UNIVERSITÀ
DEGLI STUDI
DI UDINE

Università degli studi di Udine

Classifying negative and positive points by optimal box clustering

Original

Availability:
This version is available http://hdl.handle.net/11390/871654 since

Publisher:

Published
DOI:10.1016/j.dam.2013.05.003

Terms of use:
The institutional repository of the University of Udine (http://air.uniud.it) is provided by ARIC services. The aim is to enable open access to all the world.

Publisher copyright

(Article begins on next page)

25 April 2024

Classifying negative and positive points by optimal box clustering

Paolo Serafini
Università di Udine, Dipartimento di Matematica e Informatica
Via delle Scienze 206. 33100 Udine, Italy

Abstract. In this paper we address the problem of classifying positive and negative data with the technique known as box clustering. A box is homogeneous if it contains only positive (negative) points. Box clustering means finding a family of homogeneous boxes jointly containing all and only positive (negative) points. We first consider the problem of finding a family with the minimum number of boxes. Then we refine this problem into finding a family which not only consists of the minimum number of boxes but also the points are covered as many times as possible by the boxes in the family. We call this problem the maximum redundancy problem. We model both problems as set covering problems with column generation. The pricing problem is a maximum box problem. Although this problem is NP-hard, there is available in the literature a combinatorial algorithm which performs well. Since the pricing has to be carried out also in the branch-and-bound search of the set covering problem we consider also how the pricing has to be modified to take care of the branching constraints. The computational results show a good behavior of the set covering approach.

1 Introduction

A fundamental task in data analysis is the classification of data. Most often the information gathered from an individual (whatever it may be) is a set of real numbers and the classification consists in deciding whether the data related to this individual belongs to one class or to a complementary class. We may denote these two classes as 'positive' and 'negative'. There is a wide range of applications which can be framed into a model of this type.

Since the data related to an individual is an array of numbers, it is natural to associate to the individual a point in a vector space. If the data are meaningful for the investigation for which they are collected, it is expected that the points in the same class are more or less clustered and separated from the points of the other class, thus defining two regions of space. An initial known 'training' set of data is used to identify these regions of space. Then new data are classified on the basis of the defined regions.

One of the oldest classification methods is based on the idea that the positive and negative points can be separated by a hyperplane [10]. If this is the case the hyperplane is first identified from the training set via mathematical programming techniques and then used for further classification of new data. The idea is simple and powerful. However, it is quite frequent that the data are not linearly separable. The use of nonlinear surfaces to separate points can be pursued (see again [10]). However, the choice of a suitable nonlinear function seems not only technically hard but also difficult to be justified. A piecewise linear separation has been proposed in [1] when the data are not linearly separable, but the method does not always produce satisfactory results.

Instead of defining a region of space according to the sign of a function (analytically or algorithmically defined), a different approach consists in defining a region as the union of simple and similar sets. In other words, a family of similar sets is defined such that each positive data of the training set is covered by at least

one member of the family while no negative data of the training set is covered by any set of the family. An analogous family is defined for the negative data.

For this approach to be useful some requirements have to be considered. For instance, a trivial family consists of very small sets each one covering exactly one point of the training set. Obviously, new data could be classified only if they are a replica of some data in the training set and this situation is meaningless for the investigation. Intuitively, we would like to have large sets covering many points, so that new data which are close to some point in the training set can easily fit into the set. At the same time the sets should be 'essential', in the sense that it is worth extending a set only if new points have to be covered. Having a family of large and essential sets is close to have a family with few sets. Hence we may consider the problem of covering the points with the minimum number of sets. However, it seems useful for the classification to have available some measure of 'trust' for the new data. To this aim not only we would like to cover the points with the minimum number of sets, but we would like also to have many points, i.e., many regions of space covered by more than one box. This kind of redundancy can constitute a measure of trust for the classification of new data.

The idea of covering the training set of points can in principle work with any family of subsets. However, subsets which would seem most suited for this analysis, like ellipsoids, present relevant computational problem. Perhaps the best family from the computational point of view is the family of boxes. Although boxes are far from being smooth sets (smoothness seems a natural requirement for identifying a region) they can be represented with a minimum amount of information and this simplicity can lead to viable algorithms. Box clustering has been proposed and described in [9].

Moreover, there is a strong rationale behind the idea of using boxes. If we think of medical data, the result of an analysis is usually framed as a number belonging to some interval. The interval represents a set of values denoting a healthy status. Considering all data together, an individual is declared healthy if the point representing all measurements is within the high dimensional box given by the euclidean product of the single intervals. This is a simplified picture since it assumes the data are not correlated. In case they are, more than one box may be needed to cover the points. A more complex picture can be called for the data referring to sick individuals since we do not expect these data to fit into one box.

In this paper we want to find a covering of the positive (or the negative) points with the minimum number of boxes and also with maximum redundancy. The problem is modeled as a set covering problem with column generation, where each column is associated to a particular box. It turns out that the pricing problem consists in finding a single box of maximum weight. This last problem has been fully investigated in [6] and a viable combinatorial algorithm has been proposed. We base the set covering problem on this pricing algorithm. The problem of finding a minimum cardinality box cover in the bidimensional case has been investigated in [2] providing interesting complexity and algorithmic results.

In Section 2 we define formally the problem and provide some computational complexity results. In Section 3 we develop the set covering model with column generation. The corresponding pricing problem is described in Section 4 and the problem of introducing branching constraints in the pricing is dealt with in Section 5. In Section 6 we report some computational results applied to artificial data and to real data. Finally in Section 7 we provide some conclusions.

2 Statement of the problem

There are given p *positive* points $X^i \in R^n$ $(i \in [p])$ and q *negative* points $Y^i \in R^n$, $(i \in [q])$. We use the notation $[h] := \{i \in Z : 1 \leq i \leq h\}$. We are interested in sets covering either only positive points or only negative points. In this paper we consider only boxes as possible covering sets. Although models with different types of sets can be envisaged in general, we do not pursue this direction which, very likely, entails considerable difficulties. A *box* $B(\ell, u)$ is the set

$$\{X \in R^n : \ell_k \leq X_k \leq u_k,\, k \in [n]\}\,.$$

A box is *positive* (*negative*) if it covers, i.e., it contains, only positive (negative) points. A box which is either positive or negative is called *homogeneous*. With abuse of notation we denote by $|B|$ the number of points contained in the homogenous box B.

A family of boxes $\{B_j\}_{j\in J}$ is *positive* (*negative*) if every positive (negative) point is contained in at least one positive (negative) box, i.e., a positive family $\{B_j\}_{j\in J}$ satisfies

$$X^i \in \bigcup_{j\in J} B_j, \quad i \in [p], \qquad Y^i \notin \bigcup_{j\in J} B_j, \quad i \in [q],$$

and a negative family $\{B_j\}_{j\in J'}$ satisfies

$$X^i \notin \bigcup_{j\in J'} B_j, \quad i \in [p], \qquad Y^i \in \bigcup_{j\in J'} B_j, \quad i \in [q].$$

Positive and negative families are called homogeneous.

For a given subset S of points, the *box-closure* of S, denoted as $[S]$, is the smallest box containing all points in S, i.e.,

$$[S] := \left\{ Z \in R^n : \min_{X\in S} X_k \leq Z_k \leq \max_{X\in S} X_k,\, k \in [n] \right\}$$

A box is *essential* if no upper box coordinate can be decreased and no lower box coordinate can be increased without reducing the number of covered points. Clearly, a box-closure is essential. It makes sense to limit the search to essential boxes. The coordinates (both lower and upper) of an essential box take values in $\left\{X_k^1, X_k^2, \ldots, X_k^p\right\}$, for $k \in [n]$ (but not vice-versa, boxes with these coordinates are not necessarily essential). As a consequence the total number of positive boxes is upper bounded by $(p\,(p-1)/2)^n$. For a fixed n (and in particular for $n = 2$) this is a polynomial number of boxes.

An essential homogeneous box is *maximal* if it is not properly contained in any other essential homogenous box.

Among the homogeneous families we may wish to find a family with some special properties. Since the problems for positive and negative families are equivalent (it is only matter of exchanging the roles of positive and negative) we simply present the problems for the positive families.

The positive family is built with the purpose of classifying future data with the highest accuracy and therefore we would like the family to have the property that all future positive points are covered by the

family and that all future negative points do not fall in the family. Clearly, this is a goal that in general cannot be fulfilled and also cannot be pursued directly because future points are not known in advance. However, it can be pursued indirectly under the reasonable assumption that future points are similar to the training set (i.e., the points X^i and Y^i).

A first simple desirable property for a family is to consist of the smallest number of boxes. The motivation for such a property does not come from minimality itself, although one could argue that pursuing a minimum amount of information to explain the data is just an application of Occam's Razor principle (see for instance the discussion in [8] supporting the choice of minimality). Rather it relies on the fact that, in general, a small number of boxes corresponds to large boxes and having large boxes is preferable than small ones for the subsequent classification of new data. We call the problem of finding a minimum cardinality family the *cardinality problem*. The cardinality problem has been also proposed in [9] together with a heuristic procedure for its solution.

In addition, we may consider a desirable property for a family if there are many points covered by many boxes. Hence we want to find a positive family of boxes with minimum cardinality such that the total number of positive points covered by the boxes and counted with multiplicity is maximized. We may call *redundancy* r_i of the point i the number of boxes covering i and *redundancy* R *of the family* the sum of the redundancies over all points, i.e.,

$$R = \sum_{i \in [p]} r_i = \sum_{j \in J} |B_j|$$

We call the problem of finding a minimum cardinality family with maximum redundancy the *max redundancy problem*.

Actually, since each point must be covered by at least one box, the redundancy of the point i would be more correctly defined as $r_i - 1$. Given the fixed number of boxes the two definitions differ by a constant and so there is no real difference in using either one.

Note that a max redundancy family has the property that any box of the family is maximal. In fact, if it were not so, we could increase a non-maximal box by adding points to the box, thereby improving the redundancy. Another property shared by optimal families is that for each box there is at least one point covered by that box only, otherwise the box could be dropped from the family thereby violating the minimum cardinality property.

We may further extend the redundancy problem by assuming that each point has a weight $0 \leq w_i \leq 1$ and the weighted redundancy W is defined as the sum of the weights over all boxes and all covered positive points, i.e.,

$$W = \sum_{j \in J} \sum_{i \in B_j} w_i = \sum_{i \in [p]} w_i \cdot |\{j : i \in B_j\}|.$$

We call the problem of finding a family with minimum cost the *max weighted redundancy problem*.

The stated problems are all NP-hard. This result can be inferred by the result in [2] which proves NP-hardness for the case $n = 2$. Here we provide a simple alternative proof for general n.

Theorem 1. *The cardinality problem is NP-hard.*

Proof: We use the same tools as in [6]. We reduce the problem of finding the chromatic number of a graph $G = (V, E)$ to the problem of finding a minimum cardinality positive family. Given a graph $G = (V, E)$ with $V = \{v_1, \ldots, v_n\}$ we define points in R^n as follows. To each vertex v_i we associate the positive point $X^i \in R^n$ with coordinates

$$X^i_k = \begin{cases} 1 & \text{if } k = i \\ 0 & \text{if } k \neq i \end{cases}$$

and to each edge (v_i, v_j) we associate the negative point Y^{ij} with coordinates

$$Y^{ij}_k = \begin{cases} 1 & \text{if } k = i \text{ or } k = j \\ 0 & \text{otherwise} \end{cases}$$

A positive box cannot cover two positive points X^i and X^j for which there exists the edge (i, j) in G. Hence a positive box covers points corresponding to a stable set in G. Conversely, the box closure of a stable set in G gives raise to a positive box. Hence there is a one-to-one correspondence between stable sets in G and positive boxes. Hence the problem of finding a minimum cardinality family is the exact counterpart of finding a covering of the nodes by a minimum cardinality family of stable sets, i.e., finding the chromatic number of G. ∎

Theorem 2. *The max redundancy problem is NP-hard.*

Proof: A max redundancy family is also one of minimum cardinality. Hence the thesis follows from the previous theorem. ∎

Theorem 3. *The max weighted redundancy problem is NP-hard.*

Proof: The max redundancy problem is just a particular case of the max weighted redundancy problem. ∎

We mention that a two-dimensional box covering problem can be rephrased as a coloring problem on a so called incompatibility graph. An incompatibility graph (for the positive points) has vertices associated to the positive points and an edge (i, j) if and only if the box closure $[\{X^i, X^j\}]$ contains at least one negative point. In dimension two the box closure of an independent set in the incompatibility graph is a positive box (this property is not true for $n > 2$; take for instance $X^1 = (0, 0, 0)$, $X^2 = (1, 1/2, 1/2)$, $X^3 = (1/2, 1, 1)$ and $Y = (3/4, 3/4, 1/4)$; we have $Y \notin [\{X^i, X^j\}]$ for any $i, j \in \{1, 2, 3\}$, so the incompatibility graph is given by three vertices without edges; however, $Y \in [\{X^1, X^2, X^3\}]$).

Hence for $n = 2$ finding a minimum cardinality family corresponds to finding the chromatic number of the incompatibility graph. For the max redundancy problem we have to allow a vertex to be colored with more than one color and a multi-coloring is feasible if two adjacent vertices do not share any of their colors. A max redundancy family is given by a multi-coloring with no more colors than the chromatic number and the maximum possible number of extra colors for all vertices. Rephrasing the problem in terms of graphs, stable sets and coloring allows to translate some graph properties to the box clustering problem. Indeed as observed in [4] the minimum cardinality problem for some special instances with $n = 2$ can be solved in polynomial time. However, as already remarked, the result in in [2] shows that NP-hardness holds also for $n = 2$.

3 A set covering model

Let us assume that all positive boxes are indexed by the finite index set $\mathcal{J}$. Any family of boxes is in a one-to-one correspondence with a particular subset $J \subset \mathcal{J}$. As such we identify families of boxes with subsets of $\mathcal{J}$ and say that a set J is feasible if and only if the corresponding family is positive.

In order to build a 0-1 LP model we define

$$a_i^j = \begin{cases} 1 & \text{if box } B_j \text{ covers } X^i \\ 0 & \text{otherwise} \end{cases}$$

and

$$x_j = \begin{cases} 1 & \text{if box } B_j \text{ is in the positive family to be found} \\ 0 & \text{otherwise.} \end{cases}$$

Then the cardinality problem can be formulated as

$$\begin{aligned} V = \min \ & \sum_{j \in \mathcal{J}} x_j \\ & \sum_{j \in \mathcal{J}} a_i^j \, x_j \geq 1, \qquad i \in [p] \\ & x_j \in \{0,1\} \qquad\quad j \in \mathcal{J} \, . \end{aligned} \tag{1}$$

For the max redundancy problem we should maximize $R(x) = \sum_j \sum_i a_i^j \, x_j$ and add to the constraints in (1) the constraints $\sum_j x_j = V$ to impose the minimum cardinality requirement. However, it is more convenient to tackle the problem directly by defining for the box B_j a cost

$$c_j = K - \sum_{i \in [p]} a_i^j = K - |B_j| \tag{2}$$

and minimizing $\sum_{j \in \mathcal{J}} c_j \, x_j$ over the same constraints of (1). In (2) K is a suitably large constant so that the solution is guaranteed to be one of minimum cardinality. To this purpose we prove the following theorem.

Theorem 4. *If $K \geq p^2$, a family $J^* \subset \mathcal{J}$ of boxes minimizing*

$$\sum_{j \in J} c_j = K \, |J| - \sum_{j \in J} |B_j|$$

for all feasible $J \subset \mathcal{J}$ has minimum cardinality and maximum redundancy.

Proof: Let J^* be a family minimizing $\sum_{j \in J} c_j$. Then for any other feasible family J we have

$$K \, |J^*| - \sum_{j \in J^*} |B_j| \leq K \, |J| - \sum_{j \in J} |B_j|. \tag{3}$$

We may assume $|J^*| > 1$ since $|J^*| = 1$ already implies minimum cardinality of J^*. Note that $|B_j| \leq p$, for any box. Furthermore, only one box of the family J^* can have $|B_j| = p$ otherwise the family would consist

of the same essential box covering all points replicated many times and this family could not be optimal for the cost. Hence we derive from (3)

$$K\,|J^*| - p\,|J^*| < K\,|J| - \sum_{j\in J} |B_j|. \tag{4}$$

Among the feasible families of boxes there is a trivial one consisting of p boxes covering single points. Its cost is $K\,p - p$. Hence we have

$$K\,|J^*| - p\,|J^*| < K\,p - p,$$

i.e.,

$$|J^*| < p\,\frac{K-1}{K-p}.$$

For any $K \geq p^2$ we derive the bound

$$|J^*| < p\,\frac{p^2-1}{p^2-p} = p + 1. \tag{5}$$

Note that $\sum_{j\in J} |B_j| \geq p$ for any feasible J since each point must be covered. Then we derive from (4) by using also the bound (5)

$$K\,|J^*| - p\,(p+1) < K\,|J| - p,$$

i.e.,

$$|J^*| < |J| + \frac{p^2}{K},$$

which, for $K \geq p^2$ implies $|J^*| \leq |J|$.

Among all subsets of minimum cardinality the term $K\,|J|$ is constant so that minimizing $\sum_{j\in J} c_j$ is equivalent to maximizing $\sum_{j\in J} |B_j|$, i.e., to maximize the redundancy. ∎

For the max weighted redundancy model the cost of the box j is $c_j = K - \sum_{i\in[p]} a_i^j\, w_i$, where K can be again chosen larger than p^2 to guarantee a solution of minimum cardinality since we have assumed $w_i \leq 1$ (we leave to the reader rephrasing the proof of Theorem 4 for this case). The 0-1 LP problem for the max redundancy problem (both unweighted and weighted) is therefore

$$\begin{aligned} \min\ & \sum_{j\in\mathcal{J}} c_j\, x_j \\ & \sum_{j\in\mathcal{J}} a_i^j\, x_j \geq 1, \qquad i \in [p] \\ & x_j \in \{0,1\} \qquad\quad j \in \mathcal{J}\,. \end{aligned} \tag{6}$$

The relaxation of (1) and (6) can be solved via column generation. The dual pricing constraints for (1) are (note that $x_j \in \{0,1\}$ can be relaxed as $x_j \geq 0$)

$$\sum_{i=1}^{p} a_j^i\, y_i \leq 1 \qquad j \in \mathcal{J} \tag{7}$$

with $y_i \geq 0$ dual variables, and for (6)

$$\sum_{i=1}^{p} a_j^i \, y_i \leq c_j \qquad j \in \mathcal{J}$$

which can be rewritten for the unweighted case as

$$\sum_{i=1}^{p} a_j^i \, (1 + y_i) \leq K \qquad j \in \mathcal{J} \tag{8}$$

and for the weighted case as

$$\sum_{i=1}^{p} a_j^i \, (w_i + y_i) \leq K \qquad j \in \mathcal{J} \tag{9}$$

In all three cases finding a violated dual constraint can be carried out by solving a Maximum Box problem with weights W_i, respectively, $W_i := y_i$, $W_i := (1+y_i)$ and $W_i := (w_i+y_i)$. The Maximum Box problem is NP-hard for general n. It can be solved either via Integer Linear Programming or via a specialized combinatorial algorithm as shown in [6]. The combinatorial algorithm exhibits reasonable computing times and therefore can be used in the pricing problem.

4 The pricing problem

Both the ILP and the combinatorial formulations are described in detail in [6]. Clearly both approaches can be used for the pricing problem. Since the pricing problem has to be solved within a branch-and-bound method, thus with possibly additional constraints, we need to shortly recall here the approaches proposed in [6] (with a different notation). As already remarked, the k-th coordinate of a positive box can be assumed to take values in the set

$$V^k := \bigcup_{i=1}^{p} \left\{ X_k^i \right\}, \qquad k \in [n] \, .$$

So in order to define a box we need assigning ℓ_k to exactly one value in V^k and similarly for u_k. For the ILP model let

$$\zeta_{kv} = \begin{cases} 1 & \text{if } \ell_k = v \in V^k \\ 0 & \text{otherwise} \, , \end{cases} \qquad \xi_{kv} = \begin{cases} 1 & \text{if } u_k = v \in V^k \\ 0 & \text{otherwise} \, . \end{cases}$$

The assignment is realized through the constraints

$$\sum_{v \in V^k} \zeta_{kv} = 1, \qquad \sum_{v \in V^k} \xi_{kv} = 1, \qquad k \in [n] \, . \tag{10}$$

We need counting the points covered by the box. To this aim let

$$\eta_i = \begin{cases} 1 & \text{if } X^i \text{ is covered} \\ 0 & \text{otherwise} \end{cases}$$

subject to the constraints

$$\eta_i \le \sum_{v \le X_k^i} \zeta_{kv}, \qquad \eta_i \le \sum_{v \ge X_k^i} \xi_{kv}, \qquad k \in [n], \quad i \in [p]\,. \tag{11}$$

For a box not to cover negative points the following constraints must be introduced

$$\sum_{k=1}^{n} \big(\sum_{v > Y_k^i} \zeta_{kv} + \sum_{v < Y_k^i} \xi_{kv} \big) \ge 1, \qquad i \in [q]\,. \tag{12}$$

Then the pricing problem consists in maximizing

$$\sum_{i \in [p]} W_i\, \eta_i$$

subject to (10), (11) and (12).

The combinatorial algorithm is based on a branching scheme where a non-homogeneous box is split into $2\,n$ smaller boxes 'around' a negative point contained in the box. We present here a slightly simpler version than the one in [6]. For each coordinate the box is split into two boxes, one above the corresponding coordinate of the negative point and the other one below, all other coordinates remaining the same. In more detail, let $B(\ell, u)$ be the non-homogeneous box and let Y be the negative point in $B(\ell, u)$, i.e., $\ell_k \le Y_k \le u_k$, $k \in [n]$, and let

$$I := \big\{ i : X^i \in B(\ell, u) \big\}$$

the index set of the points in the box $B(\ell, u)$. For each coordinate k we compute

$$I_k^- := \big\{ i \in I : X_k^i < Y_k \big\}\,, \qquad I_k^+ := \big\{ i \in I : X_k^i > Y_k \big\}\,, \qquad k \in [n]$$

and the $2\,n$ box-closures

$$\Big[\bigcup_{i \in I_k^-} \{X^i\} \Big], \qquad \Big[\bigcup_{i \in I_k^+} \{X^i\} \Big], \qquad k \in [n]\,.$$

It can be proved that any positive box contained in the box is also contained in at least one of these $2\,n$ boxes. This branching scheme is accompanied by an effective upper bound function which is also used to select the best (in term of bound) negative point in the box. According to [6] this combinatorial algorithm is much faster than the ILP formulation. We support this statement after extensive computational tests.

In order to speed-up the pricing phase, especially in the first column generations, we have also devised a heuristic algorithm for the Maximum Box problem. The pricing is first carried out by the heuristics and the exact algorithm is run only if a violated dual inequality is not found by the heuristics. The heuristics is a greedy one and is based on the following idea: preliminarily for all pairs of positive points X^i, X^j, the L_1 distances

$$d_{ij} := \sum_{h=1}^{n} |X_h^i - X_h^j|$$

are computed. We remark the the computation of the distances d_{ij} is carried out only once. Then, for each point X^i such that its weight W_i is strictly above a fixed threshold (that we have chosen equal to 1 for the max redundancy problem, i.e., all points with dual variable $y_i = 0$ are excluded) the other points X^j are sorted in order of decreasing values W_j/d_{ij}. A box is built starting from each point X^i by adding one point at a time according to this order and discarding points whose inclusion would yield a non homogeneous box. The order of the starting points is given by decreasing weights W_i. As soon as a box violating a dual inequality is found the procedure is stopped. A similar procedure is done for the negative points.

5 Pricing and branching

Both exact pricing methods can be used as such only at the root node of the branch-and-bound tree. At other nodes we must take care also of branching constraints. If branching is carried out by imposing either $x_j = 0$ or $x_j = 1$ in case of a fractional variable x_j corresponding to the box B_j, these constraints must be embedded in the pricing problem.

Imposing $x_j = 0$ means that in the pricing problem we don't want the box B_j as a possible solution. Since only maximal boxes can be considered for an optimal solution, this means that we don't want also any box contained in B_j.

To this aim it is enough in the ILP model that at least one of the assignment variables ζ_{kv_ℓ} and ξ_{kv_u} identifying B_j and the boxes in B_j must be zero, which can be accomplished by

$$\sum_{k=1}^{n} \Big(\sum_{\ell_k^j \le v_\ell \le u_k^j} \zeta_{kv_\ell} + \sum_{\ell_k^j \le v_u \le u_k^j} \xi_{kv_u} \Big) \le 2\,n - 1\,. \tag{13}$$

Therefore in the ILP model we need only adding constraints of the type (13) for each variable fixed to zero.

In the combinatorial model the box B_j must be considered infeasible, like any other non-homogeneous box. As observed, also the boxes contained in B_j have to be considered infeasible. Hence the only action to take, as soon as the box B_j (or some of its included boxes) appears at a node of the search tree of the combinatorial algorithm is to directly fathom that node. We just need to add to the algorithm some bookkeeping scheme to easily identify 'forbidden' boxes.

This method to avoid generating forbidden boxes can be generalized to avoid the generation of boxes with undesirable features, e.g., boxes covering less than a fixed amount of points.

Imposing $x_j = 1$ is equivalent to cover all positive points in B_j. Therefore we might think of reformulating the problem by simply dropping these points. However, we have to be careful. Some of the dropped points could be fruitfully covered also by other boxes to increase the redundancy. Hence all points must be retained in the computation. Let us consider the following three problems, labeled as (14)-(a), (b) and (c), where, for

the sake of notation, the variable to be fixed to one is x_1.

$$\begin{array}{lll} \text{(a)} & \text{(b)} & \text{(c)} \\ \min \sum_j c_j\, x_j & \min \sum_j c_j\, x_j & \min \sum_j c_j\, x_j \\ \sum_j \bar{a}_i^j\, x_j \geq 1 & \sum_j a_i^j\, x_j \geq b_i & \sum_j a_i^j\, x_j \geq 1 \\ & & x_1 \geq 1 \\ x_j \geq 0 & x_j \geq 0 & x_j \geq 0 \end{array} \tag{14}$$

In problem (14)-(a) the matrix $\bar{a}_i^j$ is the matrix a_i^j without the rows such that $a_i^1 = 1$ (hence the first column $\bar{a}^1$ has all zeros). In other words, all points covered by box 1 are dropped from the problem. Let x^1 and y^1 be primal and dual variables of problem (14)-(a). In problem (14)-(b) $b_i = 1 - a_i^1$. In other words, points already covered by box 1 need not to be covered in problem (14)-(b), but they are present. Let x^2 and y^2 be primal and dual variables of problem (14)-(b). In problem (14)-(c) there is an explicit constraint imposing $x_1 = 1$. Let x^3 and (y^3, t) be primal and dual variables of problem (14)-(c), where t refers to the constraint $x_1 \geq 1$.

Proposition 1. *There are dual optima such that $y_i^1 = y_i^2 = y_i^3$ for each row i in $\bar{a}_i^j$ and $y_i^2 = y_i^3 = 0$ otherwise (note that $y_i^3 = b_i\, y_i^2$) and $t = c_1$.*

Proof: It is immediate that there are optima such that $x_j^1 = x_j^2 = x_j^3$ for $j > 1$, $x_1^1 = x_1^2 = 0$ (assuming $c_1 > 0$) and $x_1^3 = 1$. Let y^1 be dual optimum for problem (14)-(a). Define $y_i^2 = y_i^3 = y_i^1$ for each row i in $\bar{a}_i^j$ and $y_i^2 = y_i^3 = 0$ otherwise. Then y^2 is dual feasible for problem (14)-(b). Similarly, (y^3, t) is dual feasible for problem (14)-(c) for all $j > 1$ and for $j = 1$ we have $\sum_i a_i^1\, y_i^3 + t = 0 + c_1$, i.e., (y^3, t) is feasible. We also have, by exploiting strong duality for x^1 and y^1,

$$\sum_j c_j\, x_j^2 = \sum_j c_j\, x_j^1 = \sum_i y_i^1 = \sum_i b_i\, y_i^2$$

which establishes strong duality for x^2 and y^2, so that y^2 must be optimal. Moreover, by exploiting strong duality for x^2 and y^2,

$$\sum_j c_j\, x_j^3 = \sum_j c_j\, x_j^2 + c_1 = \sum_i b_i\, y_i^2 + c_1 = \sum_i y_i^3 + t$$

which establishes strong duality between x^3 and y^3 and optimality of y^3. ∎

In conclusion, the dual optima for pricing can be computed by problem (14)-(a), which has the advantage to be smaller than the others. However, since the cost c_j are computed by taking into account all points, the pricing must be done on the totality of positive points.

The type of branching we have considered is the usual branching on fractional variables. On our computational experiments we have observed a different behavior of the computation on the two branches. While setting a variable to 1, i.e., forcing a box to be chosen, is quite effective in terms of increasing the lower

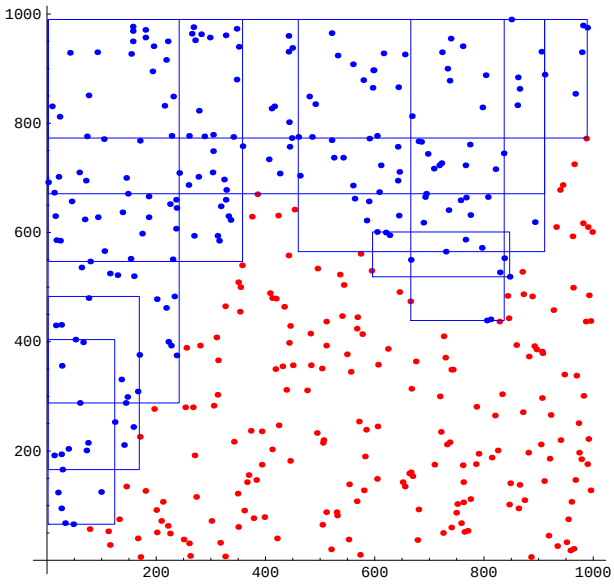

Fig. 1. A cardinality optimal family of boxes

bound, on the contrary, setting a variable to 0, i.e., forbidding a particular box, is not as effective. What happens is that the computation tries to 'fix' the prohibition to use that particular box by finding a slightly different one. Since there may be many almost equal boxes, especially with high dimensional boxes, the lower bound increases very slowly on the paths of the B&B tree with variables fixed to 0. Finding a better branching is matter of future investigation.

6 Computational results

We have carried out two types of tests. The first one is based on random artificial instances with points generated to form clusters of some preassigned form. The second one is based on real data. A first set contains data from healthy and sick individuals who have been medically tested at the Rome University Hospital for carpal tunnel syndrome [11] and a second set is the Breast Cancer Wisconsin data set available at the UCI Machine Learning Repository [14].

The first artificial test is two dimensional. There are 200 positive points and 200 negative points in a square. They have been generated randomly so as to be separated by a cubic function. The points can be seen in Figure 1, where blue and red points are positive and negative respectively. We have carried out four different computations, according to the pricing method (either ILP or the combinatorial algorithm) and to the initial solution (either one very large cost non-homogeneous giant box covering all positive points or 200 homogeneous degenerate boxes each one covering a single positive point).

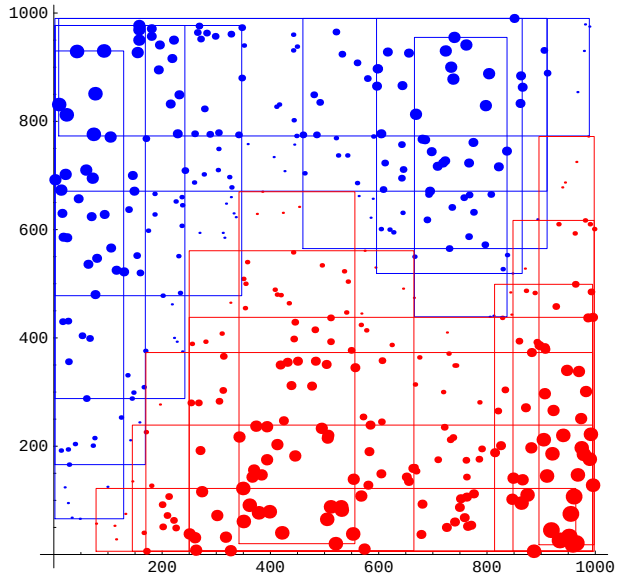

Fig. 2. Redundancy optimal families of boxes

The set covering problem with ILP pricing method has been tested with the commercial LP solver Lingo, whereas the problem with the combinatorial pricing method has been tested with *Mathematica 5.1*. Clearly CPU time comparisons cannot be done because *Mathematica* is designed to be very flexible and not very fast. In addition all computations have been carried out in exact arithmetic. Yet, the two computations with Mathematica required much less time than the two with Lingo. The sequences of columns priced out by the two procedures are different although starting from the same data. This has to be ascribed to the fact that optimal solution of the pricing problem are in general non-unique and thus the pricing can generate different columns. It turned out that the pricing with the combinatorial algorithm has required much less column generations than the ILP method. Due to the very poor performance of the ILP pricing all our tests have been carried out with the combinatorial pricing computed with Mathematica.

In general we have observed a better performance if the initial solution consists of boxes covering single points. Almost half column generations are needed with respect to starting with the giant non-homogeneous box.

To solve the cardinality problem for the positive points of the two dimensional example of Figure 1, 13 column generations were needed with a minimum of 7, a maximum of 40 and an average of 30.3 B&B nodes of the combinatorial pricing algorithm. The cardinality optimal solution requiring 9 boxes (shown in the figure) was integral at the root node of the B&B tree. The time to find this solution was 13 minutes. We stress that the algorithm has run within the Mathematica software. We may consider a speed-up of two orders of magnitude with a compiled code calling a fast LP solver.

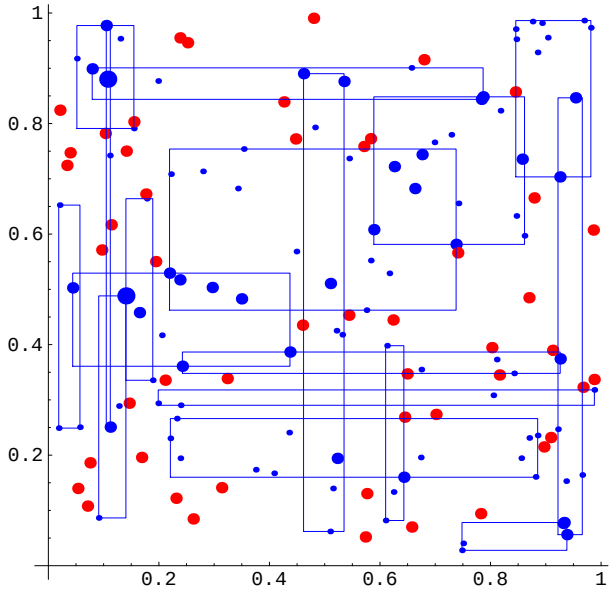

Fig. 3. Optimal family of boxes for random data

For the max redundancy problem we have chosen $K = p$ instead of the theoretical value $K = p^2$. We have observed that by choosing $K = p$, so that the cost of a box is the number of points not covered by the box, the solution was always one of minimum cardinality. Solving the max redundancy problem for the positive points of the same instance has required 16 column generations, with a minimum of 6, a maximum of 56 and an average of 32.5 B&B nodes of the combinatorial pricing algorithm. The solution consists of 9 boxes and has redundancy 584. The max redundancy for the negative points has required 15 column generations with a minimum of 14, a maximum of 38 and an average of 26.6 B&B nodes. Also this solution consists of 9 boxes and its redundancy is 719.

In Figure 2 both positive and negative families are depicted. The point size has been chosen roughly proportional to the redundancy of the point. By comparing the max redundancy positive covering with the min cardinality in Figure 1 we may appreciate how the max redundancy provides a better covering of the points. We may note that there are regions, in fact boxes, where positive and negative boxes overlap. Of course no point can belong to these regions. However, new points to be analyzed could fall within these regions and their classification would require additional tests. Also there are regions not covered by any box. Also in this case further testing would be needed.

As a second artificial test we have generated 100 positive random points and 50 negative random points in a square (see Figure 3). Due to the mixing of positive and negative points we expect this test to be severe for the algorithm. We have run the algorithm with objective function the maximum redundancy for the positive points ($K = p = 100$). At the root node of the B&B tree the solution was fractional with value 1,487.25. This solution has fractional cardinality 16.125 and fractional redundancy 125.25. During the column generation

process an integral solution of value 1566 with cardinality 17 and redundancy 134 has been produced. This solution is cardinality optimal as it has been possible to check by separately running the cardinality objective function. The combinatorial pricing algorithm at the root node has been called 49 times with minimum and maximum number of B&B nodes 24 and 265 respectively, and an average of 126.6 nodes.

Then the B&B search for the optimal redundancy family has been stopped after having generated 200 nodes in the B&B tree. In one of the earliest subproblems (level 3 by imposing three boxes) a better solution with value 1,549, 17 boxes and redundancy 151 was found. This is the solution shown in Figure 3. The blue points are reproduced with size proportional to their redundancy.

Finally we have generated a higher dimensional test with $n = 4$ and points almost linearly separable. A random point X was generated uniformly within the unit box. Then it was accepted as a positive point with probability $e^{-\alpha\, t}$ with $t := \max\{0, \sum_i X_i - n/2\}$ and α a parameter tuned to the value 10^4. Similarly a random point Y was generated uniformly within the unit box and accepted as a negative point with probability $e^{-\alpha\, t}$ with $t := \max\{0, -\sum_i X_i + n/2\}$. The instance has 103 positive points and 118 negative points and we have solved the max redundancy problem for the positive points ($K = p = 103$). After 37 column generations a fractional solution of value 468.33 has been found. Since during the process a solution with value 469 was found, this solution is optimal. This solution has 7 boxes with redundancy 252. The number of B&B nodes of the pricing problems ranged from a minimum of 43 nodes to a maximum of 531 with an average of 242.5. On Mathematica the overall computation time has been quite high, about one hour. However, we recall again that this value is not significant due the characteristics of Mathematica.

As already remarked we have observed a very modest increasing of the lower bound in the B&B tree when a variable is set to the value 0, for the reasons already explained. It is indeed this slow increasing that prevented a faster computation.

The second type of data are based on real observation. The first test is based on ultrasonic and electromyography measurements taken on 102 individuals, 64 of them healthy and 38 sick of carpal tunnel syndrome. There are in total 7 different measurements for each individual. The detailed explanation of these data can be found in [11]. The 38 data referring to the sick persons are the positive data and the 64 data referring to the healthy persons are the negative data.

It has to be observed that this set of data turns out to be an easy one for the box covering problem. Indeed one box is enough to cover all negative points, and two boxes only are needed to cover the positive points. These two boxes are largely overlapping so that the redundancy is 68 (i.e. 30 points out of 38 are covered by both boxes). The computation is immediate for both positive and negative covers and therefore this test is obviously successful from the point of view of the computational performance.

From this data set we have randomly selected half positive and half negative data to be used as a training set. This data has been covered by positive and negative boxes, respectively, and the remaining data has been used as test data. For each positive (negative) test point four outcomes are possible: it belongs to at least one positive (negative) box and to no negative (positive) box, thus being correctly identified as positive (negative); it belongs to no box, neither positive nor negative, and so its status cannot be detected; it belongs to both a positive and a negative box and so its status is again undetected; it belongs to at least one negative (positive) box and to no positive (negative) box, thus being wrongly identified as negative (positive).

In the first case we have the true positive (true negative) points, labeled TP (TN) in Table 1. In the second case we have a lack of information from the training set, likely due to too few points in the training set. This situation is labeled as "?" in the table. In the third case the training set has conveyed some sort of contradictory information. We recall that positive and negative boxes may overlap but clearly in their intersection no point of the training set must be present. New points may fall in the intersection. This situation is labeled as "!" in the table. This kind of uncertainty should be considered more critical than the previous one. Finally in the fourth case we face the undesirable situation of misclassified points. Positive points are wrongly detected as negative, so called false negative (labeled FN) and negative points are wrongly detected as positive, so called false positive (FP).

Although it is possible to further refine the classification by taking into consideration other parameters like the proximity of a point to other boxes (see for instance [12, 13]), we have not pursued this line here because we consider these aspects not central to this paper.

We have carried out 25 random selections of the training sets. Each row in Table 1, except the last two, reports the results of each random selection. In particular the numbers refer to the number of points which fall into one of the situations described above. The last two rows report the average values of the 25 tests and their percentages respectively. It is noteworthy to mention that only one false positive and no false negative case has been observed by box clustering.

The second test of real data comes from the UCI benchmark instances. We have chosen the Breast Cancer Wisconsin (Diagnostic) data set [14]. This data comprises 699 instances with 9 attributes each. Each attribute is measured as an integer number between 1 and 10. Since there are missing values we have left out all instances with some missing value, thus obtaining 682 points, of which 238 are positive (malignant) and 444 negative (benign). Due to the higher number of points and the higher dimensionality this is computationally a more severe test than the previous data set.

We have first found a max redundancy cover both for the 444 negative points and for the 238 positive points. For the negative points the number of generated boxes by the pricing procedures has been 36, of which 14 have been generated by the heuristics (almost all at the beginning) and 26 by the exact procedure. The 26 runs of the exact combinatorial pricing have required the following number of B&B nodes

204, 185, 204, 194, 261, 491, 493, 315, 302, 384, 433,
501, 565, 573, 574, 662, 655, 424, 701, 729, 734, 757,

with an average of about 400 B&B nodes for each box generation.

The best solution found at the root node of the B&B tree consisted of the following five boxes (identified by the ℓ and u values) for which the redundancy is 1124 (on the average roughly 2.5 boxes cover a single

point and a box covers about 225 points, i.e., almost half the test size)

$$\begin{array}{l}
\left(\{1,1,1,1,2,1,1,1,1\},\ \{5,4,4,10,5,10,7,2,8\}\right)\\
\left(\{1,1,1,1,1,1,1,1,1\},\ \{8,4,6,6,5,3,7,3,2\}\right)\\
\left(\{3,3,2,2,3,1,1,2,1\},\ \{8,4,4,6,7,10,7,8,3\}\right)\\
\left(\{1,1,1,1,1,1,1,1,1\},\ \{4,4,5,6,10,4,7,8,8\}\right)\\
\left(\{4,1,4,1,1,1,1,1,1\},\ \{6,9,8,5,8,8,7,7,1\}\right)
\end{array}$$

For the positive points the number of generated boxes by the pricing procedures has been 54, of which 38 have been generated by the heuristics and 16 by the exact procedure. The 16 runs of the exact combinatorial pricing have required the following number of B&B nodes

$$\begin{array}{c}
256,\ 390,\ 123,\ 501,\ 632,\ 877,\ 582,\ 799,\\
632,\ 863,\ 669,\ 619,\ 925,\ 565,\ 1440,\ 1409
\end{array}$$

with an average of about 705 B&B nodes for each box generation.

The best solution found at the root node of the B&B tree consisted of the following eight boxes for which the redundancy is 633 (on the average roughly 2.6 boxes cover a single point and a box covers about 79 points, i.e., almost one third of the test size, note the difference with respect to the negative points)

$$\begin{array}{l}
\left(\{1,1,3,1,3,1,2,9,1\},\ \{10,10,10,10,10,10,10,10,10\}\right)\\
\left(\{1,5,2,2,2,1,2,3,1\},\ \{10,10,10,10,10,10,10,10,10\}\right)\\
\left(\{5,1,1,1,2,3,2,1,1\},\ \{10,6,8,4,8,10,7,6,8\}\right)\\
\left(\{9,1,1,1,2,1,1,1,1\},\ \{10,10,10,10,10,10,10,10,10\}\right)\\
\left(\{1,4,2,6,2,1,2,1,1\},\ \{10,10,10,10,10,10,10,10,10\}\right)\\
\left(\{1,5,2,1,2,1,5,1,1\},\ \{10,10,10,10,10,10,10,10,10\}\right)\\
\left(\{3,2,2,1,5,1,1,1,1\},\ \{10,10,6,10,6,10,6,10,4\}\right)\\
\left(\{2,1,1,2,1,4,2,1,1\},\ \{10,8,10,9,5,10,10,6,2\}\right)
\end{array}$$

Then, as for the carpal tunnel data case, we have randomly selected a training set to test the other data. In this case we have selected 100 positive and 100 negative instances out of the whole data set. We have found a max redundancy cover (both negative and positive) of these points and used these boxes to test the remaining 138 positive and 344 negative points. We have run this test 25 times. The results are reported in Table 2 with the last two rows reporting the average and the percentage values respectively.

We have noted that in all cases only two boxes were necessary to cover the negative points to confirm the observation that healthy individuals tend to have the attribute measures within an interval and these intervals are little correlated. On the contrary four/five boxes were needed to cover the positive data.

7 Conclusions

In this paper we have considered a box-clustering classification method for positive and negative data. In particular we have developed a column-generation integer linear programming model to find both a positive and a negative box family covering of the data. The box family can be chosen to be one of minimum cardinality and, among the many minimum cardinality families, we may prefer the ones that maximize coverage. To this purpose we have introduced the concept of redundancy. The problems we want to solve are NP-hard and the method we propose for their solution is exact, although clearly suffers from the fact that the pricing itself is NP-hard. Further investigation is needed to develop a reliable heuristics for pricing in order to speed-up the pricing process and to devise a better branching scheme in presence of fractional solutions.

The method is flexible enough to introduce particular constraints, like for instance allowing only sufficiently large boxes (this can be achieved in the pricing procedure) and/or requiring at least a specified number of boxes (this can be achieved by simply adding one row constraint to (6) with very little modifications).

Acknowledgments

I am indebted to Federica Ricca and Vincenzo Spinelli for providing the carpal tunnel syndrome data.

References

1. K.P. Bennett and O.L. Mangasarian: Robust linear programming discrimination of two linearly inseparable sets. *Optimization methods and software*, **1**, 23–34, (1992).
2. S. Bereg, S. Cabello, J. M. Díaz-Báñez, P. Pérez-Lantero, C. Seara, and I. Ventura: The class cover problem with boxes. *Computational Geometry*, **45**, 294–304, (2012).
3. E. Boros, P.L. Hammer, T. Ibaraki and A. Kogan: Logical analysis of numerical data. *Mathematical Programming*, **79**, 163–190, (1997).
4. E. Boros, V. Spinelli and F. Ricca: Incompatibility graphs and data mining, in: *Proceedings of the 10th Cologne-Twente Workshop on Graphs and combinatorial optimization*, Frascati, June 14-16, 2011, `http://ctw2011.dia.uniroma3.it/ctw_proceedings.pdf` (2011).
5. Chunhui Chen and O.L. Mangasarian: Hybrid misclassification minimization. *Advances in Computational Mathematics*, **5**, 127–136, (1996).
6. J. Eckstein, P.L. Hammer, Y. Liu, M. Nediak, and B. Simeone: The maximum box problem and its application to data analysis. *Computational Optimization and Applications*, **23**, 285–298, (2002).
7. G. Felici, B. Simeone and V. Spinelli: Classification Techniques and Error Control in Logic Mining, in: *Data Mining:* special issue in *Annals of Information Systems,* **8***,* 99–119, R. Stahlbock, S.F. Crone and S. Lessmann (eds), (2010).
8. B.J. Gao, M. Ester, H. Xiong, J. Cai and O. Schulte: The Minimum Consistent Subset Cover Problem: A Minimization View of Data Mining. *IEEE Transactions on Knowledge and Data Engineering*, **99**, , (2011).

9. P.L. Hammer, Y. Liu, S. Szedmák and B. Simeone: Saturated systems of homogeneous boxes and the logical analysys of numerical data. *Discrete and Applied Mathematics*, **144**, 103–109, (2004).
10. O.L. Mangasarian: Linear and nonlinear separation of patterns by linear programming. *Operations Research*, **13**, 444–451, (1965).
11. M. Maravalle, F. Ricca, B. Simeone and V. Spinelli: Carpal Tunnel Syndrome Automatic Classification: Electromyography vs. Ultrasound imaging. Dipartimento di Scienze Statistiche, Sapienza, Università di Roma, Serie A - Ricerche, n. 11/2011. (2011).
12. S. Salzberg: A nearest hyperrectangle learning method. *Machine Learning*, **6**, 251–276, (1991).
13. D. Wettschereck and T.G. Dietterich: An experimental comparison of the nearestneighbor and nearest-hyperrectangle algorithms. *Machine Learning*, **19**, 5–27, (1995).
14. W.H. Wolberg: Wisconsin Diagnostic Breast Cancer (WDBC), visited on: June 5, 2012, `http://archive.ics.uci.edu/ml/datasets/Breast+Can cer+Wisconsin+%28Original%29` (1991).

positive test points				negative test points			
TP	?	!	FN	TN	?	!	FP
10	9	0	0	23	9	0	0
4	15	0	0	24	8	0	0
8	11	0	0	28	4	0	0
13	6	0	0	24	8	0	0
10	9	0	0	25	7	0	0
9	10	0	0	23	9	0	0
10	9	0	0	21	11	0	0
10	9	0	0	23	9	0	0
6	13	0	0	22	10	0	0
13	6	0	0	26	6	0	0
10	9	0	0	22	10	0	0
12	7	0	0	23	9	0	0
9	10	0	0	20	11	0	1
8	11	0	0	29	3	0	0
13	6	0	0	31	1	0	0
8	11	0	0	22	10	0	0
13	6	0	0	20	11	1	0
6	13	0	0	29	3	0	0
11	8	0	0	23	9	0	0
6	13	0	0	19	13	0	0
8	11	0	0	28	4	0	0
11	8	0	0	22	10	0	0
9	10	0	0	24	8	0	0
12	7	0	0	26	5	1	0
12	7	0	0	21	11	0	0
9.64	9.36	0.00	0.00	23.92	7.96	0.08	0.04
50.73	49.27	0.00	0.00	74.75	24.87	0.25	0.13

Table 1. Carpal tunnel syndrome data. TP: true positive; TN: true negative - positive and negative points detected as such; ?: undecided - not belonging to any box; !: overdecided - belonging to both positive and negative boxes; FP: false positive; FN: false negative

positive test points				negative test points			
TP	?	!	FN	TN	?	!	FP
127	8	3	0	317	10	6	11
109	19	5	5	328	11	3	2
127	8	1	2	319	11	3	11
132	3	2	1	320	7	4	13
129	6	2	1	309	15	8	12
126	9	2	1	318	13	3	10
128	6	3	1	306	19	5	14
117	15	1	5	317	19	3	5
116	17	5	0	312	6	16	10
126	7	3	2	330	5	3	6
120	10	4	4	325	7	7	5
125	12	1	0	318	10	5	11
128	7	0	3	321	12	1	10
126	4	2	6	323	7	3	11
125	12	1	0	318	10	5	11
130	7	0	1	313	11	9	11
124	9	4	1	309	14	4	17
133	4	1	0	315	8	9	12
133	4	0	1	309	7	11	17
119	15	2	2	318	17	1	8
117	18	1	2	324	7	5	8
115	12	8	3	327	8	2	7
134	4	0	0	314	7	9	14
129	6	2	1	306	19	9	10
125	7	3	3	321	13	2	8
124.80	9.16	2.24	1.80	317.48	10.92	5.44	10.16
90.44	6.64	1.62	1.30	92.29	3.17	1.58	2.96

Table 2. Wisconsin diagnostic breast cancer data. TP: true positive; TN: true negative - positive and negative points detected as such; ?: undecided - not belonging to any box; !: overdecided - belonging to both positive and negative boxes; FP: false positive; FN: false negative