Mathematical Programming 79 (1997) 163–190

Logical analysis of numerical data [1]

Endre Boros [a,2], Peter L. Hammer [a,*], Toshihide Ibaraki [b,3], Alexander Kogan [a,c,4]

[a] *RUTCOR, Rutgers University, P.O. Box 5062, New Brunswick, NJ 08903, USA*
[b] *Department of Applied Mathematics and Physics, Graduate School of Engineering, Kyoto University, 606 Kyoto, Japan*
[c] *Department of Accounting and Information Systems, Faculty of Management, Rutgers University, Newark, NJ 07102, USA*

Received 27 February 1997; accepted 21 April 1997

Abstract

"Logical analysis of data" (LAD) is a methodology developed since the late eighties, aimed at discovering hidden structural information in data sets. LAD was originally developed for analyzing binary data by using the theory of partially defined Boolean functions. An extension of LAD for the analysis of numerical data sets is achieved through the process of "binarization" consisting in the replacement of each numerical variable by binary "indicator" variables, each showing whether the value of the original variable is above or below a certain level. Binarization was successfully applied to the analysis of a variety of real life data sets. This paper develops the theoretical foundations of the binarization process studying the combinatorial optimization problems related to the minimization of the number of binary variables. To provide an algorithmic framework for the practical solution of such problems, we construct compact linear integer programming formulations of them. We develop polynomial time algorithms for some of these minimization problems, and prove NP-hardness of others. © 1997 The Mathematical Programming Society, Inc. Published by Elsevier Science B.V.

Keywords: Data analysis; Boolean functions; Machine learning; Binarization; Set covering; Monotonicity; Thresholdness; Computational complexity

* Corresponding author. Email: hammer@rutcor.rutgers.edu.
[1] The authors gratefully acknowledge the partial support by the Office of Naval Research (grants N00014-92-J1375 and N00014-92-J4083).
[2] Email: boros@rutcor.rutgers.edu.
[3] Email: ibaraki@kuamp.kyoto-u.ac.jp.
[4] Email: kogan@rutcor.rutgers.edu.

Published by Elsevier Science B.V.
PII S0025-5610(97)00050-6

1. Introduction

1.1. Logical analysis of data

The study of numerical data, common to a large variety of disciplines, leads to interesting problems of combinatorial optimization, the solution of which can have an immediate impact on the understanding of the nature of the phenomena under investigation, the factors governing them, and on the ways we can influence their development.

The data we are examining here consists of collections of "observations" represented as points in $\mathbb{R}^d$. These observations can be associated to patients in a medical study, consumers in a marketing analysis, probes in a potential oil field, etc. The components of the vectors, called "attributes" stand for the results of various medical tests, financial parameters, geological measurements, etc. The observations fall in two classes: "positive" ones (e.g. healthy patients, potential buyers, oil rich areas, etc.) and "negative" ones (e.g. sick patients, etc.).

The goal of data analysis is to arrive to logical explanations distinguishing positive and negative observations. It was shown in [11,15] that – in the case of binary data – this goal can be achieved by the use of partially defined Boolean functions.

The usefulness of logical or Boolean techniques is demonstrated by many approaches in this field [1,10,12,17,22,23]. The classification power of the Boolean-based methodology of the *logical analysis of data* (LAD) and its applicability to oncology, psychiatry, oil exploration, economic analysis, and other fields are described in [7,16,19]. Preliminary results reported in [7] already indicate that the LAD approach not only has high classification power for distinguishing between positive and negative examples, but also is quite successful in extracting underlying structural information from the given data, which is useful in understanding why and how phenomena occur. While the techniques of LAD are entirely different from those of the pioneering works of Mangasarian (e.g. [18]), the two approaches provide strongly complementary and mutually reinforcing results.

The methodology of LAD is extended to the case of numerical data by a process called *binarization*, consisting in the transformation of numerical (real valued) data to binary (0,1) ones. In this transformation we map each observation $u = (u_A, u_B, \ldots)$ of the given numerical data set to a binary vector $x(u) = (x_1, x_2, \ldots) \in \{0,1\}^n$ by defining e.g. $x_1 = 1$ iff $u_A \geqslant \alpha_1$, $x_2 = 1$ iff $u_B \geqslant \alpha_2$, etc., and in such a way that if u and v represent a positive and a negative observation point respectively, then $x(u) \neq x(v)$. The binary variables x_i, $i = 1, \ldots, n$ associated to the real attributes are called *indicator variables*, and the real parameters α_i, $i = 1, \ldots, n$ used in the above process are called *cut points*.

The binarization process described above has been included in a current implementation of LAD and successfully used in the computational experiments reported in [7].

The object of this paper is to provide a theoretical foundation for the binarization process with the primary focus on the combinatorial optimization problems related to the minimization of the number of cut points. The paper studies the computational

Table 1
A numerical data set (S^+, S^-)

Attributes	A	B	C
S^+: positive examples	3.5	3.8	2.8
	2.6	1.6	5.2
	1.0	2.1	3.8
S^-: negative examples	3.5	1.6	3.8
	2.3	2.1	1.0

Table 2
A binarization of Table 1

Boolean variables	x_A	x_B	x_C
T: true points	1	1	0
	0	0	1
	0	1	1
F: false points	1	0	1
	0	1	0

complexity of these problems, develops polynomial time algorithms for some of them, and provides compact integer programming formulations for them.

1.2. Examples

The meaning of a "logical" explanation of numerical data can be best illustrated by an example. Let us consider Table 1, containing a set S^+ of positive observations and a set S^- of negative observations having the attributes A, B, C. To be more specific, we may think of the phenomenon X as a headache and the attributes A, B and C as representing say, blood pressure, body temperature, and the pulse rate, respectively (the numerical values being normalized appropriately).

In order to extract a logical explanation from this data set, let us first introduce the following cut points

$$\alpha_A = 3.0, \quad \alpha_B = 2.0, \quad \alpha_C = 3.0$$

for the attributes A, B, C, respectively, and then transform the numerical attributes to binary indicator variables by defining $x_A = 1$ iff $u_A \geqslant \alpha_A$, $x_B = 1$ iff $u_B \geqslant \alpha_B$ and $x_C = 1$ iff $u_C \geqslant \alpha_C$ for all points $u = (u_A, u_B, u_C) \in S^+ \cup S^-$. The result of this binarization of Table 1 is given in Table 2.

It would be natural to view Table 2 as a truth table of a *partially defined Boolean function* (pdBf) over the Boolean variables x_A, x_B, x_C. In a pdBf like Table 2, the binary vectors resulting from the numerical vectors in S^+ are called *true points*, and those resulting from S^- *false points*. The sets of true and false points in a pdBf are denoted by T and F respectively. Table 2 represents a partially defined function since some Boolean vectors do not appear in the table. If all Boolean vectors do appear in

Table 3
Another binarization of Table 1

Boolean variables	x_B	x'_C	x''_C
T: true points	1	0	1
	0	1	1
	1	0	1
F: false points	0	0	1
	1	0	0

a table, such a table represents a mapping $\{0,1\}^n \to \{0,1\}$, and is called a *Boolean function*.

Now consider a pdBf $g = (T, F)$ and a function f defined over the same set of n variables. If f is consistent with g, i.e., if all true points (respectively false points) of g are also true points (respectively false points) of f, the function f is called an *extension* of the pdBf g. In general, there are many extensions f of a given pdBf g. For example, an extension of the pdBf in Table 2 is given by the following disjunctive normal form (DNF):

$$f = x_A x_B \vee \bar{x}_A \bar{x}_B \vee x_B x_C. \tag{1}$$

We view this as a logical explanation of Table 1; it tells that the phenomenon X occurs if the values of attributes A and B are both high, or both low, or the values of attributes B and C are both high.

Although exactly one cut point was introduced for each attribute in the example above, there is no reason to prohibit the introduction of more than one, or of no, cut point for some attributes. As another example, let us introduce the following cut points

$$\alpha_B = 2.0, \quad \alpha'_C = 5.0, \quad \alpha''_C = 1.5,$$

that is, one cut point for the attribute B, two cut points for C, and none for A. Using two cuts points per one attribute may be natural; $x'_C = x''_C = 1$ would mean "high", $x'_C = 0$ and $x''_C = 1$ would mean "intermediate", and $x'_C = x''_C = 0$ would mean "low" (note that the combination $x'_C = 1$ and $x''_C = 0$ is not feasible). The resulting pdBf is shown in Table 3,[5] has an extension given by the following DNF:

$$f = x_B x''_C \vee x'_C. \tag{2}$$

The DNF (2) may be considered simpler than the DNF (1) since it has fewer terms and literals. Furthermore, the DNF (2) is positive (i.e., monotone increasing) in all its variables, meaning that a change of the value of one of the variables from 0 to 1 cannot decrease the value of the function from 1 to 0, whatever the values of the other variables are. In some cases, this fact by itself may be regarded as an important information for the understanding of the mechanism causing the phenomenon.

[5] Note that the first and the third vectors of T coincide.

1.3. Outline of the paper

The simple examples in Section 1.2 show that there are many possible ways to introduce cut points, as well as to interpret (i.e., to derive extensions of) the resulting logical data. It is frequently desired to find an extension which is as simple as possible under some well-defined criterion of simplicity, or to find an extension having a special functional form (e.g., positive), or both.

The problem of checking whether a pdBf (i.e., a binary data set partitioned into two classes) has an extension in a specified class $\mathcal{C}$ has been studied in relation to the detection of cause-effect relationships [11,15], learning and identification of a Boolean function [1,4,24], structural analysis of binary data [5], and others. Beside studying extensions in the class of *all* Boolean functions, we shall pay special attention to the frequently occurring classes of *monotone*, *Horn*, *threshold*, *linear* and *quadratic* functions (see Section 2.1 for their definitions). The classes of "decomposable" Boolean functions and decomposable extensions of pdBfs were studied in [5]. The complexity of finding extensions of a pdBf in various classes of functions was considered in [8], and extended in [9] to the more general cases in which some of the bits in the data vectors may be missing. These problems were shown to be polynomially solvable for all of the above six classes, as will be recalled in Section 2.1. Several classes of Boolean functions which may be particularly relevant to LAD were studied in [13].

In this paper we show first that for any numerical data set and for any of the six classes of functions above, it can be checked in polynomial time whether there exists a binarization admitting an extension in the given class. We also consider the special case in which at most one cut point per numerical attribute is used in the binarization, and show that this variant is NP-hard for all of the six classes considered above.

The main focus of this paper is on the problem of finding the minimum number of cut points (and their locations) under the constraint that the resulting pdBf has an extension in some given class $\mathcal{C}$.

In order to provide an algorithmic framework for this minimization problem we formulate it as a linear integer programming problem for all of the six classes above. For some of the classes this formulation turns out to be a set covering problem. If the problem sizes are not too large, these formulations can be used to solve the problems exactly, and even if the sizes are large, they can serve as starting points for the development of effective heuristic algorithms. Some heuristic algorithms were already implemented in the report [7], and we are currently elaborating on them to pursue further improvements.

Further, we analyze the restricted case of the minimization problem in which the dimension of the numerical data set is a fixed constant, and show that this restricted problem is polynomially solvable for the class of linear functions. We also prove that the minimization problem for the class of positive Boolean functions can be solved in polynomial time when the numerical data set is of dimension 2.

Finally, it is shown that the minimization problem becomes NP-hard for all other classes considered in this paper, even if the dimension of the numerical data set is restricted.

2. Definitions and preliminaries

2.1. Partially defined Boolean functions and their extensions

For a *Boolean function* (or simply, *function*) $f : \{0,1\}^n \to \{0,1\}$, the vector $x \in \{0,1\}^n$ is called a *true point* (respectively *false point*) if $f(x) = 1$ (respectively $f(x) = 0$). $T(f)$ (respectively $F(f)$) denotes the set of true points (respectively false points) of f; clearly, $T(f) \cap F(f) = \emptyset$ and $T(f) \cup F(f) = \{0,1\}^n$. For two functions f and h, we denote $f \leqslant h$ if $f(x) \leqslant h(x)$ for all $x \in \{0,1\}^n$. A function is *positive* (or *monotone*) if $x \leqslant y$ (i.e., $x_j \leqslant y_j$ for $j = 1,2,\dots,n$) implies $f(x) \leqslant f(y)$. A function f is called *threshold* if there exist $n+1$ real numbers w_j, $j = 0,1,2,\dots,n$, such that

$$f(x) = \begin{cases} 1 & \text{if } \sum_{j=1}^{n} w_j x_j \geqslant w_0, \\ 0 & \text{if } \sum_{j=1}^{n} w_j x_j < w_0. \end{cases} \tag{3}$$

The Boolean variables $x_1, x_2, \dots, x_n$ and their complements $\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n$ are called *literals*. A conjunction of literals (or simply, a *term*) $t = \bigwedge_{j\in P_t} x_j \bigwedge_{j\in N_t} \bar{x}_j$, with $P_t \cap N_t = \emptyset$, is called an *implicant* of f if $t \leqslant f$ (where t is understood as the function it represents). An implicant t is called *prime* if there is no implicant $t' = \bigwedge_{j\in P_{t'}} x_j \bigwedge_{j\in N_{t'}} \bar{x}_j$ such that $P_{t'} \subseteq P_t, N_{t'} \subseteq N_t$ and $P_{t'} \cup N_{t'} \neq P_t \cup N_t$. A disjunction $\bigvee_{i=1}^{m} t_i$ of terms $t_i = \bigwedge_{j\in P_{t_i}} x_j \bigwedge_{j\in N_{t_i}} \bar{x}_j$, $i = 1,2,\dots,m$, is called a *disjunctive normal form* (DNF), and it is well known that every Boolean function can be represented by a DNF. A DNF is a k-DNF if $|P_{t_i} \cup N_{t_i}| \leqslant k$ for all i. We shall call a Boolean function *quadratic* if it has a 2-DNF representation, and *linear* if it has a 1-DNF representation. A DNF $\bigvee_{i=1}^{m} t_i$ is *Horn* if every t_i satisfies $|N_{t_i}| \leqslant 1$. A function f is called *Horn* if it has a Horn DNF. It is known (e.g., [12]) that f is Horn if and only if $F(f)$ is closed under intersection operations (i.e., $x, y \in F(f)$ implies $x \wedge y \in F(f)$, where $(x \wedge y)_j = x_j \wedge y_j$, $j = 1,2,\dots,n$).

We shall consider in this paper the following special classes of Boolean functions: the class $\mathcal{C}_{\mathrm{ALL}}$ of all Boolean functions, the class $\mathcal{C}_{\mathrm{P}}$ of positive functions, the class $\mathcal{C}_{\mathrm{HORN}}$ of Horn functions, the class $\mathcal{C}_{\mathrm{TH}}$ of threshold functions, the class $\mathcal{C}_{\mathrm{LINEAR}}$ of linear functions, and the class $\mathcal{C}_{\mathrm{QUAD}}$ of quadratic functions.

A *partially defined Boolean function* (pdBf, for short) is defined by a pair (T, F), where $T, F \subseteq \{0,1\}^n$. A function f is an *extension* of pdBf (T, F) if $T \subseteq T(f)$ and $F \subseteq F(f)$ hold. The following six lemmas, proved in [8,11], give the basis for our discussion.

Lemma 1. *A pdBf (T, F) has an extension in $\mathcal{C}_{\mathrm{ALL}}$ if and only if $T \cap F = \emptyset$.* □

Lemma 2. *A pdBf (T, F) has an extension in $\mathcal{C}_P$ if and only if there is no $a \in T$ and $b \in F$ such that $a \leqslant b$.* $\square$

Lemma 3. *A pdBf (T, F) has an extension in $\mathcal{C}_{\text{HORN}}$ if and only if there is no $a \in T$ such that $a = \bigwedge_{b \in F \text{ s.t. } b \geqslant a} b$.* $\square$

Lemma 4. *A pdBf (T, F) has an extension in $\mathcal{C}_{\text{TH}}$ if and only if the system of inequalities*

$$\sum_{j=1}^{n} a_j w_j \geqslant w_0, \text{ for all } a \in T \quad \text{and} \quad \sum_{j=1}^{n} b_j w_j \leqslant w_0 - 1, \text{ for all } b \in F,$$

has a solution. $\square$

Lemma 5. *Define I_0 and I_1 for a given pdBf (T, F) by*

$$\begin{aligned} I_0 &= \{ j \in [n] \mid \text{all } b \in F \text{ satisfy } b_j = 1 \}, \\ I_1 &= \{ j \in [n] \mid \text{all } b \in F \text{ satisfy } b_j = 0 \}, \end{aligned} \tag{4}$$

where $[n] = \{1, 2, \ldots, n\}$. Then (T, F) has an extension in $\mathcal{C}_{\text{LINEAR}}$ if and only if $f = \bigvee_{j \in I_1} x_j \bigvee_{j \in I_0} \bar{x}_i$ satisfies $T(f) \supseteq T$. $\square$

Lemma 6. *Let I_0 and I_1 be as in (4), and define I_{00}, I_{01} and I_{11} for a given pdBf (T, F) by*

$$\begin{aligned} I_{00} &= \{ (i, j) \in [n] \times [n] \mid \text{every } b \in F \text{ satisfies either } b_i = 1 \text{ or } b_j = 1 \}, \\ I_{01} &= \{ (i, j) \in [n] \times [n] \mid \text{every } b \in F \text{ satisfies either } b_i = 1 \text{ or } b_j = 0 \}, \\ I_{11} &= \{ (i, j) \in [n] \times [n] \mid \text{every } b \in F \text{ satisfies either } b_i = 0 \text{ or } b_j = 0 \}, \end{aligned} \tag{5}$$

where $[n] = \{1, 2, \ldots, n\}$. Then the pdBf (T, F) has an extension in $\mathcal{C}_{\text{QUAD}}$ if and only if

$$f = \bigvee_{j \in I_1} x_j \bigvee_{j \in I_0} \bar{x}_i \bigvee_{(i,j) \in I_{11}} x_i x_j \bigvee_{(i,j) \in I_{01}} \bar{x}_i x_j \bigvee_{(i,j) \in I_{00}} \bar{x}_i \bar{x}_j$$

satisfies $T(f) \supseteq T$. $\square$

It is easy to see that the conditions in these lemmas can be checked in time polynomial in the input length of (T, F), i.e., $n(|T| + |F|)$. For example, the condition of Lemma 4 can be tested by linear programming in polynomial time.

2.2. Cut points, master pdBf and extensions

Let us now return to the numerical data as shown in Table 1. We assume that each observation is a d-dimensional real vector (i.e., there are d attributes), and denote the set of positive observations by S^+ and the set of negative observations by S^-, respectively.

Table 4
The master pdBf of the set of numerical data in Table 1

Attributes	A			B		C		
Variables	x_{A1}	x_{A2}	x_{A3}	x_{B1}	x_{B2}	x_{C1}	x_{C2}	x_{C3}
T	1	1	1	1	1	0	0	1
	0	1	1	0	0	1	1	1
	0	0	0	0	1	0	1	1
F	1	1	1	0	0	0	1	1
	0	0	1	0	1	0	0	0

For each attribute A, let $u_A^{(0)} > u_A^{(1)} > \cdots > u_A^{(K)}$ be all the distinct values of the A-component of $u \in S^+ \cup S^-$ arranged in decreasing order. We shall associate to A several cut points α_{Ai} and the corresponding binary *indicator variables* x_{Ai} defined by

$$x_{Ai} = \begin{cases} 1 & \text{if } A \geqslant \alpha_{Ai}, \\ 0 & \text{if } A < \alpha_{Ai}. \end{cases}$$

The cut points will be chosen in a way which allows to distinguish between positive and negative observations, if possible. Clearly, if we introduce two cut points between the consecutive values of attribute A, e.g. between $u_A^{(i)}$ and $u_A^{(i+1)}$, then the corresponding binary variables will be equivalent, and therefore will not help us in any way to make a distinction between the points in S^+ and S^-. Similarly, cut points above $u_A^{(0)}$ or below $u_A^{(K)}$, or between any two consecutive attribute values $u_A^{(i)}$ and $u_A^{(i+1)}$ realized by observations belonging only to S^+ or only to S^-, will not provide any help in distinguishing points of S^+ and S^-. Therefore, we shall consider at most one cut point between any two consecutive values of attribute A, and no cut points above $u_A^{(0)}$ or below $u_A^{(K)}$. More precisely, the largest set of binary variables we shall consider is obtained when one cut point $\alpha_{Ai} = (u_A^{(i)} + u_A^{(i+1)})/2$ is introduced between each pair of $u_A^{(i)}$ and $u_A^{(i+1)}$ for which there are observations $u' \in S^+$ and $u'' \in S^-$ (or $u' \in S^-$ and $u'' \in S^+$) such that $u'_A = u_A^{(i)}$ and $u''_A = u_A^{(i+1)}$. If for an attribute A we introduce K' cut points, then the value u_A is transformed to a K'-dimensional Boolean vector. When we introduce the largest set of cut points for each attribute, the resulting pdBf (T^*, F^*) is called the *master* pdBf of (S^+, S^-). Table 4 is the master pdBf of the numerical data in Table 1, where

$$\alpha_{A1} = 3.0, \quad \alpha_{A2} = 2.4, \quad \alpha_{A3} = 1.5,$$
$$\alpha_{B1} = 3.0, \quad \alpha_{B2} = 2.0,$$
$$\alpha_{C1} = 4.0, \quad \alpha_{C2} = 3.0, \quad \alpha_{C3} = 2.0.$$

Remark that the binarizations of Table 1 introduced in Tables 2 and 3 correspond to subsets of columns of the master pdBf of Table 4.

Let us call a set of cut points *consistent* if in the resulting pdBf (T, F) the sets T and F are disjoint. The first problem considered in this paper is to decide whether there exists a set of consistent cut points such that the resulting pdBf has an extension in a specified class C.

Lemma 7. *Let C be one of the six classes of functions listed in Section 1. If there is a set of cut points such that the resulting pdBf has an extension in C, then the master pdBf also has an extension in C.*

Proof. Immediate from the definitions of these classes, since adding variables (corresponding to new cut points) to binary vectors does not prevent (T, F) from having an extension in C. □

Lemma 7 shows that the master pdBf is the richest representation needed for finding an extension in the classes considered.

Given a numerical data set (S^+, S^-), we can first construct its master pdBf (T^*, F^*), and then check whether it has an extension in the class C. This can be done by testing the conditions in the corresponding lemma that describes when an extension in this class exists. The whole task can be carried out in time polynomial in the input length $d(|S^+| + |S^-|)$, since the construction of the master pdBf (T^*, F^*) can be done in $d(|S^+| + |S^-|)^2$ time, and checking the conditions of Lemmas 1–6 can also be done in time polynomial in $n(|T^*| + |F^*|)$ $(\leqslant d(|S^+| + |S^-|)^2)$. This establishes the following theorem.

Theorem 8. *It can be checked in polynomial time whether the master pdBf (T^*, F^*) of a numerical data set (S^+, S^-) has an extension in any of the six classes of functions listed in Section 1.* □

For some classes of functions, the condition for the existence of an extension for the master pdBf can be stated directly in terms of the original numerical data set.

Lemma 9. *Let us denote the master pdBf of a numerical data set (S^+, S^-) by (T^*, F^*). Then,*

(i) *(T^*, F^*) has an extension in C_{ALL} if and only if $S^+ \cap S^- = \emptyset$,*

(ii) *(T^*, F^*) has an extension in C_{P} if and only if there is no pair $u \in S^+$ and $v \in S^-$ such that $u \leqslant v$.*

Proof. Immediate from Lemmas 1 and 2, and the definition of the master pdBf. □

Let us consider the class C_{TH}, and let us recall that two subsets S^+ and S^- of the Euclidean space $\mathbb{R}^d$ are said to be *linearly separable* if there exist $d + 1$ real numbers W_j, $j = 0, 1, 2, \ldots, d$, such that

$$\sum_{j=1}^{d} W_j u_j \geqslant W_0, \quad \text{for all } u \in S^+ \quad \text{and} \quad \sum_{j=1}^{d} W_j v_j < W_0, \quad \text{for all } v \in S^-. \tag{6}$$

Linear separability of two sets S^+ and S^- is well known to be equivalent to the disjointness of their convex hulls, i.e. to $\operatorname{conv} S^+ \cap \operatorname{conv} S^- = \emptyset$. Let us remark that for finite sets the integrality of the coefficients W_0 and W_j, $j = 1, \ldots, d$ can also be assumed.

One can easily see that there are examples of (S^+, S^-), which are not linearly separable, but for which a corresponding pdBf (T, F) does have a threshold extension. If for example

$$S^+ = \{u^{(1)} = (1.0, 1.0)\} \quad \text{and} \quad S^- = \{v^{(1)} = (2.0, 0.0), v^{(2)} = (0.0, 2.0)\},$$

then

$$u^{(1)} = \frac{1}{2}v^{(1)} + \frac{1}{2}v^{(2)},$$

showing that S^+ and S^- are not linearly separable. However, if we introduce the cut points $\alpha_A = 0.5$ and $\alpha_B = 0.5$, then the resulting pdBf (T, F),

$$T = \{a^{(1)} = (1, 1)\} \quad \text{and} \quad F = \{b^{(1)} = (1, 0), b^{(2)} = (0, 1)\},$$

has a threshold extension with coefficients $w_1 = w_2 = 1.0$ and $w_0 = 2.0$.

Let us show next that whenever the sets S^+ and S^- are linearly separable, i.e. there exist reals W_j, $j = 0, 1, \ldots, d$ satisfying (6), the master pdBf (T^*, F^*) does always have a threshold extension.

Theorem 10. *A numerical data set (S^+, S^-) is linearly separable only if its master pdBf (T^*, F^*) has a threshold extension.*

Proof. Let us recall that $u_A^{(0)} > u_A^{(1)} > \cdots > u_A^{(K_A)}$ denote the distinct values of attribute A (for $A = 1, \ldots, d$) occurring in $S^+ \cup S^-$, and that the binary values of the indicator variables

$$x_{Ai} = \begin{cases} 1 & \text{if } u_A > u_A^{(i)}, \\ 0 & \text{otherwise}, \end{cases}$$

for $i = 1, \ldots, K_A$ correspond to $u \in S^+ \cup S^-$ in the master pdBf (T^*, F^*). Hence we have

$$\begin{aligned} \sum_{A=1}^{d} W_A u_A &= \sum_{A=1}^{d} W_A \left(u_A^{(K_A)} + \sum_{i=1}^{K_A} x_{Ai} (u_A^{(i-1)} - u_A^{(i)}) \right) \\ &= \sum_{A=1}^{d} W_A u_A^{(K_A)} + \sum_{A=1}^{d} \sum_{i=1}^{K_A} x_{Ai} W_A (u_A^{(i-1)} - u_A^{(i)}). \end{aligned}$$

Thus the coefficients $w_0 = W_0 - \sum_{A=1}^{d} W_A u_A^{(K_A)}$, and $w_{Ai} = W_A(u_A^{(i-1)} - u_A^{(i)})$ for $A = 1, \ldots, d$ and $i = 1, \ldots, K_A$ define a threshold extension of the master pdBf (T^*, F^*). $\square$

3. Single cut point for each attribute

In some cases, it is important to restrict the number of cut points introduced for each attribute. As a special case, we consider the following problem.

SINGLE-CP($\mathcal{C}$)
Input: A numerical data set (S^+, S^-).
Question: Is there a set of cut points, with at most one cut point per attribute, such that the resulting pdBf (T, F) has an extension in $\mathcal{C}$?

In this section we show that the problem SINGLE-CP($\mathcal{C}$) is NP-complete for all six classes introduced in Section 1. First,

Theorem 11. *The problem* SINGLE-CP($\mathcal{C}$) *is* NP-*complete for each of the classes* $\mathcal{C}_{\mathrm{ALL}}$, $\mathcal{C}_{\mathrm{HORN}}$, $\mathcal{C}_{\mathrm{TH}}$, $\mathcal{C}_{\mathrm{LINEAR}}$, *and* $\mathcal{C}_{\mathrm{QUAD}}$.

Proof. It is clear that SINGLE-CP($\mathcal{C}$) is in NP for all the classes listed in the theorem statement. In order to prove that SINGLE-CP($\mathcal{C}$) is NP-hard, we shall reduce to it the 3SAT problem, a well-known NP-complete problem.

3SAT
Input: A set of clauses C_i, $i = 1, 2, \ldots, m$, each containing at most three literals from $\{X_1, \bar{X}_1, \ldots, X_n, \bar{X}_n\}$.
Question: Is there an assignment of values to the Boolean variables $X_1, X_2, \ldots, X_n$ such that all the clauses are satisfied?

We shall associate to a given set of clauses C_i, $i = 1, 2, \ldots, m$, a numerical data set (S^+, S^-), involving n attributes and $m + 1$ input vectors, such that SINGLE-CP($\mathcal{C}$) has a solution for this (S^+, S^-) if and only if the answer of 3SAT is YES. The set S^- will consist of a single negative point $(1, 1, \ldots, 1)$, while the set S^+ will include a positive point $u^{(i)}$ for every clause $C_i = \bigvee_{j \in P_i} X_j \bigvee_{j \in N_i} \overline{X}_j$, defined by

$$u_j^{(i)} = \begin{cases} 2 & \text{if } j \in P_i, \\ 0 & \text{if } j \in N_i, \\ 1 & \text{if } j \notin P_i \cup N_i. \end{cases}$$

For example, for the clauses $C_1 = (X_2 \vee \overline{X}_3 \vee \overline{X}_5)$, $C_2 = (\overline{X}_1 \vee X_4)$, $C_3 = (X_1 \vee \overline{X}_2 \vee X_5)$, and $C_4 = (X_3 \vee \overline{X}_4 \vee \overline{X}_5)$ the corresponding numerical data set (S^+, S^-) will be

S^+: positive examples	1	2	0	1	0
	0	1	1	2	1
	2	0	1	1	2
	1	1	2	0	0
S^-: negative examples	1	1	1	1	1

To solve SINGLE-CP($\mathcal{C}_{\mathrm{ALL}}$), we shall introduce at most one cut point per attribute in such a way that the resulting pdBf (T, F) has an extension in $\mathcal{C}_{\mathrm{ALL}}$ (i.e. $T \cap F = \emptyset$). Clearly, for each attribute j, the cut point α_j must be either 0.5 or 1.5. Let us associate to the cut points the following binary assignment of the Boolean variables X_j:

$$X_j = \begin{cases} 1 & \text{if } \alpha_j = 1.5, \\ 0 & \text{if } \alpha_j = 0.5. \end{cases} \tag{7}$$

With this definition, there is an assignment that satisfies all clauses C_i if and only if the associated binarization produces a consistent pdBf (T, F) with $T = \{x(u^{(i)}) \mid u^{(i)} \in S^+\}$ and $F = \{x(v) \mid v = (1, 1, \ldots, 1)\}$. Indeed, point $u^{(i)} \in S^+$ can be separated from point $v \in S^-$ either by setting $\alpha_j = 1.5$ for some $j \in P_i$, or by setting $\alpha_j = 0.5$ for some $j \in N_i$, i.e. exactly when the associated binary assignment satisfies clause C_i.

Note that if the pdBf (T, F) is consistent, then it has an extension represented by the DNF $\bigvee_{j=1}^{n} X_j^{\alpha_j}$, where $X_j^{\alpha_j} = X_j$ if $\alpha_j = 1.5$ and $X_j^{\alpha_j} = \bar{X}_j$ if $\alpha_j = 0.5$. Clearly, this function is linear, and hence it is also quadratic, Horn, and threshold. Therefore, (S^+, S^-) has an extension in $\mathcal{C}_{\mathrm{ALL}}$ if and only if it has an extension in all of the classes $\mathcal{C}_{\mathrm{HORN}}$, $\mathcal{C}_{\mathrm{TH}}$, $\mathcal{C}_{\mathrm{LINEAR}}$, and $\mathcal{C}_{\mathrm{QUAD}}$. □

Theorem 12. *The problem* SINGLE-CP($\mathcal{C}_{\mathrm{P}}$) *is* NP-*complete.*

Proof. It is clear that SINGLE-CP($\mathcal{C}_{\mathrm{P}}$) is in NP. We shall prove the NP-hardness by a reduction from 3SAT, as defined in the proof of Theorem 11.

To accomplish this, we shall associate to a given set of clauses C_i $i = 1, \ldots, m$ a numerical data set (S^+, S^-), involving n real attributes and $2m$ input vectors. Let us choose subsets $A_i \subset [n]$ of size $|A_i| = \lfloor n/2 \rfloor$ and $B_i = [n] \setminus A_i$, for $i = 1, \ldots, m$, such that $|A_i \cap B_k| \geqslant 7$ if $i \neq k$, and $|A_i \cap B_k| = 0$ if $i = k$. It is easy to see that such subsets exist if n is large enough, since $m \leqslant \binom{\lfloor n/7 \rfloor}{\lfloor n/14 \rfloor}$ for large n. Furthermore, let us denote by I_i the set of indices of the variables in the clause C_i, $i = 1, \ldots, m$, and define $S^+ = \{a^{(i)} \mid i = 1, \ldots, m\}$ and $S^- = \{b^{(i)} \mid i = 1, \ldots, m\}$ by

$$a_j^{(i)} = \begin{cases} +1 & \text{if } X_j \in C_i \text{ or } j \in A_i \setminus I_i, \\ 0 & \text{if } \bar{X}_j \in C_i, \\ -1 & \text{if } j \in B_i \setminus I_i, \end{cases}$$

and

$$b_j^{(i)} = \begin{cases} +1 & \text{if } j \in A_i \setminus I_i, \\ 0 & \text{if } X_j \in C_i, \\ -1 & \text{if } \bar{X}_j \in C_i \text{ or } j \in B_i \setminus I_i. \end{cases}$$

We can observe that, since each attribute can have only three distinct values, $1, 0, -1$, there are only two possible cut points $\alpha_j = 0.5$ and $\alpha_j = -0.5$ for each j. Therefore, let us associate an assignment to the Boolean variables X_j by

$$X_j = \begin{cases} 1 & \text{if } \alpha_j = 0.5, \\ 0 & \text{if } \alpha_j = -0.5. \end{cases} \tag{8}$$

We claim that the cut points α_j, $j = 1, \ldots, n$, define a pdBf which has a positive extension if and only if the associated assignment satisfies all clauses C_i, $i = 1, \ldots, m$. Let us remark first that for $i \neq k$, $a^{(i)}$ and $b^{(k)}$ are always positively separated in $j \in (A_i \cap B_k) \setminus (I_i \cup I_k)$, (i.e., $a_j^{(i)} > \alpha_j > b_j^{(k)}$), regardless of the value of α_j. Since $|I_i \cup I_k| \leqslant 6$ and $|A_i \cap B_k| \geqslant 7$ for $i \neq k$, such an index j always exists. We can also notice that the vectors $a^{(i)}$ and $b^{(i)}$ have the same components in all $j \notin I_i$. Hence, to separate them, there must be a $j \in I_i$ such that $X_j \in C_i$ and $\alpha_j = 0.5$, or $\bar{X}_j \in C_i$ and $\alpha_j = -0.5$. In other words, $a^{(i)}$ and $b^{(i)}$ will be separated positively if and only if the corresponding assignment (8) satisfies the clause C_i.

This proves the claim, and concludes the proof. □

4. Minimization of the number of cut points

Given a numerical data set (S^+, S^-) we have shown above that the corresponding master pdBf (T^*, F^*) can be used to check in polynomial time the existence of an extension in a given class $\mathcal{C}$. If such extensions exist, the next fundamental problem is to find a "most compact" extension. In this paper, the number of cut points will be used as a measure of compactness. Therefore, we shall study the problem

MIN-CP($\mathcal{C}$)
Input: A numerical data set (S^+, S^-) such that its master pdBf has an extension in $\mathcal{C}$.
Output: A minimum cardinality set of cut points such that the corresponding pdBf (T, F) has an extension in $\mathcal{C}$.

It will be seen in Section 4.1 that the problem MIN-CP($\mathcal{C}$) is NP-hard for all the six classes. Motivated by this, we shall also consider in Section 5 and in the Appendix the special cases in which the dimension d of S^+ and S^- is fixed to 2, 3, and an arbitrary constant k (positive integer). These problems will be denoted as MIN-2CP($\mathcal{C}$), MIN-3CP($\mathcal{C}$), and MIN-kCP($\mathcal{C}$) respectively. It will be seen that even in these special cases only the problems MIN-2CP($\mathcal{C}_{\mathrm{P}}$) and MIN-kCP($\mathcal{C}_{\mathrm{LINEAR}}$) are polynomially solvable.

4.1. NP-hardness results

In this subsection we show that the problem MIN-CP($\mathcal{C}$) is NP-hard for all the six classes considered in Section 1. To prove this, we shall use a reduction from the minimum set covering problem, MIN-COVER, which is known to be NP-hard [14], and which is defined as follows.

MIN-COVER
Input: An $m \times d$ matrix $A = \{a_{ij}\}$, where $a_{ij} = 0$ or 1 for all $i \in [m]$ and $j \in [d]$, satisfying $\sum_{j=1}^{d} a_{ij} \geqslant 1$ for all $i \in [m]$.

Output: A set of columns J of A such that $\sum_{j\in J} a_{ij} \geqslant 1$ for all $i \in [m]$, and the size $|J|$ is minimal.

Theorem 13. *The problem* MIN-CP($\mathcal{C}$) *is* NP-*hard for all of the six classes defined in Section* 2.

Proof. Given a MIN-COVER problem, let us define a numerical data set (S^+, S^-) by

$$S^+ = \{(a_{11}, a_{12}, \ldots, a_{1d}), \ldots, (a_{m1}, a_{m2}, \ldots, a_{md})\} \quad \text{and} \quad S^- = \{(0, 0, \ldots, 0)\}$$

Although the data are already binary, it is necessary to introduce a cut point $\alpha_j = 0.5$ if the attribute j is used as a Boolean variable x_j in the pdBf (T, F) obtained after binarization. Let J be the set of indices j for which the cut points $\alpha_j = 0.5$ are introduced. Then

$$T = \{(a_{1j}, j \in J), (a_{2j}, j \in J), \ldots, (a_{mj}, j \in J)\} \quad \text{and} \quad F = \{(0, 0, \ldots, 0)\},$$

where each vector has dimension $|J|$. By Lemma 1, this pdBf (T, F) has an extension in $\mathcal{C}_{\mathrm{ALL}}$ if and only if $\sum_{j\in J} a_{ij} \geqslant 1$ holds for all $i \in [m]$. Since the number of cut points is $|J|$, (T, F) is a desired pdBf if and only if J is a solution to MIN-COVER. This proves that MIN-COVER is reduced to MIN-CP($\mathcal{C}_{\mathrm{ALL}}$), and hence MIN-CP($\mathcal{C}_{\mathrm{ALL}}$) is NP-hard.

Furthermore, notice that, if J allows extensions in $\mathcal{C}_{\mathrm{ALL}}$, one of them can be represented as $f = \bigvee_{j\in J} x_j$, which is positive, Horn, threshold and linear (hence quadratic). This means that MIN-CP($\mathcal{C}$) is NP-hard for all the six classes $\mathcal{C}_{\mathrm{ALL}}$, $\mathcal{C}_{P}$, $\mathcal{C}_{\mathrm{HORN}}$, $\mathcal{C}_{\mathrm{TH}}$, $\mathcal{C}_{\mathrm{LINEAR}}$, and $\mathcal{C}_{\mathrm{QUAD}}$. $\square$

As a remark, we note that the above construction uses at most one cut point $\alpha_j = 0.5$ for each attribute j of (S^+, S^-). Therefore, MIN-CP($\mathcal{C}$) remains NP-hard for all the six classes even after imposing the additional constraint of using at most one cut point for each attribute.

4.2. Integer programming formulations

In this subsection, we formulate MIN-CP($\mathcal{C}$) as integer programming problems for all the classes $\mathcal{C}$ under consideration. In the first part we formulate MIN-CP($\mathcal{C}$) problems for the classes $\mathcal{C}_{\mathrm{ALL}}$, $\mathcal{C}_{\mathrm{P}}$, and $\mathcal{C}_{\mathrm{LINEAR}}$ as MIN-COVER problems (see also [11]). In the second part we give general integer programming formulations of MIN-CP($\mathcal{C}$) for $\mathcal{C}_{\mathrm{HORN}}$, $\mathcal{C}_{\mathrm{QUAD}}$ and $\mathcal{C}_{\mathrm{TH}}$.

The integer programming formulations given in this subsection provide a good starting point for developing practical algorithms for the MIN-CP($\mathcal{C}$) problems. Although MIN-COVER is known to be NP-hard, various exact and heuristic algorithms have been developed for this problem, since it is encountered frequently in many operations research applications [2]. If problem sizes are not too large, there is a good chance of finding an exact solution. Heuristic algorithms can be used to solve large size problems, or to solve moderate size problems faster.

Let us first consider $\mathcal{C}_{\mathrm{ALL}}$. Given a numerical data set (S^+, S^-), we first construct the master pdBf (T^*, F^*) as in Section 2.2. Since each indicator variable in the master pdBf corresponds to a cut point, solving MIN-CP($\mathcal{C}_{\mathrm{ALL}}$) is equivalent to finding a minimum subset of the indicator variables which can still separate the true vectors in T^* from the false vectors in F^*. To model this as MIN-COVER, let y_j be a decision variable, $j = 1, \ldots, n$, describing whether the indicator variable x_j of the master pdBf (and hence the cut point associated with it) is chosen $(y_j = 1)$ or not $(y_j = 0)$ in a minimum subset. For every pair of $u \in S^+$ and $v \in S^-$ let $x(u) \in T^*$ and $x(v) \in F^*$ respectively denote the n-dimensional Boolean vectors obtained from u and v. For every $j = 1, \ldots, n$ let us define

$$a_j^{(u,v)} = \begin{cases} 1 & \text{if } x_j(u) \neq x_j(v), \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

Consider now the following MIN-COVER problem.

$$\begin{aligned} \text{minimize} \quad & \sum_{j=1}^{n} y_j \\ \text{subject to} \quad & \sum_{j=1}^{n} a_j^{(u,v)} y_j \geqslant 1, \quad \text{for all pairs } (u,v) \text{ s.t. } u \in S^+,\ v \in S^- \\ & y_j = 0, 1, \quad j = 1, 2, \ldots, n. \end{aligned} \tag{10}$$

Here each constraint $\sum a_j^{(u,v)} y_j \geqslant 1$ guarantees that $u \in S^+$ and $v \in S^-$ are transformed into distinct Boolean vectors using only the chosen variables x_j. If a solution y is feasible (i.e., satisfies all the constraints), then the resulting pdBf (T, F) satisfies the condition of Lemma 1, and if y is optimal, it gives the minimum number of cut points necessary to satisfy Lemma 1. Let n be the number of cut points introduced to define the master pdBf. This MIN-COVER problem has n variables, and $|S^+| \times |S^-|$ constraints.

We extend now the same idea to the case of $\mathcal{C}_P$. This can be accomplished by simply changing the definition of $a_j^{(u,v)}$ as follows.

$$a_j^{(u,v)} = \begin{cases} 1 & \text{if } x_j(u) = 1 \text{ and } x_j(v) = 0, \\ 0 & \text{otherwise.} \end{cases}$$

The constraint $\sum a_j^{(u,v)} y_j \geqslant 1$ now guarantees that, for each pair of $u \in S^+$ and $v \in S^-$, at least one j having the property $x_j(u) > x_j(v)$ is selected. Therefore, the resulting (T, F) constructed from (T^*, F^*), by using the coordinates j for which $y_j = 1$ satisfies the condition in Lemma 2. This MIN-COVER formulation for $\mathcal{C}_P$ has the same number of variables and constraints as the above formulation for $\mathcal{C}_{\mathrm{ALL}}$.

As the third case of MIN-COVER formulation, we consider $\mathcal{C}_{\mathrm{LINEAR}}$. In this case, Lemma 5 states that a cut point α_{Ai} for the attribute A of the given data set (S^+, S^-)

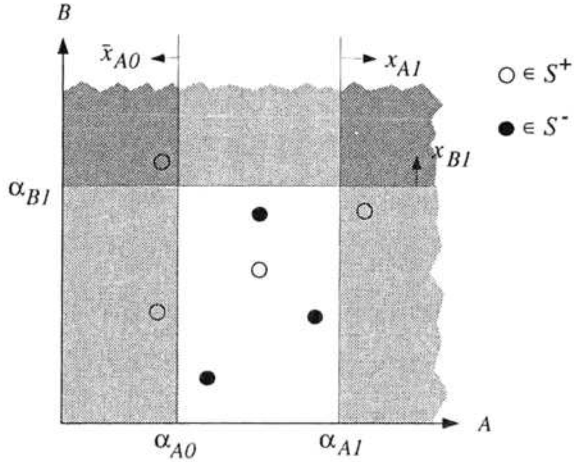


Fig. 1. Possible cut points for $\mathcal{C}_{\text{LINEAR}}$.

is useful only if the resulting indicator variable x_{Ai} belongs to one of the sets I_0 or I_1 defined in Lemma 5. This is possible if and only if either (i) $v_A \geqslant \alpha_{Ai}$ for all $v \in S^-$, or (ii) $v_A < \alpha_{Ai}$ for all $v \in S^-$.

Therefore, without loss of generality we can restrict our attention to the following two cut points for each attribute A:

$$\alpha_{A0} = \min\{v_A \mid v \in S^-\} - \Delta_0/2 \quad \text{and} \quad \alpha_{A1} = \max\{v_A \mid v \in S^-\} + \Delta_1/2; \tag{11}$$

here $\Delta_0 > 0$ is the gap between $\min\{v_A \mid v \in S^-\}$ and the largest $u_A, u \in S^+$, which is smaller than $\min\{v_A \mid v \in S^-\}$ (if there is no such u_A, the cut point α_{A0} is not needed), and $\Delta_1 > 0$ is defined in a similar way. The situation is illustrated in Fig. 1 for the two-dimensional case, where x_{Ai} and x_{Bj} are the indicator variables associated with the cut points α_{Ai} and α_{Bj}, respectively.

The problem MIN-COVER can now be constructed for $\mathcal{C}_{\text{LINEAR}}$ in the same manner as for $\mathcal{C}_{\text{ALL}}$. The only difference from the case $\mathcal{C}_{\text{ALL}}$ is that here each y_j corresponds to α_{A0} or to α_{A1} for some attribute A. A solution y of MIN-COVER gives the 1-DNF

$$f = \Big(\bigvee_{y_{A1}=1} x_{A1}\Big) \vee \Big(\bigvee_{y_{A0}=1} \bar{x}_{A0}\Big),$$

which is an extension of (T, F). This MIN-COVER problem has only $2d$ variables, where d is the dimension of vectors in (S^+, S^-). The number of constraints remains the same as in the above two cases.

We shall present now integer programming formulations of MIN-CP($\mathcal{C}$) for $\mathcal{C}_{\text{QUAD}}$, $\mathcal{C}_{\text{HORN}}$ and $\mathcal{C}_{\text{TH}}$. The problem MIN-CP($\mathcal{C}_{\text{QUAD}}$) can be stated as an integer program, which is resembling MIN-COVER. For each linear or quadratic term t_k, which corresponds to an element in $I_0, I_1, I_{00}, I_{01}, I_{11}$ of Lemma 6, let us introduce a decision variable $z_k \in \{0,1\}$ $(k = 1,2,\dots,K)$. Clearly, $K \leqslant n + 3\binom{n}{2}$, unless $F^* = \emptyset$. Let us define for each $u \in S^+$

$$a_k^{(u)} = \begin{cases} 1 & \text{if } t_k(x(u)) = 1, \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

Similarly to the previously discussed cases, we introduce a decision variable y_j for each indicator variable x_j in the master pdBf (and hence to the cut point associated with it) to describe whether x_j is selected ($y_j = 1$) or not ($y_j = 0$). Then an integer program for MIN-CP($\mathcal{C}_{\mathrm{QUAD}}$) is the following.

$$\begin{aligned} \text{minimize} \quad & \sum_{j=1}^{n} y_j \\ \text{subject to} \quad & \sum_{k=1}^{K} a_k^{(u)} z_k \geqslant 1, \quad \text{for all } u \in S^+ \\ & y_j \geqslant z_k, \quad \text{for all } j \text{ for which } x_j \text{ or } \bar{x}_j \text{ is involved in } t_k, \\ & \qquad\qquad \text{for all } k = 1,\dots,K \\ & y_j = 0,1, \quad j = 1,2,\dots,n \\ & z_k = 0,1, \quad k = 1,2,\dots,K. \end{aligned} \tag{13}$$

The constraint $\sum_{k=1}^{K} a_k^{(u)} z_k \geqslant 1$ guarantees that at least one quadratic (or linear) term t_k with $t_k(x(u)) = 1$ is chosen for every $u \in S^+$. The constraint $y_j \geqslant z_k$ becomes redundant if $z_k = 0$, but if $z_k = 1$ it enforces $y_j = 1$ whenever x_j or $\bar{x}_j$ is involved in t_k. This guarantees that all the indicator variables x_j in each chosen term t_k (i.e., with $z_k = 1$) are selected. Therefore, any feasible solution of (13) defines a pdBf that satisfies the conditions of Lemma 6, and the optimum value of the objective function gives the minimum number of cut points necessary to satisfy Lemma 6. The integer program (13) has at most $2n + 3\binom{n}{2}$ variables and at most $|S^+| + n + 3\binom{n}{2}$ constraints.

It is straightforward to see how the above integer program can be modified for finding the minimum number of cut points such that the resulting pdBf has an extension in the class of d-DNFs, for any fixed d.

The integer program (13) can be modified for solving MIN-CP($\mathcal{C}_{\mathrm{HORN}}$), based on the following observation. For $u \in S^+$, let $J_u = \{j \mid x_j(u) = 1,\ j = 1,2,\dots,n\}$. Then a Horn extension exists if and only if, for every $u \in S^+$, at least one of the Horn terms $\bigwedge_{j\in J_u} x_j$ and $\bar{x}_i \bigwedge_{j\in J_u} x_j$ for $i \in \{1,2,\dots,n\} \setminus J_u$ takes the value 0 on all points $x(v)$, $v \in S^-$. Therefore, a Horn DNF (if any) can be constructed by taking the disjunction of a number of such terms.

It should be noted that some literals may be removed from the selected terms in order to decrease the number of cut points, as long as the resulting DNF still represents a Horn extension of (S^+, S^-). To achieve this goal, let t_k, $k = 1,2,\dots,K$, denote all the terms of the above type. Clearly, $K \leqslant n|S^+|$. Let us introduce decision variables z_k $(k = 1,2,\dots,K)$ for such terms, and define

$$a_k^{(u)} = \begin{cases} 1 & \text{if } t_k(x(u)) = 1, \\ 0 & \text{otherwise,} \end{cases}$$

and

$$b_j^{(v,k)} = \begin{cases} 1 & \text{if } x_j(v) = 0 \text{ and } t_k \text{ contains literal } x_j, \\ & \text{or } x_j(v) = 1 \text{ and } t_k \text{ contains } \bar{x}_j, \\ 0 & \text{otherwise.} \end{cases}$$

Let us further introduce a decision variable y_j to describe whether the Boolean variable x_j in the master pdBf (and hence the cut point associated with it) is chosen ($y_j = 1$) or not ($y_j = 0$). The resulting integer program for the solution of MIN-CP($\mathcal{C}_{\text{HORN}}$) becomes

$$\begin{aligned} \text{minimize} \quad & \sum_{j=1}^{n} y_j \\ \text{subject to} \quad & \sum_{k=1}^{K} a_k^{(u)} z_k \geqslant 1, \quad \text{for all } u \in S^+ \\ & \sum_{j=1}^{n} b_j^{(v,k)} y_j \geqslant z_k, \quad \text{for all } k = 1, 2, \ldots, K \text{ and for all } v \in S^- \\ & y_j = 0, 1, \quad j = 1, 2, \ldots, n \\ & z_k = 0, 1, \quad k = 1, 2, \ldots, K. \end{aligned} \tag{14}$$

The constraint $\sum_{k=1}^{K} a_k^{(u)} z_k \geqslant 1$ guarantees that there is at least one Horn term t_k for every $u \in S^+$, which satisfies $t_k(x(u)) = 1$. The constraint $\sum_{j=1}^{n} b_j^{(v,k)} y_j \geqslant z_k$ guarantees that, for every selected term t_k and $v \in S^-$, at least one indicator variable x_j involved in this term will also be selected, guaranteeing that this term takes the value 0 on v. Therefore, any feasible solution of (14) defines a pdBf which satisfies the conditions of Lemma 3, and if y is optimal, the optimum value gives the minimum number of cut points necessary to satisfy Lemma 3. The integer linear program (14) has at most $n + n|S^+|$ variables and at most $|S^+| + n|S^+| \times |S^-|$ constraints.

The integer program for the solution of MIN-CP($\mathcal{C}_{\text{TH}}$) is based on Lemma 4 that requires the following linear program to have a solution:

$$\begin{aligned} & \sum_{j=1}^{n} a_j w_j \geqslant w_0, \quad \text{for all } a \in T^*, \\ & \sum_{j=1}^{n} b_j w_j \leqslant w_0 - 1, \quad \text{for all } b \in F^*. \end{aligned} \tag{15}$$

For the development that follows, we need to establish an upper bound M on the absolute values of the w's. Since the number of variables in (15) is $n + 1$ and since all the coefficients are $+1$, -1, or 0, each component of a solution (if any) can be represented as the ratio of two determinants of dimension $n+1$ with elements $+1$, -1, or 0 (see, e.g., [3]). Based on this, we can use the Hadamard inequality to show that M satisfies

$$M \leqslant n^{n/2}. \tag{16}$$

Introducing now a decision variable y_j to describe whether the Boolean variable x_j in the master pdBf (and hence the cut point associated with it) is chosen ($y_j = 1$) or not ($y_j = 0$), a mixed integer program for MIN-CP($\mathcal{C}_{\mathrm{TH}}$) can be formulated as follows:

$$\begin{aligned} &\text{minimize} && \sum_{j=1}^{n} y_j \\ &\text{subject to} && \sum_{j=1}^{n} a_j w_j \geqslant w_0, \quad \text{for all } a \in T^* \\ &&& \sum_{j=1}^{n} b_j w_j \leqslant w_0 - 1, \quad \text{for all } b \in F^* \\ &&& w_j \geqslant -My_j, \quad j = 1,2,\dots,n \\ &&& w_j \leqslant My_j, \quad j = 1,2,\dots,n \\ &&& y_j = 0,1, \quad j = 1,2,\dots,n, \end{aligned} \tag{17}$$

where M satisfies (16).

If $y_j = 0$, the constraints $w_j \geqslant -My_j$ and $w_j \leqslant My_j$ force $w_j = 0$, while if $y_j = 1$, these constraints have no influence on the existence of a solution. The first two groups of constraints in (17) are the separation conditions of Lemma 4. Therefore, any feasible solution of (17) defines a pdBf that satisfies the conditions of Lemma 4, and if y is optimal, it gives the minimum number of cut points necessary to satisfy Lemma 4. The mixed integer linear program (17) has n binary variables, $n+1$ real variables, and $2n + |S^+| + |S^-|$ constraints.

Finally note that all the integer programs formulated in this subsection can be easily modified for the case of a single cut point for each attribute. Indeed, if the set J_A denotes all the indices of cut points for the attribute A, then it is sufficient to add to each problem the following d constraints:

$$\sum_{j \in J_A} y_j \leqslant 1, \quad \text{for all attributes } A.$$

5. Low dimensional data sets

5.1. Polynomially solvable cases

Although MIN-CP($\mathcal{C}$) is NP-hard for all the classes under consideration (as shown in Theorem 13), we point out in this subsection that in certain cases this problem can be solved in polynomial time if the dimension d of the given data set is bounded. Two such cases are MIN-kCP($\mathcal{C}_{\text{LINEAR}}$) and MIN-2CP($\mathcal{C}_{\text{P}}$).

We shall first consider the problem MIN-kCP($\mathcal{C}_{\text{LINEAR}}$). As described in Section 4.1, this problem can be formulated as MIN-COVER having $2d$ variables and $|S^+| \times |S^-|$ constraints. Since d is limited by a constant k, in order to solve this problem, we generate all the possible vectors $y \in \{0,1\}^{2k}$ to find a feasible vector that minimizes $\sum y_j$. Since the number of possible Boolean vectors is bounded by a constant 2^{2k}, the whole process can be performed in polynomial time, thus proving the next theorem.

Theorem 14. *If the dimension d of the given data set (S^+, S^-) is limited by a constant k, then the problem* MIN-kCP($\mathcal{C}_{\text{LINEAR}}$) *can be solved in polynomial time.* □

Before studying the problem MIN-2CP($\mathcal{C}_{\text{P}}$), let us consider a data set (S^+, S^-) and its extension in the two-dimensional space (i.e., $d = 2$), where the two attributes are denoted A and B. The positive and negative observations are represented in the 2-dimensional plane as shown in Fig. 2(a). We draw vertical lines corresponding to the cut points introduced for A, and horizontal lines for B, thus creating a number of rectangular regions. A minimal rectangular region (not containing other rectangular regions) is called a *cell*, and each cell corresponds to a unique Boolean vector resulting from the binarization using the cut points (see Fig. 2(b)). The resulting pdBf (T, F) is obtained by defining T as the set of all the Boolean vectors corresponding to the cells containing positive observations, and defining F as the set of all the Boolean vectors corresponding to the cells containing negative observations.

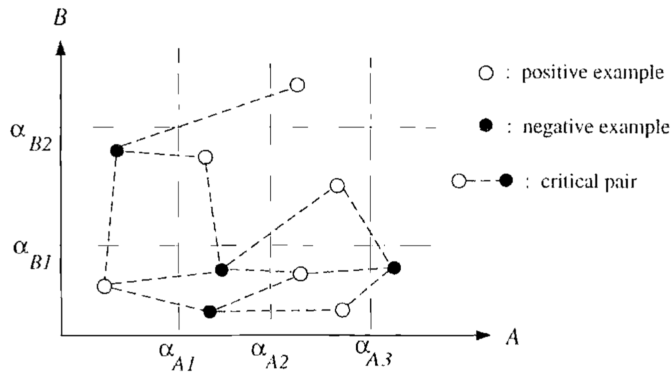
Recall that the master pdBf of a numerical data set (S^+, S^-) has a positive extension if and only if no pair of $u \in S^+$ and $v \in S^-$ satisfies $u \leqslant v$ (by Lemma 9). We say that (S^+, S^-) is *positively oriented* if this condition is satisfied. For example, the data set (S^+, S^-) in Fig. 3 is positively oriented.

We shall call a pair of positive and negative points in the 2-dimensional plane *critical* if the minimum rectangle containing them contains no other point of $S^+ \cup S^-$. The broken line segments in Fig. 2 indicate all critical pairs in this data set. We say that a line (vertical or horizontal) *separates* a critical pair if the two points of the pair are located on opposite sides of the line. We shall say that a vertical (horizontal) line *positively separates* the critical pair (u, v), $u \in S^+$, $v \in S^-$, if u is to the right (above) of the line. The following lemma follows immediately from Lemma 2.

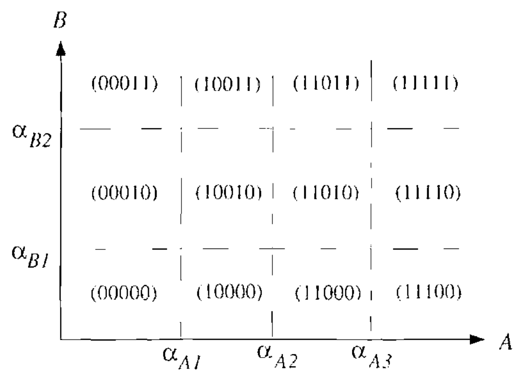
Lemma 15. *Given a 2-dimensional positively oriented data set (S^+, S^-), let (T, F) denote the pdBf defined by a set of cut points. Then (T, F) has an extension in $\mathcal{C}_{\text{P}}$ if and only if every critical pair is positively separated by at least one of the lines corresponding to the considered cut points.* □

We shall describe now a polynomial time procedure to solve MIN-2CP($\mathcal{C}_{\text{P}}$).

Let $u', u'' \in S^+$ and $v', v'' \in S^-$. The two pairs (u', v') and (u'', v'') will be called *independent* if there is no cut point that separates both these pairs (i.e., none of the



(a) Positive and negative examples, and critical pairs



(b) Boolean vectors $(x_{A1}, x_{A2}, x_{A3}, x_{B1}, x_{B2})$

Fig. 2. A 2-dimensional data set.

vertical or horizontal lines separates both (u', v') and (u'', v'')). In other words, the open interval (v'_A, u'_A) does not intersect the interval (v''_A, u''_A), and the interval (v'_B, u'_B) does not intersect the interval (v''_B, u''_B). For example, the positively oriented (S^+, S^-) in Fig. 4 contains five pairwise independent pairs indicated by solid line segments. It is clear that the number of cut points needed for the pdBf (T, F) to be consistent is at least the maximum number of pairwise independent pairs in (S^+, S^-), because no line can separate more than one pair of a family of pairwise independent pairs. The polynomial time algorithm to be given below is a by-product of the following minimax result.

Theorem 16. *If a given data set (S^+, S^-) is positively oriented, then the maximum number of pairwise independent pairs in (S^+, S^-) equals the minimum number of cut points needed to produce a consistent pdBf (T, F).*

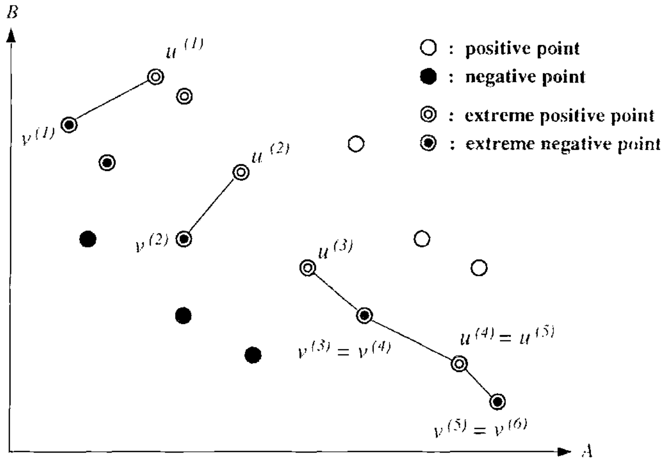


Fig. 3. An example of positively oriented data set (S^+, S^-).

After introducing some more notations and definitions, we shall prove this statement by actually presenting a construction of a consistent separation that involves as many cut points as the number of pairwise independent pairs.

Let us call a positive point $u \in S^+$ *extreme* if there is no distinct positive point $u' \in S^+$ such that $u' < u$. Let us denote by S^+_{ex} the set of extreme positive points in S^+. Similarly, a negative point $v \in S^-$ is called *extreme* if there is no distinct negative point $v' \in S^-$ such that $v' > v$. Let us denote by S^-_{ex} the set of extreme negative points in S^-. The significance of extreme points consists in the fact that any separation of all critical pairs of extreme points will be at the same time a separation of all critical pairs in Lemma 15. In Fig. 3, there are five extreme positive points and five extreme negative points, which are indicated as double circles.

For a point w, let us denote by $NW(w)$ the region of all 2-dimensional points located to the "north-west" from w. More precisely,

$$NW(w) = \{w' \mid w'_A \leqslant w_A, w'_B \geqslant w_B\}.$$

Similarly,

$$NE(w) = \{w' \mid w'_A \geqslant w_A, w'_B \geqslant w_B\},$$

$$SW(w) = \{w' \mid w'_A \leqslant w_A, w'_B \leqslant w_B\},$$

$$SE(w) = \{w' \mid w'_A \geqslant w_A, w'_B \leqslant w_B\}.$$

We say that a point w' *precedes* another point w'' if $w' \in NW(w'')$, and denote this relation by $w' \preceq w''$. Since any two extreme points $u', u'' \in S^+_{\text{ex}}$ satisfy either $u' \in NW(u'')$ or $u'' \in NW(u')$, this precedence relation is a linear order on S^+_{ex}. Similarly, this relation is also a linear order on S^-_{ex}.

We shall describe now a method to construct a set of extreme points which will be used to find pairwise independent pairs for Theorem 16, and then to find cut points to separate them. Let $u^{(1)}$ be the $\preceq$-smallest element in $S^+_{\rm ex}$, and $v^{(1)}$ be the $\preceq$-smallest element in $S^-_{\rm ex}$. Recursively, let $u^{(i+1)}$ be the $\preceq$-smallest element in $SE(v^{(i)}) \cap S^+_{\rm ex}$, and $v^{(i+1)}$ be the $\preceq$-smallest element in $SE(u^{(i)}) \cap S^-_{\rm ex}$. In this process, if $SE(v^{(k)}) \cap S^+_{\rm ex}$ or $SE(u^{(k)}) \cap S^-_{\rm ex}$ is empty, then the corresponding $(k+1)$th point is not defined, and the construction thereafter is stopped. Let us call all the constructed points $u^{(i)}$ and $v^{(i)}$, $i = 1, 2, \ldots$ *corner* positive and *corner* negative points, respectively. Note that $SE(u^{(i+1)}) \subseteq SE(u^{(i)})$ and $SE(v^{(i+1)}) \subseteq SE(v^{(i)})$ hold for all i. Fig. 3 contains the corner positive points $u^{(1)}, u^{(2)}, \ldots, u^{(5)}$ and corner negative points $v^{(1)}, v^{(2)}, \ldots, v^{(6)}$, for which $v^{(3)} = v^{(4)}, v^{(5)} = v^{(6)}$ and $u^{(4)} = u^{(5)}$ hold.

Assume that the positive corner points $u^{(1)}, u^{(2)}, \ldots, u^{(k_u)}$ and the negative corner points $v^{(1)}, v^{(2)}, \ldots, v^{(k_v)}$ are constructed in the above process. Since the construction alternates between positive and negative points, it is easy to see that k_u and k_v differ at most by one. Let us denote

$$k = \min\{k_u, k_v\}.$$

Clearly, $k \geqslant 1$ holds if $S^+ \neq \emptyset$ and $S^- \neq \emptyset$. It is important to notice that the above construction rule implies that the pairs $(u^{(i)}, v^{(i)})$ and $(u^{(j)}, v^{(j)})$ are independent for any $i \neq j$. Therefore, we find k pairwise independent pairs $(u^{(i)}, v^{(i)})$, $i = i, 2, \ldots, k$ (represented in the example of Fig. 3 by solid line segments).

For a pair $(u^{(i)}, v^{(i)})$, the two points $u^{(i)}$ and $v^{(i)}$ may be either $\preceq$-comparable or not. If they are not $\preceq$-comparable, then $v^{(i)} \leqslant u^{(i)}$. On the other hand, if they are $\preceq$-comparable, the subsequent pairs have a very special structure, as shown in the next lemma.

Lemma 17. *If there exists i such that $u^{(i)} \preceq v^{(i)}$, then $v^{(i)} = v^{(i+1)}$ and $v^{(i+1)} \preceq u^{(i+1)}$. Similarly, if $v^{(i)} \preceq u^{(i)}$, then $u^{(i)} = u^{(i+1)}$ and $u^{(i+1)} \preceq v^{(i+1)}$.*

Proof. We consider only the case of $u^{(i)} \preceq v^{(i)}$, since the other case can be proved in a similar way. In this case, we have $v^{(i)} \in SE(u^{(i)})$. Moreover, by definition, $v^{(i)}$ is the $\preceq$-smallest element in $SE(u^{(i)}) \cap S^-_{\rm ex}$, since if there were a $\preceq$-smaller element v' in $SE(u^{(i)}) \cap S^-_e$, then v' would satisfy $v' \in SE(u^{(i)}) \cap S^-_{\rm ex} \subseteq SE(u^{(i-1)}) \cap S^-_{\rm ex}$ and v' would play the role of $v^{(i)}$, a contradiction. Therefore, $v^{(i+1)} = v^{(i)}$. If $u^{(i+1)}$ exists, then $u^{(i+1)} \in SE(v^{(i)}) \cap S^+_e$ holds by definition, and it follows that $v^{(i+1)} = v^{(i)} \preceq u^{(i+1)}$. $\square$

As a result of Lemma 17, if there is a pair $(u^{(i)}, v^{(i)})$ such that $u^{(i)} \preceq v^{(i)}$, then

$$u^{(i)} \preceq v^{(i)} = v^{(i+1)} \preceq u^{(i+1)} = u^{(i+2)} \preceq \cdots \preceq v^{(k_v-1)} = v^{(k_v)} \quad (\text{or } u^{(k_u-1)} = v^{(k_u)}).$$

A similar property holds also for the case of $v^{(i)} \preceq u^{(i)}$.

In the example of Fig. 3, the five independent pairs of corner points satisfy $v^{(1)} \leqslant u^{(1)}$, $v^{(2)} \leqslant u^{(2)}$, and $u^{(3)} \preceq v^{(3)} = v^{(4)} \preceq u^{(4)} = u^{(5)} \preceq v^{(5)} = v^{(6)}$.

Let

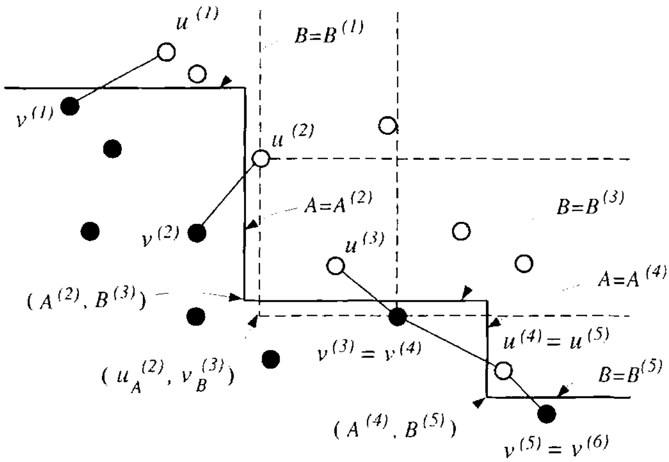


Fig. 4. Separation of the independent pairs in the previous data set.

$$\Delta_A = \min\{|w'_A - w''_A| \mid w', w'' \in S^+ \cup S^-,\ w'_A \neq w''_A\},$$

$$\Delta_B = \min\{|w'_B - w''_B| \mid w', w'' \in S^+ \cup S^-,\ w'_B \neq w''_B\}.$$

It can be seen that if points $w, w' \in S^+ \cup S^-$, then $w'_A \notin (w_A - \frac{1}{2}\Delta_A, w_A)$ or w'_B and $w'_B \notin (w_B, w_B + \frac{1}{2}\Delta_B)$. For the subsequent construction we shall introduce the potential cut points $A = A^{(i)}$ and $B = B^{(i)}$:

$$A^{(i)} = u_A^{(i)} - \tfrac{1}{2}\Delta_A, \qquad B^{(i)} = v_B^{(i)} + \tfrac{1}{2}\Delta_B.$$

We are now ready to turn to

Proof of Theorem 16. We have seen that the above procedure constructs k pairwise independent pairs. We shall prove the theorem by constructing the same number k of cut points, which separate positively all critical pairs.

The selection of the cut points starts from the very last (i.e., the kth) pair $u^{(k)}$ and $v^{(k)}$. Let us consider first the case in which there exists an $i \leqslant k$ such that the points $u^{(i)}$ and $v^{(i)}$ are $\preceq$-comparable. Then the corner points $u^{(k)}$ and $v^{(k)}$ are $\preceq$-comparable by Lemma 17. If $u^{(k)} \preceq v^{(k)}$, then we choose $B = B^{(k)}$ as the first cut point, and continue choosing cut points alternating between Bs and As, $(B = B^{(k)}, A = A^{(k-1)}, B = B^{(k-2)}, \ldots)$ until the first pair $(u^{(1)}, v^{(1)})$ is separated. If, on the other hand, $v^{(k)} \preceq u^{(k)}$, then we choose alternatively $A = A^{(k)}, B = B^{(k-1)}, A = A^{(k-2)}, \ldots$ until the first pair $(u^{(1)}, v^{(1)})$ is separated. These alternating cut points form a staircase as shown in Fig. 4. To show that this staircase separates positively all critical pairs in (S^+, S^-), it is sufficient to prove that there is no positive point to the south-west of this staircase, and no negative point to the north-east of it.

Let us consider in detail the case of $u^{(k)} \preceq v^{(k)}$, since the case of $v^{(k)} \preceq u^{(k)}$ can be treated in a similar way. We show first that there is no negative point to the north-east of the staircase. For the area $NE(A^{(k-1)}, B^{(k)})$, this amounts to showing that there is no negative point in $NE(u_A^{(k-1)}, v_B^{(k)})$ (except for $v^{(k)}$). Obviously,

$$NE(u_A^{(k-1)}, v_B^{(k)}) = NE(u^{(k-1)}) \cup NE(v^{(k)}) \cup (SE(u^{(k-1)}) \cap NW(v^{(k)})).$$

The area $NE(u^{(k-1)})$ does not contain a negative point, since any $w \in NE(u^{(k-1)})$ satisfies $w \geqslant u^{(k-1)}$ and the data set (S^+, S^-) is positively oriented. The area $NE(v^{(k)})$ does not contain any negative point, since $v^{(k)}$ is an extreme negative point. Finally, the area $(SE(u^{(k-1)}) \cap NW(v^{(k)}))$ does not contain any negative point, since $v^{(k)}$ is the $\preceq$-smallest negative point in $SE(u^{(k-1)})$. In a similar way, it can be shown that there is no negative point in $NE(A^{(k-3)}, B^{(k-2)})$, and so on. In Fig. 4, it is illustrated by broken lines how $NE(u_A^{(2)}, v_B^{(3)})$ can be represented as $NE(u^{(2)}) \cup NE(v^{(3)}) \cup (SE(u^{(2)}) \cap NW(v^{(3)}))$.

If k is even, the argument is complete, since the north-east area of the staircase is the union of $k/2$ north-east regions $NE(A^{(2j-1)}, B^{(2j)})$, $j = 1, 2, \ldots, k/2$. If k is odd, then it remains to be shown that there is no negative point with $B \geqslant B^{(1)}$, or equivalently, that there is no negative point in the region with $B \geqslant v_B^{(1)}$, which is $NE(v^{(1)}) \cup NW(v^{(1)})$. Indeed, $NE(v^{(1)})$ contains no negative point, since $v^{(1)}$ is an extreme negative point. Also, $NW(v^{(1)})$ contains no negative point, since $v^{(1)}$ is the first corner negative point.

Let us now discuss why there is no positive point to the south-east of the staircase. Notice first that there is no positive points with $B \leqslant B^{(k)}$ (equivalently, $B \leqslant v_B^{(k)}$). Indeed, $SW(v^{(k)})$ does not contain any positive point, since (S^+, S^-) is positively oriented, and $SE(v^{(k)})$ does not contain any positive point, since $u^{(k)}$ ($\preceq v^{(k)}$) is the last corner positive point. In order to show that there is no positive point in $SW(A^{(k-1)}, B^{(k-2)})$ (equivalently, $SW(u_A^{(k-1)}, v_B^{(k-2)})$), we need the following representation:

$$SW(u_A^{(k-1)}, v_B^{(k-2)}) = SW(u^{(k-1)}) \cup SW(v^{(k-2)}) \cup (SE(v^{(k-2)}) \cap NW(u^{(k-1)})).$$

The area $SW(u^{(k-1)})$ does not contain any positive point, since $u^{(k-1)}$ is an extreme positive point. The area $SW(v^{(k-2)})$ does not contain any positive point w, since (S^+, S^-) is positively oriented. Finally, the area $(SE(v^{(k-2)}) \cap NW(u^{(k-1)}))$ does not contain any positive point, since $u^{(k-1)}$ is the $\preceq$-smallest positive point in $SE(v^{(k-2)})$. In a similar way, it can be shown that there is no negative point in $SW(A^{(k-3)}, B^{(k-4)})$, and so on.

If k is odd, the argument is complete, since the south-west area of the staircase is the union of $(k-1)/2$ south-west regions $SW(A^{(2j)}, B^{(2j-1)})$ and of the first half-plane defined by $B \leqslant v_B^{(k)}$. If k is even, then it remains to be shown that there is no positive point with $A \leqslant A^{(1)}$ (equivalently, $A \leqslant u_A^{(1)}$). Indeed, the area $SW(u^{(1)})$ contains no positive point, since $u^{(1)}$ is an extreme positive point; and $NW(u^{(1)})$ contains no positive point, since $u^{(1)}$ is the first corner positive point. This completes the analysis of the case $u^{(k)} \preceq v^{(k)}$.

Finally, let us consider the case in which there is no $i \leqslant k$ such that the points $u^{(i)}$ and $v^{(i)}$ are $\preceq$-comparable. Let us analyze first the case $k_u = k_v$. There are two ways of constructing the separating staircase using k cut points: either $B^{(k)}, A^{(k-1)}, B^{(k-2)}, \ldots$ or $A^{(k)}, B^{(k-1)}, A^{(k-2)}, \ldots$. In either case, an analysis similar to the above shows that there is no negative point to the north-east of the staircase, and no positive point to

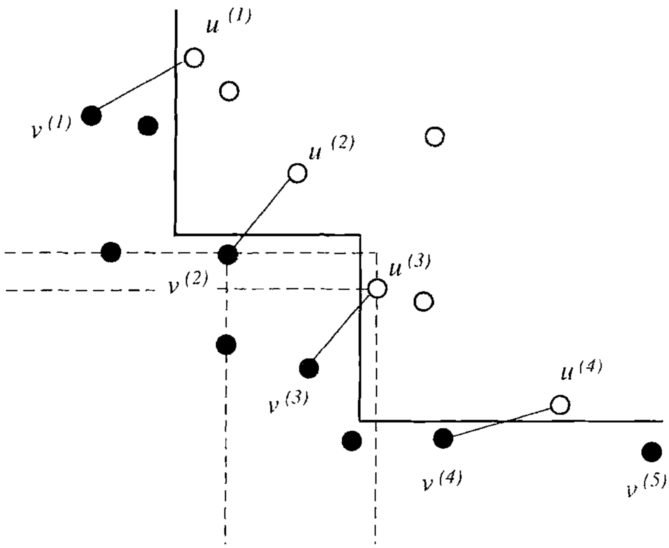


Fig. 5. Separation of independent pairs without $\preceq$-comparability.

the south-west of it. Let us consider now the case $k_v = k_u + 1$ (hence $k = k_u$). In this case $v^{(k+1)} \neq v^{(k)}$, and the staircase is constructed by choosing the cut points $B^{(k)}, A^{(k-1)}, B^{(k-2)}, \ldots$. Notice that in this case there is no positive point with $B \leqslant B^{(k)}$ (equivalently, $B \leqslant v_B^{(k)}$). Indeed, $SW(v^{(k)})$ contains no positive point (by the assumption that (S^+, S^-) is positively oriented), and $SE(v^{(k)})$ contains no positive point (since $u^{(k)}$ is the last corner positive point). The rest of the argument proceeds as above. (Note that the construction of the staircase using the cut points $A^{(k)}, B^{(k-1)}, A^{(k-2)}, \ldots$ would not work, since the point $v^{(k+1)}$ is located to the north-east of the staircase.) The last case $k_u = k_v + 1$ (hence $k = k_v$) is similar to the above, and we omit its proof. Fig. 5 shows the structure with four independent corner pairs when $k_u = 4$ and $k_v = 5$. It also shows the staircase constructed as in the proof. The broken lines in the figure indicate how $SW(u_A^{(3)}, v_B^{(2)})$ is represented as $SW(u^{(3)}) \cup SW(v^{(2)}) \cup (SE(v^{(2)}) \cap NW(u^{(3)}))$. $\square$

Finally we have arrived at the next main theorem of this subsection.

Theorem 18. *The problem* MIN-2CP($\mathcal{C}_P$) *can be solved in polynomial time.*

Proof. Immediate from the construction in the proof of Theorem 16, which can obviously be completed in polynomial time. In fact, it is not difficult to implement the algorithm so that the running time is $O(d|S^+ \cup S^-| \log |S^+ \cup S^-|)$. $\square$

5.2. NP-hard cases

We have seen in the previous subsection that MIN-CP($\mathcal{C}$) is solvable in polynomial time for some classes $\mathcal{C}$ if the dimension d is bounded. However, no substantial extensions of these results can be expected because of the NP-hardness of the problem as shown by the following result.

Theorem 19. *The problem* MIN-2CP($\mathcal{C}$) *for the classes* $\mathcal{C}_{\mathrm{ALL}}$, $\mathcal{C}_{\mathrm{HORN}}$, $\mathcal{C}_{\mathrm{TH}}$ *and* $\mathcal{C}_{\mathrm{QUAD}}$ *as well as the problem* MIN-3CP($\mathcal{C}_{\mathrm{P}}$) *are all* NP-*hard.*

The proof of this theorem is based on a reduction from the well-known NP-hard problem MAX-2SAT [14]. For the lengthy and technical proof we refer the reader to [6].

References

[1] D. Angluin, Queries and concept learning, *Machine Learning* 2 (1988) 319–342.
[2] E. Balas and A. Ho, Set covering algorithms using cutting planes, heuristics and subgradient optimization: A computational study, *Mathematical Programming* 12 (1980) 37–60.
[3] R. Bellman, *Introduction to Matrix Analysis* (McGraw-Hill, New York, 1960).
[4] J.C. Bioch and T. Ibaraki, Complexity of identification and dualization of positive Boolean functions, *Information and Computation* 123 (1995) 50–63.
[5] E. Boros, V. Gurvich, P.L. Hammer, T. Ibaraki and A. Kogan, Decompositions of partially defined Boolean functions, *Discrete Applied Mathematics* 62 (1995) 51–75.
[6] E. Boros, P.L. Hammer, T. Ibaraki, A. Kogan, Logical analysis of numerical data, RUTCOR Research Report RRR 04-97, RUTCOR, Rutgers University, 1997.
[7] E. Boros, P.L. Hammer, T. Ibaraki, A. Kogan, E. Mayoraz and I. Muchnik, An implementation of logical analysis of data, RUTCOR Research Report RRR 22-96, RUTCOR, Rutgers University, 1996.
[8] E. Boros, T. Ibaraki and K. Makino, Error-free and best-fit extensions of partially defined Boolean functions, RUTCOR Research Report RRR 14-95, RUTCOR, Rutgers University, 1995.
[9] E. Boros, T. Ibaraki and K. Makino, Extensions of partially defined Boolean functions with missing data, RUTCOR Research Report RRR 06-96, RUTCOR, Rutgers University, 1996.
[10] P. Clark and T. Niblett, The CN2 induction algorithm, *Machine Learning* 3 (1989) 261–283.
[11] Y. Crama, P.L. Hammer and T. Ibaraki, Cause-effect relationships and partially defined Boolean functions, *Annals of Operations Research* 16 (1988) 299–326.
[12] R. Dechter and J. Pearl, Structure identification in relational data, *Artificial Intelligence* 58 (1992) 237–270.
[13] O. Ekin, P.L. Hammer and A. Kogan, On connected Boolean functions, RUTCOR Research Report RRR 36-96, RUTCOR, Rutgers University, 1996.
[14] M.R. Garey and D.S. Johnson, *Computers and Intractability* (Freeman, New York, 1979).
[15] P.L. Hammer, Partially defined Boolean functions and cause-effect relationships, in: *Proceedings of the International Conference on Multi-Attribute Decision Making Via OR-Based Expert Systems*, University of Passau, Passau, Germany, 1986.
[16] A.B. Hammer, P.L. Hammer and I. Muchnik, Logical analysis of Chinese productivity patterns, RUTCOR Report RR 1-96, RUTCOR, Rutgers University, 1996.
[17] S.J. Hong, R-MINI: An iterative approach for generating minimal rules from examples, *IEEE Transactions on Knowledge and Data Engineering*, to appear.
[18] O.L. Mangasarian, Mathematical programming in machine learning, in: G. Di Pillo and F. Giannessi, eds., *Nonlinear Optimization and Applications* (Plenum Publishing, New York, 1996) 283–295.
[19] I. Muchnik and L. Yampolsky, A pilot study of the concurrent validity of logical analysis of data, as applied to two Beck inventories, RUTCOR Technical Report RTR 02-95, RUTCOR, Rutgers University, 1995.

[20] S. Muroga, *Threshold Logic and Its Applications* (Wiley-Interscience, New York, 1971).
[21] P.M. Murphy and D.W. Aha, UCI repository of machine learning databases: Machine readable data repository, Department of Computer Science, University of California, Irvine, 1994.
[22] J.R. Quinlan, Induction of decision trees, *Machine Learning* 1 (1986) 81–106.
[23] J.R. Quinlan and R. Rivest, Inferring decision trees using minimum description length principle, *Information and Computation* 80 (1989) 227–248.
[24] L.G. Valiant, A theory of the learnable, *Communications of the ACM* 27 (1984) 1134–1142.