Comparing Linear and Spherical Separation Using Grossone-Based Numerical Infinities in Classification Problems

Annabella Astorino and Antonio Fuduli

Abstract We investigate the role played by the linear and spherical separations in binary supervised learning and in Multiple Instance Learning (MIL), in connection with the use of the grossone-based numerical infinities. While in the binary supervised learning the objective is to separate two sets of samples, a binary MIL problem consists in separating two different type of sets (positive and negative), each of them constituted by a finite number of samples. We remind that using the spherical separation in classification problems provides an advantage especially in terms of computational time, since, when the center of the separating sphere is (judiciously) fixed in advance, the corresponding optimization problem reduces to a structured linear program, easily solvable by an ad hoc algorithm. In particular, by embedding the grossone idea, here we analyze the case where the center of the sphere is selected far from both the two sets, obtaining in this way a kind of linear separation. This approach is easily extensible to the margin concept (of the type adopted in the Support Vector Machine technique) and to MIL problems. Some numerical results are reported on classical binary datasets drawn from the literature.

A. Astorino (✉)
Institute for High Performance Computing and Networking (ICAR), National Research Council, Rende, Italy
e-mail: annabella.astorino@icar.cnr.it

A. Fuduli
Department of Mathematics and Computer Science, University of Calabria, Rende, Italy
e-mail: antonio.fuduli@unical.it

Y. D. Sergeyev and R. De Leone (eds.), *Numerical Infinities and Infinitesimals in Optimization*, Emergence, Complexity and Computation 43,
https://doi.org/10.1007/978-3-030-93642-6_10

1 Introduction

Classification problems in mathematical programming concern separation of sample sets by means of an appropriate surface. This field, entered by many researchers in optimization community in the last years, is a part of the more general machine learning area, aimed at providing automated systems able to learn from human experiences.

The objective of pattern classification is to categorize samples into different classes on the basis of their similarities. More formally, given a set of labelled and unlabelled samples, characterized by some features, for each of them we want to express a particular feature, the class label, as a function of the remaining ones. This is done by constructing a prediction function, by means of which we would like to predict the class of each sample. In machine learning literature many approaches [14] have been indeed devised for automatically distinguishing among different samples on the basis of their patterns: approaches of supervised, unsupervised and semi-supervised learning, and more recently approaches of Multiple Instance Learning. In particular, in the supervised case most of the learning models apply the inductive inference concept, where the prediction function, derived only from the labelled input data, is used to predict the label of any future object. A well established supervised technique is the Support Vector Machine (SVM) [33, 58], which has revealed a powerful classification tool in many application areas.

A widely adopted alternative to supervised classification is the unsupervised one, where all the objects are unlabelled: as a consequence, in such case, the prediction function is constructed by clustering the data on the basis of their similarities [22, 28]. In the middle we find the semisupervised techniques [29], that apply the transductive inference concept: the prediction function is derived from the information concerning all the available data (both labelled and unlabelled samples). This function is not aimed at predicting the class label of newly incoming samples, but only at making a decision about the currently available unlabelled objects. Some useful references are [5, 30], the latter being a semisupervised version of the SVM technique.

A more recent classification framework is constituted by the Multiple Instance Learning (MIL) [42], whose main difference with respect to the traditional supervised learning scenario resides in the nature of the learning samples. In fact, each sample is not represented by a fixed-length vector of features but by a bag of feature vectors that are referred to as instances. The classification labels are only provided for the entire training bags whereas the labels of the instances inside them are unknown. The task is to learn a model that predicts the labels of the new incoming bags, possibly together with the labels of the instances inside them. A seminal SVM-type MIL paper is [1], while some recent articles are [11–13, 15, 17, 21, 24, 40, 49].

In this work, strictly connected with [7], starting from the spherical binary supervised classification approach reported in [16], we introduce some spherical separation models for both supervised learning and Multiple Instance Learning. Such models are obtained by embedding the grossone-based numerical methodology [53], which allows to select the center of the sphere far from both the two sets (of samples or of bags), providing a kind of linear separation.

Spherical separation falls into the class of the nonlinear separation surfaces [10, 14], differently, for example, from the well known supervised learning SVM technique [33, 58], where a classifier is constructed by generating a hyperplane far away from the points of the two sets. Also the SVM approach allows to obtain general nonlinear classifiers by adopting kernel transformations. In this case the basic idea is to map the data into a higher dimensional space (the feature space) and to separate the two transformed sets by means of a hyperplane, that corresponds to a nonlinear surface in the original input space. The main advantage of spherical separation is that, once the center of the sphere is heuristically fixed in advance, the optimal radius can be found quite effectively by means of simple sorting algorithms such as those ones reported in [9, 16]. No analogous simplification strategy is apparently available if one adopts the SVM approach. Moreover, another advantage is to work directly in the input space. In fact to keep, whenever possible, the data in the original space seems appealing in order to stay close to the real life modeled processes. Of course kernel methods are characterized by high flexibility, even if sometimes they provide results which are hard to be interpreted in the original input space, differently from the nonlinear classifiers acting directly in such space (see for example [19, 20]).

The chapter is organized in the following way. In the next section we focus on supervised classification, distinguishing between linear and spherical separation, the latter suitable for grossone application (see [7]). In Sect. 3 we discuss the possibility to extend the grossone spherical separation to Multiple Instance Learning, while in Sect. 4 we comment the numerical results published in [7], which confirm the practical applicability of the grossone-based numerical infinities in classification problems. Finally some conclusions are drawn in Sect. 5.

2 Linear and Spherical Separability for Supervised Classification

Let

$$\mathcal{A} = \{a_1, \ldots, a_m\}, \quad \text{with } a_i \in \mathbb{R}^n, \ i = 1, \ldots, m$$

and

$$\mathcal{B} = \{b_1, \ldots, b_k\}, \quad \text{with } b_l \in \mathbb{R}^n, \ l = 1, \ldots, k,$$

be the two finite sets of samples (points in $\mathbb{R}^n$). The classical binary classification problem consists in discriminating between $\mathcal{A}$ and $\mathcal{B}$ by means of a separating surface, obtained by minimizing any classification error. Such surface can be a hyperplane (linear separation) or a nonlinear surface, such as a sphere (spherical separation). A seminal paper on linear separation appeared in 1965 by Mangasarian [47], while the first approach for pattern classification based on a minimum volume sphere dates back to 1999 by Tax and Duin [56].

2.1 Linear Separation

The two sets $\mathcal{A}$ and $\mathcal{B}$ are linearly separable if and only if there exists a hyperplane

$$H(w, \gamma) = \{x \in \mathbb{R}^n \mid w^T x = \gamma\}, \ \text{with } w \in \mathbb{R}^n \text{ and } \gamma \in \mathbb{R},$$

such that

$$w^T a_i \leq \gamma - 1 \qquad i = 1, \ldots, m$$

and

$$w^T b_l \geq \gamma + 1 \qquad l = 1, \ldots, k.$$

A geometrical characterization of linear separability is that $\mathcal{A}$ and $\mathcal{B}$ are linearly separable if and only if their convex hulls do not intersect, i.e.

$$\text{conv}(\mathcal{A}) \cap \text{conv}(\mathcal{B}) = \emptyset,$$

as depicted in Fig. 1, where the two cases of linearly separable and inseparable sets are considered.

The problem of finding a separating hyperplane can be formulated as a linear program [23], but several other approaches have been proposed, such as the SVM technique [33, 58], where the idea is to generate a separation hyperplane far away from the objects of both the two sets. This is done by

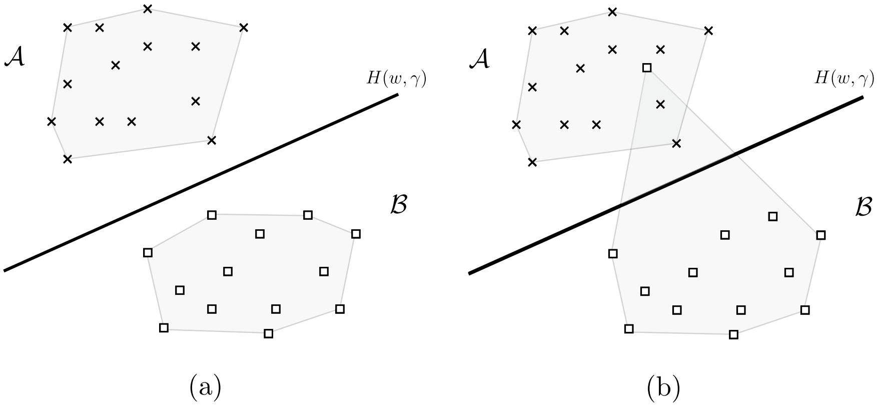

Fig. 1 Linear separation: **a** $\mathcal{A}$ and $\mathcal{B}$ are separable since $\text{conv}(\mathcal{A}) \cap \text{conv}(\mathcal{B}) = \emptyset$; **b** $\mathcal{A}$ and $\mathcal{B}$ are not separable since $\text{conv}(\mathcal{A}) \cap \text{conv}(\mathcal{B}) \neq \emptyset$

maximizing the margin (i.e. the distance between two parallel hyperplanes supporting the sets), representing a measure of the generalization capability, i.e. the ability of the classifier to correctly classify any new sample (see Fig. 2). In particular, from the mathematical point of view, the SVM provides a separating hyperplane $H(w, \gamma)$ by minimizing the following error function:

$$\min_{w,\gamma} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m} \max\{0, a_i^T w - \gamma + 1\} + C\sum_{l=1}^{k} \max\{0, -b_l^T w + \gamma + 1\}, \tag{1}$$

where the minimization of first term corresponds to the maximization of the margin, and the last two terms represent the misclassification errors in correspondence to the two point sets $\mathcal{A}$ and $\mathcal{B}$, respectively. The parameter C is a positive constant giving the tradeoff between these two objectives. We conclude this subsection, reminding that the above nonsmooth minimization problem can be easily rewritten as a smooth quadratic programming problem.

2.2 Spherical Separation

In the spherical separation the idea is to find a sphere

$$S(x_0, R) = \{x \in \mathbb{R}^n \mid \ \|x - x_0\|^2 = R^2\},$$

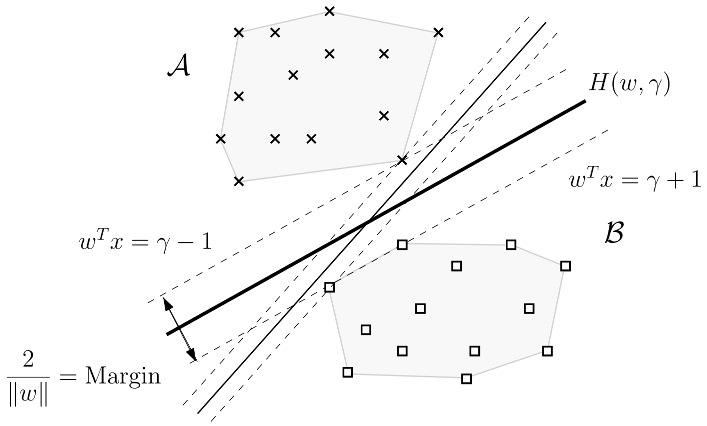

Fig. 2 Among all the separating hyperplanes, the SVM approach selects that one with the largest margin

with center $x_0 \in \mathbb{R}^n$ and radius R, enclosing all points of $\mathcal{A}$ and no points of $\mathcal{B}$.

2.2.1 Spherical Separation Without Margin

The set $\mathcal{A}$ is spherically separable from the set $\mathcal{B}$ if and only if there exists a sphere $S(x_0, R)$ such that

$$\|a_i - x_0\|^2 \leq R^2 \qquad i = 1, \ldots, m$$

and

$$\|b_l - x_0\|^2 \geq R^2 \qquad l = 1, \ldots, k.$$

We observe that, in this case, the role played by the two sets is not symmetric; in fact a necessary (but not sufficient) condition for the existence of a separation sphere is the following (see Fig. 3):

$$\text{conv}(\mathcal{A}) \cap \mathcal{B} = \emptyset.$$

Based on the above spherical separability definition, the classification error associated to any sphere $S(x_0, R)$ is

$$\sum_{i=1}^{m} \max\{0, \|a_i - x_0\|^2 - R^2\} + \sum_{l=1}^{k} \max\{0, R^2 - \|b_l - x_0\|^2\}.$$

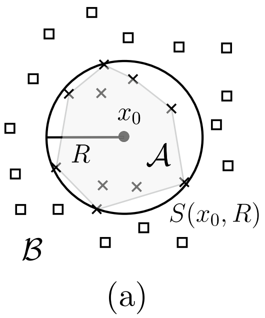

Fig. 3 Spherical separation of $\mathcal{A}$ from $\mathcal{B}$: **a** $\mathcal{A}$ is separable from $\mathcal{B}$ and then $\text{conv}(\mathcal{A}) \cap \mathcal{B} = \emptyset$; **b** $\mathcal{A}$ is not separable from $\mathcal{B}$ even if $\text{conv}(\mathcal{A}) \cap \mathcal{B} = \emptyset$; **c** $\mathcal{A}$ is not separable from $\mathcal{B}$ since $\text{conv}(\mathcal{A}) \cap \mathcal{B} \neq \emptyset$

To take into account the generalization capability, in [16] the authors have proposed to construct a minimal volume separation sphere by solving the following problem:

$$\min_{x_0, z} z + C \sum_{i=1}^{m} \max\{0, \|a_i - x_0\|^2 - z\} + C \sum_{l=1}^{k} \max\{0, z - \|b_l - x_0\|^2\}, \tag{2}$$

with $z \overset{\triangle}{=} R^2 \geq 0$ and $C > 0$ being the parameter tuning the tradeoff between the minimization of the volume and the minimization of the classification error.

Some works devoted to spherical separation are [2, 3, 8, 9, 16, 18, 44]. In particular, the approach presented in [16] assumes that the center x_0 of the sphere is fixed (for example, equal to the barycenter of $\mathcal{A}$): in such case it is easy to see that problem (2) reduces to a univariate, convex, nonsmooth optimization problem and it is rewritable as a structured linear program, whose dual can be solved in time $O(p \log p)$, where p is the cardinality of the biggest set between $\mathcal{A}$ and $\mathcal{B}$. In fact the optimal value of the variable z (the square of the radius) is computable by simply comparing the distances, preliminarily sorted, between the center x_0 and each point in the two sets. For further technical details on such approach we refer the reader directly to [16].

2.2.2 Spherical Separation with Margin

Now we consider a margin spherical separation, where we extend the SVM concept of margin to the spherical case with the aim at providing a better quality classifier. In particular the set $\mathcal{A}$ is strictly spherically separable from

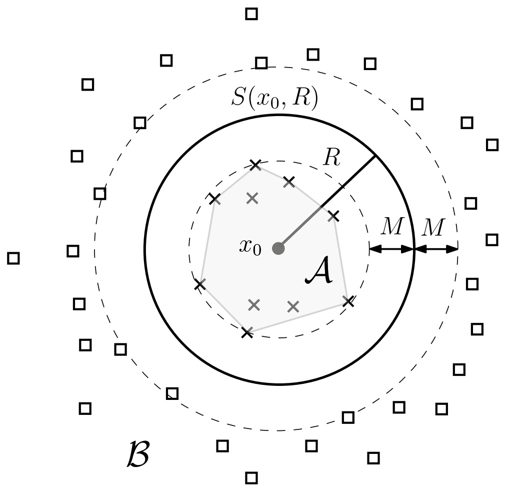

Fig. 4 Strict spherical separation of $\mathcal{A}$ from $\mathcal{B}$

the set $\mathcal{B}$ if there exists a sphere $S(x_0, R)$ such that

$$\|a_i - x_0\|^2 \le (R - M)^2, \qquad i = 1, \dots, m$$

and

$$\|b_l - x_0\|^2 \ge (R + M)^2, \qquad l = 1, \dots, k,$$

for some margin M, $0 < M \le R$ (see Fig. 4).

Based on the above definition, the classification error becomes

$$\sum_{i=1}^{m} \max\{0, \|a_i - x_0\|^2 - (R - M)^2\} + \sum_{l=1}^{k} \max\{0, (R + M)^2 - \|b_l - x_0\|^2\},$$

which, by setting $z \overset{\triangle}{=} R^2 + M^2$ and $q \overset{\triangle}{=} 2RM$, can be rewritten as:

$$\sum_{i=1}^{m} \max\{0, q - z + \|a_i - x_0\|^2\} + \sum_{l=1}^{k} \max\{0, q + z - \|b_l - x_0\|^2\}.$$

In [9] the authors have proposed to solve the following optimization problem:

$$\min_{x_0, 0 \le q \le z} C \left(\sum_{i=1}^{m} \max\{0, q - z + \|a_i - x_0\|^2\} + \sum_{l=1}^{k} \max\{0, q + z - \|b_l - x_0\|^2\} \right) - q,$$

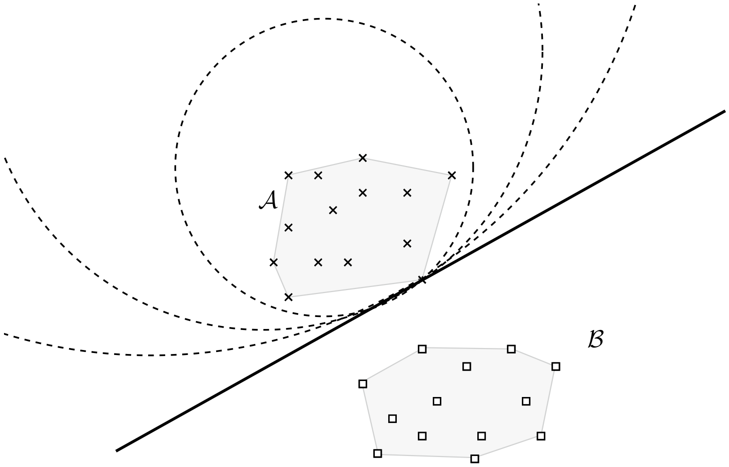

Fig. 5 A hyperplane can be interpreted as a sphere with an infinitely far center

where the objective of margin maximization is represented by the term $-q$, while the tradeoff between classification error and margin is accounted by the positive weighting parameter C.

In case the center x_0 of the sphere is given, the above problem reduces to the minimization of a nonsmooth and convex function [4, 37] in the two variables z and q. Such problem can be easily put in the form of a structured linear program, which is solvable by an extended version of the algorithm presented in [16] (see [9] for the details).

2.3 Comparing Linear and Spherical Separation in the Grossone Framework

From the mathematical point of view, both the linear and the spherical separations are characterized by the same number of variables to be determined: in fact a separation hyperplane is identified by the bias and the normal, while a sphere is obtained by computing the center and the radius. In this perspective, a hyperplane can be viewed as a particular sphere where the center is infinitely far (see Fig. 5).

Then a possible choice of the center x_0 is to take a point far from both the sets $\mathcal{A}$ and $\mathcal{B}$, i.e.

$$x_0 = x_0^{\mathcal{A}} + M\left(x_0^{\mathcal{A}} - x_0^{\mathcal{B}}\right), \tag{3}$$

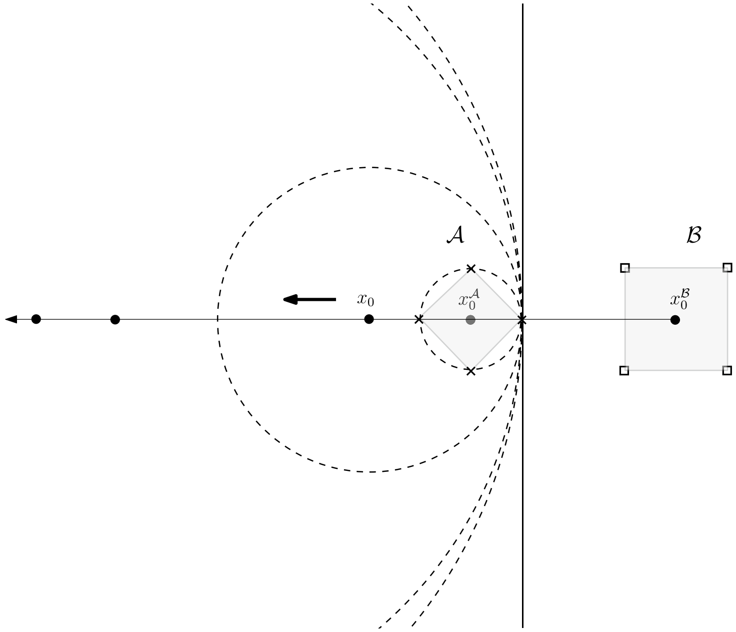

Fig. 6 Spherical separation with a far center

where

$$x_0^{\mathcal{A}} \triangleq \frac{1}{m}\sum_{i=1}^{m} a_i \quad \text{and} \quad x_0^{\mathcal{B}} \triangleq \frac{1}{k}\sum_{l=1}^{k} b_l$$

are the barycenters of $\mathcal{A}$ and $\mathcal{B}$, respectively, while M is a sufficiently large positive parameter, commonly named "big M" (see for example [32]).

Formula (3) corresponds to computing x_0 from $x_0^{\mathcal{A}}$ along the direction $x_0^{\mathcal{A}} - x_0^{\mathcal{B}}$ with stepsize equal to M (see Fig. 6).

Notice that, in general, the "big M" constant is not easy to be managed from the numerical point of view, since indeed it is not evident how to quantify the minimum threshold value such that M could be considered sufficiently big: as a consequence, in the practical cases, the necessity to test many trial values arises. A possible way to overcome this numerical difficulty is to obtain an infinitely far center by exploiting the grossone theory [53], setting M equal to ①, where the symbol ① denotes the new numeral *grossone*.

Differently from [16], where various values of M in formula (3) have been tested in order to obtain a good classification performance, a remarkable advantage in using the grossone resides in avoiding the necessity to repeat several tests with larger and larger values of M.

We conclude the subsection by highlighting that the new grossone-based computational methodology, which is not related to the nonstandard analysis [54], is applied in various fields, such as in optimization [31, 34–36, 41, 55], in numerical differentiation [52], in ordinary differential equations [43] and so on. To the best of our knowledge, it seems that the only machine learning paper involving the grossone idea is [7]. Finally some more theoretical works are in logics and philosophy [45, 46, 50], in probability [27] and in fractals analysis [25, 26].

3 Linear and Spherical Separability for Multiple Instance Learning

Multiple Instance Learning (MIL) [42] is a machine learning paradigm, consisting in classifying sets of samples: the samples are called instances and the sets are called bags. The main peculiarity of a MIL problem stays in the learning phase, where only the labels of the bags are known while the labels of the instances are unknown.

The first MIL paper [38] has appeared in 1997: in such work a drug design problem has been tackled, with the aim at discriminating between active and non-active molecules. A drug molecule is active (i.e. it has the desired drug effect) if one or more of its conformations binds to a particular target site (typically a larger protein molecule): the peculiarity of the problem is that it is not known a priori which conformation makes a molecule active, being available only the label of the overall molecule. In the MIL perspective, each molecule is a bag and the corresponding conformations are the instances.

We focus on binary MIL problems, aimed at discriminating between positive and negative bags, in the presence of only two classes of instances. We adopt the so-called standard MIL assumption (very common in the literature), stating that a bag is positive if and only if it contains at least a positive instance and it is negative otherwise.

Since the considerations reported in Sect. 2.3 for supervised classification can be extended to MIL, in the sequel we first remind the SVM type model for MIL introduced in [1] and, successively, we propose our modification of such model based on the spherical separation.

3.1 The SVM Type Model for MIL

The SVM type model proposed for MIL in [1] provides, in the instance space, a separating hyperplane $H(w, \gamma)$ by solving the following optimization problem:

$$\begin{cases} \displaystyle\min_{w,\gamma,y} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\sum_{j\in J_i^+} \max\{0, y_j(x_j^T w - \gamma) + 1\} \\ \displaystyle\qquad\qquad + C\sum_{l=1}^{k}\sum_{j\in J_l^-} \max\{0, -x_j^T w + \gamma + 1\} \\ \displaystyle\sum_{j\in J_i^+} \frac{y_j + 1}{2} \geq 1 \quad i = 1, \dots, m \\ y_j \in \{-1, +1\} \quad j \in J_i^+, \quad i = 1, \dots, m, \end{cases} \tag{4}$$

where m is the number of positive bags indexed by the sets $J_i^+, i = 1, \dots, m$, k is the number of negative bags indexed by the sets $J_l^-, l = 1, \dots, k, x_j$ is the jth instance belonging to a bag and y_j is the class label of the instance x_j. The m constraints involved in the above nonlinear mixed integer program impose that at least one instance of each positive bag is labelled positively by $y_j = +1$, i.e. the satisfaction of the standard MIL assumption. A separating MIL hyperplane is depicted in Fig. 7, where the two dashed polygons represent the positive bags and the three continuous polygons are the negative bags.

Notice that in case each bag is a singleton and $y_j = 1$ for any j, problem (4) reduces to the classical SVM problem (1).

3.2 A Grossone MIL Spherical Model

In this subsection, in order to embed the grossone framework into the MIL paradigm, we propose to modify problem (4) by substituting the hyperplane for a sphere. We obtain the following nonlinear mixed integer optimization problem:

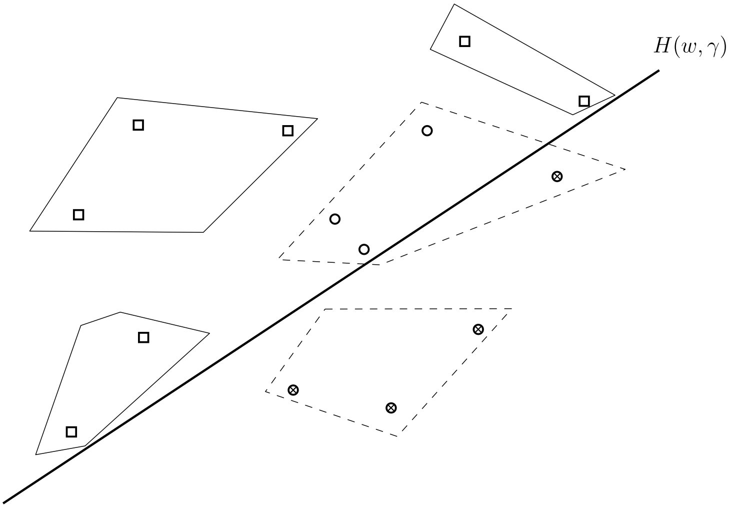

Fig. 7 MIL separating hyperplane: two positive bags (dashed polygons) and three negative bags (continuous polygons). The circles and the squares inside the bags represent the instances

$$\begin{cases} \min\limits_{x_0,R,y} R^2 + C\sum\limits_{i=1}^{m}\sum\limits_{j\in J_i^+}\max\{0,\, y_j(\|x_j-x_0\|^2 - R^2)\} \\ \qquad\qquad + C\sum\limits_{l=1}^{k}\sum\limits_{j\in J_l^-}\max\{0,\, R^2 - \|x_j-x_0\|^2\} \\ \qquad\qquad \sum\limits_{j\in J_i^+}\dfrac{y_j+1}{2} \geq 1 \quad i=1,\dots,m \\ \qquad\qquad y_j \in \{-1,+1\} \quad j\in J_i^+, \quad i=1,\dots,m. \end{cases} \tag{5}$$

According to the concept of spherical separation reported in Sect. 2.2.1, the above model takes into account the standard MIL assumption, which, in case of a separating sphere, imposes that a bag is positive if at least one of its instances is inside the sphere and it is negative otherwise (see Fig. 8, where the represented bags are the same as in Fig. 7).

A possible approach to solve heuristically problem (5) could be to use a BDC (Block Coordinate Descent) [57] type algorithm, consisting in the alternation between the computation of the vector y when the couple (x_0, R) is fixed and, vice-versa, the computation of the couple (x_0, R) when y is fixed. In particular, when y is fixed, x_0 could be set by adopting the following formula (analogous to formula (3), with M substituted by ①):

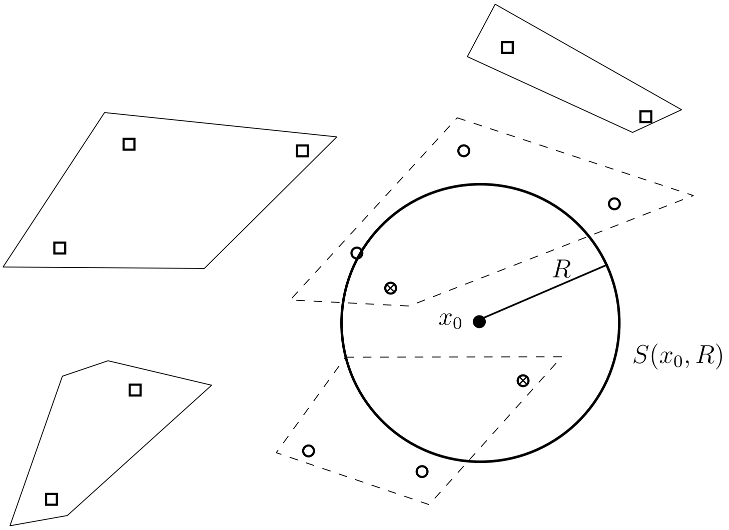

Fig. 8 MIL separating sphere: two positive bags (dashed polygons) and three negative bags (continuous polygons). The circles and the squares inside the bags represent the instances

$$x_0 = x_0^+ + ①(x_0^+ - x_0^-), \tag{6}$$

where x_0^+ and x_0^- are the barycenters of the currently positive and negative instances, respectively. Once y is fixed and x_0 is computed by formula (6), the corresponding optimal radius of the sphere is obtainable by using the ad hoc algorithm presented in [16].

4 Some Numerical Results

In [7] some numerical experiments have been performed to test the grossone idea in the supervised spherical separation without margin. In fact the center of the sphere has been chosen as follows:

$$x_0 = x_0^{\mathcal{A}} + ① \left(x_0^{\mathcal{A}} - x_0^{\mathcal{B}} \right),$$

i.e. by setting $M = ①$ in formula (3).

The code, named in [7] $\text{FC}_{①}$, has been implemented in Matlab and it has been tested on thirteen data sets drawn from the literature and listed in Table 1.

Table 1 Data sets

Data set	Dimension	Points
Cancer	9	699
Diagnostic	30	569
Heart	13	297
Pima	9	769
Ionosphere	34	351
Sonar	60	208
Mushrooms	22	8124
Prognosis	32	110
Tic Tac Toe	9	958
Votes	16	435
Galaxy	14	4192
g50c	50	550
g10n	10	550

The first ten test problems have been taken from the UCI Machine Learning Repository [48], a collection of databases, domain theories, and data generators that are used by the machine learning community. Galaxy is the data set used in galaxy discrimination with neural networks [51], while an accurate description of g50c and g10n is reported in [30].

In order to manage the grossone arithmetic operations, the authors have used the Matlab Environment of the new Simulink-based solution of the Infinity Computer [39], where an arithmetic C++ library is integrated within a Matlab environment. In particular, given the two gross-numbers x and y, from such library the following C++ subroutines have been used:

- `TestGrossMatrix(x,y,'-')`, returning the difference between x and y;
- `TestGrossMatrix(x,y,'+')`, returning the sum of x and y;
- `TestGrossMatrix(x,y,'*')`, returning the product of x and y;
- `GROSS_cmp(x,y)`, returning 1 if $x > y$, -1 if $x < y$ and 0 if $x = y$.

Using the Matlab notation, any vector g of n gross-number elements (that in the sequel, for the sake of simplicity, we call gross-vector) has been expressed as a couple `(G,fg)`, with

$$\mathtt{G} = [\mathtt{g1}; \mathtt{g2}; \ldots; \mathtt{gn}] \quad \text{and} \quad \mathtt{fg} = [\mathtt{fg1}\ \mathtt{fg2} \ldots \mathtt{fgn}],$$

where `gj`, $j = 1, \ldots, n$, is an array of appropriate dimension representing a gross-number. For each row of `gj`, the first element contains a gross-digit, while the second one contains the corresponding gross-power. The scalar `fgj`, $j = 1, \ldots, n$, is necessary to provide the position in `G` of the last component of `gj`.

To manage the gross-vectors, in [7] the following new Matlab subroutines have been implemented:

- `realToGrossone(r)`, returning a grossone representation `(G,fg)` of a real vector `r`;
- `extract(G,fg,i)`, returning the `i`th gross-number in the gross-vector `(G, fg)`;
- `normGrossone(G,fg)`, computing the squared Euclidean norm of the gross-vector `(G,fg)`;
- `scalProdG(G1,fg1,G2,fg2)`, computing the scalar product between the two gross-vectors `(G1,fg1)` and `(G2,fg2)`;
- `BubbleSortGrossone(G,fg,sign)`, sorting the gross-vector `(G, fg)` in the ascending order if `sign` $= 1$ and in the descending order if `sign` $= -1$.

For each data set, in order to compute the best value of the parameter C, a bilevel cross-validation strategy [6] has been adopted, by varying C in the grid $\{10^{-1}, 10^{0}, 10^{1}, 10^{2}\}$: such choice of the grid has been suggested by the necessity to obtain a nonzero optimal value of z, which in turn provides the optimal value of the radius R, as shown in [16].

In Table 2 we report the results, provided by Algorithm $\text{FC}_{①}$ and published in [7], expressed in terms of average testing correctness. Such results have been compared by the authors with those ones relative to the two following fixed-center classical variants, obtained by setting

$$x_0 = x_0^{\mathcal{A}} \qquad (\text{Algorithm } \text{FC}_{\mathcal{A}})$$

and

$$x_0 = x_0^{\mathcal{A}} + x_0^{\mathcal{B}} \qquad (\text{Algorithm } \text{FC}_{\mathcal{AB}}),$$

respectively, and with the results obtained by a variant of the standard linear SVM (Algorithm SVM_0), where, in order to have a fair comparison, the margin term has been dropped by setting, in the `fitcsvm` Matlab subroutine, the penalty parameter `BoxConstraint` equal to 10^6. We recall in fact the spherical approach implemented in [7] does not involve any margin concept. In Table 2, for each data set, the best result is underlined.

In comparison with $\text{FC}_{\mathcal{A}}$ and $\text{FC}_{\mathcal{AB}}$, the choice of the infinitely far center appears to be the best one: in fact Algorithm $\text{FC}_{①}$ outperforms the other two

Table 2 Numerical results

Data set	$\mathrm{FC}_{\mathcal{A}}$	$\mathrm{FC}_{\mathcal{A}\mathcal{B}}$	$\mathrm{FC}_{\textcircled{1}}$	SVM_0
Cancer	97.00	95.71	97.57	71.86
Diagnostic	83.86	53.33	89.65	92.11
Heart	74.33	55.00	87.33	68.67
Pima	69.35	66.23	61.43	61.82
Ionosphere	51.14	40.75	78.86	69.43
Sonar	59.05	52.86	65.71	75.24
Mushrooms	76.44	64.50	78.19	49.59
Prognosis	56.00	45.00	68.00	53.00
Tic Tac Toe	71.79	70.42	57.79	50.11
Votes	82.79	53.35	86.74	76.51
Galaxy	80.24	51.36	89.19	54.32
g50c	67.62	50.26	90.58	86.56
g10n	53.58	45.02	77.66	90.24

approaches on all the data sets except Pima and Tic Tac Toe, where the best performance is got by fixing x_0 as the barycenter of $\mathcal{A}$. We note also that choosing x_0 as the barycenter of all the points is not a good strategy, since the corresponding results are very poor on all the test problems, but Cancer and Tic Tac Toe, where the testing correctnesses appear comparable.

Also with respect to SVM_0, Algorithm $\mathrm{FC}_{\textcircled{1}}$ is characterized by a good performance, except on Diagnostic, Sonar and g10n, while on Pima both the approaches behave almost the same. These results were expected because, even if taking the radius infinitely far makes the spherical separability tend to the linear separability, the two approaches differ substantially. We recall in fact that, if two sets are linearly separable, they are also spherical separable (even taking a very large radius), but the vice-versa is not true.

5 Conclusions

In this work we have examined the main differences between linear and spherical separation in the light of the grossone theory. In particular, we have recalled the main observations reported in [7] for supervised classification, extending them to the cases of the supervised spherical separation with margin and of the Multiple Instance Learning.

We have focused on the possibility to construct binary spherical classifiers characterized by an infinitely far center. As shown by the preliminary numerical results reported in [7], adopting the grossone theory allows to obtain a good performance in terms of average testing correctness, managing very easily the numerical computations, which do not require any tuning of the "big M" parameter.

Future research could consist in extending such approach to the kernel trick, which is well suitable in the fixed-center spherical separation, as shown in [16], and to practically implement the grossone idea for solving MIL problems.

References

1. Andrews, S., Tsochantaridis, I., Hofmann, T.: Support vector machines for multiple-instance learning. In: Becker, S., Thrun, S., Obermayer, K. (eds.) Advances in Neural Information Processing Systems, pp. 561–568. MIT Press, Cambridge (2003)
2. Astorino, A., Bomze, I., Brito, P. Gaudioso, M.: Two spherical separation procedures via non-smooth convex optimization. In: De Simone, V., Di Serafino, D., Toraldo, G. (eds.) Recent Advances in Nonlinear Optimization and Equilibrium Problems: A Tribute to Marco D'Apuzzo, Quaderni di Matematica, Dipartimento di Matematica della Seconda Universitá di Napoli, vol. 27, pp. 1–16. Aracne (2012)
3. Astorino, A., Bomze, I., Fuduli, A., Gaudioso, M.: Robust spherical separation. Optimization **66**(6), 925–938 (2017)
4. Astorino, A., Frangioni, A., Fuduli, A., Gorgone, E.: A nonmonotone proximal bundle method with (potentially) continuous step decisions. SIAM J. Optim. **23**(3), 1784–1809 (2013)
5. Astorino, A., Fuduli, A.: Nonsmooth optimization techniques for semisupervised classification. IEEE Trans. Pattern Anal. Mach. Intell. **29**(12), 2135–2142 (2007)
6. Astorino, A., Fuduli, A.: The proximal trajectory algorithm in SVM cross validation. IEEE Trans. Neural Netw. Learn. Syst. **27**(5), 966–977 (2016)
7. Astorino, A., Fuduli, A.: Spherical separation with infinitely far center. Soft Comput. **24**(23), 17751–17759 (2020)
8. Astorino, A., Fuduli, A., Gaudioso, M.: DC models for spherical separation. J. Global Optim. **48**(4), 657–669 (2010)
9. Astorino, A., Fuduli, A., Gaudioso, M.: Margin maximization in spherical separation. Comput. Optim. Appl. **53**(2), 301–322 (2012)
10. Astorino, A., Fuduli, A., Gaudioso, M.: Nonlinear programming for classification problems in machine learning. In: AIP Conference Proceedings, vol. 1776 (2016)
11. Astorino, A., Fuduli, A., Gaudioso, M.: A Lagrangian relaxation approach for binary multiple instance classification. IEEE Trans. Neural Netw. Learn. Syst. **30**(9), 2662–2671 (2019)
12. Astorino, A., Fuduli, A., Gaudioso, M., Vocaturo, E.: Multiple instance learning algorithm for medical image classification. In: CEUR Workshop Proceedings, vol. 2400 (2019)

13. Astorino, A., Fuduli, A., Giallombardo, G., Miglionico, G.: SVM-based multiple instance classification via DC optimization. Algorithms **12**(12) (2019)
14. Astorino, A., Fuduli, A., Gorgone, E.: Non-smoothness in classification problems. Optim. Methods Softw. **23**(5), 675–688 (2008)
15. Astorino, A., Fuduli, A., Veltri, P., Vocaturo, E.: Melanoma detection by means of multiple instance learning. Interdiscip. Sci.: Comput. Life Sci. **12**(1), 24–31 (2020)
16. Astorino, A., Gaudioso, M.: A fixed-center spherical separation algorithm with kernel transformations for classification problems. Comput. Manag. Sci. **6**(3), 357–372 (2009)
17. Astorino, A., Gaudioso, M., Fuduli, A., Vocaturo, E.: A multiple instance learning algorithm for color images classification. In: ACM International Conference Proceeding Series, pp. 262–266 (2018). www.scopus.com. Cited By :15
18. Astorino, A., Gaudioso, M., Khalaf, W.: Edge detection by spherical separation. Comput. Manag. Sci. **11**(4), 517–530 (2014)
19. Astorino, A., Gaudioso, M., Seeger, A.: Conic separation of finite sets. I. The homogeneous case. J. Convex Anal. **21** (2014)
20. Astorino, A., Gaudioso, M., Seeger, A.: Conic separation of finite sets II. The nonhomogeneous case. J. Convex Anal. **21**(3), 819–831 (2014)
21. Avolio, M., Fuduli, A.: A semiproximal support vector machine approach for binary multiple instance learning. IEEE Trans. Neural Netw. Learn. Syst. **32**(8), 3566–3577 (2021)
22. Bagirov, A., Karmitsa, N., Taheri, S.: Partitional Clustering via Nonsmooth Optimization. Springer, Berlin (2020)
23. Bennett, K.P., Mangasarian, O.L.: Robust linear programming discrimination of two linearly inseparable sets. Optim. Methods Softw. **1**, 23–34 (1992)
24. Bergeron, C., Moore, G., Zaretzki, J., Breneman, C., Bennett, K.: Fast bundle algorithm for multiple instance learning. IEEE Trans. Pattern Anal. Mach. Intell. **34**(6), 1068–1079 (2012)
25. Caldarola, F.: The exact measures of the Sierpinski d-dimensional tetrahedron in connection with a diophantine nonlinear system. Commun. Nonlinear Sci. Numer. Simul. **63**, 228–238 (2018)
26. Caldarola, F.: The Sierpinski curve viewed by numerical computations with infinities and infinitesimals. Appl. Math. Comput. **318**, 321–328 (2018)
27. Calude, C.S., Dumitrescu, M.: Infinitesimal probabilities based on grossone. SN Comput. Sci. **1**, article number: 36 (2020)
28. Celebi, M.E. (ed.): Partitional Clustering Algorithms. Springer International Publishing, Berlin (2015)
29. Chapelle, O., Schölkopf, B., Zien, A. (eds.): Semi-supervised learning. MIT Press, Cambridge (2006)
30. Chapelle, O., Zien, A.: Semi-supervised classification by low density separation. In: Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics, pp. 57–64 (2005)
31. Cococcioni, M., Cudazzo, A., Pappalardo, M., Sergeyev, Y.D.: Solving the lexicographic multi-objective mixed-integer linear programming problem using branch-and-bound and grossone methodology. Commun. Nonlinear Sci. Numer. Simul. **84**, 105177 (2020)
32. Cococcioni, M., Fiaschi, L.: The Big-M method with the numerical infinite M. Optim. Lett, **15**, 2455–2468 (2021)

33. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press, Cambridge (2000)
34. De Cosmis, S., De Leone, R.: The use of grossone in mathematical programming and operations research. Appl. Math. Comput. **218**(16), 8029–8038 (2012)
35. De Leone, R.: Nonlinear programming and grossone: quadratic programming and the role of constraint qualifications. Appl. Math. Comput. **318**, 290–297 (2018)
36. De Leone, R., Fasano, G., Sergeyev, Y.D.: Planar methods and grossone for the conjugate gradient breakdown in nonlinear programming. Comput. Optim. Appl. **71**(1), 73–93 (2018)
37. Demyanov, A., Fuduli, A., Miglionico, G.: A bundle modification strategy for convex minimization. Eur. J. Oper. Res. **180**, 38–47 (2007)
38. Dietterich, T.G., Lathrop, R.H., Lozano-Pérez, T.: Solving the multiple instance problem with axis-parallel rectangles. Artif. Intell. **89**(1–2), 31–71 (1997)
39. Falcone, A., Garro, A., Mukhametzhanov, M.S., Sergeyev, Y.D.: A simulink-based infinity computer simulator and some applications. In: Sergeyev, Y.D., Kvasov, D.E. (eds.) Numerical Computations: Theory and Algorithms, pp. 362–369. Springer International Publishing, Cham (2020)
40. Gaudioso, M., Giallombardo, G., Miglionico, G., Vocaturo, E.: Classification in the multiple instance learning framework via spherical separation. Soft Comput. **24**, 5071–5077 (2020). https://doi.org/10.1007/s00500-019-04255-1
41. Gaudioso, M., Giallombardo, G., Mukhametzhanov, M.S.: Numerical infinitesimals in a variable metric method for convex nonsmooth optimization. Appl. Math. Comput. **318**, 312–320 (2018)
42. Herrera, F., Ventura, S., Bello, R., Cornelis, C., Zafra, A., Sánchez-Tarragó, D., Vluymans, S.: Multiple Instance Learning: Foundations and Algorithms. Springer International Publishing, Berlin (2016)
43. Iavernaro, F., Mazzia, F.: Solving ordinary differential equations by generalized Adams methods: properties and implementation techniques. Appl. Numer. Math. **28**(2–4), 107–126 (1998)
44. Le Thi, H.A., Minh, L.H., Pham Dinh, T., Ngai, V.H.: Binary classification via spherical separator by DC programming and DCA. J. Global Optim. **56**, 1393–1407 (2013)
45. Lolli, G.: Infinitesimals and infinites in the history of mathematics: a brief survey. App. Math. Comput. **218**(16), 7979–7988 (2012)
46. Lolli, G.: Metamathematical investigations on the theory of grossone. Appl. Math. Comput. **255**, 3–14 (2015)
47. Mangasarian, O.L.: Linear and nonlinear separation of patterns by linear programming. Oper. Res. **13**(3), 444–452 (1965)
48. Murphy, P.M., Aha, D.W.: UCI repository of machine learning databases (1992). www.ics.uci.edu/~mlearn/MLRepository.html
49. Plastria, F., Carrizosa, E., Gordillo, J.: Multi-instance classification through spherical separation and VNS. Comput. & Oper. Res. **52**, 326–333 (2014)
50. Rizza, D.: A study of mathematical determination through Bertrand's Paradox. Philos. Math. **26**(3), 375–395 (2018)
51. Odewahn, S., Stockwell, E., Pennington, R., Humphreys, R., Zumach, W.: Automated star/galaxy discrimination with neural networks. Astron. J. **103**(1), 318–331 (1992)
52. Sergeyev, Y.D.: Higher order numerical differentiation on the infinity computer. Optim. Lett. **5**(4), 575–585 (2011)

53. Sergeyev, Y.D.: Numerical infinities and infinitesimals: methodology, applications, and repercussions on two Hilbert problems. EMS Surv. Math. Sci. **4**(2), 219–320 (2017)
54. Sergeyev, Y.D.: Independence of the grossone-based infinity methodology from non-standard analysis and comments upon logical fallacies in some texts asserting the opposite. Found. Sci. **24**(1), 153–170 (2019)
55. Sergeyev, Y.D., Kvasov, D.E., Mukhametzhanov, M.S.: On strong homogeneity of a class of global optimization algorithms working with infinite and infinitesimal scales. Comm. Nonlinear Sci. Num. Sim. **59**, 319–330 (2018)
56. Tax, D.M.J., Duin, R.P.W.: Data domain description using support vectors. In: ESANN'1999 proceedings Bruges, pp. 251–256. Belgium (1999)
57. Tseng, P.: Convergence of a block coordinate descent method for nondifferentiable minimization. J. Optim. Theory App. **109**(3), 475–494 (2001)
58. Vapnik, V.: The Nature of the Statistical Learning Theory. Springer, New York (1995)