4OR-Q J Oper Res
https://doi.org/10.1007/s10288-018-0378-2

INVITED SURVEY

Nonlinear optimization and support vector machines

Veronica Piccialli[1] · **Marco Sciandrone**[2]

Received: 22 February 2018 / Revised: 25 April 2018

Abstract Support Vector Machine (SVM) is one of the most important class of machine learning models and algorithms, and has been successfully applied in various fields. Nonlinear optimization plays a crucial role in SVM methodology, both in defining the machine learning models and in designing convergent and efficient algorithms for large-scale training problems. In this paper we present the convex programming problems underlying SVM focusing on supervised binary classification. We analyze the most important and used optimization methods for SVM training problems, and we discuss how the properties of these problems can be incorporated in designing useful algorithms.

Keywords Statistical learning theory · Support vector machine · Convex quadratic programming · Wolfe's dual theory · Kernel functions · Nonlinear optimization methods

Mathematics Subject Classification 65K05 Mathematical programming methods · 90C25 Convex programming · 90C30 Nonlinear programming

✉ Marco Sciandrone
marco.sciandrone@unifi.it

Veronica Piccialli
veronica.piccialli@uniroma2.it

[1] Dipartimento di Ingegneria Civile e Ingegneria Informatica, Università degli Studi di Roma "Tor Vergata", via del Politecnico, 100133 Rome, Italy

[2] Dipartimento di Ingegneria dell'Informazione, Università di Firenze, via di Santa Marta 3, 50139 Firenze, Italy

Published online: 23 May 2018

1 Introduction

The Support Vector Machine (SVM) is widely used as a simple and efficient tool for linear and nonlinear classification as well as for regression problems. The basic training principle of SVM, motivated by statistical learning theory Vapnik (1998), is that the expected classification error for unseen test samples is minimized, so that, SVMs define good predictive models.

In this paper we focus on *supervised* (linear and nonlinear) binary SVM classifiers, whose task is to classify objects (patterns) into two groups using the features describing the objects and a labelled dataset (the *training set*). We will not enter into the details of statistical issues concerning SVM models, nor we will analyze the standard cross-validation techniques used for adjusting SVM hyperparameters in order to optimize the predictive performance as machine learning models. A suitable analysis of statistical and machine learning issues can be found, for instance, in Bishop (2006), Scholkopf and Smola (2001), Shawe-Taylor and Cristianini (2004). Here we will limit our analysis to theoretical, algorithmic and computational issues related to the optimization problem underlying the training of SVMs.

SVM training requires solving (large-scale) convex programming problems, whose difficulties are mainly related to the possibly huge number of training instances, that leads to a huge number of either variables or constraints. The particular structure of the SVM training problems has favored the design and the development of ad hoc optimization algorithms to solve large-scale problems. Thanks to the convexity of the constrained problem, optimization algorithms for SVM are required to quickly converge towards any minimum. Thus the requirements are well-defined from an optimization point of view, and this has motivated a wide research activity (even of the optimization community) to define efficient and convergent algorithms for SVM training (see, for instance, Astorino and Fuduli 2015; Boser et al. 1992; Byrd et al. 2011; Carrizosa and Romero Morales 2013; Cortes and Vapnik 1995; Fan et al. 2008; Ferris and Munson 2004; Franc and Sonnenburg 2009; Fung and Mangasarian 2001; Gaudioso et al. 2017; Hsu and Lin 2002a; Keerthi and Lin 2003; Lee et al. 2015; Lee and Mangasarian 2001; Mangasarian and Musicant 2001; Mangasarian 2006; Mavroforakis and Theodoridis 2006; Osuna et al. 1997; Glasmachers and Dogan 2013; Tsang et al. 2005; Wang and Lin 2014; Wang et al. 2012). We observe that in neural network training, where the unconstrained optimization problem is nonconvex and suitable safeguards (for instance, early stopping) must be adopted in order to avoid converging too quickly towards undesired minima (in terms of generalization capabilities), the requirements of a training algorithm are not well-defined from an optimization point of view.

The SVM training problem can be equivalently formulated as a (linearly constrained) quadratic convex problem or, by Wolfe's duality theory, as a quadratic convex problem with one linear constraint and box constraints. Depending on the formulation, several optimization algorithms have been specifically designed for SVM training. Thus, we present the most important contributions for the *primal formulations*, i.e., Newton methods, least-squares algorithms, stochastic sub-gradient methods, cutting plane algorithms, and for the *dual formulations* decomposition methods. Interior point methods were developed both for the primal and the dual formulations. We observe

that the design of convergent and efficient decomposition methods for SVM training has yielded relevant advances both from a theoretical and computational point of view. Indeed, the "classical" decomposition methods for nonlinear optimization, such as the successive over-relaxation algorithm and the Jacobi and Gauss-Seidel algorithms, are applicable only when the feasible set is the Cartesian product of subsets defined in smaller subspaces. Since the SVM training problem contains an equality constraint, such methods cannot be directly employed, and this has motivated the study and the design of new decomposition algorithms improving the state-of-art.

The paper is organized as follows. We formally introduce in Sect. 2 the concept of *optimal separating hyperplane* underlying linear SVM, we give the primal formulation of the linear SVM training problem, and we recall the fundamental concepts of the Wolfe's dual theory necessary for defining the dual formulation of the linear SVM training problem. The dual formulation allows us, through the so-called *kernel trick*, to immediately extend in Sect. 3 the approach of linear SVM to the case of nonlinear classifiers. Sections 4 and 5 contain the analysis of unconstrained and constrained methods, respectively, for the primal formulations. The wide class of decomposition methods for the dual formulation is analyzed in Sect. 6. Interior point methods are presented in Sect. 7. Finally, in Sect. 8 we direct the reader to the available software for SVM training related to the presented methods. In the appendices we provide the proofs of important results concerning: (1) the existence and uniqueness of the optimal hyperplane; (2) Wolfe's dual theory both in the general and in the quadratic case; (3) the kernel functions. As regards (1), although the result is well-known, we believe that the kind of proof is novel and technically interesting. Concerning (2) and (3), they represent pillars of SVM methodology, and a reader might find them of interest to obtain some related technical insights.

2 The optimal separating hyperplane and linear SVM

The Training Set (TS) is a set of l observations:

$$TS = \{(x^i, y^i),\ x^i \in X \subseteq \Re^n,\ y^i \in Y \subseteq \Re,\ i = 1, \ldots, l\}.$$

The vectors x^i are the patterns belonging to the input space. The scalars y^i are the labels (targets). In a *classification problem* we have that $y^i \in \{-1, 1\}$, in a *regression problem* $y^i \in \Re$. We will focus only on classification problems.

Let us consider two disjoint sets A and B of points in $\Re^n$ to be classified. Assume that A and B are *linearly separable*, that is, there exists a hyperplane $H = \{x \in \Re^n : w^T x + b = 0\}$ such that the points $x^i \in A$ belong to one half-space, and the points $x^j \in B$ belong to the other half-space. More precisely, we can assume that there exist a vector $w \in \Re^n$ and a scalar $b \in \Re$ such that

$$\begin{array}{ll} w^T x^i + b & \geq 1, \quad \forall x^i \in A \\ w^T x^j + b & \leq -1, \ \forall x^j \in B \end{array} \tag{1}$$

A hyperplane will be indicated by $H(w, b)$. We say that $H(w, b)$ is a *separating hyperplane* if the pair (w, b) is such that (1) holds. The decision function of a linear

classifier associated with a separating hyperplane is $f_d(x) = \text{sgn}(w^T x + b)$. We introduce the concept of *margin* of a separating hyperplane.

Definition 1 Let $H(w, b)$ be a separating hyperplane. The margin of $H(w, b)$ is the minimum distance ρ between points in $A \cup B$ and the hyperplane $H(w, b)$, that is

$$\rho(w, b) = \min_{x^i \in A \cup B} \left\{ \frac{|w^T x^i + b|}{\|w\|} \right\}.$$

It is quite intuitive that the margin of a given separating hyperplane is related to the generalization capability of the corresponding linear classifier, i.e., to correctly classify unseen data. The relationship between the margin and the generalization capability of linear classifiers is analyzed by the statistical learning theory Vapnik (1998), which theoretically motivates the importance of defining the hyperplane with *maximum margin*, the so-called *optimal separating hyperplane*.

Definition 2 Given two linearly separable sets A and B, the optimal separating hyperplane is a separating hyperplane $H(w^\star, b^\star)$ having maximum margin.

It can be proved that the optimal hyperplane *exists and is unique* (see "Appendix A"). From the above definition we get that the optimal hyperplane is the unique solution of the following problem

$$\begin{aligned} &\max_{w \in \Re^n, b \in \Re} \ \min_{x^i \in A \cup B} \left\{ \frac{|w^T x^i + b|}{\|w\|} \right\} \\ &\text{s.t. } y^i \left[w^T x^i + b \right] - 1 \geq 0 \quad i = 1, \ldots, l. \end{aligned} \tag{2}$$

It can be proved that problem (2) is equivalent to the convex quadratic programming problem

$$\begin{aligned} &\min_{w \in \Re^n, b \in \Re} F(w) = \frac{1}{2} \|w\|^2 \\ &\text{s.t. } y^i \left[w^T x^i + b \right] - 1 \geq 0, i = 1, \ldots, l. \end{aligned} \tag{3}$$

Now assume that the two sets A are B are not linearly separable. This means that the system of linear inequalities (1) does not admit solution. Let us introduce *slack* variables ξ_i, with $i = 1, \ldots, l$:

$$y^i \left[w^T x^i + b \right] - 1 + \xi_i \geq 0, \ i = 1, \ldots, l. \tag{4}$$

Note that whenever a vector x^i is not correctly classified the corresponding variable ξ^i is greater than 1. The variables ξ_i corresponding to vectors correctly classified and belonging to the "separation zone" are such that $0 < \xi^i < 1$. Therefore, the term $\sum_{i=1}^{l} \xi_i$ is an *upper bound* on the number of the classification errors on the training vectors. Then, it is quite natural to add to the objective function of problem (3) the

term $C\sum_{i=1}^{l}\xi_i$, where $C > 0$ is a parameter to assess the training error. The primal problem becomes

$$\begin{array}{ll} \min\limits_{w,b,\xi} F(w,\xi) = \dfrac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\xi_i & \\ \text{s.t.} \quad y^i\left[w^T x^i + b\right] - 1 + \xi_i \geq 0 & i = 1,\ldots,l \\ \xi_i \geq 0 \quad i = 1,\ldots,l & \end{array} \tag{5}$$

For reasons explained later, the dual problem of (5) is often considered. We direct the reader to Bertsekas (1999), Mangasarian (1994), Fletcher (1987) for insights on duality in nonlinear programming. Let us consider the convex programming problem

$$\begin{array}{l} \min\limits_{x} f(x) \\ Ax - b \leq 0, \end{array} \tag{6}$$

where $f : \Re^n \to \Re$ is a convex, continuously differentiable function, $A \in \Re^{m\times n}$, $b \in \Re^m$. Introducing the Lagrangian function $L(x,\lambda) = f(x) + \lambda^T(Ax - b)$, *Wolfe's dual* of (6) is defined as follows

$$\begin{array}{l} \max\limits_{x,\lambda} L(x,\lambda) \\ \nabla_x L(x,\lambda) = 0 \\ \lambda \geq 0. \end{array} \tag{7}$$

It can be proved (see "Appendix B") that, if problem (6) admits a solution $x^\star$, then there exists a vector of Lagrange multipliers $\lambda^\star$ such that $(x^\star, \lambda^\star)$ is a solution of (7).

In the general case, given a solution $(\bar{x}, \bar{\lambda})$ of Wolfe's dual, we can not draw conclusions with respect to the primal problem (6). In the particular case of convex quadratic programming problems the following result holds (see "Appendix B").

Proposition 1 *Let $f(x) = \frac{1}{2}x^T Qx + c^T x$, and suppose that the matrix Q is symmetric and positive semidefinite. Let $(\bar{x}, \bar{\lambda})$ be a solution of Wolfe's dual (7). Then, there exists a vector $x^\star$ (not necessarily equal to $\bar{x}$) such that*

(i) $Q(x^\star - \bar{x}) = 0$;
(ii) *$x^\star$ is a solution of problem (6); and*
(iii) *x^* is a global minimum of (6) with associated multipliers $\bar{\lambda}$.*

Now let us consider the convex quadratic programming problem (5). Here the primal variables are (w, b, ξ), and the condition $\nabla_x L(x,\lambda) = 0$ gives two constraints

$$w = \sum_{i=1}^{l}\lambda_i y^i x^i \qquad \sum_{i=1}^{l}\lambda_i y^i = 0.$$

Then, setting $X = \left[y^1 x^1, \ldots, y^l x^l\right]$, $\lambda^T = \left[\lambda^1, \ldots, \lambda^l\right]$, Wolfe's dual of (5) is a *convex quadratic programming problem* of the form

$$\begin{aligned} \min_{\lambda} \ & \Gamma(\lambda) = \frac{1}{2}\lambda^T X^T X \lambda - e^T \lambda \\ s.t. \ & \sum_{i=1}^{l} \lambda_i y^i = 0 \\ & 0 \le \lambda \le C, \end{aligned} \tag{8}$$

where $e^T = [1, \ldots, 1]$.

Once a solution $\lambda^\star$ is computed, the primal vector $w^\star$ can be determined as follows

$$w^\star = \sum_{i=1}^{l} \lambda_i^\star y^i x^i,$$

i.e., $w^\star$ depends only on the so-called (*support vectors*) x^i whose corresponding multipliers $\lambda_i^\star$ are not null. The support vectors corresponding to multipliers $\lambda_i^\star$ such that $0 < \lambda_i^\star < C$ are called *free support vectors*, those corresponding to multipliers $\lambda_i^\star = C$ are called *bounded support vectors*. We also observe that assertion (iii) of Proposition 1 ensures that an optimal solution $(w^\star, b^\star)$ satisfies the complementarity conditions with multipliers equal to $\lambda^\star$. Thus, by considering any free support vector x^i, we have $0 < \lambda_i^\star < C$, which implies

$$y^i \left((w^\star)^T x^i + b^\star \right) - 1 = 0, \qquad i = 1, \ldots, l, \tag{9}$$

so that, once $w^\star$ is computed, the scalar $b^\star$ can be determined by means of the corresponding complementarity condition defined by (9).

Finally, we observe that the decision function of a linear SVM is

$$f_d(x) = \operatorname{sgn}\left((w^*)^T x + b^\star \right) = \operatorname{sgn}\left(\sum_{i=1}^{l} \lambda_i^\star y^i (x^i)^T x + b^\star \right).$$

Summarizing, we have that the duality theory leads to a convenient way to deal with the constraints. Moreover, the dual optimization problem can be written in terms of dot products, as well as the decision function, and this allows us to easily extend the approach to the case of nonlinear classifiers.

3 Nonlinear SVM

The idea underlying the nonlinear SVM is that of mapping the data of the input space onto a higher dimensional space called *feature space* and to define a linear classifier in this feature space.

Let us consider a mapping $\phi : \Re^n \to \mathcal{H}$ where $\mathcal{H}$ is an Euclidean space (the *feature space*) whose dimension is greater than n (the dimension can be even infinite). The input training vectors x^i are mapped onto $\phi(x^i)$, with $i = 1, \ldots, l$.

We can think to define a linear SVM in the feature space by replacing x^i with $\phi(x^i)$. Then we have

- the dual problem (8) is replaced by the following problem

$$\begin{aligned} \min_{\lambda} \ & \Gamma(\lambda) = \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} y^i y^j \phi(x^i)^T \phi(x^j)\lambda_i \lambda_j - \sum_{i=1}^{l} \lambda_i \\ s.t. \ & \sum_{i=1}^{l} \lambda_i y^i = 0 \\ & 0 \le \lambda_i \le C \qquad i = 1, \ldots, l; \end{aligned} \tag{10}$$

- the optimal primal vector $w^\star$ is

$$w^\star = \sum_{i=1}^{l} \lambda_i^\star y^i \phi(x^i);$$

- given $w^\star$ and any $0 < \lambda_i^\star < C$, the scalar $b^\star$ can be determined using the complementarity conditions

$$y^i \left(\sum_{j=1}^{l} \lambda_j^\star y^j \phi(x^j)^T \phi(x^i) + b^\star \right) - 1 = 0; \quad \text{and} \tag{11}$$

- the decision function takes the form

$$f_d(x) = \operatorname{sgn}\left((w^*)^T \phi(x) + b^\star \right). \tag{12}$$

Remark 1 The primal/dual relation in infinite dimensional spaces has been rigorously discussed in Lin (2001a).

From (12) we get that the separation surface is:

- *linear* in the *feature space*;
- *non linear* in the *input space*.

It is important to observe that both in the dual formulation (10) and in formula (12) concerning the decision function it is not necessary to explicitly know the mapping ϕ, but it is sufficient to know the inner product $\phi(x)^T \phi(z)$ of the feature space. This leads to the fundamental concept of *kernel function*.

Definition 3 Given a set $X \subseteq \Re^n$, a function

$$K : X \times X \to \Re$$

is a *kernel* if

$$K(x, y) = \phi(x)^T \phi(y) \qquad \forall x, y \in X, \tag{13}$$

where ϕ is an application $X \to \mathcal{H}$ and $\mathcal{H}$ is an Euclidean space, that is, a linear space with a fixed inner product.

We observe that a kernel is necessarily a symmetric function. It can be proved that $K(x, z)$ is a kernel if and only if the $l \times l$ matrix

$$\left(K(x^i, x^j)\right)_{i,j=1}^{l} = \begin{pmatrix} K(x^1, x^1) & \dots & K(x^1, x^l) \\ & \vdots & \\ K(x^l, x^1) & \dots & K(x^l, x^l) \end{pmatrix}$$

is positive semidefinite for any set of training vectors $\{x^1, \dots, x^l\}$. The kernel is often referred to as the Mercer kernel in the literature. We have the following result, whose proof is reported in "Appendix C".

Proposition 2 *Let $K : X \times X \to \Re$ be a symmetric function. Then K is a kernel if and only if, for any choice of the vectors $x^1, \dots, x^\ell$ in X the matrix*

$$K = \left[K(x_i, x_j)\right]_{i,j=1,\dots,\ell}$$

is positive semidefinite.

Using the definition of kernel problem (10) can be written as follows

$$\begin{aligned} \min_{\lambda} \ & \Gamma(\lambda) = \tfrac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} y^i y^j K(x^i, x^j)\lambda_i \lambda_j - \sum_{i=1}^{l} \lambda_i \\ s.t. \ & \sum_{i=1}^{l} \lambda_i y^i = 0 \\ & 0 \le \lambda_i \le C \qquad i = 1, \dots, l. \end{aligned} \tag{14}$$

By Proposition 2 it follows that problem (14) is a convex quadratic programming problem.
Examples of kernel functions are:

$K(x, z) = (x^T z + 1)^p$ *polynomial kernel* (p integer ≥ 1)

$K(x, z) = e^{-\|x-z\|^2/2\sigma^2}$ *Gaussian kernel* ($\sigma > 0$)

$K(x, z) = tanh(\beta x^T z + \gamma)$ *hyperbolic tangent kernel* (for suitable values of β and γ)

It can be shown that the Gaussian kernel is an inner product in an infinite dimensional space. Using the definition of kernel function the decision function is

$$f_d(x) = \text{sgn}\left(\sum_{i=1}^{l} \lambda_i^\star y^i K(x, x^i) + b^\star\right).$$

4 Unconstrained primal formulations

Let us consider the linearly constrained primal formulation (5) for linear SVM. It can be shown that problem (5) is equivalent to the following unconstrained nonsmooth problem

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l} \max\{0, 1 - y^i(w^T x^i + b)\}. \tag{15}$$

The above formulation penalizes slacks (ξ) linearly and is called L_1-SVM. An unconstrained smooth formulation is that of the so-called L_2-SVM, where slacks are quadratically penalized, i.e.,

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l} \max^2\{0, 1 - y^i(w^T x^i + b)\}. \tag{16}$$

Least Squares SVM (LS-SVM) considers the primal formulation (5), where the inequality constraints

$$y^i(w^T x^i + b) \geq 1 - \xi^i,$$

are replaced by the equality constraints

$$y^i(w^T x^i + b) = 1 - \xi^i.$$

This leads to a regularized linear least squares problem

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l} (y^i(w^T x^i + b) - 1)^2. \tag{17}$$

The general unconstrained formulation takes the form

$$\min_{w,b} R(w, b) + C\sum_{i=1}^{l} L(w, b; x^i, y^i), \tag{18}$$

where $R(w, b)$ is the *regularization term* and $L(w, b; x^i, y^i)$ is the *loss function* associated with the observation (x^i, y^i).

We observe that the bias term b plays a crucial role both in the learning model, i.e., it may be critical for successful learning (especially in unbalanced datasets), and in the optimization-based training process. The simplest approach to learn the bias term is that of adding one more feature to each instance, with constant value equal to 1. In this way, in L_1-SVM, L_2-SVM and LS-SVM, the regularization term becomes $\frac{1}{2}(\|w\|^2 + b^2)$ with the advantages of having convex properties of the objective function useful for convergence analysis and the possibility to directly apply

algorithms designed for models without the bias term. The conceptual disadvantage of this approach is that the statistical learning theory underlying SVM models is based on an *unregularized* bias term. We will not go into the details of the issues concerning the bias term.

The extension of the unconstrained approach to nonlinear SVM, where the data x^i are mapped onto the feature space $\mathcal{H}$ by the mapping $\phi : \Re^n \to \mathcal{H}$, are often done by means of the *representer theorem* Kimeldorf and Wahba (1970). Using this theorem we have that the solution of SVM formulations can be expressed as a linear combination of the mapped training instances. Then, we can train a nonlinear SVM without direct access to the mapped instances, but using their inner products through the kernel trick. For instance, setting $w = \sum_{i=1}^{l} \beta_i \phi(x^i)$, the optimization problem corresponding to L_2-SVM with regularized bias term is the following unconstrained problem

$$\min_{\beta,b} \frac{1}{2}\beta^T K\beta + C\sum_{i=1}^{l} \max^2\{0, 1 - y^i \beta^T K_i\}, \tag{19}$$

where K is the kernel matrix associated to the mapping ϕ and K_i is the $i-$th column. Note that both (16) and (19) are piecewise convex quadratic functions.

4.1 Methods for primal formulations

First let us consider the nonsmooth formulation (15) without considering the bias term b. A simple and effective stochastic sub-gradient descent algorithm has been proposed in Shalev-Shwartz et al. (2011). The vector w is initially set to 0. At iteration t, a pair (x^{i_t}, y^{i_t}) is randomly chosen in the training set, and the objective function

$$f(w) = \frac{\lambda}{2}\|w\|^2 + \frac{1}{l}\sum_{i=1}^{l} \max\{0, 1 - y^i (w^T x^i)\}$$

is approximated as follows

$$f(w; i_t) = \frac{\lambda}{2}\|w\|^2 + \max\{0, 1 - y^{i_t}(w_t^T x^{i_t})\}.$$

The sub-gradient of $f(w; i_t)$ is

$$\nabla_t = \lambda w_t - 1\left[y^{i_t} w_t^T x^{i_t} < 1\right] y^{i_t} x^{i_t},$$

where $1\left[y^{i_t} w_t^T x^{i_t} < 1\right]$ is the indicator function which takes the value one if its argument is true and zero otherwise. The vector w is updated as follows

$$w_{t+1} = w_t - \eta_t \nabla_t,$$

where $\eta_t = \frac{1}{\lambda t}$, and $\lambda > 0$. A more general version of the algorithm is the one based on *mini-batch* iterations, where instead of using a single example (x^{i_t}, y^{i_t}) of the training set, a subset of training examples, defined by the set $A_t \subset \{1, \ldots, P\}$, with $|A_t| = r$, is considered. The objective function is approximated as follows

$$f(w; A_t) = \frac{\lambda}{2}\|w\|^2 + \frac{1}{r}\sum_{i \in A_t} \max\{0, 1 - y^i(w^T x^i)\}$$

whose sub-gradient is

$$\nabla_t = \lambda w_t - \frac{1}{r}\sum_{i \in A_t} 1\left[y^{i_t} w_t^T x^{i_t} < 1\right] y^{i_t} x^{i_t}.$$

The updating rule is again

$$w_{t+1} = w_t - \eta_t \nabla_t.$$

In the *deterministic* case, that is, when all the training examples are used at each iteration, i.e., $A_t = \{1, \ldots, l\}$, the complexity analysis shows that the number of iterations required to obtain an $\epsilon-$approximate solution is $O(1/\lambda\epsilon)$. In the *stochastic* case, i.e., $A_t \subset \{1, \ldots, l\}$, a similar result in probability is given. We observe that the complexity analysis relies on the property that the objective function is $\lambda-$*strongly convex*, i.e.,

$$f(w) - \frac{\lambda}{2}\|w\|^2$$

is a convex function.

The extension to nonlinear SVM is performed taking into account that, once mapped the input data x^i onto $\phi(x^i)$, thanks to the fact that w is initialized to 0, we can write

$$w_{t+1} = \frac{1}{\lambda t}\sum_{i=1}^{l} \alpha_{t+1}[i] y^i \phi(x^i),$$

where $\alpha_{t+1}[i]$ counts how many times example i has been selected so far and we had a non-zero loss on it. It can be shown that the algorithm does not require the explicit access to the weight vector w. To this aim, we show how the vector α, initialized to zero, is iteratively updated. At iteration t, the index i_t is randomly chosen in $\{1, \ldots, l\}$, and we set

$$\alpha_{t+1}[i] = \alpha_t[i] \qquad i \neq i_t.$$

If

$$y^{i_t} \frac{1}{\lambda t} \sum_{i=1}^{l} \alpha_t[i] y^j K(x^{i_t}, x^i) < 1$$

then set $\alpha_{t+1}[i_t] = \alpha_t[i_t] + 1$, otherwise set $\alpha_{t+1}[i_t] = \alpha_t[i_t]$. Thus, the algorithm can be implemented by maintaining the vector α, using only kernel evaluations, without direct access to the feature vectors $\phi(x)$.

Newton-type methods for formulation (16) of L_2-SVM have been proposed first in Mangasarian (2002) and then in Keerthi and DeCoste (2005). The main difficulty of this formulation concerns the fact that the objective function is not twice continuously differentiable, so that the *generalized Hessian* must be considered. Finite convergence is proved in both papers. The main peculiarities of the algorithm designed in Keerthi and DeCoste (2005) are: (1) the formulation of a linear least square problem for computing the search direction (i.e., the violated constraints, depending on the current solution, are replaced by equality constraints); (2) the adoption of an exact line search for determining the stepsize. The matrix of the least square problem has a number of rows equal to $n + n_v$, where n_v is the number of *violated inequality constraints*, i.e., the constraints such that $y^i w^T x^i < 1$.

Newton optimization for problem (19) and the relationship with the dual formulation have been deeply discussed in Chapelle (2007). In particular, it is shown that the complexity of one Newton step is $O(ln_{sv} + n_{sv}^3)$, where again n_{sv} is the number of *violated inequality constraints*, i.e., the constraints such that $y^i(\beta^T K_i) < 1$.

In Chang et al. (2008), the primal unconstrained formulation for linear classification (18) is considered, with L2 regularization and L2 loss function, i.e., $f(w) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l} \max\{0, 1 - y^i w^T x^i\}^2$. The authors propose a coordinate descent algorithm, where w^{k+1} is constructed by sequentially updating each component of w^k. Define

$$w^{k,i} = (w_1^{k+1}, \ldots, w_{i-1}^{k+1}, w_i^k, \ldots, w_n^k) \text{ for } i = 2, \ldots, n$$

with $w^{k,1} = w^k$ and $w^{k,n+1} = w^{k+1}$. In order to update the i-th component defining $w^{k,i+1}$, the following one variable unconstrained problem is approximately solved:

$$\min_z f(w^{k,i} + ze_i)$$

The obtained function is a piecewise quadratic function, and the problem is solved by means of a line search along the Newton direction computed using the generalized second derivative proposed in Mangasarian (2002). The authors prove that the algorithm converges to an ϵ accurate solution in $O\left(nC^3P^6(\#nz)^3 \log(\frac{1}{\epsilon})\right)$ where #nz is total number of nonzero values of training data, and $P = \max |x_j^i|$.

Finally, standard algorithms for the least squares formulation (17) concerning LS-SVM have been presented in Suykens and Vandewalle (1999) and in Cassioli et al. (2013). In this latter paper an incremental recursive algorithm, which requires storing a square matrix (whose dimension is equal to the number of features of the data), has been employed and could be used, in principle, even for online learning.

5 Constrained primal formulations and cutting plane algorithms

A useful tool in optimization is represented by cutting planes technique. Depending on the class of problems, this kind of tool can be used for strengthening a relaxation, for solving a convex problem by means of a sequence of LP relaxations, or for making tractable a problem with an exponential number of constraints.

This type of machinery is applied in Joachims (2006), Joachims et al. (2009), Joachims and Yu (2009) for training an SVM. The main idea is to reformulate SVM training as a problem with quadratic objective and an exponential number of constraints, but with only one slack variable that measures the overall loss in accuracy in the training set. The constraints are obtained as the combination of all the possible subsets of constraints in problem (5). Then, a master problem that is the training of a smaller size SVM is solved at each iteration, and the constraint that is most violated in the solution is added for the next iteration.

The advantage is that it can be proved that the number of iteration is bounded and the bound is independent on the size of the problem, but depends only on the desired level of accuracy.

More specifically, in Joachims (2006), the primal formulation (5) with $b = 0$ is considered where the error term is divided by the number of elements in the training set, i.e.,

$$\begin{aligned} \min_{w,\xi} & F(w,\xi) = \frac{1}{2}\|w\|^2 + \frac{C}{l}\sum_{i=1}^{l}\xi_i \\ \text{s.t.} \quad & y^i\left[w^T x^i\right] - 1 + \xi_i \geq 0 \qquad i = 1,\ldots,l \\ & \xi_i \geq 0 \qquad i = 1,\ldots,l. \end{aligned} \tag{20}$$

Then, an equivalent formulation called Structural Classification SVM is defined:

$$\begin{aligned} \min_{w,\xi} & F(w,\xi) = \frac{1}{2}\|w\|^2 + C\xi \\ \text{s.t.} \quad & \forall \mathbf{c} \in \{0,1\}^l \; : \; \frac{1}{l} w^T \sum_{i=1}^{l} c_i y^i x^i \geq \frac{1}{l}\sum_{i=1}^{l} c_i - \xi. \\ & \xi \geq 0 \end{aligned} \tag{21}$$

This formulation corresponds to summing up all the possible subsets of the constraints in (20), and has an exponential number of constraints, one for each vector $\mathbf{c} \in \{0,1\}^l$, but there is only one slack variable ξ. The two formulations can be shown to be equivalent, in the sense that any solution w^* of problem (21) is also a solution of problem (20), with $\xi^* = \frac{1}{l}\sum \xi_i^*$. The proof relies on the observation that for any value of w the slack variables for the two problems that attain the minimum objective value satisfy $\xi = \frac{1}{l}\sum \xi_i$. Indeed, for a given w the smallest feasible slack variables in (20) are $\xi_i = \max\{0, 1 - y^i w^T x^i\}$. In a similar way, in (21) for a given w the smallest feasible slack variable can be found by solving

$$\xi = \max_{\mathbf{c}\in\{0,1\}^l} \left\{ \frac{1}{l}\sum_{i=1}^{l} c_i - \frac{1}{l}\sum_{i=1}^{l} c_i y_i w^T x^i \right\}. \tag{22}$$

However, problem (22) can be decomposed into l problems, one for each component of the vector $\mathbf{c}$, i.e.,

$$\min \xi = \sum_{i=1}^{l} \max_{c_i \in \{0,1\}} \left\{ \frac{1}{l} c_i - \frac{1}{l} c_i y_i w^T x^i \right\} = \sum_{i=1}^{l} \max \left\{ 0, \frac{1}{l} - \frac{1}{l} y_i w^T x^i \right\} = \min \frac{1}{l} \sum_{i=1}^{l} \xi_i,$$

so that the objective values of problems (20) and (21) coincide at the optimum. This equivalence result implies that it is possible to solve (21) instead of (20).

The advantage of this problem is that there is a single slack variable that is directly related to the infeasibility, since if (w, ξ) satisfies all the constraints with precision ϵ, then the point $(w, \xi + \epsilon)$ is feasible. This allows one to establish an effective and straightforward stopping criterion related to the accuracy on the training loss.

The cutting plane algorithm for solving problem (21) is the following:

Cutting Plane Algorithm

Data. The training set TS, C, ϵ.

Inizialization. $\mathcal{W} = \emptyset$.

Repeat

1. update (w, ξ) with the solution of

$$\begin{aligned} &\min \ \frac{1}{2}\|w\|^2 + C\xi \\ &\text{s.t.} \ \ \forall \mathbf{c} \in \mathcal{W} \ : \ \frac{1}{l} w^T \sum_{i=1}^{l} c_i y^i x^i \geq \frac{1}{l} \sum_{i=1}^{l} c_i - \xi \end{aligned} \tag{23}$$

2. **for** $i = 1, \ldots, l$

$$c_i = \begin{cases} 1 \text{ if } y^i w^T x^i < 1 \\ 0 \text{ otherwise.} \end{cases}$$

end for

3. set $\mathcal{W} = \mathcal{W} \cup \{\mathbf{c}\}$.

Until $\left(\frac{1}{l} \sum_{i=1}^{l} c_i - \frac{1}{l} \sum_{i=1}^{l} c_i y^i w^T x^i \leq \xi + \epsilon \right)$

Return (w, ξ)

This algorithm starts with an empty set of violated constraints, and then iteratively builds a sufficient subset of the constraints of problem (21). Step 1 solves the problem with the current set of constraints. The vector $\mathbf{c}$ computed at Step 2 corresponds to selecting the constraint in (21) that requires the largest ξ to make it feasible given the current w, i.e., it finds the most violated constraint. The stopping criterion implies that the algorithm stops when the accuracy on the training loss is considered acceptable. Problem (23) can be solved either by solving the primal or by solving the dual, with

any training algorithm for SVM. It can be shown that the algorithm terminates after at most $\max\left\{\frac{2}{\epsilon}, \frac{8CR^2}{\epsilon^2}\right\}$ iterations, where $R = \max_i \|x_i\|$, and this number also bounds the size of the working set $\mathcal{W}$ to a constant that is independent on n and l. Furthermore, for a constant size of the working set $\mathcal{W}$, each iteration takes $O(sl)$, where s is the number of nonzero features for each element of the working set. This algorithm is thus extremely competitive when the problem is highly sparse, and has been extended to handle structural SVM training in Joachims et al. (2009). It is also possible to obtain a straightforward extension of this approach to non linear kernels, defining a dual version of the algorithm. However, whereas the fixed number of iteration properties does not change, the time complexity per iteration worsens significantly, becoming $O(m^3 + ml^2)$ where m is the number of constraints added in the primal. The idea in Joachims and Yu (2009) is then to use arbitrary basis vectors to represent the learned rule, not only the support vectors, in order to find sparser solutions and keep the iteration cost lower. In particular, instead of using the Representer Theorem, setting $w = \sum_{i=1}^{l} \alpha_i y^i \phi(x^i)$ and considering the subspace $\mathcal{F} = \text{span}\{\phi(x^1), \ldots, \phi(x^l)\}$, they consider a smaller subspace $\mathcal{F}' = \text{span}\{\phi(b^1), \ldots, \phi(b^k)\}$ for some small k and the basis vectors b^i are built during the algorithm. In this setting, each iteration has time complexity at most $O(m^3 + mk^2 + kl)$.

Finally in Teo et al. (2010) and Le et al. (2008) a bundle method is defined for regularized risk minimization problems, that is shown to converge in $O(1/\epsilon)$ steps for linear classification problems, and that is further optimized in Franc and Sonnenburg (2009) and Franc and Sonnenburg (2008), where an optimized choice of the cutting planes is described.

6 Decomposition algorithms for the dual formulation

Let us consider the convex quadratic programming problem for SVM training in the case of classification problems:

$$\begin{aligned} \min_{\alpha} \ & f(\alpha) = \tfrac{1}{2}\alpha^T Q\alpha - e^T\alpha \\ s.t. \quad & y^T\alpha = 0 \\ & 0 \le \alpha \le C, \end{aligned} \tag{24}$$

where $\alpha \in \Re^l$, l is the number of training data, Q is a $l \times l$ symmetric and positive semidefinite matrix, $e \in \Re^l$ is the vector of ones, $y \in \{-1, 1\}^l$, and C is a positive scalar. The generic element q_{ij} of Q is $y_i y_j K(x^i, x^j)$, where $K(x, z) = \phi(x)^T\phi(z)$ is the kernel function related to the nonlinear function ϕ that maps the data from the input space into the feature space. We prefer to adopt here the symbol α (instead of λ as in (14)) for the dual variables, since it is a choice of notation often adopted in the SVM literature.

The structure of problem (24) is very simple, but we assume that the number l of training data is huge (as in many big data applications) and the Hessian matrix Q, which

is dense, cannot be fully stored so that standard methods for quadratic programming cannot be used. Hence, the adopted strategy to solve the SVM problem is usually based on the decomposition of the original problem into a sequence of smaller subproblems obtained by fixing subsets of variables.

We remark that the need to design specific decomposition algorithms, instead of the well-known *block coordinate descent* methods, arises from the presence of the equality constraints that, in particular, makes the convergence analysis difficult. The classical decomposition methods for nonlinear optimization, such as the successive over-relaxation algorithm and the Jacobi and GaussSeidel algorithms Bertsekas (1999), are applicable only when the feasible set is the Cartesian product of subsets defined in smaller subspaces.

In a general decomposition framework, at each iteration k, the vector of variables α^k is partitioned into two subvectors $(\alpha^k_W, \alpha^k_{\overline{W}})$, where the index set $W \subset \{1, \ldots, l\}$ identifies the variables of the subproblem to be solved and is called *working set*, and $\overline{W} = \{1, \ldots, l\} \setminus W$ (for notational convenience, we omit the dependence on k).

Starting from the current solution $\alpha^k = (\alpha^k_W, \alpha^k_{\overline{W}})$, which is a feasible point, the subvector α^{k+1}_W is computed as the solution of the subproblem

$$\begin{aligned} \min_{\alpha_W} \; & f(\alpha_W, \alpha^k_{\overline{W}}) \\ & y_W^T \alpha_W = -y_{\overline{W}}^T \alpha^k_{\overline{W}} \\ & 0 \le \alpha_W \le C. \end{aligned} \tag{25}$$

The variables corresponding to $\overline{W}$ are unchanged, that is, $\alpha^{k+1}_{\overline{W}} = \alpha^k_{\overline{W}}$, and the current solution is updated setting $\alpha^{k+1} = (\alpha^{k+1}_W, \alpha^{k+1}_{\overline{W}})$. The general framework of a decomposition scheme is reported below.

Decomposition framework

Data. A feasible point α^0 (usually $\alpha^0 = 0$).

Inizialization. Set $k = 0$.

While (the stopping criterion is not satisfied)

1. select the working set W^k;
2. set $W = W^k$ and compute a solution α^*_W of the problem (25);
3. set $\alpha_i^{k+1} = \begin{cases} \alpha_i^* & for\ i \in W \\ \alpha_i^k & otherwise; \end{cases}$
4. set $\nabla f(\alpha^{k+1}) = \nabla f(\alpha^k) + Q\left(\alpha^{k+1} - \alpha^k\right)$.
5. set $k = k + 1$.

end while

Return $\alpha^* = \alpha^k$

The choice $\alpha^0 = 0$ for the starting point is motivated by the fact that this point is a feasible point and such that the computation of the gradient $\nabla f(\alpha^0)$ does not require any element of the matrix Q, being $\nabla f(0) = -e$. The cardinality q of the working set, namely the dimension of the subproblem, must be *greater than or equal to* 2, due to the presence of the linear constraint, otherwise we would have $\alpha^{k+1} = \alpha^k$.

The selection rule of the working set strongly affects both the speed of the algorithm and its convergence properties. In computational terms, the most expensive step at each iteration of a decomposition method is the evaluation of the kernel to compute the columns of the Hessian matrix, corresponding to the indices in the working set W. These columns are needed for updating the gradient.

We distinguish between:

- *Sequential Minimal Optimization* (SMO) algorithms, where the size of the working set is exactly equal to two; and
- *General Decomposition Algorithms*, where the size size of the working set is strictly greater than two.

In the sequel we will mainly focus on SMO algorithms, since they are the most used algorithms to solve large quadratic programs for SVM training.

6.1 Sequential Minimal Optimization (SMO) algorithms

The decomposition methods usually adopted are the so-called "Sequential Minimal Optimization" (SMO) algorithms, since at each iteration they update the minimum number of variables, that is two. At each iteration, an SMO algorithm requires the solution of a convex quadratic programming of two variables with one linear equality constraint and box constraints. Note that the solution of a subproblem in two variables of the above form can be analytically determined (and this is one of the reasons motivating the interest in defining SMO algorithms). SMO algorithms were the first methods proposed for SVM training and the related literature is wide (see, e.g., Joachims 1999; Keerthi and Gilbert 2002; Lin 2001b; Osuna et al. 1997; Platt 1999).

The analysis of SMO algorithms relies on feasible and descent directions having only two nonzero elements. In order to characterize these directions, given a feasible point $\bar{\alpha}$, let us introduce the following index sets

$$\begin{aligned} R(\bar{\alpha}) &= L^+(\bar{\alpha}) \cup U^-(\alpha) \cup \{i : \ 0 < \bar{\alpha}_i < C\} \\ S(\bar{\alpha}) &= L^-(\bar{\alpha}) \cup U^+(\bar{\alpha}) \cup \{i : \ 0 < \bar{\alpha}_i < C\}, \end{aligned} \tag{26}$$

where

$$L^+(\bar{\alpha}) = \{i : \ \bar{\alpha}_i = 0, \ y_i > 0\}, \ \ L^-(\bar{\alpha}) = \{i : \ \bar{\alpha}_i = 0, \ y_i < 0\}$$

$$U^+(\bar{\alpha}) = \{i : \ \bar{\alpha}_i = C, \ y_i > 0\}, \ U^-(\bar{\alpha}) = \{i : \ \bar{\alpha}_i = C, \ y_i < 0\}.$$

Note that

$$R(\bar{\alpha}) \cap S(\bar{\alpha}) = \{i : 0 < \bar{\alpha}_i < C\} \qquad R(\bar{\alpha}) \cup S(\bar{\alpha}) = \{1, \ldots, l\}.$$

The introduction of the index sets $R(\alpha)$ and $S(\alpha)$ allows us to state the optimality conditions in the following form (see, e.g., Lucidi et al. 2007).

Proposition 3 *A feasible point α^* is a solution of (24) if and only if*

$$\max_{i \in R(\alpha^\star)} \left\{ -\frac{(\nabla f(\alpha^\star))_i}{y_i} \right\} \leq \min_{j \in S(\alpha^\star)} \left\{ -\frac{(\nabla f(\alpha^\star))_j}{y_j} \right\}. \tag{27}$$

Given a feasible point $\bar{\alpha}$, which is not a solution of problem (24), a pair $i \in R(\bar{\alpha})$, $j \in S(\bar{\alpha})$ such that

$$\left\{ -\frac{(\nabla f(\bar{\alpha}))_i}{y_i} \right\} > \left\{ -\frac{(\nabla f(\bar{\alpha}))_j}{y_j} \right\}$$

is said to be a *violating pair*.

Given a violating pair (i, j), let us consider the direction $d^{i,j}$ with two nonzero elements defined as follows

$$d_h^{i,j} = \begin{cases} 1/y_i & \text{if } h = i \\ -1/y_j & \text{if } h = j \\ 0 & \text{otherwise.} \end{cases}$$

It can be easily shown that $d^{i,j}$ is a feasible direction at $\bar{\alpha}$ and we have $\nabla f(\bar{\alpha})^T d^{i,j} < 0$, i.e., $d^{i,j}$ is a descent direction. This implies that the selection of a violating pair of an SMO-type algorithm implies a strict decrease of the objective function. However, the use of generic violating pairs as working sets is not sufficient to guarantee convergence properties of the sequence generated by a decomposition algorithm.

A convergent SMO algorithm can be defined using as indices of the working set those corresponding to the"maximal violation" of the KKT conditions. More specifically, given again a feasible point α which is not a solution of problem (24), let us define

$$I(\alpha) = \left\{ i : i \in \arg \max_{i \in R(\alpha)} \left\{ -\frac{(\nabla f(\alpha))_i}{y_i} \right\} \right\}$$
$$J(\alpha) = \left\{ j : j \in \arg \min_{j \in S(\alpha)} \left\{ -\frac{(\nabla f(\alpha))_j}{y_j} \right\} \right\}$$

Taking into account the KKT conditions as stated in (27), a pair $i \in I(\alpha)$, $j \in J(\alpha)$ most violates the optimality conditions, and therefore, it is said to be a *maximal violating pair*. Note that the selection of the maximal violating pair involves $O(l)$ operations. An SMO-type algorithm using maximal violating pairs as working sets is usually called *most violating pair* (MVP) algorithm which is formally described below.

SMO-MVP Algorithm

Data. The starting point $\alpha^0 = 0$ and the gradient $\nabla f(\alpha^0) = -e$.

Inizialization. Set $k = 0$.

While (the stopping criterion is not satisfied)

1. select $i \in I(\alpha^k)$, $j \in J(\alpha^k)$, and set $W = \{i, j\}$;
2. compute analytically a solution $\alpha^* = \begin{pmatrix} \alpha_i^\star & \alpha_j^\star \end{pmatrix}^T$ of (25);
3. set $\alpha_h^{k+1} = \begin{cases} \alpha_i^* & for \text{ h=i} \\ \alpha_j^* & for \text{ h=j} \\ \alpha_h^k & otherwise; \end{cases}$
4. set $\nabla f(\alpha^{k+1}) = \nabla f(\alpha^k) + (\alpha_i^{k+1} - \alpha_i^k) Q_i + (\alpha_j^{k+1} - \alpha_j^k) Q_j$;
5. set $k = k + 1$.

end while

Return $\alpha^* = \alpha^k$

The scheme requires storing a vector of size l (the gradient $\nabla f(\alpha^k)$) and *to get* two columns, Q_i and Q_j, of the matrix Q.

We remark that the condition on the working set selection rule, i.e., $i \in I(\alpha^k)$, $j \in J(\alpha^k)$, can be viewed as a *Gauss-Soutwell* rule, since it is based on the maximum violation of the optimality conditions. It can be proved (see Lin 2001b, 2002a) that SMO-MVP Algorithm is globally convergent provided that the Hessian matrix Q is positive semidefinite.

A usual requirement to establish convergence properties in the context of a decomposition strategy is that

$$\lim_{k \to \infty} \left(\alpha^{k+1} - \alpha^k \right) = 0. \tag{28}$$

Indeed, in a decomposition method, at the end of each iteration k, only the satisfaction of the optimality conditions with respect to the variables associated to W is ensured. Therefore, to get convergence towards KKT points, it may be necessary to ensure that consecutive points, which are solutions of the corresponding subproblems, tend to the same limit point.

It can be proved (Lin 2002a) that SMO algorithms guarantee property (28) (the proof fully exploits that the subproblems are convex, quadratic problems into two variables).

The global convergence result of SMO algorithms can be obtained even using working set rules different from that selecting the maximal violating pair. For instance, the so-called *constant-factor violating pair* rule (Chen et al. 2006) guarantees global convergence properties of the SMO algorithm adopting it, and requires to select any violating pair $u \in R(\alpha^k)$, $v \in S(\alpha^k)$ such that

$$\frac{(\nabla f(\alpha^k))_u}{y_u} - \frac{(\nabla f(\alpha^k))_v}{y_v} \leq \sigma \left(\frac{(\nabla f(\alpha^k))_i}{y_i} - \frac{(\nabla f(\alpha^k))_j}{y_j} \right), \tag{29}$$

where $0 < \sigma \leq 1$ and (i, j) is a maximal violating pair.

The SMO-MVP algorithm is globally convergent and is based on first order information, since the maximal violating pair is related to the minimization of the first order approximation:

$$f(\alpha^k + d) \simeq f(\alpha^k) + \nabla f(\alpha^k)^T d.$$

An SMO algorithm using second order information has been proposed in Fan et al. (2005), where the designed working set selection rule takes into account that f is quadratic and we can write

$$f(\alpha^k + d) = f(\alpha^k) + \nabla f(\alpha^k)^T d + \frac{1}{2} d^T Q d. \tag{30}$$

In particular, the working set selection rule of Fan et al. (2005) exploits second order information using (30), requires $O(l)$ operations, and provides a pair defining the working set which is a constant-factor violating pair. Then, the resulting SMO algorithms, based on second order information, is globally convergent.

Other convergent SMO algorithms, not based on the MVP selection rule, have been proposed in Chang et al. (2000), Lin et al. (2009), and Lucidi et al. (2007).

We conclude the analysis of SMO algorithms focusing on the stopping criterion. To this aim let us introduce the functions $m(\alpha)$, $M(\alpha)$:

$$m(\alpha) = \begin{cases} \max\limits_{h \in R(\alpha)} -\dfrac{(\nabla f(\alpha))_h}{y_h} & \text{if } R(\alpha) \neq \emptyset \\ -\infty & \text{otherwise} \end{cases}$$

$$M(\alpha) = \begin{cases} \min\limits_{h \in S(\alpha)} -\dfrac{(\nabla f(\alpha))_h}{y_h} & \text{if } S(\alpha) \neq \emptyset \\ +\infty & \text{otherwise,} \end{cases}$$

where $R(\alpha)$ and $S(\alpha)$ are the index sets previously defined. From the definitions of $m(\alpha)$ and $M(\alpha)$, and using Proposition 3, it follows that $\bar{\alpha}$ is solution of (24) if and only if $m(\bar{\alpha}) \leq M(\bar{\alpha})$.

Let us consider a sequence of feasible points $\{\alpha^k\}$ converging to a solution $\bar{\alpha}$. At each iteration k, if α^k is not a solution then (using again Proposition 3) we have $m(\alpha^k) > M(\alpha^k)$.

Therefore, one of the adopted stopping criteria is

$$m(\alpha^k) \leq M(\alpha^k) + \epsilon, \tag{31}$$

where $\epsilon > 0$.

Note that the functions $m(\alpha)$ and $M(\alpha)$ are not continuous. Indeed, even assuming $\alpha^k \to \bar{\alpha}$ for $k \to \infty$, it may happen that $R(\alpha^k) \neq R(\bar{\alpha})$ or $S(\alpha^k) \neq S(\bar{\alpha})$ for k sufficiently large. However, it can be proved (Lin 2002b) that an SMO Algorithm using the constant-factor violating pair rule generates a sequence $\{\alpha^k\}$ such that $m(\alpha^k) - M(\alpha^k) \to 0$ for $k \to \infty$. Hence, for any $\epsilon > 0$, an SMO algorithm of this type satisfies the stopping criterion (31) in a finite number of iterations. To our knowledge, this finite convergence result has not been proved for other asymptotically convergent SMO algorithms not based on the constant-factor violating pair rule.

6.2 General decomposition algorithms

In this section we briefly present decomposition algorithms using working sets of size greater than two. To this aim we will refer to the decomposition framework previously defined. The distinguishing features of the decomposition algorithms are:

(a) the working set selection rule; and
(b) the iterative method used to solve the quadratic programming subproblem.

The dimension of the subproblems is usually on the order of ten variables. A working set selection rule, based on the violation of the optimality conditions of Proposition 3, has been proposed in Joachims (1999) and analyzed in Lin (2001b). The rule includes, as particular case, the one selecting the most violating pair and used by SMO-MVP algorithm. Let $q \geq 2$ be an even integer defining the size of the working set W. The working set selection rule is the following.

(i) Select $q/2$ indices in $R(\alpha^k)$ sequentially so that

$$\left\{-\frac{(\nabla f(\alpha^k))_{i_1}}{y_{i_1}}\right\} \geq \left\{-\frac{(\nabla f(\alpha^k))_{i_2}}{y_{i_2}}\right\} \geq \cdots \geq \left\{-\frac{(\nabla f(\alpha^k))_{i_{q/2}}}{y_{i_{q/2}}}\right\}$$

with $i_1 \in I(\alpha_k)$.

(ii) Select $q/2$ indices in $S(\alpha^k)$ sequentially so that

$$\left\{-\frac{(\nabla f(\alpha^k))_{j_1}}{y_{j_1}}\right\} \leq \left\{-\frac{(\nabla f(\alpha^k))_{j_2}}{y_{j_2}}\right\} \leq \cdots \leq \left\{-\frac{(\nabla f(\alpha^k))_{j_{q/2}}}{y_{j_{q/2}}}\right\}$$

with $j_1 \in J(\alpha_k)$.

(iii) Set $W = \{i_1, i_2, \ldots, i_{q/2}, j_1, j_2, \ldots, j_{q/2}\}$.

Note that the working set rule employed by the SMO-MVP algorithm is a particular case of the above rule, with $q = 2$. The asymptotic convergence of the decomposition algorithm based on the above working set rule and on the computation of the exact solution of the subproblem has been established in Lin (2001b) under the assumption that the objective function is strictly convex with respect to block components of cardinality less than or equal to q. This assumption is used to guarantee condition (28), but it may not hold, for instance, if some training data are the same. A proximal point-based modification of the subproblem has been proposed in Palagi and Sciandrone

(2005), and the global convergence of the decomposition algorithm using the above working set selection rule has been proved without strict convexity assumptions on the objective function.

Remark 2 We observe that the above working set selection rule (see (i)–(ii)) requires considering subproblem variables that mostly violate (in a decreasing order) the optimality conditions. This guarantees global convergence, but the degree of freedom for selecting the whole working set is limited. An open theoretical question concerns the convergence of a decomposition algorithm where the working set, besides the most violating pair, includes other arbitrary indices. This issue is very important to exploit the use of a *caching* technique that allocates some memory (the cache) to store the recently used columns of the Hessian matrix, thus avoiding in some cases the recomputation of these columns. To minimize the number of kernel evaluations and to reduce the computational time, it is convenient to select working sets containing as many elements corresponding to columns stored in the cache memory as possible. However, to guarantee the global convergence of a decomposition method, the working set selection cannot be completely arbitrary. The study of decomposition methods specifically designed to couple both the theoretical aspects of convergence and an efficient use of a caching strategy has motivated some works (see, e.g., Glasmachers and Igel 2006; Lin et al. 2009; Lucidi et al. 2009).

Concerning point (b), we observe that a closed form of the solution of the subproblem whose dimension is greater than two is not available, and this motivates the need to adopt an iterative method. In Joachims (1999) a primal-dual interior-point solver is used to solve the quadratic programming subproblems.

Gradient projection methods are suitable methods since they consist in a sequence of projections onto the feasible region that are inexpensive due to the special structure of the feasible set of (25). In fact, a projection onto the feasible set can be performed by efficient algorithms like those proposed in Dai and Fletcher (2006), Kiwiel (2008), and Pardalos and Kovoor (1990). Gradient projection methods for SVM have been proposed in Dai and Fletcher (2006) and Serafini and Zanni (2005).

Finally, the approach proposed in Mangasarian and Musicant (1999), where the square of the bias term is added to the objective function, leads by the Wolfe dual to a quadratic programming problem with only box constraints, called *Bound-constrained* SVM formulation (BSVM). In Hsu and Lin (2002b), this simpler formulation has been considered, suitable working set selection rules have been defined, and the software TRON Lin and Morè (1999), designed for large sparse bound-constrained problems, has been adapted to solve small (say of dimension 10) fully dense subproblems.

In Hsieh et al. (2008), by exploiting the bound-constrained formulation for the specific class of linear SVM, a dual coordinate descent algorithm has been defined where the dual variables are updated once at a time. The subproblem is solved analytically, the algorithm converges with convergence rate at least linear, and obtains an ϵ-accurate solution in $O(\log(1/\epsilon))$ iterations. A parallel version has been defined in Chiang et al. (2016). Also in Glasmachers and Dogan (2013) an adaptive coordinate selection has been introduced that does not select all coordinates equally often for optimization. Instead, the relative frequencies of coordinates are subject to online adaptation leading to a significant speedup.

7 Interior point methods

Interior point methods are a valuable option for solving convex quadratic optimization problems of the form

$$\begin{aligned} \min_{z} \ & \tfrac{1}{2} z^T Q z + c^T z \\ \text{s.t.} \ & Az = b \\ & 0 \le z \le u. \end{aligned} \tag{32}$$

Primal-dual interior point methods consider at each step a perturbed version of the (necessary and sufficient) primal dual optimality conditions,

$$Az = b \tag{33}$$

$$-Qz + A^T \lambda + s - v = -c \tag{34}$$

$$ZSe = \mu e \tag{35}$$

$$(U - Z)Ve = \mu e \tag{36}$$

where $S = \text{Diag(s)}$, $\text{V} = \text{Diag(v)}$, $\text{Z} = \text{Diag(z)}$, $\text{U} = \text{Diag(u)}$, and solve this system by applying the Newton method, i.e., compute the search direction $(\Delta z, \Delta\lambda, \Delta s, \Delta v)$ by solving:

$$\begin{pmatrix} A & 0 & 0 & 0 \\ -Q & A^T & I & -I \\ S & 0 & Z & 0 \\ -V & 0 & 0 & U - Z \end{pmatrix} \begin{pmatrix} \Delta z \\ \Delta\lambda \\ \Delta s \\ \Delta v \end{pmatrix} = \begin{pmatrix} -r_z \\ -r_\lambda \\ -r_s \\ -r_v \end{pmatrix} \tag{37}$$

for suitable residuals. The variables Δs and Δv can be eliminated, obtaining the augmented system:

$$\begin{pmatrix} -(Q + \Theta^{-1}) & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \Delta z \\ \Delta\lambda \end{pmatrix} = \begin{pmatrix} -r_c \\ -r_b \end{pmatrix} \tag{38}$$

where $\Theta \equiv Z^{-1}S + (U - Z)^{-1}V$ and r_c and r_b are suitable residuals. Finally Δz is eliminated, ending up with the normal equations, that require calculating

$$M \equiv A(Q + \Theta^{-1})^{-1} A^T \tag{39}$$

and factorizing it to solve $M\Delta\lambda = -\hat{r}_b$.

The advantage of interior point methods is that the number of iterations is almost independent of the size of the problem, whereas the main computational burden at each iteration is the solution of system (38). IPMs have been applied to linear SVM training in Ferris and Munson (2002), Fine and Scheinberg (2001), Gertz and Griffin (2010), Goldfarb and Scheinberg (2008), Woodsend and Gondzio (2009), Woodsend and Gondzio (2011). The main differences are the formulations of the problem considered and the linear algebra tools used in order to solve the corresponding system (38).

In Ferris and Munson (2002), different versions of the primal-dual pair for SVM are considered: the standard one, given by (5) and (8), is one where the bias term is included in the objective function:

$$\begin{array}{ll} \min\limits_{w,b,\xi} & = \dfrac{1}{2}\|w,b\|^2 + C\sum\limits_{i=1}^{l}\xi_i \\ \text{s.t.} & y^i\left[w^T x^i + b\right] - 1 + \xi_i \geq 0 \qquad i = 1,\ldots,l \\ & \xi_i \geq 0 \qquad i = 1,\ldots,l \end{array} \tag{40}$$

with corresponding dual

$$\begin{array}{l} \min\limits_{\alpha} \ \frac{1}{2}\alpha^T Y X^T X Y \alpha + \frac{1}{2}\alpha^T Y e e^T Y \alpha - e^T \alpha \\ \qquad 0 \leq \alpha \leq Ce, \end{array} \tag{41}$$

and

$$\begin{array}{ll} \min\limits_{w,b,\xi} & = \dfrac{1}{2}\|w,b\|^2 + \dfrac{C}{2}\|\xi\|_2^2 \\ \text{s.t.} & y^i\left[w^T x^i + b\right] - 1 + \xi_i \geq 0 \qquad i = 1,\ldots,l \\ & \xi_i \geq 0 \qquad i = 1,\ldots,l \end{array} \tag{42}$$

with corresponding dual

$$\begin{array}{l} \min\limits_{\alpha} \ \frac{1}{2C}\alpha^T \alpha + \frac{1}{2}\alpha^T Y X^T X Y \alpha + \frac{1}{2}\alpha^T Y e e^T Y \alpha - e^T \alpha \\ \qquad \alpha \geq 0. \end{array} \tag{43}$$

The simplest situation for IPMs is problem (43), where the linear system (38) simplifies into

$$(C + RHR^T)\Delta\lambda = r_1, \tag{44}$$

with $C = \frac{1}{2C}I + \Theta^{-1}$, $H = I$ and $R = Y[X^T - e]$. The matrix $C + RR^T$ can be easily inverted using the Sherman-Morrison-Woodbury formula Golub and Loan (1996):

$$(C + RHR^T)^{-1} = C^{-1} - C^{-1}R\left(H^{-1} + R^T C^{-1} R\right)^{-1} R^T C^{-1} \tag{45}$$

where C^{-1} and H^{-1} are diagonal and positive definite and the matrix $H^{-1} + R^T C^{-1} R$ is of size n and needs to be computed only once per iteration. The approach can be extended by using some (slightly more complex) variations of this formula for solving (41), whereas for solving problem (8) some proximal point is needed.

In Gertz and Griffin (2010), an interior point method is defined for solving the primal problem (5). In this case, we consider the dual variables α associated to the classification constraints with the corresponding slack variables s, and μ the multipliers associated to the nonnegativity constraints on the ξ vector. In this case, the primal dual optimality conditions lead to the following reduced system:

$$\begin{pmatrix} I & 0 & -X^T Y \\ 0 & 0 & y^T \\ YX & -y & \Omega \end{pmatrix} \begin{pmatrix} \Delta w \\ \Delta b \\ \Delta \alpha \end{pmatrix} = \begin{pmatrix} -r_w \\ -r_b \\ -r_\Omega \end{pmatrix}, \tag{46}$$

where $\Omega = \text{Diag}(\alpha)^{-1}\text{S} + \text{Diag}(\mu)^{-1}\text{Diag}(\xi)$. By row elimination, system (46) can be transformed into

$$\begin{pmatrix} I + X^T Y \Omega^{-1} Y X & -X^T Y \Omega^{-1} y & 0 \\ -y^T \Omega^{-1} Y X & y^T \Omega^{-1} y & 0 \\ YX & -y & \Omega \end{pmatrix} \begin{pmatrix} \Delta w \\ \Delta b \\ \Delta \alpha \end{pmatrix} = \begin{pmatrix} -\hat{r}_w \\ -\hat{r}_b \\ -\hat{r}_\Theta \end{pmatrix}. \tag{47}$$

Finally, this system can be reduced into

$$(I + X^T Y \Omega^{-1} Y X - \frac{1}{y^T \Omega^{-1} y} y_d^T y_d) \Delta w = -\tilde{r}_w \tag{48}$$

$$\Delta b = \frac{1}{\sigma}(-\hat{r}_b + y_d^T \Delta w) \tag{49}$$

where $y_d = X^T Y \Omega^{-1} y$. The main cost in solving this system is computing and factorizing the matrix $I + X^T Y \Omega^{-1} Y X - \frac{1}{y^T \Omega^{-1} y} y_d^T y_d$. In Gertz and Griffin (2010), the idea is to solve system (48) by a preconditioned linear conjugate gradient that requires only a mechanism for computing matrix-vector products of the form Mx, and they define a new preconditioner exploiting the structure of problem (5). The method is applicable when the number of features n is relatively large. Both the methods proposed in Ferris and Munson (2002) and Gertz and Griffin (2010) exploit the Sherman–Morrison–Woodbury formula, but it has been shown (see Goldfarb and Scheinberg 2008) that this approach leads to numerical issues, especially when the matrix Θ (or Ω) is ill-conditioned and if there is near degeneracy in the matrix XY, which occurs if there are multiple samples close to the separating hyperplane.

In Goldfarb and Scheinberg (2008), an alternative approach is proposed for solving problem (8) (note that in this section we stick to the notation α instead of λ) based on Product Form Cholesky Factorization. Here it is assumed that the matrix $YX^T XY$ can be approximated by a low rank matrix VV^T, and an efficient and numerically stable Cholesky Factorization of the matrix $VV^T + \text{Diag}(\alpha)^{-1}\text{S} + (\text{Diag}(\text{Ce}) - \text{Diag}(\alpha))^{-1}\text{Z}$ is computed. The advantage with respect to methods using the SMW formula is that the LDL^T Cholesky factorization of the IPM normal equation matrix enjoys the property that the matrix L is stable even if D becomes ill-conditioned.

A different approach to overcoming the numerical issues related to the SMW formula is the one described in Woodsend and Gondzio (2011), where a primal-dual interior point method is proposed based on a different formulation of the training problem. In particular, the authors consider the dual formulation (8), and include the substitution

$$w = XY\alpha$$

in order to get the following primal-dual formulation:

$$\begin{aligned} \min_{w,\alpha} \ & \tfrac{1}{2} w^T w - e^T \alpha \\ \text{s.t.} \ & w - XY\alpha = 0 \\ & y^T \alpha = 0 \\ & 0 \le \alpha \le Ce. \end{aligned} \tag{50}$$

In order to apply standard interior point methods, that require all the variables to be bounded, some bounds are added on the variable w, so that the problem to be solved becomes:

$$\begin{aligned} \min_{w,\alpha} \ & \tfrac{1}{2} w^T w - e^T \alpha \\ \text{s.t.} \ & w - XY\alpha = 0 \\ & y^T \alpha = 0 \\ & 0 \le w \le u_w \\ & 0 \le \alpha \le Ce. \end{aligned} \tag{51}$$

The advantage of this formulation is that the objective function matrix Q is sparse, since it only has a non zero diagonal block corresponding to w (that is the identity matrix). Specializing the matrix M in (39) for this specific problem, if we define

$$\begin{aligned} \Theta_w^{-1} &= (W^{-1} S_w + (\mathrm{Diag}(\mathrm{u_w}) - \mathrm{W})^{-1} \mathrm{V_w}) \\ \Theta_\alpha^{-1} &= (\mathrm{Diag}(\alpha)^{-1} \mathrm{S}_\alpha + (\mathrm{Diag(Ce)} - \mathrm{Diag}(\alpha))^{-1} \mathrm{V}_\alpha) \end{aligned}$$

we get

$$\begin{aligned} M &= A \left(Q + \Theta^{-1} \right)^{-1} A^T \\ &= \begin{pmatrix} \left(I_n + \Theta_w^{-1}\right) + XY\Theta_\alpha YX^T & -XY\Theta_\alpha y \\ -y^T \Theta_w YX^T & y^T \Theta_\alpha y \end{pmatrix}. \end{aligned} \tag{52}$$

Building the matrix M is the most expensive operation, of order $\mathcal{O}(l(n+1)^2)$ while inverting the matrix is of order $\mathcal{O}((n+1)^3)$. In order to get the optimal hyperplane, it is possible to directly get the bias b since it is the element of λ corresponding to the constraint $y^T \alpha = 0$.

The method uses as stopping condition the stability of the set of support vectors monitored by measuring the change in the angle ϕ of the normal to the hyperplane between iterations i and $i-1$:

$$\cos(\phi) = \frac{(w^{(i-1)})^T w^{(i)}}{\|w^{(i-1)}\| \|w^{(i)}\|}. \tag{53}$$

Furthermore the number of iterations of IPMs can be reduced by using multiple correctors (that all use the same factorization of M) to improve the centrality of the current

point, and also an accurate estimate of the bounds on w can help to speed up the approach.

A parallel version of this algorithm has been introduced in Woodsend and Gondzio (2009).

8 Software

Most of the methods described in this survey are open source and can be downloaded. Here we report for the reader's convenience a list of the algorithms and the link to the corresponding software.

Algorithms for solving SVM in the primal:

1. the stochastic sub-gradient methods described in Shalev-Shwartz et al. (2011) are implemented in the software PEGASOS that can be downloaded from https://www.cs.huji.ac.il/~shais/code/index.html
2. The cutting plane algorithm proposed in Joachims (2006) is implemented in the software SVM^{perf} that can be downloaded from http://www.cs.cornell.edu/people/tj/svm_light/svm_perf.html

Interior point Methods:

1. The methods described in Ferris and Munson (2002); Gertz and Griffin (2010) are part of the software for quadratic programming OOQP downloadable at http://pages.cs.wisc.edu/~swright/ooqp/
2. The method described in Woodsend and Gondzio (2011) is implemented in the software SVM-OOPS that can be downloaded at http://www.maths.ed.ac.uk/ERGO/svm-oops/

Decomposition methods for solving SVM in the dual:

1. SMO-type algorithms and general decomposition algorithms have been implemented both in the software SVM^{light} that can be downloaded at http://svmlight.joachims.org/ and in the software LIBSVM that can be downloaded at https://www.csie.ntu.edu.tw/~cjlin/libsvm/index.html.
2. An efficient library for linear classification is implemented in the software LIBLINEAR that can be downloaded at https://www.csie.ntu.edu.tw/~cjlin/liblinear/

9 Concluding remarks

In this paper we have presented an overview of the nonlinear optimization methods for SVM training, which typically involves convex programming problems, whose difficulties are related to the dimensions, i.e., to the number of training instances, and/or to the number of features. We have considered different equivalent formulations, pointing out the main theoretical and computational difficulties of the problems. We have described the most important and used optimization methods for SVM training, and we have discussed how the algorithms have been specifically designed and adapted to take into account the structure of the considered problems.

In our analysis we have limited ourselves to models and algorithms for binary classification since by nature SVM are mainly binary classifiers. Although the paper is a survey, in a field as vast as SVM we had to leave out several related important topics, such as Multiclass-Classification, One-Class SVM, Support Vector Regression, Semi-Supervised SVM, and Online Incremental SVM. However, we believe that most of the concepts, models and algorithms developed for SVM binary classification may represent a sound and useful basis to analyze the other classes of SVM models.

Appendix A: Proof of existence and uniqueness of the optimal hyperplane

The idea underlying the proof of existence and uniqueness of the optimal hyperplane is based on the following steps:

– for each separating hyperplane $H(w, b)$, there exists a separating hyperplane $H(\hat{w}, \hat{b})$ such that

$$\frac{1}{\|w\|} \leq \rho(w, b) \leq \frac{1}{\|\hat{w}\|};$$

– the above condition implies that problem (2), i.e.,

$$\begin{array}{ll} \max\limits_{w\in\Re^n, b\in\Re} & \rho(w, b) \\ \text{s.t.} & w^T x^i + b \geq 1, \quad \forall x^i \in A \\ & w^T x^j + b \leq -1, \quad \forall x^j \in B \end{array}$$

admits solution provided that the following problem

$$\begin{array}{ll} \max\limits_{w\in\Re^n, b\in\Re} & \frac{1}{\|w\|} \\ \text{s.t.} & w^T x^i + b \geq 1, \quad \forall x^i \in A \\ & w^T x^j + b \leq -1, \quad \forall x^j \in B \end{array} \tag{54}$$

admits solution;

– problem (54) is obviously equivalent to

$$\begin{array}{ll} \min\limits_{w\in\Re^n, b\in\Re} & \|w\|^2 \\ \text{s.t.} & w^T x^i + b \geq 1, \quad \forall x^i \in A \\ & w^T x^j + b \leq -1, \quad \forall x^j \in B; \end{array} \tag{55}$$

– then we prove that (55) admits a unique solution, which is also the unique solution of (2).

Lemma 1 *Let $H(\hat{w}, \hat{b})$ be a separating hyperplane. Then*

$$\rho(\hat{w}, \hat{b}) \geq \frac{1}{\|\hat{w}\|}.$$

Proof Since

$$|\hat{w}^T x^\ell + \hat{b}| \geq 1, \quad \forall x^\ell \in A \cup B,$$

it follows

$$\rho(\hat{w}, \hat{b}) = \min_{x^\ell \in A \cup B} \left\{ \frac{|\hat{w}^T x^\ell + \hat{b}|}{\|\hat{w}\|} \right\} \geq \frac{1}{\|\hat{w}\|}.$$

□

Lemma 2 *Given any separating hyperplane $H(\hat{w}, \hat{b})$, there exists a separating hyperplane $H(\bar{w}, \bar{b})$ such that*

$$\rho(\hat{w}, \hat{b}) \leq \rho(\bar{w}, \bar{b}) = \frac{1}{\|\bar{w}\|}. \tag{56}$$

Moreover there exist two points $x^+ \in A$ and $x^- \in B$ such that

$$\begin{aligned} \bar{w}^T x^+ + \bar{b} &= 1 \\ \bar{w}^T x^- + \bar{b} &= -1 \end{aligned} \tag{57}$$

Proof Let $\hat{x}^i \in A$ and $\hat{x}^j \in B$ be the closest points to $H(\hat{w}, \hat{b})$, that is, the two points such that

$$\begin{aligned} \hat{d}_i &= \frac{|\hat{w}^T \hat{x}^i + \hat{b}|}{\|\hat{w}\|} \leq \frac{|\hat{w}^T x^i + \hat{b}|}{\|\hat{w}\|}, \quad \forall x^i \in A \\ \hat{d}_j &= \frac{|\hat{w}^T \hat{x}^j + \hat{b}|}{\|\hat{w}\|} \leq \frac{|\hat{w}^T x^j + \hat{b}|}{\|\hat{w}\|}, \quad \forall x^j \in B \end{aligned} \tag{58}$$

from which it follows

$$\rho(\hat{w}, \hat{b}) = \min\{\hat{d}_i, \hat{d}_j\} \leq \frac{1}{2}(\hat{d}_i + \hat{d}_j) = \frac{\hat{w}^T(\hat{x}^i - \hat{x}^j)}{2\|\hat{w}\|}. \tag{59}$$

Let us consider the numbers α and β such that

$$\begin{aligned} \alpha \hat{w}^T \hat{x}^i + \beta &= 1 \\ \alpha \hat{w}^T \hat{x}^j + \beta &= -1 \end{aligned} \tag{60}$$

that is, the numbers

$$\alpha = \frac{2}{\hat{w}^T(\hat{x}^i - \hat{x}^j)}, \qquad \beta = -\frac{\hat{w}^T(\hat{x}^i + \hat{x}^j)}{\hat{w}^T(\hat{x}_i - \hat{x}_j)}.$$

It can be easily verified that $0 < \alpha \leq 1$. We will show that the hyperplane $H(\bar{w}, \bar{b}) \equiv H(\alpha\hat{w}, \beta)$ is a separating hyperplane for the sets A and B, and it is such that (56) holds. Indeed, using (58), we have

$$\begin{aligned} \hat{w}^T x^i &\geq \hat{w}^T \hat{x}^i, \quad \forall x^i \in A \\ \hat{w}^T x^j &\leq \hat{w}^T \hat{x}^j, \quad \forall x^j \in B. \end{aligned}$$

As $\alpha > 0$, we can write

$$\begin{aligned} \alpha \hat{w}^T x^i + \beta &\geq \alpha \hat{w}^T \hat{x}^i + \beta = 1, \qquad \forall x^i \in A \\ \alpha \hat{w}^T x^j + \beta &\leq \alpha \hat{w}^T \hat{x}^j + \beta = -1, \quad \forall x^j \in B \end{aligned} \tag{61}$$

from which we get that $\bar{w}$ and $\bar{b}$ satisifies (1), and hence, that $H(\bar{w}, \bar{b})$ is a separating hyperplane for the sets A and B.

Furthermore, taking into account (61) and the value of α, we have

$$\rho(\bar{w}, \bar{b}) = \min_{x^\ell \in A \cup B} \left\{ \frac{|\bar{w}^T x^\ell + \bar{b}|}{\|\bar{w}\|} \right\} = \frac{1}{\|\bar{w}\|} = \frac{1}{\alpha \|\hat{w}\|} = \frac{\hat{w}^T (\hat{x}^i - \hat{x}^j)}{2\|\hat{w}\|}.$$

Condition (56) follows from the above equality and (59). Using (60) we obtain that (57) holds with $x^+ = \hat{x}^i$ and $x^- = \hat{x}^j$. □

Proposition 4 *The following problem*

$$\begin{aligned} &\min \ \|w\|^2 \\ &\text{t.c.} \ \ w^T x^i + b \geq 1, \qquad \forall x^i \in A \\ &\qquad w^T x^j + b \leq -1, \quad \forall x^j \in B \end{aligned} \tag{62}$$

admits a unique solution $(w^\star, b^\star)$.

Proof Let $\mathcal{F}$ the feasible set, that is,

$$\mathcal{F} = \{(w, b) \in \Re^n \times \Re : \ w^T x^i + b \geq 1, \forall x^i \in A, \ w^T x^j + b \leq -1, \forall x^j \in B\}.$$

Given any $(w_o, b_o) \in \mathcal{F}$, let us consider the level set

$$\mathcal{L}_o = \{(w, b) \in \mathcal{F} : \ \|w\|^2 \leq \|w_o\|^2\}.$$

The set $\mathcal{L}_o$ is closed, and we will show that is also bounded. To this aim, assume by contradiction that there exists an unbounded sequence $\{(w_k, b_k)\}$ belonging to $\mathcal{L}_o$. Since $\|w_k\| \leq \|w_o\|, \forall k$, we must have $|b_k| \to \infty$. For any k we can write

$$\begin{aligned} w_k^T x^i + b_k &\geq 1, \qquad \forall x^i \in A \\ w_k^T x^j + b_k &\leq -1, \quad \forall x^j \in B \end{aligned}$$

and hence, as $|b_k| \to \infty$, for k sufficiently large, we have $\|w_k\|^2 > \|w_o\|^2$, and this contradicts the fact that $\{(w_k, b_k)\}$ belongs to $\mathcal{L}_o$. Thus $\mathcal{L}_o$ is a compact set.

Weirstrass's theorem implies that the function $\|w\|^2$ admits a minimum $(w^\star, b^\star)$ on $\mathcal{L}_o$, and hence, on $\mathcal{F}$. As consequence, $(w^\star, b^\star)$ is a solution of (62).

In order to prove that $(w^\star, b^\star)$ is the unique solution, by contradiction assume that there exists a pair $(\bar{w}, \bar{b}) \in \mathcal{F}$, $(\bar{w}, \bar{b}) \neq (w^\star, b^\star)$, such that $\|\bar{w}\|^2 = \|w^\star\|^2$. Suppose $\bar{w} \neq w^\star$. The set $\mathcal{F}$ is convex, so that

$$\lambda(w^\star, b^\star) + (1-\lambda)(\bar{w}, \bar{b}) \in \mathcal{F}, \quad \forall \lambda \in [0, 1].$$

Since $\|w\|^2$ is a strictly convex function, for any $\lambda \in (0, 1)$ it follows

$$\|\lambda w^\star + (1-\lambda)\bar{w}\|^2 < \lambda\|w^\star\|^2 + (1-\lambda)\|\bar{w}\|^2.$$

Getting $\lambda = 1/2$, which corresponds to consider the pair $(\tilde{w}, \tilde{b}) \equiv \left(\frac{1}{2}w^\star + \frac{1}{2}\bar{w}, \frac{1}{2}b^\star + \frac{1}{2}\bar{b}\right)$, we have $(\tilde{w}, \tilde{b}) \in \mathcal{F}$ and

$$\|\tilde{w}\|^2 < \frac{1}{2}\|w^\star\|^2 + \frac{1}{2}\|\bar{w}\|^2 = \|w^\star\|^2,$$

and this contradicts the fact that $(w^\star, b^\star)$ is a global minimum. Therefore, we must have $\bar{w} \equiv w^\star$.

Assume $b^\star > \bar{b}$ (the case $b^\star < \bar{b}$ is analogous), and consider the point $\hat{x}^i \in A$ such that

$$w^{\star T}\hat{x}^i + b^\star = 1$$

(the existence of such a point follows from (57) of Lemma 2). We have

$$1 = w^{\star T}\hat{x}^i + b^\star = \bar{w}^T\hat{x}^i + b^\star > \bar{w}^T\hat{x}^i + \bar{b}$$

and this contradicts the fact that $\bar{w}^T x^i + \bar{b} \geq 1$, $\forall x^i \in A$. As consequence, we must have $\bar{b} \equiv b^\star$, and hence the uniqueness of the solution is proved. □

Proposition 5 *Let $(w^\star, b^\star)$ be the solution of (62). Then, $(w^\star, b^\star)$ is the unique solution of the following problem*

$$\begin{array}{ll} \max & \rho(w, b) \\ \text{t.c.} & w^T x^i + b \geq 1, \quad \forall x^i \in A \\ & w^T x^j + b \leq -1, \quad \forall x^j \in B \end{array} \tag{63}$$

Proof We observe that $(w^\star, b^\star)$ is the unique solution of the problem

$$\begin{array}{ll} \max & \frac{1}{\|w\|} \\ \text{t.c.} & w^T x^i + b \geq 1, \quad \forall x^i \in A \\ & w^T x^j + b \leq -1, \quad \forall x^j \in B. \end{array}$$

Lemmas 1 and 2 imply that, for any separating hyperplane $H(w, b)$, we have

$$\frac{1}{\|w\|} \leq \rho(w, b) \leq \frac{1}{\|w^\star\|}$$

and hence, for the separating hyperplane $H(w^\star, b^\star)$ we obtain $\rho(w^\star, b^\star) = \dfrac{1}{\|w^\star\|}$, which implies that $H(w^\star, b^\star)$ is the optimal separating hyperplane. □

Appendix B: The Wolfe dual and its properties

Consider the convex problem

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & g(x) \leq 0 \\ & h(x) = 0 \end{array} \tag{64}$$

with $f : \Re^n \to \Re$ convex and continuously differentiable, $g : \Re^n \to \Re^m$ convex and continuously differentiable, and $h : \Re^n \to \Re^p$ affine functions. Then its Wolfe dual is

$$\begin{array}{ll} \max\limits_{x,\lambda,\mu} & L(x, \lambda, \mu) \\ \text{s.t.} & \nabla_x L(x, \lambda, \mu) = 0 \\ & \lambda \geq 0, \end{array} \tag{65}$$

where $L(x, \lambda, \mu) = f(x) + \lambda^T g(x) + \mu^T h(x)$.

Proposition 6 *Let x^* be a global solution of problem (64) with multipliers (λ^*, μ^*). Then it is also a solution of problem (65) and there is zero duality gap, i.e., $f(x^*) = L(x^*, \lambda^*, \mu^*)$.*

Proof The point (x^*, λ^*, μ^*) is clearly feasible for problem (65) since it satisfies the KKT conditions of problem (64). Furthermore, by complementarity $((\lambda^*)^T g(x^*) = 0)$ and feasibility $(h(x^*) = 0)$

$$L(x^*, \lambda^*, \mu^*) = f(x^*) + (\lambda^*)^T g(x^*) + (\mu^*)^T h(x^*) = f(x^*)$$

so that there is zero duality gap. Furthermore, for any $\lambda \geq 0$, $\mu \in \Re^p$, by the feasibility of x^*, we have

$$L(x^*, \lambda^*, \mu^*) = f(x^*) \geq f(x^*) + \lambda^T g(x^*) + \mu^T h(x^*) = L(x^*, \lambda, \mu). \tag{66}$$

By the convexity assumptions on f and g, the nonnegativity of λ and by the linearity of h, we get that $L(\cdot, \lambda, \mu)$ is a convex function in x and hence, for any feasible (x, λ, μ), we can write

$$L(x^*, \lambda, \mu) \geq L(x, \lambda, \mu) + \nabla_x L(x, \lambda, \mu)^T (x^* - x) = L(x, \lambda, \mu), \tag{67}$$

where the last equality derives from the constraints of problem (65). By combining (66) and (67), we get

$$L(x^*, \lambda^*, \mu^*) \geq L(x, \lambda, \mu) \text{ for all } (x, \lambda, \mu) \text{ feasible for problem (65)}$$

and hence (x^*, λ^*, μ^*) is a global solution of problem (65). □

A stronger result can be proved when the primal problem is a convex quadratic programming problem defined by (6).

Proposition 7 *Let $f(x) = \frac{1}{2}x^T Qx + c^T x$, and suppose that the matrix Q is symmetric and positive semidefinite. Let $(\bar{x}, \bar{\lambda})$ be a solution of Wolfe's dual (7). Then, there exists a vector $x^\star$ (not necessarily equal to $\bar{x}$) such that*

(i) $Q(x^\star - \bar{x}) = 0$;
(ii) $x^\star$ *is a solution of problem (6); and*
(iii) x^* *is a global minimum of (6) with associated multipliers* $\bar{\lambda}$.

Proof First, we show how in this case problem (7) is a convex quadratic programming problem. In particular, problem (7) becomes for the quadratic case:

$$\max_{x,\lambda} \frac{1}{2}x^T Qx + c^T x + \lambda^T (Ax - b) \tag{68}$$

$$Qx + c + A^T \lambda = 0 \tag{69}$$

$$\lambda \geq 0. \tag{70}$$

Multiplying the constraints (68) by x^T we get

$$x^T Qx + c^T x + x^T A^T \lambda = 0,$$

which implies that the objective function (68) can be rewritten as

$$\max -\frac{1}{2}x^T Qx + c^T x - \lambda^T b = -\min \frac{1}{2}x^T Qx + \lambda^T b,$$

which shows how problem (68) is actually a convex quadratic optimization problem. For this problem, the KKT conditions are necessary and sufficient for global optimality, and, if we denote by v the multipliers of the equality constraints (69) and by z the multipliers of the constraints (70), we get that there must exist multipliers v and z such that the following conditions hold;

$$Q\bar{x} - Qv = 0 \tag{71}$$

$$b - Av - z = 0 \tag{72}$$

$$z^T \bar{\lambda} = 0 \tag{73}$$

$$z \geq 0 \tag{74}$$

$$Q\bar{x} + c + A^T \bar{\lambda} = 0 \tag{75}$$

$$\bar{\lambda} \geq 0. \tag{76}$$

The expression of z can be derived by constraints (72), and substituted in (73) and (74), implying:

$$Av - b \leq 0 \tag{77}$$

$$\bar{\lambda}^T (Av - b) = 0. \tag{78}$$

Furthermore by subtracting (71) from (75), we get

$$Qv + c + A^T \bar{\lambda} = 0. \tag{79}$$

By combining (79), (78), (77) and (76) we get that the pair $(v, \bar{\lambda})$ satisfies the KKT conditions of problem (6), and hence setting $x^* = v$ we get the thesis, keeping into account that point (i) derives from (71). □

Appendix C: Kernel characterization

Proposition 8 *Let $K : X \times X \to \Re$ be a symmetric function. Function K is a kernel if and only if the $l \times l$ matrix*

$$\left(K(x^i, x^j)\right)_{i,j=1}^{l} = \begin{pmatrix} K(x^1, x^1) & \dots & K(x^1, x^l) \\ & \vdots & \\ K(x^l, x^1) & \dots & K(x^l, x^l) \end{pmatrix}$$

is positive semidefinite for any set of training vectors $\{x^1, \dots, x^l\}$.

Proof **necessity** Symmetry derives from the symmetry of the function K. To prove positive semidefiniteness we look at the quadratic form, for any $v \in \Re^l$:

$$\begin{aligned} v^T K v &= \sum_{i=1}^{l} \sum_{j=1}^{l} v_i v_j K(x^i, x^j) = \sum_{i=1}^{l} \sum_{j=1}^{l} v_i v_j \langle \phi(x^i), \phi(x^j) \rangle \\ &= \left\langle \sum_{i=1}^{l} v_i \phi(x^i), \sum_{j=1}^{l} v_j \phi(x_j) \right\rangle \\ &= \langle z, z \rangle \geq 0 \end{aligned}$$

sufficiency Assume

$$\begin{pmatrix} K(x^1, x^1) & \dots & K(x^1, x^l) \\ & \vdots & \\ K(x^l, x^1) & \dots & K(x^l, x^l) \end{pmatrix} \succeq 0 \tag{80}$$

We need to prove that there exists a linear space $\mathcal{H}$, a function $\phi : X \to \mathcal{H}$ and a scalar product $\langle \cdot, \cdot \rangle$ defined on $\mathcal{H}$ such that $k(x, y) = \langle \phi(x), \phi(y) \rangle$ for all $x, y \in X$.
Consider the linear space

$$\mathcal{H} = lin\{K(\cdot, y) : y \in X\}$$

with the generic element $f(\cdot)$

$$f = \sum_{i=1}^{m} \alpha_i K(\cdot, x^i)$$

for any $m \in N$, with $\alpha_i \in \Re$ for $i = 1, \ldots, m$. Given two elements $f, g \in \mathcal{H}$, with $g(\cdot) = \sum_{j=1}^{m'} \beta_j K(\cdot, x^j)$, define the function $\rho : \mathcal{H} \times \mathcal{H} \to \Re$ defined as

$$\rho(f, g) = \sum_{i=1}^{m} \sum_{j=1}^{m'} \alpha_i \beta_j K(x^i, x^j)$$

It can be shown that the function ρ is a scalar product in the space $\mathcal{H}$, by showing that the following properties hold:

(i) $\rho(f, g) = \rho(g, f)$
(ii) $\rho(f^1 + f^2, g) = \rho(f^1, g) + \rho(f^2, g)$
(iii) $\rho(\lambda f, g) = \lambda \rho(f, g)$
(iv) $\rho(f, f) \geq 0$ and $\rho(f, f) = 0$ implies $f = 0$

The first three properties are a consequence of the definition of ρ and can be easily verified. We need to show property (iv). First, we observe that, given $f^1, \ldots, f^p$ in $\mathcal{H}$ the matrix with elements $\rho_{st} = \rho(f^s, f^t)$ is symmetric (thanks to property (i)) and positive semidefinite. Indeed,

$$\sum_{i=1}^{p} \sum_{j=1}^{p} \gamma_i \gamma_j \rho_{ij} = \sum_{i=1}^{p} \sum_{j=1}^{p} \gamma_i \gamma_j \rho(f^i, f^j) = \rho\left(\sum_{i=1}^{p} \gamma_i f^i, \sum_{j=1}^{p} \gamma_j f^j \right) \geq 0$$

This implies in turn that all principal minors have non negative determinant. Consider any 2×2 principal minor, with elements $\rho_{ij} = \rho(f^i, f^j)$. The nonnegativity of the determinant, and the symmetry of the matrix imply

$$\begin{aligned} &\rho(f^i, f^i)\rho(f^j, f^j) - \rho(f^i, f^j)\rho(f^j, f^i) \\ &= \rho(f^i, f^i)\rho(f^j, f^j) - \rho(f^i, f^j)^2 \geq 0 \end{aligned}$$

so that

$$\rho(f^i, f^j)^2 \leq \rho(f^i, f^i)\rho(f^j, f^j) \tag{81}$$

We note that, setting $m' = 1$, $g(\cdot) = k(\cdot, x)$, $f(x)$ can be written as

$$f(x) = \sum_{i=1}^{m} \alpha_i K(x, x^i) = \rho(K(\cdot, x), f)$$

with $K(\cdot, x) \in \mathcal{H}$. Furthermore, for any $x, y \in X$, we get

$$\rho(K(\cdot, x), K(\cdot, y)) = K(x, y)$$

Using (81) with $f^i = K(\cdot, x)$ and $f^j = f(x)$ we get

$$f(x)^2 = \rho(K(\cdot, x), f) \leq \rho(f^i, f^i)\rho(f^j, f^j) = \rho(K(\cdot, x), K(\cdot, x))\rho(f, f) = K(x, x)\rho(f, f)$$

that implies, thanks to (80), both $\rho(f, f) \geq 0$ and that if $\rho(f, f) = 0$, then $f(x)^2 \leq 0$ for all $x \in X$ and hence $f = 0$. □

References

Astorino A, Fuduli A (2015) Support vector machine polyhedral separability in semisupervised learning. J Optim Theory Appl 164:1039–1050

Bertsekas DP (1999) Nonlinear programming. Athena Scientific, Belmont

Bishop CM (2006) Pattern recognition and machine learning. Springer, Berlin

Boser BE, Guyon IM, Vapnik VN (1992) A training algorithm for optimal margin classifiers. In: Proceedings of the fifth annual workshop on computational learning theory, COLT '92. ACM, New York, pp 144–152

Byrd RH, Chin GM, Neveitt W, Nocedal J (2011) On the use of stochastic hessian information in optimization methods for machine learning. SIAM J Optim 21:977–995

Carrizosa E, Romero Morales D (2013) Supervised classification and mathematical optimization. Comput Oper Res 40:150–165

Cassioli A, Chiavaioli A, Manes C, Sciandrone M (2013) An incremental least squares algorithm for large scale linear classification. Eur J Oper Res 224:560–565

Chang CC, Hsu CW, Lin CJ (2000) The analysis of decomposition methods for support vector machines. IEEE Trans Neural Netw Learn Syst 11:1003–1008

Chang KW, Hsieh CJ, Lin CJ (2008) Coordinate descent method for large-scale l2-loss linear support vector machines. J Mach Learn Res 9:1369–1398

Chapelle O (2007) Training a support vector machine in the primal. Neural Comput 19:1155–1178

Chen PH, Fan RE, Lin CJ (2006) A study on smo-type decomposition methods for support vector machines. IEEE Trans Neural Netw 17:893–908

Chiang WL, Lee MC, Lin CJ (2016) Parallel dual coordinate descent method for large-scale linear classification in multi-core environments. In: Proceedings of the 22Nd ACM SIGKDD international conference on knowledge discovery and data mining, KDD '16. ACM, New York, pp 1485–1494

Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20:273–297

Dai HY, Fletcher R (2006) New algorithms for singly linearly constrained quadratic programs subject to lower and upper bounds. Math Programm 106:403–421

Fan RE, Chang KW, Hsieh CJ, Wang XR, Lin CJ (2008) Liblinear: a library for large linear classification. J Mach Learn Res 9:1871–1874
Fan RE, Chen PH, Lin CJ (2005) Working set selection using second order information for training support vector machines. J Mach Learn Res 6:1889–1918
Ferris MC, Munson TS (2004) Semismooth support vector machines. Math Programm B 101:185–204
Ferris MC, Munson TS (2002) Interior-point methods for massive support vector machines. SIAM J Optim 13:783–804
Fine S, Scheinberg K (2001) Efficient svm training using low-rank kernel representations. J Mach Learn Res 2:243–264
Fletcher R (1987) Practical methods of optimization, 2nd edn. Wiley, New York
Franc V, Sonnenburg S (2009) Optimized cutting plane algorithm for large-scale risk minimization. J Mach Learn Res 10:2157–2192
Franc V, Sonnenburg S (2008) Optimized cutting plane algorithm for support vector machines. In: Proceedings of the 25th international conference on machine learning, ICML '08. ACM, New York, pp 320–327
Fung G, Mangasarian OL (2001) Proximal support vector machine classifiers. In: Proceedings of the seventh ACM SIGKDD international conference on knowledge discovery and data mining, KDD '01. ACM New York, pp 77–86
Gaudioso M, Gorgone E, Labbé M, Rodríguez-Chía AM (2017) Lagrangian relaxation for svm feature selection. Comput OR 87:137–145
Gertz EM, Griffin JD (2010) Using an iterative linear solver in an interior-point method for generating support vector machines. Comput Optim Appl 47:431–453
Glasmachers T, Igel C (2006) Maximum-gain working set selection for svms. J Mach Learn Res 7:1437–1466
Goldfarb D, Scheinberg K (2008) Numerically stable ldlt factorizations in interior point methods for convex quadratic programming. IMA J Numer Anal 28:806–826
Golub GH, Van Loan CF (1996) Matrix computations, 3rd edn. Johns Hopkins University Press, Baltimore
Hsieh CJ, Chang KW, Lin CJ, Keerthi SS, Sundararajan S (2008) A dual coordinate descent method for large-scale linear svm. In: Proceedings of the 25th international conference on machine learning, ICML '08. ACM, New York, pp 408–415
Hsu CW, Lin CJ (2002) A comparison of methods for multiclass support vector machines. IEEE Trans Neural Netw 13:415–425
Hsu CW, Lin CJ (2002) A simple decomposition method for support vector machines. Mach Learn 46:291–314
Teo CH, Vishwanthan SVN, Smola AJ, Le QV (2010) Bundle methods for regularized risk minimization. J Mach Learn Res 11:311–365
Joachims T (1999) Advances in kernel methods. Chapter making large-scale support vector machine learning practical. MIT Press, Cambridge, pp 169–184
Joachims T (2006) Training linear svms in linear time. In: Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining, KDD '06. ACM, New York, pp 217–226
Joachims T, Finley T, Yu CNJ (2009) Cutting-plane training of structural svms. Mach Learn 77:27–59
Joachims T, Yu CNJ (2009) Sparse kernel svms via cutting-plane training. Mach Learn 76:179–193
Keerthi SS, DeCoste D (2005) A modified finite Newton method for fast solution of large scale linear svms. J Mach Learn Res 6:341–361
Keerthi SS, Gilbert EG (2002) Convergence of a generalized smo algorithm for svm classifier design. Mach Learn 46:351–360
Keerthi SS, Lin CJ (2003) Asymptotic behaviors of support vector machines with gaussian kernel. Neural Comput 15:1667–1689
Kimeldorf GS, Wahba G (1970) A correspondence between bayesian estimation on stochastic processes and smoothing by splines. Ann Math Stat 41:495–502
Kiwiel KC (2008) Breakpoint searching algorithms for the continuous quadratic knapsack problem. Math Program 112:473–491
Le QV, Smola AJ, Vishwanathan S (2008) Bundle methods for machine learning. In: Platt JC, Koller D, Singer Y, Roweis ST (eds) Advances in neural information processing systems 20. Curran Associates Inc, New York, pp 1377–1384

Lee MC, Chiang WL, Lin CJ (2015) Fast matrix-vector multiplications for large-scale logistic regression on shared-memory systems. In: Aggarwal C, Zhou Z-H, Tuzhilin A, Xiong H, Wu X (eds) ICDM. IEEE Computer Society, pp 835–840

Lee YJ, Mangasarian OL (2001) SSVM: a smooth support vector machine for classification. Comput Optim Appl 20:5–22

Lin C-J, Lucidi S, Palagi L, Risi A, Sciandrone M (2009) A decomposition algorithm model for singly linearly constrained problems subject to lower and upper bounds. J Optim Theory Appl 141:107–126

Lin CJ (2001) Formulations of support vector machines: a note from an optimization point of view. Neural Comput 13:307–317

Lin CJ (2001) On the convergence of the decomposition method for support vector machines. IEEE Trans Neural Netw 12:1288–1298

Lin CJ (2002) Asymptotic convergence of an SMO algorithm without any assumptions. IEEE Trans Neural Netw 13:248–250

Lin CJ (2002) A formal analysis of stopping criteria of decomposition methods for support vector machines. IEEE Trans Neural Netw 13:1045–1052

Lin CJ, Morè JJ (1999) Newton's method for large bound-constrained optimization problems. SIAM J Optim 9:1100–1127

Lucidi S, Palagi L, Risi A, Sciandrone M (2007) A convergent decomposition algorithm for support vector machines. Comput Optim Appl 38:217–234

Lucidi S, Palagi L, Risi A, Sciandrone M (2009) A convergent hybrid decomposition algorithm model for svm training. IEEE Trans Neural Netw 20:1055–1060

Mangasarian OL (1994) Nonlinear programming. Classics in applied mathematics. Society for Industrial and Applied Mathematics. ISBN: 9780898713411

Mangasarian OL (2002) A finite Newton method for classification. Optim Methods Softw 17:913–929

Mangasarian OL (2006) Exact 1-norm support vector machines via unconstrained convex differentiable minimization. J Mach Learn Res 7:1517–1530

Mangasarian OL, Musicant DR (1999) Successive overrelaxation for support vector machines. IEEE Trans Neural Netw 10:1032–1037

Mangasarian OL, Musicant DR (2001) Lagrangian support vector machines. J Mach Learn Res 1:161–177

Mavroforakis ME, Theodoridis S (2006) A geometric approach to support vector machine (SVM) classification. IEEE Trans Neural Netw 17:671–682

Osuna E, Freund R, Girosi F (1997) An improved training algorithm for support vector machines. In: Neural networks for signal processing VII. Proceedings of the 1997 IEEE signal processing society workshop, pp 276–285

Osuna E, Freund R, Girosit F (1997) Training support vector machines: an application to face detection. In Proceedings of IEEE computer society conference on computer vision and pattern recognition, pp 130–136

Palagi L, Sciandrone M (2005) On the convergence of a modified version of the svmlight algorithm. Optim Methods Softw 20:315–332

Pardalos PM, Kovoor N (1990) An algorithm for a singly constrained class of quadratic programs subject to upper and lower bounds. Math Program 46:321–328

Platt JC (1999) Advances in kernel methods. Chapter fast training of support vector machines using sequential minimal optimization. MIT Press, Cambridge, pp 185–208

Scholkopf B, Smola AJ (2001) Learning with Kernels: support vector machines, regularization, optimization, and beyond. MIT Press, Cambridge

Shalev-Shwartz S, Singer Y, Srebro N, Cotter A (2011) Pegasos: primal estimated sub-gradient solver for svm. Math Program 127:3–30

Shawe-Taylor J, Cristianini N (2004) Kernel methods for pattern analysis. Cambridge University Press, New York

Suykens JAK, Vandewalle J (1999) Least squares support vector machine classifiers. Neural Process Lett 9:293–300

Glasmachers T, Dogan U (2013) Accelerated coordinate descent with adaptive coordinate frequencies. In: Asian conference on machine learning, ACML 2013, Canberra, ACT, Australia, pp 72–86

Serafini T, Zanni L (2005) On the working set selection in gradient projection-based decomposition techniques for support vector machines. Optim Methods Softw 20:583–596

Tsang IW, Kwok JT, Cheung PM (2005) Core vector machines: fast svm training on very large data sets. J Mach Learn Res 6:363–392

Vapnik VN (1998) Statistical learning theory. Wiley-Interscience, New York
Wang PW, Lin CJ (2014) Iteration complexity of feasible descent methods for convex optimization. J Mach Learn Res 15:1523–1548
Wang Z, Crammer K, Vucetic S (2012) Breaking the curse of kernelization: budgeted stochastic gradient descent for large-scale svm training. J Mach Learn Res 13:3103–3131
Woodsend K, Gondzio J (2009) Hybrid mpi/openmp parallel linear support vector machine training. J Mach Learn Res 10:1937–1953
Woodsend K, Gondzio J (2011) Exploiting separability in large-scale linear support vector machine training. Comput Optim Appl 49:241–269