$. Any teaching set must show that under any context, we have the same statement $y_1 \succ \cdots \succ y_m$.

Thus, a minimal teaching set restricted to CPT(v_i) is a smallest set of examples $U_k[i]$ such that if projected to any subset R of size k, $U_k[i]$ contains m^k contexts. Each of the statements of the form $y_1 \succ \cdots \succ y_m$ can be taught by $(m-1)$ labelled examples, and one does so for each element of $U_k[i]$. For each variable v_i, a respective set of examples is included in the teaching set. Obviously, fewer examples are not sufficient for teaching a separable CP-net. We denote the cardinality of $U_k[i]$, which is independent of i, by $\mathcal{U}_k$. In the binary case, $U_k[i]$ is known as a "$(n-1,k)$-universal set of minimum size" [27, 28].

In combination with some obvious bounds, we obtain the following theorem.

Theorem 3. *For $0 \leq k \leq n-1$, we have $n(m-1)m^k \leq \mathrm{TD}(\mathcal{C}_{ac}^k) = n(m-1)\mathcal{U}_k \leq n(m-1)\binom{n-1}{k}m^k$. If $k=0$, then $\mathcal{U}_k = 1$, so that $\mathrm{TD}(\mathcal{C}_{ac}^0) = (m-1)n$. If $k=1$, then $\mathcal{U}_k = m$, so that $\mathrm{TD}(\mathcal{C}_{ac}^1) = (m-1)mn$. If $k=n-1$, then $\mathcal{U}_k = m^{n-1}$, so that $\mathrm{TD}(\mathcal{C}_{ac}^{n-1}) = (m-1)nm^{n-1}$.*

Theorem 3 implies that, for $\mathcal{C}_{ac}^{n-1}$, the ratio of TD over instance space size $|\mathcal{X}_{swap}|$ is $\frac{2}{m}$. In particular, in the case of binary CP-nets (i.e., when $m=2$), which is the focus of most of the literature on learning CP-nets, the TD equals the instance space size. However, maximal concepts have a TD far below the worst-case TD.

3.2.1 An Example Illustrating Teaching Sets

We will now illustrate our results on the teaching dimension and recursive teaching dimension through examples. Let us consider the case where $V = \{A, B, C\}$, $D_A = \{a, \bar{a}\}$, $D_B = \{b, \bar{b}\}$, and $D_C = \{c, \bar{c}\}$. We consider the class $\mathcal{C}_{ac}^{n-1}$ of all complete acyclic CP-nets defined over V. Figure 5 shows three concepts, N_1, N_2, and N_3, from this class.

In order to illustrate Lemma 3, let us first compute an upper bound on the teaching dimension of the maximal concept N_1. The claim is that $\mathrm{TD}(N_1, \mathcal{C}_{ac}^{n-1})$ is less than or equal to the size of N_1, which is 7. Consider a set of entailments $\mathcal{E}$ corresponding to the teaching set T as described in the proof of Lemma 3. One possibility for $\mathcal{E}$ is the set consisting of the entailments $abc \succ \bar{a}bc$, $abc \succ a\bar{b}c$, $\bar{a}b\bar{c} \prec \bar{a}\bar{b}\bar{c}$, $abc \succ ab\bar{c}$, $a\bar{b}c \succ a\bar{b}\bar{c}$, $\bar{a}bc \succ \bar{a}b\bar{c}$, and $\bar{a}\bar{b}c \prec \bar{a}\bar{b}\bar{c}$. $\mathcal{E}$ is obviously consistent with N_1. A closer look shows that it is also a teaching set for N_1 with respect to $\mathcal{C}_{ac}^{n-1}$. To see why, consider the partition $\mathcal{E} = \mathcal{E}_A \cup \mathcal{E}_B \cup \mathcal{E}_C$, where

- $\mathcal{E}_A = \{abc \succ \bar{a}bc\}$,
- $\mathcal{E}_B = \{abc \succ a\bar{b}c, \bar{a}b\bar{c} \prec \bar{a}\bar{b}\bar{c}\}$, and
- $\mathcal{E}_C = \{abc \succ ab\bar{c}, a\bar{b}c \succ a\bar{b}\bar{c}, \bar{a}bc \succ \bar{a}b\bar{c}, \bar{a}\bar{b}c \prec \bar{a}\bar{b}\bar{c}\}$.

The set $\mathcal{E}_C$ shows that C has at least two parents, which, in our case, have to be A and B. As a result, $Pa(C) = \{A, B\}$ and $\mathrm{CPT}(C)$ is defined precisely:

it is the CPT with the maximum possible parent set $\{A, B\}$ and $\mathcal{E}_C$ contains a statement for every context over this parent set. Similarly, $\mathcal{E}_B$ shows that there must be at least one parent for B. Given the fact that $B \in Pa(C)$ we conclude that $Pa(B) = A$ as there is no other way to explain $\mathcal{E}_C$ and $\mathcal{E}_B$ together. Therefore, we have identified CPT(B) precisely. To this end, there is no way for CPT(A) except to be unconditional (which is consistent with $\mathcal{E}_A$ having one entailment only). Thus, $\mathcal{E}$ is a teaching set for N_1 and $\text{TD}(N_1) \leq size(N_1) = 7$.

Now, let us illustrate Lemma 4. For this purpose, consider the teaching dimension of N_3 which is not maximal w.r.t. $\mathcal{C}_{ac}^{n-1}$. According to Lemma 4, there exists a concept N that subsumes N_3 and with which $\text{TD}(N_3) \geq \text{TD}(N)$. We follow the proof of Lemma 4. If G' is the graph of N_3, we construct a new graph G as follows: G results from G' by selecting the variable C and adding two incoming edges (i.e., the maximum possible number of edges) to the node labeled by this variable. A concept with such graph is N_1. Let $\mathcal{E}$ be a set of entailments corresponding to a teaching set for N_3 with size equal to $\text{TD}(N_3, \mathcal{C}_{ac}^{n-1})$. We claim that the number of entailments in $\mathcal{E}$ whose swapped variable is C has to be greater than or equal 4. To see this, consider the CP-net N with graph G where CPT(A) and CPT(B) are identical to the corresponding conditional preference tables in N_3. Moreover, for CPT(C) in N, for every entailment in $e \in \mathcal{E}$ whose swapped variable is C, a statement $u : c \succ \bar{c}$ or $u : \bar{c} \succ c$ is created in agreement with e, where u provides the values of A and B in e. For instance if the subset of $\mathcal{E}$ referring to swapped variable C is $\{abc \succ ab\bar{c}, \bar{a}bc \succ \bar{a}b\bar{c}\}$, then one possible table for CPT(C) in N is $\{b : c \succ \bar{c}, \bar{b} : \bar{c} \succ c\}$. (Alternatively, the CPT for C in N_1 can also be used.)

Note that $\text{TD}(N_3, \mathcal{C}_{ac}^{n-1}) \leq \text{TD}(\mathcal{C}_{ac}^{n-1})$. By Theorem 3, $\text{TD}(\mathcal{C}_{ac}^{n-1})$ equals $(m-1)nm^{n-1}$ which, for $m = 2$ and $n = 3$ evaluates to 12.

In the sum, we discussed why $\text{TD}(N_1, \mathcal{C}_{ac}^{n-1}) \leq 7$ and $\text{TD}(N_1, \mathcal{C}_{ac}^{n-1}) \leq \text{TD}(N_3, \mathcal{C}_{ac}^{n-1}) \leq 12$. It is actually the case that $\text{TD}(N_1) = 7$, $\text{TD}(N_2) = 9$ and $\text{TD}(N_3) = 10$. Figure 6 shows one possible example of minimum teaching sets for the three networks. In the displayed choice of teaching sets, we selected 10 instances for teaching N_3, use all but one of them (some with flipped labels) to teach N_2 and use seven of them (some with flipped labels) to teach N_1. While this is not the only choice of smallest possible teaching sets for these concepts, it illustrates that some subset of the instances that are used to teach some concept, with appropriate labelling, can also be used to teach a concept that subsumes it. In Figure 6, we show which instances are removed from the teaching set of N_3 by crossing them out. For instance, the entailment $ab\bar{c} \succ \bar{a}b\bar{c}$ is not needed in the teaching set

$abc \succ \bar{a}bc$	$abc \succ \bar{a}bc$	$abc \succ \bar{a}bc$
~~$ab\bar{c} \succ \bar{a}b\bar{c}$~~	~~$ab\bar{c} \succ \bar{a}b\bar{c}$~~	$ab\bar{c} \succ \bar{a}b\bar{c}$
$abc \succ a\bar{b}c$	$abc \succ a\bar{b}c$	$abc \succ a\bar{b}c$
~~$ab\bar{c} \succ a\bar{b}\bar{c}$~~	$ab\bar{c} \succ a\bar{b}\bar{c}$	$ab\bar{c} \succ a\bar{b}\bar{c}$
$\bar{a}b\bar{c} \prec \bar{a}\bar{b}\bar{c}$	$\bar{a}b\bar{c} \prec \bar{a}\bar{b}\bar{c}$	$\bar{a}b\bar{c} \prec \bar{a}\bar{b}\bar{c}$
~~$\bar{a}bc \prec \bar{a}\bar{b}c$~~	$\bar{a}bc \prec \bar{a}\bar{b}c$	$\bar{a}bc \prec \bar{a}\bar{b}c$
$abc \succ ab\bar{c}$	$abc \succ ab\bar{c}$	$abc \succ ab\bar{c}$
$a\bar{b}c \succ a\bar{b}\bar{c}$	$a\bar{b}c \succ a\bar{b}\bar{c}$	$a\bar{b}c \succ a\bar{b}\bar{c}$
$\bar{a}\bar{b}c \prec \bar{a}\bar{b}\bar{c}$	$\bar{a}\bar{b}c \prec \bar{a}\bar{b}\bar{c}$	$\bar{a}\bar{b}c \succ \bar{a}\bar{b}\bar{c}$
$\bar{a}bc \succ \bar{a}b\bar{c}$	$\bar{a}bc \prec \bar{a}b\bar{c}$	$\bar{a}bc \succ \bar{a}b\bar{c}$
N_1	N_2	N_3

Figure 6: Teaching sets for the three networks in Figure 5 w.r.t. the class $\mathcal{C}_{ac}^{n-1}$.

of N_2 as the remaining examples provide enough evidence to conclude that $A \in Pa(B)$ and $A \in Pa(C)$; thus one entailment for CPT(A) suffices for teaching CPT(A).

3.3 Optimal Mistake Bound and Self-Directed Complexity

To conclude our study of information complexity parameters, we determine the self-directed learning complexity as well as a lower bound on the optimal mistake bound for classes of complete acyclic CP-nets.

Theorem 4. $\mathrm{SDC}(\mathcal{C}_{ac}^k) = (m-1)\mathcal{M}_k$.

Proof. From [24] and Theorem 2 we get $\mathrm{SDC}(\mathcal{C}_{ac}^k) \geq \mathrm{RTD}(\mathcal{C}_{ac}^k) = (m-1)\mathcal{M}_k$. For the upper bound, note that any concept in $\mathcal{C}_{ac}^k$, when fixing a variable v and a context $\gamma \in \mathcal{O}_{V\setminus\{v\}}$, induces a total order on D_v. Goldman et al. (1993) discussed a prediction strategy (basically an insertion sort) to learn a total order over m items while making at most $m-1$ mistakes. Separately for each variable v, a self-directed learner fixes an arbitrary context $\gamma \in \mathcal{O}_{V\setminus\{v\}}$ and learns a preference over D_v with Goldman et al.'s strategy. For other contexts on v, the learner will assume the same preference relation unless it makes a mistake (which will cause it to learn a new preference over D_v). For each preference to be learned (at most $\mathcal{M}_k$ in total), the learner makes at most $(m-1)$ mistakes, for a total of $\leq (m-1)\mathcal{M}_k$ mistakes. ■

Theorem 5. $\mathrm{OPT}(\mathcal{C}_{ac}^{k}) \geq \lceil \log(m!) \rceil \mathcal{M}_k$.

Proof. Any learner must identify up to $\mathcal{M}_k$ preference statements (for a fixed variable and context) separately. Each such statement is a permutation of m elements, and its identification requires at least $\lceil \log(m!) \rceil$ comparisons, in the worst case. The adversary can force the learner to make as many mistakes as comparisons are needed, yielding the lower bound.

■

4 Structural Properties of CP-net Classes

The class $\mathcal{C}_{ac}^{n-1}$ is interesting from a structural point of view, as its VC dimension equals its recursive teaching dimension, see Table 2. In general, the VC dimension can exceed the recursive teaching dimension by an arbitrary amount, and it can also be smaller than the recursive teaching dimension [24]. Simon and Zilles [29] posed the question whether the recursive teaching dimension can be upper-bounded by a function that is linear in the VC dimension. So far, the best known upper bound on the recursive teaching dimension is quadratic in the VC dimension [30].

The computational learning theory literature knows of a number of structural properties under which the VC dimension and the recursive teaching dimension coincide. The purpose of this section is to investigate which of these structural properties apply to certain classes of acyclic CP-nets. The main result is that the class $\mathcal{C}_{ac}^{n-1}$ does not satisfy any of the known general structural properties sufficient for VCD and RTD to coincide; therefore this class may serve as an interesting starting point for formulating new general properties of a concept class $\mathcal{C}$ that are sufficient for establishing $\mathrm{RTD}(\mathcal{C}) = \mathrm{VCD}(\mathcal{C})$.

A finite concept class $\mathcal{C}$ is said to be maximum if its cardinality meets the Sauer bound with equality. That is, $|\mathcal{C}| = \sum_{i=0}^{d} \binom{|\mathcal{X}|}{i}$ where $|\mathcal{X}|$ is the instance space size and d is the VC dimension of $\mathcal{C}$. $\mathcal{C}$ is maximal if adding any concept $c \notin \mathcal{C}$ to the class will increase its VC dimension. For a given VC dimension and a given instance space size, maximum classes are the largest possible classes in terms of *cardinality*, while maximal classes are largest with respect to *inclusion*. Every maximum class is also maximal but the converse does not hold [18]. Moreover, $\mathcal{C}$ is said to be an extremal class if $\mathcal{C}$ strongly shatters every set that it shatters. $\mathcal{C}$ strongly shatters $S \subseteq \mathcal{X}$ if there is a subclass $\mathcal{C}'$ of $\mathcal{C}$ that shatters S such that all concepts in $\mathcal{C}'$ agree on the labelling of all instances in $\mathcal{X} \backslash S$. Every maximum class is also an

Table 3: Structural properties of CP-net concept classes.

class	maximum	maximal	intersection-closed	extremal
$\mathcal{C}_{ac}^{0}$ with $m = 2$ (over $\mathcal{X}_{sep}$)	yes	yes	yes	yes
$\mathcal{C}_{ac}^{0}$ with $m > 2$ (over $\mathcal{X}_{sep}$ or $\mathcal{X}_{swap}$)	no	no	no	no
$\mathcal{C}_{ac}^{k}$ with $m \geq 2$, $1 \leq k \leq n-1$	no	no	no	no

extremal class, but not vice versa [31]. A concept class $\mathcal{C}$ is intersection-closed if $c \cap \bar{c} \in \mathcal{C}$ for any two concepts $c, \bar{c} \in \mathcal{C}$.

It was proven that any finite maximum class $\mathcal{C}$ that can be *corner-peeled* by the algorithm proposed by Rubinstein and Rubinstein [23] satisfies $\mathrm{RTD}(\mathcal{C}) = \mathrm{VCD}(\mathcal{C})$ [24]. The same equality holds when $\mathcal{C}$ is of VC dimension 1 or when $\mathcal{C}$ is intersection-closed [24].

Within this section, we assume that $n \geq 2$. The main results of this section are summarized in Table 3. For the proofs of some of these results, the following lemma will be useful.

Lemma 5. *Let $m \geq 2$ and $k \in \{0, \ldots, n-1\}$. If $c \in \mathcal{C}_{ac}^{k}$, then $\bar{c} \in \mathcal{C}_{ac}^{k}$, where $\bar{c}(x) = 1 - c(x)$ for all $x \in \mathcal{X}_{swap}$.*

Proof. The CP-net $\bar{N}$ corresponding to $\bar{c}$ is obtained from the CP-net N corresponding to c by reversing each preference statement in each CPT of N. Obviously, in $\bar{N}$, each variable has the same parent set as in N, so that $\bar{c} \in \mathcal{C}_{ac}^{k}$. ∎

First of all, we discuss separable CP-nets. Note that there is a one-to-one correspondence between separable CP-nets and n-tuples of total orders of $\{1, \ldots, m\}$: since there are no dependencies between the variables, each separable CP-net simply determines an order over the m domain values of a variable, and it does so for each variable independently. This way, for a separable CP-net, the swap example $(v_{i_1}^1 v_{i_2}^2 \ldots v_{i_{l-1}}^{l-1} \alpha v_{i_{l+1}}^{l+1} \ldots v_{i_n}^n, v_{i_1}^1 v_{i_2}^2 \ldots v_{i_{l-1}}^{l-1} \beta v_{i_{l+1}}^{l+1} \ldots v_{i_n}^n)$ will always be labelled exactly the same way as *any* other swap example $x = (x.1, x.2)$ whose swapped variable is v_l and for which the lth positions of $x.1$ and $x.2$ are α and β, respectively. We may therefore consider separable CP-nets over an instance space that is a proper subset of $\mathcal{X}_{swap}$, namely one that contains exactly one swap example for each pair of domain values of each variable. Assuming a fixed choice of such pairs, we denote this subset of $\mathcal{X}_{swap}$ by $\mathcal{X}_{sep}$ and remark that $|\mathcal{X}_{sep}| = \binom{m}{2} n$. Note that, for the class of separable CP-nets, each instance $x \in \mathcal{X}_{swap} \setminus \mathcal{X}_{sep}$ is redundant in the following sense: there exists some instance $x' \in \mathcal{X}_{sep}$ such that

- either $c(x) = c(x')$ for all $c \in \mathcal{C}_{ac}^0$,
- or $c(x) = 1 - c(x')$ for all $c \in \mathcal{C}_{ac}^0$.

It is now easy to see the following for the binary case.

Proposition 1. *Let $m = 2$. Over $\mathcal{X}_{sep}$, the concept class $\mathcal{C}_{ac}^0$ is maximum (in particular, also maximal and extremal) and intersection-closed.*

Proof. Given the instance space $\mathcal{X}_{sep}$, the claim is immediate from the fact that $\text{VCD}(\mathcal{C}_{ac}^0) = (m-1)n = n = \binom{m}{2}n = |\mathcal{X}_{sep}|$, which means that $\mathcal{C}_{ac}^0$ is the class of *all* possible concepts over $\mathcal{X}_{sep}$.

■

In the non-binary case, we will see below that the situation is different. We start by showing that the class of separable CP-nets is not intersection-closed in the non-binary case.

Up to now, a subtlety in the definition of intersection-closedness has been ignored in our discussions. This is best explained using a very simple example. Consider a concept class $\mathcal{C}$ over $\mathcal{X} = \{x_1, x_2\}$ that contains the concepts $\{x_1\}$, $\{x_2\}$, and the empty concept. Obviously, $\mathcal{C}$ is intersection-closed. From a purely learning-theoretic point of view, and certainly for the calculation of any of the information complexity parameters studied above, $\mathcal{C}$ is equivalent to the class $\mathcal{C}' = \{\{x_2\}, \{x_1\}, \{x_1, x_2\}\}$ that results from $\mathcal{C}$ simply when flipping all labels. This class is no longer intersection-closed, as it does not contain the intersection of $\{x_2\}$ and $\{x_1\}$. Likewise, any two concept classes $\mathcal{C}$ and $\mathcal{C}'$ over some instance space $\mathcal{X}$ are equivalent if one is obtained from the other by "inverting" any of its instances, i.e., by selecting any subset $X \subseteq \mathcal{X}$ and replacing $c(x)$ by $1 - c(x)$ for all $c \in \mathcal{C}$ and all $x \in X$.

When defining the instance space $\mathcal{X}_{swap}$, we did not impose any requirements, for any swap pair (o, o'), as to whether (o, o') or (o', o) should be included in $\mathcal{X}_{swap}$. So, in fact, $\mathcal{X}_{swap}$ could be any of a whole class of instance spaces, all of which are equivalent for the purposes of calculating the information complexity parameters we studied. Thus, to show that $\mathcal{C}_{ac}^0$ is not intersection-closed, we have to consider all possible combinations in which the outcome pairs in $\mathcal{X}_{swap}$ could be arranged. In the proof of Proposition 2 this requires a distinction of only two cases, while more cases need to be considered when proving that $\mathcal{C}_{ac}^k$ is not intersection-closed for $k > 0$ (see Proposition 5 below.)

Proposition 2. *Let $m > 2$. Then $\mathcal{C}_{ac}^0$ is not intersection-closed (neither over $\mathcal{X}_{sep}$ nor over $\mathcal{X}_{swap}$).*

Proof. Let $v \in V$ be any variable. Since $m > 2$, the domain of v contains three pairwise distinct values a_1, a_2, and a_3 such that the instance space ($\mathcal{X}_{sep}$ or $\mathcal{X}_{swap}$) contains swap examples x_1, x_2, and x_3, each with the swapped variable v, and one of the following two cases holds:

- Case 1. The projections $x_i[v]$ of the swap pairs x_i to the swapped variable v are

$$x_1[v] = (a_1, a_2),\ x_2[v] = (a_1, a_3),\ x_3[v] = (a_2, a_3)\,.$$

- Case 2. The projections $x_i[v]$ of the swap pairs x_i to the swapped variable v are

$$x_1[v] = (a_1, a_2),\ x_2[v] = (a_3, a_1),\ x_3[v] = (a_2, a_3)\,.$$

It remains to show that, in either case, we can find two separable CP-nets whose intersection (as concepts over the given instance space) is not a separable CP-net.

In Case 1, let c_1 be a CP-net entailing $a_1 \succ a_2$, $a_1 \succ a_3$, and $a_3 \succ a_2$, while c_2 entails $a_2 \succ a_1$, $a_1 \succ a_3$, and $a_2 \succ a_3$. Both these sets of entailments can be realized by separable CP-nets. Both c_1 and c_2 label x_2 with 1 (as they both prefer a_1 over a_3), but they disagree in their labels for x_1 and x_3. The intersection of c_1 and c_2 thus labels x_2 with 1, while it labels both x_1 and x_3 with 0. This corresponds to a preference relation in which a_2 is preferred over a_1, then a_1 is preferred over a_3, but a_3 is preferred over a_2. This cycle cannot be realized by a separable CP-net, i.e., $c_1, c_2 \in \mathcal{C}_{ac}^0$ while $c_1 \cap c_2 \notin \mathcal{C}_{ac}^0$.

In Case 2, let c_1 be a CP-net entailing $a_1 \succ a_2$, $a_3 \succ a_1$, and $a_3 \succ a_2$, while c_2 entails $a_2 \succ a_1$, $a_1 \succ a_3$, and $a_2 \succ a_3$. Both these sets of entailments can be realized by separable CP-nets. The concepts c_1 and c_2 disagree in their labels for all of x_1, x_2, and x_3. The intersection of c_1 and c_2 thus labels all of x_1, x_2, and x_3 with 0. This corresponds to a preference relation in which a_2 is preferred over a_1, then a_1 is preferred over a_3, but a_3 is preferred over a_2. This cycle cannot be realized by a separable CP-net, i.e., $c_1, c_2 \in \mathcal{C}_{ac}^0$ while $c_1 \cap c_2 \notin \mathcal{C}_{ac}^0$. ∎

Further, it turns out that the class of non-binary separable CP-nets is neither maximal nor extremal (and thus not maximum either), independent of whether or not $\mathcal{X}_{sep}$ or $\mathcal{X}_{swap}$ is chosen as the instance space. The proofs of these claims rely on Lemma 5 and establish the same claims for the class $\mathcal{C}_{ac}^k$ for any $k \in \{1, \ldots, n-1\}$ and any $m \geq 2$. Since for $k > 0$, the set

$\mathcal{X}_{swap}$ is the more reasonable instance space, in the remainder of this section we always assume that a concept class is given over $\mathcal{X}_{swap}$. For the class of separable CP-nets though, every proof we provide will go through without modification when $\mathcal{X}_{swap}$ is replaced by$\mathcal{X}_{sep}$.

First, we show that maximality no longer holds for the class of separable CP-nets, when $m > 2$, and neither for $\mathcal{C}_{ac}^k$, when $k > 0$ and $m \geq 2$.

Proposition 3. *Let $m \geq 2$ and $0 \leq k \leq n-1$, where $(m,k) \neq (2,0)$. Then the concept class $\mathcal{C}_{ac}^k$ is not maximal, and, in particular, $\mathcal{C}_{ac}^k$ is not maximum.*

Proof. We need to prove that there exists some concept c over $\mathcal{X}_{swap}$ (not necessarily corresponding to a consistent CP-net) such that $\text{VCD}(\mathcal{C}_{ac}^k \cup \{c\}) = \text{VCD}(\mathcal{C}_{ac}^k)$. We will prove an even stronger statement, namely: for every subset $X \subseteq \mathcal{X}_{swap}$ with $|X| = \text{VCD}(\mathcal{C}_{ac}^k)+1$ and every set $\mathcal{C}$ of concepts such that $\mathcal{C}_{ac}^k \cup \mathcal{C}$ shatters X, we have $|\mathcal{C}| \geq 2$.

Let $X \subseteq \mathcal{X}_{swap}$ with $|X| = \text{VCD}(\mathcal{C}_{ac}^k) + 1$. Such a set X exists, since $\mathcal{X}_{swap}$ is not shattered by $\mathcal{C}_{ac}^k$. Moreover, let $\vec{x} = (x^1, \ldots, x^{|X|})$ be any fixed sequence of all and only the elements in X, without repetitions. Since X is not shattered by $\mathcal{C}_{ac}^k$, there is an assignment $(l_1, \ldots, l_{|X|}) \in \{0,1\}^{|X|}$ of binary values to $\vec{x}$ that is not realized by $\mathcal{C}_{ac}^k$, i.e., there is no concept $c \in \mathcal{C}_{ac}^k$ such that $c(x^i) = l_i$ for all i. Since no concept in $\mathcal{C}_{ac}^k$ realizes the assignment $(l_1, \ldots, l_{|X|})$ on $\vec{x}$, by Lemma 5, no concept in $\mathcal{C}_{ac}^k$ realizes the assignment $(1-l_1, \ldots, 1-l_{|X|})$ on $\vec{x}$. Thus, to shatter X, one would need to add at least two concepts to $\mathcal{C}_{ac}^k$. ■

Second, the following proposition establishes that, under the same conditions as in Proposition 3, the class $\mathcal{C}_{ac}^k$ is not extremal.

Proposition 4. *Let $m \geq 2$ and $0 \leq k \leq n-1$, where $(m,k) \neq (2,0)$. Then the concept class $\mathcal{C}_{ac}^k$ is not extremal.*

Proof. Let $X \subseteq \mathcal{X}_{swap}$ be a set of instances that is shattered by $\mathcal{C}_{ac}^k$, such that $|X| = \text{VCD}(\mathcal{C}_{ac}^k$. Since $\mathcal{X}_{swap}$ is not shattered by $\mathcal{C}_{ac}^k$, we can fix some $\hat{x} \in \mathcal{X}_{swap} \setminus X$. Moreover, let $\vec{x} = (x^1, \ldots, x^{|X|})$ be any fixed sequence of all and only the elements in X, without repetitions.

Suppose $\mathcal{C}_{ac}^k$ were extremal. Then X is strongly shattered, so that, in particular, there exists some $\hat{l} \in \{0,1\}$ such that, for each choice of $(l_1, \ldots, l_{|X|}) \in \{0,1\}^{|X|}$, the labelling $(l_1, \ldots, l_{|X|}, \hat{l})$ of the instance vector $(x^1, \ldots, x^{|X|}, \hat{x})$ is realized by $\mathcal{C}_{ac}^k$. Lemma 5 then implies that, for each choice of $(l_1, \ldots, l_{|X|}) \in \{0,1\}^{|X|}$, the labelling $(1-l_1, \ldots, 1-l_{|X|}, 1-\hat{l})$

of the instance vector $(x^1, \ldots, x^{|X|}, \hat{x})$ is realized by $\mathcal{C}^k_{ac}$. This is equivalent to saying that, for each choice of $(l_1, \ldots, l_{|X|}) \in \{0,1\}^{|X|}$, the labelling $(l_1, \ldots, l_{|X|}, 1 - \hat{l})$ of the instance vector $(x^1, \ldots, x^{|X|}, \hat{x})$ is realized by $\mathcal{C}^k_{ac}$. To sum up, for each choice of $(l_1, \ldots, l_{|X|}) \in \{0,1\}^{|X|}$, both $(l_1, \ldots, l_{|X|}, \hat{l})$ and $(l_1, \ldots, l_{|X|}, 1-\hat{l})$ as labellings of the instance vector $(x^1, \ldots, x^{|X|}, \hat{x})$ are realized by $\mathcal{C}^k_{ac}$. This means that $X \cup \{\hat{x}\}$ is shattered by $\mathcal{C}^k_{ac}$, in contradiction to $|X| = \mathrm{VCD}(\mathcal{C}^k_{ac})$. ■

As a last result of our study of structural properties of CP-net classes, we show that $\mathcal{C}^k_{ac}$ is not intersection-closed, when $k \geq 1$ or when $m \geq 3$.

Proposition 5. *Let $m \geq 2$ and $0 \leq k \leq n-1$, where $(m,k) \neq (2,0)$. Then the concept class $\mathcal{C}^k_{ac}$ is not intersection-closed.*

Proof. Let A and B be two distinct variables. Let a_1 and a_2 be two distinct values in D_A, and b_1 and b_2 two distinct values in D_B. Further, let γ be any fixed context over the remaining variables, i.e., those in $V \setminus \{A, B\}$. We will argue over the possible preferences over outcomes of the form $ab\gamma$, where $a \in \{a_1, a_2\}$ is an assignment to A and $b \in \{b_1, b_2\}$ is an assignment to B.

Without loss of generality, suppose that $\mathcal{X}_{swap}$ contains the instance $(a_1b_1\gamma, a_1b_2\gamma)$ (instead of $(a_1b_2\gamma, a_1b_1\gamma)$.) If that were not the case, one could rename variables and values accordingly to make $\mathcal{X}_{swap}$ contains the instance $(a_1b_1\gamma, a_1b_2\gamma)$.

There are then various cases to consider for the swap pairs representing the comparisons between outcomes of the form $ab\gamma$, where $a \in \{a_1, a_2\}$ and $b \in \{b_1, b_2\}$. For each case, we provide two concepts $c_1, c_2 \in \mathcal{C}^1_{ac}$ for which $c_1 \cap c_2$ is not acyclic and thus is not in $\mathcal{C}^k_{ac}$ for any k.

Case 1. $\mathcal{X}_{swap}$ contains $(a_1b_1\gamma, a_2b_1\gamma)$, $(a_2b_1\gamma, a_2b_2\gamma)$, and $(a_1b_2\gamma, a_2b_2\gamma)$.
Case 2. $\mathcal{X}_{swap}$ contains $(a_1b_1\gamma, a_2b_1\gamma)$, $(a_2b_2\gamma, a_2b_1\gamma)$, and $(a_1b_2\gamma, a_2b_2\gamma)$.
Case 3. $\mathcal{X}_{swap}$ contains $(a_1b_1\gamma, a_2b_1\gamma)$, $(a_2b_1\gamma, a_2b_2\gamma)$, and $(a_2b_2\gamma, a_1b_2\gamma)$.
Case 4. $\mathcal{X}_{swap}$ contains $(a_1b_1\gamma, a_2b_1\gamma)$, $(a_2b_2\gamma, a_2b_1\gamma)$, and $(a_2b_2\gamma, a_1b_2\gamma)$.
Case 5. $\mathcal{X}_{swap}$ contains $(a_2b_1\gamma, a_1b_1\gamma)$, $(a_2b_1\gamma, a_2b_2\gamma)$, and $(a_1b_2\gamma, a_2b_2\gamma)$.
Case 6. $\mathcal{X}_{swap}$ contains $(a_2b_1\gamma, a_1b_1\gamma)$, $(a_2b_2\gamma, a_2b_1\gamma)$, and $(a_1b_2\gamma, a_2b_2\gamma)$.
Case 7. $\mathcal{X}_{swap}$ contains $(a_2b_1\gamma, a_1b_1\gamma)$, $(a_2b_1\gamma, a_2b_2\gamma)$, and $(a_2b_2\gamma, a_1b_2\gamma)$.
Case 8. $\mathcal{X}_{swap}$ contains $(a_2b_1\gamma, a_1b_1\gamma)$, $(a_2b_2\gamma, a_2b_1\gamma)$, and $(a_2b_2\gamma, a_1b_2\gamma)$.

Cases 1 and 7 are discussed in Table 4. A violation of the property of intersection-closedness in Cases 2, 3, and 4 can then be immediately deduced from Case 1 by inverting the binary values in the table for column 3, column

Case 1	1 $(a_1b_1\gamma, a_1b_2\gamma)$	2 $(a_1b_1\gamma, a_2b_1\gamma)$	3 $(a_2b_1\gamma, a_2b_2\gamma)$	4 $(a_1b_2\gamma, a_2b_2\gamma)$	5 CPT(A)	6 CPT(B)
c_1	0	1	1	1	$a_1 \succ a_2$	$a_1 : b_2 \succ b_1$ $a_2 : b_1 \succ b_2$
c_2	1	0	1	1	$b_1 : a_2 \succ a_1$ $b_2 : a_1 \succ a_2$	$b_1 \succ b_2$
$c_1 \cap c_2$	0	0	1	1	$b_1 : a_2 \succ a_1$ $b_2 : a_1 \succ a_2$	$a_1 : b_2 \succ b_1$ $a_2 : b_1 \succ b_2$
Case 7	1 $(a_1b_1\gamma, a_1b_2\gamma)$	2 $(a_2b_1\gamma, a_1b_1\gamma)$	3 $(a_2b_1\gamma, a_2b_2\gamma)$	4 $(a_2b_2\gamma, a_1b_2\gamma)$	5 CPT(A)	6 CPT(B)
c_1	0	1	1	1	$a_2 \succ a_1$	$a_1 : b_2 \succ b_1$ $a_2 : b_1 \succ b_2$
c_2	1	0	1	1	$b_1 : a_1 \succ a_2$ $b_2 : a_2 \succ a_1$	$b_1 \succ b_2$
$c_1 \cap c_2$	0	0	1	1	$b_1 : a_1 \succ a_2$ $b_2 : a_2 \succ a_1$	$a_1 : b_2 \succ b_1$ $a_2 : b_1 \succ b_2$

Table 4: Cases 1 and 7 in the proof of Proposition 5. Columns 1 through 4 provide binary labels stating which of the four swap instances considered are contained in a concept. Column 5 provides the statements in the corresponding CPT for A, while column 6 does the same for B. Concepts c_1 and c_2 belong to $\mathcal{C}^1_{ac}$, but $c_1 \cap c_2$ has a cycle, in which A is a parent of B and vice versa.

4, both columns 3 and 4, respectively. In the same way, Cases 8, 5, and 6 can be handled following Case 7. ■

To conclude, there are no known structure-related theorems in the literature that would imply $\text{VCD}(\mathcal{C}^{n-1}_{ac}) = \text{RTD}(\mathcal{C}^{n-1}_{ac})$. Hence, the latter equation, which we have proven in Section 3, is of interest, as it makes the class of complete unbounded acyclic CP-nets the first "natural" class known in the literature for which VCD and RTD coincide. A deeper study of its structural properties might lead to new insights into the relationship between VCD and RTD and might thus address open problems in the field of computational learning theory [29].

5 Learning from Perfect Membership Queries

In this section, we investigate the problem of learning complete CP-nets from membership queries alone. The reason for choosing membership queries alone is twofold. Firstly, from the cognitive perspective, answering membership queries of the form "*is o better than o′?*" is more intuitive and poses less burden upon the user than comparing a proposed CP-net to the true one (which is the case when answering equivalence queries). Secondly, Koriche and Zanuttini have shown that membership queries are powerful in the sense that CP-nets are not efficiently learnable from equivalence queries alone but they are from equivalence and membership queries [2]. Thus, an immediate question is whether membership queries alone are powerful enough to efficiently learn CP-nets.

The complexity results presented in Section 3 have interesting consequences on learning CP-nets from membership queries alone. In particular, it is known that the query complexity of the optimal membership query algorithm is lower-bounded by the teaching dimension of the class [32]. Therefore, in this section, we propose strategies to learn CP-nets and use the TD results to assess their optimality. In what follows, we show near-optimal query strategies for tree CP-nets and generally for classes of bounded acyclic CP-nets. This is followed by investigating the case of learning CP-nets non-adaptively, that is, fixing the membership queries in advance without adapting them to the answers received.

5.1 Tree CP-nets

Koriche and Zanuttini [2] present an algorithm for learning a binary tree-structured CP-net N^* that may be *incomplete* in that it may have empty CPTs (i.e., it learns a superclass of $\mathcal{C}_{ac}^1$ for $m = 2$.) Their learner uses at most $n_{N^*}+1$ equivalence queries and $4n_{N^*}+e_{N^*}\log(n)$ membership queries, where n_{N^*} is the number of relevant variables and e_{N^*} the number of edges in N^*. We present a method for learning any CP-net in $\mathcal{C}_{ac}^1$ (i.e., *complete* tree CP-nets) for any m, using only membership queries.

For a CP-net N, *a conflict pair* w.r.t. v_i is a pair (x, x') of swaps such that (i) $V(x) = V(x') = v_i$, (ii) $x.1$ and $x'.1$ agree on v_i, (iii) $x.2$ and $x'.2$ agree on v_i, and (iv) N entails one of the swaps x, x', but not the other. If v_i has a conflict pair, then v_i has a parent variable v_j whose values in x and x' are different. Such a variable v_j can be found with $\log(n)$ membership queries by binary search (each query halves the number of candidate variables with different values in x and x') [27].

We use this binary search to learn tree-structured CP-nets from membership queries, by exploiting the following fact: if a variable v_i in a tree CP-net has a parent, then a conflict pair w.r.t. v_i exists and can be detected by asking membership queries to sort m "test sets" for v_i. Let $(v_1^i, \ldots, v_m^i)$ be an arbitrary but fixed permutation of D_{v_i}. Then, for all $j \in \{1, \ldots, m\}$, a test set $I_{i,j}$ for v_i is defined by $I_{i,j} = \{(v_j^1, \ldots, v_j^{i-1}, v_r^i, v_j^{i+1}, \ldots, v_j^n) \mid 1 \leq r \leq m\}$.

Since v_i has no more than one parent, determining preference orders over m such test sets of size m is sufficient for revealing conflict pairs, rather than having to test all possible contexts in $\mathcal{O}_{V\setminus\{v_i\}}$.

Example 7. *Consider the set of variables* $V = \{A, B, C\}$ *where* $D_A = \{a, a', a'', a'''\}$*,* $D_B = \{b, b', b'', b'''\}$*, and* $D_C = \{c, c', c'', c'''\}$*. The following is one possible collection of test sets for the variable A:*

$$I_{A,1} = \{abc, a'bc, a''bc, a'''bc\}$$
$$I_{A,2} = \{ab'c', a'b'c', a''b'c', a'''b'c'\}$$
$$I_{A,3} = \{ab''c'', a'b''c'', a''b''c'', a'''b''c''\}$$
$$I_{A,4} = \{ab'''c''', a'b'''c''', a''b'''c''', a'''b'''c'''\}$$

Clearly, a *complete* target CP-net imposes a *total* order on every $I_{i,j}$, which can be revealed by posing enough membership queries selected from the $\binom{m}{2}$ swaps over $I_{i,j}$; a total of $O(m\log(m))$ comparisons suffice to determine the order over $I_{i,j}$. This yields a simple algorithm for learning tree CP-nets with membership queries:

Algorithm 1. For every variable v_i, determine $Pa(v_i)$ and $\mathrm{CPT}(v_i)$ as follows:

1. For every value $j \in \{1, \ldots, m\}$, ask $O(m\log(m))$ membership queries from the $\binom{m}{2}$ swaps over $I_{i,j}$ to obtain an order over $I_{i,j}$.

2. If for all $j_1, j_2 \in \{1, \ldots, m\}$ the obtained order over I_{i,j_1} imposes the same order on D_{v_i} as the obtained order over I_{i,j_2} does, i.e., there is no conflict pair for v_i, then $Pa(v_i) = \emptyset$. In this case, $\mathrm{CPT}(v_i)$ is fully determined by the queries in Step 1, following the order over D_{v_i} that is imposed by the order over any of the $I_{i,j}$.

3. If there are some $j_1, j_2 \in \{1, \ldots, m\}$ such that the obtained order over I_{i,j_1} imposes a different order on D_{v_i} than the obtained order over I_{i,j_2} does, i.e., there is a conflict pair (x, x') for v_i, then find the only parent of v_i by $\log(n)$ further queries, as described by Damaschke [27]. From these queries, together with the ones posed in Step 1, $\mathrm{CPT}(v_i)$ is fully determined.

The procedure described by Damaschke [27] is a binary search on the set of candidates for the parent variable. Let (x, x') be the conflict pair over variable v_i, as found in Step 3, where

$$\begin{aligned} x &= (a_1 \ldots a_{i-1} a_i a_{i+1} \ldots a_n,\ a_1 \ldots a_{i-1} \overline{a_i} a_{i+1} \ldots a_n)\,, \\ x' &= (a'_1 \ldots a'_{i-1} a_i a'_{i+1} \ldots a'_n,\ a'_1 \ldots a'_{i-1} \overline{a_i} a'_{i+1} \ldots a'_n)\,. \end{aligned}$$

Initially, each variable other than v_i is a potential parent. The set of potential parents is halved recursively by asking membership queries for swaps (o, o') over v_i, with $o(v_i) = a_i$, $o'(v_i) = \bar{a}_i$, and half of the potential parent variables in o and o' having the same values as in x, while the other half of the potential parent variables has values identical to those in x'. (The variables that have been eliminated from the set of potential parent variables will all be assigned the same values as in x.)

Example 8. *Suppose the queries on the test sets revealed a conflict pair (x, x') for the variable v_5, where*

$$\begin{aligned} x &= (a_1a_2a_3a_4a,\ a_1a_2a_3a_4\overline{a}), \\ x' &= (a'_1a'_2a'_3a'_4a,\ a'_1a'_2a'_3a'_4\overline{a}), \end{aligned}$$

and

$$a_1a_2a_3a_4a \succ a_1a_2a_3a_4\overline{a},$$

while

$$a'_1a'_2a'_3a'_4\overline{a} \succ a'_1a'_2a'_3a'_4a.$$

To find the only parent of v_5, Damaschke's procedure will check whether

$$a_1a_2a'_3a'_4a \succ a_1a_2a'_3a'_4\overline{a}.$$

If yes, then either v_1 or v_2 must be the parent of v_5, and one will next check whether

$$a_1a'_2a_3a_4a \succ a_1a'_2a_3a_4\overline{a}.$$

If yes, then v_1 is the parent of v_5, else v_2 is the parent of v_5. If, however, $a_1a_2a'_3a'_4a \prec a_1a_2a'_3a'_4\overline{a}$, then the second query would have been to test whether $a_1a_2a'_3a_4a \succ a_1a_2a'_3a_4\overline{a}$, in order to determine whether the parent of v_5 is v_3 or v_4.

It is not hard to see that our algorithm learns a target CP-net $N^* \in \mathcal{C}^1_{ac}$ with

$$O(nm^2 \log(m) + e_{N^*} \log(n))$$

membership queries, where e_{N^*} is the number of edges in N^*. In particular, for the binary case, it requires on the order of $2n + e_{N^*} \log(n)$ queries at most, i.e., compared to Koriche and Zanuttini's method, when focusing only on tree CP-nets with non-empty CPTs, our method reduces the number of membership queries by a factor of 2, while at the same time dropping equivalence queries altogether.

It is a well-known and trivial fact that the teaching dimension of a concept class $\mathcal{C}$ is a lower bound on the worst-case number of membership queries required for learning concepts in $\mathcal{C}$. We have proven above that $\mathrm{TD}(\mathcal{C}_{ac}^1) = n(m-1)\mathcal{U}_1 = nm(m-1)$. That means that our method uses no more than on the order of $m\log(m) + e_{N^*}\log(n)$ queries more than an optimal one, which means, asymptotically, it uses at most an extra $e_{N^*}\log(n)$ queries when $m = 2$.

By comparison, Koriche and Zanuttini [2] provide an algorithm that learns the class $\mathcal{C}$ of complete *and incomplete* (binary) acyclic CP-nets with nodes of bounded indegree from equivalence and membership queries. To evaluate their algorithm, they compare its query consumption to $\log(4/3)\mathrm{VCD}(\mathcal{C})$, which is a lower bound on the required number of membership and equivalence queries, known from fundamental learning-theoretic studies [33]. In lieu of an exact value for $\mathrm{VCD}(\mathcal{C})$, Koriche and Zanuttini plug in a lower bound on $\mathrm{VCD}(\mathcal{C})$, cf. their Theorem 6. We show in AppendixA that this lower bound is not quite correct; consequently, here we re-assess the query consumption of Koriche and Zanuttini's algorithm.

For any k, their algorithm uses at most $s_{N^*} + e_{N^*}\log(n) + e_{N^*} + 1$ queries in total, for a target CP-net N^* with s_{N^*} statements and e_{N^*} edges. In the worst case $s_{N^*} = \mathcal{M}_k \leq \mathrm{VCD}(\mathcal{C})$ and $e_{N^*} = \binom{k}{2} + (n-k)k$ (i.e., N^* is maximal w.r.t. $\mathcal{C}$). This yields $\mathcal{M}_k + e_{N^*}(\log(n) + 1)$ queries for their algorithm, which exceeds the lower bound $\log(4/3)\mathrm{VCD}(\mathcal{C})$ by at most $\log(3/2)\mathrm{VCD}(\mathcal{C}) + e_{N^*}\log(n)$. This is a more refined assessment compared to the term $e_{N^*}\log(n)$ that they report, and it holds for *any* value of k.

5.2 Bounded Acyclic CP-nets

So far, our arguments for bounded acyclic CP-nets in general have been information-theoretic with no investigation of their query complexity. In this section, we show that the teaching dimension results for $\mathcal{C}_{ac}^k$ (cf. Theorem 3) immediately yield a general strategy for learning acyclic CP-nets from membership queries alone. Recall that $\mathrm{TD}(\mathcal{C}_{ac}^k) = n(m-1)\mathcal{U}_k$, where $\mathcal{U}_k$ is the size of an $(n-1,k)$-universal set $\mathcal{F}$ of minimum size.

Let us first consider the binary case, i.e., $m = 2$. Then, for $k \geq 2$, the quantity $\mathcal{U}_k$ is known to be $\Omega(2^k\log(n-1))$ and $O(k2^k\log(n-1))$ [34]. Thus,

$$\Omega(n2^k\log(n-1)) \ni \mathrm{TD}(\mathcal{C}_{ac}^k) \in O(nk2^k\log(n-1))\,.$$

For simplicity, assume all variables have the same domain $\{0,1\}$; all one would need is an explicit list of all the elements in $\mathcal{F}$. Then, for every variable v_i, one can query the elements of $\mathcal{F}$. Technically, every element of

$\mathcal{F}$ is a context $\gamma \in \mathcal{O}_{V \setminus v_i}$ and one queries the instance x where $V(x) = v_i$ and $x[V \setminus v_i] = \gamma$. For example, assume $n = 4$ and one tries to identify CPT(v_2). If $\gamma = (0, 0, 0)$ is the element of $\mathcal{F}$ assigning values to v_1, v_3, and v_4, then the corresponding instance is $((0, 0, 0, 0),\ (0, 1, 0, 0))$, where the ith value represents the ith variable.

Algorithm 2. For every variable v_i, determine $Pa(v_i)$ and CPT(v_i) as follows:

1. Ask membership queries for all the elements of $\mathcal{F}$.
2. If all of the queried instances for v_i together yield only a single statement, then v_i has no parents. (Since there are at most k parents for v_i and the same statement appears in every context of all potential parent sets of size k, we know that CPT(v_i) is unconditional.) Otherwise, for any $j \neq i$, the variable v_j is included in $Pa(v_i)$, if and only if there exists a conflict pair (x, x') "caused by v_j," i.e., in which v_j is the only variable (other than v_i) for which the value in x is different than the value in x'.
3. From the queries posed in Step 1, CPT(v_i) is fully determined.

The correctness of this algorithm follows by the same arguments that were used to establish Theorem 3. Thus, one can identify any concept $c \in \mathcal{C}^k_{ac}$ with $n\mathcal{U}_k$ membership queries, where $\mathcal{U}_k \in O(nk2^k \log(n-1))$. A universal set of such size was proven to exist by a probabilistic argument with no explicit construction of the set [28]. However, a construction of an $(n-1, k)$-universal set of size $2^k \log(n-1)k^{O(\log k)}$ is reported in the literature [34]. Therefore, one can effectively learn any concept in $\mathcal{C}^k_{ac}$ with a number of queries bounded by $n2^k \log(n-1)k^{O(\log k)}$, which is at most $k^{O(\log k)}$ away from the teaching dimension. Moreover, when $k2^k < \sqrt{n}$, there is an explicit construction of a (n, k)-universal set of size n, e.g., for $n = 800$ this construction is guaranteed to work with an indegree up to 3 and for $n = 1500$ with an indegree up to 4. This can be utilized if one is interested in learning sparse CP-nets over a large number of variables. Note that the same techniques can be applied to the multi-valued case, i.e., when $m > 2$.

6 Learning from Corrupted Membership Queries

So far, we have assumed that all membership queries are answered correctly. This assumption is unrealistic in many settings, especially when it comes to dealing with human experts that may have incomplete knowledge of the target function. For instance, when eliciting CP-nets from users, it could be the case that the user actually does not know which of two given outcomes is to be preferred.

In this section, we consider the situation in which there is a fixed set L of instances x for which the oracle does not provide the true classification $c^*(x)$. The set L is assumed to be chosen in advance by an adversary. There are two ways in which the oracle could deal with queries to elements in L:

- A *limited oracle* returns "I don't know" (denoted by $\perp$) when queried for any $x \in L$, and returns the true label $c^*(x)$ for any $x \notin L$ [35, 36].
- A *malicious oracle* returns the wrong label $1 - c^*(x)$ when queried for any $x \in L$, and returns the true label $c^*(x)$ for any $x \notin L$ [35, 37].

In either case, the oracle is persistent in the sense that it will return the same answer every time the same instance is queried. A concept class $\mathcal{C}$ is exactly learnable with membership queries to a limited (malicious, resp.) oracle if there is an algorithm that exactly identifies any target concept $c^* \in \mathcal{C}$ by asking a limited (malicious, resp.) oracle a number of membership queries that is polynomial in n, the size of c^*, and $|L|$ [35, 36][5].

While we assume that the learner has no prior information on the size or content of L, we will study under which conditions on L learning complete acyclic CP-nets from limited or malicious oracles is possible. We will restrict our analysis to the case of binary CP-nets.

In our analysis, we will make use of the trivial observation that exact learnability with membership queries to a malicious oracle implies exact learnability with membership queries to a limited oracle.

6.1 Limitations on the Corrupted Set

We first establish that learning $\mathcal{C}_{ac}^k$ is impossible, both from malicious and from limited oracles, when $|L| \geq 2^{n-1-k}$.

Proposition 6. *Let $m = 2$ and $1 \leq k \leq n-1$. If $|L| \geq 2^{n-1-k}$, then the class $\mathcal{C}_{ac}^k$ is not exactly learnable with membership queries to a limited oracle and not exactly learnable with membership queries to a malicious oracle.*

Proof. It suffices to prove non-learnability from limited oracles.

Consider a CP-net N in which all the variables are unconditional, except for one variable v_i, which has exactly k parents. Furthermore, let $\mathrm{CPT}(v_i)$

[5] This corresponds to the strict learning model discussed in [35, 36].

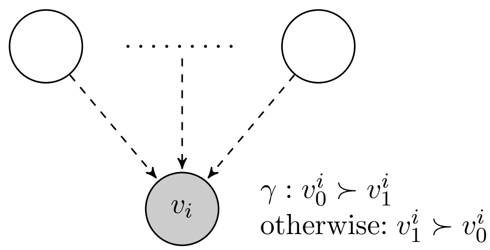


Figure 7: A CP-net $N \in \mathcal{C}_{ac}^k$ that, for $|L| \geq 2^{n-1-k}$, cannot be distinguished from any CP-net N' that differs from N only in CPT(v_i).

impose the same order on the domain of v_i for all contexts over $Pa(v_i)$, except for one context γ over $Pa(v_i)$ for which CPT(v_i) imposes the reverse order over the domain of v_i. Figure 7 shows an example of such a CP-net. The adversary can choose $L = \{x = (x_1, x_2) \in \mathcal{X}_{swap} \mid V(x) = v_i$ and $x_1[Pa(v_i)] = x_2[Pa(v_i)] = \gamma\}$. Then, in communication with a limited membership oracle, the learner will not be able to distinguish N from the CP-net N' that is equivalent to N except for having the orders over the domain of v_i swapped in CPT(v_i).

■

In essence, this negative result is due to the fact that the number of instances supporting a statement in a CPT with k parents is 2^{n-1-k}—having all these corrupted makes learning hopeless. In the remainder of this section, we will study assumptions on the structure of L that will allow a learning algorithm to overcome the corrupted answers from limited or malicious oracles at the expense of a bounded number of additional queries. That means, we will constrain the adversary in its options for selecting L, without directly limiting the size of L. The goal is to be able to obtain the correct answer for any query x made by our algorithms that learn from perfect oracles, simply by taking majority votes over a bounded number of additional "verification queries." If we know that the corrupted oracle will affect only a small subset of these verification queries (small enough so that the majority vote over them is guaranteed to yield the correct label for x,) we can use the same learning procedures as in the perfect oracle case, supplemented by a bounded number of verification queries.

Suppose that, for each $x \in \mathcal{X}_{swap}$, we could efficiently compute a small set VQ(x) $\subseteq \mathcal{X}_{swap}$ such that the limited/malicious oracle would be guaranteed to return the correct label for x on more than half of the queries for elements in VQ(x). In the case of membership queries to a limited oracle, we could then simulate Algorithms 1 or 2 with the following modification:

LIM If a query for $x \in \mathcal{X}_{swap}$ made by Algorithm 1 (or 2) is answered with $\perp$, replace this response by the majority vote of the limited oracle's responses to all queries over the set VQ(x).

In the case of learning from malicious oracles, *every* query made by Algorithms 1 or 2 would have to be supplemented by verification queries:

MAL If a query for $x \in \mathcal{X}_{swap}$ is made by Algorithm 1 (or 2), respond to that query by taking the majority vote over the malicious oracle's responses to all queries over the set VQ(x).

If q is an upper bound on the size of the set VQ(x), for any $x \in \mathcal{X}_{swap}$, the modified algorithms then would need to ask at most qz membership queries, where z is the number of queries asked by Algorithms 1 or 2.

It remains to find a suitable set VQ(x) of verification queries for any swap pair x, so that

- the size of VQ(x) is not too big, and
- it is not too unreasonable an assumption that the corrupted oracle will return the true label for x on the majority of the swap pairs in VQ(x).

To this end, we introduce some notation.

Definition 7. *Let $x = (x_1, x_2) \in \mathcal{X}_{swap}$ and $1 \leq t \leq n-1$. Then we denote by $F^t(x)$ the set of all swap instances $x' = (x'_1, x'_2)$ with the swapped variable $V(x') = V(x)$ and with a Hamming distance of exactly t between x_i and x'_i when restricted to $V \setminus \{V(x)\}$, i.e.*

$$F^t(x) = \{x' \in \mathcal{X}_{swap} \mid V(x) = V(x') = v_s \text{ and } |\{v \in V \backslash \{v_s\} \mid x[\{v\}] \neq x'[\{v\}]\}| \leq t\}\,.$$

There is a relationship between the entailment of an instance x and the entailments of the elements of $F^t(x)$ for any t: Given a preference table CPT(v_i), where v_i has k parents, and given t, it is not hard to see that

- $|F^t(x)| = \binom{n-1}{t}$, and
- the set $F^t(x)$ contains $\binom{n-1-k}{t}$ elements with the same entailment w.r.t. $V(x)$ as in x itself. These are the instances that share the same values in the parent variables of $V(x)$ and, hence, their entailments have to be identical.

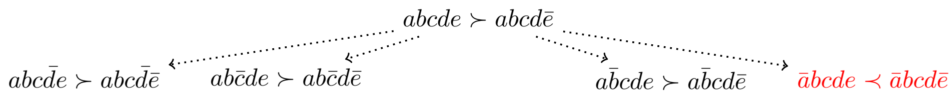


Figure 8: An example of the entailments of an instance x and $F^1(x)$ for $x = (abcde, abcd\bar{e})$ with $Pa(v_5) = \{v_1\}$ and $\text{CPT}(v_5)$ is $\{a : e \succ \bar{e}, \bar{a} : \bar{e} \succ e\}$.

If most of the elements in $F^t(x)$ share the same entailment, they could be used to compensate the oracle corruption. However, queries to elements of $F^t(x)$ again might receive corrupted answers. We will therefore impose a restriction on the overlap between the set L of instances with corrupted responses and any set $F^t(x)$, specifically for $t = 1$.

We would like to constrain L so that querying the elements of $F^1(x)$, and then picking the most frequent answer, yields the true classification of instance x. Figure 8 shows an example of an instance x with $V(x) = v_5$, its entailment and the entailments of the elements of $F^1(x)$ assuming $k = 1$ and $Pa(v_5) = \{v_1\}$.

We then obtain the following learnability result for the case that the indegree k is bounded to be sufficiently small.

Theorem 6. *Suppose $n > 2k + 2$.*

1. *If $|F^1(x) \cap L| \leq n - 2 - 2k$ for every $x \in \mathcal{X}_{swap}$, then the strategy LIM will be successful in interacting with a limited oracle.*

2. *If $|F^1(x) \cap L| \leq \lfloor \frac{n-1}{2} \rfloor - k - 1$ for every $x \in \mathcal{X}_{swap}$, then the strategy MAL will be successful in interacting with a malicious oracle.*

Proof. Note that there are $n-1$ elements in $F^1(x)$, of which at least $n-1-k$ have the same label in the target concept as x and at most k have the opposite label.

First, suppose $|F^1(x) \cap L| \leq n-2-2k$ for every $x \in \mathcal{X}_{swap}$, in the case of a limited oracle. Then, for every $x \in \mathcal{X}_{swap}$, the limited oracle will correctly respond to at least $|F^1(x)| - n + 2 + 2k = 2k + 1$ of the queries for elements in $F^1(x)$. Since at most k of these elements have the opposite label as x, the majority of these queries will return the correct label.

Second, suppose $|F^1(x) \cap L| \leq \lfloor \frac{n-1}{2} \rfloor$ for every $x \in \mathcal{X}_{swap}$, in the case of a malicious oracle. Then, for every $x \in \mathcal{X}_{swap}$, the limited oracle will correctly respond to at least $|F^1(x)| - \lfloor \frac{n-1}{2} \rfloor + k + 1 = \lceil \frac{n-1}{2} \rceil + k + 1$ of the queries for elements in $F^1(x)$. In the worst case, these $\lceil \frac{n-1}{2} \rceil + k + 1$ correctly answered queries contain all of the k elements of $F^1(x)$ that have the opposite label of x. That means that at least $\lceil \frac{n-1}{2} \rceil + 1$ of the $n-1$ queries over $F^1(x)$ (and thus a majority) return the correct label for x. ■

7 Related Work

The problem of learning CP-nets has recently gained a lot of attention [3, 38, 4, 2, 10, 5, 6, 39, 40, 41, 7].

Both in active and in passive learning, a sub-problem to be solved by many natural learning algorithms is the so-called *consistency problem.* This problem is to decide, given a set $S \subseteq \mathcal{O} \times \mathcal{O} \times \{0,1\}$ of labelled examples and a CP-net N, whether or not N is consistent with S, i.e., whether N entails $o \succ o'$ if $(o,o',1) \in S$ and N entails $o \not\succ o'$ if $(o,o',0) \in S$. The consistency problem was shown to be NP-hard even if N is restricted to be an acyclic CP-net whose nodes are of indegree at most k for some fixed $k \geq 2$ and even when, for any $(o,o',b) \in S$, the outcomes o and o' differ in the values of at most two variables [3]. Based on this, Dimopoulos et al. [3] showed that complete acyclic CP-nets with bounded indegree are not efficiently PAC-learnable, i.e., learnable in polynomial time in the PAC model. The authors, however, then showed that such CP-nets are efficiently PAC-learnable from examples that are drawn exclusively from the set of so-called transparent entailments. Specifically, this implied that complete acyclic CP-nets with indegree at most k are efficiently PAC-learnable from swap examples. Michael and Papageorgiou [40] then provided a comprehensive experimental view on the performance of the algorithm proposed in [3]. Their work also proposed an efficient method for checking whether a given entailment is transparent or not.

Lang and Mengin [4] considered the complexity of learning binary separable CP-nets in various learning settings. The literature also includes results on learning CP-nets from noisy examples [5, 39, 7], or from *inconsistent* training sets, which entail that some outcome is preferred over itself [5, 7].

As for active learning, Guerin et al. [6] proposed a heuristic online algorithm that is not limited to swap comparisons. The algorithm assumes the user is able to provide explicit answers of the form $o \succ o'$, $o' \succ o$ or $o \bowtie o'$ to any query (o,o').

To the best of our knowledge, the only studies of learning CP-nets in Angluin's query model are one by Koriche and Zanuttini [2] and one by Labernia et al. [42]. Koriche and Zanuttini assumed perfect oracles and investigated the problem of learning complete and incomplete bounded CP-nets from membership and equivalence queries over the swap instance space. They showed that complete acyclic CP-nets are not learnable from equivalence queries alone but are *attribute-efficiently* learnable from membership and equivalence queries. Attribute-efficiency means that the number of queries required is upper-bounded by a function that is logarithmic in the number of variables. In the case of tree CP-nets, their results hold true even when the equivalence queries may return non-swap examples. The setting considered in their work is more general than ours and exhibits the power of membership queries when it comes to learning CP-nets. Labernia et al. [42] investigated the problem of learning an average CP-net from multiple users using equivalence queries alone. However, neither study addresses the problem of learning complete acyclic CP-nets from membership queries alone, whether corrupted or uncorrupted.

Information complexity parameters of classes of CP-nets have been investigated in only two publications, both of which were concerned mainly with finding the VC dimension value [10, 2]. We discussed our work in relation to theirs already above, see, for example, Section 1.

8 Conclusion

We determined exact values or non-trivial bounds on the parameters VCD, TD, RTD, SDC, and OPT for the classes of complete k-bounded acyclic CP-nets for any k, and used some of the insights gained thereby for the design of algorithms for learning CP-nets from membership queries. The VCD values we determined still apply to the class of potentially incomplete k-bounded acyclic CP-nets, and thus correct a mistake in [2]. Further, we used the calculated TD values in order to show that our proposed algorithm for learning complete tree CP-nets from membership queries alone is close to optimal.

To the best of our knowledge, $\mathcal{C}_{unb}$ is the first known non-maximum (and not intersection-closed) class that is interesting from an application point of view and satisfies RTD = VCD. Thus further studies on the structure of CP-nets may be helpful toward the solution of an open problem concerning the general relationship between RTD and VCD [29].

Our results may also have implications on the study of consistent CP-nets. Since the class of acyclic CP-nets is less expressive than that of all consistent CP-nets, while having the same information complexity in terms of VCD, it would be interesting to find out whether learning algorithms for acyclic CP-nets can be easily adapted to consistent CP-nets in general.

References

[1] C. Boutilier, R. I. Brafman, C. Domshlak, H. H. Hoos, D. Poole, CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements, Journal of Artificial Intelligence Research 21 (2004) 135–191.

[2] F. Koriche, B. Zanuttini, Learning conditional preference networks, Artificial Intelligence 174 (11) (2010) 685–703.

[3] Y. Dimopoulos, L. Michael, F. Athienitou, Ceteris paribus preference elicitation with predictive guarantees, in: IJCAI, 2009, pp. 1890–1895.

[4] J. Lang, J. Mengin, The complexity of learning separable ceteris paribus preferences, in: IJCAI, 2009, pp. 848–853.

[5] J. Liu, Y. Xiong, C. Wu, Z. Yao, W. Liu, Learning conditional preference networks from inconsistent examples, IEEE Transactions on Knowledge and Data Engineering 99 (2012) 1.

[6] J. T. Guerin, T. E. Allen, J. Goldsmith, Learning CP-net preferences online from user queries, in: ADT, 2013, pp. 208–220.

[7] T. E. Allen, C. Siler, J. Goldsmith, Learning tree-structured cp-nets with local search, in: Proceedings of the Thirtieth International Florida Artificial Intelligence Research Society Conference, 2017, pp. 8–13.

[8] D. Angluin, Queries and concept learning, Machine Learning 2 (4) (1988) 319–342.

[9] V. N. Vapnik, A. Y. Chervonenkis, On the uniform convergence of relative frequencies of events to their probabilities, Theory of Probability and its Applications 16 (2) (1971) 264–280.

[10] Y. Chevaleyre, F. Koriche, J. Lang, J. Megine, B. Zanuttini, Learning ordinal preferences on multiattribute domains: the case of CP-nets, in: Preference Learning, Springer-Verlag, 2010, pp. 273–296.

[11] S. A. Goldman, M. J. Kearns, On the complexity of teaching, Journal of Computer and System Sciences 50 (1995) 20–31.

[12] S. Zilles, S. Lange, R. Holte, M. Zinkevich, Models of cooperative teaching and learning, Journal of Machine Learning Research 12 (2011) 349–384.

[13] S. A. Goldman, R. L. Rivest, R. E. Schapire, Learning Binary Relations and Total Orders, SIAM Journal on Computing 22 (5) (1993) 1006–1034.

[14] N. Littlestone, Learning quickly when irrelevant attributes abound: a new linear threshold algorithm, Machine Learning 2 (4) (1988) 245––318.

[15] E. Alanazi, M. Mouhoub, S. Zilles, The complexity of learning acyclic CP-nets, in: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI), 2016, pp. 1361–1367.

[16] J. Goldsmith, J. Lang, M. Truszczynski, N. Wilson, The computational complexity of dominance and consistency in cp-nets, J. Artif. Intell. Res. (JAIR) 33 (2008) 403–432.

[17] D. Bigot, B. Zanuttini, H. Fargier, J. Mengin, Probabilistic conditional preference networks, in: Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence, UAI 2013, Bellevue, WA, USA, August 11-15, 2013, 2013.

[18] M. J. Kearns, U. V. Vazirani, An Introduction to Computational Learning Theory, MIT Press, Cambridge, MA, USA, 1994.

[19] L. G. Valiant, A theory of the learnable, Commun. ACM 27 (11) (1984) 1134–1142.

[20] A. Blumer, A. Ehrenfeucht, D. Haussler, M. K. Warmuth, Learnability and the Vapnik-Chervonenkis dimension, Journal of the ACM 36 (4) (1989) 929–965.

[21] A. Shinohara, S. Miyano, Teachability in computational learning, New Generation Computing 8 (4) (1991) 337–347.

[22] N. Sauer, On the density of families of sets, Journal of Combinatorial Theory, Series A 13 (1) (1972) 145–147.

[23] B. I. P. Rubinstein, J. H. Rubinstein, A geometric approach to sample compression, Journal of Machine Learning Research 13 (2012) 1221–1261.

[24] T. Doliwa, G. Fan, H. U. Simon, S. Zilles, Recursive teaching dimension, VC-dimension and sample compression, Journal of Machine Learning Research 15 (2014) 3107–3131.

[25] N. J. A. Sloane, The on-line encyclopedia of integer sequences, http://oeis.org, sequence A036604.

[26] R. Booth, Y. Chevaleyre, J. Lang, J. Mengin, C. Sombattheera, Learning conditionally lexicographic preference relations, in: ECAI, 2010, pp. 269–274.

[27] P. Damaschke, Adaptive versus nonadaptive attribute-efficient learning, Machine Learning 41 (2) (2000) 197–215.

[28] S. Jukna, Extremal Combinatorics: With Applications in Computer Science, 1st Edition, Springer Publishing Company, Incorporated, 2010.

[29] H. U. Simon, S. Zilles, Open problem: Recursive teaching dimension versus VC dimension, in: COLT, 2015, pp. 1770–1772.

[30] T. L. Lunjia Hu, Ruihan Wu, L. Wang, Quadratic upper bound for recursive teaching dimension of finite VC classes, in: COLT, 2017.

[31] S. Moran, M. K. Warmuth, Labeled compression schemes for extremal classes, CoRR abs/1506.00165.

[32] D. Angluin, Queries revisited, in: N. Abe, R. Khardon, T. Zeugmann (Eds.), Algorithmic Learning Theory, Vol. 2225 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2001, pp. 12–31.

[33] P. Auer, P. Long, Structural results about on-line learning models with and without queries, Machine Learning 36 (3) (1999) 147—-181.

[34] G. Seroussi, N. H. Bshouty, Vector sets for exhaustive testing of logic circuits, IEEE Trans. Information Theory 34 (3) (1988) 513–522.

[35] D. Angluin, M. Krikis, R. H. Sloan, G. Turán, Malicious omissions and errors in answers to membership queries, Machine Learning 28 (2-3) (1997) 211–255.

[36] L. Bisht, N. H. Bshouty, L. Khoury, Learning with errors in answers to membership queries, J. Comput. Syst. Sci. 74 (1) (2008) 2–15.

[37] R. Bennet, N. H. Bshouty, Learning attribute-efficiently with corrupt oracles, Theor. Comput. Sci. 387 (1) (2007) 32–50.

[38] F. Koriche, B. Zanuttini, Learning conditional preference networks with queries, in: IJCAI, 2009, pp. 1930–1935.

[39] J. Liu, Z. Yao, Y. Xiong, W. Liu, C. Wu, Learning conditional preference network from noisy samples using hypothesis testing, Knowledge-Based Systems 40 (2013) 7–16.

[40] L. Michael, E. Papageorgiou, An empirical investigation of ceteris paribus learnability, in: IJCAI, 2013.

[41] D. Bigot, J. Mengin, B. Zanuttini, Learning probabilistic cp-nets from observations of optimal items, in: STAIRS 2014 - Proceedings of the 7th European Starting AI Researcher Symposium, Prague, Czech Republic, August 18-22, 2014, 2014, pp. 81–90.

[42] F. Labernia, F. Yger, B. Mayag, J. Atif, Query-based learning of acyclic conditional preference networks from noisy data, in: DA2PL, 2016.

A Revising a Lower Bound on VCD from [2]

Let $\mathcal{C}_{ac}^{*,k,e}$ be the class of all binary acyclic CP-nets with indegree at most k and at most e edges, where $0 \leq k < n$ and $k \leq e \leq \binom{n}{2}$ (note that this class includes both complete and incomplete CP-nets, whereas our study focused on complete CP-nets.) Koriche and Zanuttini [2, Theorem 6] gave a lower bound on $\mathrm{VCD}(\mathcal{C}_{ac}^{*,k,e})$ over the swap instance space. In particular, setting $u = \lfloor \frac{e}{k} \rfloor$ and $r = \lfloor \log \frac{n-u}{k} \rfloor$, they claimed that $\mathrm{VCD}(\mathcal{C}_{ac}^{*,k,e})$ is lower-bounded by

$$\mathrm{LB} = \begin{cases} 1\,, & \text{if } k = 0\,,\\ u(r+1)\,, & \text{if } k = 1\,,\\ u(2^k + k(r-1) - 1)\,, & \text{if } k > 1\,. \end{cases}$$

We claim that this bound is not generally correct for large values of k and e.

For any given k, it is easy to see that there is a target concept in $\mathcal{C}_{ac}^{*,k,e}$ whose graph has $e_{max} = \binom{k}{2} + (n-k)k$ edges. We can always construct such a graph G as follows: Let V_1 and V_2 be a partition over the n vertices of G where $|V_1| = n-k$ and $|V_2| = k$. Add an edge from each node in V_2 to each node in V_1. This results in $(n-k)k$ edges and G is clearly acyclic with indegree exactly k for every element in V_1. For the remaining $\binom{k}{2}$ edges, let $<$ be any total order on V_2; now the edge (v, w) is added to G if and only if $v < w$.

Next, we evaluate the lower bound LB for $k > 1$ when the target concept has e_{max} edges in its graph:

$$
\begin{aligned}
&u(2^k + k(r-1) - 1) \\
&= \lfloor \frac{e}{k} \rfloor (2^k + k(\lfloor \log(\frac{n - \lfloor \frac{e}{k} \rfloor}{k}) \rfloor - 1) - 1) && \text{(setting } e = e_{max}\text{)} \\
&= \frac{\binom{k}{2} + (n-k)k}{k} (2^k + k(\lfloor \log(\frac{n - \frac{\binom{k}{2} + (n-k)k}{k}}{k}) \rfloor - 1) - 1) \\
&= \frac{1}{2}(2n - k - 1)(2^k + k(\lfloor \log(\frac{k+1}{2k}) \rfloor - 1) - 1) \\
&= \frac{1}{2}(2n - k - 1)(2^k - 2k - 1)
\end{aligned}
$$

When $k = n - 1$, this term evaluates to $n2^{n-2} - n^2 + \frac{n}{2}$ which becomes larger than $2^n - 1$ for $n > 6$. More generally, given $k = n - c$ for some constant $0 < c < n - 1$, the term equals

$$
(n + (c-1))2^{n-(c+1)} - n^2 - (c-1)n + \frac{(2c-1)(n + (c-1))}{2}
$$

which exceeds $2^n - 1$ for small values of c and $n > 6$. Booth et al. [26], however, proved that $2^n - 1$ is an upper bound on the VC dimension of any class of irreflexive transitive relations over $\{0,1\}^n$, and thus also an upper bound on $\mathrm{VCD}(\mathcal{C}_{ac}^{*,k,e})$. Consequently, there must be a mistake in the bound LB. It appears that the mistake was caused by Koriche and Zanuttini assuming that there are acyclic CP-nets with $\frac{e}{k}2^k$ statements, which is not true for large values of k and e.