International Journal of Applied Mathematical Research, *2 (4) (2013) 495-504*
©Science Publishing Corporation
www.sciencepubco.com/index.php/IJAMR

A linear programming solution to data description and novelty classification

Roberto Ragona

ENEA, Dept. of Advanced Technologies for Energy and Industry, Via Anguillarese, 301 - 00123 Rome (Italy)
**Corresponding author E-mail: roberto.ragona@enea.it*

Abstract

Many real-world problems require the detection of abnormal instances of a physical process, and methods inspired by the Support Vector Machines have been developed that model reference or normal data well. These methods serve as a fundamental step to enable the classification of new data as normal or abnormal. They imply the solution of a quadratic programming problem, which can present difficulties in finding solutions with standard methods and program solvers when the number of points becomes large. In this paper, we present an approach that was developed in a different context and that leads to a linear programming problem to attain the computational advantages of a linear environment.

Keywords: *Data description, novelty classification, one-class classifier, outlier detection, supports vector data description (SVDD).*

1 Introduction

The problem of one-class classification has received increasing attention in recent years, and several solutions have been proposed. The one-class classification describes the process of learning the normality of a system from a unique set of classified and labelled examples (the reference or target class A); once the optimised (in some sense) model has been built, new incoming data (novelties) are classified as normal or abnormal according to some defined score or criterion. Many approaches are available for novelty detection, such as the Gaussian mixture model and the Parzen density estimation model. These approaches were developed by first estimating the probability distribution of normal data patterns and thereafter distinguishing a new data pattern based on the distribution level. Other solutions propose distribution-free or domain description approaches, in which attempts to learn just the boundaries of the target set are made to try to exclude superfluous space. The availability of efficient solutions has stimulated interesting applications, such as in the field of medical diagnosis or statistical process control, where control charts are one of the most applied tools for quality control. The design of the control limits in many traditional charts (for example referring to T^2-Hotelling distribution) is based on the Gaussian assumption, which in many real-life applications is questionable.

These findings have generated an interest in distribution-free methods based on different principles. Among these approaches, we remark the solutions presented in foundational papers like [1] and [2], where the classification process presents deep analogies with the concepts of support vector machines (SVM) found in [3] and leads to quadratic programming problems (QP).

In a previous paper [4] we presented a solution method for two-class optimal classifiers that proceeds by means of linear programming (LP) techniques, which presents an alternative to the quadratic programming techniques of traditional SVM; in this paper we extend the LP techniques to a one-class classifier definition and show that linear techniques can address fundamental classification problems.

Other methods based on LP techniques have been presented, from differing points of view and with different results, e.g. [13] [14], which show that LP is a viable approach to one-class classification.

In section 2, we will present the basic theory pertaining to the one-dimensional case; in section 3, we will generalise this theory to higher dimensions; a refinement of the LP method is outlined in sections 4 and 5; in section 6, methods for outlier treatment will be discussed, and comparisons between the proposed LP approach and the QP approach as outlined in [1] will finally be presented in section 7.

2 The basic theory for the one-dimensional space

We have a one-dimensional data set $\{x_i\}$, i=1, 2, ..., n defined on the x-axis of R^2 (i.e. the class **A** of interest), and we want to obtain a closed region that encloses the data, which is the segment between the leftmost and rightmost point on the x-axis in this simple case (see fig.1).

The key idea is to consider an auxiliary function (*support function*), s(x), defined on x that can favour the selection of the boundary points; for example, the function $s(x) = x^2$ comprises the ordinates $y_1 = s(x_1) = x_1^2$ and $y_n = s\ (x_n) = x_n^2$, which correspond to the boundary points x_1 and x_n, and distinguishes them from the ordinates of the remaining points.

Moreover, let us consider a generic straight line (*decision function*), *D*(x):

$$D(x) = a_1\ x + b,$$

where b is the intercept of *D*(x) on the y-axis.

To obtain the optimal line $D^*(x) = r^* = a_1^*x + b^*$ that crosses s(x) as low as possible at the two boundary points $P_1 = (x_1, x_1^2)$ and $P_n = (x_n, x_n^2)$ (see fig. 1), we must solve the LP problem:

$$\left.\begin{array}{ll} \text{minimise b, subject to} & \\ D(x_i) \geq s\ (x_i), & \text{that is,} \\ a_1\ x_i + b \geq x_i^2, & (i=1, 2, \ldots n) \end{array}\right\} \qquad (1)$$

The LP procedure (1) moves a generic line r towards the optimal line r* (see again fig. 1). The auxiliary function, $s(x) = x^2$, acts as support to the optimal line r*: only points belonging to the closed interval $[x_1, x_n]$ are compliant with the constraints (1). Moreover, the following is true at boundaries:

$$r^*(x_1) = a_1^*x_1 + b^* = s(x_1)\ = x_1^2;$$
$$r^*(x_n) = a_1^*x_n + b^* = s(x_n)\ = x_n^2;$$

x_1 and x_n assume in our formulation the name of *support vectors*, in the sense that they define the sample limits of the reference class and are sufficient to describe its extent.

New incoming data x_c can now be classified according to the following rules:

$$\text{if } D^*(x_c) = a_1^*\ x_c + b^* < \ s(x_c),\ x_c \notin \text{ class } \mathbf{A} \text{ (e.g., the point } x_{c1} \text{ of fig. 1);}$$
$$\text{if } D^*(x_c) = a_1^*\ x_c + b^* \geq s(x_c),\ x_c \in \text{ class } \mathbf{A} \text{ (e.g., the point } x_{c2} \text{ of fig. 1).} \qquad (2)$$

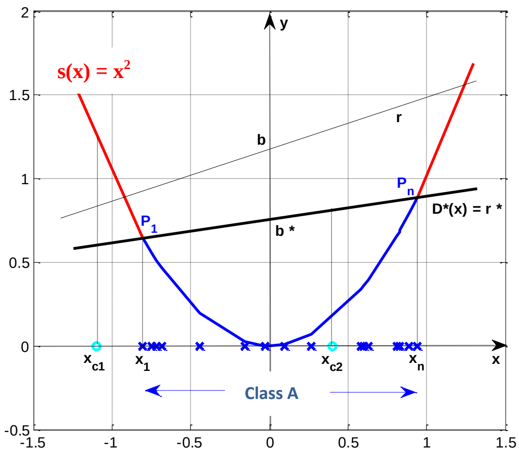


Fig. 1: The basic idea behind the LP solution

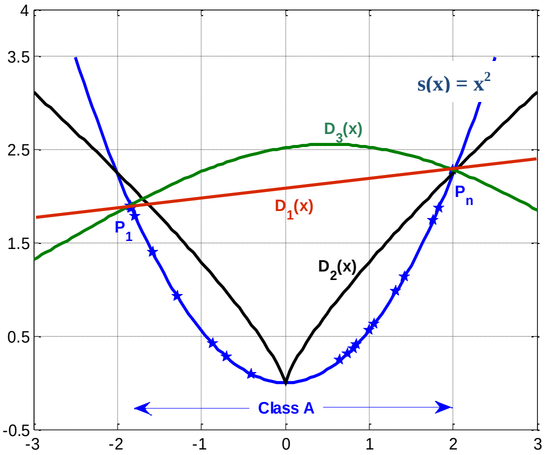


Fig. 2: Possible decision functions *D*(x)

The appropriate interaction between the decision function, *D*(x), and the support function, s(x), defines the ability to trace a closed region around the data sample and to classify new data.

A fundamental requirement of the method is that the function s(x) is convex and that its point of minimum is strictly internal to the data distribution, otherwise unbounded solutions in the LP problem (1) can be attained; this last requirement is easy to satisfy, e.g. by a preliminary mean detrending, which sets the sample data near the minimum of s(x). Other requirements involve the choice of s(x): the properties of continuity and regularity are assumed, along with a divergence to $+\infty$ with x tending to $\pm\infty$, to consider all possible data range.

Conversely, the decision function, *D*(x), can assume multiple forms (see fig. 2).

In fig. 2, $D_1(x)$ is a straight line, whereas $D_2(x) = a_1 \mid x \mid^{0.8} + b$, and $D_3(x)$ is a linear combination of two appropriate Radial Basis Functions (RBF). $D_1(x)$ and $D_3(x)$ properly pass through the points P_1 and P_n, which are the leftmost and the rightmost points of the sample.

The following relationship between derivative values must be satisfied on the outside region of the class **A** to render the classification rules (2) possible:

$$ds(x)/dx > dD^*(x)/dx.$$

3 Generalizations

Our strategy in R^m, where $m > 2$, is to refer the class **A** at the origin O and create a boundary curve as the intersection between a *hyperplane* and the support function in the input space, R^m. Thus, the generalisation to a linear decision function on the points of the space R^m is straightforward: $D(\mathbf{x})$ is now expressed as a sum of the type

$$\textstyle\sum_1^m a_i x_i + b = \boldsymbol{a}^T \bullet \mathbf{x} + b,$$

where $\mathbf{x} = [x_1, x_2, ... x_m]^T \epsilon R^m$, and problem (1) can be consequently reformulated.

The operator $(\bullet)^T$ denotes a matrix/vector transposition.

To generalise to *non-linear* decision functions on points of the space R^m, we resort to a function $D(\mathbf{x})$, which is defined via N properly chosen non-linear functions, $\varphi_i(\mathbf{x})$:

$$D(\mathbf{x}) = \textstyle\sum_1^N a_i\ \varphi_i(\mathbf{x}) + b = [a_1\, a_2\ .....\ a_N] \bullet \boldsymbol{\varphi}(\mathbf{x}) + b = \boldsymbol{a}^T \bullet \boldsymbol{\varphi}(\mathbf{x}) + b \qquad (3)$$

$$\mathbf{x}\ \epsilon\ R^m,\ \varphi_i(\mathbf{x}) \mathbf{:}\ R^m \rightarrow R,\ \boldsymbol{\varphi}(\mathbf{x}) = [\varphi_1(\mathbf{x})\ \varphi_2(\mathbf{x})\ ....\ \varphi_N(\mathbf{x})]^T\!:\ R^m \rightarrow R^N.$$

Apart from the necessary adaptations, the basic theory remains unchanged: we must find the optimal $D^*(\mathbf{x})$ by following the LP formulation introduced in (1):

$$\left.\begin{array}{l} \text{minimise b, subject to} \\ D(\mathbf{x}_j) = D(x_{j1,}\, x_{j2}\ldots\, x_{jm}) = \boldsymbol{a}^T \bullet \boldsymbol{\varphi}(x_{j1,}\, x_{j2},\, \ldots,\, x_{jm}) + b\ \geq\ s(x_{j1,}\, x_{j2},\, \ldots,\, x_{jm}), \\ j = 1, 2\ldots\, n \end{array}\right\} \qquad (4)$$

In the above definitions, m represents the dimensionality of the original space (the input space), N indicates the dimension of the feature space (generally augmented to facilitate the classification task [6]), and n the sample size.

A versatile choice of $s(\mathbf{x})$ in R^m takes the following form (elliptic paraboloid, [5]):

$$s(x_{1,}\, x_2,\, \ldots,\, x_m) = c_1x_1^{\,2} + c_2x_2^{\,2} + \qquad + c_mx_m^{\,2} \qquad (5)$$

$$c_i \in R^+,\ \forall\, i,\ i = 1, 2, ..., m$$

which satisfies the conditions of regularity, continuity and divergence to $+\infty$ when x_i tends to $\pm\infty$.

If we let c_i = constant $\in R^+$, $\forall$ i, we obtain a paraboloid of revolution. This paraboloid pertains to a parabola that revolves about its axis in R^3, in which the cross sections perpendicular to this axis are circles.

Equation (5) offers the advantage that the choice of each coefficient, c_i, can be usefully suggested by the variability along the i-th coordinate, e.g. in the space of principal components.

The new data $\mathbf{x}_c$ are classified as in (2), with the following obvious dimensional generalisation:

$$\text{if } D^*(\mathbf{x}_c) = \boldsymbol{a}^{*T} \bullet \boldsymbol{\varphi}(\mathbf{x}_c) + b^*\ <\ s(\mathbf{x}_c),\ \ \mathbf{x}_c \notin \text{ class } \mathbf{A};$$

$$\text{if } D^*(\mathbf{x}_c) = \boldsymbol{a}^{*T} \bullet \boldsymbol{\varphi}(\mathbf{x}_c) + b^*\ \geq\ s(\mathbf{x}_c),\ \ \mathbf{x}_c \in \text{ class } \mathbf{A}. \qquad (6)$$

These simple geometric concepts allow for a variety of nonlinear classifiers in the input space because of the available freedom to utilise different types of nonlinear functions. As a result, the specialisation of (3) leads to several possible forms of $D(\mathbf{x})$ in the higher dimensional case, e.g.:

a) $D(x_{1,}\, x_2\ldots\, x_m) = [a_1\ a_2\ .....\ a_m] \bullet \mathbf{[}x_1\ x_2\ .....\ x_m]^T + b$ (linear, $\varphi_i(\mathbf{x}) = x_i$, N = m)

b) $D(x_{1,}\, x_2\ldots\, x_m) = [a_1\ a_2\ ......\ a_m] \bullet \mathbf{[}|x_1|^h\ |x_2|^h\ .....\ |x_m|^h]^T + b$ (powers, $\varphi_i(\mathbf{x}) = |x_i|^h$, N = m)

c) $D(\mathbf{x}) = a_1\, K(\mathbf{x}_1, \mathbf{x}) + a_2\, K(\mathbf{x}_2, \mathbf{x}) + ... + a_n\, K(\mathbf{x}_n, \mathbf{x}) + b,$ (combination of dot products)

where h is a real positive exponent, and $K(\mathbf{x}_i, \mathbf{x})$ is any dot product of type

$$K(\mathbf{x}_i, \mathbf{x}) = \boldsymbol{\varphi}(\mathbf{x}_i)^T \bullet \boldsymbol{\varphi}(\mathbf{x}) = [\varphi_1(\mathbf{x}_i)\ \varphi_2(\mathbf{x}_i)\ ....\ \varphi_N(\mathbf{x}_i)] \bullet [\varphi_1(\mathbf{x})\ \varphi_2(\mathbf{x})\ ....\ \varphi_N(\mathbf{x})]^T$$

with possible infinite terms in $\boldsymbol{\varphi}(\bullet)$ (when $N \rightarrow \infty$), provided the dot product is finite.

In particular, this dot product can assume the following expression, which implies infinite terms in $\boldsymbol{\varphi}(\bullet)$ and a finite result associated with a closed form [7]:

$$K(\mathbf{x}_i, \mathbf{x}) = \exp(-\gamma\, \|\mathbf{x}_i - \mathbf{x}\|^2) \qquad \text{(RBF kernel)} \qquad (7)$$

The above form c) originates from a form of $D(\mathbf{x})$ of type (3) in the input space and can be justified with arguments similar to that developed in [4, §3]. It can be viewed as a tight approximation of the non-linear form (3) of $D(\mathbf{x})$ when operating implicitly over the feature space.

In fact, $D(\mathbf{x})$ realizes a mapping from the input space, R^m, to the real space, R, through the feature space, R^N, in form c), i.e. $D(\mathbf{x})$: $R^m \rightarrow R^N \rightarrow R$.

Interestingly, $K(\bullet_, \bullet)$ is not required to be a Mercer's kernel [6] in the LP context: it can consist of a generic dot product [4, §3].

4 A better formulation of the LP problem with a lasso-type term

The LP problem as formulated in (4) is generally not restrictive enough to assure a unique classifier, except for the linear case in R^2, as suggested by simple geometrical considerations.

In fact, the LP problem may present multiple solutions in practical applications of (4). The introduction of a regularisation term in (4) is appropriate to overcome this limitation, as done in statistical regression where this procedure is assumed to provide advantages, such as well-defined numerical solutions or avoiding over fitting. Different formulations are used in regression, all of which propose optimization criteria that controls the growth of coefficients, i. e. introducing in optimization regularisation terms like the following:

i) $\lambda \Sigma a_i^2$ (Ridge regression [8]);

ii) $\lambda \Sigma |a_i|$ (Lasso regression [9]).

Criterion ii) suits our context very well if we impose the additional constraints of positivity to the coefficients a_i; in this event criterion ii) simply reduces to the following:

$$\lambda \Sigma a_i$$

which ensures that problem (4) remains linear.

Therefore, our availability to accept suboptimal conditions is rewarded by addressed solutions in real terms.

In our experience, we have always found unique and well defined solutions to the LP problem (4) amended by a lasso-type term.

Therefore, problem (4) can be reformulated as follows:

$$\left.\begin{array}{l} \text{minimize } (b + \lambda \Sigma a_i), \text{ subject to} \\ D(x_{k1,} x_{k2} \ldots x_{km}) \geq s(x_{k1,} x_{k2}, \ldots, x_{km}), \\ a_i \geq 0, b \geq 0_, \quad k = 1, 2, \ldots., n; \; i = 1, 2, \ldots. \end{array}\right\} \tag{8}$$

The solution to (8) is indexed by the free parameter λ; it controls the size of the coefficients and the amount of regularisation, and tuned values based on experimental trials are to be selected.

A known effect of the lasso regularisation in statistical regression is that it may estimate some coefficients to be exactly zero, more often than other methods (coefficient shrinkage). This effect also seems to operate in our method: many optimal coefficients a_i^*, that define $D^*(\mathbf{x})$, generally approach zero.

In fig. 3 we present the optimal RBF classifier pertaining to a sample, ***S***, of 100 random points lying on the x-y plane of R^3 (the blue and green circled points visible in the right panel, which represents a top view).

The LP procedure (8) with $\gamma = 0.9$, $\lambda = 0.1$ and $s(\mathbf{x}) = x_1^2 + x_2^2$ furnishes the optimal classifier:

$$D^*(\mathbf{x}) = \Sigma_1^{100} a_i^* \exp(-\gamma \, \|\mathbf{x}_i - \mathbf{x}\|^2) + b^*$$

In our case, this classifier is composed of the combination of only 14 RBF functions that correspond to the 14 optimal coefficients, a_i^*, which are different from zero in the optimisation, and whose composition in R^3 is visible in the left panel of fig. 3. The green circled points in the right panel of fig. 3 represent the *support vectors*, i.e. the boundary points, $\mathbf{x}_S \in \boldsymbol{S}$, where the optimal classifier equals the support function, $D^*(\mathbf{x}_S) = s(\mathbf{x}_S)$.

An independent random sample ***I*** of 4000 points at the origin O was also considered and is presented in fig. 3; rule (6) classifies its points as depicted in its right panel: as black crosses (the case $D^*(\mathbf{x}_{I\_black}) < s(\mathbf{x}_{I\_black})$: $\mathbf{x}_{I\_black}$ does not belong to the class ***S***), and as red stars (the case $D^*(\mathbf{x}_{I\_red}) \geq s(\mathbf{x}_{I\_red})$: $\mathbf{x}_{I\_red}$ does belong to the class ***S***)

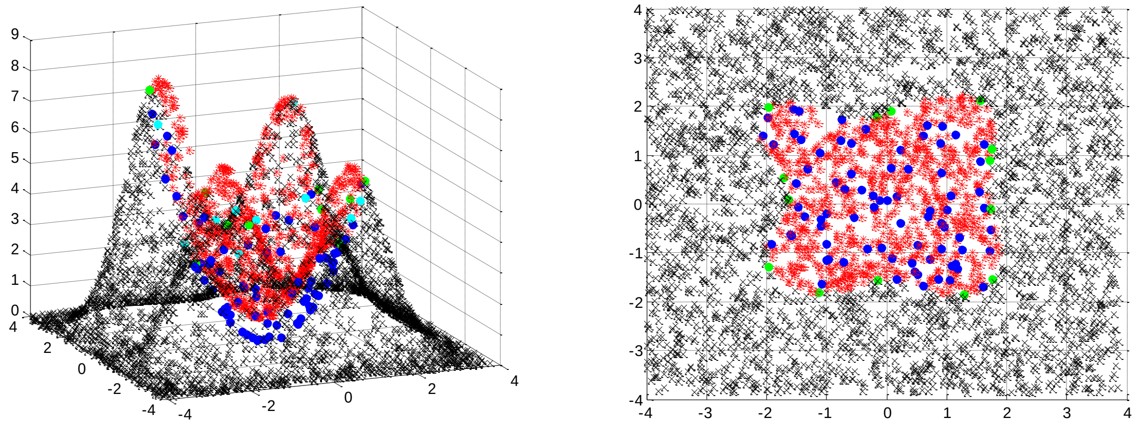


Fig. 3: A solution to (8) with RBF kernels (γ = 0.9 and λ = 0.1. Left, the representation of $D^*(\mathbf{x})$ in R^3; right, a top view)

Fig. 4 presents a second example of the classification that pertains to the same sample ***S*** of fig. 3, with a classifier that consists of power functions (see form b) of previous section). Here h = 0.5 and λ = 0.1. Procedure (8) identifies the optimal $D^*(x_1, x_2) = a_1^* |x_1|^{0.5} + a_2^* |x_2|^{0.5} + b^*$ of fig. 4, with meanings similar to fig. 3.

In particular, only two support vectors derive from the condition $D^*(\mathbf{x}_S) = s(\mathbf{x}_S) = x_{S1}^2 + x_{S2}^2$ (i.e. the two green circled points of the right panel of fig. 4).

A wider membership region (i.e., attributable to class **A** and denoted as before by red stars) can be noted in this second example, which could result in a larger generalization capability if contained into the appropriate limits. But the definition of the appropriate limits is the fundamental task of classification.

5 Negative examples

Negative examples (i.e. instances known to be external to the sample and furnished for better modelling purposes) can be incorporated in the optimisation model with relative ease. In this case, we restate the classifier definition by referring to the RBF functions, for example:

$$D(\mathbf{x}) = a_1\, K(\mathbf{x}_1, \mathbf{x}) + \ldots\ldots + a_n\, K(\mathbf{x}_n, \mathbf{x}) - a_{n+1}\, K(\mathbf{x}_{n+1}, \mathbf{x}) - \ldots\ldots - a_{n+p}\, K(\mathbf{x}_{n+p}, \mathbf{x}) + b.$$

In other words, we associate the negative examples, arranged from the (n+1)-th to the (n+p)-th instance, with negative signs to facilitate their discrimination.

The LP problem now assumes the following form:

$$\left.\begin{array}{ll} \text{Minimize } (b + \lambda\, \Sigma\, a_i), & \text{subject to} \\ D(\mathbf{x}_k) \geq s(\mathbf{x}_k), & \text{for } k = 1, 2, \ldots, n \\ D(\mathbf{x}_k) < s(\mathbf{x}_k), & \text{for } k = (n+1), (n+2), \ldots, (n+p) \\ a_i \geq 0,\ b \geq 0, & \text{for } i = 1, 2, \ldots., (n+p) \end{array}\right\} \qquad (9)$$

We always obtained good results as long as p << n. When p is comparable to n, a solution with procedures of optimal binary classification is clearly preferred.

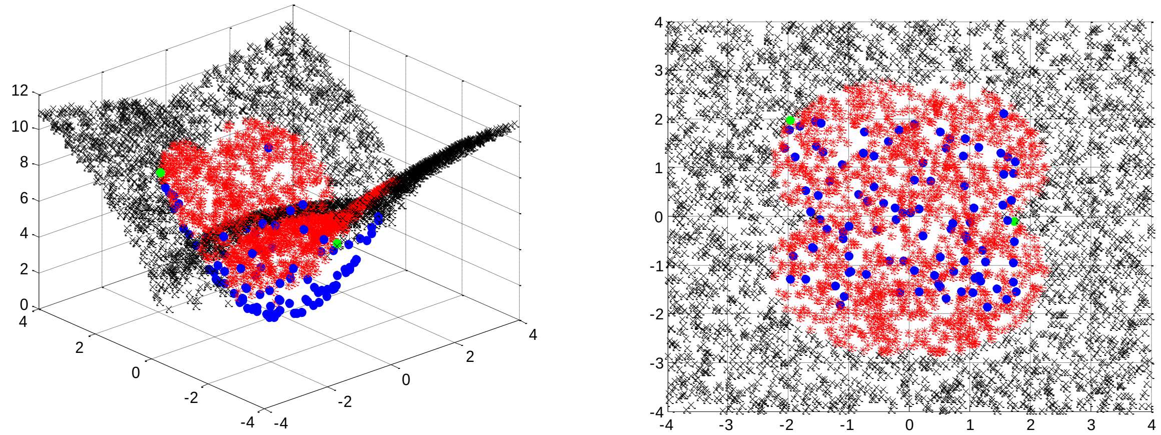


Fig. 4: A solution to (8) with power functions (left, the representation of $D^*(\mathbf{x})$ in R^3; right, a top view)

6 Outlier detection

The detection of outliers, i.e. of observations with a combination of characteristics not attributable to the population, constitutes an important task in statistical analysis. In fact, the results of this analysis can be misleading at best when applied to data containing outliers.

We now discuss two different methods that can detect outliers; the second method is specific to the LP approach presented in this paper.

6.1 A first method: introducing slack variables into the LP problem

When the identification of possible outliers in the sample is desired, the use of slack variables and modified constraints is an available resource [1].

The problem is now restated in a way to determine the optimal classifier identified by the following *modified* LP problem:

$$\text{minimize } (b + \lambda\, \Sigma\, a_i + C\, \Sigma\, \xi_i\,) \text{ subject to}$$

$$D(x_{k1,}\, x_{k2} \ldots\, x_{km}) + \xi_k \geq s\, (x_{k1,}\, x_{k2} \ldots\, x_{km}), \qquad (10)$$

$a_i \geq 0, b \geq 0, \ \xi_i \geq 0, \quad k = 1, 2, \ldots, n; \ i = 1, 2, \ldots$

where C is a free parameter that controls the degree of exclusion.

The constraints in relations (10) state that one or more points, $\mathbf{x}_k$, of the sample are permitted to produce classification values, $D^*(\mathbf{x}_k)$, that are below the support function, s($\mathbf{x}_k$), by a positive quantity, ξ_k, to remain external to the boundary curve identified by the optimal classifier (see point x_n of fig. 5, the left panel, for example).

Based on the choice of C, all points $\mathbf{x}_j$ that ensure that the relative ξ_j are strictly greater than zero will constitute the set of the points excluded from the membership region; on the contrary, the included points meet the conditions $\xi_j = 0$.

The right panel of fig. 5 presents the optimal classifier solution to (10) relative to the same random sample ***S*** of fig. 3 (now with values $\gamma = 0.9$, $\lambda = 0.1$ and C = 0.05); five points of ***S***, represented by black diamonds noted by blue arrows, are excluded from the membership region, which is denoted by red stars. This exclusion reduces the extension when compared to the analogous region of fig. 3.

The problem with this procedure resides in our incapacity to control the parameter C: we are not able to know in advance the number of excluded points that corresponds to an assigned value of C. Therefore, a post analysis is necessary to determine the retention or exclusion of each of them once the candidate outliers have been identified.

To demonstrate the effects of the outliers, we report the results of the analysis conducted on a dataset of 10 classes with 16 attributes (i.e., defined in R^{16}) engaging 10992 instances (see table 1), which were obtained from a benchmark repository [10], named "Pen-Based Recognition of Handwritten Digits Data Set" and available at the following link: http://archive.ics.uci.edu/ml/datasets/Pen-Based+Recognition+of+Handwritten+Digits.

The 2 innermost classes, labelled with codes 4 and 6, were joined together to form the reference class **A**, for a subtotal of 2200 instances. The remaining 8 classes, for a subtotal of 8792 instances, formed a class of "external conditions" (**EC**) resulting quite uniformly distributed around **A**, and assimilated in our analysis to abnormal values/outliers. In this analysis, we considered 7 situations composed of different samples obtained by lining up random subsamples extracted from the class **EC** of 0, 5, 10, 20, 50, 70 and 100 instances behind the class **A**, to create 7 optimal RBF classifiers on these situations of increasing contamination of the reference class **A** with spurious values from the class **EC**.

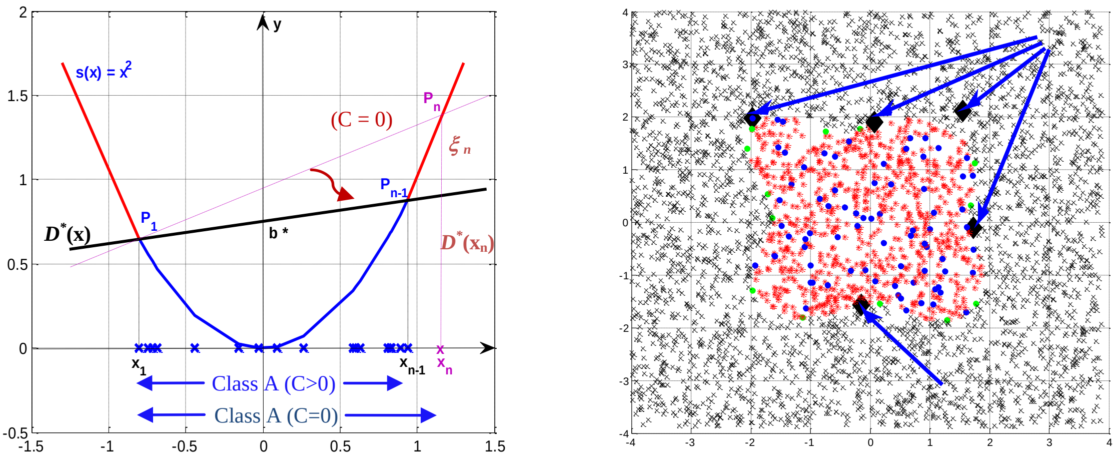


Fig. 5: The concept of variable boundaries with slack variables (left) and the results of the modified LP problem (10) applied to the random sample ***S*** of fig. 3 (right).

The aim of this analysis, whose results are presented in table 1, was to evaluate the amount of "false positiveness" for each classifier (FP; i.e. the number of instances of **EC**, among the remaining 8692, classified as belonging to **A**) and of "false negativeness" (FN; i.e. the number of instances of **A** classified as external) with varying values of C. The analysis is presented for classifiers operating in the space R^{16} built for the following values of C: C = 0, C = 0.03, C = 0.05.

Table 1 shows interesting results. First, we note that optimal classifiers $D^*(\mathbf{x})$ correspond to samples with increasing degree of contamination (from 0 to 4.55%) with a growing degree of FP error: in the worst cases (last row), this error exceeded 60%. This fact demonstrates the influence of outliers on the analysis. The worst result in term of the FP error, which corresponds to a value of 81.90%, was obtained with a classifier built for the value C = 0 (see the last row of table 1). This situation is equivalent to saying that approximately 7118 instances of **EC** (out of 8692) are improperly classified by rule (6) as belonging to **A**, while all values of **A** are properly classified (FN = 0%).

Table 1: Outlier effects on FP and FN for classifiers with increasing contamination

Classification errors (samples: class **A** + random instances extracted from **EC**; bracketed the percent degree of contamination)	C = 0.00, γ = 0.0005, λ = 0.1		C = 0.03, γ = 0.0005, λ = 0.1		C = 0.05, γ = 0.0005, λ = 0.1	
	FP	FN	FP	FN	FP	FN
2200 instances from **A**+0 from **EC** (0%)	3.13%	0%	1.55%	1.45%	2.47%	0.95%
2200 instances from **A**+5 from **EC** (0.23%)	13.79%	0%	2.06%	1.55%	3.37%	0.95%
2200 instances from **A**+10 from **EC** (0.46%)	31.55%	0%	2.50%	1.41%	4.95%	0.86%
2200 instances from **A**+20 from **EC** (0.91%)	43.15%	0%	4.45%	1.45%	16.52%	0.91%
2200 instances from **A**+50 from **EC** (2.27%)	74.82%	0%	39.08%	1.55%	58.84%	1.00%
2200 instances from **A**+70 from **EC** (3.18%)	79.30%	0%	54.72%	1.50%	70.14%	1.00%
2200 instances from **A**+100from **EC**(4.55%)	81.90%	0%	61.25%	1.45%	73.30%	0.82%

Values of C that are greater than zero alleviate the FP error, increasing in contrast the FN error; in other words, the situations with C greater than zero reduce the membership region of **A**, filtering out both instances of **EC** (FP diminishes) and instances of **A** (FN increases).

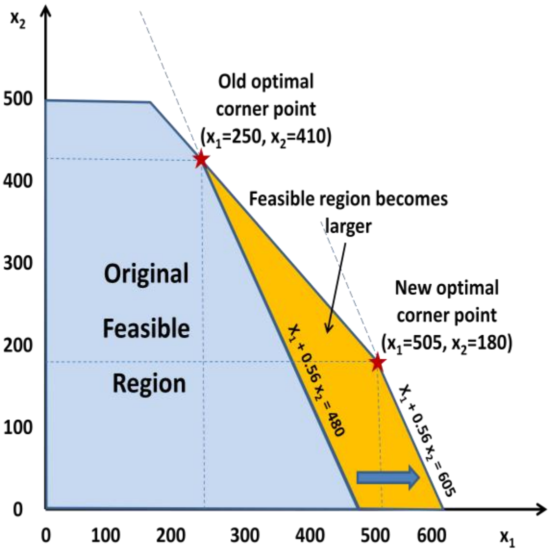
Table 1 indicates that the value C = 0.03 provided the best trade-off between FP and FN.

6.2 A second method: the LP sensitivity analysis

Sensitivity or post-optimal analysis provides additional information in the LP context about the current optimal solution; it aims to determine the effects on the optimal solution if the problem values (e.g. the objective function coefficients (OFC) or the right hand sides (RHS) values of the constraints) change. Some LP solvers furnish solutions to this analysis, which are generally in terms of admissible variation ranges for each OFC or for each RHS (sensitivity report).

Our interest focuses on the RHS sensitivity analysis (see a conceptual schema in the left panel of fig. 6), which consists of calculating the *admissible* variation range on RHS, assuring that the current basis remains optimal while the RHS remains within this range [12].

The report in the right panel of fig. 6 shows an example of a RHS sensitivity analysis produced by the LPSolve IDE [11] solver; this (partial) report states that the RHS of the 38-th constraint (the row R38; in the proposed example it is actually 17,535.77) is allowed to range from 17,002.329 to 1,901,892.561 while maintaining the same optimal basis, whereas to the 51-th constraint (the row R51) pertains a reduced range between 17,238.941 and 62,521.290 (the related RHS amounts actually to 17,322.98).


x_2
500
400
300
200
100
0
Old optimal corner point (x_1=250, x_2=410)
Feasible region becomes larger
Original Feasible Region
New optimal corner point (x_1=505, x_2=180)
$x_1 + 0.56\ x_2 = 480$
$x_1 + 0.56\ x_2 = 605$
0
100
200
300
400
500
600
x_1

Sensitivity Report

	From	Till
R38	17,002.329	1,901,892.561
R39	-inf	+inf
R40	-inf	+inf
R41	-inf	+inf
R42	-inf	+inf
R43	-inf	+inf
R44	-inf	+inf
R45	-inf	+inf
R46	-inf	+inf
R47	-inf	+inf
R48	-inf	+inf
R49	-inf	+inf
R50	-inf	+inf
R51	17,238.941	62,521.290
R52	-inf	+inf

Fig. 6: A change of one RHS in R^2 (left; from 480 to 605) with its consequences and a piece of the sensitivity report by the LPSolve IDE [11] solver (right).

In real terms, maintaining the same optimal basis indicates that the same set of constraints will remain active [12, chap. 6, page 3] and the optimum will remain at the intersection of these constraints. Thus, a wide admissible range on an

RHS implies that the related constraint has “enough room” to permit large parallel translations to the hyperplane that represents it (e.g. see the highlighted region in the left panel of fig. 6), while remaining active.
Moreover, recall that the generic k-th constraint is expressed as follows when employing RBF, for example:

$$D(\mathbf{x}_k) = a_1\,K(\mathbf{x}_1, \mathbf{x}_k) + a_2\,K(\mathbf{x}_2, \mathbf{x}_k) + \dots\dots\dots\dots + a_n\,K(\mathbf{x}_n, \mathbf{x}_k) + b \;\geq\; RHS_k = s(\mathbf{x}_k),$$

which involves the instance $\mathbf{x}_k$ of the sample. It follows then that a wide admissible range on RHS_k should be associated with a meaning of marginality of the instance $\mathbf{x}_k$ with respect to the pattern distribution because large translations are allowed to the hyperplane expressed by means of $\mathbf{x}_k$ in the kernel space $K(\bullet,\bullet)$, while holding the same optimal basis and the constraint active.
The criterion of considering an instance $\mathbf{x}_k$ marginal or of identifying it as outlier when the related constraint is active and its admissible range on RHS is large has been tested on the sample already presented in section 6.1; this method was simultaneously compared with the competing method of the slack variables.
The analysis was organised to build 10 subsamples, SS_i, each of which consisted of 220 instances randomly extracted from class **A**, followed by 10 instances randomly extracted from class **EC**, as represented below:

SS_i = {220 instances from **A** + 10 instances from **EC**}, i = 1, 2, 10.

Moreover, the 10 subsamples SS_i resulted without superposition ($SS_i \cap SS_k$ = the empty set, $\forall$ i, k = 1, 2, ..., 10). For each SS_i two optimal RBF classifiers were built: the first turned only to LP sensitivity analysis ($C = 0.0$, $\gamma = 0.0005$, $\lambda = 0.1$), the second containing slack variables ($C = 0.08$, $\gamma = 0.0005$, $\lambda = 0.1$). Both cases aimed to evaluate the number of True Outlier Attributions (#TOA; i.e. how many instances of **EC**, out of the 10 really present in the subsample, are properly identified as outliers), and the number of False Outlier Attributions (#FOA; i.e. how many instances of **A**, out of the 220 really present in the subsample, are interpreted as outliers).
Table 2 presents the results of the comparisons between the two methods. Moreover the last column expresses the percent averages of #TOA and #FOA. In each cell, the first number refers to #TOA and the second to #FOA; for example, the cell at the intersection of column (subsample) “SS_2” and of row “Sensitivity Analysis”, which reports the values (3, 2), states that 3 instances of **EC** (out of 10) and 2 of **A** (out of 220) were properly and improperly recognised, respectively, as outliers by the optimal classifier $D^*(\mathbf{x})$ built with parameters $C = 0.0$, $\gamma = 0.0005$, $\lambda = 0.1$ on SS_2.

Table 2: Comparison between the two methods of outlier detection (#TOA, #FOA)

($\gamma = 0.0005$, $\lambda = 0.1$)	SS_1	SS_2	SS_3	SS_4	SS_5	SS_6	SS_7	SS_8	SS_9	SS_{10}	Average (%)
Sensitivity Analysis (C = 0)	7, 0	3, 2	4, 3	7, 0	5, 1	10, 1	5, 0	7, 2	9, 1	7, 0	64%, 0.45%
Slack variables (C = 0.08)	5, 3	4, 2	6, 3	9, 0	6, 3	9, 3	6, 0	8, 2	8, 4	8, 1	69%, 0.95%

Based on the last column, the two methods seem to compete at similar levels of quality; the method with slack variables shows a slight advantage on #TOA (69% vs. 64%) at the cost of a lower performance on #FOA (0.95% vs. 0.45%).
From a computational point of view, the sensitivity analysis method gains an advantage over the other method in terms of computational efficiency: the LP matrix is reduced in size, because fewer variables are involved in optimization (absence of slack variables).

7 Comparing the LP model with other models of data description

In the present section we focus on the classification errors, aiming to evaluate the error of the first kind or of false negativeness (FN; i.e. the amount of rejected target patterns) and the error of the second kind or of false positiveness (FP; i.e. the amount of accepted outlying patterns) for competing models of data description.
Two categories of procedures are generally used to evaluate the error of a model. Procedures from the first category estimate the error by testing the model on an independent dataset (i.e. not used for training), while those from the second category estimate the error by theoretical bounds.
At present, the cross-validation procedure, which falls into the first category, is one of most popular methods in the statistical literature and in this section we show results obtained via this method.
The dataset of section 6.1, originally subdivided into the target class **A** with 2200 instances and the outlying class **EC** with 8792 instances, was now rearranged into 5 different *aggregations of subsamples* (AoS_i, i = 1, 2,…, 5), each of which was structured as follows:

AoS_i = {440 instances from **A** + 1760 remaining instances from **A** + **EC**}, i = 1, 2… 5.

More precisely, 440 instances were randomly extracted from **A** to form the first component of each AoS_i, without overlapping this component among the various AoS_i, acting as *independent* target data. The component with 1760 instances from **A** was used as training data to build the classification model each time. A 5-fold cross-validation procedure was then followed to estimate FN_{td} (FN on training data, intending the improperly rejected instances among the 1760 instances used in training), FN_{itd} (FN on the 440 independent instances) and FP (the improperly accepted outliers from **EC**). Table 3 shows the results of the comparison between our procedure, denoted LP model, and the model presented in [1], denoted QP model. Each cell of table 3 reports the values averaged over the 5 analyses, in terms of the number (absolute and percent) of instances recognised as support vectors (SV), and instances related to FN_{td}, FN_{itd}, and FP. The cross-validation analysis was performed in the Matlab environment, using its *linprog* and *quadprog* routines (for the LP and QP model respectively); the resultant computing times distinctly favoured the LP model.

A grid search was preliminarily conducted to determine the appropriate values of the free parameters C and γ.

Table 3: Comparison between models (averaged results from a 5-fold cross-validation)

	SV (out of 1760 instances)	*FN_{td} (out of 1760 instances)*	*FN_{itd} (out of 440 instances)*	*FP (out of 8792 instances)*
LP model (C = 0.01, γ = 0.08, λ = 0.1)	8 (0.45%)	100 (5.68%)	26.6 (6.05%)	14.6 (0.17%)
QP model (C = 0.01, γ = 0.114)	131 (7.44%)	45.6 (2.59%)	43.4 (9.86%)	0 (0%)

Several important conclusions can be drawn from table 3:

1) The number of support vectors between the LP and QP model is undoubtedly different; other analyses conducted that varied the parameters C and γ around the values of table 3 did not reduce this marked difference;
2) Notably, the mathematical modeling implemented by the two methods is different, and the definition of the support vector refers to different geometrical objects: support vectors are generally model-dependent. Thus, the two methods reported a superposition of the respective support vector sets to a low degree. For example, the fifth analysis of the cross-validation procedure reported a sharing of 2 out of 8 support vectors in the LP model and 139 support vectors in the QP model;
3) Moreover, we found that the numbers of optimal coefficients different from zero in the LP and the QP solutions are definitely unequal in the same direction and extent. Thus, the data description operated by the LP model is more parsimonious and simpler;
4) If contained within the appropriate limits, this simplified model structure could improve the generalisation capability, because LP modeling seems to concentrate more on the general properties of data distribution;
5) Table 3 shows a small percent increment in the LP model when passing from FN_{td} to FN_{itd} (from 5.68% to 6.05%), whereas the analogous increment in the QP model is more pronounced (from 2.59% to 9.86%). This difference indicates that the LP model is more stable when classifying independent target data at the cost of a worse level of target data classification (FN_{td} passes from 2.59% in the QP model to 5.68% in the LP model);
6) The FP does not seem to report significant differences (0.17% versus 0%).

This result, which was based on a single analysis, clearly requires deeper investigations to completely confirm the relative characteristics of the two models.

Nevertheless, we found a systematic trend of the LP procedure to produce models with a marked reduction of support vectors and optimal coefficients different from zero in our experience with both procedures.

8 Conclusion

This paper presented a new method that aimed to build a one-class classifier. The method is based on the effects of a support function, s(x), that acts as a selector for support vectors and a classification curve, when it results crossed by the optimal classifier in the input or feature space. The data modelling is performed in the context of an LP optimization process to ensure the advantages of linearity.

A versatile support function was proposed, as a general formulation to address regular dataset distributions, whereas particular distributions can be usefully handled by tightly fitting specialized functions to the reference data. For example, in the case of data grouped into two separated regions, the Cassini oval [5], which consists of two disconnected loops when proper values of its characteristic parameter are assumed, can be considered a support function.

The data description model presented in this paper relies on a sparse and defined set of support vectors, whose cardinality resulted undoubtedly smaller than the cardinality of the competing QP model. Thus, the data description operated by the LP model is more parsimonious and simpler.

A new method of outlier detection, based on LP sensitivity analysis, was also proposed. This method demonstrated its validity when compared to the classic method with slack variables, thus enriching the availability of useful techniques when addressing this delicate but fundamental task of statistical analysis.

A Matlab code is freely available on request.

References

[1] D.M.J. Tax, R.P.W. Duin, Support vector data description, Machine Learning, vol. 54, pp.45-66 (2004)

[2] B. Scholkopf, J.C. Platt, J. Shawe-Taylor, A.J. Smola, R.C. Williamson, Estimating the support of a high-dimensional distribution, Microsoft Research Corp., Technical Report MSR-TR-99-87 (1999)

[3] B.E. Boser, I.M.Guyon, V.Vapnik, A training algorithm for optimal margin classifiers, Proceedings Fifth ACM Workshop on Computational Learning Theory, Pittsburgh (1992)

[4] R. Ragona, A minimax Chebyshev approach to optimal binary classification, International Journal of Applied Mathematical Research, 2 (2) 175-187 (2013)

[5] G.A. Korn, T.M. Korn, Mathematical handbook for scientist and engineers, McGraw-Hill Book Company, 1968

[6] N. Cristianini, J. Shawe-Taylor, An introduction to support vector machines, Cambridge University Press, 2004

[7] C.J.C. Burges, A tutorial on support vector machines for pattern recognition, Data Mining and Knowledge Discovery, 2, 121–167 (1998)

[8] A. N. Tikhonov, V. Y. Arsenin. Solution of ill-posed problems, Winston & Sons, 1977

[9] R. Tibshirani, Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society, Series B 58 (1): 267–288 (1996)

[10] Bache, K. & Lichman, M. (2013). UCI Machine Learning Repository, at: http://archive.ics.uci.edu/ml. Irvine, CA: University of California, School of Information and Computer Science.

[11] LPSolve IDE, freely available at: http://lpsolve.sourceforge.net/5.5/IDE.htm.

[12] John W. Chinneck, Practical optimization: a gentle introduction (2012), freely available at http://www.sce.carleton.ca/faculty/chinneck/po.html

[13] C. Campbell, K.P. Bennet, A linear programming approach to novelty detection, Neural Information Processing Systems, volume 13, pages 395-401 (2000).

[14] E. Pekalska, D.M.J. Tax, R.P.W. Duin, One-class LP classifier for dissimilarity representations, Advances in Neural Information Processing Systems, pages 761-768, (2002)