Neural Process Lett (2011) 33:31–44
DOI 10.1007/s11063-010-9162-9

First and Second Order SMO Algorithms for LS-SVM Classifiers

Jorge López · Johan A. K. Suykens

Published online: 5 December 2010

Abstract Least squares support vector machine (LS-SVM) classifiers have been traditionally trained with conjugate gradient algorithms. In this work, completing the study by Keerthi et al., we explore the applicability of the SMO algorithm for solving the LS-SVM problem, by comparing First Order and Second Order working set selections concentrating on the RBF kernel, which is the most usual choice in practice. It turns out that, considering all the range of possible values of the hyperparameters, Second Order working set selection is altogether more convenient than First Order. In any case, whichever the selection scheme is, the number of kernel operations performed by SMO appears to scale quadratically with the number of patterns. Moreover, asymptotic convergence to the optimum is proved and the rate of convergence is shown to be linear for both selections.

Keywords Least squares support vector machines · Sequential minimal optimization · Support vector classification · Working set selection

1 Introduction

Least squares support vector machines (LS-SVMs) were introduced in [1] as a simplification to support vector machines (SVMs) [2], where the inequality constraints are forced to become equality constraints and a least squares loss function is taken. In a binary classification context, we have a sample of N preclassified patterns $\{X_i, y_i\}, i = 1, \ldots, N$, where the outputs

J. López
Departamento de Ingeniería Informática and Instituto de Ingeniería del Conocimiento,
Universidad Autónoma de Madrid, C/Francisco Tomás y Valiente 11, 28049 Madrid, Spain

J. A. K. Suykens (✉)
Department of Electrical Engineering, ESAT-SCD/SISTA, Katholieke Universiteit Leuven,
Kasteelpark Arenberg 10, Heverlee, 3001 Leuven, Belgium
e-mail: Johan.Suykens@esat.kuleuven.be

$y_i \in \{+1, -1\}$, and the problem to be solved to obtain the LS-SVM binary classifier is:

$$\begin{aligned} \min_{W,b,\xi} \quad & \frac{1}{2}\|W\|^2 + \frac{C}{2}\sum_{i=1}^{N}\xi_i^2 \\ s.t. \quad & y_i\left(W \cdot \Phi\left(X_i\right) + b\right) = 1 - \xi_i,\ \forall i = 1, \ldots, N, \end{aligned} \tag{1}$$

where $\cdot$ denotes the vector inner product (dot product), and $\Phi\left(X_i\right)$ is the image of pattern X_i in the feature space with feature map $\Phi(\cdot)$. Taking the Lagrangian, differentiating with respect to the primal and dual variables and setting the derivatives to zero, one obtains:

$$\mathscr{L}\left(W, b, \xi, \alpha\right) = \frac{1}{2}\|W\|^2 + \frac{C}{2}\sum_{i=1}^{N}\xi_i^2 - \sum_{i=1}^{N}\alpha_i\left[y_i\left(W \cdot \Phi\left(X_i\right) + b\right) - 1 + \xi_i\right], \tag{2}$$

$$\frac{\partial L}{\partial W} = W - \sum_{i=1}^{N}\alpha_i y_i \Phi\left(X_i\right) = 0 \Rightarrow W = \sum_{i=1}^{N}\alpha_i y_i \Phi\left(X_i\right), \tag{3}$$

$$\frac{\partial L}{\partial b} = -\sum_{i=1}^{N}\alpha_i y_i = 0 \Rightarrow \sum_{i=1}^{N}\alpha_i y_i = 0, \tag{4}$$

$$\frac{\partial L}{\partial \xi} = C\xi_i - \alpha_i = 0 \Rightarrow \alpha_i = C\xi_i. \tag{5}$$

We note in passing that (3), (4) are the same than for SVMs, whereas (5) is not. In LS-SVMs the values of the coefficients α_i are proportional to the errors ξ_i, so they can be negative and are not box constrained, whereas in SVMs we have box constraints $0 \leq \alpha_i$ if we use the 2-norm of the errors and $0 \leq \alpha_i \leq C$ if we use the 1-norm. This fact makes the LS-SVM problem (1) easier to solve, but at the same time the sparsity of the SVM solutions is lost and an upper level procedure is needed to sparsify it, if needed.

In order to solve (1), there are two alternative approaches. The first is to substitute (5) and (3) in the constraints of (1) to get:

$$y_i\left(\sum_{j=1}^{N}\alpha_j y_j K_{ji} + b\right) = 1 - \frac{\alpha_i}{C}\ \forall i, \tag{6}$$

where $K_{ji} = k\left(X_j, X_i\right) = \Phi\left(X_j\right) \cdot \Phi\left(X_i\right)$, with k a Mercer kernel function. This, together with (4), form the KKT system:

$$\begin{bmatrix} 0 & y^T \\ y & \tilde{Q} \end{bmatrix}\begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \tag{7}$$

with $\tilde{Q}$ the Gram matrix with elements $Q_{ij} = y_i y_j \tilde{K}_{ij}$, and $\tilde{K}$ the modified kernel matrix with elements $\tilde{K}_{ij} = \tilde{k}\left(X_i, X_j\right) = k\left(X_i, X_j\right) + \frac{\delta_{ij}}{C}$. This modified kernel function $\tilde{k}$ is the same that is used in SVMs with the 2-norm of the errors [3]. Consequently, the first approach consists of solving this linear system of equalities. The current way to deal with this KKT system is by means of a Hestenes–Stiefel conjugate gradient method after a transformation to an equivalent system [4]. Observe that this approach is not valid for SVMs, which always have the inequality box constraints.

The other approach is to solve the dual problem, as it is usually done in SVMs. Substituting (5) and (3) in the Lagrangian (2), and taking into account (4), the dual problem is the following:

$$\min_{\alpha} \mathscr{D}(\alpha) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j \tilde{K}_{ij} - \sum_{i=1}^{N}\alpha_i$$
$$s.t. \qquad \sum_{i=1}^{N}\alpha_i y_i = 0. \tag{8}$$

This is the approach followed in this work and in the one by Keerthi and Shevade [5]. Note that in Keerthi's work, the dual is slightly different because the errors are transformed to $\xi_i' \leftarrow y_i\xi_i$ (this does not change the primal objective function), so accordingly $\alpha_i' \leftarrow y_i\alpha_i$ and the dual problem becomes:

$$\min_{\alpha} \mathscr{D}(\alpha') = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i'\alpha_j'\tilde{K}_{ij} - \sum_{i=1}^{N}\alpha_i' y_i$$
$$s.t. \qquad \sum_{i=1}^{N}\alpha_i' = 0. \tag{9}$$

For consistency with the primal problem (1), we will work with the dual problem (8) and not with (9).

In the special case of the linear kernel (that is, $\Phi(X_i) = X_i$), there are alternative approaches, since the primal (1) is directly addressable. For instance, we can also rewrite (1) as a KKT system of equations. Notice that in the general case this is not possible, since the feature map $\Phi(\cdot)$ is usually unknown.

This case is also special in standard SVMs, where the linear kernel has recently received much attention in domains such as text mining. In such domains, the dimensionality of the patterns is so high that using a non-linear kernel does not improve performance. Examples of successful algorithms for Linear SVMs are Pegasos [6], SVM-Perf [7] and LIBLINEAR [8].

Concerning Linear LS-SVMs, these algorithms do not cover them, and the topic has not received so much attention. A very recent example of an algorithm that covers both paradigms is NESVM [9]. However, in this work we will concentrate on non-linear kernels (most notably the RBF kernel) and general datasets with not so high dimensionality. The observations and results obtained are also valid for the particular case of linear kernels, since we always deal with the general feature map $\Phi(\cdot)$.

The rest of the paper is organized as follows: in Sect. 2 we introduce the rationale of the SMO algorithm, its First Order version proposed in [5] and the Second Order version inspired by the work by Fan et al. [10] and present in the state-of-the-art software LIBSVM [11]. In Sect. 3 we discuss the convergence of both SMO versions: we shall study the asymptotic convergence to the optimum as well as the rate of convergence to this optimum. In Sect. 4 we experiment with both versions of the algorithm to see their performance and their scaling for large-scale problems. Finally, Sect. 5 discusses the results and gives pointers to future possible extensions.

2 The SMO Algorithm

The SMO algorithm was developed in [12] as a decomposition method to solve the dual problems arising in SVM formulations. In each iteration, it selects two coefficients (α_i, α_j), while the rest of the coefficients keep their current values. Observe that this is the minimum number of coefficients that can be chosen, since the constraint $\sum_{i=1}^{N}\alpha_i y_i = 0$ is also present in SVM duals.

This selection of just two coefficients makes also possible to solve analytically the optimization subproblem, which usually is an advantage compared to other more general methods like SVMLight [13] that allow for working sets of size greater than two, but in turn require an inner solver to deal with the optimization subproblems.

Since the SMO subproblems are analytically solved, the key question is how to select the two coefficients to be updated. Initially, in [12] two heuristics were proposed that resulted in a somewhat cumbersome selection. Later, in [14] the concept of violating pair was introduced to denote two coefficients that cause a violation in the KKT optimality conditions of the dual, and the authors suggested to select always the pair that violated them the most, that is, the maximum violating pair (MVP). Finally, in [10] a Second Order selection is proposed that usually results in faster training that the MVP rule, even if that selection is considerably more costly. This is the selection adopted in the software LIBSVM [11], arguably the best-known and most commonly used SVM package nowadays.

These MVP and Second Order rules can be shown to be very intuitively derived under a point of view based on the dual gain [15]. We will follow the lines of this last work, adapting them to the dual problem (8).

First, we note that the KKT dual optimality conditions are (3), (4), (5) and the primal constraints in (1). After obtaining (6) by elimination of (3) and (5), we reformulate it to:

$$\begin{aligned} & y_i \left(\sum_{j=1}^{N} \alpha_j y_j \tilde{K}_{ji} + b \right) = 1 \quad \forall i \\ & \Rightarrow y_i \tilde{W} \cdot \tilde{\Phi}(X_i) = 1 - y_i b \quad \forall i \\ & \Rightarrow \tilde{W} \cdot \tilde{\Phi}(X_i) - y_i = -b \quad \forall i \end{aligned} \tag{10}$$

where $\tilde{\Phi}$ is the mapping to the feature space associated to the kernel $\tilde{k}$ and $\tilde{W} = \sum_i \alpha_i y_i \tilde{\Phi}(X_i)$. Since a dual feasible point already satisfies (4), the KKT conditions for an optimal dual solution α^* can be summarised in the equality:

$$\max \left\{ \tilde{W}^* \cdot \tilde{\Phi}(X_i) - y_i \right\} = \min \left\{ \tilde{W}^* \cdot \tilde{\Phi}(X_i) - y_i \right\}, \tag{11}$$

with $\tilde{W}^* = \sum_i \alpha_i^* y_i \tilde{\Phi}(X_i)$. In order to have a reasonable stopping criterion, one can relax this equality so that the difference between the maximum and minimum is less than a given tolerance:

$$\max \left\{ \tilde{W}^* \cdot \tilde{\Phi}(X_i) - y_i \right\} - \min \left\{ \tilde{W}^* \cdot \tilde{\Phi}(X_i) - y_i \right\} \leq \varepsilon, \tag{12}$$

for some $\varepsilon > 0$ (if $\varepsilon = 0$ we recover the exact KKT condition). These are also the rationale for the stopping criteria of LIBSVM and SVMLight, adapting the reasoning for the SVM KKT optimality conditions.

We will use this relaxed criterion instead of the duality gap proposed in [5], which is computationally more costly to compute, since it requires the estimation of the bias term in every iteration, as well as the calculation of the duality gap itself.

2.1 First Order SMO

We turn next to justify the MVP rule. In view of (8), we have that $\mathscr{D}(\alpha) = \frac{1}{2}\|\tilde{W}\|^2 - \sum \alpha_i$. Assume that we want to modify just a pair of coefficients (α_i, α_j), that is, an update rule of the form:

$$\begin{aligned} \alpha_i &\leftarrow \alpha_i + \delta_i, \\ \alpha_j &\leftarrow \alpha_j + \delta_j, \\ \alpha_k &\leftarrow \alpha_k \quad \forall k \neq i, j. \end{aligned} \tag{13}$$

Because of the constraint $\sum \alpha_i y_i = 0$, it must be the case that $\delta_i y_i + \delta_j y_j = 0$, so we can write $\delta_i = -y_i y_j \delta_j$ and the update in $\tilde{W}$ as $\tilde{W} \leftarrow \tilde{W} + y_j \delta_j (\tilde{\Phi}(X_j) - \tilde{\Phi}(X_i)) = \tilde{W} + y_j \delta_j \tilde{Z}_{j,i}$, with $\tilde{Z}_{j,i} = \tilde{\Phi}(X_j) - \tilde{\Phi}(X_i)$.

Hence, the new dual value can be expressed as a function of the increment δ_j:

$$\begin{aligned} \mathscr{D}(\alpha) &\leftarrow \frac{1}{2}\|\tilde{W}\|^2 - \sum \alpha_i + y_j \delta_j \tilde{W} \cdot \tilde{Z}_{j,i} + \frac{1}{2}\delta_j^2 \|\tilde{Z}_{j,i}\|^2 + y_i y_j \delta_j - \delta_j \\ &= \mathscr{D}(\alpha) + y_j \delta_j \left(\tilde{W} \cdot \tilde{Z}_{j,i} - \left(y_j - y_i\right)\right) + \frac{1}{2}\delta_j^2 \|\tilde{Z}_{j,i}\|^2 \\ &= \mathscr{D}(\alpha) + y_j \delta_j \Delta + \frac{1}{2}\delta_j^2 \|\tilde{Z}_{j,i}\|^2, \end{aligned}$$

where we denote $\Delta = \tilde{W} \cdot \tilde{Z}_{j,i} - \left(y_j - y_i\right)$. In order to minimize the new value of the dual, we differentiate with respect to δ_j and equal to zero, obtaining:

$$\delta_j^* = -y_j \frac{\Delta}{\|\tilde{Z}_{j,i}\|^2}, \quad \delta_i^* = y_i \frac{\Delta}{\|\tilde{Z}_{j,i}\|^2}, \quad \mathscr{D}(\alpha) \leftarrow \mathscr{D}(\alpha) - \frac{\Delta^2}{2\|\tilde{Z}_{j,i}\|^2}. \tag{14}$$

Note that, since the coefficients are unconstrained, we always obtain these values and there is no need to clip them, which is usually the case in SVMs due to the box constraints. Therefore, for LS-SVMs it is clear that the optimal pair choice is:

$$(i, j) = \arg\max_{l,m} \left\{ \frac{\left(\tilde{W} \cdot \tilde{Z}_{m,l} - (y_m - y_l)\right)^2}{2\|\tilde{Z}_{m,l}\|^2} \right\}. \tag{15}$$

First Order selection in [5] corresponds to ignoring the denominator in the above expression, and just choosing:

$$\begin{aligned} (i, j) &= \arg\max_{l,m} \left\{ \left(\tilde{W} \cdot \tilde{Z}_{m,l} - (y_m - y_l)\right)^2 \right\} \\ &= \arg\max_{l,m} \left\{ \left|\tilde{W} \cdot \tilde{Z}_{m,l} - (y_m - y_l)\right| \right\}, \end{aligned}$$

which is clearly maximized by:

$$j = \arg\max_m \left\{ \tilde{W} \cdot \tilde{\Phi}(X_m) - y_m \right\}, \quad i = \arg\min_l \left\{ \tilde{W} \cdot \tilde{\Phi}(X_l) - y_l \right\}. \tag{16}$$

As was the case for (14), in standard SVMs the selections (15) and (16) are somewhat more complex, since it must be taken into account that the coefficients are box-constrained [15]. Selection (16), by (11) or its relaxed version (12), corresponds to choosing the MVP. If there is no such violating pair, then the algorithm finishes because we are in an (ε-)optimal point.

2.2 Second Order SMO

The so-called Second Order selection was proposed in [10] as a better approximation than (16) to the optimal choice (15). Note that, in First Order, the cost of the search of (16) is $\mathscr{O}(1)$

kernel operations, if we assume that the values $F_i = \tilde{W} \cdot \tilde{\Phi}(X_i) - y_i$ are cached throughout the algorithm. In practice, this is always the case, since the updates of these values are of the form:

$$F_l \leftarrow F_l - \frac{\Delta}{\left\| \tilde{Z}_{j,i} \right\|^2} \left(\tilde{k}(X_j, X_l) - \tilde{k}(X_i, X_l) \right), \tag{17}$$

up to an algorithmic cost of $2N$ kernel operations. However, if we want to find the optimal pair in (15) we have to check all possible pairs, which is $\mathcal{O}\left(N^2\right)$. Since this cost is impractical for most of the problems at hand, Fan's proposal is a compromise between First Order and full search, in the following way:

$$i = \arg\min_l \left\{ \tilde{W} \cdot \tilde{\Phi}(X_l) - y_l \right\}, \quad j = \arg\max_{m \neq i} \left\{ \frac{\left(\tilde{W} \cdot \tilde{Z}_{m,i} - (y_m - y_i) \right)^2}{2 \left\| \tilde{Z}_{m,i} \right\|^2} \right\}. \tag{18}$$

That is, i is selected as in First Order, and j is selected to maximize the expression in (15), keeping fixed i. This implies a cost of $\mathcal{O}(N)$ kernel operations, so in principle it is more costly than First Order. Nevertheless, in [10] it is shown that usually the gain in iterations makes up for this additional cost, altogether resulting in a faster algorithm for SVM training.

Moreover, note that this additional cost is not as much as one might at first think. Observe that $\left\| \tilde{Z}_{m,i} \right\|^2 = \tilde{K}_{ii} + \tilde{K}_{mm} - 2\tilde{K}_{mi}$. Assuming that $diag(K)$ is precomputed and stored in memory, a naïve implementation would compute the N kernel operations $\tilde{K}_{mi}$ for the pair selection and then would apply the updates in (17), up to at least $3N$ kernel operations per iteration. However, $\tilde{K}_{mi} = \tilde{K}_{im}$, so, if we store these N values when looking for the pair, we only need to compute $\tilde{K}_{jl}$ in (17). Thus, both selection schemes amount to $2N$ kernel operations per iteration, and the additional cost of Second Order is just the extra loop that searches j and the time for storing $\tilde{K}_{mi}$ in memory and recovering it for (17).

This said, there are some cases where obviously Second Order is not worth the effort. Specifically, if the denominator in (18) is basically constant, the maximization just maximizes the numerator, so it reduces to (16). Considering from now on the RBF kernel $k\left(X_i, X_j\right) = \exp(-\|X_i - X_j\|^2/\sigma^2)$ (which is by far the most commonly used in SVM literature), we have the bounds $0 \leq \tilde{K}_{mi} \leq 1$, and also $\tilde{K}_{ii} = 1 + \frac{1}{C}\ \forall i$. Hence, $\|\tilde{Z}_{m,i}\|^2$ can be written as $2\left(1 + \frac{1}{C}\right) - 2\tilde{K}_{mi}$, with $m \neq i$. There are three cases that make this denominator constant:

1. $C \approx 0$: then $\frac{1}{C} \gg 1$, so $\|\tilde{Z}_{m,i}\|^2 \approx \frac{2}{C}$.
2. $\sigma \approx 0$: then $K \to (1 + \frac{1}{C})I$, so $\tilde{K}_{mi} \approx 0$ and $\|\tilde{Z}_{m,i}\|^2 \approx 2\left(1 + \frac{1}{C}\right)$.
3. $\sigma \gg 0$: then all the off-diagonal elements of K tend to be equal to 1, so $\tilde{K}_{mi} \approx 1$ and $\|\tilde{Z}_{m,i}\|^2 \approx 2\left(1 + \frac{1}{C}\right) - 2 = \frac{2}{C}$.

Therefore, in these three cases Second Order SMO behaves almost equivalently to First Order SMO, so a very similar number of iterations is expected, but with a slightly higher cost. The experiments in Sect. 4 illustrate that this is indeed the case for the cases $C \approx 0$ and $\sigma \gg 0$ (the case $\sigma \approx 0$ is omitted since such a small value for σ is hardly ever convenient in terms of accuracy of the final models).

3 Convergence

In the previous section we have discussed the adaptation of the First and Second Order versions of SMO for LS-SVM training. A question that arises naturally is whether both versions

are guaranteed to converge to the optimal solution, and in case they are, what is the rate of convergence to the optimum.

3.1 Asymptotic Convergence

The asymptotic convergence of the First Order version was established in [5]. Next we will rewrite the proof in this reference with our notation. This will clarify that the proof does not rely on the particular pair selection (as long as the selected pair is a violating one), so the convergence of the Second Order version can also be stated. Besides, we introduce two intermediate lemmas for the sake of clarity and completeness: Lemma 1 is part of the proof in [5], whereas Lemma 2 is only used in [5], but not proved.

Lemma 1 ([5]) *The decrease of the dual function in an iteration t of SMO satisfies* $\mathscr{D}(\alpha^{t+1}) - \mathscr{D}(\alpha^t) \leq -\frac{\left\|\alpha^{t+1}-\alpha^t\right\|^2}{2C}$.

Proof Once the pair (i, j) is selected, the only coefficients changed in (14) are $\alpha_i^{t+1} = \alpha_i^t - y_i y_j \delta_j^*$ and $\alpha_j^{t+1} = \alpha_j^t + \delta_j^*$, so we can write $(\delta_j^*)^2 = \frac{\|\alpha^{t+1}-\alpha^t\|^2}{2}$. The change of the dual in (14), that is, $\mathscr{D}(\alpha^{t+1}) - \mathscr{D}(\alpha^t) = -\frac{\Delta^2}{2\|\tilde{Z}_{ji}\|^2}$ can be rewritten as $\mathscr{D}(\alpha^{t+1}) - \mathscr{D}(\alpha^t) = -\frac{(\delta_j^*)^2}{2}\|\tilde{Z}_{ji}\|^2$. Plugging in $\|\tilde{Z}_{ji}\|^2 = \frac{2}{C} + \|\Phi(X_i) - \Phi(X_j)\|^2 \geq \frac{2}{C}$, we get the result, since $\mathscr{D}(\alpha^{t+1}) - \mathscr{D}(\alpha^t) \leq -\frac{(\delta_j^*)^2}{C} = -\frac{\|\alpha^{t+1}-\alpha^t\|^2}{2C}$.

Lemma 2 *The dual function $\mathscr{D}(\alpha)$ is bounded below.*

Proof The dual can be written as $\mathscr{D}(\alpha) = \frac{1}{2}\|\tilde{W}\|^2 - \sum_i \alpha_i \geq -\sum_i \alpha_i$. Hence, we would like to bound above the summation $\sum_i \alpha_i$. By Cauchy's inequality, $\sum_i \alpha_i \leq \sqrt{N}\sqrt{\sum_i \alpha_i^2}$. From $\|\tilde{W}\|^2 = \alpha^T \tilde{Q}\alpha$ and the general result that $\lambda_{\min}(\tilde{Q})\alpha^T\alpha \leq \alpha^T\tilde{Q}\alpha$, where $\lambda_{\min}(\tilde{Q}) > 0$ stands for the minimal eigenvalue of matrix $\tilde{Q}$, it follows that $\sqrt{\sum_i \alpha_i^2} = \sqrt{\alpha^T\alpha} \leq \frac{\|W\|}{\sqrt{\lambda_{\min}(\tilde{Q})}}$. Thus, $\sum_i \alpha_i \leq \sqrt{N}\frac{\|W\|}{\sqrt{\lambda_{\min}(\tilde{Q})}}$.

Since $\alpha = 0$ is a feasible solution, we get $\frac{1}{2}\|\tilde{W}\|^2 \leq \sum_i \alpha_i \leq \sqrt{N}\frac{\|W\|}{\sqrt{\lambda_{\min}(\tilde{Q})}}$, so $\|W\| \leq \frac{2\sqrt{N}}{\sqrt{\lambda_{\min}(\tilde{Q})}}$ and therefore $\sum_i \alpha_i \leq \frac{2N}{\lambda_{\min}(\tilde{Q})}$, yielding the result with $\mathscr{D}(\alpha) \geq \frac{-2N}{\lambda_{\min}(\tilde{Q})}$.

Theorem 1 ([5, Lemma 1]) *The sequence of iterates $\{\alpha^t\}$ given by First Order SMO converges to the optimal point α^*.*

Proof Since the modified kernel matrix $\tilde{K}$ adds the term $\frac{1}{C}$ and any Mercer kernel guarantees that K is positive semidefinite, $\tilde{K}$ is positive definite, so the dual function $\mathscr{D}(\alpha)$ of (8) is strictly convex and the optimal point α^* is unique.

From Lemmas 1 and 2, the dual $\mathscr{D}(\alpha)$ decreases in every iteration and is bounded below, so the sequence $\{\mathscr{D}(\alpha^t)\}$ is convergent. This implies by Lemma 1 that the sequence $\{\alpha^{t+1} - \alpha^t\}$ converges to 0. Given an iteration t, since the SMO subproblem with a pair (i, j) is optimally solved by taking δ_j^*, the KKT condition (11) holds for the pair, so $F_j^{t+1} = F_i^{t+1}$.

If the number of iterations is finite, then (11) holds, so α^* is attained. Thus, from now on we consider an infinite number of iterations. Since the number of possible pairs is finite, there is at least one pair (i, j) that is selected an infinite number of times. Because of Lemma 2,

the set $\{\alpha | \mathscr{D}(\alpha) \leq \mathscr{D}(\alpha^0)\}$ is compact. The sequence $\{\alpha^t\}$ lies in this set, so it is a bounded sequence and has at least one limit point. Let $\{\alpha^{t_k}\}$ a convergent subsequence of $\{\alpha^t\}$ with limit point $\overline{\alpha}$. To be precise, let us consider the subsequence $\{\alpha^{t_{k_l}}\}$ of this subsequence where the selected pair for optimization is always (i, j).

The functions $\max_m\{\tilde{W} \cdot \tilde{\Phi}(X_m) - y_m\}$ and $\min_n\{\tilde{W} \cdot \tilde{\Phi}(X_n) - y_n\}$ are clearly continuous on α. Therefore, we can write $\lim_{t_{k_l}\to\infty}\{W^{t_{k_l}} \cdot \tilde{\Phi}(X_j) - y_j - (W^{t_{k_l}} \cdot \tilde{\Phi}(X_i) - y_i)\} = \lim_{t_{k_l}\to\infty}\{F_j^{t_{k_l}} - F_i^{t_{k_l}}\} = \lim_{t_{k_l}\to\infty}\{(F_j^{t_{k_l}} - F_j^{t_{k_l}+1}) + (F_j^{t_{k_l}+1} - F_i^{t_{k_l}+1}) + (F_i^{t_{k_l}+1} - F_i^{t_{k_l}})\}$.

Since $\{\alpha^{t_{k_l}+1} - \alpha^{t_{k_l}}\}$ converges to zero, the first and third term in parentheses tend to zero. As for the second one, it is always zero, since we know that $F_j^{t_{k_l}+1} = F_i^{t_{k_l}+1}$. Thus, the above limits are all zero, the KKT condition (11) holds and so the limit point is $\overline{\alpha} = \alpha^*$. The strict convexity of $\mathscr{D}(\alpha)$ ensures that the sequences $\{\alpha^{t_k}\}$ and $\{\alpha^t\}$ themselves also converge to α^*, so the result follows.

From this result and the selection in the Second Order variant, it is immediate that this variant is also asymptotically convergent.

Theorem 2 *The sequence of iterates $\{\alpha^t\}$ given by Second Order SMO converges to the optimal point α^*.*

Proof The proof is analogous to the one above, because the pair selection is irrelevant for the proof, as long as that selection guarantees the strict decrease of the dual function $\mathscr{D}(\alpha)$. This is also the case for Second Order, since (18) ensures that $\Delta \neq 0$, so (14) gives the strict decrease.

3.2 Rate of Convergence

To our knowledge, the rate of convergence of SMO applied to LS-SVM training has never been tackled. Fortunately, the results of Chih-Jen Lin and his team on the convergence of SMO for SVM training [10,16,17] are applicable here. Moreover, the assumptions they need for their calculations on the SVM case are automatically fulfilled in the LS-SVM case.

Specifically, two assumptions are needed: (1) the matrix of the quadratic form Q to be positive definite (recall that $\tilde{Q}$ is guaranteed to be so), (2) the optimal solution to be non-degenerate. This last assumption is intended so that the SVM dual can be written without the box constraints in the coefficients α_i once the iteration number is large enough. Since in LS-SVMs these coefficients are not constrained, the assumption also holds.

Therefore, their results also apply here. We will omit the lengthy proofs because there is no change in them aside from some notation. For details, we refer the interested reader to reference [17].

Theorem 3 ([17, Theorem 7]) *There exists an $\eta < 1$ such that $(\alpha^{t+1} - \alpha^*)^T \tilde{Q}(\alpha^{t+1} - \alpha^*) \leq \eta(\alpha^t - \alpha^*)^T \tilde{Q}(\alpha^t - \alpha^*)$.*

Specifically, we get the value

$$\eta = 1 - \min_P \left\{ \frac{\lambda_{\min}(\tilde{Q}_{PP}^{-1})}{2N\lambda_{\max}(\tilde{Q}^{-1})} \left(\frac{y^T \tilde{Q}^{-1} y}{\sum_m \sum_n \left\| \tilde{Q}_{mn}^{-1} \right\|} \right)^2 \rho^2 \right\}, \tag{19}$$

where P stands for every possible pair (i, j), $\tilde{Q}_{PP}^{-1}$ is the submatrix of $\tilde{Q}^{-1}$ corresponding to the pair (i, j), $\lambda_{\min}$ denotes the minimal eigenvalue of a matrix, and $\lambda_{\max}$ its maximal

eigenvalue. The value $0 \leq \rho \leq 1$ represents the relative violation of the selected pair with respect to the MVP. Therefore, for First Order $\rho = 1$, whereas for Second Order it can be shown [10] that

$$\rho = \sqrt{\frac{\min_{m,n}\{\left\|\tilde{Z}_{mn}\right\|^2\}}{\max_{m,n}\{\left\|\tilde{Z}_{mn}\right\|^2\}}}. \tag{20}$$

Observing that $\mathscr{D}(\alpha^t) - \mathscr{D}(\alpha^*) = \frac{1}{2}(\alpha^t - \alpha^*)^T \tilde{Q}(\alpha^t - \alpha^*)$, the linear rate of convergence is established in the following result:

Theorem 4 ([17, Theorem 8]) *It holds that* $\mathscr{D}(\alpha^{t+1}) - \mathscr{D}(\alpha^*) \leq \eta(\mathscr{D}(\alpha^t) - \mathscr{D}(\alpha^*))$, *where* η *is given by* (19).

Clearly, the minimum in (19) is $\leq \frac{1}{2N}$, so we have $\eta \geq 1 - \frac{1}{2N}$. By Theorem 4, the smaller η becomes, the faster the convergence is. Therefore, when the minimum in (19) approaches its maximal value $\frac{1}{2N}$ we should have a faster convergence.

Starting with First Order, this will be the case when $\tilde{Q}$ becomes the (possibly scaled by a positive scalar) identity matrix I, for then in (19) all the eigenvalues of $\tilde{Q}$ (and thus of $\tilde{Q}^{-1}$) will be the same, and also $y^T \tilde{Q}^{-1} y$ becomes $\sum_i y_i^2 \tilde{Q}_{ii}^{-1} = \sum_i \sum_j \tilde{Q}_{ij}^{-1} = \sum_i \sum_j \|\tilde{Q}_{ij}^{-1}\|$. This happens in two cases:

1. $C \approx 0$: then $\frac{1}{C} \gg 1$, so $Q \approx \frac{1}{C} I \Rightarrow Q^{-1} \approx CI$.
2. $\sigma \approx 0$: then $Q \approx (1 + \frac{1}{C})I \Rightarrow Q^{-1} \approx \frac{1}{1+1/C} I$.

Observe that these two cases are two cases where it was not convenient to use Second Order (see Sect. 2.2). Equation (19) gives another motivation for this fact, since for these cases $\min_{m,n}\{\left\|\tilde{Z}_{mn}\right\|^2\} \approx \max_{m,n}\{\left\|\tilde{Z}_{mn}\right\|^2\}$, and so $\rho \approx 1$ by (20).

The remaining case in Sect. 2.2 of $\sigma \gg 0$ is not so clear, because in that case Q is dense unless $C \approx 0$ (the off-diagonal elements tend to ± 1 and the diagonal ones are $1 + 1/C$), so the convergence rate is not known a priori. It will depend on how well or ill-conditioned Q is: if it is very ill-conditioned $1/\lambda_{\max}(Q^{-1}) = \lambda_{\min}(Q) \approx 0$, and thus $\eta \approx 1$. This will also be the situation when either the value of σ is intermediate or when C is not small. In Second Order, there is the additional influence of the term ρ, which depends on the uniformity of the kernel matrix $\tilde{K}$.

4 Experiments

In this section we will run three sets of experiments to see three basic aspects: (1) the influence of the hyperparameters C and σ on the convergence, (2) the generalization properties of LS-SVMs and (3) the scaling of the training times with respect to the training set size. To do this, we run both SMO selections (First Order and Second Order), always using the RBF kernel $k(X_i, X_j) = \exp(-\|X_i - X_j\|^2/2\sigma^2)$ and the stopping condition (12) with $\varepsilon = 10^{-3}$, i.e. $\Delta \leq 10^{-3}$, which is the default choice of SVM packages like [11] and [13]. Regarding the hardware, an Intel Core 2 Quad machine with 8 GB of RAM and 4 processors, each with 2.66 GHz, was used. However, no explicit parallelization was implemented in the source code, which was written in C++.

The first set is aimed to characterize the behaviour of both selections depending on the value of the hyperparameter C. For this purpose, we use the benchmark datasets Banana,

Table 1 Average execution times of each run (in seconds) for the SMO with First Order (SMO 1) and Second Order (SMO 2) pair selection in the datasets Banana, Image, Splice and German with different values of the hyperparameter C

Log C	Banana ($\sigma^2 = 1.8221$)		Image ($\sigma^2 = 2.7183$)		Splice ($\sigma^2 = 29.9641$)		German ($\sigma^2 = 31.2500$)	
	SMO 1	SMO 2	SMO 1	SMO 2	SMO 1	SMO 2	SMO 1	SMO 2
−4	11.500	13.251	2.271	2.588	3.621	4.202	0.291	0.349
−3	7.674	8.850	1.825	2.116	3.498	4.057	0.272	0.347
−2	6.801	7.841	1.252	1.444	2.573	2.976	0.190	0.219
−1	5.989	6.590	1.100	1.280	2.680	3.097	0.142	0.177
0	10.541	6.753	2.102	1.229	2.364	2.599	0.188	0.205
1	58.775	10.798	5.740	1.866	3.523	2.582	0.740	0.512
2	530.937	41.718	35.279	4.813	5.853	1.909	3.104	1.574
3	1391.656	119.492	350.215	19.637	5.119	1.709	8.223	2.896

The value of σ is kept fixed so as to preserve good generalization

Image, Splice and German, available in [18]. The values of the hyperparameter σ for Banana, Image and Splice are taken from [5], whereas the one for German is taken from [19], and in all cases result in good generalization values. Since the number of kernel operations is the same for both selections, we measure execution times for the different settings. Table 1 shows the average times obtained (every setting was executed 10 runs for the sake of soundness).

We can see very clearly the effect mentioned in Sect. 2.2: for small values of C it is preferable to use First Order; the computational burden of Second Order is higher than the one of First Order because both schemes are selecting the very same pairs. In any case, the difference is not very significant, since most of the time is devoted to calculating kernel operations, which are identical in number for both schemes.

However, for large values of C it is preferable to use Second Order. It scales much better than First Order in all datasets. It is for middle values of C when the trend changes (around 10^1). This better behaviour of Second Order with $C \gg 0$ is not a direct consequence of (19) and (20), but anyway seems to be a general rule (unless the value of σ is very small, where again Second Order reduces to First Order).

To further explore the influence of the hyperparameters and check the generalization properties of LS-SVMs, next we run a second set of experiments on the datasets Titanic, Heart, Breast Cancer, Thyroid and Pima (also available in [18]), using the hyperparameter values obtained in [20] by an evolutionary algorithm. This time, instead of using the whole dataset as in the first set of experiments, we use the 100 partitions of each dataset provided in [18], so as to certify the good generalization properties of those hyperparameter values (reported in Table 2). We also report the average number of iterations and execution times per run in Table 3. Again, we performed 10 executions for every partition, so that time measurements are sound.

The misclassification rates obtained are consistent with the ones reported in [20] and also show that, in general, the stopping criterion used is a good one, since the classification performance is nearly the same whichever the pair selection is (we have checked that imposing a smaller ε these performances become identical).

It can be seen that for these datasets and hyperparameters, it is better to use Second Order in Titanic, Thyroid and Pima, whereas it is better to use First Order in Heart and Breast

Table 2 Hyperparameter values for the datasets Titanic, Heart, Cancer, Thyroid and Pima

Dataset	$\text{Log}_{10} C$	$\text{Log}_{10} \sigma$
Titanic	4.00	0.58
Heart	1.41	1.83
Cancer	0.33	0.76
Thyroid	1.27	0.26
Pima	1.92	1.20

Table 3 Average misclassification rates, number of iterations (in thousands) and execution times of each run (in milliseconds) for the SMO with First Order (SMO 1) and Second Order (SMO 2) pair selection for the datasets Titanic, Heart, Cancer, Thyroid and Pima with the hyperparameter values of Table 2

Dataset	Misclassification rate		# Iters.		Execution times	
	SMO 1	SMO 2	SMO 1	SMO 2	SMO 1	SMO 2
Titanic	22.3 ± 1.1	22.5 ± 1.3	272.378 ± 59.776	2.037 ± 1.139	1706.6 ± 374.1	24.2 ± 13.3
Heart	15.6 ± 3.2	15.6 ± 3.2	0.263 ± 0.006	0.262 ± 0.008	2.3 ± 0.2	3.9 ± 0.2
Cancer	26.0 ± 4.5	26.0 ± 4.5	0.509 ± 0.011	0.396 ± 0.008	4.7 ± 0.2	6.6 ± 0.2
Thyroid	4.7 ± 2.2	4.7 ± 2.2	1.322 ± 0.075	0.525 ± 0.031	8.0 ± 0.5	6.1 ± 0.4
Pima	23.2 ± 1.6	23.2 ± 1.6	5.526 ± 0.190	1.883 ± 0.036	117.4 ± 4.1	74.3 ± 0.1

Cancer. Looking at Table 2, we see that Titanic and Pima have large values of C, whereas for Breast Cancer it is quite small, and Heart and Thyroid something in between.

The results in Table 3 show that the biggest improvement with Second Order happens for Titanic, next for Pima and next for Thyroid, which is consistent with the ordered values of their hyperparameter C. Therefore, this is further evidence on the previous observation that for large values of C Second Order outperforms First Order. Breast Cancer is better tackled by First Order because the value of C is too small for the Second Order search to be worthwhile. Nevertheless, it is not small enough for Second Order to reduce to First Order, since its number of iterations is still less than the one of First Order.

Heart has a bigger value of C than Thyroid, but for this dataset Second Order is worse than First Order because the value of σ is large. Note that in this case it is large enough for Second Order to reduce to First Order: virtually the same number of iterations are performed in both cases.

The final set of experiments aims to ascertaining how well the SMO algorithm scales for large-scale problems when it uses First Order and Second Order working set selections. Equation (19) suggests that the rate of convergence is inversely proportional to the number of patterns. In order to test this, we use the datasets Adult and Web, available in [11] for several increasing numbers of patterns.

In Figs. 1 and 2 we plot the results for Adult with $C = 1, \sigma^2 = 10$ and Web with $C = 5, \sigma^2 = 10$, respectively (those were the values used in [5]). As it can be seen, the number of iterations scales linearly with the training set size, as suggested by Eq. (19). Note that Second Order needs less iterations to converge, as expected, and that this reduction is greater for Web than for Adult, mainly because of its larger value of C. In any case, the scaling is linear in both cases.

Since, whichever the working set selection is, SMO has to perform $\mathcal{O}(N)$ kernel operations per iteration, the fact that the number of iterations scales linearly means that the

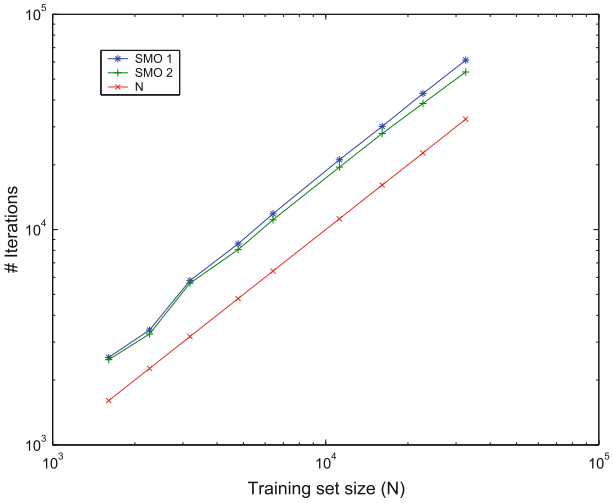

Fig. 1 Variation of number of iterations with training set size for dataset Adult

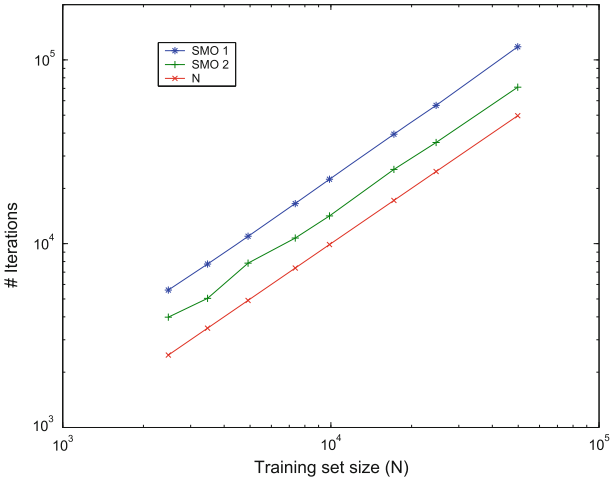

Fig. 2 Variation of number of iterations with training set size for dataset Web

number of kernel operations (and consequently the execution times) scale quadratically. This is consistent with the experiments in [5], where First Order selection was shown to scale quadratically in terms of kernel operations.

5 Discussion and Further Work

In this work we have dealt with First and Second Order working set selections for SMO regarding LS-SVM classifier training. The SMO algorithm arises naturally when one considers the dual formulation of LS-SVMs instead of the KKT linear system of equalities, the latter having been the most widely used approach so far. In its final form, the resulting SMO is simpler

than its counterpart for SVMs, since the coefficients are not box-constrained, and in terms of computational cost it empirically scales quadratically with the training set size.

Moreover, what has been said is also valid for regression. We can use a common formulation for classification and regression by substituting the constraints in (1) for $W \cdot \Phi(X_i) + b = y_i - \xi_i$. Concerning the SMO algorithm, the only changes are that $\delta_i = -\delta_j$ and $\delta_j^* = -\frac{\Delta}{\|\tilde{Z}_{j,i}\|^2}$, $\delta_i^* = \frac{\Delta}{\|\tilde{Z}_{j,i}\|^2}$ instead of (14). This corresponds to using the dual (9) in lieu of (8).

For SVMs it was shown in Fan's work that Second Order working set selection nearly always results in faster training than First Order selection despite its more expensive computational burden. We have seen that this conclusion is in general also true for LS-SVMs. Even if their primal formulation with square penalties compels us to use the modified kernel matrix $K + (1/C)I$, and this fact provokes that First Order is more convenient than Second Order in some settings, the computational cost of Second Order is such cases is only slightly higher than the one for First Order.

In particular, there are three cases where Second Order reduces to First Order: (i) $C \approx 0$, (ii) $\sigma \approx 0$, (iii) $\sigma \gg 0$. Experimentally, we check the above for cases (i) and (iii). However, for the case $C \gg 0$, Second Order is empirically shown to converge much faster than First Order. This is not a direct consequence of the theory, and is worth for further investigation. In any case, what can be said is that altogether Second Order is a more balanced choice, since it can result in very noticeable savings for the case $C \gg 0$, whereas for the rest of cases its performance is nearly the same as that of First Order.

We have also seen how both SMO selections are guaranteed to converge to the optimal solution. Besides, the rate of convergence is shown to be linear for both types of pair selection. A bound given for this rate suggests that for cases (i) and (ii) the convergence of SMO is faster. For the rest of cases, the speed of convergence depends on the condition number of the Gram matrix, and in Second Order the uniformity of the kernel matrix is also seen to have an influence.

In order to improve even more the performance of SMO, the use of cycle-based acceleration [21] will also be addressed. This idea has worked very well with First Order SMO for SVMs, so it is conjectured that it is potentially useful with Second Order SMO. In addition, the fact that the LS-SVM dual problem is less constrained will probably mean that the improvements are greater than for SVMs.

For the particular case of Linear LS-SVMs, another interesting topic is the comparison between SMO and algorithms such as NESVM [9]. It is conjectured that the results obtained will be similar to the ones reported in that work, since the LIBSVM software internally uses Second Order SMO.

Acknowledgments This research work was carried out at the ESAT laboratory of the Katholieke Universiteit Leuven. Jorge López is a doctoral researcher kindly supported by an FPU (Formación de Profesorado Universitario) grant from the Spanish Ministry of Education, reference AP2007-00142. Johan Suykens is a professor with K.U. Leuven: Research supported by Research Council K.U. Leuven: GOA AMBioRICS, GOA-MaNet, CoE EF/05/006, OT/03/12, PhD/postdoc & fellow grants; Flemish Government: FWO PhD/postdoc grants, FWO projects G.0499.04, G.0211.05, G.0226.06, G.0302.07; Research communities (ICCoS, ANMMM, MLDM); AWI: BIL/05/43, IWT: PhD Grants; Belgian Federal Science Policy Office: IUAP DYSCO.

References

1. Suykens JAK, Vandewalle J (1999) Least squares support vector machine classifiers. Neural Process Lett 9(3):293–300
2. Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297

3. Keerthi SS, Shevade SK, Bhattacharyya C, Murthy KRK (2000) A fast iterative nearest point algorithm for support vector machine classifier design. IEEE Trans Neural Netw 11(1):124–136
4. Suykens JAK, Lukas L, Van Dooren P, De Moor B, Vandewalle J (1999) Least squares support vector machine classifiers: a large scale algorithm. In: Proceedings of the European conference on circuit theory and design (ECCTD), pp 839–842
5. Keerthi SS, Shevade SK (2003) SMO algorithm for least-squares SVM formulations. Neural Comput 15(2):487–507
6. Shalev-Shwartz S, Singer Y, Srebro N (2007) Pegasos: primal estimated sub-gradient solver for SVM. In: Proceedings of the 24th international conference on machine learning (ICML), pp 807–814
7. Joachims T (2006) Training linear SVMs in linear time. In: Proceedings of the 12th international conference on knowledge discovery and data mining (SIGKDD), pp 217–226
8. Fan R-E, Chang K-W, Hsieh C-J, Wang X-R, Lin C-J (2008) LIBLINEAR: a library for large linear classification. J Mach Learn Res 9:1871–1874
9. Zhou T, Tao D, Wu X (2010) NESVM: a fast gradient method for support vector machines. In: Proceedings of the 12th international conference on data mining (ICDM)
10. Fan R-E, Chen P-H, Lin C-J (2005) Working set selection using second order information for training support vector machines. J Mach Learn Res 6:1889–1918
11. Chang C-C, Lin C-J (2001) LIBSVM: a library for support vector machines software. http://www.csie.ntu.edu.tw/~cjlin/libsvm
12. Platt J-C (1999) Fast training of support vector machines using sequential minimal optimization. In: Advances in kernel methods: support vector learning. MIT Press, Cambridge, pp 185–208
13. Joachims T (1999) Making large-scale support vector machine learning practical. In: Advances in kernel methods: support vector learning. MIT Press, Cambridge, pp 169–184
14. Keerthi SS, Shevade SK, Bhattacharyya C, Murthy KRK (2001) Improvements to Platt's SMO algorithm for SVM classifier design. Neural Comput 13(3):637–649
15. López J, Barbero Á, Dorronsoro JR (2008) On the equivalence of the SMO and MDM algorithms for SVM training. In: Lecture notes in computer science: machine learning and knowledge discovery in databases, vol 5211. Springer, New York, pp 288–300
16. Lin C-J (2001) Linear convergence of a decomposition method for support vector machines. Technical report
17. Chen P-H, Fan R-E, Lin C-J (2006) A study on SMO-type decomposition methods for support vector machines. IEEE Trans Neural Netw 17:893–908
18. Rätsch G (2000) Benchmark repository. http://ida.first.fhg.de/projects/bench/benchmarks.htm
19. Van Gestel T, Suykens JAK, Baesens B, Viaene S, Vanthienen J, Dedene G, De Moor B, Vandewalle J (2004) Benchmarking least squares support vector machine classifiers. Mach Learn 54(1):5–32
20. Guo XC, Yang JH, Wu CG, Wang CY, Liang YC (2008) A novel LS-SVMs hyper-parameter selection based on particle swarm optimization. Neurocomputing 71(16–18):3211–3215
21. Barbero Á, López J, Dorronsoro JR (2009) Cycle-breaking acceleration of SVM training. Neurocomputing 72(7–9):1398–1406