*International Journal of Applied Mathematical Research, 3 (3) (2014) 196-206*
*©Science Publishing Corporation*
*www.sciencepubco.com/index.php/IJAMR*
*doi: 10.14419/ijamr.v3i3.2449*
*Research Paper*

# Sensitivity analysis in linear programming approach to optimal SVM classification

**Roberto Ragona**

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

## Abstract

At present, linear programming (LP) techniques for optimal one-class and two-class classification can be considered well established and feasible; they pose an alternative to the quadratic programming (QP) approach, which is usually credited with having greater complexity. Sensitivity analysis, well developed in the LP context, is generally employed to furnish answers describing how an optimal solution changes when varying the parameters in an LP problem; as a possible application in optimal classification, it can be employed for the definition of the free parameters present in LP procedures, reducing the events of computational restart from scratch when searching for a satisfactory classifier through repeated trials. The proposed method is demonstrated on a simple example which exhibits its effectiveness in reducing the computational burden, but this procedure can be extrapolated to large problems as well.

***Keywords**: Linear Programming, Optimal Classification, Sensitivity Analysis, Support Vector Machines.*

## 1. Introduction

Classic Support Vector Machine (SVM) techniques for one-class and two-class classification have led to successful solutions following a non-parametric approach oriented by a quadratic optimisation criterion; in fact neither SVM technique assumes any knowledge of the forms of the underlying probability distributions. Classic SVM techniques for both one-class and two-class contexts are now well developed and have been presented in a series of foundational papers and books, e.g., [1], [2], [3], [4]. The related applications are formulated as constrained quadratic optimisations which, under conditions of positive semi-definiteness [5, p. 172] of the matrix which describes the objective function, lead to a convex quadratic programming problem, and can be assured of finding a global maximum (minimum) of the objective function, although there might be cases in which the solution is not unique.

Other authors report optimal methods of supervised one-class and two-class classification when using linear programming techniques, e.g., [5, p. 230], [6], [7], [8], which gained approval as competing methods, because they are usually credited with having lower complexity. For two-class classification, researchers have mostly looked at maximizing margins measured using the $L_1$ or $L_\infty$ norms of the weight vector (instead of the $L_2$ norm), both of which in fact lead to linear programming formulations. Different solutions to one-class and two-class problems following a geometric approach were presented instead in [9], [10].

The $L_2$ norm SVM for two classes has stimulated a lot of research and efficient implementations for solving it are available (e.g., SVMlight [11], LIBSVM [12]).

Also in the competing LP context for two classes, methods (see e.g. [13], [14], and [7]) attempting to make learning $L_1$ norm SVMs practical for large data sets were developed, so at present the two competing approaches can be considered well developed and feasible.

Linear techniques offer the advantage of an increased capability of an analytical treatment, which paves the way to many analyses and results typical of this context; among other things, sensitivity analysis (SA). A number of problems can be asked concerning the sensitivity of an optimal solution to changes in the parameters, and SA addresses those that can be answered easily.

The main aim of this paper is to apply SA concepts to optimal classifiers defined through linear techniques.

In Section 2 we will focus on the basic formulations for one-class and two-class problems, as reported in [9] and [10], with some changes in order to attain a common framework; in Section 3 we present some fundamental results of SA, and in Section 4 a practical analysis will be conducted to demonstrate the possible advantages of its application.

## 2. LP one-class and two-class optimal classifiers

We report the formulations proposed in [10] and [9] to describe a possible approach to solve the one-class and the two-class classification problem via LP, and our SA will be developed around them. Similar considerations can be developed around any other classifier inspired by linear techniques.

### 2.1. LP formulation for supervised one-class classification

The one-class LP classifier problem takes the following form [10, problem (10)]:

$$\left.\begin{array}{l}\text{minimise } (b + \lambda\ \Sigma a_i + C\Sigma\ \xi_k\ ), \text{ subject to}\\ D\,(\mathbf{x}_k) + \ \xi_k\ \geq\ s\,(\mathbf{x}_k),\\ a_i \geq 0,\ b \geq 0,\ \lambda \geq 0,\ C > 0_,\ \xi_k\ \geq\ 0,\ k = 1, 2, \ldots, n;\ i = 1, 2, \ldots, N\end{array}\right\} \qquad (1)$$

where the target class **A** is composed of points $\mathbf{x}_k = [x_{k1}, x_{k2}... x_{km}]^T$ (k = 1, 2... n) defined in the real vector space $R^m$, and $s(\mathbf{x}_k)$ is the value of the support function calculated on $\mathbf{x}_k$.

The non-linear classifier $D(\mathbf{x})$ is defined through N arbitrary function $\varphi_i(\mathbf{x})$, so implying a projection from the space $R^m$ to the space $R^N$:

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

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

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

A very common form assumed by the classifier $D(\mathbf{x})$ is the following:

$$D(\mathbf{x}) = a_1\ K(\mathbf{x}_1, \mathbf{x}) + a_2\ K(\mathbf{x}_2, \mathbf{x}) + \ldots\ldots\ldots. + a_n\ K(\mathbf{x}_n, \mathbf{x}) + b,$$

where $K(\mathbf{x}_i, \mathbf{x})$ is any dot product of the type

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

with possibly infinitely many terms in $\boldsymbol{\varphi}\,(\bullet)$ $(N\rightarrow\infty)$, provided the dot product is finite.

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

$$K(\mathbf{x}_i, \mathbf{x}) = \exp\,(-\gamma\ \|\mathbf{x}_i - \mathbf{x}\|^2) \qquad (i = 1, 2... n) \qquad \text{(RBF kernel)}$$

Interestingly, in general $K\,(\bullet_,\bullet)$ is not required to be a Mercer kernel [2] in the LP context: it can be a generic dot product [10, §3].

The term

$$\lambda\ \Sigma a_i \qquad (2)$$

occurring in (1) derives from a consideration of *regularisation*: if the $a_i$'s are unconstrained, they can be very large and hence susceptible to high variance.

In (1) a LASSO-type [15] criterion of regularisation, consisting in general of an additional term of the type

$$\lambda\ \Sigma|a_i| \qquad (3)$$

in the objective function, was assumed. In fact LASSO suits our context very well if we impose the additional constraint of the positivity of the $a_i$: then (3) simply reduces to (2), which ensures that problem (1) remains linear.

Therefore, our willingness to accept suboptimal conditions is rewarded by viable solutions in real terms; in [10] an analysis to account for this term was accomplished, reporting satisfying results.

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 (1): in our experience many optimal coefficients $a_i^*$, which define the optimal classifier $D^*(\mathbf{x})$, generally approach zero.

The term

$$C\ \Sigma\ \xi_k$$

in (1) accounts for outlier treatment.

### 2.2. LP formulation for the supervised two-class classification

The two-class LP classifier problem assumes the following form [9, problem (11)]

$$\left.\begin{array}{l} \text{minimise } (m + C\sum_1^p \xi_i\ ), \text{ subject to} \\ m - c_i D(\mathbf{x}_i) \geq 0 \\ c_i D(\mathbf{x}_i) + \xi_i \geq 1 \\ \xi_i \geq 0, C > 0, i = 1, 2\ldots p \end{array}\right\} \qquad (4)$$

Here we have two training classes, **A** and **B**, composed respectively of $n_A$ and $n_B$ points, and a non-linear classifier defined through N functions $\varphi_i(\mathbf{x})$

$$D(\mathbf{x}) = \sum_{i=1}^{N} w_i\varphi_i(\mathbf{x}) + b = [w_1 w_2 \ldots\ldots w_N] \bullet [\varphi_1(\mathbf{x})\ \varphi_2(\mathbf{x}) \ldots\ldots \varphi_N(\mathbf{x})]^T + b = \mathbf{w}^T \bullet \boldsymbol{\varphi}(\mathbf{x}) + b$$

Moreover

$p = n_{\mathbf{A}} + n_{\mathbf{B}}$ (sample size), $c_i = +1$ if $\mathbf{x}_i \in \mathbf{A}$, $c_i = -1$ if $\mathbf{x}_i \in \mathbf{B}$.

The term

$$C\sum_1^p \xi_i$$

in the objective function accounts for the treatment of inseparable classes.

It can be demonstrated [9] that the optimal classifier $D^*(\mathbf{x})$ can be expressed up to a tight approximation as:

$$D^*(\mathbf{x}) = \sum_1^N w_i^*\ \varphi_i(\mathbf{x}) + b^* \cong \sum_1^p \theta_i^*\ K(\mathbf{x}_i, \mathbf{x}) + b^*, \text{ where}$$

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

is a dot product between N-dimensional vectors.

A usual assumption is the following

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

Here we propose a variant to (4) obtained by introducing in the objective function a LASSO-type term, to pursue the advantages already described in the one-class case. The analysis of possible advantages is beyond the scope of the present work and will be discussed in a subsequent paper, so we assume here this formulation with only a rapid analysis about the obtainable benefits: in any case, this term does not invalidate the effectiveness of the present analysis.

The new formulation becomes the following:

$$\left.\begin{array}{l} \text{minimise } (m + \lambda \sum_1^p \theta_i\ + C\sum_1^p \xi_i\ ), \text{ subject to} \\ m - c_i D(\mathbf{x}_i) \geq 0 \\ c_i D(\mathbf{x}_i) + \xi_i \geq 1 \\ \theta_i\ \geq 0, \xi_i \geq 0, \lambda \geq 0, C \geq 0, i = 1, 2\ldots p \end{array}\right\} \qquad (5)$$

The term

$$\lambda \sum_1^p \theta_i$$

represents the LASSO-type term in the objective function deriving from a term of the type (3), with all $\theta_i$ positive to retain linearity.

Moreover, a second variation is considered to facilitate a solution when all $\theta_i$ are assumed positive: the classifier $D(\mathbf{x})$ is split into three terms and forced to assume the following configuration:

$$D(\mathbf{x}) = \sum_1^{nA} \theta_i\ K(\mathbf{x}_i, \mathbf{x}) - \sum_1^{nB} \theta_i\ K(\mathbf{x}_i, \mathbf{x}) + b \qquad (6)$$

in other words, the terms $\theta_i$ concerning the class **A** furnish a positive contribution to $D(\mathbf{x})$, when $K(\mathbf{x}_i, \mathbf{x}) \geq 0$, whereas those concerning the class **B** give rise to negative contributions, in order to have a higher degree of compliance with its fundamental aim. To justify this choice, let us assume for example the RBF kernel and consider a point $x_k \in \mathbf{B}$.

All $K(\mathbf{x}_i, \mathbf{x}_k)$ are positive by definition; moreover, they will have (relatively) large values if $x_i \in \mathbf{B}$, and will have small values if $x_i \in \mathbf{A}$. Therefore, because all $\theta_i$ are positive, it is easier for $D^*(\mathbf{x}_k)$ to become negative (in order to assign $\mathbf{x}_k$ to class **B**) when subtracting the first two terms in (6), more than in the opposite case of addition of the first two terms.

Fig. 1 presents an example of classification developed on a random sample of class **A** (the brown balls) and class **B** (the black diamonds) of 100 points located in $R^2$. Moreover, the left panel presents the class attribution of 4000 independent random points located in $R^2$ around O, once problem (4) has been solved, i.e., without any regularisation; the right panel shows the class attribution of the same 4000 points, once problem (5)–(6) is solved, i.e., with a regularisation parameter of $\lambda = 0.01$. The blue crosses in Fig. 1 are points assigned by the classifiers to class **A**, the red crosses to class **B**; together they form the sample of 4000 points.

The optimal classifier built without regularisation wound up being made up of 13 coefficients $\theta_i^*$ different from zero, one of them being as high as 2.56e+13; with a regularisation parameter $\lambda = 0.01$ the resulting optimal classifier was instead composed of 3 coefficients $\theta_i^*$ different from zero, the greatest of them being equal to 44.1.

As a consequence, the data description used by the model containing the LASSO-type regularization term is more parsimonious and simpler; if maintained within the right limits, this simplified model structure should mean a better generalization capability, because this modeling seems to concentrate more on the general properties of the data distribution.

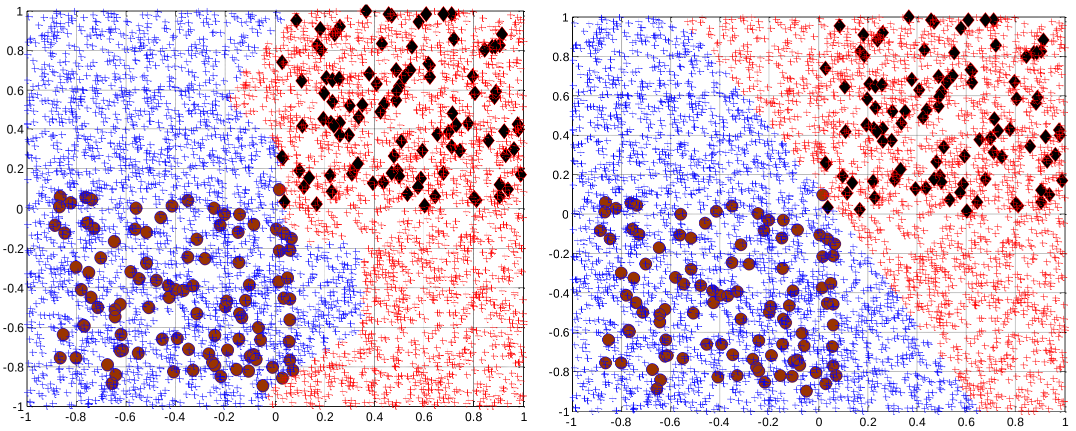

**Fig. 1:** Two Optimal Classifications with RBF Kernel, C = 0 and $\gamma = 0.06$ Build on the Same Random Sample (Left Panel, without Regularisation – Right Panel, with A Regularisation Parameter $\lambda = 0.01$).

Looking at Fig. 1, right panel, the partition into regions attributable to class **A** and class **B** is regular and quasi-linear, as expected from this simple data distribution; the competing classification (left panel) is more involved.

The programming expressed by (1) (one-class classifier) and by (5) (two-class classifier) share a similar structure and attain a common linear context; SA develops in the two situations along similar lines, therefore it will be detailed only for the case of two-class classifications.

## 3. Results of linear programming and sensitivity analysis

We present a review of results concerning LP and SA; extensive reference to the textbook [5] will be made, because of its well-developed algebraic approach. In particular we will refer to chapter 6 of [5].

The canonical form of linear programming assumes the representation

$$\left.\begin{aligned}&\text{minimize } \; z_v = \mathbf{p}^T\mathbf{v}\\ &\text{subject to } \; \mathbf{A}\,\mathbf{v} = \mathbf{b},\\ &\mathbf{v} \geq \mathbf{0}\end{aligned}\right\} \tag{7}$$

A tableau for the canonical form problem is fully specified [5, p. 52] by giving the set B of *basic* variables and the complementary set N of *nonbasic* variables. The tableau for a given basis B is the following (Table 1)

**Table 1:** A Tableau for The Canonical LP

| No. of columns: dim(**v**) | 1 column |
|---|---|
| $-\mathbf{A}_B^{-1}\mathbf{A}_N$ | $\mathbf{A}_B^{-1}\mathbf{b}$ |
| $\mathbf{p}_N^T - \mathbf{p}_B^T\,\mathbf{A}_B^{-1}\mathbf{A}_N$ | $\mathbf{p}_B^T\,\mathbf{A}_B^{-1}\mathbf{b}$ |

In Table 1, $\mathbf{A}_B$, $\mathbf{A}_N$, $\mathbf{p}_B$ and $\mathbf{p}_N$ denote the partition of **A** and **p** into basic and nonbasic component.

The tableau is optimal for (7) if and only if the reduced costs and the final column $\mathbf{A}_B^{-1}\,\mathbf{b}$ are all nonnegative [5, p. 152]; that is, iff

$$\mathbf{c}^T = \mathbf{p}_N^T - \mathbf{p}_B^T\,\mathbf{A}_B^{-1}\mathbf{A}_N \text{ (vector of reduced costs) } \geq \mathbf{0}, \textit{ and}$$

$$\mathbf{A}_B^{-1}\,\mathbf{b} \geq \mathbf{0}$$

In particular, when the tableau is optimal, we obtain the following results for the optimal solution $\mathbf{v}^*$ and the optimal objective z* in canonical form:

$$\mathbf{v}^* = \mathbf{A}_B^{-1}\,\mathbf{b}$$

$$z^* = \mathbf{p}_B^T\,\mathbf{A}_B^{-1}\,\mathbf{b}.$$

Looking at (1) or (5), we see that these problems are affected by free parameters ($\lambda$ and/or C) in the objective function, and we stress such a situation by a reformulation which gives evidence only to one parameter for the sake of simplicity.

This new formulation of the objective function, called the cost-parametrised version [5, p. 159], is the following:

$$z(t) = (\mathbf{p}^T + t\,\mathbf{q}^T)\,\mathbf{v} = (\mathbf{p}^t)^T\,\mathbf{v}, \tag{8}$$

where the dependence on a free parameter t is represented, and **q** is a vector of fixed variations with respect to **p**.

It is a simple matter to demonstrate that the objective functions in (1) or (5) can be expressed as reported in (8) when we assume either $\lambda$ or C as free parameter, but not both.

Sensitivity analysis, among other things, says how an optimal solution changes when the parameter t varies in this objective function; this analysis is of paramount importance because modellers know the problem of a proper choice of $\lambda$ and/or C, which often requires the computational restarting of an LP procedure from scratch, to consider modified values of $\lambda$ and/or C during the search for a satisfactory classifier.

We now recall from [5, theorem 6.3.1 and corollary 6.3.2] an important result which describes the infimum (greatest lower bound) of z(t) as a function of t.

**Theorem 1:** *Let X be a nonempty polyhedral set in* $R^n$ *(that is, a set defined by a finite number of equality and inequality linear constraints), and let* z(t) *be defined by*

$$z(t) \triangleq \inf_v (\mathbf{p}^T + t\,\mathbf{q}^T)\,\mathbf{v} = \inf_v (\mathbf{p}^t)^T\,\mathbf{v},\ \mathbf{v} \in X.$$

Then z(t) is a *piecewise-linear concave* function.

This result is useful to demonstrate that in an LP cost-parametrised problem the range (-∝, +∝) of t can be partitioned into a finite number of subintervals separated by breakpoints $t_1$, $t_2$, ..., $t_M$ at which the optimal cost function z*(t) switches from -∝ to finite or switches from one vertex of *X* to another [5, p. 163], keeping (when finite) linearity in each subinterval. Correspondingly, the components of the optimal solution $\mathbf{v}^* = \mathbf{A}_B^{-1}\,\mathbf{b}$ (when finite) at breakpoints switch among *constant* values, because they do not depend on t as long as the current basis B remains the same (only the terms $\mathbf{p}^t_B$ and $\mathbf{p}^t_N$ are affected by the parameter t); therefore, the components of **v*** take the form of stepwise functions.

To arrange (5) in canonical form, see (7), the vector **v** has to assume the following structure:

$$\mathbf{v} = [m\ \theta_1\ \theta_2\ ......\ \theta_p\ \ b\ \xi_1\ \ ......\ \xi_p\ \ s_1\ s_2\ ....]^T$$

where the terms $s_k \geq 0$ are slack variables, required to transform inequality constraints into equality constraints.

Fig. 2 presents a situation of variation of z*(λ) and $v_1$*(λ) = *m**(λ); it was developed around a “toy example” of class **A** and class **B** formed of (6+6) random points in $R^2$, by solving repeatedly problem (5) with C = 0, γ = 0.6 for the RBF kernel and λ varying in the range [0.005, 1] with step size Δλ = 0.001, therefore having about 1000 LP problems solved to explore the reported range. There is evidence of a piecewise-linear behaviour of z*(λ) and of a stepwise behaviour of *m**(λ): to any slope variation of z*(λ) there corresponds a jump of *m**(λ), lined up with the same breakpoint.

In Fig. 2 two values of λ are shown, pertaining to two breakpoints and derived from the repeated solution of problem (5). Inside any subinterval the optimal **v*** is unchanged, therefore the same optimal classifier remains associated to each of them; breakpoints are important because they denote a change in the classifiers, each of them with an assigned number of support vectors, and can address modellers toward a proper choice, but a “brute force” definition of them (as done in Fig. 2 with a systematic variation of λ) is in general computationally heavy.

We will show how SA in effect can reduce drastically this computational burden, attaining the same results.

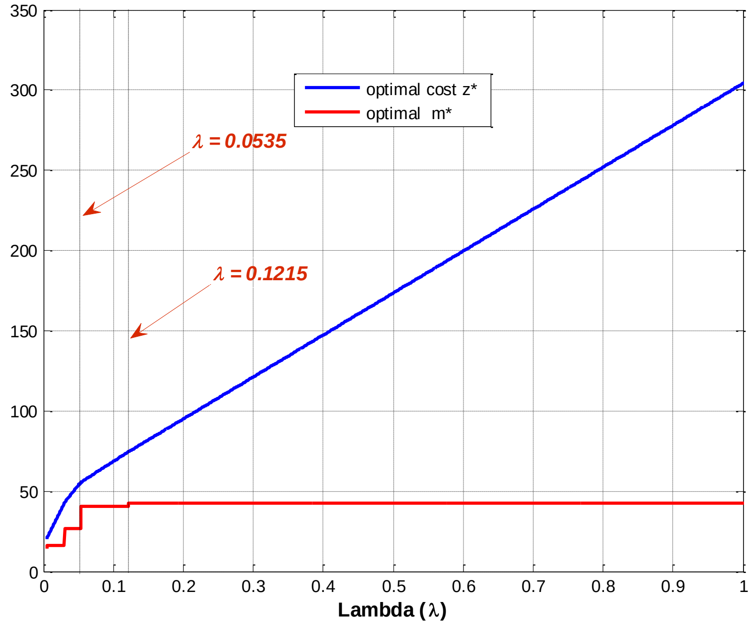

**Fig. 2:** Variations of Z* and *M** As A Function of the Regularization Parameter λ (C = 0, $\gamma_{RBF}$ = 0.6)

## 4. Sensitivity analysis in practice

To present SA in practice and to illustrate the analysis step by step, we continue to consider the toy example of 12 random points in $R^2$ presented above.

A variant of the procedure suggested in [5, §6.3, p. 162] is employed; moreover, Matlab functions developed and presented there [5, App. B.2] are also used to obtain the results.

In particular, the commands:

a) H = totbl(**A**, **b**, **p**) (make the matrices/vectors **A**, **b**, **p** into tableau H);

b) H = addrow(H, **x**, lbl, z) (add row vector **x** with label lbl as row z of the tableau H);

c) B = ljx(H, r, s) (perform a labelled Jordan exchange with pivot H(r, s) to obtain the tableau B),

will play a fundamental role in the analysis that we now go on to present.

## 4.1. The case of LP classifiers with one free parameter (the parameter λ)

Let us suppose we have the following problem, derived from (5) when C = 0:

$$\left.\begin{array}{l} \text{minimise } z(\mathbf{v}, \lambda)_{\text{v}} = (m + \lambda \sum_1^{12} \theta_i\ ), \text{ subject to} \\ m - c_i D(\mathbf{x}_i) \geq 0 \\ c_i D(\mathbf{x}_i) \geq 1 \\ \theta_i\ \geq 0, \lambda \geq 0, \text{ i} = 1, 2\ldots 12 \end{array}\right\} \qquad (9)$$

to be solved for the toy example; this is a formulation which excludes the parameter C.

We fix an initial large value for λ, say $\lambda_0$ = 10, solve for it and next organise the procedure in accordance with (8) to consider decrements of the free parameter λ towards zero. Therefore, we arrange the problem in the following way:

$$\left.\begin{array}{l} \text{minimise } z(\mathbf{v}, \lambda)_{\text{v}} = ([1\ \lambda_0\,\lambda_0 \ldots\ldots \lambda_0\ 0\ 0] + (\lambda_0 - \lambda)\ [0\ \text{-}1\ \text{-}1\ \ldots\ \text{-}1\ 0\ 0])\ [m\ \theta_1\ \ \theta_2\ \ldots\ldots\ \theta_{12}\ b_1\ b_2]^{\text{T}}, \text{ i.e.} \\ \text{minimise } [\mathbf{p}^{\text{T}} + (\lambda_0 - \lambda)\ \mathbf{q}^{\text{T}}]\ [m\ \theta_1\ \ \theta_2\ \ldots\ldots\ \theta_{12}\ b_1\, b_2]^{\text{T}} \triangleq \min\ [\mathbf{p}^{\text{T}} + (\lambda_0 - \lambda)\ \mathbf{q}^{\text{T}}]\ \mathbf{v}, \text{ subject to} \\ m - c_i D(\mathbf{x}_i) \geq 0 \\ c_i D(\mathbf{x}_i) \geq 1 \\ m \geq 0, \theta_i\ \geq 0, b = b_1 - b_2, b_1 \geq 0, b_2 \geq 0, \lambda_0 = 10, 0 \leq \lambda \leq 10, x_i \in R^2, \text{ i} = 1, 2, \ldots\ldots, 12. \end{array}\right\} \qquad (10)$$

In (10) we used the trick of representing the sign-free parameter *b* (the bias term of the classifier *D*(**x**)) as the difference between $b_1$ and $b_2$ (both positive), in order to maintain all the 15 components of **v** positive or null.

As a first step, we solve the LP programming (10) in its primal and dual symmetric forms ([16], p. 85) relatively to the initial situation $\lambda_0$=10 and λ =10: as is known, the primal solution **v*** (generally furnished without slack variables) identifies also the optimal basis, the dual solution **d*** shows with its non-zero components which constraints are active at optimality ([16], theorem 2, condition iii), p. 96).

In our case we obtain the following results from a call to Matlab LP solver (*linprog*), able to furnish both solutions:

$$\left.\begin{array}{l} \mathbf{v}^* = [42.65\ \ 0\ \ 81.87\ \ 0\ \ 0\ \ 0\ \ 90.57\ \ 0\ \ 0\ \ 89.04\ \ 0\ \ 0\ \ 0\ \ 0\ \ 62.53]^{\text{T}} \\ \mathbf{d}^* = [0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 1\ \ 0\ \ 0\ \ 0\ \ 0\ \ 887.6\ \ 0\ \ 0\ 440.6\ \ 0\ \ 1085.3\ \ 0\ \ 0\ \ 244.0\ \ 0\ \ 0]^{\text{T}} \\ z^*(\lambda_0 = 10, \lambda = 10) = 2657.5 \end{array}\right\} \qquad (11)$$

with the obvious correspondence (see (10))

$$\mathbf{v}^* = [m^*\ \theta_1^*\ \theta_2^* \ldots\ldots\ldots\ldots\ \theta_{12}^*\ \ b^*_1\ b^*_2]^{\text{T}}$$

which assigns the following solutions:

$$m^* = v^*_1 = 42.65,\ \ \theta_2^* = 81.87,\ \ \theta_6^* = 90.57,\ \ \theta_9^* = 89.04,\ b^*_1 = 0,\ b^*_2 = 62.53.$$

The primal solution says that the optimal basis $B_1$* is constituted by the non-zero components of **v***, i.e.:

$B_1$* = (1st, 3rd, 7th, 10th, 15th components)

The dual solution **d*** says that at optimality the following set of constraints is active, represented as $AC_1$*:

$AC_1$* = (9th, 14th, 17th, 19th, 22nd constraints).

Following [5, p. 160], we now generate the initial tableau of the simplex method, augmented by an extra row (the 26th) containing the vector $\mathbf{q}^{\text{T}}$ of (8), labelled by **z0** and corresponding to the decrement vector $\mathbf{q}^{\text{T}}$ = [0 -1 .... -1 0 0] in (10). This sequence of commands generates the initial augmented tableau T (see Table 2):

T = totbl (**A**, **b**, **p**);
T = addrow (T, [**q**' 0], '**z0**', 26],

where **A**, **b** and **p** comply with (7) and (10). In particular, **A** (24 x 15) represents the matrix of the constraint system, the vector **b** (24 x 1) is the right hand term of the constraint system (**b** = $[0\ \ldots\ldots\ 0\ 1\ \ldots\ldots\ 1]^{\text{T}}$), and **p** (15 x 1) is the vector of the cost function present in (10) (**p** = $[1\ \ \lambda_0\ \lambda_0 \ldots\ldots\ldots\ \lambda_0\ 0\ \ 0]^{\text{T}}$, with $\lambda_0$ = 10); moreover, **q**' is equivalent to $\mathbf{q}^{\text{T}}$ in Matlab language and represents as already said the decrement cost row vector (**q**' = [0 -1 -1 ...... -1 0 0]).

The initial deriving arrangement (see Table 2) is organised in such a way that the original problem variables $v_i$, i = 1, 2, ..., 15, are nonbasic (independent) variables, and the slack variables $s_i$, i = 1, 2, ...., 24, are basic (dependent) variables; moreover, $\mathbf{p}^{\text{T}}$ corresponds to the row labelled by **zf**, $\mathbf{q}^{\text{T}}$ to the row labelled by **z0**, and the vector – **b** is on the last column. To obtain the final optimal tableau, we have to force the basis $B_1$* = (1st, 3rd, 7th, 10th, 15th components) to enter the set of basic variables (at optimality they are to be non-zero), and $AC_1$* = (9th, 14th, 17th, 19th, 22nd constraints) to enter the set of nonbasic variables (at optimality they are to be zero): e.g., $s_9$ = 0 means that the 9th constraint becomes active.

This can be done by the succession of commands:

T = ljx (T, 9, 1);
T = ljx (T, 14, 3);
T = ljx (T, 17, 7);
T = ljx (T, 19, 10);
T = ljx (T, 22, 15).

The first command T = ljx (T, 9, 1) performs a Jordan exchange with pivot (9,1), which forces the slack $s_9$ to enter the set of nonbasic variables and the variable $v_1$ to enter the set of basic variables; and so on.

**Table 2:** The Initial Augmented Tableau

| | v1 | v2 | v3 | v4 | v5 | v6 | v7 | v8 | v9 | v10 | v11 | v12 | v13 | v14 | v15 | **-b** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| s1 | 1.00 | -1.00 | -0.87 | -0.89 | -0.96 | -0.62 | -0.74 | 0.76 | 0.66 | 0.55 | 0.87 | 0.79 | 0.84 | -1.00 | 1.00 | 0.00 |
| s2 | 1.00 | -0.87 | -1.00 | -0.89 | -0.93 | -0.76 | -0.62 | 0.93 | 0.93 | 0.84 | 0.84 | 0.99 | 0.98 | -1.00 | 1.00 | 0.00 |
| s3 | 1.00 | -0.89 | -0.89 | -1.00 | -0.98 | -0.89 | -0.89 | 0.93 | 0.75 | 0.70 | 0.67 | 0.86 | 0.80 | -1.00 | 1.00 | 0.00 |
| s4 | 1.00 | -0.96 | -0.93 | -0.98 | -1.00 | -0.80 | -0.82 | 0.90 | 0.77 | 0.69 | 0.79 | 0.89 | 0.87 | -1.00 | 1.00 | 0.00 |
| s5 | 1.00 | -0.62 | -0.76 | -0.89 | -0.80 | -1.00 | -0.79 | 0.93 | 0.74 | 0.78 | 0.43 | 0.78 | 0.65 | -1.00 | 1.00 | 0.00 |
| s6 | 1.00 | -0.74 | -0.62 | -0.89 | -0.82 | -0.79 | -1.00 | 0.72 | 0.47 | 0.44 | 0.44 | 0.58 | 0.52 | -1.00 | 1.00 | 0.00 |
| s7 | 1.00 | 0.76 | 0.93 | 0.93 | 0.90 | 0.93 | 0.72 | -1.00 | -0.90 | -0.89 | -0.63 | -0.95 | -0.85 | 1.00 | -1.00 | 0.00 |
| s8 | 1.00 | 0.66 | 0.93 | 0.75 | 0.77 | 0.74 | 0.47 | -0.90 | -1.00 | -0.97 | -0.69 | -0.97 | -0.92 | 1.00 | -1.00 | 0.00 |
| s9 | 1.00 | 0.55 | 0.84 | 0.70 | 0.69 | 0.78 | 0.44 | -0.89 | -0.97 | -1.00 | -0.53 | -0.91 | -0.81 | 1.00 | -1.00 | 0.00 |
| s10 | 1.00 | 0.87 | 0.84 | 0.67 | 0.79 | 0.43 | 0.44 | -0.63 | -0.69 | -0.53 | -1.00 | -0.77 | -0.89 | 1.00 | -1.00 | 0.00 |
| s11 | 1.00 | 0.79 | 0.99 | 0.86 | 0.89 | 0.78 | 0.58 | -0.95 | -0.97 | -0.91 | -0.77 | -1.00 | -0.97 | 1.00 | -1.00 | 0.00 |
| s12 | 1.00 | 0.84 | 0.98 | 0.80 | 0.87 | 0.65 | 0.52 | -0.85 | -0.92 | -0.81 | -0.89 | -0.97 | -1.00 | 1.00 | -1.00 | 0.00 |
| s13 | 0.00 | 1.00 | 0.87 | 0.89 | 0.96 | 0.62 | 0.74 | -0.76 | -0.66 | -0.55 | -0.87 | -0.79 | -0.84 | 1.00 | -1.00 | -1.00 |
| s14 | 0.00 | 0.87 | 1.00 | 0.89 | 0.93 | 0.76 | 0.62 | -0.93 | -0.93 | -0.84 | -0.84 | -0.99 | -0.98 | 1.00 | -1.00 | -1.00 |
| s15 | 0.00 | 0.89 | 0.89 | 1.00 | 0.98 | 0.89 | 0.89 | -0.93 | -0.75 | -0.70 | -0.67 | -0.86 | -0.80 | 1.00 | -1.00 | -1.00 |
| s16 | 0.00 | 0.96 | 0.93 | 0.98 | 1.00 | 0.80 | 0.82 | -0.90 | -0.77 | -0.69 | -0.79 | -0.89 | -0.87 | 1.00 | -1.00 | -1.00 |
| s17 | 0.00 | 0.62 | 0.76 | 0.89 | 0.80 | 1.00 | 0.79 | -0.93 | -0.74 | -0.78 | -0.43 | -0.78 | -0.65 | 1.00 | -1.00 | -1.00 |
| s18 | 0.00 | 0.74 | 0.62 | 0.89 | 0.82 | 0.79 | 1.00 | -0.72 | -0.47 | -0.44 | -0.44 | -0.58 | -0.52 | 1.00 | -1.00 | -1.00 |
| s19 | 0.00 | -0.76 | -0.93 | -0.93 | -0.90 | -0.93 | -0.72 | 1.00 | 0.90 | 0.89 | 0.63 | 0.95 | 0.85 | -1.00 | 1.00 | -1.00 |
| s20 | 0.00 | -0.66 | -0.93 | -0.75 | -0.77 | -0.74 | -0.47 | 0.90 | 1.00 | 0.97 | 0.69 | 0.97 | 0.92 | -1.00 | 1.00 | -1.00 |
| s21 | 0.00 | -0.55 | -0.84 | -0.70 | -0.69 | -0.78 | -0.44 | 0.89 | 0.97 | 1.00 | 0.53 | 0.91 | 0.81 | -1.00 | 1.00 | -1.00 |
| s22 | 0.00 | -0.87 | -0.84 | -0.67 | -0.79 | -0.43 | -0.44 | 0.63 | 0.69 | 0.53 | 1.00 | 0.77 | 0.89 | -1.00 | 1.00 | -1.00 |
| s23 | 0.00 | -0.79 | -0.99 | -0.86 | -0.89 | -0.78 | -0.58 | 0.95 | 0.97 | 0.91 | 0.77 | 1.00 | 0.97 | -1.00 | 1.00 | -1.00 |
| s24 | 0.00 | -0.84 | -0.98 | -0.80 | -0.87 | -0.65 | -0.52 | 0.85 | 0.92 | 0.81 | 0.89 | 0.97 | 1.00 | -1.00 | 1.00 | -1.00 |
| **zf** | 1.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 0.00 | 0.00 | 0.00 |
| **z0** | 0.00 | -1.00 | -1.00 | -1.00 | -1.00 | -1.00 | -1.00 | -1.00 | -1.00 | -1.00 | -1.00 | -1.00 | -1.00 | 0.00 | 0.00 | 0.00 |

The final tableau is shown in Table 3 and is optimal because the row vector of reduced costs, labelled by **zf**, and its last column are both positive in all their components (see §2); moreover, the yellow highlighted rows have in their last component the optimal values of **v*** already presented in (11), the other values of **v*** being zero because pertaining to the set of nonbasic variables:

$v_1^* = m^* = 42.65,\ v_3^* = 81.87,\ v_7^* = 90.57,\ v_{10}^* = 89.04,\ v_{15}^* = 62.53.$

Finally, the blue highlighted value of **zf** in Table 3 confirms the optimal value of the cost function, see (11):

$z^*(\lambda_0=10, \lambda=10) = 2657.5.$

The parametric reduced cost of the tableau is furnished instead by the following linear combination [5, p. 160]:

$\mathbf{PRC}(\lambda_0=10, \lambda) = \mathbf{zf} + (\lambda_0 - \lambda)\,\mathbf{z0} \quad = \mathbf{zf} + (10 - \lambda)\,\mathbf{z0} \qquad (0 \le \lambda \le 10)$

which is to be examined one component at a time (except for the last blue highlighted component of Table 3) to obtain the first condition of negativity, which identifies the nearest breakpoint with varying λ. This is equivalent to finding the smallest positive value of the decrement (10 - λ) causing the condition $\mathbf{PRC}(\lambda_0=10, \lambda) \le 0$.

In the case of Table 3, this happens in correspondence to the eighth component of $\mathbf{PRC}(\lambda_0=10, \lambda)$ (green highlighted in Table 3), which with a more complete representation by 5 decimal digits results

$\mathbf{PRC}(\lambda_0=10, \lambda_8) = 6.46027 - (\lambda_0 - \lambda_8) \bullet 0.65397 = 6.46027 - (10 - \lambda_8) \bullet 0.65397 \le 0,$

and is equivalent to the condition

$$\lambda_8 \le 0.12146\ . \qquad (12)$$

This result fixes the first breakpoint $\lambda_{b1}$ to the left of $\lambda_0 = 10$ to the greatest admissible value of $\lambda_8$:

$\lambda_{b1} = \lambda_8^{\max} = 0.12146,$

and fully agrees with Figure 2, where the value λ = 0.1215 there presented was found through a procedure of repeated solutions with a step size Δλ = 0.001.

Another noticeable result is that in the right-open interval [0.12146, 10) of variation of λ, the optimal **v*** does not vary (see §2), whereas the optimal cost z*(λ) obeys the following linear variation rule:

$z^*(\lambda) = 2657.47 - (10 - \lambda) \bullet 261.48 = 261.48 \bullet \lambda + 42.67,$

deriving from the blue highlighted components (represented by 5 decimal digits) of the last column of Table 3 [5, p. 160] and confirmed in Figure 2.

To proceed further with the second breakpoint towards $0^+$, we repeat the above procedure for a different starting value of $\lambda_0$, say $\lambda_0 = 0.12$, immediately to the left of $\lambda_{b1} = 0.12146$.

The Matlab LP solver (*linprog*) furnishes for the values $\lambda_0 = 0.12$ and $\lambda = 0.12$ the following optimal basis $B_2$* and set $AC_2$* of active constraints:

$B_2^*$ = (1st, 3rd, 7th, 8th, 10th, 15th components);

$AC_2^*$ = (6th, 9th, 14th, 17th, 19th, 22nd constraints).

The variable $v_8$ now enters the optimal basis $B_2$*, and the 6th constraint enters the set $AC_2$*.

After a series of commands dedicated to performing proper Jordan exchanges which comply with $B_2$* and $AC_2$*, we obtain the results of Table 4; for the sake of brevity only the two last rows of the optimal final tableau are shown.

**Table 3:** The Final Tableau ($\lambda_0 = 10$, $\lambda = 10$)

| | s9 | v2 | s14 | v4 | v5 | v6 | s17 | v8 | v9 | s19 | v11 | v12 | v13 | v14 | s22 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| s1 | 1.00 | -0.01 | 4.61 | 0.04 | 0.01 | 0.12 | 2.14 | -0.07 | 0.03 | 6.71 | -0.04 | 0.00 | 0.02 | 0.00 | 2.04 | 15.49 |
| s2 | 1.00 | 0.17 | 12.55 | 0.10 | 0.11 | 0.07 | 7.27 | -0.08 | 0.08 | 17.79 | -0.13 | 0.02 | 0.07 | 0.00 | 4.03 | 41.65 |
| s3 | 1.00 | 0.03 | 4.35 | 0.05 | 0.03 | 0.11 | 1.85 | -0.06 | 0.04 | 6.72 | -0.04 | 0.01 | 0.03 | 0.00 | 1.49 | 14.41 |
| s4 | 1.00 | 0.03 | 4.68 | 0.04 | 0.03 | 0.10 | 2.39 | -0.06 | 0.03 | 7.34 | -0.04 | 0.00 | 0.03 | 0.00 | 1.73 | 16.14 |
| s5 | 1.00 | 0.17 | 13.55 | 0.10 | 0.11 | 0.07 | 6.27 | -0.08 | 0.08 | 17.79 | -0.13 | 0.02 | 0.07 | 0.00 | 4.03 | 41.65 |
| s6 | 1.00 | -0.01 | 1.65 | 0.07 | 0.03 | 0.24 | -1.15 | -0.11 | 0.05 | 1.65 | -0.02 | 0.00 | 0.04 | 0.00 | 0.85 | 3.00 |
| s7 | 1.00 | 0.17 | 13.55 | 0.10 | 0.11 | 0.07 | 7.27 | -0.08 | 0.08 | 16.79 | -0.13 | 0.02 | 0.07 | 0.00 | 4.03 | 41.65 |
| s8 | 1.00 | 0.03 | 4.33 | 0.02 | 0.02 | 0.02 | 1.90 | -0.02 | 0.02 | 5.07 | -0.03 | 0.00 | 0.01 | 0.00 | 1.16 | 12.46 |
| v1 | 1.00 | 0.17 | 13.55 | 0.10 | 0.11 | 0.07 | 7.27 | -0.08 | 0.08 | 17.79 | -0.13 | 0.02 | 0.07 | 0.00 | 4.03 | 42.65 |
| s10 | 1.00 | 0.17 | 13.55 | 0.10 | 0.11 | 0.07 | 7.27 | -0.08 | 0.08 | 17.79 | -0.13 | 0.02 | 0.07 | 0.00 | 3.03 | 41.65 |
| s11 | 1.00 | 0.12 | 10.99 | 0.07 | 0.08 | 0.06 | 5.36 | -0.06 | 0.06 | 13.26 | -0.09 | 0.02 | 0.05 | 0.00 | 3.09 | 32.69 |
| s12 | 1.00 | 0.12 | 11.76 | 0.07 | 0.07 | 0.07 | 5.70 | -0.06 | 0.06 | 14.42 | -0.10 | 0.02 | 0.05 | 0.00 | 3.03 | 34.92 |
| s13 | 0.00 | 0.18 | 8.95 | 0.06 | 0.09 | -0.05 | 5.13 | -0.01 | 0.05 | 11.08 | -0.09 | 0.02 | 0.05 | 0.00 | 2.00 | 26.16 |
| v3 | -0.0 | -0.90 | 30.03 | -0.27 | -0.67 | 0.65 | 10.91 | 0.03 | 0.69 | 32.99 | 1.32 | 0.81 | 1.35 | 0.00 | 7.95 | 81.87 |
| s15 | 0.00 | 0.14 | 9.20 | 0.05 | 0.08 | -0.04 | 5.42 | -0.02 | 0.04 | 11.07 | -0.09 | 0.02 | 0.04 | 0.00 | 2.55 | 27.24 |
| s16 | 0.00 | 0.15 | 8.87 | 0.05 | 0.08 | -0.03 | 4.89 | -0.02 | 0.05 | 10.45 | -0.08 | 0.02 | 0.04 | 0.00 | 2.31 | 25.52 |
| v7 | -0.0 | 0.26 | 28.43 | -0.50 | -0.18 | -0.84 | 16.86 | 0.36 | 0.01 | 35.72 | -0.80 | 0.06 | -0.15 | 0.00 | 9.57 | 90.57 |
| s18 | 0.00 | 0.18 | 11.90 | 0.03 | 0.08 | -0.17 | 8.43 | 0.03 | 0.03 | 16.14 | -0.11 | 0.02 | 0.03 | 0.00 | 3.18 | 38.65 |
| v10 | -0.0 | -0.30 | 28.95 | 0.27 | 0.02 | 0.90 | 15.57 | -0.74 | -0.44 | 38.05 | 0.72 | -0.25 | 0.32 | 0.00 | 6.48 | 89.04 |
| s20 | 0.00 | 0.14 | 9.22 | 0.08 | 0.09 | 0.05 | 5.37 | -0.06 | 0.07 | 12.72 | -0.10 | 0.02 | 0.05 | 0.00 | 2.87 | 29.19 |
| s21 | 0.00 | 0.17 | 13.55 | 0.10 | 0.11 | 0.07 | 7.27 | -0.08 | 0.08 | 17.79 | -0.13 | 0.02 | 0.07 | 0.00 | 4.03 | 41.65 |
| v15 | 0.00 | 0.39 | 22.43 | 0.08 | 0.13 | 0.13 | 8.34 | -0.05 | 0.13 | 23.29 | -0.62 | 0.07 | 0.00 | 1.00 | 8.48 | 62.53 |
| s23 | 0.00 | 0.05 | 2.57 | 0.03 | 0.03 | 0.01 | 1.92 | -0.02 | 0.02 | 4.54 | -0.03 | 0.01 | 0.02 | 0.00 | 0.94 | 8.96 |
| s24 | 0.00 | 0.06 | 1.79 | 0.02 | 0.03 | 0.00 | 1.57 | -0.01 | 0.02 | 3.37 | -0.03 | 0.01 | 0.02 | 0.00 | 1.00 | 6.74 |
| **zf** | 1.00 | 0.42 | 887.6 | 5.09 | 1.81 | 17.20 | 440.6 | 6.46 | 12.74 | 1085. | 22.37 | 16.25 | 25.17 | 0.00 | 243.9 | 2657.5 |
| **z0** | 0.00 | -0.02 | -87.41 | -0.50 | -0.17 | -1.71 | -43.34 | -0.65 | -1.27 | -106.7 | -2.25 | -1.62 | -2.51 | 0.00 | -24.0 | -261.5 |

The parametric reduced cost of this tableau is furnished by

$$\mathbf{PRC}(\lambda_0 = 0.12, \lambda) = \mathbf{zf} + (\lambda_0 - \lambda)\,\mathbf{z0} = \mathbf{zf} + (0.12 - \lambda)\,\mathbf{z0} \qquad (0 \le \lambda \le 0.12)$$

which, when analysed with regard to the negativity of its 11th column (green highlighted in Table 4 and represented by 5 decimal digits), gives

$$\mathbf{PRC}(\lambda_0 = 0.12, \lambda_{11}) = 0.14351 - (0.12 - \lambda_{11}) \bullet 2.16045 \le 0, \text{ that is,}$$

$$(0.12 - \lambda_{11}) \geq 0.06643, \text{ and } \lambda_{11} \le 0.05357.$$

**Table 4:** The 2 Last Rows of the Final Tableau ($\lambda_0 = 0.12$, $\lambda = 0.12$)

| | s6 | v2 | s9 | v4 | v5 | v6 | s14 | s17 | v9 | s19 | v11 | v12 | v13 | v14 | s22 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **zf** | 0.01 | 0.18 | 0.99 | 0.16 | 0.13 | 0.27 | 24.03 | 12.48 | 0.23 | 30.59 | 0.14 | 0.22 | 0.37 | 0.00 | 6.90 | 74.01 |
| **z0** | 5.73 | 0.03 | -5.73 | -0.90 | -0.33 | -3.11 | -96.87 | -36.73 | -1.53 | -116.20 | -2.16 | -1.64 | -2.74 | 0.00 | -28.85 | -278.66 |

The 11th column identifies indeed the smallest positive decrement $(0.12 - \lambda)$ producing $\mathbf{PRC}(\lambda_0 = 0.12, \lambda) \le 0$.
This result assigns a correspondence between the greatest admissible value of $\lambda_{11}$ and the second breakpoint $\lambda_{b2}$:

$$\lambda_{b2} = \lambda_{11}^{\max} = 0.05357,$$

once again in agreement with Figure 2.
In the right-open interval [0.05357, 0.12) of $\lambda$ the optimal cost $z^*(\lambda)$ presents the following linear variation rule:

$$z^*(\lambda) = 74.01 - (0.12 - \lambda) \bullet 278.66 = 278.66 \bullet \lambda + 40.57$$

(derived from the last blue highlighted column of Table 4) [5, p. 160], whereas the components of the optimal solution $\mathbf{v}^*$ maintain a constant value.
Subsequent breakpoints towards $0^+$ can be identified by repeating the illustrated procedure; this way SA demonstrates its capabilities to explore the entire range of $\lambda$ following a computationally convenient procedure.

## 4.2. The case of LP classifiers with two parameters (λ and C)

The case of SA with two free parameters is difficult to treat analytically. Nevertheless, this situation can be faced with repeated analyses, which fix one parameter (for example $\lambda$) to a value and define through SA the variations consequent to the second parameter C; by repeating SA analysis on different values of $\lambda$, we can obtain a grid of breakpoints on the plane ($\lambda$, C) which helps modellers to outline a general conformation.
Following (5), the problem in its decremental representation is now:

$$\left.\begin{array}{l} \text{minimise } z(\mathbf{v}, \lambda, C)_{\mathrm{v}} = [\mathbf{p}^{\mathrm{T}} + (C_0 - C)\,\mathbf{q}^{\mathrm{T}}]\,[m\ \theta_1\ \theta_2\ \ldots\ldots\ \theta_{12}\ b_1\ b_2\ \xi_1\ \ldots\ldots\ \xi_{12}]^{\mathrm{T}} \triangleq \\ \triangleq [\mathbf{p}^{\mathrm{T}} + (C_0 - C)\,\mathbf{q}^{\mathrm{T}}]\,\mathbf{v}, \text{ subject to} \\ m - c_i\, D\,(\mathbf{x}_i) \geq 0 \\ c_i\, D\,(\mathbf{x}_i) + \xi_i \geq 1 \\ \theta_i \geq 0,\ \xi_i \geq 0,\ \lambda \geq 0,\ C_0 \geq 0,\ 0 \le C \le C_0,\ x_i \in R^2, \quad i = 1, 2, \ldots.., 12 \end{array}\right\} \qquad (13)$$

where

$\mathbf{p}^T + (C_0 - C)\,\mathbf{q}^T = [1\ \ \lambda\ \ldots\ldots\ \lambda\ \ 0\ \ 0\ \ C_0\ \ldots\ldots\ C_0] + (C_0 - C)\,[0\ \ 0\ \ldots\ldots 0\ \ 0\ \ 0\ \ {-1}\ \ {-1}\ \ldots\ldots\ {-1}]$

Figure 3 shows the “brute force” LP solution to about 1000 problems of the type (13), with $C_0$ = 50, C varying in the range [0.1, 50] with a step size ΔC = 0.05 and λ fixed to the value 0.12, in terms of the optimal cost z* and of the optimal $v_1$* = m*; piecewise-linear and stepwise behaviours are confirmed in Fig. 3.

The decremental representation associated to Fig. 3 is:

$\mathbf{p}^T + (C_0 - C)\,\mathbf{q}^T = [1\ 0.12\ \ldots\ldots\ 0.12\ \ 0\ \ 0\ \ 50\ldots\ldots\ 50] + (50 - C)\,[0\ \ 0\ \ldots\ldots 0\ \ 0\ \ 0\ \ {-1}\ {-1}\ \ldots\ldots\ {-1}]$

The Matlab LP solver (*linprog*) furnishes for the value $C_0$ = 50 and C = 50 the following optimal basis $B_3$* and the following set $AC_3$* of active constraints:

$B_3$* = (1st, 3rd, 7th, 8th, 10th, 15th components);

$AC_3$* = (6th, 9th, 14th, 17th, 19th, 22nd constraints).

The results of Table 5 are then obtained, once the right succession of Jordan exchanges which comply with $B_3$* and $AC_3$* are performed; Table 5 shows only the last two rows of the final tableau, composed in this case of 28 elements.

The parametric reduced cost of this tableau is furnished by

$\mathbf{PRC}(\lambda = 0.12, C_0 = 50, C) = \mathbf{zf} + (C_0 - C)\,\mathbf{z0} = \mathbf{zf} + (50 - C)\,\mathbf{z0} \qquad (0 \le C \le 50)$

**Table 5:** The 2 Last Rows of the Final Tableau ($\lambda_0$ = 0.12, $C_0$ = 50, C = 50; the Set of Nonbasic Variables Is Not Reported)

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **zf** | 0.01 | 0.18 | 0.99 | 0.16 | 0.13 | 0.27 | 24.03 | 12.48 | 0.23 | 30.59 | 0.14 | 0.22 | 0.37 | 0.00 | 6.90 | 50.00 |
| (continued from previous row) | | | | | 25.97 | 50.00 | 50.00 | 37.52 | 50.00 | 19.41 | 50.00 | 50.00 | 43.10 | 50.00 | 50.00 | 74.01 |
| **z0** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | -0.00 | -0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | -1.00 |
| (continued from previous row) | | | | | -1.00 | -1.00 | -1.00 | -1.00 | -1.00 | -1.00 | -1.00 | -1.00 | -1.00 | -1.00 | -1.00 | 0.00 |

which, when analysed with regard to the negativity of its 22nd column (green highlighted in Table 5 and represented by 5 decimal digits), gives

$\mathbf{PRC}(\lambda = 0.12, C_0 = 50, C_{22}) = 19.4112 - (50 - C_{22}) \bullet 1 \le 0$, that is,

$(50 - C_{22}) \ge 19.4112$, and $C_{22} \le 30.5888$.

The 22nd column yields indeed the smallest positive decrement (50 − C) producing $\mathbf{PRC}(\lambda_0 = 0.12, C_0 = 50, C) \le 0$

The greatest admissible value for $C_{22}$ assigns this value to the first breakpoint $\eta_1$:

$\eta_1 = C_{22}^{max} = 30.5888$,

resulting in agreement with the situation of Fig. 3, obtained via repeated solutions to the LP problem (13).

In the right-open interval [30.5888, 50) of C the optimal cost z*(λ = 0.12, C) presents the following linear variation rule (derived from the last blue highlighted column of Table 5) [5, p. 160]:

$z^*(\lambda = 0.12, C) = 74.01 + (50 - C) \bullet 0 = 74.01$,

in perfect agreement with the top value of z* in Fig. 3.

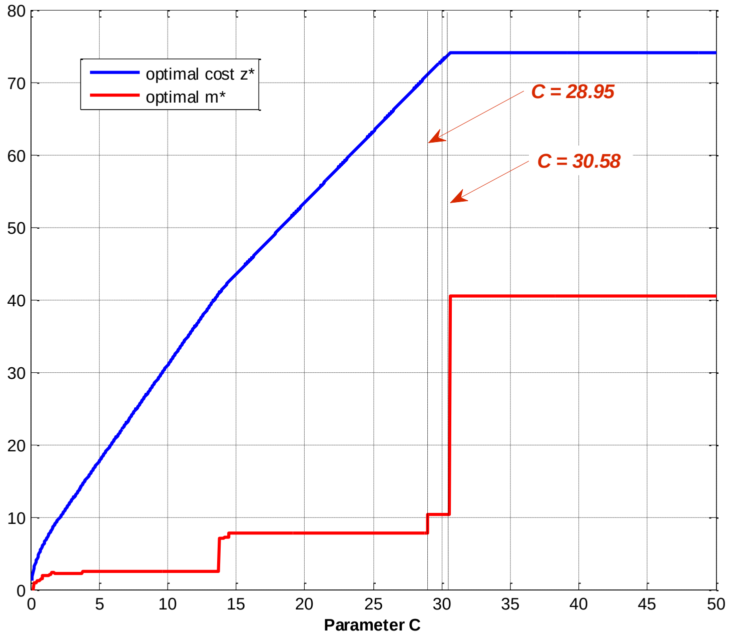

**Fig. 3:** Variations of Z* and *M** As A Function of the Parameter C (λ = 0.12, $\gamma_{RBF}$ = 0.6)

As a last step, we continue further our analysis by calculating the second breakpoint $\eta_2$ towards $0^+$, as done in §4.1.

We fix $C_0$ and C to a value immediately to the left of $\eta_1$= 30.5888, say $C_0$ = C = 30, and solve the consequent problem (13) through the Matlab *linprog* routine, obtaining:

$B_4$* = (1st, 3rd, 7th, 10th, 15th, 22nd components);

$AC_4^*$ = (6th, 9th, 14th, 17th, 19th, 22nd constraints).

The variable $v_8$ of $B_3^*$ leaves now the optimal basis $B_4^*$, the variable $v_{10}$ enters in its stead. The set $AC_4^*$ of active constraints at optimality remains unaltered.

The succession of Jordan exchanges which comply with $B_4^*$ and $AC_4^*$ yields the results of Table 6, which shows only the last two rows of the final tableau, composed of 28 elements.

The parametric reduced cost of this tableau is furnished by

$$\mathbf{PRC}(\lambda = 0.12, C_0 = 30, C) = \mathbf{zf} + (C_0 - C)\,\mathbf{z0} = \mathbf{zf} + (30 - C)\,\mathbf{z0} \qquad (0 \le C \le 30)$$

The analysis of **PRC**(C) with regard to the negativity of its 3rd column (green highlighted in Table 6 and represented by 5 decimal digits), gives

$$\mathbf{PRC}(\lambda = 0.12, C_0 = 30, C_3) = 0.63374 - (30 - C_3) \bullet 0.60569 \le 0, \text{ that is,}$$

$$(30 - C_3) \geq 1.04631, \text{ and } C_3 \le 28.95369.$$

The 3rd column yields indeed the smallest positive decrement (30 − C) producing $\mathbf{PRC}(\lambda_0 = 0.12, C_0 = 30, C) \le 0$.

**Table 6:** The 2 Last Rows of the Final Tableau ($\lambda$ = 0.12, $C_0$ = 30, C = 30; the Set of Nonbasic Variables Is Not Reported)

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **zf** | 0.37 | 0.18 | 0.63 | 0.13 | 0.12 | 0.19 | 23.44 | 0.04 | 0.22 | 12.90 | 0.15 | 0.22 | 0.35 | 0.00 | 30.00 | 30.00 |
| | (continued from previous row) | | | | | 30.00 | 30.00 | 17.10 | 30.00 | 6.60 | 30.00 | 30.00 | 23.40 | 30.00 | 30.00 | 72.93 |
| **z0** | 0.61 | 0.01 | -0.61 | -0.04 | -0.02 | -0.15 | -1.00 | 0.07 | -0.03 | 0.70 | 0.01 | -0.00 | -0.02 | 0.00 | -1.00 | -1.00 |
| | (continued from previous row) | | | | | -1.00 | -1.00 | -1.70 | -1.00 | -0.51 | -1.00 | -1.00 | -0.49 | -1.00 | -1.00 | -1.82 |

This result fixes the breakpoint $\eta_2$ immediately to the left of $C_0$ = 30 to the greatest admissible value of $C_3$:

$$\eta_2 = C_3^{max} = 28.95369,$$

once again confirmed in Fig. 3.

In the right-open interval [28.95369, 30) of C the optimal cost $z^*(\lambda = 0.12, C)$ presents the following linear variation rule, as a result produced by the last blue highlighted terms of Table 6 [5, p. 160]:

$$z^*(\lambda = 0.12, C) = 72.93 - (30 - C) \bullet 1.82 = 1.82 \bullet C + 18.33.$$

By repeating this scheme, the analysis can be pushed forward as far as we want; this way SA demonstrates a capability of analysis over the plane ($\lambda$, C) with computationally efficient procedures.

## 5. Conclusion

Techniques of optimal SVM classification implemented through linear programming offer the advantages of greater analytical tractability than classic SVM techniques inspired by quadratic programming.

The availability in the LP context of well-established and viable SA procedures is an important advantage, because it makes possible the implementation of parametric programming, which mainly concerns how solutions are affected by changes in the data of the problem and how characteristic situations of optimality (e.g., breakpoints) can be defined.

From a computational point of view this is of paramount importance for large scale problems, to avoid repeated blind searches for solutions, restarting each time from scratch when we try to determine satisfactory values for the free parameters.

In this paper we demonstrated how SA in effect can help modellers to build a representation on the line $\lambda$, in the case of one free parameter, or on the plane ($\lambda$, C), in the case of two free parameters, of the cost z* and of the solution **v*** (which defines the optimal classifier coefficients), identifying at the same time the breakpoints which separate different solutions; therefore, the modeller can be helped in the subsequent decisions by the knowledge that in the intervals between two consecutive breakpoints the solution is invariant, and that to each solution corresponds an evaluable degree of complexity (i.e., a known number of support vectors).

In this paper SA was presented as a practical analysis developed around a simple example of two random classes, but the general lines of this procedure can be extrapolated to large problems as well; in any case, it always demonstrated a good agreement with the results coming from the comparing procedures of “brute force” solutions.

## References

[1] 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

[2] N. Cristianini, J. Shawe-Taylor, An Introduction to Support Vector Machines, Cambridge University Press, 2000

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

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

[5] M. C. Ferris, O. L. Mangasarian, S. J. Wright, Linear Programming with MATLAB, MPS-SIAM, Philadelphia, 2007

[6] J.P. Pedroso, N. Murata, Support Vector Machines for Linear Programming: Motivations and Formulations, BSIS Tech. Report No. 99-XXX, August 1999

[7] O.L. Mangasarian, Exact 1-Norm Support Vector Machines via Unconstrained Convex Differentiable Minimization, Journal of Machine Learning Research, 2006, vol. 7, pp. 1517–1530

[8] J. Zhu, S. Rosset, T. Hastie, and R. Tibshirani. 1-norm Support Vector Machines, Advances in Neural Information Processing Systems, 2004, vol. 16

[9] R. Ragona, A Minimax Chebyshev Approach to Optimal Binary Classification, Int. Journal of Applied Mathematical Research, 2013, vol. 2, No. 2, pp. 175-187

[10] R. Ragona, A Linear Programming Solution to Data Description and Novelty Classification, Int. Journal of Applied Mathematical Research, 2013, vol. 2, No. 4, pp. 495-504

[11] T. Joachims, Making large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning, B. Schölkopf and C. Burges and A. Smola (ed.), MIT-Press, 1999.

[12] C.-C. Chang and C.-J. Lin. LIBSVM: a Library for Support Vector Machines. ACM Transactions on Intelligent Systems and Technology, 2011, vol. 2, No. 3

[13] P. S. Bradley and O. L. Mangasarian. Massive Data Discrimination via Linear Support Vector Machines. Optimization Methods and Software, 2000, vol. 13, No. 1

[14] S. Sra, Efficient Large Scale Linear Programming Support Vector Machines, ECML 2006, Proc. of the 17th European Conference on Machine Learning, Springer-Verlag, Berlin, 2006, pp. 767-774

[15] R. Tibshirani, Regression Shrinkage and Selection via the LASSO. Journal of the Royal Statistical Society, 1996, Series B vol. 58, No. 1

[16] D.G. Luenberger, Linear and Nonlinear Programming, second edition, Stanford University, 1984