Building separating concentric balls to solve a multi-instance classification problem *

Emilio Carrizosa, José Gordillo[†]
Universidad de Sevilla (Spain)
{ecarrizosa,jgordillo}@us.es

Frank Plastria
Vrije Universiteit Brussel (Belgium)
Frank.Plastria@vub.ac.be

3rd February 2008

Abstract

In this work, we consider a classification problem where the objects to be classified are bags of instances which are vectors measuring d different attributes. The classification rule is defined in terms of a ball, whose center and radius are the parameters to be computed. Given a bag, it is assigned to the positive class if at least one element is strictly included inside the ball, and it is labelled as negative otherwise. We model this question as a margin optimization problem. Several necessary optimality conditions are derived leading to a polynomial algorithm in fixed dimension. A VNS type heuristic is proposed and experimentally tested.

Keywords: Supervised Classification, Multi-Instance Learning, Mixed-Integer Programming, Variable Neighborhood Search.

[*] This work has been partially supported by the grants MTM2005-09362-103-01 of MEC, Spain, and FQM-329 of Junta de Andalucía, Spain. Their financial support is gratefully acknowledged.

[†] Corresponding author.

1 Introduction

Consider a classification problem where each of the objects which compose the dataset may be represented by several feature vectors, and only some of them (even only one in some cases) are responsible for the class that the object belongs to. For approaching this task, a set of tools grouped under the name of Multiple Instance Learning have been developed during the last decade.

A Multiple Instance Problem is a supervised classification problem where the objects to be classified are bags of instances which are vectors measuring d different attributes (see e.g. [11, 39, 42] for a description). According to these measurements, a label ($+1$ or -1 in the two-class case) must be assigned to each bag.

Although there are several ways to assign a label to the complete set of vectors (see [39]), in our problem, the class label of a bag will be chosen according to the so-called MI assumption (see [39]), which states that a bag is considered as positive (label $+1$) if at least one of its instances satisfies a determined condition and as negative (label -1) otherwise.

1.1 Origin and contributions to the problem

Multi-Instance Learning was first applied to drug activity prediction by Dietterich et al. in ([11]). In that problem, the objects are molecules, whereas the instances are different low-energy conformations, that is, shapes that the molecule can adopt by rotating its bonds. Only some of these conformations can bind to a target binding site (which is a part of a larger molecule) and then, the molecule becomes active, producing a determined drug. But the only information biochemists know is if a molecule is qualified or not to make a drug, not having any information about the correct conformations. The aim is thus, given a new molecule, to be able to predict if it will be active or inactive by taking into account the whole set of its conformations.

In [11], an axis-parallel hyper-rectangle is built such that it must contain at least one instance of each positive bag and it cannot contain any instance of the negative bags. Three algorithms are derived for this. The first algorithm constructs the smallest rectangle covering all the instances of the positive bags, and then it excludes all the negative instances, by eliminating, in turns, the negative instance which requires eliminating the smallest number of positive instances. The second algorithm is a modification of the first one, which assigns a cost for eliminating each positive instance (this is useful, for example, to avoid excluding from the rectangle a positive instance which is the last survivor from its bag). The third algorithm builds the smallest rectangle covering at least one instance of each positive bag, by using a backfitting procedure to choose the positive instances and by selecting the most relevant features in each step.

Since the appearance of this paper, several authors have developed new algorithms to try to improve the results obtained by Dietterich et al., following different strategies.

- **Diverse Density:** In [25], the concept of *Diverse Density* is introduced to deal with multi-instance problems. A point with high Diverse Density is near to many instances of different positive bags, and far from the instances of the negative bags. The problem is thus to find the point with the maximum Diverse Density and, when having a new bag, it is classified as positive if the smallest distance from the bag to that point is smaller than a certain threshold, and as negative otherwise. In [41], an Expectation-Maximization algorithm is proposed to maximize the Diverse Density, by transforming the multi-instance problem into a single instance and by using the EM algorithm to maximize the responsability for the label of each bag. This same strategy is also used in [32].

- ***k*-Nearest Neighbour:** Two variants of the k-NN algorithm are proposed in [37] to solve multi-instance problems. The minimal and maximal Hausdorff distances are defined to measure the proximity between bags. A Bayesian framework is used for the so-called *Bayesian k-NN* algorithm. For the *Citation k-NN* algorithm, the concepts of references and citers are applied to the k-NN methodology, by studying for each bag not only its neighbours, but also the bags for which the concerned bag is a neighbour. These concepts are also used in [43].

- **Support Vector Machines:** In [1], two formulations as a mixed integer quadratic program are proposed. The *mi-SVM* approach introduces integer variables for modeling the individual labels of the instances of the positive bags, while the *MI-SVM* formulation selects one representative from each positive bag as that one which determines the sign of the label of its bag. Both algorithms compute optimal hyperplanes, given an initial assignment for the integer variables, and they update those integer values by assigning a positive label to the instance with the highest value for the classification rule.

  A separating hyperplane via SVMs is also proposed in [24], by using that a positive bag is correctly classified if at least one element of the convex hull of the instances of the bag is included in the positive halfspace. A set of bilinear constraints are derived and, in turns, one set of variables is held constant and the underlying linear program is solved. This successive linearization algorithm converges in a few iterations to a local optimum.

  In [10], an image categorization problem is solved by using SMVs and the k-means algorithm. Given an image, this is represented as a set of image patches, and, via the k-means clustering algorithm, the patch descriptors are assigned to a predetermined number of clusters. Every image is described via a feature vector, counting the number of its patches in each cluster. The one-versus-rest SVM algorithm is used to solve the problem of assigning each image to a cluster.

  Transductive SVMs, a modification of SVM which forces to unlabeled data (coming from positive bags) to be as far as possible from the separating hyperplane, are used in [2], obtaining good results when the positive bags are sparse (few positive instances).

- **Kernels:** A general kernel on multi-instance data is defined in [16], which separates positive and negative bags under natural assumptions. The kernel procedure described in [23] consists in mapping the database to a Hilbert space via a first kernel, fitting a Gaussian model to each bag in that feature space and defining a second kernel as the Bhattacharyya's affinity between such Gaussian models.

In [6], a similarity measurement is defined between a bag and an instance, and a mapping is defined in terms of that measure, transforming the multi-instance problem into a single-instance, solved via SVMs. A similar strategy is followed in [7], but using, to define the mapping, the maximum distance between a bag and a set of instances prototypes obtained via Diverse Density. Other kernels on sets of vectors can be found in [13].

- **Other techniques:** DC Optimization [8], Propositional Learning [14], Generalized Multi-Instance Learning [38], and so on.

1.2 Some application fields

Apart from drug activity prediction, Multi-Instance Learning has been successfully applied to other different fields. In image classification or retrieval (see [1, 6, 7, 13, 25, 26, 32, 40]), one has a series of images, and the goal is to train a classifier to detect a given object. The pictures are the bags of the multi-instance problem, which are segmented in sets of pixels (blobs), the instances of the problem. An image is classified as positive if it contains at least a determined blob, which is characteristic from the object to detect, and as negative otherwise.

In [43], a problem in web mining is described, where the aim is to provide recommendation on web index pages to an user, based on the browsing history of that web user. The index pages (which are web pages with links to other pages, containing only the titles of brief information about the content of those linked pages) are the bags of the multi-instance problem, while each linked page is an instance. Each page is represented by its d more frequent terms. A user is interested in an index page if he or she is interested in at least one of the links (satisfying thus the MI assumption).

Other different applications have been studied in handwriting recognition (see [23]), text categorization (see [1]) or disease prediction (see [39]).

1.3 Our approach

In this paper, we model the multi-instance classification problem by using two concentric balls as classifiers and by following the strategy of maximizing the margin, as done in Support Vector Machines with hyperplanes (see e.g. [9]). We obtain a formulation as a nonlinear mixed-integer program, which must be solved through heuristic techniques. Necessary conditions of optimality are derived for this problem, which are used to build a Variable Neighbourhood Search algorithm (see [27]) to obtain a solution.

This work is structured as follows. In Section 2, the problem is formulated as one of maximizing a margin, in a similar way to the strategy used in Support Vector Machines methods. In Section 3, the existence of optimal solution in our problem is studied, and in Section 4, several necessary conditions of optimality are derived, leading to a finite dominating set which can be determined in polynomial time, as seen in Section 5. In Section 6, a VNS algorithm is proposed, by using the optimality conditions, to solve the problem. Two extensions of this VNS algorithm, when we introduce p separating balls in the classification rule and when there are more than two possible labels in the database,

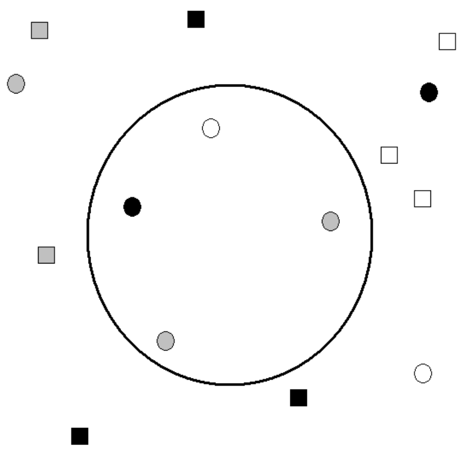

Figure 1: Construction of the classifier

are studied in 7. The algorithm is tested on several real and artificial databases and the results are shown in Section 8. Finally, Section 9 includes some discussion and conclusions.

2 Modeling the problem

Consider a database Ω formed by objects $i = (X_i, Y_i) \in \Omega$, where each X_i is a finite set of feature vectors, $X_i = \{x_1, x_2, \ldots, x_{K_i}\}$, with $x_k \in \mathbb{R}^d$, $k = 1, \ldots, K_i$, and where Y_i is the corresponding class defined by means of a label +1 or -1.

In this work, the classification rule is defined in terms of a ball with center x_0 and radius r. Thus, our problem will be to compute the parameters $x_0 \in \mathbb{R}^d$ and $r \in \mathbb{R}_+$ which define the corresponding ball, such that the classification rule, expressed according to the MI assumption, is the following:

Given a bag $X \subset \mathbb{R}^d$,

$$\begin{array}{l} \bullet \text{ classify in } G_{+1}, \text{ if } \exists x \in X \text{ such that } \|x - x_0\|^2 < r^2, \\ \bullet \text{ classify in } G_{-1}, \text{ otherwise, i.e., if } \forall x \in X, \|x - x_0\|^2 \geq r^2. \end{array} \tag{1}$$

In other words, once x_0, r are obtained, a bag X will be labelled as member of G_{-1} only if X is fully contained in the complementary set of the open ball $B(x_0, r)$ centered at x_0 with radius r (and as member of G_{+1} otherwise).

In Figure 1, an example in dimension 2 is drawn. The circles of the same colour represent the instances of each bag of G_{+1}, whereas the squares of a determined colour represent the instances of a bag of G_{-1}. Our problem is to build a ball such that it contains at least one circle of each colour and it does not contain any squares.

Different balls may exist separating the instances in G_{+1} and G_{-1}. For instance, for the example depicted in Figure 1, an alternative separating ball is given in Figure 2. In order to choose one ball, we follow the strategy successfully used in Support Vector Machines,

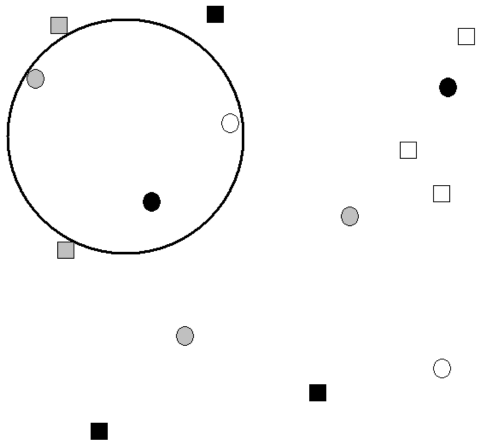

Figure 2: An alternative separating ball

[3, 9, 28], and maximize a margin, which will be defined in a similar way to that used in Support Vector Machines (see e.g. [9]).
Then, given a database, a training sample $I \subset \Omega$, extracted from it, will be used to build the optimization problem which must be solved in order to obtain optimal parameters (optimal in the sense that the margin is maximized), center and radius of $B(x_0, r)$, for constructing the classification rule.

2.1 Defining the optimization problem

According to the classification rule defined in (1), given an object i and its set of feature vectors X_i, it is assigned to the group

$$\begin{array}{lllr} G_{+1} & : \text{ if } \exists x \in X_i : \|x - x_0\|^2 < r^2 & \leftrightarrow & \min\limits_{x \in X_i} \|x - x_0\|^2 < r^2 \\ & & \leftrightarrow & \max\limits_{x \in X_i} (r^2 - \|x - x_0\|^2) > 0, \\ G_{-1} & : \text{ if } \forall x \in X_i, \|x - x_0\|^2 \geq r^2 & \leftrightarrow & \min\limits_{x \in X_i} \|x - x_0\|^2 \geq r^2 \\ & & \leftrightarrow & \min\limits_{x \in X_i} (\|x - x_0\|^2 - r^2) \geq 0. \end{array} \tag{2}$$

Given a training set $I = \{(X_1, Y_1), (X_2, Y_2), \ldots, (X_N, Y_N)\}$, we define

$$\begin{aligned} G_{+1} &= \{i \in I : Y_i = +1\}, \\ G_{-1} &= \{i \in I : Y_i = -1\}. \end{aligned}$$

The optimization problem we want to solve is

$$\max_{x_0, r} \min \left\{ \min_{i \in G_{+1}} \max_{x \in X_i} (r^2 - \|x - x_0\|^2), \min_{i \in G_{-1}} \min_{x \in X_i} (\|x - x_0\|^2 - r^2) \right\} \tag{3}$$

which can also be written as

$$\max_{(x_j)_j \in \prod\limits_{j \in G_{+1}} X_j} \max_{x_0, r} \min \left\{ \min_{j \in G_{+1}} (r^2 - \|x_j - x_0\|^2), \min_{i \in G_{-1}} \min_{x \in X_i} (\|x - x_0\|^2 - r^2) \right\}. \tag{4}$$

Denote by Δ the margin, that is, the minimum between the two distances considered in the formulation of Problem (4),

$$\Delta = \min \left\{ \min_{j \in G_{+1}} (r^2 - \|x_j - x_0\|^2), \min_{i \in G_{-1}} \min_{x \in X_i} (\|x - x_0\|^2 - r^2) \right\}. \tag{5}$$

Then, the problem remains as follows,

$$\begin{array}{lll} \max\limits_{(x_j)_j \in \prod_{j \in G_{+1}} X_j} & \max\limits_{x_0, r, \Delta} & \Delta \\ & \text{s.t.} & \Delta \leq \min\limits_{j \in G_{+1}} (r^2 - \|x_j - x_0\|^2) \\ & & \Delta \leq \min\limits_{i \in G_{-1}} \min\limits_{x \in X_i} (\|x - x_0\|^2 - r^2) \end{array} \tag{6}$$

or equivalently,

$$\begin{array}{lll} \max\limits_{(x_j)_j \in \prod_{j \in G_{+1}} X_j} & \max\limits_{x_0, r, \Delta} & \Delta \\ & \text{s.t.} & \|x_j - x_0\|^2 \leq r^2 - \Delta, \ \forall j \in G_{+1} \\ & & \|x - x_0\|^2 \geq r^2 + \Delta, \ \forall x \in \bigcup\limits_{i \in G_{-1}} X_i. \end{array} \tag{7}$$

If we denote by $r_{+1}^2 = r^2 - \Delta$, $r_{-1}^2 = r^2 + \Delta$, one has that $\Delta = \dfrac{r_{-1}^2 - r_{+1}^2}{2}$, and Problem (7) can be rewritten as

$$\begin{array}{lll} \max\limits_{(x_j)_j \in \prod_{j \in G_{+1}} X_j} & \max\limits_{x_0, r_{+1}, r_{-1}} & r_{-1}^2 - r_{+1}^2 \\ & \text{s.t.} & \|x_j - x_0\|^2 \leq r_{+1}^2, \ \forall j \in G_{+1} \\ & & \|x - x_0\|^2 \geq r_{-1}^2, \ \forall x \in \bigcup\limits_{i \in G_{-1}} X_i. \end{array} \tag{8}$$

Hence, our problem can be seen as that of obtaining two concentric balls $B(x_0, r_{+1})$, $B(x_0, r_{-1})$, where $B(x_0, r_{+1})$ contains every instance x_j (of those previously selected) from the bags of the group G_{+1}, $B(x_0, r_{-1})$ does not contain strictly any instance from the bags of the group G_{-1} and the difference between the squares of the radii is as large as possible (see Figure 3 for the example in dimension 2). Thus, we use (x_0, r_{+1}, r_{-1}) to denote a finite solution of Problem (8).

Our aim will be to characterize the existence of optimal solutions and to obtain some necessary conditions to describe an optimal solution of Problem (8), in order to obtain an optimal solution of our original problem. The reasonings used for obtaining these conditions look like others which appear in some problems in Location Theory (see e.g. [5, 33]).

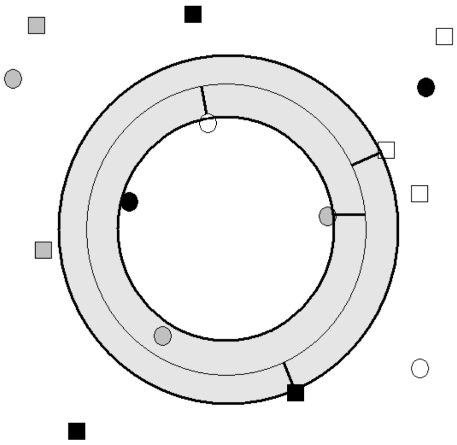

Figure 3: Construction of the two concentric balls

3 Existence of finite optimal solution

The following result explains the assumptions under which a finite optimal solution of Problem (8) can be obtained. For the proof of this result, some notions on Voronoi diagrams (see [30, 36]) are needed.
Given the sets of points $\{y_1, \dots, y_m\}$, the *nearest-point Voronoi polytope* associated with the point y_k is defined as

$$V_k = \bigcap_{i=1,\dots,m} \{x : \|x - y_k\| \leq \|x - y_i\|\}. \tag{9}$$

Analogously, the *farthest-point Voronoi polytope* associated with the point y_k is defined as

$$W_k = \bigcap_{i=1,\dots,m} \{x : \|x - y_k\| \geq \|x - y_i\|\}. \tag{10}$$

The sets $V = \{V_k : \ k = 1, \dots, m\}$ and $W = \{W_k : \ k = 1, \dots, m\}$ are called the *nearest-point* and the *farthest-point Voronoi diagrams*, respectively (see [30, 36]).

Theorem 3.1. *There exists a finite optimal solution (x_0, r_{+1}, r_{-1}) of Problem (8) iff for every choice of representatives from bags of G_{+1}, $S = (x_j)_j \in \prod_{j \in G_{+1}} X_j$, the intersection of the convex hull of this set of instances S and the convex hull of all the instances of bags of G_{-1} is non-empty, that is, if $CH(S) \cap CH(\bigcup_{i \in G_{-1}} X_i) \neq \emptyset$, $\forall S \in \prod_{j \in G_{+1}} X_j$.*

Proof.
Denote by $S = (x_j)_j \in \prod_{j \in G_{+1}} X_j$, a choice of representatives of bags in G_{+1}, define $P(S)$

as

$$\begin{array}{lll} \max\limits_{x_0, r_{+1}, r_{-1}} & r_{-1}^2 - r_{+1}^2 & \\ \text{s.t.} & \|x_j - x_0\|^2 \leq r_{+1}^2, \ \forall x_j \in S & \\ & \|x - x_0\|^2 \geq r_{-1}^2, \ \forall x \in \bigcup\limits_{i \in G_{-1}} X_i & \end{array} \tag{11}$$

and denote by $z(S)$ its optimal value, possibly $+\infty$.
With this notation, Problem (8) can be written as

$$\max_{(x_j)_j \in \prod\limits_{j \in G_{+1}} X_j} \quad z(\{x_j : \ j \in G_{+1}\}). \tag{12}$$

Firstly, we will see that if there exists a selection of representatives S for which the intersection $CH(S) \cap CH(\bigcup_{i \in G_{-1}} X_i)$ is empty, then the solution of Problem (8) is unbounded. Indeed, if this intersection is empty for this selection of representatives, one can find a hyperplane $H : \{p^\top x = c\}$, with $p \in \mathbb{R}^d$, $\|p\| = 1$ and $c \in \mathbb{R}$, separating strictly the two convex hulls, satisfying that the halfspace $\{p^\top x > c\}$ contains $CH(S)$.
Construct the nearest-point Voronoi diagram V for the complete set of instances of bags in G_{-1} and the farthest-point Voronoi diagram W associated to the set S, as defined in expressions (9)-(10). Consider the intersection of the two diagrams in the space of dimension d. The obtained tessellation has a finite number of cells.
Let C an unbounded cell of that tessellation such that, for each x in the cell, one has that $x + \lambda p$ also belongs to the cell, for any $\lambda \geq 0$, with p the vector defined previously for the hyperplane H. For this cell, there exists a point $a \in \bigcup_{i \in G_{-1}} X_i$ which is the nearest one of this set to all the points in the cell and a point $b \in S$ which is the farthest one of S to every point in the cell.
Consider x_0 any point belonging to $C \cap \{p^\top x > c\}$ and define the straight line $r = x_0 + \lambda p$, for $\lambda \in \mathbb{R}$. The radii r_{+1} and r_{-1} can be expressed as the distances from x_0 to b and a, that is, $r_{+1} = \|x_0 - b\|$ and $r_{-1} = \|x_0 - a\|$ (see [29, 31, 36] for a detailed description on the topic).
Consider a_0 and b_0 the orthogonal projections of a and b on r. Observe that $a_0 \neq b_0$, since r is orthogonal to the separating hyperplane H and then, a_0 and b_0 are also separated by H. The objective function for the feasible solution (x_0, r_{+1}, r_{-1}) can be expressed as follows,

$$\begin{aligned} r_{-1}^2 - r_{+1}^2 &= \|x_0 - a_0\|^2 + \|a - a_0\|^2 - \|x_0 - b_0\|^2 - \|b - b_0\|^2 \\ &= \|x_0 - a_0\|^2 - \|x_0 - b_0\|^2 + C. \end{aligned} \tag{13}$$

Since $p = \frac{(b_0 - a_0)}{\|b_0 - a_0\|}$, if we move x_0 along the straight line $r = x_0 + \lambda p$, for $\lambda > 0$ increasing, the closest point in $\bigcup_{i \in G_{-1}} X_i$ and the farthest point in S to the new center $x_0 + \lambda p$ continue being the same points a and b, since the new center is also included in $C \cap \{p^\top x > c\}$, and the objective function increases linearly in λ,

$$\begin{aligned} r'^2_{-1} - r'^2_{+1} &= \|x_0 + \lambda p - a_0\|^2 - \|x_0 + \lambda p - b_0\|^2 + C \\ &= \|x_0 - a_0\|^2 + 2\lambda (x_0 - a_0)^\top p - \|x_0 - b_0\|^2 - 2\lambda (x_0 - b_0)^\top p + C \\ &= \|x_0 - a_0\|^2 - \|x_0 - b_0\|^2 + C + 2\lambda (b_0 - a_0)^\top \tfrac{(b_0 - a_0)}{\|b_0 - a_0\|} \\ &= r_{-1}^2 - r_{+1}^2 + 2\lambda \|b_0 - a_0\| > r_{-1}^2 - r_{+1}^2. \end{aligned} \tag{14}$$

In fact, the larger the value of λ, the better the solution. The solution is thus unbounded.

Now, we will see that if for every choice S of representatives from bags of G_{+1}, the intersection $CH(S) \cap CH(\bigcup_{i\in G_{-1}} X_i)$ is non-empty, then a finite optimal solution of Problem (8) can be found.

For every possible set of representatives S, consider Problem (11) and build the nearest-point Voronoi diagram V for $\bigcup_{i\in G_{-1}} X_i$, the farthest-point Voronoi diagram W for S, and the tessellation obtained by intersecting the two diagrams, whose number of cells is finite. For each cell, consider $a \in \bigcup_{i\in G_{-1}} X_i$, the nearest point of this set to all the points in the cell, and $b \in S$, the farthest point of S to every point in the cell.

Thus, for any possible center x_0 in this cell, the radii r_{+1} and r_{-1} can be computed as the distances to those two points, $r_{-1} = \|x_0 - a\|$ and $r_{+1} = \|x_0 - b\|$. Problem (11) restricted to the cell can be expressed as follows,

$$\begin{array}{ll} \max\limits_{x_0} & \|x_0 - a\|^2 - \|x_0 - b\|^2 \\ \text{s.t.} & \|x_j - x_0\|^2 \leq \|x_0 - b\|^2, \ \forall x_j \in S \\ & \|x_k - x_0\|^2 \geq \|x_0 - a\|^2, \ \forall x_k \in \bigcup\limits_{i\in G_{-1}} X_i. \end{array} \tag{15}$$

In fact, Problem (15) can be rewritten as the following linear program in x_0,

$$\begin{array}{ll} \min\limits_{x_0} & 2x_0^\top(a - b) + \|b\|^2 - \|a\|^2 \\ \text{s.t.} & 2x_0^\top(b - x_j) + \|x_j\|^2 - \|b\|^2 \leq 0, \ \forall x_j \in S \\ & 2x_0^\top(x_k - a) + \|a\|^2 - \|x_k\|^2 \leq 0, \ \forall x_k \in \bigcup\limits_{i\in G_{-1}} X_i. \end{array} \tag{16}$$

The dual problem corresponding to the minimization problem (16) is the following,

$$\begin{array}{ll} \max\limits_{\lambda_j,\mu_k} & \|b\|^2 - \|a\|^2 + \sum\limits_j \lambda_j \ (\|x_j\|^2 - \|b\|^2) - \sum\limits_k \mu_k \ (\|x_k\|^2 - \|a\|^2) \\ \text{s.t.} & (1 - \sum\limits_k \mu_k) \ a + \sum\limits_k \mu_k x_k = (1 - \sum\limits_j \lambda_j) \ b + \sum\limits_j \lambda_j x_j \\ & \lambda_j, \ \mu_k \geq 0, \ \forall j, \ \forall k. \end{array} \tag{17}$$

If we denote μ_a and λ_b the lagrangian multipliers corresponding to the constraints in Problem (16) defined by a and b, the constraint in Problem (17) can also be expressed as

$$(1 - \sum_{k\neq a} \mu_k) \ a + \sum_{k\neq a} \mu_k x_k = (1 - \sum_{j\neq b} \lambda_j) \ b + \sum_{j\neq b} \lambda_j x_j. \tag{18}$$

Let $x \in CH(S) \cap CH(\bigcup_{i\in G_{-1}} X_i)$ (this point can be defined because this intersection in non-empty by assumption). This means, there exist $\mu'_k \geq 0$ ($\forall k : x_k \in \bigcup_{i\in G_{-1}} X_i$), $\sum_k \mu'_k = 1$ and $\lambda'_j \geq 0$ ($\forall j : x_j \in S$), $\sum_j \lambda'_j = 1$, such that $x = \sum_k \mu'_k x_k = \sum_j \lambda'_j x_j$. Defining $\mu_k = \mu'_k$, for $k \neq a$, and $\lambda_j = \lambda'_j$, for $j \neq b$, we obtain (18).

Hence, the dual problem is feasible and consequently its corresponding primal problem (11) will have a finite optimal solution. If we repeat the process for every cell (the number of cells is finite) and for every possible S, we conclude that there is a finite optimal solution for Problem (8).

□

Remark 3.1. *If the solution of Problem (8) is unbounded, according to the proof of Theorem 3.1, we can move the center x_0 of any original feasible solution along the direction p and the objective function continues improving. Then, the separating concentric balls are transformed in two hyperplanes $\{p^\top x = d\}$ and $\{p^\top x = e\}$, with $d > c > e$, and such that the closed halfspace $\{p^\top x \geq d\}$ contains $CH((x_j)_j)$ whereas $\{p^\top x \leq e\}$ contains $CH(\bigcup_{i\in G_{-1}} X_i)$.*

Theorem 3.1 characterizes the existence of a finite optimal solution in Problem (8). The following results will be used to describe a procedure to check the existence of solution of Problem (8) (or equivalently, to check that the two groups are not linearly separable) by solving a mixed-integer linear problem.

Theorem 3.2. *Problem (8) does not have a finite optimal solution iff there exists a selection of representatives from bags in G_{+1}, $S \in \prod\limits_{j\in G_{+1}} X_j$ and there exist $p \in \mathbb{R}^d$, $\beta \in \mathbb{R}$ such that*

$$p^\top x + \beta \geq 1, \qquad \forall x \in S, \tag{19}$$

$$p^\top x + \beta \geq 0, \quad \forall x \in \textstyle\bigcup_{j\in G_{+1}} X_j, \tag{20}$$

$$p^\top x + \beta < 1, \quad \forall x \in \textstyle\bigcup_{i\in G_{-1}} X_i. \tag{21}$$

Proof.
By Theorem 3.1, there does not exist a finite optimal solution of Problem (8) iff there exists a selection $S \in \prod\limits_{j\in G_{+1}} X_j$ such that a hyperplane $H : \{q^\top x = \alpha\}$ separates strictly the convex hulls of the sets of points S and $\bigcup_{i\in G_{-1}} X_i$. In that case, one has that

$$q^\top x \geq \alpha_S, \qquad \forall x \in S, \qquad \text{with } \alpha_S = \min_{x\in S} q^\top x, \tag{22}$$

$$q^\top x \geq \alpha_+, \quad \forall x \in \textstyle\bigcup_{j\in G_{+1}} X_j, \quad \text{with } \alpha_+ = \displaystyle\min_{x\in \bigcup_{j\in G_{+1}} X_j} q^\top x, \tag{23}$$

$$q^\top x \leq \alpha_-, \quad \forall x \in \textstyle\bigcup_{i\in G_{-1}} X_i, \quad \text{with } \alpha_- = \displaystyle\max_{x\in \bigcup_{i\in G_{-1}} X_i} q^\top x, \tag{24}$$

satisfying $\alpha_S \geq \alpha_+$ and $\alpha_S > \alpha_-$.
In case $\alpha_S > \alpha_+$, by subtracting α_+ in both sides of inequalities (22) and (23) and by dividing by $\alpha_S - \alpha_+$, we obtain

$$\frac{q^\top x - \alpha_+}{\alpha_S - \alpha_+} \geq \frac{\alpha_S - \alpha_+}{\alpha_S - \alpha_+} \quad \leftrightarrow \quad \frac{q^\top x - \alpha_+}{\alpha_S - \alpha_+} \geq 1, \ \forall x \in S, \tag{25}$$

$$\frac{q^\top x - \alpha_+}{\alpha_S - \alpha_+} \geq \frac{\alpha_+ - \alpha_+}{\alpha_S - \alpha_+} \quad \leftrightarrow \quad \frac{q^\top x - \alpha_+}{\alpha_S - \alpha_+} \geq 0, \ \forall x \in \bigcup_{j\in G_{+1}} X_j. \tag{26}$$

Likewise, by performing the same operations in expression (24) and by taking into account that $\alpha_S > \alpha_-$, one has that

$$\frac{q^\top x - \alpha_+}{\alpha_S - \alpha_+} \leq \frac{\alpha_- - \alpha_+}{\alpha_S - \alpha_+} < 1, \ \forall x \in \bigcup_{i\in G_{-1}} X_i. \tag{27}$$

Calling $p = \frac{1}{\alpha_S - \alpha_+} q$ and $\beta = -\frac{\alpha_+}{\alpha_S - \alpha_+}$ in expressions (25)-(27), we obtain expressions (19)-(21).

In case $\alpha_S = \alpha_+$, we sum $1 - \alpha_S$ in both sides of constraints (22)-(24), and we obtain

$$\begin{array}{lcll} q^\top x - \alpha_S + 1 & \geq & \alpha_S - \alpha_S + 1 & \leftrightarrow \; q^\top x + 1 - \alpha_S \geq 1, \; \forall x \in S, \\ q^\top x - \alpha_S + 1 & \geq & \alpha_+ - \alpha_S + 1 & \leftrightarrow \; q^\top x + 1 - \alpha_S \geq 1 \geq 0, \; \forall x \in \bigcup_{j \in G_{+1}} X_j, \\ q^\top x - \alpha_S + 1 & \leq & \alpha_- - \alpha_S + 1 < 1, & \quad \forall x \in \bigcup_{i \in G_{-1}} X_i, \end{array}$$

by taking into account that $\alpha_- - \alpha_S < 0$. Calling $p = q$ and $\beta = 1 - \alpha_S$, expressions (19)-(21) are obtained.

□

Theorem 3.3. *Problem (8) does not have a finite optimal solution iff the solution of the problem*

$$\begin{array}{rcr} \min\limits_{p \in \mathbb{R}^d, \beta, z \in \mathbb{R}, y_l^j \in \{0,1\}} & z & \\ s.t. & y_l^j \in \{0,1\}, \; \forall x_l \in X_j, \forall j \in G_{+1} & (28) \\ & \sum_l y_l^j = 1, \; \forall j \in G_{+1} & (29) \\ & p^\top x_l + \beta \geq y_l^j, \; \forall x_l \in X_j, \forall j \in G_{+1} & (30) \\ & p^\top x + \beta \leq z, \; \forall x \in \bigcup_{i \in G_{-1}} X_i, & (31) \end{array}$$

is strictly smaller than 1.

Proof.

By Theorem 3.2, Problem (8) does not have a finite optimal solution iff constraints (19)-(21) are satisfied for a determined set of representatives $S \in \prod_{j \in G_{+1}} X_j$, and for parameters $p \in \mathbb{R}^d$, $\beta \in \mathbb{R}$.

In the mixed-integer linear program with constraints (28)-(31), the binary variables defined in constraints (28) express that an instance x_l is chosen as the representative from its bag X_j when the corresponding variable y_l^j is equal to 1. Likewise, constraints (29) impose that exactly one representative is chosen from each bag X_j. Then, for any feasible solution of this problem, a selection of representatives from bags in G_{+1} is selected in constraints (28)-(29).

Constraints (30) represent that the expressions (19)-(20) are reached, while constraints (31) are related to expression (21). Then, expressions (19)-(21) are fully satisfied iff a solution of the mixed-integer linear problem is found with $z < 1$.

□

From now on, we will assume that the existence of optimal solution of Problem (8) has been already checked and that the two groups are not linearly separable, and hence, the classifier cannot be a hyperplane, but a ball.

4 Necessary conditions for optimality

Below, we derive some conditions that a finite feasible solution (x_0, r_{+1}, r_{-1}) must satisfy to be eligible for optimality.

For describing these optimal solutions, the concept of active point will also be necessary. An instance x from a bag of the group G_{+1} is an *active point* for the ball $B(x_0, r_{+1})$ iff the distance from x to x_0 is exactly r_{+1}, that is, $\|x - x_0\| = r_{+1}$. Thus, the set of active points of G_{+1}, which is denoted by A_{+1}, is formed by the instances which lie on the boundary of the ball $B(x_0, r_{+1})$ (respectively, the set A_{-1} of the active points of G_{-1} is formed by the instances lying on the boundary of $B(x_0, r_{-1})$).

During the proofs, we will see that a determined solution is not optimal by finding another solution which gives a better value of the objective function.

Theorem 4.1. *If (x_0, r_{+1}, r_{-1}) is an optimal solution, then there exists at least one active point in each group G_{+1} and G_{-1}, that is, the sets A_{+1} and A_{-1} of active points are non-empty.*

Proof.

Suppose that A_{+1} is empty. Since (x_0, r_{+1}, r_{-1}) is a feasible solution of Problem (8), at least one instance x_j from each bag X_j of the group G_{+1} is strictly (due to the emptiness of A_{+1}) contained inside the ball $B(x_0, r_{+1})$, that is, $\|x_j - x_0\|^2 < r_{+1}^2$.

Then, it is sufficient to define r'_{+1} as the distance from x_0 to the farthest instance (of a bag of G_{+1}) contained in the ball $B(x_0, r_{+1})$, that is, $r'_{+1} = \max\limits_{x \in (\bigcup_{j \in G_{+1}} X_j) \cap B(x_0, r_{+1})} \|x - x_0\|$, which is strictly smaller than r_{+1} and (x_0, r'_{+1}, r_{-1}) is a feasible solution which improves the value of the objective function.

On the other hand, suppose that A_{-1} is an empty set. Then, for any x belonging to any bag of G_{-1}, one has that $\|x - x_0\|^2 > r_{-1}^2$, and it is sufficient to take $r'_{-1} = \min\limits_{x \in \bigcup_{i \in G_{-1}} X_i} \|x - x_0\|$, which is strictly bigger than r_{-1}. Hence, (x_0, r_{+1}, r'_{-1}) improves the objective function.

In both cases, we conclude that the initial solution (x_0, r_{+1}, r_{-1}) cannot be optimal.

□

Theorem 4.2. *If (x_0, r_{+1}, r_{-1}) is an optimal solution, one has that:*

1. *If $r_{+1} < r_{-1}$, then there must exist at least two active points in G_{-1}.*

2. *If $r_{+1} > r_{-1}$, then there must exist at least two active points in G_{+1}.*

3. *If $r_{+1} = r_{-1}$ and $\left(\bigcup_{j \in G_{+1}} X_j\right) \cap \left(\bigcup_{i \in G_{-1}} X_i\right) = \emptyset$ (i.e., there is not any instance common to a bag in G_{+1} and to a bag in G_{-1}), then there must exist at least two active points in G_{+1} and two in G_{-1}.*

Proof.

By Theorem 4.1, if (x_0, r_{+1}, r_{-1}) is optimal, there must exist at least an active point a in G_{-1} and an active point b in G_{+1}. Below, we will see there are more active points in each case.

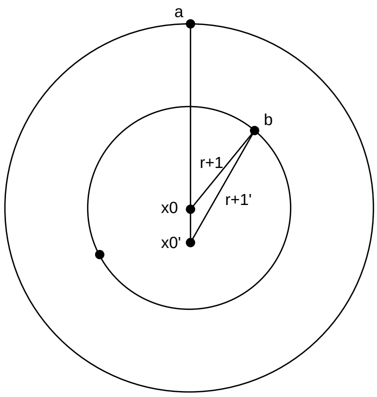

Figure 4: Illustration for the proof of Theorem 4.2, case $r_{+1} < r_{-1}$

1. When $r_{+1} < r_{-1}$, suppose there is only one active point a in the set A_{-1} (see Figure 4). Consider the direction $p = x_0 - a$ and it will be proved to be a direction of improvement for the objective function. Indeed, if we move x_0 an amount $\epsilon > 0$, small enough (for not touching new active points) in the direction $u = \frac{p}{\|p\|}$, one has that $r'_{-1} = r_{-1} + \epsilon$.

   And the other radius must be measured as the maximum distance from $x'_0 = x_0 + \epsilon u$ to the points belonging to A_{+1}.

   If the new center x'_0 is nearer to all the active points in A_{+1}, the radius $r'_{+1} \leq r_{+1}$ and the difference between the squares of the radii increases.

   Otherwise, the new radius r'_{+1} will be, at most, the distance from x'_0 to the point b which belonged to A_{+1} and which is now the furthest. Anyway, one has that $r'_{+1} \leq r_{+1} + \epsilon$, by triangular inequality on b, x_0 and x'_0, and the value of the objective function improves strictly since

$$\begin{aligned} r'^2_{-1} - r'^2_{+1} &\geq (r_{-1} + \epsilon)^2 - (r_{+1} + \epsilon)^2 \\ &= r^2_{-1} - r^2_{+1} + 2\epsilon(r_{-1} - r_{+1}) > r^2_{-1} - r^2_{+1}. \end{aligned}$$

   Hence, we have found a new feasible solution (x'_0, r'_{+1}, r'_{-1}) with a better value of the objective function of Problem (8), contradicting the optimality of (x_0, r_{+1}, r_{-1}).

2. When $r_{+1} > r_{-1}$, suppose now there is only one active point b in the set A_{+1} and consider the direction $q = b - x_0$. If we move x_0 an amount $\epsilon > 0$, small enough, in direction $v = \frac{q}{\|q\|}$, one has that $r'_{+1} = r_{+1} - \epsilon$, whereas the other radius, r'_{-1} must be taken as the minimum distance from $x'_0 = x_0 + \epsilon v$ to the points belonging to A_{-1}. And now, with a symmetric reasoning to that used in the case $r_{+1} < r_{-1}$, we obtain that (x'_0, r'_{+1}, r'_{-1}) improves the objective function.

3. When $r_{+1} = r_{-1}$, suppose there is only one active point a in the set A_{-1} and consider the direction $p = x_0 - a$. If we move x_0 along the direction $u = \frac{p}{\|p\|}$ and

with an analogous reasoning to that used in case $r_{+1} < r_{-1}$, we obtain a new feasible solution (x'_0, r'_{+1}, r'_{-1}), with $x'_0 = x_0 + \epsilon u$, $r'_{-1} = r_{-1} + \epsilon$ and r'_{+1} the distance from x'_0 to a point b in A_{+1} (the farthest one to x'_0).

If b is nearer to x'_0 than to x_0, r'_{+1} has decreased and then, the objective function has improved. Otherwise, two situations may arise: either x'_0, a and b are not collinear or $a = b$, but this latter situation cannot occur under the assumption that there are not any common instances in the bags of G_{+1} and G_{-1}. Therefore, by strict triangular inequality $r'_{+1} < r_{+1} + \epsilon$, the objective function improves since

$$r'^2_{-1} - r'^2_{+1} > (r_{-1} + \epsilon)^2 - (r_{+1} + \epsilon)^2 = r^2_{-1} - r^2_{+1}.$$

If we suppose there is only one active point b in A_{+1}, the reasoning to follow is analogous (by symmetry).

□

Remark 4.1. *For the following results, we need to add the assumption that the instances of the database Ω are in general position, in the sense that any set of n instances of the database, with $1 \leq n \leq d + 1$, are always affinely independent, that is, its rank is equal to $n - 1$.*

Remark 4.2. *Note that assumption stated in part 3 of Theorem 4.2 is less restrictive and is included in the assumption stated in Remark 4.1.*

Theorem 4.3. *Under the assumption that the data are in general position, any optimal solution has at least $d + 2$ active points associated.*

Proof.
Let $A_{+1} = \{x_{j_1}, \ldots, x_{j_s}\}$ and $A_{-1} = \{x_{k_1}, \ldots, x_{k_t}\}$ be the active points of the groups G_{+1} and G_{-1}, respectively (where $s, t \geq 1$, according to Theorem 4.1).
Consider the mediatrices of these two sets of active points,

$$\begin{aligned} med(A_{+1}) &= \{x \in \mathbb{R}^d : \|x - x_{j_1}\|^2 = \|x - x_{j_2}\|^2, \ldots, \|x - x_{j_1}\|^2 = \|x - x_{j_s}\|^2\} \quad (32) \\ med(A_{-1}) &= \{x \in \mathbb{R}^d : \|x - x_{k_1}\|^2 = \|x - x_{k_2}\|^2, \ldots, \|x - x_{k_1}\|^2 = \|x - x_{k_t}\|^2\} \quad (33) \end{aligned}$$

and the intersection of these two mediatrices, $med(A_{+1}) \cap med(A_{-1})$, is formed by these points in $\mathbb{R}^d$ satisfying simultaneously the two sets of equations, that is,

$$\begin{aligned} \|x - x_{j_1}\|^2 &= \|x - x_{j_2}\|^2 \\ &\vdots \\ \|x - x_{j_1}\|^2 &= \|x - x_{j_s}\|^2 \\ \|x - x_{k_1}\|^2 &= \|x - x_{k_2}\|^2 \\ &\vdots \\ \|x - x_{k_1}\|^2 &= \|x - x_{k_t}\|^2. \end{aligned} \quad (34)$$

Observe that (34) is equivalent to the linear system

$$\begin{array}{rcl} 2(x_{j_2}-x_{j_1})^\top x &=& \|x_{j_2}\|^2-\|x_{j_1}\|^2 \\ &\vdots& \\ 2(x_{j_s}-x_{j_1})^\top x &=& \|x_{j_s}\|^2-\|x_{j_1}\|^2 \\ 2(x_{k_2}-x_{k_1})^\top x &=& \|x_{k_2}\|^2-\|x_{k_1}\|^2 \\ &\vdots& \\ 2(x_{k_t}-x_{k_1})^\top x &=& \|x_{k_t}\|^2-\|x_{k_1}\|^2. \end{array} \tag{35}$$

which contains at most $s+t-2$ linearly independent equations.
Hence, $dim(med(A_{+1})\cap med(A_{-1}))$, the dimension of the affine space obtained by intersecting the mediatrices $med(A_{+1})$ and $med(A_{-1})$, satisfies

$$dim(med(A_{+1})\cap med(A_{-1}))\geq d-(s+t-2).$$

Suppose that the number of active points is strictly smaller than $d+2$, that is, $s+t\leq d+1$. Then,

$$dim(med(A_{+1})\cap med(A_{-1}))\geq 1.$$

In that case, the intersection of the mediatrices would contain at least a straight line. We will build one straight line where we will find a better solution.
Since the center x_0 must be at the same distance of every active point in G_{+1} (and at the same distance of every active point in G_{-1}, respectively), it must be in the intersection of the mediatrices of A_{+1} and A_{-1}, that is, x_0 satisfies the linear system (35).
Consider $a\in A_{-1}$ and $b\in A_{+1}$, two active points (see Figure 5). The objective function of Problem (8) is linear in x_0,

$$r_{-1}^2-r_{+1}^2=\|x_0-a\|^2-\|x_0-b\|^2=2(b-a)^\top x_0+\|a\|^2-\|b\|^2$$

Consider $y\in med(A_{+1})\cap med(A_{-1})$, with $y\neq x_0$, and the straight line $r=x_0+\lambda(y-x_0)$, with $\lambda\in\mathbb{R}$. This straight line is included in the intersection of the mediatrices, in fact these two manifolds coincide when $dim(med(A_{+1})\cap med(A_{-1}))=1$.
Consider a_0 and b_0 the orthogonal projections of the points a and b on r. Then, the objective function can be written as

$$\begin{array}{rcl} r_{-1}^2-r_{+1}^2 &=& \|x_0-a\|^2-\|x_0-b\|^2 \\ &=& \|x_0-a_0\|^2+\|a_0-a\|^2-\|x_0-b_0\|^2-\|b_0-b\|^2 \\ &=& 2(b_0-a_0)^\top x_0+\|a_0\|^2-\|b_0\|^2+\|a_0-a\|^2-\|b_0-b\|^2 \\ &=& 2(b_0-a_0)^\top x_0+C, \end{array} \tag{36}$$

where C is a constant depending on a, b, a_0 and b_0.
And if we take $p=b_0-a_0$, for $a_0\neq b_0$, we obtain a direction of improvement along the line in a neighborhood of the point x_0. Indeed, if we move x_0 an amount $\epsilon>0$ (small enough for not finding new active points associated to the solution) in the direction p, the new value of the objective function is

$$\begin{array}{rcl} r'^2_{-1}-r'^2_{+1} &=& 2(b_0-a_0)^\top(x_0+\epsilon(b_0-a_0))+C=r_{-1}^2-r_{+1}^2+2\epsilon\|b_0-a_0\|^2 \\ &>& r_{-1}^2-r_{+1}^2. \end{array}$$

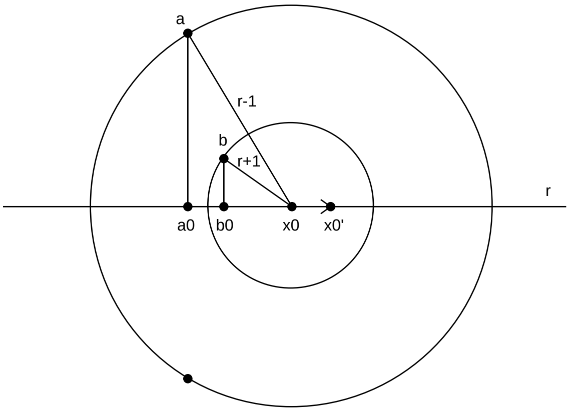

Figure 5: Illustration for the proof of Theorem 4.3

Thus, we have obtained a new solution $(x_0+\epsilon p, r'_{+1}, r'_{-1})$, with the same sets of active points and a better value of the objective function, therefore, the original solution (x_0, r_{+1}, r_{-1}) cannot be optimal.

In case $a_0 = b_0$, it is sufficient to take other two active points whose orthogonal projections on r do not coincide. This is always possible if $dim(med(A_{+1}) \cap med(A_{-1})) > 1$, because, in that case, we only have to choose a different straight line r. If $dim(med(A_{+1}) \cap med(A_{-1})) = 1$, we can always find at least two active points, $a' \in A_{-1}$ and $b' \in A_{+1}$, with different orthogonal projections, otherwise, the $d+1$ active points would lie on the same hyperplane and this is not possible under the assumption of the data being in general position.

□

Remark 4.3. *Without the general position assumption, one still obtains existence of an optimal solution with at least $d+2$ active points (although the uniqueness is not guaranteed, since other solutions with the same value of the objective function and only $d+1$ active points can be found).*

Indeed, if $dim(med(A_{+1}) \cap med(A_{-1})) = 1$ and the $d+1$ active points are cohyperplanar, their orthogonal projections on r will coincide. In that case, since $a_0 = b_0$, expression (36) remains as follows

$$r_{-1}^2 - r_{+1}^2 = \|a_0 - a\|^2 - \|b_0 - b\|^2$$

which does not depend on x_0, therefore, we can move x_0 along the straight line until a new point becomes active and the value of the objective function remains constant. Then, a solution (x'_0, r'_{+1}, r'_{-1}) with $d+2$ active points can be reached, although the solution is not unique, because any solution with x^*_0 belonging to the interval $[x_0, x'_0]$ would have the same value of the objective function (and only $d+1$ active points).

Definition 4.1. *Given two sets of points $A^+ \subset \bigcup_{j\in G_{+1}} X_j$ and $A^- \subset \bigcup_{i\in G_{-1}} X_i$, consider the intersection $med(A^+) \cap med(A^-)$. If $card(A^+ \cup A^-) = d+2$, $med(A^+) \cap med(A^-) = \{x_0\}$ and all points of A^+ and A^- are active at x_0, we say that x_0 is generated by A^+ and A^-.*

Theorem 4.3 asserts that any optimal solution is generated by its active points.

Definition 4.2. *Given a problem called P of type (8), any new problem obtained by adding instances to bags in G_{-1} and/or removing instances from bags in G_{+1} is called an extension of P.*

Lemma 4.1. *If P' is an extension of P, then the optimal value of P' is smaller than or equal to the optimal value of P.*

Proof.
Given P a problem of type (8), if we add instances to bags in G_{-1}, the resulting problem has more constraints. Likewise, if we remove instances from the bags in G_{+1}, the number of possible choices of representatives of these bags is smaller (in the combinatorial part of the problem). In both cases, the optimal value of the resulting problem P' will be smaller than or equal to the optimal value of P.

□

Theorem 4.4. *Under the assumption of the data being in general position, if (x_0, r_{+1}, r_{-1}) is an optimal solution generated by the sets A^+ and A^-, all points of A^+ come from different bags.*

Proof.
Suppose we have $d+2$ active points associated to the solution (x_0, r_{+1}, r_{-1}) of Problem (8), P, and a set B of those instances, with cardinality at least equal to two, coming from the same bag X_j in G_{+1}.
Drop that set B of active points, except for one of them, b, and consider the problem P' which is an extension of P where we have removed the set $B\backslash\{b\}$ in the bag X_j in G_{+1}. Then, (x_0, r_{+1}, r_{-1}) continues being a feasible solution with the same value of the objective function, but the number of active points is now strictly smaller than $d+2$, then the solution cannot be optimal and a solution with a better value of the objective function can be found. By Lemma 4.1, the optimal value of problem P' is a lower bound of the optimal value of problem P. Then, the solution (x_0, r_{+1}, r_{-1}) cannot be optimal for P.

□

Theorem 4.5. *If (x_0, r_{+1}, r_{-1}) is an optimal solution, then the intersection of the convex hulls of the two groups of active points A_{+1} and A_{-1} is a non-empty set.*

Proof.
Suppose $CH(A_{+1}) \cap CH(A_{-1})$, the intersection of the convex hulls of the sets of active points A_{+1} and A_{-1}, is empty. Then, one can find a hyperplane $H : \{p^\top x = c\}$ which strictly separates these two convex hulls, with $p \in \mathbb{R}^d$, $\|p\| = 1$ and $c \in \mathbb{R}$, such that the halfspace which contains $CH(A_{+1})$ is defined by $\{p^\top x > c\}$. Consider the straight line

$r = x_0 + \lambda p$, with $\lambda \in \mathbb{R}$. If we move x_0 along this straight line a certain amount $\epsilon > 0$, small enough, the objective function will be improved.
To prove this latter, it is sufficient to consider as a the active point from A_{-1} closest to the new center $x_0+\epsilon p$, and as b the point from A_{+1} which maximizes the distance from the new center to A_{+1}, and their corresponding orthogonal projections a_0 and b_0 to the straight line r. We obtain a new feasible solution $(x_0 + \epsilon p, r'_{+1}, r'_{-1})$, where $r'_{-1} = \|x_0 + \epsilon p - a\|$ and $r'_{+1} = \|x_0 + \epsilon p - b\|$. Observe that $a_0 \neq b_0$, because the straight line is orthogonal to the separating hyperplane and hence, a_0 and b_0 are also separated by the hyperplane H.
The objective function for the initial solution (x_0, r_{+1}, r_{-1}) is

$$\begin{aligned} r_{-1}^2 - r_{+1}^2 &= \|x_0 - a_0\|^2 + \|a - a_0\|^2 - \|x_0 - b_0\|^2 - \|b - b_0\|^2 \\ &= \|x_0 - a_0\|^2 - \|x_0 - b_0\|^2 + C \end{aligned}$$

On the other hand, since $p = \frac{(b_0 - a_0)}{\|b_0 - a_0\|}$, the value of the objective function for $(x_0 + \epsilon p, r'_{+1}, r'_{-1})$ is better,

$$\begin{aligned} r'^2_{-1} - r'^2_{+1} &= \|x_0 + \epsilon p - a_0\|^2 - \|x_0 + \epsilon p - b_0\|^2 + C \\ &= \|x_0 - a_0\|^2 + 2\epsilon(x_0 - a_0)^\top p - \|x_0 - b_0\|^2 - 2\epsilon(x_0 - b_0)^\top p + C \\ &= \|x_0 - a_0\|^2 - \|x_0 - b_0\|^2 + C + 2\epsilon(b_0 - a_0)^\top \frac{(b_0 - a_0)}{\|b_0 - a_0\|} \\ &= r_{-1}^2 - r_{+1}^2 + 2\epsilon\|b_0 - a_0\| > r_{-1}^2 - r_{+1}^2. \end{aligned}$$

Then, we conclude that (x_0, r_{+1}, r_{-1}) with $CH(A_{+1}) \cap CH(A_{-1}) = \emptyset$ cannot be an optimal solution.

□

5 A polynomial algorithm in fixed dimension

The results obtained in the previous section show that a finite dominating set of solutions can be built in polynomial time, when the dimension d is fixed.
Hence, an algorithm to find an optimal solution can be constructed as follows.

Algorithm 5.1.

1. *Choose $d + 2$ active points, by taking into account the conditions obtained in Theorems 4.1-4.5.*

2. *Compute the center and the radii associated.*

3. *Check the feasibility of the solution*

4. *Once given the finite dominating set of solutions (for every possible choice of $d + 2$ active points), choose the optimal one.*

For every possible choice of $d+2$ active points, satisfying the necessary conditions of optimality, one has to compute the center and the radii of the associated solution.
The center x_0 is built as the intersection of the mediatrices of the two sets of active points, A_{+1} and A_{-1}.
In other words, if $x_{i_1}, \ldots, x_{i_s}$ are the points in A_{+1}, with $1 \leq s \leq d+1$ (by Theorem 4.1), and $x_{i_{s+1}}, \ldots, x_{i_{d+2}}$ in A_{-1}, x_0 is obtained as solution to

$$\begin{array}{rcl} \|x_0 - x_{i_1}\|^2 & = & \|x_0 - x_{i_2}\|^2 \\ & \vdots & \\ \|x_0 - x_{i_1}\|^2 & = & \|x_0 - x_{i_s}\|^2 \\ \|x_0 - x_{i_{s+1}}\|^2 & = & \|x_0 - x_{i_{s+2}}\|^2 \\ & \vdots & \\ \|x_0 - x_{i_{s+1}}\|^2 & = & \|x_0 - x_{i_{d+2}}\|^2. \end{array} \tag{37}$$

Observe that (37) is equivalent to the linear system

$$\begin{array}{rcl} 2(x_{i_2} - x_{i_1})^\top x_0 & = & \|x_{i_2}\|^2 - \|x_{i_1}\|^2 \\ & \vdots & \\ 2(x_{i_s} - x_{i_1})^\top x_0 & = & \|x_{i_s}\|^2 - \|x_{i_1}\|^2 \\ 2(x_{i_{s+2}} - x_{i_{s+1}})^\top x_0 & = & \|x_{i_{s+2}}\|^2 - \|x_{i_{s+1}}\|^2 \\ & \vdots & \\ 2(x_{i_{d+2}} - x_{i_{s+1}})^\top x_0 & = & \|x_{i_{d+2}}\|^2 - \|x_{i_{s+1}}\|^2. \end{array} \tag{38}$$

and should have a unique solution (Theorem 4.3).
And the radii are computed as the distances from the center x_0 to any of the active points in each group, that is,

$$r_{-1} = \|x_0 - a\|, \quad a \in A_{-1}, \tag{39}$$

$$r_{+1} = \|x_0 - b\|, \quad b \in A_{+1}. \tag{40}$$

To check the feasibility of this solution (x_0, r_{+1}, r_{-1}), one has to study the distances from the center x_0 to each bag of G_{+1} and to each instance of a bag of G_{-1} and the following conditions must be satisfied,

$$d(x_0, X_j)^2 = \min_{x \in X_j} \|x - x_0\|^2 \leq r_{+1}^2, \forall j \in G_{+1} \tag{41}$$

$$d(x_0, x)^2 = \|x - x_0\|^2 \geq r_{-1}^2, \forall x \in X_i, \forall i \in G_{-1}. \tag{42}$$

And finally, an optimal solution will be that with the best value of the objective function. Since the solution of the linear system defined in (38) can be obtained via Gauss elimination with complexity $\mathcal{O}(d^3)$, where d is the dimension of the space of our problem, an optimal solution by using Algorithm 5.1 can be found with complexity $\mathcal{O}(d^3 n^{d+2})$, where n represents the total number of instances in the two groups. For high dimensions, this algorithm cannot be applied in an efficient way, and a heuristic methodology, described in the following section, will be used to find the solution.

6 A VNS strategy to solve the problem

For solving Problem (8), which is a mixed-integer nonlinear problem, a heuristic method, called Variable Neighborhood Search (see e.g. [17, 27] for a description), will be used to take advantage of the combinatorial structure of the problem. This method is a recent metaheuristic based on systematic change of neighborhood within a local search, for solving combinatorial and global optimization problems, which has been successfully applied in different fields, such as in location theory (see e.g. [4, 17]).
The basic VNS algorithm works as follows.

Algorithm 6.1 (Mladenovic and Hansen)**.**

- ***Initialization step.***
  Select the set of neighborhood structures $\mathcal{N}_k$, $k = 1, \ldots, k_{max}$ to be used in the search.
  Find an initial solution x.
  Choose a stopping condition.

- ***Main step.***

  1. *Set $k := 1$.*
  2. *Until $k = k_{max}$, repeat the following steps:*
     - *Generate a feasible solution x' at random from the k^{th} neighborhood of x (that is, $x' \in \mathcal{N}_k(x)$).*
     - *Apply some local search method with x' as initial solution (the new local optimum will be denoted by x'').*
     - *If the solution obtained x'' is better than x, move there ($x := x''$) and continue the search with $\mathcal{N}_1$; otherwise, set $k := k + 1$.*

Below, we describe the search space, the neighborhood structure and the local search used for performing the algorithm.

6.1 Search space

By Theorems 4.1 and 4.3, an optimal solution must have at least $d + 2$ active points, one at least in each group. These conditions are used in order to define the search space of the algorithm.
The different solutions will be determined in terms of a selection of active points of each group. Each possible selection will contain s points from the bags of group G_{+1}, with $1 \leq s \leq d + 1$, and where all these points must come from different bags, that is, no more than one instance from each bag can be chosen (by Theorem 4.4).
Then, we take $d-s+2$ points from the bags of the group G_{-1} (belonging or not to different bags).
These points will represent, respectively, the sets of active points A_{+1} and A_{-1}.

6.2 Initial solution

Two different strategies have been attempted to construct an initial solution for the VNS algorithm.

In the first strategy, the initial solution for the algorithm is built by choosing at random a set of $d + 2$ active points belonging to the search space.

In the second strategy, we perform the following steps:

- Computing the centroid C_i of each bag i in G_{+1} (that is, the arithmetical mean of the coordinates of the instances). Computing the centroid C of this set of centroids C_i.
- Choosing, for each bag in G_{+1}, the closest instance to the centroid C as the representative of that bag.
- Selecting $d + 2$ active points:
  - $s = \min(card(G_{+1}), d)$ instances from G_{+1}, the s representatives of bags that are farthest from the centroid C,
  - $d - s + 2$ instances from G_{-1}, the closest ones to C.

Observe that this set of $d + 2$ active points belongs to the search space by construction.

6.3 Neighborhood structure

The neighborhood structure is defined by taking into account the possible choices of active points. Thus, the k-th neighborhood $\mathcal{N}_k(x)$ will be formed by all the possible sets of active points obtained by modifying k elements in the configuration of active points which generate x.

The only restrictions to be imposed will be that there must be at least one instance from each group and no more than one representative from each bag of the group G_{+1} in each configuration (by Theorems 4.1 and 4.4).

6.4 Calculating the center and the radii

Given the $d+2$ active points, the center x_0 is computed as the solution of the linear system (38). If the solution is not unique, a new active point is added (and consequently, a new equation is added to the linear system).

Once the center x_0 has been found, the radii are built in order to guarantee the feasibility of the solution. The radius r_{+1} is thus obtained as

$$r_{+1}(x_0) = \max_{i \in G_{+1}} \min_{x \in X_i} \|x_0 - x\| \tag{43}$$

and the new set of active points of the group G_{+1}, will be formed by the points of the bags of G_{+1} (one or several points) with distance to the center equal to $r_{+1}(x_0)$.

Analogously, the radius r_{-1} is obtained as

$$r_{-1}(x_0) = \min_{i \in G_{-1}} \min_{x \in X_i} \|x_0 - x\| \tag{44}$$

and the new set of active points of G_{-1} is formed by the set of points (at least one) with distance to the center equal to $r_{-1}(x_0)$.

6.5 Local search

Given a set of $d+2$ active points, the corresponding solution $(x_0, r_{+1}(x_0), r_{-1}(x_0))$ is computed as explained before. But expressions (43)-(44) only guarantee two active points associated to that solution. Then, we try to recover $d+2$ active points.
For choosing the $d+2$ active points, the following steps must be performed:

- Computing the distance matrices between:
  - the instances of G_{+1} and x_0,
  - the instances of G_{-1} and x_0.
- For each bag in G_{+1}, selecting the closest instance to x_0 as its representative.
- Selecting $d+2$ active points:
  - $s = \min(card(G_{+1}), d)$ instances from G_{+1}, the s representatives of bags the furthest ones to x_0,
  - $d-s+2$ instances from G_{-1}, the closest ones to x_0.
- Computing the new center and radii.

We repeat the process until we obtain $d+2$ active points. Although the convergence is guaranteed (since the set of instances is finite), in practice, a maximum number of iterations is fixed (this is especially advisable for high values d).

6.6 Main step of the algorithm

Given a set of $d+2$ active points (an element of the search space) defining a solution x_0, we choose at random another feasible solution x_0' from the first neighbourhood of x_0, $x_0' \in \mathcal{N}_1(x_0)$. We compute the radii $r_{+1}(x_0')$ and $r_{-1}(x_0')$. We apply the local search procedure to compute the new solution x_0''.
Now, we evaluate the objective function for the new solution x_0''. If the objective value has improved, we move to x_0'', that is, we set $x_0 := x_0''$ and we continue the process in $\mathcal{N}_1(x_0)$. If the objective function has not been improved, we choose another random x_0' from the same neighbourhood and we repeat the process until having selected a maximum number of h solutions in each neighbourhood.
After h iterations in a neighbourhood, if the solution has not improved, we set $k := k+1$ and we continue the search in $\mathcal{N}_k(x_0)$, until $k = k_{max}$, where k_{max} is fixed to $d+2$ in our problem (this way, we can obtain a configuration of $d+2$ active points completely different to the original one).
Finally, the stopping condition is given by a fixed number of iterations.

7 Extensions of the VNS algorithm

7.1 The p-balls VNS algorithm

In most real databases, the value of the objective function for the solution of Problem (8) is negative, because one cannot construct the two separating concentric balls satisfying the constraints in Problem (8) and satisfying simultaneously that $r_{+1} \leq r_{-1}$. In that case, one obtains a high misclassification rate for the training sample, when trying to separate the two groups, and consequently, bad results for classification in the test sample.
A strategy to improve these results is to modify the initial classification rule (1) to allow the introduction of p separating balls. Thus, the new classification rule is defined now in terms of p balls, each ball l with center $x_{0,l} \in \mathbb{R}^d$ and radius $r_l \in \mathbb{R}_+$, $l = 1, \ldots, p$. According to the MI assumption, the classification rule remains as follows:
Given a bag $X \subset \mathbb{R}^d$,

$$\begin{array}{l} \bullet \text{ classify in } G_{+1}, \text{ if } \exists x \in X, \ \exists l \in \{1, \ldots, p\} \text{ such that } \|x - x_{0,l}\|^2 < r_l^2 \\ \bullet \text{ classify in } G_{-1}, \text{ otherwise, i.e., if } \forall x \in X, \ \forall l = 1, \ldots, p, \ \|x - x_{0,l}\|^2 \geq r_l^2. \end{array} \tag{45}$$

In the algorithm, before the training of the classifier, we apply the k-means clustering algorithm (see e.g. [18]), with $k = p$, to build clusters with the bags in G_{+1}. Given the bags of G_{+1}, the following steps are performed:

1. Computing the centroid C_i of each bag i in G_{+1}.

2. Initial assignment: the set of bags is partitioned at random in p clusters (with the same size).

3. Computing the centroid $\tilde{C}_l$ of each cluster l, $l = 1, \ldots, p$ (the mean of the centroids C_i of the bags assigned to that cluster).

4. Computing the distance matrix between the centroid C_i of the i-th bag, $i \in G_{+1}$, and the centroid $\tilde{C}_l$ of the l-th cluster, $l = 1, \ldots, p$.

5. Assigning the bag i to the cluster whose centroid $\tilde{C}_l$ is the closest one to C_i.

6. Repeating steps 3-5 while there are some changes in the assignment or while a fixed number of iterations is not reached.

Once the clusters have been constructed, we apply the VNS algorithm described in Section 6 for $G_{+1,l}$ being the bags of the cluster l and $G_{-1,l} = G_{-1}$, $l = 1, \ldots, p$, we compute the p balls (for the p clusters) and we classify the testing sample with the rule (45).

7.2 Multi-class case

So far, we have only dealt with the classification problem for two groups. However, in many real situations, more than two groups appear in a classification problem. Different strategies can be found in the literature, most of them proposing to transform a multi-class problem in a series of two-class problems to be solved (see e.g., [19, 21, 35]).
We will use the *one-versus-one* (see [15]) algorithm (1-v-1) for our experiments. In this algorithm, one has to construct a classifier for every possible pair of groups i and j. Since

the classification rule is not symmetric, we will need to build the ball where the group i is G_{+1} and the group j is G_{-1} (we denote it by $B(x_0^{i,j}, r^{i,j})$) and the opposite (the ball $B(x_0^{j,i}, r^{j,i})$). Then, we need to construct $N(N-1)$ classifiers in total.
Given a bag X to be classified, for every pair of groups i and j, we compute:

$$intensity(i,j) = \frac{\min_{x \in X} d(x, x_0^{i,j})^2}{(r^{i,j})^2}, \tag{46}$$

and we give one vote to group i, if $intensity(i,j) < intensity(j,i)$, or one vote to group j, otherwise. X is finally assigned to the group with the highest number of votes (following the Max Wins algorithm, [15]).
In case of tie between two groups i and j, we go back to compare the intensities for these two groups and we assign the bag to group i if $intensity(i,j) < intensity(j,i)$ (else, to group j).

8 Computational experiment

The classification problem and the VNS algorithm proposed for building the classifier have been implemented by using Matlab 6.5 on a computer with Pentium IV CPU 3.06 GHz. Several numerical experiments have been performed, with artificial and real databases.
In the experiments with artificial databases, we have solved the classification problem through 10-fold cross validation, that is, the bags of the database are grouped in 10 sets (these sets form a partition) and, in turns, 9 of these sets are used for training the model and the last one is used to test the problem, that is, the process is repeated ten times (see [20, 22] for a description of the method). With the real databases in Subsection 8.5, we have performed 5-fold cross validation.
With the training sample, we have applied the VNS algorithm to build the center x_0 and the radius r which define the classifier, and we have measured the classification accuracy, i.e., the percentage of well-classified bags, for the training sample and, later, for the test sample. The parameter k_{max} in the VNS algorithm has been fixed to $d+2$ (hence, it depends on the dimension of the problem), while the parameter h (the number of different solutions taken in each neighbourhood) and the maximum number of iterations in the local search step (see Subsection 6.5) have been both set equal to 5.

8.1 Full enumeration vs VNS algorithm

In this first experiment, we compare the results obtained via the VNS algorithm with those obtained via a full enumeration of the finite dominating set of solutions (obtained by Algorithm 5.1) in the optimization problem (8).
Since full enumeration is not an efficient way of obtaining solutions in high dimension, two sets G_{+1} and G_{-1} have been built in dimension $d = 2$ with 50 instances in each one. Polar coordinates have been used for generating the instances. Thus, for an instance $(\rho \cos\theta, \rho \sin\theta)$ of G_{-1}, θ comes from a uniform distribution $U(-\pi, \pi)$, and ρ is chosen from a uniform distribution $U(r_1, r_2)$, where $0 < r_1 < r_2$ (r_1 was fixed to 1 and r_2 to 2 for the experiment). The instances in G_{+1} were generated in the same way, excepting

Number of bags	1	2	5	10	50
Complete enumeration	1.9315	3.8101	0.3592	0.3036	0.0464
VNS (random initial solution)	1053.4	1664.6	0.3592	0.3036	0.0464
VNS (heuristic initial solution)	1053.4	1664.6	0.3592	0.3036	0.0464

Table 1: Value of the objective function for complete enumeration and for VNS

one instance in each bag which must be included in $B(0, r_1)$ (that is, ρ was chosen from a uniform distribution $U(0, r_1)$). That way, the spherical separability of the bags (in the sense that there exists a feasible solution (x_0, r_{+1}, r_{-1}) of Problem (8) satisfying that $r_{+1} < r_{-1}$) is guaranteed.

The experiment was repeated with 50 instances in each group and changing the number of bags in the group G_{+1} (the numbers of bags were 1, 2, 5, 10 and 50), each bag with the same number of instances. Observe that the problem remains the same by changing the number of bags in G_{-1}.

In Table 1, we show the values of the objective function obtained for Problem (8) by using a complete enumeration and the VNS algorithm, for 1000 iterations, with the two possible initial solutions described in Subsection 6.2 (random and heuristic centroid-based initial solutions).

One can observe that the VNS algorithm obtains quite similar results to those obtained via a full enumeration, excepting the case of only one and two bags in G_{+1}. However, in that case, the real solution is $+\infty$ for the two cases, since the procedure to check the existence of finite optimal solution (described in Section 4) indicates that the sets are linearly separable.

A comparison of the behaviour of the objective function, for the two types of initial solution (random and heuristic) with respect to the number of iterations is depicted in Figure 6. We can observe that, in the three datasets, the two versions of the algorithm reach the optimal solution (or a very close value) with only a few iterations. However, the algorithm with the heuristic initial solution reaches, in general, the optimal solution before, since a better value to initialize the process is given.

Anyway, in this work, we are not really interested in obtaining the best solution of Problem (8) but rather in obtaining competitive results for the classification problem.

8.2 Artificial database with spherically separable sets of instances

From now on, the experiments described herein illustrate the classification problem.

In this experiment, we have built one artificial database, where the instances of G_{+1} and G_{-1} are spherically separable , in the following way. For each group, 2000 instances have been generated coming from uniform distributions: the instances of G_{+1} via a uniform distribution with parameters -10 and 10, $U(-10, 10)$, and the instances of G_{-1} via a uniform distribution with parameters -20 and 20, $U(-20, 20)$, and by taking into account that at least one coordinate does not belong to the interval $(-10, 10)$.

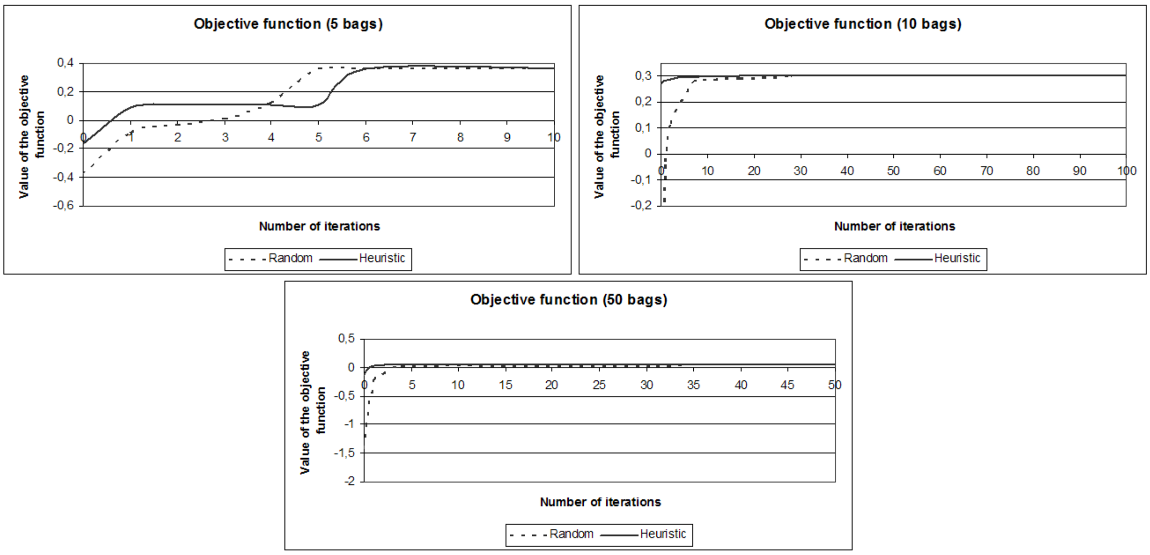

Figure 6: Behaviour of the objective function with the VNS algorithm (sets with 5, 10 and 50 bags)

The instances of each class have been grouped in 100 bags (20 instances per bag) and we have repeated the process for different dimensions ($d = 2, 3, 10, 20, 30, 50, 100$).
In all the cases, an accuracy of 100% was found in each group for the training and the test sample in every dimension.

8.3 Artificial database with spherically separable sets of bags

Another separable artificial database for the classification problem has been constructed as follows. For each class (G_{+1} and G_{-1}), we have generated a total of 2000 instances and 100 bags (20 instances per bag). The instances of G_{-1} are generated as in the latter database, that is, via a uniform distribution $U(-20, 20)$, with at least one coordinate not belonging to the interval $(-10, 10)$. On the other hand, one instance of each bag of G_{+1} is generated via $U(-10, 10)$, while the rest of instances come from $U(-20, 20)$.
The average accuracy for the 10 runs for different dimensions and for the test and training samples are given in Table 2. The number of iterations for the VNS algorithm was fixed to 2000 (without local search).
One can remark that the accuracies are quite satisfactory. Moreover, for the highest dimension ($d = 100$), the accuracy is 100% in both cases (test and training sample). This is probably due to the fact that the higher the dimension, the more solutions are considered in the VNS algorithm, since the $k_{\max}$, that is, the maximum neighborhood radius taken into account, is fixed in our implementation to $d + 2$. Hence, the number of solutions studied depends on the dimension of the problem.

	Dim	d=2	d=3	d=10	d=20	d=30	d=50	d=100
Test	G_{+1}	100	92	93	97	97	97	100
	G_{-1}	100	90	87.69	93	97	99	100
	Total	100	91	90.35	95	97	98	100
Train	G_{+1}	100	92.22	96.44	99.78	99.89	100	100
	G_{-1}	100	95.57	78.86	98.56	100	100	100
	Total	100	93.9	87.65	99.17	99.95	100	100

Table 2: Accuracy for uniform artificial database

	Dim	d=2	d=3	d=5	d=10	d=20	d=30	d=50	d=100
Test	G_{+1}	96	88	97	100	100	100	100	100
	G_{-1}	87	89	96	100	99	100	100	100
	Total	91.5	88.5	96.5	100	99.5	100	100	100
Train	G_{+1}	100	99.89	99.67	100	100	100	100	100
	G_{-1}	87.56	89.33	97.45	100	100	100	100	100
	Total	93.78	94.61	98.56	100	100	100	100	100

Table 3: Accuracy for gaussian artificial database

8.4 Artificial dataset based on a gaussian distribution

For this database, 100 bags, each one with 20 instances, have been generated for each group, G_{+1} and G_{-1}, by using a gaussian distribution.

Each coordinate of the mean vector of each bag in G_{+1} comes from a uniform distribution $U(-1, 1)$, while the coordinates of the mean vector of the bags in G_{-1} come from $U(-5, 5)$. The instances of each bag are generated from a multivariate normal distribution with the corresponding mean vector and the identity as the covariance matrix.

Table 3 shows the average accuracy for the 10 runs of the cross-validation process in the test and training samples, for different dimensions (with 5000 iterations in the VNS algorithm which builds the classifier and without local search).

One can observe that the accuracy for the highest dimensions is better, in both samples (training and test), because for a high dimension, the databases built herein become more easily separable.

8.5 Real database for image categorization

Finally, we have applied our algorithm to a real database for image categorization. Image categorization consists in labeling images into a set of predefined categories.

The image database is a set of 2000 images in JPEG format taken from 20 CD-ROMs published by the COREL Corporation, each CD-ROM containing 100 images representing a different concept. This dataset was previously used for Multi-instance Learning in [6, 7], and it is available at the webpage http://www.cs.olemiss.edu/∼ychen/ddsvm.html.

Class	Class name	Instances per bag (average)
0	African people and villages	4.84
1	Beach	3.54
2	Historical building	3.1
3	Buses	7.59
4	Dinosaurs	2.00
5	Elephants	3.02
6	Flowers	4.46
7	Horses	3.89
8	Mountain and glaciers	3.38
9	Food	7.24
10	Dogs	3.80
11	Lizards	2.80
12	Fashion models	5.19
13	Sunset scenes	3.52
14	Cars	4.93
15	Waterfalls	2.56
16	Antique furniture	2.30
17	Battle ships	4.32
18	Skiing	3.34
19	Desserts	3.65

Table 4: Description of the image database

A segmentation process was applied to these images to extract some properties about luminance, color and texture of the pictures and they were encoded into feature vectors. These feature vectors were grouped into clusters, representing the regions of the segmented image. Then, each image has several regions, where each region is characterized by a feature vector in dimension $d = 9$, representing the color, texture and shape properties of that region (see [6, 7] for a more detailed description).
From the Multiple Instance Learning framework, the different concepts (CD-ROMs) are the groups which the images will be assigned to, the images are the bags of the database, and the regions are the instances of each bag. In this dataset, there are 20 groups, 100 bags in each group and the average number of instances per bag for the different groups is displayed in Table 4 (along with the name of the groups). The dimension of the problem is $d = 9$. We have performed several experiments with only the first 10 groups (1000-Image database) and with the complete database (2000-Image database).

Since the database has more than two classes, the 1-v-1 algorithm, explained in Subsection 7.2, is the selected tool to solve the multi-class problem, and for every pair of groups i and j, the p-balls VNS algorithm, described in Subsection 7.1, is used to build the classifier.
First, we have considered the problem with only the first ten classes. For selecting the training and test samples, we have used 5-fold cross validation on the database. Different

		Assigned class									
		0	1	2	3	4	5	6	7	8	9
Real class	0	**67**	3	4	0	2	11	1	3	3	6
	1	2	**58**	7	3	1	8	1	0	17	3
	2	4	3	**75**	2	1	7	1	0	5	2
	3	1	3	10	**67**	9	2	0	1	1	6
	4	0	0	0	0	**100**	0	0	0	0	0
	5	10	4	2	0	0	**71**	0	5	8	0
	6	2	0	0	0	0	1	**94**	2	0	1
	7	3	1	0	0	0	14	0	**81**	1	0
	8	0	18	3	0	0	10	0	0	**69**	0
	9	5	4	1	2	5	4	0	3	5	**71**

Table 5: Confusion matrix for the 1000-Image database

values for p (in the optimization algorithm to construct the separating balls) have been considered (from $p = 1$ to $p = 20$), although we only show the best results, which were obtained for $p = 15$. The number of iterations to obtain each ball is set equal to 50, and the solution based on the centroid (see Subsection 6.2) was taken as the initial solution.

Table 5 displays the confusion matrix for the test samples in the database with the first 10 classes. Each element (i, j) of this matrix represents the percentage of elements of the class i which has been assigned to the class j. The elements of the diagonal (in bold) represent the percentage of elements correctly labeled for each class. Then, one can observe that the class 4 (dinosaurs) is easily separable from the rest of classes, since all its elements have been correctly classified. However, some problems appear to distinguish between classes 0 and 6 (African people or villages and elephants), or especially between two kind of landscapes: images of beaches and images of mountains or glaciers (classes 1 and 8), where we obtain 17% of beaches misclassified as mountains or glaciers, and 18% in the other direction. Likewise, 14% of horses (class 7) is labeled as elephants (class 5).
Table 6 shows the accuracy for every class, that is, the percentage of elements of every class which has been correctly labeled into its class, in the training and the test samples. One can observe that the performance of the algorithm is quite good in most of the classes in the training sample, and, in general, a class which is easily separable from the rest in the training sample, continues being easily discriminated in the test sample. This is the case of class 4, with 100% of accuracy in both the training and the test samples. However, we can also find some classes, like class 3 (buses) with a much better accuracy in the training (96.5%) than in the test sample (only 67%).
Table 7 presents the classification accuracy for every class in the complete dataset (2000-Image database). The performance of our algorithm is good in the training sample in most of the classes (except for class 8), showing the power of our methodology to separate the bags of the different concepts. In the test sample, the accuracy is lower than for the 1000-Image database, although good results are obtained for separating classes such as number 4 and 6 (dinosaurs and flowers).

Class	0	1	2	3	4	5	6	7	8	9
Test	67	58	75	67	100	71	94	81	69	71
Train	83.75	78.5	90.5	96.5	100	86.25	99.5	96.25	75.5	97.75

Table 6: Accuracy for the 1000-Image database

Class	0	1	2	3	4	5	6	7	8	9
Test	52	58	69	67	97	55	88	78	42	67
Train	84.25	83.5	91	93.25	100	83.25	98.75	92	68.75	95.25
Class	10	11	12	13	14	15	16	17	18	19
Test	44	63	64	57	57	76	81	60	49	28
Train	86	82.5	96	92.25	92	92.5	99.25	95.5	84	74.75

Table 7: Accuracy for the 2000-Image database

In Table 8, the accuracy for the test and training samples in the two databases are shown. One can observe that we obtain very good results for the training sample (even with the complete dataset), around 90%, and the results for the test sample are quite competitive. Finally, we compare the results we have obtained, with the results obtained via other methods which have been used in this database: MILES algorithm [6], DD-SVM [7], MI-SVM [1] and k-means-SVM [10]. These other algorithms have also been tested over five test sets extracted at random from the database, but the technique is not cross validation (see [6]). Although our algorithm does not get to improve the best results obtained so far, our results are comparable with the solutions obtained for this database, and in fact, we get to improve the performance of other algorithms based on SVMs (like MI-SVM and k-means-SVM) in this multi-class problem in both datasets.

9 Conclusions and extensions

In this work, a new tool for solving classification problems in Multiple Instance Learning has been described. The problem, which has an easy geometric interpretation, has been formulated as a nonlinear mixed integer optimization problem. The existence of finite optimal solution has been studied and necessary conditions for optimal solutions have been deduced.

	1000-Image database	2000-Image database
Test	75.3	62.6
Train	90.45	89.24

Table 8: Accuracy for the two databases

Algorithms	1000-Image database	2000-Image database
MILES	82.6	68.7
DD-SVM	81.5	67.5
MI-SVM	74.7	54.6
k-means-SVM	69.8	52.3
p-balls VNS algorithm	75.3	62.6

Table 9: Accuracy for different algorithms for the image database

These optimality conditions have been considered to develop the heuristic algorithm (a Variable Neighbourhood Search algorithm) to solve the model. The computational results for the classifier show that our tool is competitive, especially with separable databases.
The introduction of p balls in the classification rule for the benchmark datasets has improved remarkably the performance of the algorithm. The selection of the optimal value for p is a topic to be considered. In fact, a method to choose automatically the most suited value of p in each training sample during the cross-validation process seems to be an interesting problem for further works.
The problem can be extended by considering different assumptions for building the classification rule, by changing the objective function of the problem, even by proposing a biobjective problem since some constraints may be relaxed.
The introduction of kernel structures in the problem and the extension to the Multi-Instance Regression problem, as described in [12, 34], are other topics which deserve further studies. Likewise, another interesting extension could be the application of this model in location theory (for example, in location of semi-obnoxious facilities).

References

[1] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support Vector Machines for Multiple-Instance Learning. In *Proc. Advances in Neural Information Processing Systems*, volume 15, pages 561–568, (2002).

[2] R. C. Bunescu and R. J. Mooney. Multiple Instance Learning for Sparse Positive Bags. In *Proc. 24th International Conference on Machine Learning*, pages 105–112, (2007).

[3] C. J. C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, (1998).

[4] E. Carrizosa, J. Gordillo, and D. R. Santos-Peñate. Covering Models with Time-dependent Demand. Optimization Online, http://www.optimization-online.org/DB_FILE/2007/03/1613.pdf, (2007).

[5] E. Carrizosa and F. Plastria. On Minquantile and Maxcovering Optimisation. *Mathematical Programming*, 71:101–112, (1995).

[6] Y. Chen, J. Bi, and J. Z. Wang. MILES: Multiple-Instance Learning via Embedded Instance Selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):1931–1947, (2006).

[7] Y. Chen and J. Z. Wang. Image Categorization by Learning and Reasoning with Regions. *Journal of Machine Learning Research*, 5:913–939, (2004).

[8] P-M. Cheung and J. T. Kwok. A Regularization Framework for Multiple-Instance Learning. In *Proc. 23rd International Conference on Machine Learning*, pages 193–200, (2006).

[9] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and other Kernel-based Learning Methods.* Cambridge University Press, Cambridge, (2000).

[10] G. Csurka, C. Bray, C. Dance, and L. Fan. Visual Categorization with Bags of Keypoints. In *Proc. ECCV'04 Workshop Statistical Learning in Computer Vision*, pages 59–74, (2004).

[11] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez. Solving the Multiple Instance Problem with Axis-Parallel Rectangles. *Artificial Intelligence*, 89:31–71, (1997).

[12] D. R. Dooly, Q. Zhang, S. A. Goldman, and R. A. Amar. Multiple-Instance Learning of real-valued Data. *Journal of Machine Learning Research*, 3:651–678, (2002).

[13] J. Eichborn and O. Chapelle. Object Categorization with SVM: Kernels for Local Features. Technical report, Max Planck Institute for Biological Cybernetics, (2004).

[14] E. Frank and X. Xu. Applying Propositional Learning Algorithms to Multi-instance Data. Working Paper. University of Waikato, (2003).

[15] J. H. Friedman. Another Approach to Polychotomous Classification. Technical report, Stanford University, (1996).

[16] T. Gärtner, P. A. Flach, A. Kowalczyk, and A. J. Smola. Multi-Instance Kernels. In *Proc. 19th International Conference on Machine Learning*, pages 179–186, (2002).

[17] P. Hansen and N. Mladenovic. Variable Neighborhood Search: Principles and Applications. *European Journal of Operational Research*, 130:449–467, (2001).

[18] J. A. Hartigan. *Clustering Algorithms.* John Wiley and Sons, New York, (1975).

[19] T. Hastie and R. Tibshirani. Classification by Pairwise Coupling. *The Annals of Statistics*, 26(2):451–471, (1998).

[20] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer-Verlag, New York, (2001).

[21] C-W. Hsu and C-J. Lin. A Comparison of Methods for Multi-class Support Vector Machines. *IEEE Transactions on Neural Networks*, 13(2):415–425, (2002).

[22] R. Kohavi. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 1137–1143, (1995).

[23] R. Kondor and T. Jebaram. A Kernel between Sets of Vectors. In *Proc. 21th International Conference on Machine Learning*, pages 361–368, (2003).

[24] O. L. Mangasarian and E. W. Wild. Multiple Instance Classification via Successive Linear Programming. *Journal of Optimization Theory and Applications*, To appear, (2008).

[25] O. Maron and T. Lozano-Pérez. A Framework for Multiple-Instance Learning. In *Proc. Advances in Neural Information Processing Systems*, volume 10, pages 570–576, (1998).

[26] O. Maron and A. L. Ratan. Multiple-Instance Learning for Natural Scene Classification. In *Proc. 15th International Conference on Machine Learning*, pages 341–349, (1998).

[27] N. Mladenovic and P. Hansen. Variable Neighborhood Search. *Computers and Operations Research*, 24:1097–1100, (1997).

[28] J. M. Moguerza and A. Muñoz. Support Vector Machines with Applications. *Statistical Science*, 21(3):322–336, (2006).

[29] Y. Ohsawa. Bicriteria Euclidean Location Associated with Maximin and Minimax Criteria. *Naval Research Logistics*, 47:581–592, (2000).

[30] A. Okabe, B. Boots, and K. Sugihara. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams.* John Wiley and Sons, Chichester, (1992).

[31] A. Okabe and A. Suzuki. Locational Optimization Problems Solved through Voronoi Diagrams. *European Journal of Operational Research*, 98:445–456, (1997).

[32] H. T. Pao, S. C. Chung, Y. Y. Xu, and H-C Fu. An EM based Multiple Instance Learning Method for Image Classification. *Expert Systems with Applications*, To appear, (2008).

[33] F. Plastria and E. Carrizosa. Undesirable Facility Location with Minimal Covering Objective. *European Journal of Operational Research*, 119(1):158–180, (1999).

[34] S. Ray and D. Page. Multiple Instance Regression. In *Proc. of the 18th International Conference on Machine Learning*, pages 425–432, (2001).

[35] B. Schölkopf and A. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond.* MIT Press, Cambridge, (2002).

[36] M. I. Shamos and D. Hoey. Closest-Point Problems. In *Proc. of the 16th Annual IIEE Symposium on Foundations of Computer Science*, pages 151–162, (1975).

[37] J. Wang and J. D. Zucker. Solving the Multiple-Instance Problem: A Lazy Learning Approach. In *Proc. 17th International Conference on Machine Learning*, pages 1119 – 1126, (2000).

[38] N. Weidmann, E. Frank, and B. Pfahringer. A Two-level Learning Method for Generalized Multi-instance Problems. In *Proc. 14th European Conference on Machine Learning*, pages 468–479, (2003).

[39] X. Xu. Statistical Learning in Multiple Instance Problems. Master's thesis, University of Waikato, (2003).

[40] C. Zhang, X. Chen, M. Chen, S. C. Chen, and M. L. Shyu. A Multiple Instance Learning Approach for Content Based Image Retrieval Using One-Class Support Vector Machine. In *Proc. IEEE International Conference on Multimedia and Expo*, pages 1142– 1145, (2005).

[41] Q. Zhang and S. A. Goldman. EM-DD: An Improved Multiple-Instance Learning Technique. In *Proc. Advances in Neural Information Processing Systems*, volume 14, pages 1073–1080, (2002).

[42] Z-H. Zhou. Multi-Instance Learning: A Survey. Technical report, AI Lab, Department of Computer Sciences and Technology, Nanging University, (2004).

[43] Z-H. Zhou, K. Jiang, and M. Li. Multi-Instance Learning Based Web Mining. *Applied Intelligence*, 22(2):135–147, (2005).