Computer Science Department Technical Report
University of California
Los Angeles, CA 90024-1596

STRUCTURE IDENTIFICATION IN RELATIONAL DATA

R. Dechter
J. Pearl

March 1992
CSD-920014

In Proceedings of the Canadian AI Conference 1992,
Toronto, Canada, January 1992.

Technical Report
R-172 (short)
March, 1992

Structure Identification in relational data *

Rina Dechter
Information and Computer Science
University of California
Irvine, CA 92717
dechter@ics.uci.edu

Judea Pearl
Cognitive Systems Laboratory
Computer Science Department
University of California
Los Angeles, CA 90024
judea@cs.ucla.edu

Abstract

This paper presents several investigations into the prospects of identifying meaningful structures in empirical data, structures that permit effective organization of the data to meet requirements of future queries. We propose a general framework whereby the notion of identifiability is given a precise formal definition, similar to that of learnability. Using this framework, we then address the problem of expressing a given relation as a k-Horn theory and, if this is impossible, finding a best k-Horn approximation to the given relation.

1 Introduction

Discovering meaningful structures in empirical data has long been regarded as the hallmark of scientific activity. Yet, despite the mystical aura surrounding such discoveries we often find that computational considerations of efficiency and economy play a major role in determining what structures are considered meaningful by scientists. Along this vein, we address the task of finding a computationally attractive description of the data, a description that, both, is economical in storage, and permits future queries to be answered in a tractable way.

Invariably, the existence of such a desirable description rests on whether the dependencies among the data items are decomposable into local, more basic dependencies, possessing some desirable features. A classical example would be to find a finite state machine (with the least number of states) that accounts for observed dependencies among successive symbols in a very long string. In more elaborate settings the dependencies can form a graph (as in the analysis of Markov fields) or a hypergraph (as in relational databases), and the task is to find the topology of these structures. Structure identification includes such tasks as finding effective representations for probability distributions, finding economical decompositions of database schema, finding simple Boolean expressions for truth tables, or finding logical theories that render subsequent processing tractable.

*This work was supported in part by the Air Force Office of Scientific Research, AFOSR 900136 and by NSF grant IRI-9157936.

Despite the generality of the task at hand, very few formal results have been established, and these were primarily confined to probabilistic analysis [Chow and Liu, 1968; Pearl and Verma, 1991]. In this paper we focus on categorical data and categorical descriptions of the data. Given a relation ρ in the form of an explicit listing of the tuples of ρ, we ask whether we can find a more desirable description of ρ, say a constraint network possessing desirable topological features, or a logical theory possessing desirable syntactic features (e.g., Horn theories). The former is treated in a recent report [Dechter and Pearl, 1991] and the latter in section 3. In both cases the desirable features would be such that facilitate efficient query processing routines.

We view this task as an exercise in automatic **identification**, because our main concern will be to recognize cases for which desirable descriptions exist and to identify the parameters of at least one such description. Thus, we explore the existence of a tractable identification procedure that takes data as input, returns a theory and works in time polynomial in the size of the input. Given that the data was generated from a theory that has a desirable structure, our procedure should identify the underlying structure if it is unique, or an equivalent structure in case it is not unique. Conversely, if the data does not lend itself to effective organization, we wish our procedure to acknowledge this fact, so as to save further explorations. An additional requirement is sometimes imposed on the procedure, to identify a "best" approximated theory, in case an exact desirable theory does not exist. We call this latter requirement "strong identifiability".

Our analysis bears close relationships to that of Selman and Kautz [Selman and Kautz, 1991], where theory formation is treated as a task of "knowledge compilation". The main difference between the two approaches is that Selman and Kautz begin with a preformed *theory* in the form of a (reasonably sized) set of clauses, while we start with the bare observations, namely, a (reasonably sized) set of tuples which represent the models of the desired theory. This enables us to easily project the relation onto subsets of variables and solve subtasks which would be intractable had we started with a clausal theory. Another difference is that we require definite determination of whether the theory approximates or describes the data.

This paper is organized as follows: Section 2 introduces a general framework of the identification task. We define weak and strong notions of identifiability and compare them to Valiant's [Valiant, 1984] notion of learnability using familiar examples. Section 3 focuses on identifying Horn theories and shows that k-Horn theories (in which every clause contain at most k literals) can be identified and updated in polynomial time, when k is bounded. All theorems will be stated without proofs, which can be found in [Dechter and Pearl, 1991].

2 Preliminaries and Basic Definitions

2.1 Theories: Networks and Formulas

We denote propositional symbols, also called *variables*, by upper case letters $P, Q, R, X, Y, Z, ...$, propositional literals (i.e. $P, \neg P$) by lower case letters $p, q, r, x, y, z, ...$ and disjunctions of literals, or *clauses*, by $\alpha, \beta, ...$. The complement operator $\sim$ over literals is defined as usual: If $p = \neg Q$, then $\sim p = Q$, If $p = Q$ then $\sim p = \neg Q$. A *formula* in conjunctive normal form (CNF) is a set of clauses, $\varphi = \{\alpha_1, ..., \alpha_t\}$ and it denotes their conjunction. The *models* of a formula φ, $M(\varphi)$, is the set of all satisfying truth assignments to all its symbols. A clause α is *entailed* by φ, written $\varphi \models \alpha$, iff α is true in all models of φ. A clause α is a *prime implicant* of φ iff $\varphi \models \alpha$ and $\nexists \beta \subseteq \alpha$ s.t. $\varphi \models \beta$. A Horn formula is a CNF formula whose clauses all have at most one positive literal. A k-CNF formula is one in which clauses are all of length k or less, and a k-Horn formula is defined accordingly.

Given a clause α we denote by $base(\alpha)$ the set of all propositional symbols on which α is defined. For instance, if $\alpha = \{P \vee \neg Q \vee R\}$ then $base(\alpha) = \{P, Q, R\}$. To characterize the structure of a formula φ we define its *scheme* to be the set of variables on which clauses are defined. Formally:

Definition 1 *(Scheme)*
Let $\varphi = \varphi(x_1, ..., x_n) = \{\alpha_1, ..., \alpha_r\}$, *then*

$$scheme(\varphi) = \{base(\alpha_j) | 1 \leq j \leq r\}. \quad (1)$$

Example 1 *Consider the formula*

$$\varphi = \{(\neg P \vee Q \vee R), (P \vee S), (\neg P \vee \neg S), (\neg P \vee R)\}. \quad (2)$$

In this case,

$$scheme(\varphi) = \{\{P, Q, R\}, \{P, S\}, \{P, R\}\}, \quad (3)$$

We next define the notions of *constraint networks* and *relations* which parallel the notions of formulas and their satisfying models for the case of multi-valued variables. A *relation* associates multi-valued variables with a set of tuples specifying their allowed combinations of values. A *constraint network* is a set of such relations, each defined on a subset of the variables and, together, are taken as conjunction of constraints, namely, they restrict value assignments to comply with each and every constituent relation. The theory of relations was studied extensively in the database literature [Maier, 1983].

Definition 2 *(Relations and Networks)*
Given a set of multi-valued variables $X = \{X_1, ..., X_n\}$, *each associated with a domain of discrete values,* $D_1, ..., D_n$, *respectively, a* **relation** *or a* **constraint** $\rho = \rho(X_1, ..., X_n)$ *is a subset of value assignments to the variables in* X,

$$\rho = \{(x_1, ..., x_n) | (x_1, ..., x_n) \in \{D_1 \times D_2 \times ... \times D_n\}\}. \quad (4)$$

A **constraint network** *over* X, $N(X)$, *consists of a set of such relations* $\rho_1, ..., \rho_t$ *each defined on a subset of variables* $S_1, .., S_t, S_i \subseteq X$. *The set* $S = \{S_1, .., S_t\}$ *is called the* **scheme** *of the constraint network, denoted* $scheme(N)$. *The network* N *represents a unique relation* $rel(N)$ *defined on* X, *which stands for all consistent assignments, namely:*

$$rel(N) = \{\overline{x} = (x_1, ..., x_n) | \forall S_i \in S, \Pi_{S_i}(\overline{x}) \in \rho_i\}. \quad (5)$$

where $\Pi_{S_i}(\overline{x})$ *denotes the projection of* $\overline{x}$ *onto* $S_i \subseteq X$. *If* $rel(N) = \rho$ *we say that* N **describes** ρ.

Clearly, any CNF formula can be viewed as a special kind of constraint network, where the domains are bi-valued, and where the models of each clause specify a constraint on the variables contained in that clause. Accordingly, we say that a bi-valued relation $\rho = \rho(X_1, ..., X_n)$ is **described** (or **represented**) by a formula $\varphi = \varphi(x_1, ..., x_n)$ iff $M(\varphi) = \rho$. We will use the term *theory* to denote either a network or a formula and, correspondingly, use $scheme(T)$ and $M(T)$ (or $rel(T)$).

When considering ways of approximating a relation ρ by a theory T we will examine primarily upper bound approximations, namely, theories T such that $\rho \subseteq M(T)$.

Definition 3 *A theory* $T \in C$ *is said to be a* **tightest** *approximation of* ρ *relative to a class* C *of theories if* $\rho \subseteq M(T)$, *and there is no* $T' \in C$ *such that* $\rho \subseteq M(T') \subset M(T)$.

Example 2 *The following relation:*

P	Q	R	S
1	0	1	0
1	1	1	0
0	1	0	1
0	0	1	1
0	1	1	1
0	0	0	1

can be defined by the network:

P	Q	R
1	0	1
1	1	1
0	1	0
0	0	1
0	1	1
0	0	0

P	S
0	1
1	0

P	R
0	0
0	1
1	1

Being bi-valued, this relation can also be described by the formula:

$$\varphi = \{(\neg P \vee Q \vee R), (P \vee S), (\neg P \vee \neg S), (\neg P \vee R)\}. \quad (6)$$

2.2 Identifiability

We are now ready to give a formal definition of identifiability – a property intrinsic to any class of theories and which governs our ability to decide whether a given

relation ρ has a description within the class or not. As a preliminary and trivial example, we will then show (in subsection 2.4) that the class of k-CNF formulas is identifiable only for $k = 2$, namely, there is no tractable way of deciding whether an arbitrary relation ρ has a description as a k-CNF formula, unless $k = 2$. The class of 2-CNF theories, however, will turn out to be strongly identifiable, namely, not only can we decide the existence of a 2-CNF description, but we can also produce such a description if it exists, or, produce the tightest 2-CNF theory if a precise description does not exists (hence the term "strong".)

To motivate the definition below we should notice that the decisions above depend on what we know apriori about the observed relation ρ. For example, were we given assurance that ρ has a description in k-CNF, it would be easy to produce one such a description. Thus, it is necessary to define the notion of identifiability relative to a background class C' of theories from which ρ is chosen. We will adopt the convention that unless stated otherwise, C' is presumed to be the class of all theories, namely, ρ is arbitrary.

Definition 4 : *(Identifiability)*
A class of theories C is said to be **identifiable** *relative to a background class C', iff:*

1. *(Recognition) For every relation ρ that is describable by some theory T in C', there is an algorithm A, polynomial in $|\rho|$, that determines if ρ has a description in C, and*
2. *(Description) If the answer to (1) is positive, A finds one theory of $T \in C$ that describes ρ, (i,e., $\rho = M(T)$)*
   C is said to be **strongly** *identifiable if, in addition to (1) and (2) above:*
3. *(Tightness) A always finds a theory T_0 in C that is a tightest approximation of ρ.*

By convention, a class in which the recognition or description tasks are NP-hard will be defined as non-identifiable. Note however that the complexity of A is measured relative to the size of ρ, and not relative to the size of its description T. Thus, the notion of identifiability will be applicable to highly constrained theories where the number of distinct observations grows polynomially with the number of variables.

2.3 An example: Identifiability of k-CNF theories.

Let C' be the set of all theories defined on n binary variables. Consider whether the class $C_k \subset C'$ of relations expressible by k-CNF theories is identifiable relative to C'. Although we have algorithms for meeting requirement (3) (and hence (2)) of constructing a tightest k-CNF approximation for any given relation ρ, we do not have an effective way of testing whether this approximation represents the relation ρ exactly, or a superset thereof. Even generating a single model of a tightest k-CNF theory is an NP-hard problem for $k > 2$. We thus conclude that C_k is not identifiable[1].

[1]The non-existence of a tractable procedure for testing exact match with ρ is based on an unpublished theorem conveyed to us by J. Ullman.

Now consider the case where the background class C' is known in advance to consist of k-CNF theories, namely, ρ has a k-CNF description. It is easy then to identify one k-CNF theory which describes ρ. We simply project ρ onto every subset of k or less variables and, for every r-tuple t, $(r \leq k)$ that is not found in the projection of ρ on $X_1, X_2, ..., X_r$ we introduce the clause $(x_1 \vee x_2 \vee ...x_r)$, with x_i being a positive literal iff x_i is false in t. This can be accomplished in time proportional to $|\rho|(2n)^k$.

The theory found can be shown (see section 3) to be a tightest approximation of ρ relative to C_k and, since ρ is assured to have a description in C_k, this theory must be a precise description of ρ. We thus conclude that C_k is strongly identifiable relative to itself.

As a final variant of this example, consider the class of 2-CNF theories. This class is identifiable relative to any arbitrary C', and the reason is as follows: Given an arbitrary relation ρ, we can find a tightest 2-CNF approximation τ of ρ by the projection method described above. There remains to determine whether τ represents ρ precisely. This last task can be accomplished by simply comparing the size of $M(\tau)$ to that of ρ. If the two sizes are equal, τ is obviously a description of ρ, because $M(\tau)$ contains ρ. The distinct feature that renders 2-CNF theories identifiable (unlike k-CNF, $k > 2$) is the tractability of the size-comparison task. A recent result [Dechter and Itai, 1991] states that for every theory T satisfiable in time t, deciding whether $|M(T)| > c$ takes time $O(ct)$. Now , since 2-SAT is satisfiable in polynomial time, testing $M(\tau) > |\rho|$ can also be accomplished in polynomial time.

2.4 Identifiability vs. Learnability

There is a strong resemblance between the notion of identifiability and that of learnability [Valiant, 1984]. If we associate theories with concepts (or functions) and the models of a theory with the learning examples, we see that in both cases we seek a polynomial algorithm that will take in a polynomial number of examples and will produce a concept (or a function) consistent with those examples, from some family of concepts C. Moreover, it is known that in order for a family C to be learnable (with one-sided errors) it must be closed under intersection, and the algorithm must produce the tightest concept in C consistent with the observations [Natarajan, 1987]. This is identical to condition (3) of strong identifiability.

The main difference between the problems described in this paper and those addressed by Valiant's model of learning is that in the latter we are given the concept class C and our task is to identify an individual member of C that is (probably) responsible for the observed instances (in the sense of assuming a small probability of error on the next instance). By contrast, in structure identification we are not given the concept class C. Rather, our objective is to decide whether a fully observed concept ρ, taken from some broad class C' (e.g., all relations) is also a member of a narrower class C

of concepts, one that possesses desirable syntactical features (e.g., 2-CNF, a constraint-tree, or a Horn theory). Thus, the task is not to infer the semantic extension of a concept from a subset of its examples (the entire extension is assumed to be directly observed), but to decide if the concept admits a given syntactical description.

It turns out that deciding whether the tightest approximation exactly describes a given concept, even when the concept is of small size, might require insurmountable computation; a problem not normally addressed in the literature on PAC learning.

The differences between learnability and identifiability can be well demonstrated using our previous example of the class C_k of k-CNF theories. We have established earlier that while C_k is not identifiable relative to the class C' of all relations, it is nevertheless strongly identifiable relative to $C' = C_k$. By comparison the class C_k is known to be polynomially learnable [Valiant, 1984] since, given a collection of instances I of $M(\varphi)$, one can find in polynomial time the tightest k-CNF expression that contains I (see section 3.1). The fact that C_k is not identifiable is not too disturbing in PAC learning tasks, because there we assume that the examples must be drawn from some k-CNF theory, so in the long run, the tightest k-CNF approximation to φ will eventually coincide with the theory from which φ is drawn. However, non-identifiability could be very disturbing if the possibility exists that the examples are taken from a theory outside C_k. In this case the tightest k-CNF theory consistent with the examples might lead to substantial (one-sided) errors.

In general, if we set $C' = C$, then, if C is learnable, it must also be strongly identifiable, because condition (1) is satisfied automatically, and the learnability requirement of zero error on negative examples is equivalent to (3). (Note that since the learner is entitled to observe the entire concept, the PAC requirement of limited error plays no role in identifiability tasks.) However, there are concept classes that are identifiable but not learnable under the condition $C' = C$, a simple example of which is the class of constraint trees. This class is not learnable because it is not closed under intersection, still, it has been shown to be identifiable [Meiri *et al.*, 1990; Dechter, 1990; Dechter and Pearl, 1991]. The same applies to star-structured networks. On the other hand, chains and k-trees are not identifiable [Dechter and Pearl, 1991].

3 Identifying Horn theories

In general, determining whether a given query formula follows from a given CNF formula is intractable. However, when the latter contains only Horn clauses the problem can be solved in linear time [Dowling and Gallier, 1984]. Moreover, experience with logic programming and databases suggests that humans find it natural to communicate knowledge in terms of Horn expressions. Thus, it would be useful to determine whether a given set of observations (the data ρ) can be described as a Horn theory.

The tractability of Horn theories stems not from the topology of the interactions among their clauses but, rather, from the syntactic restriction imposed on each individual clause. However, there are several impediments to the prospects of identifying general Horn theories. First, Selman and Kautz have shown that finding a tightest Horn approximation to a given CNF formula is NP-hard [Selman and Kautz, 1991]. All indications are that starting with a given relation does not make this task any easier. Second, Selman and Kautz also observed that some CNF theories can be converted into Horn expressions only after invoking exponentially many clauses (in the size of the source theory). In such cases it will be futile to use the Horn theory instead of the observations themselves. The more practical question to ask then is whether a given relation can be described as a Horn theory of a reasonable size. To that end, we first analyze the identifiability of k-Horn formulas, namely, Horn formulas in which every clause contains at most k literals, and then extend the results to Horn theories of limited overall size. We start by analyzing general CNF formulas parameterized by their scheme.

3.1 Canonical and Maximal Formulas

Paralleling the multi-valued case, we will first extend the auxiliary notions of *projection network* and *minimal network* to those of *projection formula* and *maximal formula*.

Definition 5 *Let ρ be a bi-valued relation over $X = X_1, ..., X_n$. We define*

$$canonical(\rho) = \{(\sim x_1 \vee \sim x_2 \vee ... \vee \sim x_n) | (x_1, x_2, ..., x_n) \notin \rho\} \quad (7)$$

Example 3 *Let $\rho(P, Q, R) = \{(100), (010), (001)\}$. Then, $canonical(\rho) = \{(\neg P \vee \neg Q \vee R), (P \vee \neg Q \vee \neg R), (\neg P \vee Q \vee \neg R), (P \vee Q \vee R), (\neg P \vee \neg Q \vee \neg R\}$.*

Similarly,

Definition 6 *Given a constraint network $N = \{\rho_1, ..., \rho_t\}$, we define $canonical(N)$ as the formula generated by collecting the canonical formulas of every constituent relation in N. Namely,*

$$canonical(N) = \cup\{canonical(\rho_i) | \rho_i \in N\}. \quad (8)$$

Clearly, $M(canonical(\rho)) = \rho$, and $M(canonical(N)) = rel(N)$.

We are now ready to extend the notion of projection network to a projection formula:

Definition 7 *Given a relation ρ and a scheme S, the* **projection formula** *of ρ w.r.t S, denoted $\Gamma_S(\rho)$, is given by:*

$$\Gamma_S(\rho) = canonical(\Pi_S(\rho)). \quad (9)$$

Theorem 1 *Let F_S be the class of CNF formulas having scheme S. The formula $\Gamma_S(\rho)$ is a tightest approximation of ρ relative to F_S.*

Paralleling the notion of minimal networks in multi-valued relations, we will now show that among all formulas φ in F_S that are equivalent to $\Gamma_S(\rho)$, $\Gamma_S(\rho)$ is **maximal** w.r.t. the partial order $\subseteq$ defined by set inclusion (of clauses). Clearly the class F_S is closed under union. The next theorem proves that among all equivalent formulas in F_S, $\Gamma_S(\rho)$ is the unique maximal formula.

Theorem 2 *Let $\varphi, \tau \in F_S$ and let $\rho = M(\varphi)$, then*

1. $\varphi \approx \tau \Longrightarrow \varphi \cup \tau \approx \varphi$
2. *There exists a unique maximal (w.r.t $\subseteq$) formula μ_S representing ρ given by $\mu_S = \Gamma_S(\rho)$.*

A clause that contains another is clearly redundant, hence we prefer to consider formulas in **reduced** form:

Definition 8 *A formula φ is* **reduced** *if none of its clauses contains another. The formula obtained after eliminating clause subsumption from φ is denoted reduced(φ).*

Theorem 3 *Let μ_S be a maximal formula of some relation, then reduced(μ_S) contains all and only the prime-implicants of μ_S that are restricted to the subsets in S.*

3.2 k-Horn formulas

We now restrict our attention to k-Horn formulas and their identifiability. We will first present a tractable algorithm for generating the maximal tightest k-Horn approximation to a given relation, followed by a tractable test for exactness.

Let S^{*k} denote the set of all subsets of X of size k or less. Our algorithm can be stated as follows: Given a relation ρ on n variables and a constant k, generate the formula $\Gamma_{S^{*k}}(\rho)$ and throw away all non-Horn clauses. We claim that the resulting Horn theory is a tightest k-Horn approximation of ρ. Since, as we will show, this is also the longest form of the tightest approximation, we then generate its equivalent reduced version. To test if the resulting Horn theory represents ρ exactly, we enumerate its models and test that no one lies outside ρ. A formal justification to this process is given in the following paragraphs.

Given a formula φ, we denote by $Horn(\varphi)$ the formula resulting from eliminating all non-Horn clauses from φ.

Theorem 4 *Let ρ be an n-ary bi-valued relation, k a constant, $\pi = \Gamma_{S^{*k}}(\rho)$ and $\eta = Horn(\pi)$. Let H_k be the family of k-Horn formulas, then,*

1. *η is a tightest k-Horn approximation of π.*
2. *η is maximal w.r.t. H_k.*
3. *Both η and reduced(η) are tightest k-Horn approximations of ρ.*
4. *if $M(\eta) \supset \rho$, no k-Horn formula describes ρ.*
5. *reduced(η) equals the set of all k-Horn prime-implicants of η.*

Theorem 4 implies that the algorithm given below which generates the formula $reduced(Horn(\Gamma_{S^{*k}}(\rho)))$, is guaranteed to return a tightest k-Horn approximation of ρ. The algorithm also returns a statement as to whether the formula found is an exact representation of ρ.

Algorithm Horn-generation(ρ, k)
Input: a relation $\rho(X_1, ..., X_n)$ and an integer k.
Output: A k-Horn formula describing ρ or a k-Horn tightest approximation of ρ.

1. begin
2. generate $\pi \Longleftarrow \Gamma_{S^{*k}}(\rho)$ (by projecting ρ on all subsets and performing the canonical transformation)
3. Let $\mu \Longleftarrow Horn(\pi)$ (by eliminating all non-Horn clauses from π).
4. $\eta \Longleftarrow reduced(\mu)$. (by eliminating subsumptions)
5. Sequentially enumerate the models of η, $\{m_1, m_2, ...\}$, using the method in [Dechter and Itai, 1991], and
   - If for some $i \leq |\rho|$, $m_i \notin \rho$, or if $M(\eta)$ contains more than $|\rho|$ elements, then return: "η is a tightest k-Horn approximation." else, return: "η describes ρ".
6. end.

In [Dechter and Itai, 1991] we showed that the models of a Horn formula can be enumerated in time linear in the number of models and the size of the formula. However, in the above algorithm we do not need to compute more then $|\rho|$ models, thus this computation is bounded by $|\rho|$.

To summarize:

Theorem 5 *Algorithm Horn-generation provides a tightest k-Horn approximation of an arbitrary relation ρ. Moreover, this approximation equals the k-Horn prime-implicants of ρ.* □

Example 4 *Consider again the relation*

P	Q	R
1	0	0
0	1	0
0	0	1

and let $k = 2$. We have

$$\Pi_{S^{*2}}(\rho) = \begin{array}{cc} P & Q \\ \hline 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{array} \quad \begin{array}{cc} P & R \\ \hline 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{array} \quad \begin{array}{cc} R & Q \\ \hline 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{array}$$

and $P = \{0,1\}$, $Q = \{0,1\}$, $R = \{0,1\}$. When applying the canonical transformation to each of these relations we get the (already reduced) formula:

$$\Gamma_{S^{*2}}(\rho) = \{(\neg P \vee \neg Q), (\neg P \vee \neg R), ((\neg R \vee \neg Q)\}.$$

Since this is a Horn formula we do not throw clauses away. Computing the number of models of this theory yields 4 models (there is an additional (0,0,0) tuple), thus we conclude that the formula is a tightest 2-Horn approximation of ρ, and that ρ is not 2-Horn identifiable. If we generate the 3-Horn approximation for ρ we get the same formula. (The reason being that in this case, the 2-Horn approximation already contains all its Horn-prime implicants.) Going through the Horn-generation *algorithm, step 2 yields:*

$$\Gamma_{S^{*3}}(\rho) = \{(\neg P \vee \neg Q \vee R), (P \vee \neg Q \vee \neg R), (\neg P \vee Q \vee \neg R), (P \vee Q \vee R), (\neg P \vee \neg Q \neg R), (\neg P \vee \neg Q), (\neg P \vee \neg R), \neg R \vee \neg Q)\}.$$

Step 3 eliminates the only non-Horn clause: $(P \vee Q \vee R)$ and the result of further eliminating subsumptions is the same formula:

$$Horn(reduced(\Gamma_{S^{*3}}(\rho))) = \{(\neg P \vee \neg Q), (\neg P \vee \neg R), \neg R \vee \neg Q)\}. \quad (10)$$

This suggests an *anytime* variation of our algorithm. Instead of applying the algorithm to all subsets of size k, we first apply the algorithm to subsets of size 2, then add the result of processing subsets of size 3, and so on, until we get a satisfying approximation. The next theorem assesses the complexity of our transformation and the size of its resulting Horn theory.

Theorem 6 *(complexity)*

1. *The length (number of clauses) of* $reduced(Horn(\Gamma_{S^*k}(\rho)))$ *is* $O(kn^{k+1})$.
2. *The complexity of* Horn-generation(ρ, k) *is* $O(n^k((k+1)|\rho| + 2^k))$.

Another important variant of the method described above is its *on-line* version, which is useful for stream processing. Assume the tuples of ρ are not available all at once, but are obtained sequentially as a stream of observations, normally containing many repetitions. In this case it might be advantageous to store a parsimonious theory of past data, rather than the data itself, and to update the theory incrementally whenever an observation arrives that contradicts the theory.

Assume we are given a theory h which is a tightest k-Horn approximation of all past data, ρ, and a new tuple t arrives that contradicts h. In principle, updating h requires finding a tightest k-Horn theory that agrees with $\rho \cup \{t\}$. but, since ρ is no longer available, the best we can do is to find a tightest k-Horn approximation of $M(h) \cup \{t\}$. Fortunately, since H_k is closed under intersection, we are guranteed that the two approximations are equivalent, namely, no information is lost by storing h instead of the exact stream of past observations.

The next theorem states that updating h can be done in polynomial time. Although each update may, in the worst case require as many as $O(n^{k+1})$ steps, it is nevertheless polynomial, and is more efficient than approximating $\rho \cup \{t\}$ from scratch when the size of ρ is exponential in n.

Theorem 7 *Incremental updating of best k-Horn approximations takes* $O(n^{k+1})$ *steps per update.*

Clearly, the facility for incremental on-line updating would be useful only when the size of ρ is the main factor that limits our ability to find a useful description of the data.

3.3 Extensions to general Horn formulas

A recent algorithm by [Angluin *et al.*, 1990] permits us to extend the results of the last section to the identification of Horn theories of size $q(n)$, for any fixed polynomial q.[2] The algorithm of Angluin et al. exactly learns Horn theories from equivalence queries and membership queries. An equivalence query is a conjectured Horn theory, and the response by the teacher is a counterexample to the correctness of the conjectured Horn theory (i.e. an assignment that satisfies the correct theory but not the conjectured theory, or vice versa). In the case that there are no counterexamples, then the learning algorithm has succeeded in identifying the correct theory. Membership queries allow the algorithm to ask if a given assignment satisfies the target (i.e., correct) Horn theory, and it is answered yes or no by the teacher.

To be able to answer equivalence queries in polynomial time, Angluin et al. assumed that the target theory is Horn (in general, testing equivalence of two given theories is intractable). If we are given a relation ρ, then we can answer equivalence queries and provide counterexamples in polynomial time even when ρ is non-Horn. Given a conjectured Horn theory H, we first check that every tuple of ρ satisfies H. If not, we return the unsatisfying tuple as a counterexample. Otherwise, $M(H)$ contains ρ, and we then determine whether or not $M(H) = \rho$ by the polynomial enumeration method of [Dechter and Itai, 1991].

Thus, since we can polynomially answer the two basic queries of Angluin's learning algorithm, the algorithm must output an exact Horn representation of ρ if one exists. To determine whether or not one exists of size at most $q(n)$, we can run the algorithm for $t(n, q(n)$ time, where $t(n, k)$ is the time needed by the algorithm to exactly learn a Horn theory of size k over n variables. If the algorithm succeeds in exactly learning ρ within $t(n, q(n))$ time, then there clearly is a Horn theory for ρ of size at most $q(n)$. Otherwise, there is not. Of course, in this case the algorithm does not supply a tightest Horn approximation, and the strong identifiability of $q(n)$-Horn theories remains an open problem.

4 Conclusions

This paper summarizes several investigations into the prospects of identifying meaningful structures in empirical data. The central theme is to identify a computationally attractive description, in cases where the observed data possess such a description and a best approximate description otherwise. This feasibility of performing this task in reasonable time has been given a formal definition through the notion of identifiability, which is normally weaker (if $C' = C$) than that of learnability.

In a related paper [Dechter and Pearl, 1991] we have explored more generally the decomposition of data into a given scheme of smaller relations, as illustrated in Section 2.3. It can be shown that, whereas a best approximation can be found, it is only in cases where the scheme is intrinsically tractable (e.g., 2-CFN) that we can (tractably) decide if the resulting approximation constitutes an exact representation of the data. The decomposition of data into a structure taken from a class of schemes turned out to be a harder task, intractable even in cases where each individual member of the class is tractable. The class of tree-structured schemes is an exception. Here it was shown that an effective procedure exists for determining whether a given relation is decomposable into a tree of binary relations and, if the answer is positive, identifying the topology of such a tree. The procedure runs in time proportional to the size of the relation, but it is still an open question whether it provides the best tree-structured approximation in case the answer is negative.

Focusing on bi-valued data, this paper has explored the identification of descriptions whose tractability stems

[2] This possibility was brought to our attention by an anonymous reviewer of [Dechter and Pearl, 1991].

from syntactical rather than structural features. In particular, we showed that k-Horn theories can be identified in polynomial time, when k is bounded. Finally, the paper presents both any-time and on-line algorithms for identifying Horn theories.

An important issue that was not dealt in this paper is assessing the goodness of the approximations provided by k-Horn theories. Another question is the feasibility of constructing both an upper bound and a lower bound approximations of ρ, in the manner discussed in [Selman and Kautz, 1991] and also in [Dechter, 1990]. Finally, we should mention that the methods presented in this paper will also handle partial observations, namely, observations of truncated tuples of ρ.

5 Acknowledgments

We thank Itay Meiri and Amir Weinshtain for useful discussions.

References

[Angluin *et al.*, 1990] D. Angluin, M. Frazier, and L. Pitt. Leaving conjunctions of horn clauses. In *Proceedings of the 31st Annual Symposium on Foundation of Computer Science, Volume I*, St. Louis, MS, October 1990. IEEE Computer Society Press.

[Chow and Liu, 1968] C. K. Chow and C. N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Trans. on Info. Theory*, IT-(14):462–467, 1968.

[Dechter and Itai, 1991] R. Dechter and A. Itai. The complexity of finding all solutions. Technical report, University of California at Irvine, 1991.

[Dechter and Pearl, 1991] R. Dechter and J. Pearl. Structure identification in relational data. Technical Report R-172, Cognitive Systems Laboratory, UCLA, 1991. Forthcoming *Artificial Intelligence*.

[Dechter, 1990] R. Dechter. Decomposing a relation into a tree of binary relations. *Journal of Computer and System Sciences*, 41(Special Issue on the Theory of Relational Databases):2–24, 1990.

[Dowling and Gallier, 1984] W. F. Dowling and J. H. Gallier. Linear time algorithms for testing the satisfiability of propositional horn formula. *Journal of Logic Programming*, 3:267–284, 1984.

[Maier, 1983] D. Maier. *The Theory of Relational Databases*. Computer Science Press, Rockville, Maryland, 1983.

[Meiri *et al.*, 1990] I. Meiri, R. Dechter, and J. Pearl. Tree decomposition with applications to constraint processing. In *Proceedings of the the American Association of Artificial Intelligence (AAAI-90)*, pages 10–16, Boston, MA, 1990.

[Natarajan, 1987] B.K. Natarajan. On learning boolean functions. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computation*, pages 296–304, 1987.

[Pearl and Verma, 1991] J. Pearl and T. Verma. A theory of inferred causation. In J. A. Allen, R. Fikes, and E. Sandewall, editors, *In Principles of Knowledge Representation and Reasoning: Proceedings of the Second International Conference*, pages 441–452, San Mateo, CA, 1991. Morgan Kaufmann.

[Selman and Kautz, 1991] B. Selman and H. Kautz. Knowledge compilation using horn approximation. In *In Proceedings of AAAI-91*, Aneheim, CA, 1991.

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