RUTCOR
RESEARCH
REPORT

The Maximum Box Problem and its Application to Data Analysis

Jonathan Eckstein[a] Peter L. Hammer[b] Ying Liu[c]
Mikhail Nediak[d] Bruno Simeone[e]

RRR 4-2002, January, 2002

RUTCOR
Rutgers Center for
Operations Research
Rutgers University
640 Bartholomew Road
Piscataway, New Jersey
08854-8003
Telephone: 732-445-3804
Telefax: 732-445-5472
Email: rrr@rutcor.rutgers.edu
http://rutcor.rutgers.edu/~rrr

[a]Rutgers Business School and RUTCOR, Rutgers University, 640 Bartholomew Road, Piscataway, NJ 08854 USA. E-mail: jeckstei@rutcor.rutgers.edu

[b]RUTCOR, Rutgers University, 640 Bartholomew Road, Piscataway, NJ 08854 USA. E-mail: hammer@rutcor.rutgers.edu

[c]RUTCOR, Rutgers University, 640 Bartholomew Road, Piscataway, NJ 08854 USA. E-mail: yingliu@rutcor.rutgers.edu

[d]RUTCOR, Rutgers University, 640 Bartholomew Road, Piscataway, NJ 08854 USA. E-mail: msnediak@rutcor.rutgers.edu

[e]Department of Statistics, "La Sapienza" University, Piazzale Aldo Moro 5, 00185 Rome Italy. E-mail: marsalis@rosd.sta.uniroma1.it

Rutcor Research Report
RRR 4-2002, January, 2002

The Maximum Box Problem and its Application to Data Analysis

Jonathan Eckstein Peter L. Hammer Ying Liu Mikhail Nediak
Bruno Simeone

Abstract. Given two finite sets of points X^+ and X^- in $\mathbb{R}^n$, the maximum box problem consists in finding an interval ("box") $B = \{x : l \leq x \leq u\}$ such that $B \cap X^- = \emptyset$, and the cardinality of $B \cap X^+$ is maximized. A simple generalization can be obtained by instead maximizing a weighted sum of the elements of $B \cap X^+$. While polynomial for any fixed n, the maximum box problem is NP-complete in general. We construct an efficient branch-and-bound algorithm for this problem and apply it to a standard problem in data analysis. We test this method on nine data sets, seven of which are drawn from the UCI standard machine learning repository.

Acknowledgements: This work was supported in part by NSF grants CCR-9902092 and DMS-9806389, and ONR grant N00014-92-J-1375. The work of the third and fourth authors was also supported in part by DIMACS.

1 Introduction

A *box* is simply an interval of $\mathbb{R}^n$, i.e. a set of the form $B = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}\}$, $\mathbf{l}, \mathbf{u} \in \mathbb{R}^n$. The concept of a box generalizes the concept of *pattern* used in the logical analysis of data (LAD) [7], and for the purposes of this paper we will consider the two terms synonymous. The methodology of LAD was originally developed for the analysis of binary data, viewed as vertices of the binary cube, and was generalized in [4] to general numerical data. The method proposed there associates multiple binary variables with each numerical variable, each indicating whether a certain threshold has been exceeded. In this paper, numerical data are handled directly, without introducing such level indicators.

In numerous data analysis problems, one is given two (usually disjoint) sets of points $X^+, X^- \subset \mathbb{R}^n$, and one would like to find patterns which intersect exactly one of these sets. A systematic study of criteria for selecting the most useful patterns for classification of data using LAD was presented in [8]. It was shown there that the best results are obtained by using *strong* (*i.e.* inclusion-wise maximal) boxes.

In this paper, we further strengthen this concept, by considering *maximum* boxes, *i.e.* boxes containing a maximum number of points in the given dataset. More precisely, given two finite sets of points $X^+, X^- \subset \mathbb{R}^n$, we consider the problem of finding a box B, such that the intersection $B \cap X^+$ has the maximum possible cardinality, subject to the constraint $X^- \cap B = \emptyset$. It will be seen that these boxes provide a valuable tool for data analysis, thus reinforcing the conclusions of [6] about the special role of "large" patterns in LAD.

Section 2 defines the terminology to be used, and also states a weighted version of the problem of finding a maximum box. Section 3 considers the complexity of this problem, which is shown to be polynomial for fixed n, but $\mathcal{NP}$-complete in the general case. Section 4 gives an integer programming formulation of the weighted problem, which inspires the branch-and-bound algorithm described in Section 5. Section 6 gives some further generalizations of the problem and describes how the branch-and-bound method can be modified to solve them.

Finally, Section 7 applies the branch-and-bound method to obtain maximum boxes for several standard benchmark data sets used in machine learning. We evaluate the performance of maximum boxes and conclude that they can be obtained efficiently and have strong robustness features, making them extremely useful in data analysis.

2 Formal definition and terminology

Let $\mathbf{l} = (l_1, \ldots, l_n), \mathbf{u} = (u_1, \ldots, u_n)$ be two vectors in $\mathbb{R}^n$ with $\mathbf{l} \leq \mathbf{u}$, *i.e.*, $l_i \leq u_i$ for $i = 1, \ldots, n$. We define a (closed) *box* to be the set

$$[\mathbf{l}, \mathbf{u}] = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}\}$$

Let us denote the set of those points of $\mathbf{X}$ which belong to the box $[\mathbf{l}, \mathbf{u}]$ by

$$X[\mathbf{l}, \mathbf{u}] = \{\mathbf{a} \in X : \mathbf{l} \leq \mathbf{a} \leq \mathbf{u}\}$$

Consider a set $X = \{\mathbf{a}_1, \ldots, \mathbf{a}_m\} \subseteq \mathbb{R}^n$, where $\mathbf{a}_i = (a_{i1}, \ldots, a_{in}), (i = 1, \ldots, m)$, and define its componentwise minimum and maximum as being the respective vectors

$$\begin{aligned} \min X &= (\min_i a_{i1}, \ldots, \min_i a_{in}) \\ \max X &= (\max_i a_{i1}, \ldots, \max_i a_{in}). \end{aligned}$$

We say that a box B is *spanned* by the set X if $B = [\min X, \max X]$.

Suppose we are given two finite sets $X^+, X^- \subseteq \mathbb{R}^n$, which are assumed in most applications to be disjoint. The sets X^+ and X^-, as well as their elements, are called *positive* and *negative* respectively. A box $[\mathbf{l}, \mathbf{u}]$ is called *homogeneous* with respect to X^- if $X^-[\mathbf{l}, \mathbf{u}] = \emptyset$. We define the

Maximum Box Problem: Find a box $[\mathbf{l}, \mathbf{u}]$, homogeneous with respect to X^-, which maximizes the cardinality of $X^+[\mathbf{l}, \mathbf{u}]$.

A slightly generalized version of the above problem is the

Weighted Maximum Box Problem: Given a function $c : X^+ \to \mathbb{R}_+$, find a box $[\mathbf{l}, \mathbf{u}]$, homogeneous with respect to X^-, such that

$$c(X^+[\mathbf{l}, \mathbf{u}]) = \sum_{\mathbf{z} \in X^+[\mathbf{l}, \mathbf{u}]} c(\mathbf{z})$$

is maximized.

3 Complexity analysis

In both the cardinality and weighted versions of the problem, we may without loss of generality restrict our attention to the boxes spanned by subsets of points in X^+, *i.e.* boxes of the form $[\min X, \max X]$, where $X \subseteq X^+$. Clearly, in the case of such spanned boxes, each component of the vectors $\mathbf{l}$ and $\mathbf{u}$ will be equal to a component of one of the vectors in X^+. Therefore, in order to solve the (weighted) maximum box problem, it is sufficient to examine at most $|X^+|^{2n}$ candidates. For each candidate box B, two calculations have to be performed: first, one has to check whether $B \cap X^- = \emptyset$, and, if so, compute the weight or cardinality of $B \cap X^+$. Clearly, $O(n(|X^+| + |X^-|))$ operations suffice to accomplish these steps. In conclusion,

Proposition 1. *For a fixed n, the (weighted) maximum box problem is in class $\mathcal{P}$.*

In order to address the general case, recall the

Maximum Clique Problem: Given a graph $G(V, E)$, find a maximum subset $V' \subseteq V$ such that every pair of vertices in V' are adjacent.

Proposition 2. *The maximum box problem is in class $\mathcal{NPC}$.*

Proof. We shall prove this statement by using a polynomial reduction of the maximum clique problem on the graph $G(V,E)$, known to be $\mathcal{NP}$-complete, to the maximum box problem.

Let the *characteristic vector* $\chi(V') \in \mathbb{R}^{|V|}$ of $V' \subseteq V$ be given by

$$\chi_i(V') = \begin{cases} 1, & i \in V' \\ 0, & i \notin V' \end{cases}$$

We define a set of positive points X^+ and a set of negative points X^- in $\mathbb{R}^{|V|}$ by:

$$\begin{aligned} X^+ &= \{\chi(\{v\}) : v \in V\} \\ X^- &= \{\chi(\{u,v\}) : \{u,v\} \notin E,\ u,v \in V\} \end{aligned}$$

Note that these sets can be generated in time polynomial in $|V|$ and $|E|$.

Consider a homogeneous box $B = [\min X', \max X']$, spanned by a set $X' \subseteq X^+$, containing at least 2 points ($|X'| \geq 2$). Note that

$$\min X' = \mathbf{0},\ \max X' = \chi(X').$$

Consider any $\chi(\{u\}), \chi(\{v\}) \in X'$. Then

$$\min X' \leq \chi(\{u,v\}) \leq \max X'$$

and $\chi(\{u,v\}) \in B$. By homogeneity, $\{u,v\} \in E$. Thus, any homogeneous box corresponds to a clique.

Conversely, let V' be a clique of $G(V,E)$, and let $X' = \{\chi(v) : v \in V'\} \subseteq X^+$. Suppose $B = [\min X', \max X']$ is not homogeneous, *i.e.* there is a negative point $\mathbf{x} \in B$. Since $\mathbf{x} = \chi(\{u,v\})$ for some $\{u,v\} \in \bar{E}$, and

$$\mathbf{0} \leq \chi(\{u\}) \leq \chi(\{u,v\}),\ \mathbf{0} \leq \chi(\{v\}) \leq \chi(\{u,v\})$$

it follows that $\chi(\{u\}), \chi(\{v\}) \in B$. Then $u, v \in V'$ with $(u,v) \notin E$, which contradicts the fact that V' is a clique. Hence, every clique corresponds to a homogeneous box. □

4 Integer programming formulation of the weighted maximum box problem

In order to formulate the weighted maximum box problem as an integer program, let us consider a box B spanned by some set of positive points.

For each coordinate j, let $v_{1j} < \ldots < v_{k_j j}$ be the distinct values taken by the jth coordinate of all the points in X^+. Since it is sufficient to consider spanned boxes of the form $X^+[\mathbf{l}, \mathbf{u}]$, the jth components of the vectors $\mathbf{l}$ and $\mathbf{u}$ may be restricted to the set $\{v_{1j}, \ldots, v_{k_j j}\}$.

For each $j = 1, \ldots, n$, and for each $i = 1, \ldots, k_j$, let the binary variable x_{ij} (y_{ij} respectively) be 1 if $l_j = v_{i_j j}$ ($u_j = v_{i_j j}$, respectively), and 0 otherwise.

With this notation, the box B will be defined by those values of the 0-1 variables x_{ij} and y_{ij} which satisfy the constraints:

$$\sum_{i=1}^{k_j} x_{ij} = 1, \quad j = 1, \ldots, n, \tag{1}$$

$$\sum_{i=1}^{k_j} y_{ij} = 1, \quad j = 1, \ldots, n, \tag{2}$$

Let us denote the elements of X^+ by $\mathbf{a}_t$ with $t = 1, \ldots, |X^+|$, and the elements of X^- by $\mathbf{a}_t$, with $t = |X^+| + 1, \ldots, |X^+| + |X^-|$. In order to distinguish the positive points contained in the box B from those not in B, let us associate now a binary variable z_t to every positive point $\mathbf{a}_t \in X^+$, by defining $z_t = 1$ if $\mathbf{a}_t \in B$, and $z_t = 0$ otherwise. Clearly, if a_{tj} is the jth coordinate of point $\mathbf{a}_t$, then $\mathbf{a}_t \in B$:

$$z_t - \sum_{\{i|v_{ij} \leq a_{tj}\}} x_{ij} \leq 0, \; j = 1, \ldots, n, \; t = 1, \ldots, |X^+| \tag{3}$$

$$z_t - \sum_{\{i|v_{ij} \geq a_{tj}\}} y_{ij} \leq 0, \; j = 1, \ldots, n, \; t = 1, \ldots, |X^+| \tag{4}$$

Similarly, if $\mathbf{a}_t \in X^-, t = 1 + |X^+|, \ldots, |X^+| + |X^-|$, then $\mathbf{a}_t \notin B$ if and only if

$$\sum_{j=1}^{n} \left(\sum_{\{i|v_{ij} > a_{tj}\}} x_{ij} + \sum_{\{i|v_{ij} < a_{tj}\}} y_{ij} \right) \geq 1, \; t = 1 + |X^+|, \ldots, |X^+| + |X^-|; \tag{5}$$

The weighted maximum box problem can be formulated now as the integer program requiring

$$\max \sum_{\mathbf{a}_t \in X^+} c_t z_t \tag{6}$$

under the conditions (1), (2), (3), (4), (5) and

$$x_{ij}, \; y_{ij}, \; z_t \in \{0, 1\} \tag{7}$$

Although we do not use this integer programming formulation directly in the branch-and-bound algorithm presented below, the formulation is helpful in understanding the branching procedure. Also, the continuous relaxation of this integer program provides a potentially useful upper bound on the number of positive points in a box.

5 Branch-and-bound algorithm

To completely specify a branch-and-bound algorithm, we have to first describe the form of the subproblems generated, and then define the following components: the branching rule, the queue discipline and the bounding function. The branching rule determines how a currently-selected problem is decomposed into finer subproblems. The queue discipline determines the order in which problems are selected. Finally, the bounding function determines which subproblems cannot produce a solution better than the incumbent, and should be pruned.

5.1 Subproblem representation

Let us consider two pairs of vectors $\underline{\mathbf{l}} \leq \bar{\mathbf{l}}$ and $\underline{\mathbf{u}} \leq \bar{\mathbf{u}}$. The family $P = (\underline{\mathbf{l}}, \bar{\mathbf{l}}, \underline{\mathbf{u}}, \bar{\mathbf{u}})$ of all homogeneous boxes $[\mathbf{l}, \mathbf{u}]$ such that

$$\underline{\mathbf{l}} \leq \mathbf{l} \leq \bar{\mathbf{l}},\ \underline{\mathbf{u}} \leq \mathbf{u} \leq \bar{\mathbf{u}}. \tag{8}$$

will be called a *subproblem*, and any box $[\mathbf{l}, \mathbf{u}]$ in this family will be said to be *covered* by P. We define *solving* $P = (\underline{\mathbf{l}}, \bar{\mathbf{l}}, \underline{\mathbf{u}}, \bar{\mathbf{u}})$ to be equivalent to solving the original integer program (1)-(7) with the additional constraints

$$x_{ij} = 0 \text{ for all those } i \text{ for which } v_{ij} < \underline{l}_j \text{ or } v_{ij} > \bar{l}_j, j = 1, \ldots, n, \tag{9}$$

$$y_{ij} = 0 \text{ for all those } i \text{ for which } v_{ij} < \underline{u}_j \text{ or } v_{ij} > \bar{u}_j, j = 1, \ldots, n. \tag{10}$$

Initially, at the root of the branch-and-bound tree, we consider the subproblem

$$P_0 = (\min X^+, \max X^+, \min X^+, \max X^+).$$

P is considered to be *feasible* if $\underline{\mathbf{l}} \leq \bar{\mathbf{u}}$, $|X^+[\underline{\mathbf{l}}, \bar{\mathbf{u}}]| > 0$, $|X^-[\bar{\mathbf{l}}, \underline{\mathbf{u}}]| = 0$, $\underline{\mathbf{l}} \leq \bar{\mathbf{l}}$, and $\underline{\mathbf{u}} \leq \bar{\mathbf{u}}$.

5.2 Branching rule

Given a subproblem $P = (\underline{\mathbf{l}}, \bar{\mathbf{l}}, \underline{\mathbf{u}}, \bar{\mathbf{u}})$, if there exists an $\mathbf{a} \in X^-[\underline{\mathbf{l}}, \bar{\mathbf{u}}]$, then $[\underline{\mathbf{l}}, \bar{\mathbf{u}}]$ is not homogeneous, and therefore, any homogeneous box must be located strictly on one side or the other of $\mathbf{a}$ in at least one dimension. As an example, we show in Figure 1 the eight possible positions of a homogeneous box in two dimensions, relative to a negative point $\mathbf{a}$. We shall use such a negative point $\mathbf{a}$ as the basis for splitting the subproblem $P = (\underline{\mathbf{l}}, \bar{\mathbf{l}}, \underline{\mathbf{u}}, \bar{\mathbf{u}})$.

The same observation can be made from the point of view of the IP formulation. Consider a constraint in (5) that corresponds to the point $\mathbf{a}$. According to this constraint, at least one of the quantities

$$p_j = \sum_{\{i | v_{ij} > a_j\}} x_{ij}, \quad q_j = \sum_{\{i | v_{ij} < a_j\}} y_{ij}, \quad j = 1, \ldots, n$$

must be equal to one for every feasible solution to the integer program (or, equivalently, for every spanned homogeneous box).

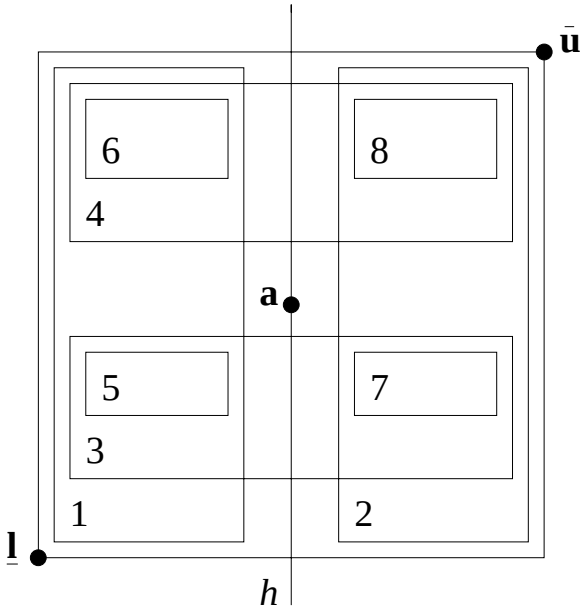

Figure 1: Eight possible positions of a homogeneous box in 2 dimensions with respect to a negative point **a**.

We will subdivide the problem $P = (\underline{\mathbf{l}}, \bar{\mathbf{l}}, \underline{\mathbf{u}}, \bar{\mathbf{u}})$ by assuming for each $k = 1, \ldots, n$ that

$$p_j = q_j = 0 \text{ for all } j < k \tag{11}$$

and that either $p_k = 1$ or $q_k = 1$. The subproblems obtained in this way will be denoted P_{a}^k (above in the kth coordinate) and P_{b}^k (below in the kth coordinate).

To obtain a description of these subproblems based on the bounds $\underline{\mathbf{l}}, \bar{\mathbf{l}}, \underline{\mathbf{u}}, \bar{\mathbf{u}}$ on the box vectors $\mathbf{l}$ and $\mathbf{u}$, we note that condition (11) is equivalent to the requirement that $l_j \leq a_j$, and that $u_j \geq a_j$ for $j < k$. Thus, we define a new vector $\bar{\mathbf{l}}^k$ to replace $\bar{\mathbf{l}}$ by putting

$$\bar{l}_j^k = \begin{cases} \min\{\bar{l}_j, a_j\}, & j < k \\ \bar{l}_j, & j \geq k \end{cases}$$

Similarly, we define a new vector $\underline{\mathbf{u}}^k$ to replace $\underline{\mathbf{u}}$ by putting

$$\underline{u}_j^k = \begin{cases} \max\{\underline{u}_j, a_j\}, & j < k \\ \underline{u}_j, & j \geq k \end{cases}$$

On the other hand, the requirement $q_k = 1$, is equivalent to $u_k < a_k$. Since we restrict our attention to spanned boxes we can, for each $k = 1, \ldots, n$, tighten the lower bounds on $\mathbf{l}$ and the upper bounds on $\mathbf{u}$ to

$$\begin{aligned} \underline{\mathbf{l}}_{\mathrm{b}}^k &= \min(X^+[\underline{\mathbf{l}}, \bar{\mathbf{u}}] \cap \{\mathbf{x} : x_k < a_k\}), \\ \bar{\mathbf{u}}_{\mathrm{b}}^k &= \max(X^+[\underline{\mathbf{l}}, \bar{\mathbf{u}}] \cap \{\mathbf{x} : x_k < a_k\}), \end{aligned}$$

thus defining the subproblem $P_{\rm b}^k = (\underline{\mathbf{l}}_{\rm b}^k, \bar{\mathbf{l}}^k, \underline{\mathbf{u}}^k, \bar{\mathbf{u}}_{\rm b}^k)$ which is "below $\mathbf{a}$ in the axis k".

Similarly, we can tighten the bounds

$$\begin{aligned} \underline{\mathbf{l}}_{\rm a}^k &= \min(X^+[\underline{\mathbf{l}}, \bar{\mathbf{u}}] \cap \{\mathbf{x} : x_k > a_k\}), \\ \bar{\mathbf{u}}_{\rm a}^k &= \max(X^+[\underline{\mathbf{l}}, \bar{\mathbf{u}}] \cap \{\mathbf{x} : x_k > a_k\}) \end{aligned}$$

defining the subproblem $P_{\rm a}^k = (\underline{\mathbf{l}}_{\rm a}^k, \bar{\mathbf{l}}^k, \underline{\mathbf{u}}^k, \bar{\mathbf{u}}_{\rm a}^k)$ which is "above $\mathbf{a}$ in the axis k".

Our branching rule is to divide $P = (\underline{\mathbf{l}}, \bar{\mathbf{l}}, \underline{\mathbf{u}}, \bar{\mathbf{u}})$ into the set of all $P_{\rm a}^k$ and $P_{\rm b}^k$, $k = 1, \ldots, n$ that are feasible. If there is no $\mathbf{a} \in X^-[\underline{\mathbf{l}}, \bar{\mathbf{u}}]$, we stop with a homogeneous box $[\underline{\mathbf{l}}, \bar{\mathbf{u}}]$.

The following proposition immediately follows from our choice of the branching rule:

Proposition 3. *Consider a subproblem $P = (\underline{\mathbf{l}}, \bar{\mathbf{l}}, \underline{\mathbf{u}}, \bar{\mathbf{u}})$ and a spanned homogeneous box $[\mathbf{l}, \mathbf{u}]$ covered by it. For any negative point $\mathbf{a} \in X^-[\underline{\mathbf{l}}, \bar{\mathbf{u}}]$, the box $[\mathbf{l}, \mathbf{u}]$ occurs in exactly one of the subproblems $P_{\rm b}^k$, $P_{\rm a}^k$, $k = 1, \ldots, n$.*

The only part of the branching rule left to specify is the choice of the negative point $\mathbf{a}$, which will be discussed in the next section. Note also that the ordering of coordinates does not affect the correctness of the proposed algorithm, and may be changed with each branching calculation.

5.3 Upper bound function

Our branch and bound method requires an *upper bound* $U(\underline{\mathbf{l}}, \bar{\mathbf{l}}, \underline{\mathbf{u}}, \bar{\mathbf{u}})$ on the total weight $c(X^+[\mathbf{l}, \mathbf{u}])$ of the positive points in the box $[\mathbf{l}, \mathbf{u}]$, satisfying

(i) $c(X^+[\mathbf{l}, \mathbf{u}]) \le U(\underline{\mathbf{l}}, \bar{\mathbf{l}}, \underline{\mathbf{u}}, \bar{\mathbf{u}})$ for any homogeneous $[\mathbf{l}, \mathbf{u}]$ covered by P

(ii) $c(X^+[\underline{\mathbf{l}}, \bar{\mathbf{u}}]) = U(\underline{\mathbf{l}}, \bar{\mathbf{l}}, \underline{\mathbf{u}}, \bar{\mathbf{u}})$ if $[\underline{\mathbf{l}}, \bar{\mathbf{u}}]$ is homogeneous.

There are many possible upper bound functions U. An easily-computable choice is

$$U(\underline{\mathbf{l}}, \bar{\mathbf{l}}, \underline{\mathbf{u}}, \bar{\mathbf{u}}) = \begin{cases} c(X^+[\underline{\mathbf{l}}, \bar{\mathbf{u}}]), \text{ if } [\underline{\mathbf{l}}, \bar{\mathbf{u}}] \text{ is homogeneous} \\ c(X^+[\underline{\mathbf{l}}, \bar{\mathbf{u}}]) - \min\{c(\mathbf{z}) : \mathbf{z} \in X^+[\underline{\mathbf{l}}, \bar{\mathbf{u}}]\}, \quad \text{otherwise} \end{cases} \tag{12}$$

However, this bound is not very tight, and leads to a large number of subproblems. Another possible upper bound is

$$U(\underline{\mathbf{l}}, \bar{\mathbf{l}}, \underline{\mathbf{u}}, \bar{\mathbf{u}}) = \min_{\mathbf{a} \in X^-[\underline{\mathbf{l}}, \bar{\mathbf{u}}]} \left\{ \max_{j=1,\ldots,n} \left\{ \max \left\{ \begin{array}{l} c(X^+[\underline{\mathbf{l}}, \bar{\mathbf{u}}] \cap \{\mathbf{x} : x_j < a_j\}), \\ c(X^+[\underline{\mathbf{l}}, \bar{\mathbf{u}}] \cap \{\mathbf{x} : x_j > a_j\}) \end{array} \right\} \right\} \right\} \tag{13}$$

This formula is based on the observation that, for any $\mathbf{a} \in X^-[\underline{\mathbf{l}}, \bar{\mathbf{u}}]$, the weight of a homogeneous box covered by $P = (\underline{\mathbf{l}}, \bar{\mathbf{l}}, \underline{\mathbf{u}}, \bar{\mathbf{u}})$ can not exceed the value of the positive sets corresponding to the subproblems of P. Moreover, one can take the minimum over $X^-[\underline{\mathbf{l}}, \bar{\mathbf{u}}]$, since any one of the negative points can be selected for branching.

We tested this upper bound function and found it to be very useful. In practice, we further tighten the bound by restricting the formula to those sets $[\underline{\mathbf{l}}, \bar{\mathbf{u}}] \cap \{\mathbf{x} : x_k < a_k\}$ and $[\underline{\mathbf{l}}, \bar{\mathbf{u}}] \cap \{\mathbf{x} : x_k > a_k\}$ that correspond to feasible subproblems.

Another advantage of using this upper bound is that it provides a natural choice of the negative point to be used in the branching rule. In our experiments, we branched on a negative point that attained the minimum in the function (13). Note that in this technique, we essentially perform some exploratory bounding work on all the subproblems that could arise from every possible separation of the current subproblem, and use the resulting information to optimize the branching decision. This kind of approach, which can also be applied to a heuristically-selected candidate list of possible branches rather than every possibility, is sometimes referred to as a *strong branching* (see for example [2]).

Finally, one could also use an upper bound of the form

$$U(\underline{\mathbf{l}}, \bar{\mathbf{l}}, \underline{\mathbf{u}}, \bar{\mathbf{u}}) = LP(\underline{\mathbf{l}}, \bar{\mathbf{l}}, \underline{\mathbf{u}}, \bar{\mathbf{u}}) \tag{14}$$

where $LP(\underline{\mathbf{l}}, \bar{\mathbf{l}}, \underline{\mathbf{u}}, \bar{\mathbf{u}})$ is the optimal value of the continuous relaxation of the integer program (1)-(7) with the additional constraints (9) and (10). We leave computational experiments using this bound within our branching procedure for future research. We did experiment briefly with applying a commercial mixed integer programming solver directly to the formulation (1)-(7), using its default branching rules, and the initial results were not very promising.

5.4 Queue discipline, including initial diving

A standard queue discipline is *best-first search*, which in this case would always select a subproblem with the highest possible value of $U(\underline{\mathbf{l}}, \bar{\mathbf{l}}, \underline{\mathbf{u}}, \bar{\mathbf{u}})$. This choice, however, turned out to be unsuitable for the application, leading to an explosive growth of the list of active subproblems because of our method's high branching factor. Another possibility is to select a subproblem with the lowest cardinality of $X^-[\underline{\mathbf{l}}, \bar{\mathbf{u}}]$; in this case, ties can be broken by selecting a subproblem with the highest upper bound. We tested this approach, and it performed reasonably well. An alternative approach is to select the subproblem with a highest value of $|X^+[\underline{\mathbf{l}}, \bar{\mathbf{u}}]|/|X^-[\underline{\mathbf{l}}, \bar{\mathbf{u}}]|$. Such a subproblem is relatively likely to produce a large homogeneous box — a conclusion confirmed through computational experimentation.

Since we do not use a best-first queue discipline, our algorithm may examine non-critical subproblems, that is, subproblems whose bounds are worse than those of the optimal solution. This phenomenon can slow down the algorithm, and, since we do not use a pure depth-first strategy, may also waste memory. It is therefore important to identify a good feasible solution early in the run, permitting early pruning of the search tree.

To identify good feasible solutions early in the run, we use a form of "diving", altering the queue discipline used at the outset of the run; this kind of technique is a traditional one in mixed-integer programming [1]. When our algorithm starts, it uses a depth-first strategy, selecting the subproblem with largest upper bound on the lowest level of the search tree. When all feasible subproblems on the lowest level correspond to homogeneous

boxes, we revert to the principal queue discipline, either minimizing $|X^-[\underline{\mathbf{l}}, \bar{\mathbf{u}}]|$ or maximizing $|X^+[\underline{\mathbf{l}}, \bar{\mathbf{u}}]|/|X^-[\underline{\mathbf{l}}, \bar{\mathbf{u}}]|$.

6 Generalizations

In the previous section, we have dealt with the problem of finding a maximum homogeneous box, *i.e.*, one which contains no negative point. In some practical applications, this condition of homogeneity could be replaced by one of "relative homogeneity", *i.e.*, a condition which allows a limited number of negative points to be contained in a positive box.

If r is a nonnegative integer, we shall say that a box $[\mathbf{l}, \mathbf{u}]$ is *r-homogeneous* if $|X^-[\mathbf{l}, \mathbf{u}]| \leq r$. Similarly, if α is a nonnegative real number, we shall say that a box $[\mathbf{l}, \mathbf{u}]$ is *α-relatively homogeneous* if $|X^-[\mathbf{l}, \mathbf{u}]| \leq \alpha |X^+[\mathbf{l}, \mathbf{u}]|$. It is natural to generalize the weighted maximum box problem to that of finding an r-homogeneous or an α-relatively homogeneous box $[\mathbf{l}, \mathbf{u}]$ such that $c(X^+[\mathbf{l}, \mathbf{u}])$ is maximized. These generalized problems will be seen below to lead naturally to integer programming formulations.

By introducing, besides the binary variable z_k $(k = 1, \ldots, |X^+|)$ associated with the positive points, additional binary variables z_k $(k = |X^+| + 1, \ldots, |X^+| + |X^-|)$ associated with the negative points, the constraint (5) becomes:

$$z_k + \sum_{j=1}^{n} \left(\sum_{\{i | v_{ij} > a_{kj}\}} x_{ij} + \sum_{\{i | v_{ij} < a_{kj}\}} y_{ij} \right) \geq 1, \quad k = |X^+| + 1, \ldots, |X^+| + |X^-| \tag{15}$$

We also include the constraint

$$\sum_{i=|X^+|+1}^{|X^+|+|X^-|} z_i \leq r \tag{16}$$

in the case of r-homogeneous boxes, and the constraint

$$\sum_{i=|X^+|+1}^{|X^+|+|X^-|} z_i \leq \alpha \sum_{i=1}^{|X^+|} z_i \tag{17}$$

in the case of α-relatively homogeneous boxes.

The integer programming formulations of the generalizations consist of (1)-(4), (6), and (7), with the conditions (15), (16) or (15), (17) replacing (5).

Our algorithm can also be modified to solve these generalized problems. At every branching step, besides the subproblems discussed previously, there will be one additional subproblem containing the negative branching point. The set of all negative points that must be inside an r-homogeneous (α-relatively homogeneous) box covered by the subproblem $P = (\underline{\mathbf{l}}, \bar{\mathbf{l}}, \underline{\mathbf{u}}, \bar{\mathbf{u}})$ can be found as $X^-[\bar{\mathbf{l}}, \underline{\mathbf{u}}]$ if $\bar{\mathbf{l}} \leq \underline{\mathbf{u}}$, and is empty otherwise. A subproblem with $p = |X^-[\bar{\mathbf{l}}, \underline{\mathbf{u}}]| > r$ will be called *infeasible* and discarded. In this case too, we shall still branch on a negative point $\mathbf{a} \in X^-[\underline{\mathbf{l}}, \bar{\mathbf{u}}]$, but in addition to the $2n$ children described in

Section 5.2, we shall also consider, if $p < r$, a single additional subproblem $P = (\underline{\mathbf{l}}, \bar{\mathbf{l}}', \underline{\mathbf{u}}', \bar{\mathbf{u}})$, which is constrained to contain the point $\mathbf{a}$ by the relations

$$\begin{aligned} \bar{\mathbf{l}}' &= \min\{\bar{\mathbf{l}}, \mathbf{a}\} \\ \underline{\mathbf{u}}' &= \max\{\underline{\mathbf{u}}, \mathbf{a}\}. \end{aligned}$$

These relations express that, if $\mathbf{a}$ is in the box $[\mathbf{l}, \mathbf{u}]$ then one must have:

$$\mathbf{l} \leq \mathbf{a},\ \mathbf{u} \geq \mathbf{a}.$$

A similar modification is needed to treat the case of α-relatively homogeneous boxes. In this case, each subproblem $P = (\underline{\mathbf{l}}, \bar{\mathbf{l}}, \underline{\mathbf{u}}, \bar{\mathbf{u}})$ can only be added to the subproblem queue if $p \leq \alpha |X^+[\underline{\mathbf{l}}, \bar{\mathbf{u}}]|$.

Finally, note that one can not use the upper bound function (13) for the subproblems with $p < r$ (or $p < \alpha |X^+[\mathbf{l}, \mathbf{u}]|$ for the α-relatively homogeneous case). Thus, one would have to use the bound of either (12) or (14).

7 An application to data analysis

Consider a data set X in $\mathbb{R}^n$ partitioned into two sets X^+ and X^- of *positive* and *negative* points, respectively. We say that a point in $\mathbb{R}^n$ contained in a pattern is *covered* by this pattern. The *coverage* of a positive (negative) homogeneous pattern is the number of positive (negative) points covered by it. The *domain* of a positive (negative) box is the percentage of the points in X^+ (respectively, X^-) covered by that pattern.

In order to empirically evaluate the efficiency of producing maximum boxes and using them in data analysis, we have constructed the maximum positive and negative patterns for nine frequently used data sets. Seven of these data sets were taken from the repository of the University of California, Irvine (UCI) [3]: Wisconsin breast cancer (`breastcancer`), Australian credit card (`credit`), Pima Indian diabetes (`diabetes`), Cleveland heart disease (`heart`), Boston housing (`housing`), congressional voting records (`voting`) and mushrooms (`mushrooms`). Two further datasets, `oil1` and `oil2`, were provided by the Chevron Corporation [5]. The key parameters of these nine data sets are listed in Table 1.

Table 1 also indicates the number of observations for which some data values are missing. In our experiments, we removed the observations with missing values, with the exception of the `voting` data set, where almost half of the observations contained missing values. In this data set, there are sixteen attributes, and all of them are binary. In this case, we have substituted the missing binary values with the majority value in the corresponding (positive or negative) class. Also, `mushrooms` has 22 categorical attributes, which cannot be directly processed by our method. Therefore, we binarized this data set using the method described in [4]), obtaining 126 binary attributes. For more information, see [3, 5, 9].

Table 2 shows the CPU time and number of subproblems (nodes) explored by our branch and bound method in computing both the maximum positive and maximum negative boxes.

Dataset	Number of Observations			Attributes		Observations with Missing Values	
	+	−	Total	Binary or Categorical	Numerical	Number	Handling
breastcancer	241	458	699	0	9	16	removed
credit	307	383	690	9	6	37	removed
diabetes	268	500	768	0	8	0	
heart	139	164	303	7	6	6	removed
housing	256	250	506	1	13	0	
voting	267	168	435	16	0	203	substituted
mushrooms	3916	4208	8124	22	0	0	
oil1	702	930	1632	0	7	0	
oil2	773	1620	2393	0	7	0	

Table 1: Experimental datasets.

Dataset	Generation					
	Subproblems		Time (Seconds)		Domain(%)	
	+	−	+	−	+	−
breastcancer	5	7	2.54	0.87	50.2	89.5
credit	70	24	20.48	12.76	32.4	39.2
diabetes	352	359	23.87	22.53	19.8	28.0
heart	74	16	2.93	1.98	41.0	49.4
housing	8	30	3.18	2.7	66.0	51.2
voting	5	4	1.23	1.89	72.5	86.5
oil1	7	5	9.04	12.66	76.1	74.5
oil2	65	18	26.15	39.08	38.1	55.8
mushroom	5	4	17	56.65	96.9	93.9

Table 2: Subproblem counts, run times, and maximum box domains

We timed our implementation on a 550 MHz Pentium Xeon II server running RedHat Linux 7.1 and the GNU 2.96 C compiler.

Two main conclusions can be drawn from this table.

- First, the computation time to generate maximum boxes is modest, ranging from below a second to slightly less than a minute, and averaging 14.3 seconds.
- Second, the domains of the maximum positive and negative boxes are remarkably large. Indeed, the maximum positive boxes contain on the average 54.8% of the given positive points, while the maximum negative boxes contain on the average 63.1% of the negative points.

We have also evaluated the robustness of the maximum positive and negative boxes by the *five-fold cross-validation* method. The method consists in dividing the positive and

negative subsets S^+ and S^- of a data set S into five approximately equal parts $S_1^+, \ldots, S_5^+$ and $S_1^-, \ldots, S_5^-$, respectively. Let $T_i^+ = S^+ \setminus S_i^+$ and $T_i^- = S^- \setminus S_i^-$. Now let $T_i^* = T_i^+ \cup T_i^-$ and $S_i^* = S_i^+ \cup S_i^-$. We now apply the algorithm for generating maximum positive and negative boxes to each of the (training) data set T_i^*, $i = 1, \ldots, 5$, validating the results using the respective data set S_i^*.

Table 3 reports average results for this procedure, which give three indications of the robustness of the maximum patterns:

(i) First, the average domains of the maximum positive and negative patterns represent 60.8% of the size of the positive and negative datasets, respectively. The corresponding measurement for the testing set is 57.6%, showing that the domain of the maximum boxes in the testing set is almost unchanged compared with the training set. Note also that the domains reported for the training sets are uniformly slightly larger than the corresponding values in Table 2, most likely due to the smaller data set sizes.

(ii) Obviously, the maximum positive and negative boxes in the training sets are homogeneous, *i.e.* they contain only positive (respectively negative) points. In order to measure to what extent this property is maintained in the testing set, we report in the "homogeneity" columns the fraction of the positive (respectively negative) points contained in the maximum positive (respectively negative) boxes in the testing set. These fractions are extremely high, averaging more than 95.5%.

(iii) Finally, in the "overlap" column, we consider the number of points in the testing set simultaneously belonging to both the maximum positive and negative boxes found using the training set. The almost uniform disjointness of the maximum positive and negative boxes in the testing set gives a third strong argument for their robustness.

The large domain and the high robustness of the maximum boxes clearly indicate their usefulness for data analysis.

References

[1] Evelyn M. L. Beale. Branch and bound methods for mathematical programming systems. *Ann. Discrete Math.*, 5:201–219, 1979. Discrete optimization (Proc. Adv. Res. Inst. Discrete Optimization and Systems Appl., Banff, Alberta, 1977), II.

[2] Robert E. Bixby, Mary Fenelon, Zonghao Gu, Ed Rothberg, and Roland Wunderling. MIP: theory and practice—closing the gap. In *System modeling and optimization (Cambridge, 1999)*, pages 19–49. Kluwer Acad. Publ., Boston, MA, 2000.

[3] Catherine L. Blake and Christopher J. Merz. UCI repository of machine learning databases. http://www.ics.uci.edu/~mlearn/MLRepository.html, 1998. University of California, Irvine, Department of Information and Computer Sciences.

	Domain(%)				Homogeneity(%)		Overlap(%)
	Training		Testing		as % of Testing Set		
Dataset	+	−	+	−	+	−	
breastcancer	52.6	90.4	50.6	90.4	97.3	99.0	0
credit	35.9	44.0	32.1	42.6	93.8	96.1	0
diabetes	21.5	30.8	13.1	30.8	77.9	93.0	0
heart	44.6	51.2	32.3	43.3	86.1	92.7	1
housing	67.2	54.7	60.4	46.8	97.2	92.8	0
voting	74.9	86.5	75.4	86.5	98.4	100.0	0
oil1	77.1	75.1	75.9	72.5	99.0	99.3	0
oil2	39.3	57.9	37.0	57.1	97.2	99.4	0
mushroom	96.9	93.9	96.9	93.9	100	100.0	0

Table 3: Robustness of maximum boxes

[4] Endre Boros, Peter L. Hammer, Toshihide Ibaraki, and Alexander Kogan. Logical analysis of numerical data. *Mathematical Programming*, 79:163–190, 1997.

[5] Endre Boros, Peter L. Hammer, Toshihide Ibaraki, Alexander Kogan, Eddy Mayoraz, and Ilya Muchnik. An implementation of logical analysis of data. *IEEE Transactions of Knowledge and Data Engineering*, 12(2):292–306, 2000.

[6] Endre Boros, Toshihide Ibaraki, L. Shi, and M. Yagiura. Generating all "good" patterns in polynomial expected time. Lecture at the *6th International Symposium on Artificial Intelligence and Mathematics*, Ft. Lauderdale, Florida, January 2000.

[7] Yves Crama, Peter L. Hammer, and Toshihide Ibaraki. Cause-effect relationships and partially defined Boolean functions. *Annals of Operations Research*, 16:299–325, 1988.

[8] Peter L. Hammer, Alexander Kogan, Bruno Simeone, and Sandor Szedmak. Pareto-optimal patterns in logical analysis of data. RUTCOR Research Report 7-2001, 2001.

[9] Tjen-Sien Lim, Wei-Yin Loh, and Yu-Shan Shih. A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. *Machine Learning*, 40:203–228, 2000.