A unified framework for structure identification

Bruno Zanuttini, Jean-Jacques Hébrard

A unified framework for structure identification

Bruno Zanuttini and Jean-Jacques Hébrard

Département d'Informatique, Université de Caen, F 14032 Caen Cedex France

Abstract

We propose a general framework for structure identification, as defined by Dechter and Pearl. It is based on the notion of prime implicate, and handles Horn, bijunctive and affine, as well as Horn-renamable formulas, for which, to our knowledge, no polynomial algorithm has been proposed before. This framework, although quite general, gives good complexity results, and in particular we get for Horn formulas the same running time and better output size than the algorithms previously known.

Key words: Combinatorial problems. Structure identification. Prime implicate. Horn-renamable. Affine.

1 Introduction

The problem of structure identification in relational data was formalized by Dechter and Pearl [3]. We focus here on bivalued (boolean) data. The problem consists in finding, if one exists, a propositional formula with predetermined properties (e.g., being Horn), and admitting a given set of models. The main motivation comes from Artificial Intelligence ; indeed, a truth assignment can be seen as an *observation* of the world, and a variable as a *property*. Thus finding a formula admitting given observations as its models can be seen as a *knowledge compilation* task, into a smaller representation, or one that allows efficient reasoning, updating, etc.

We will therefore focus on the identification of classes of formulas that satisfy these requirements, and particularly classes for which the satisfiability problem can be solved in polynomial time. Kavvadias and Sideri [6] show

Email address: {zanutti,hebrard}@info.unicaen.fr (Bruno Zanuttini and Jean-Jacques Hébrard).

that Schaefer's dichotomy theorem for the generalized satisfiability problem [9] extends in some sense to the identification problem : deciding whether given observations are the models of at least one formula in a class $\mathcal{C}$ is either polynomial or CoNP-complete, where $\mathcal{C}$ is defined by fixed constraints. The polynomial classes are the same as for the generalized satisfiability problem : Horn, bijunctive, reversed-Horn and affine. Moreover, Dechter and Pearl [3] give algorithms for computing a formula when a Horn or bijunctive one exists ; the first finds in time $O(|R|^2n^2)$ a Horn formula with $O(|R|n^2)$ clauses, where $|R|$ is the number of observations and n the number of variables, and the second gives in time $O(|R|n^2)$ a bijunctive formula with $O(n^2)$ clauses. The reversed-Horn case is similar to the Horn case, and linear algebra gives a polynomial algorithm for affine formulas.

We introduce here a new framework for the identification problem (section 4), based on the notion of *prime implicate*. This framework handles Horn and bijunctive formulas, with a better output size for Horn formulas than the algorithms previously known ($O(|R|n)$ clauses in time $O(|R|^2n^2)$). It also handles affine formulas (section 5), without needing the usual tools of linear algebra : we show how prime implicates allow to link the conjunctive normal forms and the algebraic representations of these formulas. Finally, our framework handles Horn-renamable formulas (section 5), for which, to our knowledge, no polynomial algorithm has been proposed before. Beyond these results, our approach gives a unified procedure for all these classes.

2 Preliminaries

We assume a countable number of propositional variables $x_1, x_2, \ldots$. We call x_i a *positive literal*, and $\neg x_i$ a *negative literal*. A *clause* is a disjunction of literals in which each variable appears at most once, and a formula is in *conjunctive normal form* (CNF) if it is written as a conjunction of clauses. For instance, the formula $\psi = (x_1 \vee \neg x_2) \wedge (x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_4)$ is in CNF. A vector $m \in \{0,1\}^n$ is a *model* of a CNF ψ (denoted by $m \models \psi$) if m satifies at least one literal t in each clause C of ψ (we say m satisfies C *via* t). A *relation* on $\{0,1\}$ is a subset of $\{0,1\}^n$. A CNF ψ *describes* a relation R if R is exactly the set of models of ψ. For $m \in \{0,1\}^n$, we write $m[i]$ for its ith component, and the number of vectors in R is denoted by $|R|$. For $m, m' \in \{0,1\}^n$, $m < m'$ and $m \leq m'$ refer to the lexicographic order. Throughout the paper, R stands for a fixed nonempty relation on $\{0,1\}$.

A CNF is *Horn* if each of its clauses contains at most one positive literal (e.g., ψ above is Horn), and *bijunctive* if each of its clauses contains at most two literals (ψ is *not* bijunctive). We denote by HORN (resp. BIJUNCTIVE) the class of Horn (resp. bijunctive) formulas. A relation is said to be Horn (resp.

bijunctive) if it has at least one Horn (resp. bijunctive) description. We now define the problem IDENTIF[$\mathcal{C}$] for a class $\mathcal{C}$ of propositional formulas.

Problem IDENTIF[$\mathcal{C}$]
Input : A relation R
Output : 'No' if R has no description in $\mathcal{C}$, otherwise some CNF $\psi \in \mathcal{C}$ describing R.

We are interested in finding efficient algorithms for IDENTIF[$\mathcal{C}$] for different classes $\mathcal{C}$: those of Horn, bijunctive, affine and Horn-renamable formulas.

3 A polynomial algorithm for description in CNF

As a preliminar step, we consider the problem of computing in polynomial time a CNF ψ describing R ; the rest of our work is indeed based on an efficient algorithm for this problem. Note that such a CNF always exists (see for instance [3]), but computing the classical canonical one requires computing a clause for each vector not in R, and thus is not polynomial. The rest of this section introduces a polynomial solution to this problem.

For $m \in R$, let $p_0(m)$ (resp. $p_1(m)$) be the length of the longest common prefix of m and its predecessor (resp. successor) $m' \in R$ in the lexicographic order, or -1 if m' does not exist. We define the clause $C(m,i)$ for all $m \in R$ and $i = 1, \ldots, n$ in one of the two cases below, where $\ell_j = x_j$ if $m[j] = 0$ and $\ell_j = \neg x_j$ if $m[j] = 1$:

- For $i > p_0(m) + 1$ and $m[i] = 1$, $C(m,i) = \ell_1 \vee \ldots \vee \ell_{i-1} \vee x_i$
- For $i > p_1(m) + 1$ and $m[i] = 0$, $C(m,i) = \ell_1 \vee \ldots \vee \ell_{i-1} \vee \neg x_i$.

Otherwise, we let $C(m,i)$ undefined. Now we define the CNF Describe(R) to be the conjunction of all the defined clauses $C(m,i)$.

Example 1 *Let* $R = \{001, 100, 110, 111\}$ *; we have :* $p_0(001) = -1$*, and we get the clause* $C(001,3) = x_1 \vee x_2 \vee x_3$ *;* $p_1(001) = 0$*, and* $C(001,2) = x_1 \vee \neg x_2$ *;* $p_1(100) = 1$*, and* $C(100,3) = \neg x_1 \vee x_2 \vee \neg x_3$*. The other clauses* $C(m,i)$ *are undefined, and we finally get the CNF* Describe(R)$= (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2 \vee \neg x_3)$.

Note that R can be seen as a binary tree T, with each $m \in R$ corresponding to a branch of T. Then $C(m,i)$ corresponds intuitively to a missing subtree of T (a son of the ith node of the branch m), i.e., $C(m,i)$ forbids a set of vectors not in R.

Proposition 2 *The CNF* ψ=Describe(R) *describes* R*. It contains* $O(|R|n)$

clauses and can be computed in time $O(|R|n^2)$.

PROOF. Let $m \in R$. We show $m \models \psi$. By definition, $m \models C(m,i)$ holds for every clause $C(m,i)$ of ψ. Now let $m' \in R$, $m' \neq m$, and $C(m',i)$ a clause of ψ. Assume $m < m'$ (the case $m' < m$ is similar). Let i_0 be the minimal index such that $m[i_0] = 0$ and $m'[i_0] = 1$. If $i > i_0$, then m satisfies $C(m',i)$ via $\neg x_{i_0}$. Otherwise $i < i_0$, thus $m[i] = m'[i]$ and $m \models C(m',i)$. Finally, m is a model of ψ. Conversely, let $R = \{m_1, \ldots, m_{|R|}\}$, with $m_j < m_{j+1}$ $(0 < j < |R|)$, and let $m \in \{0,1\}^n \backslash R$. We show that m does not satisfy ψ. If $m < m_1$, let i_0 be the minimal index such that $m[i_0] = 0$ and $m_1[i_0] = 1$; m does not satisfy $C(m_1, i_0)$. The case $m_{|R|} < m$ is similar. Now assume $m_j < m < m_{j+1}$, and let i_0, i_1, i_2 be the minimal indexes such that $m_j[i_0] = 0$, $m_{j+1}[i_0] = 1$, $m_j[i_1] = 0$, $m[i_1] = 1$, $m[i_2] = 0$ and $m_{j+1}[i_2] = 1$. We have $p_1(m_j) = p_0(m_{j+1}) = i_0 - 1$, and necessarily ($i_0 < i_1$ and $i_0 = i_2$) or ($i_0 < i_2$ and $i_0 = i_1$). If $i_0 < i_1$, then we have $i_1 > p_1(m_j) + 1$ and thus m does not satisfy $C(m_j, i_1)$, and if $i_0 < i_2$, then $i_2 > p_0(m_{j+1}) + 1$ and m does not satisfy $C(m_{j+1}, i_2)$. Finally, m is not a model of ψ.
The number of clauses is $O(|R|n)$ by definition. Sorting R by lexicographic order requires $O(|R|n)$ steps (radix sort). For a given $m \in R$, computing $p_0(m)$ or $p_1(m)$ requires $O(n)$ steps, and writing the clauses, $O(n^2)$. □

4 A framework for identification

We now give a general procedure for `IDENTIF[`$\mathcal{C}$`]`, and exemplify it with the classes HORN and BIJUNCTIVE. We will see in section 5 that our framework also handles Horn-renamable and affine formulas. We use the notion of prime implicate. Let us recall that a clause C is called a *prime implicate* of a formula ψ if ψ logically implies C, but implies no proper subclause of C. We call a CNF *prime* if each of its clauses is a prime implicate of it.

Proposition 3 ([4, Lemma 3.2]) *Every prime CNF describing a Horn (or bijunctive) relation is Horn (resp. bijunctive).*

The proof given in [4] is for Horn formulas, but still works for bijunctive formulas. The idea is that every prime implicate of a formula ψ can be eventually obtained by resolution from any CNF logically equivalent to ψ [8], and that the resolvent of two Horn (resp. bijunctive) clauses is also Horn (resp. bijunctive).

We now define the procedure `Identify[`$\mathcal{C}$`]` for the problem `IDENTIF[`$\mathcal{C}$`]`.

Procedure `Identify[`$\mathcal{C}$`]`(R)
Step 1 : Compute a prime CNF ϕ describing R ;

$\phi \leftarrow$ `Describe(`R`)` ;
for every clause $C = t_1 \vee \ldots \vee t_k$ of ϕ **do**
  for every $m \in R$ **do**
    `last` $\leftarrow 1$;
    for $i = k$ **to** 1 **do**
      if m satisfies C via t_i
      then if `last` $= 1$ **then** [$T[t_i, m] \leftarrow$ 'lastyes' ; `last`$\leftarrow 0$] **else** $T[t_i, m] \leftarrow$ 'yes'
      else $T[t_i, m] \leftarrow$ 'no'
    endfor ;
  endfor ;
  `models`$\leftarrow \emptyset$;
  for $i = 1$ **to** k **do**
    if there exists $m \in R \backslash$`models` such that $T[t_i, m] =$'lastyes'
    then `models`$\leftarrow$`models` $\cup \{m' \in R, T[t_i, m'] =$ 'yes' or 'lastyes'$\}$ */* keep t_i */*
    else cancel t_i from C
  endfor ;
endfor ;
return ϕ ;

Fig. 1. Construction of `DescribePI(`R`)`

Step 2 : If ϕ is in $\mathcal{C}$ then return ϕ, otherwise return 'No'.

By Proposition 3, `Identify[`HORN`]` solves `IDENTIF[`HORN`]`, and the same holds for the class BIJUNCTIVE.

Example 4 *Let* $R = \{001, 100, 110, 111\}$*, as in example 1. The CNF* $(x_1 \vee x_3) \wedge (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2 \vee \neg x_3)$ *is prime and describes* R*. As it is not Horn (resp. bijunctive), we can deduce immediately that* R *admits no Horn (resp. bijunctive) description.*

Now we need a polynomial procedure for computing ϕ given R. We define the CNF `DecribePI(`R`)` to this end. We use R and the CNF $\psi =$ `Describe(`R`)` for reducing each clause C of ψ into a prime implicate of ψ. Given C, we drop a maximal number of literals t_i from C such that every $m \in R$ still satisfies C. We give a constructive definition of `DescribePI(`R`)` in figure 1 : T is a two-dimensional array indexed by the literals $t_i \in C$ and the vectors $m \in R$, with $T[t_i, m] =$'yes' if m satisfies C via t_i, and 'no' otherwise, and `models` is a set of vectors. When we keep a literal t_i in C, we add to `models` the vectors that satisfy C via t_i.

Proposition 5 *The CNF* `DescribePI(`R`)` *describes* R*. It is prime, contains* $O(|R|n)$ *clauses and can be computed in time* $O(|R|^2 n^2)$*.*

PROOF. By construction, `DescribePI(`R`)` logically implies `Describe(`R`)`, so its models are in R. The converse holds, for we keep one 'yes' or 'lastyes' per m in T. Now let C be a clause in `DescribePI(`R`)`, and t_i a literal in C.

As we have kept t_i in C, there exists $m \in R$ such that $T[t_i, m]$ ='lastyes' and $m \notin$ `models` when we consider t_i. Thus m does not satisfy the clause obtained from C by removing t_i, and C is a prime implicate of `DescribePI(`R`)`. Finally, `DescribePI(`R`)` contains as many clauses as the CNF `Describe(`R`)`, i.e., $O(|R|n)$, and it is easily seen that the time complexity of its construction is $O(|R|^2 n^2)$. □

By Proposition 5, the procedure `Identify[`HORN`]` solves `IDENTIF[`HORN`]` in time $O(|R|^2 n^2)$ with $O(|R|n)$ clauses, and the same holds for the class BIJUNCTIVE.

5 Horn-renamable and affine relations

We now consider the class HORN-RENAMABLE of *Horn-renamable* CNFs. A CNF ψ is called *Horn-renamable* if there exists a subset S of its variables such that replacing, for every $x \in S$, each occurrence of x in ψ with $\neg x$, and conversely (*renaming* x in ψ) yields a Horn formula. A relation is called *Horn-renamable* if it has a Horn-renamable description. *A priori*, there may exist no simpler method for identifying those formulas than testing the 2^n possible renamings and using algorithms for HORN ; thus the following result is not obvious.

Proposition 6 `IDENTIF[`HORN-RENAMABLE`]` *is solvable in time* $O(|R|^2 n^2)$ *with* $O(|R|n)$ *clauses.*

PROOF. Horn-renamable formulas are recognizable in linear time [1,2,5]. Since every prime CNF describing a Horn relation is Horn, and renaming preserves the notion of prime implicate, every prime CNF describing a Horn-renamable relation is Horn-renamable. Thus the procedure `Identify[`HORN-RENAMABLE`]` solves `IDENTIF[`HORN-RENAMABLE`]` in time $O(|R|^2 n^2)$ with $O(|R|n)$ clauses. □

We finally turn our attention to AFFINE, the class of *affine* formulas [6,9]. This class is one of the classes for which Schaefer shows that the generalized satisfiability problem is polynomial.

An *affine* formula is a system of linear equations on the two-element field, and a relation is *affine* if it is the set of solutions of such a system. For example, $R = \{0001, 0010, 1100, 1111\}$ is affine, since R is the set of solutions of the formula $(x_1 \oplus x_2 = 0) \wedge (x_1 \oplus x_3 \oplus x_4 = 1)$. In some sense, addition modulo

2 plays the same role in affine formulas as disjunction in CNF formulas. We denote by AFFINE the class of affine formulas.

The identification problem for affine formulas thus corresponds to finding a system of equations admitting a given set of solutions. This problem can be solved by using the tools of linear algebra [7], but what we show here is that our procedure can be straightforwardly applied to this class, despite the fact that being affine is not a syntactic property about CNFs. For this purpose, we exhibit a purely syntactic link between affine and CNF representations of a formula.

Let $C = t_1 \vee \ldots \vee t_p$ be a clause. We denote by $E(C)$ the equation $e(t_1) \oplus \ldots \oplus e(t_p) = 1$, where $e(t_i) = x_j$ if $t_i = x_j$ and $e(t_i) = x_j \oplus 1$ if $t_i = \neg x_j$.

Example 7 *Let $C = x_1 \vee \neg x_2 \vee \neg x_3 \vee x_4 \vee \neg x_5$. The equation $E(C)$ is $x_1 \oplus x_2 \oplus 1 \oplus x_3 \oplus 1 \oplus x_4 \oplus x_5 \oplus 1 = 1$, i.e., $\bigoplus_{i=1}^{5} x_i = 0$.*

Proposition 8 *Assume R is affine, and let $\phi = C_1 \wedge \ldots \wedge C_m$ be a prime CNF describing it. Then the affine formula $A(\phi) = E(C_1) \wedge \ldots \wedge E(C_m)$ describes R.*

PROOF. We first show that every solution of $A(\phi)$ is a model of ϕ. Let $C = \bigvee_i t_i$ be a clause in ϕ ; the equation $E(C)$ admits the same solutions as the equation $\bigoplus_i t_i = 1$. Consequently, if $m \in \{0,1\}^n$ satisfies $E(C)$, then an odd number of t_i's are assigned 1, thus at least one and m satisfies C. Thus the solutions of $A(\phi)$ all belong to R. Conversely, we know that if R is affine, then for all $m_1, m_2, \ldots, m_k$ in R with k odd, the vector $m = m_1 \oplus \ldots \oplus m_k$ is in R, where $(m_1 \oplus \ldots \oplus m_k)[i] = m_1[i] \oplus \ldots \oplus m_k[i]$ for all i. Indeed, if $E = (x_{i_1} \oplus \ldots \oplus x_{i_p} = b)$, with $b \in \{0,1\}$, is an equation satisfied by $m_1, \ldots, m_k$, then $(\bigoplus_{j=1}^{k} m_j)[i_1] \oplus \ldots \oplus (\bigoplus_{j=1}^{k} m_j)[i_p] = \bigoplus_{j=1}^{k} (m_j[i_1] \oplus \ldots \oplus m_j[i_p]) = b \oplus \ldots \oplus b$ (k times) ; since k is odd, this is equal to b, thus $m_1 \oplus \ldots \oplus m_k$ is a solution of E. So let $C = \bigvee_i t_i$ be a clause of ϕ, and, to obtain a contradiction, $m \in R$ that is not a solution of $E(C)$. Then without loss of generality, m assigns 1 to $t_1, t_2, \ldots, t_{2p}$, and 0 to the other literals of C. As C is a prime implicate of ϕ, there exists $\mu_1, \ldots, \mu_{2p} \in R$ such that for all i, μ_i satisfies C but not the clause obtained from it by removing t_i ; we deduce that for all i, μ_i assigns 1 to t_i and 0 to the other literals of C. Now the vector $m \oplus \mu_1 \oplus \ldots \mu_{2p}$ is in R by the remark above (the sum has $2p+1$ terms), but it assigns 0 to all the literals in C, a contradiction. □

We deduce that our framework applies to affine formulas. Indeed, we use the procedure `Identify[`AFFINE`]`, slightly modified for step 2 (due to the fact that being affine is not a syntactic property about CNFs) : we first compute $A(\phi)$, where ϕ is the prime CNF of step 1, and then test whether each m in

R is a solution of $A(\phi)$. If yes, R is the set of solutions of $A(\phi)$, since every solution of $A(\phi)$ is in R (cf. proof of Proposition 8). Otherwise, Proposition 8 ensures that R is not affine. Computing $A(\phi)$ is linear in the size of ϕ, and testing whether each $m \in R$ is a solution of ϕ requires $O(|R|^2 n^2)$ steps. Finally, we have a time complexity $O(|R|^2 n^2)$ and $O(|R|n)$ output equations.

Example 9 *Let again* $R = \{001, 100, 110, 111\}$ *;* `DescribePI(R)` *is* $\phi = (x_1 \vee x_3) \wedge (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2 \vee \neg x_3)$. *Thus* $A(\phi)$ *is the system* $(x_1 \oplus x_3 = 1) \wedge (x_1 \oplus x_2 = 0) \wedge (x_1 \oplus x_2 \oplus x_3 = 1)$. *But* $111 \in R$ *is not a solution of the first equation, thus* R *is not the set of solutions of* $A(\phi)$ *and we can deduce that* R *is not affine.*

Finally, let us recall (cf. for instance [6,9]) the well-known characterization of affine relations : R is affine if and only if for every $m_1, m_2, m_3 \in R$, $m_1 \oplus m_2 \oplus m_3$ is in R. The construction of Proposition 8 gives a new proof of this criterion, without using the tools of algebra.

Acknowledgements

The authors wish to thank an anonymous referee for suggesting the remark before Proposition 2.

References

[1] Aspvall, B., Recognizing disguised nr(1) instances of the satisfiability problem, J. of Algorithms 1 (1980) 97–103

[2] Chandru, V., Coullard, C.R., Hammer, P.L., Montanez, M. and Sun, X., On renamable Horn and generalized Horn functions, Ann. Math. AI 1 (1990) 33-47

[3] Dechter, R. and Pearl, J., Structure identification in relational data, Artificial Intelligence 58 (1992) 237–270

[4] Hammer, P.L. and Kogan, A., Horn functions and their DNFs, Inform. Process. Lett. 44 (1992) 23–29

[5] Hébrard, J.-J., A linear algorithm for renaming a set of clauses as a Horn set, Theoret. Comp. Sci. 124 (1994) 343–350

[6] Kavvadias, D. and Sideri, M., The inverse satisfiability problem, SIAM J. Comput. 28 (1998) 152–163

[7] Lang, S., Linear algebra, Addison-Wesley (1966)

[8] Quine, W.V., On cores and prime implicants of truth functions, Am. Math. Monthly 66 (1959) 755-760

[9] Schaefer, T.J., The complexity of satisfiability problems, in: Proc. 10th Annual ACM Symposium on Theory Of Computing, San Diego, CA (1978) 216–226