Convergence of the IRWLS procedure to the Support Vector Machine solution

Fernando Pérez-Cruz, Carlos Bousoño-Calzón, and
Antonio Artés-Rodríguez

25th May 2004

Abstract

An Iterative Re-Weighted Least Squares (IRWLS) procedure recently proposed is shown to converge to the Support Vector Machine solution. The convergence to a stationary point is assured by modifying the original IRWLS procedure.

1 Introduction

Support vector machines (SVMs) are state-of-the-art tools for linear and non-linear input-output knowledge discovery (Vapnik, 1998; Schölkopf and Smola, 2001). The SVM relies on the minimization of a quadratic problem, which is frequently solved using Quadratic Programming (QP) (Burges, 1998). The Iterative Re-Weighted Least Square (IRWLS) procedure for solving SVM for classification was first introduced in (Pérez-Cruz et al., 1999; Pérez-Cruz et al., 2001) and it was used in (Pérez-Cruz et al., 2000a) to construct the fastest SVM solver of the time. It solves a sequence of weighted least square problems that, unlike other least square procedures such as Lagrangian SVMs (Mangasarian and Musicant, 2000) or Least Square SVMs (Suykens and Vandewalle, 1999; Van Gestel et al., 2004), leads to the true SVM solution, as we will show herein. However, to prove its convergence to the SVM solution, the IRWLS procedure

has to be modified with respect to the formulation that appears in (Pérez-Cruz et al., 1999; Pérez-Cruz et al., 2001).

The IRWLS has been also proposed for solving regression problems (Pérez-Cruz et al., 2000b). Although, we will only deal with the IRWLS for classification, the extension of this proof to regression is straightforward.

The paper outline goes as follows. We proof the convergence of the IRWLS procedure to the SVM solution in Section 2 and we summarize in Section 3 the algorithmic implementation of it. We conclude the paper with some comments in Section 4.

2 Proof of convergence of the IRWLS algorithm to the SVC solution

The support vector classifier (SVC) seeks to compute the dependency between a set of patterns $\mathbf{x}_i \in \mathbb{R}^d$ $(i = 1, \ldots, n)$ and its corresponding labels $y_i \in \{\pm 1\}$, given a transformation to a feature space $\boldsymbol{\phi}(\cdot)$ ($\mathbb{R}^d \xrightarrow{\boldsymbol{\phi}(.)} \mathbb{R}^H$ and $d \leq H$). The SVC solves

$$\min_{\mathbf{w},\xi_i,b} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i \right\}$$

subject to:

$$y_i(\boldsymbol{\phi}^T(\mathbf{x}_i)\mathbf{w} + b) \geq 1 - \xi_i$$
$$\xi_i \geq 0$$

where $\mathbf{w}$ and b define the linear classifier in the feature space (nonlinear in the input space, unless $\boldsymbol{\phi}(\mathbf{x}) = \mathbf{x}$) and C is the penalty applied over training errors. This problem is equivalent to the following unconstrained problem, in which we need to minimize

$$L_P(\mathbf{w}, b) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} L(u_i) \tag{1}$$

with respect to $\mathbf{w}$ and b, where $u_i = 1 - y_i(\boldsymbol{\phi}^T(\mathbf{x}_i)\mathbf{w} + b)$ and $L(u) = \max(u, 0)$.

To prove the convergence of the algorithm, we need $L_P(\mathbf{w}, b)$ not only to be continuous but also differentiable, therefore we would replace $L(u)$ by a

smooth approximation:

$$L(u) = \begin{cases} 0, & u < 0 \\ Ku^2/2, & 0 \leq u < 1/K \\ u - 1/(2K), & u \geq 1/K \end{cases}$$

which tends to $\max(u, 0)$ as K approaches infinity $\left(\lim_{K \longrightarrow \infty} L(u) = \max(u, 0)\right)$.

Being the problem convex the SVM solution is achieved at $\mathbf{w}^*$ and b^* that makes the gradient vanish:

$$\nabla L_P(\mathbf{w}^*, b^*) = \begin{bmatrix} \nabla_{\mathbf{w}} L_P(\mathbf{w}^*, b^*) \\ \nabla_b L_P(\mathbf{w}^*, b^*) \end{bmatrix} = \begin{bmatrix} \mathbf{w}^* - C \sum_{i=1}^{n} \boldsymbol{\phi}(\mathbf{x}_i) y_i \left. \frac{dL(u)}{du} \right|_{u_i^*} \\ -C \sum_{i=1}^{n} y_i \left. \frac{dL(u)}{du} \right|_{u_i^*} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ 0 \end{bmatrix} \tag{2}$$

where $u_i^* = 1 - y_i(\boldsymbol{\phi}^T(\mathbf{x}_i)\mathbf{w}^* + b^*)$.

Optimization problems are solved using iterative procedures that relies in each iteration in the previous solution ($\mathbf{w}^k$ and b^k, in our case) to obtain the following one, until the optimal solution has been reached. To construct the IRWLS procedure, we modify (1) using a first order Taylor expansion of $L(u)$ over the previous solution, as it is common in other optimization procedures (Nocedal and Wright, 1999), leading to:

$$L_P^{'}(\mathbf{w}, b) = \frac{1}{2}\|\mathbf{w}\|^2 + C \left(\sum_{i=1}^{n} L(u_i^k) + \left. \frac{dL(u)}{du} \right|_{u_i^k} [u_i - u_i^k] \right)$$

where $u_i^k = 1 - y_i(\boldsymbol{\phi}^T(\mathbf{x}_i)\mathbf{w}^k + b^k)$, $L_P^{'}(\mathbf{w}^k, b^k) = L_P(\mathbf{w}^k, b^k)$ and $\nabla L_P^{'}(\mathbf{w}^k, b^k) = \nabla L_P(\mathbf{w}^k, b^k)$. Now, we construct a quadratic approximation imposing that $L_P^{''}(\mathbf{w}^k, b^k) = L_P(\mathbf{w}^k, b^k)$ and $\nabla L_P^{''}(\mathbf{w}^k, b^k) = \nabla L_P(\mathbf{w}^k, b^k)$, leading to:

$$L_P^{''}(\mathbf{w}, b) = \frac{1}{2}\|\mathbf{w}\|^2 + C \left(\sum_{i=1}^{n} L(u_i^k) + \left. \frac{dL(u)}{du} \right|_{u_i^k} \frac{(u_i)^2 - (u_i^k)^2}{2u_i^k} \right) =$$

$$= \frac{1}{2}\|\mathbf{w}\|^2 + \frac{1}{2} \sum_{i=1}^{n} a_i (1 - y_i(\boldsymbol{\phi}^T(\mathbf{x}_i)\mathbf{w} + b))^2 + \text{CT} \tag{3}$$

where

$$a_i = \frac{C}{u_i^k} \left. \frac{dL(u)}{du} \right|_{u_i^k} = \begin{cases} 0, & u_i^k < 0 \\ KC, & 0 \leq u_i^k < 1/K \\ C/u_i^k, & u_i^k \geq 1/K \end{cases}$$

and CT are constant terms that do not depend on neither $\mathbf{w}$ nor b. The IRWLS procedure consists in minimizing (3), then recomputing a_i with the obtained solution, and continue until the solution has been reached. We will focus on the algorithmic implementation in the following section, meanwhile we will demonstrate the following items to prove that the IRWLS procedure converges to the SVM solution:

- the sequence $(\mathbf{w}^0, b^0), \ldots, (\mathbf{w}^k, b^k), \ldots$ converges to $(\mathbf{w}^{op}, b^{op})$; and
- $\mathbf{w}^{op} = \mathbf{w}^*$ and $b^{op} = b^*$.

First we need to prove that the sequence of solutions converges to a limiting point in solution space $(\mathbf{w}^{op}, b^{op})$. Then, we need to assess that this limit point corresponds with the SVM solution in (2). Line search algorithms, for advancing towards the optimum, look in the minimizing functional for a descending direction, $\mathbf{p}^k$, and modifies the previous solution, $\mathbf{z}^k$, and amount η^k to obtain the following one, $\mathbf{z}^{k+1} = \mathbf{z}^k + \eta^k \mathbf{p}^k$. Wolfe conditions (Nocedal and Wright, 1999) ensure that line search methods make sufficient progress in each iteration, so the limit point is reached with any required precision, being:

$$L_P(\mathbf{z}^k + \eta^k \mathbf{p}^k) \leq L_P(\mathbf{z}^k) + c_1 \nabla L_P(\mathbf{z}^k)^T \mathbf{p}^k \tag{4}$$

$$\nabla L_P(\mathbf{z}^k + \eta^k \mathbf{p}^k)^T \mathbf{p}^k \geq c_2 \nabla L_P(\mathbf{z}^k)^T \mathbf{p}^k \tag{5}$$

for $0 < c_1 < c_2 < 1$. Wolfe conditions can be applied to the IRWLS procedure, because we can describe it as a line search method, where $\mathbf{z}^k = [(\mathbf{w}^k)^T \quad b^k]^T$, $\mathbf{p}^k = [(\mathbf{w}^s - \mathbf{w}^k)^T \quad (b^s - b^k)]^T$, where $\mathbf{w}^s$ and b^s represent the minimum of the weighted least square problem in (3).

To prove the first Wolfe condition, also known as the strictly decreasing property, we will first show that $L_P(\mathbf{z}^k) > L_P(\mathbf{z}^k + \eta^k \mathbf{p}^k) = L_P(\mathbf{z}^{k+1})$. We know that $L_P(\mathbf{w}^k, b^k) = L_P''(\mathbf{w}^k, b^k)$ and, being $\mathbf{w}^s$ and b^s the minimum of (3), $L_P''(\mathbf{w}^k, b^k) \geq L_P''(\mathbf{w}^s, b^s)$, equality will only hold if $\mathbf{w}^s = \mathbf{w}^k$ and $b^s = b^k$, due to we are solving a least square problem. Consequently, $L_P''(\mathbf{w}^k, b^k) \geq L_P''(\mathbf{w}^{k+1}, b^{k+1})$ $\forall \eta_k \in (0, 1]$, because $(\mathbf{w}^{k+1}, b^{k+1})$ are a convex combination of $(\mathbf{w}^k, b^k)$ and $(\mathbf{w}^s, b^s)$ and $L_P''(\mathbf{w}, b)$ is a convex functional, equality will only hold if $\mathbf{w}^{k+1} = \mathbf{w}^k = \mathbf{w}^s$ and $b^{k+1} = b^k = b^s$. Now

we will set η^k to enforce that $L_P''(\mathbf{w}^{k+1}, b^{k+1}) \geq L_P(\mathbf{w}^{k+1}, b^{k+1})$, to guarantee that $L_P(\mathbf{w}^k, b^k) = L_P''(\mathbf{w}^k, b^k) > L_P''(\mathbf{w}^{k+1}, b^{k+1}) \geq L_P(\mathbf{w}^{k+1}, b^{k+1})$. To show that $L_P''(\mathbf{w}^{k+1}, b^{k+1}) \geq L_P(\mathbf{w}^{k+1}, b^{k+1})$, it is sufficient to prove that $L(u_i^k) + \left.\frac{dL(u)}{du}\right|_{u_i^k} \frac{(u_i^{k+1})^2 - (u_i^k)^2}{2u_i^k} \geq L(u_i^{k+1}) \; \forall i = 1, \ldots, n$.

For $u_i^k \geq 0$, $L(u_i^k) + \left.\frac{dL(u)}{du}\right|_{u_i^k} \frac{(u_i)^2 - (u_i^k)^2}{2u_i^k}$ is tangent to $L(u_i)$ at $u_i = u_i^k$, its minimum is attained at $u_i = 0$, and its minimal value is greater or equal than zero. Therefore, in this case, $L(u_i^k) + \left.\frac{dL(u)}{du}\right|_{u_i^k} \frac{(u_i)^2 - (u_i^k)^2}{2u_i^k} \geq L(u_i)$ for any $u_i \in \mathbb{R}$. We show an example for $u_i^k = 1$ in Figure 1.

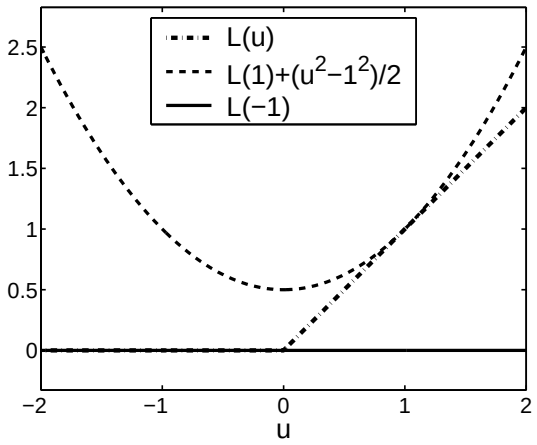

Figure 1: The dash-dotted line represents the actual SVM loss-function $L(u)$. The dashed line represents the approximation to the lost function used in (3) when $u_i^k = 1$, and the solid line represents this approximation when $u_i^k < 0$.

For $u_i^k < 0$, we need to ensure that $L(u_i^{k+1}) \leq 0$, which can only be obtained for $u_i^{k+1} \leq 0$. As $(\mathbf{w}^{k+1}, b^{k+1})$ are a convex combination of $(\mathbf{w}^k, b^k)$ and $(\mathbf{w}^s, b^s)$, u_i^{k+1} can only be greater than zero if $u_i^s > 0$. For the samples, whose $u_i^k < 0$ and $u_i^s > 0$, we will need to set $\eta_i^k \leq \frac{u_i^k}{u_i^k - u_i^s}$ to ensure that $u_i^{k+1} \leq 0$ and it can be easily checked that $0 < \eta^k < 1$. Then, if we set

$$\eta^k = \min_{\mathcal{S}} \frac{u_i^k}{u_i^k - u_i^s} \tag{6}$$

where $\mathcal{S} = \{i | \quad u_i^k < 0 \quad \& \quad u_i^s > 0\}$, we will be ensuring that $L_P''(\mathbf{w}^{k+1}, b^{k+1}) \geq L_P(\mathbf{w}^{k+1}, b^{k+1})$. In the case $\mathcal{S} = \emptyset$, we set $\mathbf{w}^{k+1} = \mathbf{w}^s$ and $b^{k+1} = b^s$ (i.e., $\eta^k = 1$), which proves that $L_P(\mathbf{z}^k + \eta^k \mathbf{p}^k) < L_P(\mathbf{z}^k)$. Now, we can set

$c_1 \in (0, c_1^*]$ to fulfill (4), where $c_1^* = \dfrac{L_P(\mathbf{z}^k + \eta^k \mathbf{p}^k) - L_P(\mathbf{z}^k)}{\nabla L_P(\mathbf{z}^k)^T \mathbf{p}^k}$ is greater than zero because $\nabla L_P(\mathbf{z}^k)^T \mathbf{p}^k < 0$, otherwise $\mathbf{p}^k$ would not be a descending direction.

Before proving the second Wolfe condition for the IRWLS, let us rewrite $L_P^{'}(\mathbf{w}, b)$ as follows:

$$L_P^{'}(\mathbf{w}, b) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} L(u_i^k) + \left.\frac{dL(u)}{du}\right|_{u_i^k} y_i[\boldsymbol{\phi}^T(\mathbf{x}_i)(\mathbf{w}^k - \mathbf{w}) + (b^k - b)]$$

and let us define

$$L_P^{'''}(\mathbf{w}, b) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} L(u_i^{k+1}) + \left.\frac{dL(u)}{du}\right|_{u_i^{k+1}} y_i[\boldsymbol{\phi}^T(\mathbf{x}_i)(\mathbf{w}^{k+1} - \mathbf{w}) + (b^{k+1} - b)]$$

which is equivalent $L_P^{'}(\mathbf{w}, b)$ but defined over the actual solution instead. Being $L_P(\mathbf{w}, b)$ convex, it can be readily seen that $L_P(\mathbf{w}, b) \geq L_P^{'}(\mathbf{w}, b)$ and $L_P(\mathbf{w}, b) \geq L_P^{'''}(\mathbf{w}, b)$ $\forall \mathbf{w} \in \mathbb{R}^H$ and $\forall b \in \mathbb{R}$.

As $(\mathbf{w}^{k+1}, b^{k+1})$ are a convex combination of $(\mathbf{w}^k, b^k)$ and $(\mathbf{w}^s, b^s)$, we can rewrite $\mathbf{p}^k = [(\mathbf{w}^s - \mathbf{w}^k)^T \quad (b^s - b^k)]^T = [(\mathbf{w}^{k+1} - \mathbf{w}^k)^T \quad (b^{k+1} - b^k)]^T / \eta^k$, leading in the left hand side of (5) to:

$$\begin{aligned}
&\eta^k \left(\nabla L_P(\mathbf{z}^{k+1})^T \mathbf{p}^k\right) = \\
&= \left[(\mathbf{w}^{k+1})^T - C\sum_{i=1}^{n} \boldsymbol{\phi}^T(\mathbf{x}_i) y_i \left.\frac{dL(u)}{du}\right|_{u_i^{k+1}} \; - C\sum_{i=1}^{n} y_i \left.\frac{dL(u)}{du}\right|_{u_i^{k+1}} \right] \begin{bmatrix} \mathbf{w}^{k+1} - \mathbf{w}^k \\ b^{k+1} - b^k \end{bmatrix} = \\
&= \|\mathbf{w}^{k+1}\|^2 - (\mathbf{w}^{k+1})^T \mathbf{w}^k - C\sum_{i=1}^{n} \left.\frac{dL(u)}{du}\right|_{u_i^{k+1}} y_i[\boldsymbol{\phi}^T(\mathbf{x}_i)(\mathbf{w}^{k+1} - \mathbf{w}^k) + (b^{k+1} - b^k)] = \\
&= \|\mathbf{w}^{k+1}\|^2 - (\mathbf{w}^{k+1})^T \mathbf{w}^k - C\sum_{i=1}^{n} \left.\frac{dL(u)}{du}\right|_{u_i^{k+1}} y_i[\boldsymbol{\phi}^T(\mathbf{x}_i)(\mathbf{w}^{k+1} - \mathbf{w}^k) + (b^{k+1} - b^k)] - \\
&\qquad - \frac{1}{2}\|\mathbf{w}^k\|^2 - C\sum_{i=1}^{n} L(u_i^{k+1}) + \frac{1}{2}\|\mathbf{w}^k\|^2 + C\sum_{i=1}^{n} L(u_i^{k+1}) = \\
&= \frac{1}{2}\|\mathbf{w}^{k+1}\|^2 - (\mathbf{w}^{k+1})^T \mathbf{w}^k + \frac{1}{2}\|\mathbf{w}^k\|^2 + \frac{1}{2}\|\mathbf{w}^{k+1}\|^2 + C\sum_{i=1}^{n} L(u_i^{k+1}) - L_P^{'''}(\mathbf{w}^k, b^k) = \\
&\qquad = \frac{1}{2}\|\mathbf{w}^{k+1} - \mathbf{w}^k\|^2 + L_P(\mathbf{w}^{k+1}, b^{k+1}) - L_P^{'''}(\mathbf{w}^k, b^k)
\end{aligned}$$

we will now repeat the same algebraic transformations over the right hand size

of (5), leading to:

$$\eta^k \left(\nabla L_P(\mathbf{z}^k)^T \mathbf{p}^k\right) =$$

$$= \left[\; (\mathbf{w}^k)^T - C\sum_{i=1}^{n} \boldsymbol{\phi}^T(\mathbf{x}_i) y_i \frac{dL(u)}{du}\bigg|_{u_i^k} - C\sum_{i=1}^{n} y_i \frac{dL(u)}{du}\bigg|_{u_i^k} \; \right] \left[\begin{array}{c} \mathbf{w}^{k+1} - \mathbf{w}^k \\ b^{k+1} - b^k \end{array} \right] =$$

$$= (\mathbf{w}^{k+1})^T \mathbf{w}^k - \|\mathbf{w}^k\|^2 + C\sum_{i=1}^{n} \frac{dL(u)}{du}\bigg|_{u_i^k} y_i[\boldsymbol{\phi}^T(\mathbf{x}_i)(\mathbf{w}^k - \mathbf{w}^{k+1}) + (b^k - b^{k+1})] +$$

$$+ \frac{1}{2}\|\mathbf{w}^{k+1}\|^2 + C\sum_{i=1}^{n} L(u_i^k) - \frac{1}{2}\|\mathbf{w}^{k+1}\|^2 - C\sum_{i=1}^{n} L(u_i^k) =$$

$$= -\frac{1}{2}\|\mathbf{w}^{k+1} - \mathbf{w}^k\|^2 - L_P(\mathbf{w}^k, b^k) + L_P^{'}(\mathbf{w}^{k+1}, b^{k+1})$$

We will now show that:

$$\frac{\|\mathbf{w}^{k+1} - \mathbf{w}^k\|^2/2 + L_P(\mathbf{w}^{k+1}, b^{k+1}) - L_P^{'''}(\mathbf{w}^k, b^k)}{\eta^k} >$$

$$> \frac{-\|\mathbf{w}^{k+1} - \mathbf{w}^k\|^2/2 - L_P(\mathbf{w}^k, b^k) + L_P^{'}(\mathbf{w}^{k+1}, b^{k+1})}{\eta^k}$$

which is equivalent to $\|\mathbf{w}^{k+1} - \mathbf{w}^k\|^2 + [L_P(\mathbf{w}^{k+1}, b^{k+1}) - L_P^{'}(\mathbf{w}^{k+1}, b^{k+1})] + [L_P(\mathbf{w}^k, b^k) - L_P^{'''}(\mathbf{w}^k, b^k)] > 0$, because $\eta^k \in (0, 1]$. The terms $L(\mathbf{w}^{k+1}, b^{k+1}) - L^{'}(\mathbf{w}^{k+1}, b^{k+1})$ and $L(\mathbf{w}^k, b^k) - L^{'''}(\mathbf{w}^k, b^k)$ are equal or greater than zero because the loss-function is convex. Moreover, $\|\mathbf{w}^{k+1} - \mathbf{w}^k\|^2 \geq 0$ and it is only zero if $\mathbf{w}^{k+1} = \mathbf{w}^k$, therefore if we are not at the solution $\nabla L_P(\mathbf{z}^{k+1})^T \mathbf{p}^k > \nabla L_P(\mathbf{z}^k)^T \mathbf{p}^k$. Now, we can set $c_2 \in [c_2^*, 1)$[1] to fulfill (5), where $c_2^* = \dfrac{\nabla L_P(\mathbf{z}^{k+1})^T \mathbf{p}^k}{\nabla L_P(\mathbf{z}^k)^T \mathbf{p}^k}$ is less than one because $\nabla L_P(\mathbf{z}^k)^T \mathbf{p}^k < 0$, otherwise $\mathbf{p}^k$ would not be a descending direction.

We now need to prove that the proposed algorithm stops when the gradient of $L_P(\mathbf{w}, b)$ vanishes. The Zoutendijk Condition (Nocedal and Wright, 1999) tell us that, if $L_P(\mathbf{w}, b)$ is bounded below and it is Lipschitz continuous[2], and the optimization procedure fulfills Wolfe Conditions, then $\|\nabla L_P(\mathbf{w}^k, b^k)\|^2 \cos^2\theta_k \to 0$ as $k \to \infty$, where $\cos^2\theta_k = \dfrac{\nabla L_P(\mathbf{w}^k, b^k)^T \mathbf{p}^k}{\|\nabla L_P(\mathbf{w}^k, b^k)\| \|\mathbf{p}^k\|}$. If we prove that θ_k does

[1] If $c_2^* < 0$, the minimum value c_2 can take is c_1, and if $c_1^* > 1$, the highest value c_1 can take is c_2, but this does not affect the given proof.

[2] $L_P(\mathbf{w}, b)$ is equal or greater than zero and it is Lipschitz continuous, because we made it differentiable.

not tend to $\pi/2$ as $k \to \infty$, we would have proven that the gradient of $L_P(\mathbf{w}, b)$ vanishes and that the proposed algorithm converges to a minimum.

Finally, we would need to prove that the achieved solution correspond to the SVM solution, which we will first prove. The minimum of (3) is obtained by solving the following linear system:

$$\begin{bmatrix} \mathbf{w} - \sum_{i=1}^{n} \boldsymbol{\phi}(\mathbf{x}_i) y_i a_i (1 - y_i(\boldsymbol{\phi}^T(\mathbf{x}_i)\mathbf{w} + b)) \\ -\sum_{i=1}^{n} y_i a_i (1 - y_i(\boldsymbol{\phi}^T(\mathbf{x}_i)\mathbf{w} + b) \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ 0 \end{bmatrix} \tag{7}$$

The IRWLS procedure stops when $\mathbf{w}^s = \mathbf{w}^k$ and $b^s = b^k$, if we replace them in (7) we are lead to:

$$\begin{bmatrix} \mathbf{w}^s - \sum_{i=1}^{n} \boldsymbol{\phi}(\mathbf{x}_i) y_i \frac{C}{u_i^k} \frac{dL(u)}{du}\Big|_{u_i^k} (1 - y_i(\boldsymbol{\phi}^T(\mathbf{x}_i)\mathbf{w}^s + b^s)) \\ -\sum_{i=1}^{n} y_i \frac{C}{u_i^k} \frac{dL(u)}{du}\Big|_{u_i^k} (1 - y_i(\boldsymbol{\phi}^T(\mathbf{x}_i)\mathbf{w}^s + b^s) \end{bmatrix} =$$

$$= \begin{bmatrix} \mathbf{w}^s - C \sum_{i=1}^{n} \boldsymbol{\phi}(\mathbf{x}_i) y_i \frac{dL(u)}{du}\Big|_{u_i^s} \\ -C \sum_{i=1}^{n} y_i \frac{dL(u)}{du}\Big|_{u_i^s} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ 0 \end{bmatrix} \tag{8}$$

which is equal to (2), consequently the IRWLS algorithm stops when it has reached the SVM solution.

To proof the sufficient condition, we need to show that if $\mathbf{w}^k = \mathbf{w}^*$ and $b^k = b^*$ the IRWLS has stopped. Suppose it has not, we can find $\mathbf{w}^s \neq \mathbf{w}^k$ and $b^s \neq b^k$ such that $L_P^{''}(\mathbf{w}^k, b^k) > L_P^{''}(\mathbf{w}^s, b^s)$, and the strictly decreasing property will lead to $L_P(\mathbf{w}^*, b^*) > L_P(\mathbf{w}^s, b^s)$, which is a contradiction because $\mathbf{w}^*$ and b^* give the minimum of $L_P(\mathbf{w}, b)$. We have just proven that if the IRWLS has stopped we will be at the SVM solution and if we are at the SVM solution the IRWLS has stopped.

Finally, we do not need to prove that θ_k does not tend to $\pi/2$, because we have just shown that the algorithms stops iff we are at the SVM solution and, consequently, this is the point in which the gradient of $L_P(\mathbf{w}, b)$ vanishes, that was what we still needed to prove, ending the proof of convergence.

3 Iterative Re-Weighted Least Squares for Support Vector Classifiers

The IRWLS procedure, when first introduced in (Pérez-Cruz et al., 1999), did not consider the modification to ensure convergence presented in the previous section (i.e. $\eta^k < 1$ in some iterations). We will know describe the algorithmic implementation of the procedure. But, before presenting the algorithm, let us rewrite (7) in matrix form,

$$\begin{bmatrix} \mathbf{\Phi}^T\mathbf{D}_a\mathbf{\Phi}+\mathbf{I} & \mathbf{\Phi}^T\mathbf{a} \\ \mathbf{a}^T\mathbf{\Phi} & \mathbf{a}^T\mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ b \end{bmatrix} = \begin{bmatrix} \mathbf{\Phi}^T\mathbf{D}_a\mathbf{y} \\ \mathbf{a}^T\mathbf{y} \end{bmatrix} \tag{9}$$

where $\mathbf{\Phi} = [\boldsymbol{\phi}(\mathbf{x}_1), \boldsymbol{\phi}(\mathbf{x}_2), \ldots, \boldsymbol{\phi}(\mathbf{x}_n)]^T$, $\mathbf{y} = [y_1, \ldots, y_n]^T$, $\mathbf{a} = [a_1, \ldots, a_n]^T$, $(\mathbf{D}_a)_{ij} = a_i\delta_{ij}$ $(\forall i, j = 1, \ldots, n)$, $\mathbf{I}$ is the identity matrix and $\mathbf{1}$ is a column-vector of n ones.

This system can be solved using kernels, as well as the regular SVM, by imposing that $\mathbf{w} = \sum_i \boldsymbol{\phi}(\mathbf{x}_i)y_i\alpha_i$ and $\sum_i \alpha_i y_i = 0$. These conditions can be obtained from the regular SVM solution (KKT conditions), see (Schölkopf and Smola, 2001) for further details. Also, they can be derived from (2) in which the α_i have replaced the derivative of $L(u_i)$. The system in (9) becomes

$$\begin{bmatrix} \mathbf{H}+\mathbf{D}_a{}^{-1} & \mathbf{y} \\ \mathbf{y}^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ b \end{bmatrix} = \begin{bmatrix} \mathbf{1} \\ 0 \end{bmatrix} \tag{10}$$

where $(\mathbf{H})_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) = \boldsymbol{\phi}^T(\mathbf{x}_i)\boldsymbol{\phi}(\mathbf{x}_j)$ and $k(\cdot,\cdot)$ is the kernel of the nonlinear transformation $\boldsymbol{\phi}(\cdot)$ (Schölkopf and Smola, 2001). The steps to derive (10) from (9) can be found in (Pérez-Cruz et al., 2001).

The Iterative Re-Weighted Least Square (IRWLS) can be summarized in the following steps:

1. Initialization: set $k = 0$, $\boldsymbol{\alpha}^0 = \mathbf{0}$, $b^0 = 0$ and $u_i^0 = 1 \quad \forall i = 1, \ldots, n$.
2. Solve (10) to obtain $\boldsymbol{\alpha}^s$ and b^s.
3. Compute u_i^s. Construct $\mathcal{S} = \{i | u_i^k < 0 \quad \& \quad u_i^s > 0\}$. If $\mathcal{S} = \emptyset$, set $\boldsymbol{\alpha}^{k+1} = \boldsymbol{\alpha}^s$ and $b^{k+1} = b^s$ and go to 5.

4. Compute η^k, using (6), and $\boldsymbol{\alpha}^{k+1}$ and b^{k+1}. If $L(\boldsymbol{\alpha}^{k+1}, b^{k+1}) > L(\boldsymbol{\alpha}^s, b^s)$, set $\boldsymbol{\alpha}^{k+1} = \boldsymbol{\alpha}^s$ and $b^{k+1} = b^s$.

5. Set $k = k + 1$ and go to 2 until convergence.

The modification in the third step helps to give a further decrease in the SVM functional. Because the value of η^k in (6) is a sufficient condition, but it is not a necessary one, and in some cases, $\boldsymbol{\alpha}^s$ and b^s can produce a further decrease in $L(\boldsymbol{\alpha}, b)$ than using $\boldsymbol{\alpha}^{k+1}$ and b^{k+1}.

It is worth pointing out that the solution achieved in the first step coincides with the Least Square Support Vector Machine Solution (Suykens et al., 2003), as $\mathbf{D_a}$ is the identity matrix multiplied by C. In a way, we can say that the starting point of the IRWLS procedure is the LS-SVM solution.

4 Comments on the IRWLS procedure

In the present article, we have proven the convergence of the IRWLS procedure to the SVM solution. This algorithm was particularly devised for solving SVMs and presents several properties that makes it desirable. First, the IRWLS algorithm, as it name indicates, only needs to solve a simple least square problem in each iteration. Moreover the linear system is only formed by the samples whose $u_i^k > 0$, while those samples whose $u_i^k < 0$ will not affect the functional $L_P''(\mathbf{w}, b)$ in (3). This property allows to work only with part of the kernel matrix, significantly reducing the runtime complexity. Second, during the first iterations, that are the most computationally costly, many samples changes the value of u_i from positive to negative. Moreover if $\eta^k < 1$ a sample whose $u_i^k < 0$ end with $u_i^{k+1} = 0$, and in the next iteration its $a_i = KC$. Therefore if any sample whose $u_i^* \geq 0$, but in some intermediate iteration $u_i^k < 0$, the algorithm will recover it. The IRWLS changes several constraints from active to inactive in one iteration, while most QP algorithms stop when a constraint changes. We illustrate this property with a simple example in Figure 2. Third, we would like to point out that the value of η^k is 1 in most iterations and only seldom a sample changes from $u_i^k < 0$ to $u_i^s > 0$

and $L(\mathbf{w}^{k+1}, b^{k+1}) \leq L(\mathbf{w}^s, b^s)$, which explains why the IRWLS was working correctly without this modification.

The role of K can be analysed from the proof and implementation perspectives. A finite value of K allows to demonstrate that the IRWLS procedure converges to the SVM solution. If K was infinite, the functional would not be differentiable, and although (2) and (8) will be equal, would not mean that $\mathbf{w}^{op}$ (b^{op}) is equal to $\mathbf{w}^*$ (b^*). From the implementation viewpoint, we are adding at least $1/K$ to the diagonal of $\mathbf{H}$, therefore if $\mathbf{H}$ is nearly singular (as it is for most problems of interest even for infinite VC dimension kernels) a finite value of K would avoid numerical instabilities. We usually fix K between 10^4 and 10^{10}, depending on the machine precision and do not modify it during training. For this range of K the solution does not vary significantly, meaning that we are close to or at the SVM solution. A software package in MATLAB can be downloaded from our webpage http://www.gatsby.ucl.ac.uk/∼fernando.

4.1 Extending the IRWLS procedure

We have proven the convergence of the IRWLS procedure to the standard SVM solution, but this procedure is sufficiently general to be applied to other loss-functions. It can be directly applied to any convex, continuous and differentiable almost everywhere loss-function. As we rely on the derivative of the loss, to demonstrate that the limiting point of the IRWLS procedure is the solution to the actual functional, and as we need the first order Taylor expansion to be a lower bound on the loss-function, to show the sufficient decreasing property. To complete the IRWLS procedure, one needs to come up with a quadratic approximation, which has to be at least locally an upper bound to the loss-function to ensure the strictly decreasing property. This upper bound has to take the same value and derivative the loss-function does at the actual solution, to ensure that the sequence of solutions converges to the solution of our functional.

The question that readily arises is if it can be employed for non-convex loss-functions. First of all, if at all used, it would only lead to a local minimum and another method would be needed to assess the quality of the obtained

solution. If the loss-function is not convex, the sufficient decreasing property would not hold for every possible solution found by the second step of the IRWLS procedure and further analysis would be necessary, taking into account the shape of the non-convex function, to demonstrate the convergence of the algorithm.

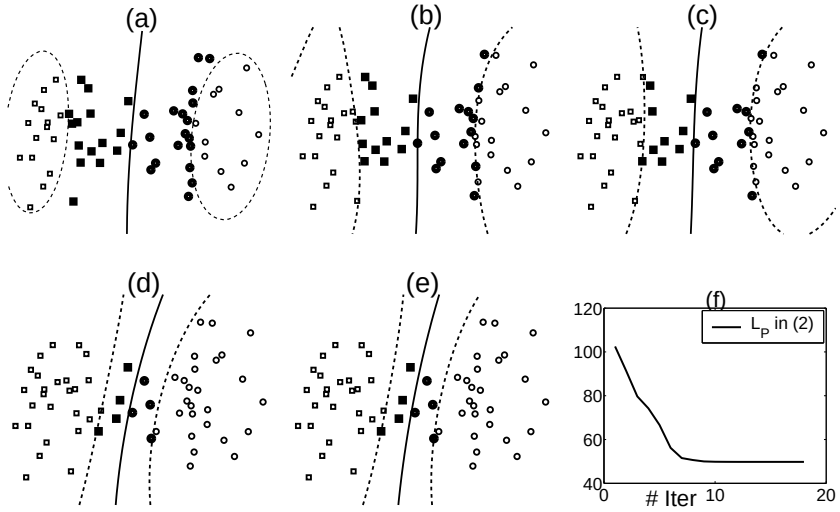

Figure 2: We show the intermediate solution of the IRWLS procedure for a simple problem in (a)-(e), respectively, for iterations 1, 2, 3, 8 and 18 (final). In (f) the value of L_P for every iteration. The squares represent the negative-class samples and the circles the positive-class samples. The solid circles and squares are those samples, whose $u_i^{k+1} > 0$. The solid line is the classification boundary and the dashed lines represent the ± 1 margins. It can be seen that in the first step almost half of the samples changes from a $u_i^0 > 0$ to a $u_i^1 < 0$, significantly advancing towards the optimum and reducing the complexity of subsequent iterations. It can be seen that solution in iteration 8 is almost equal to the solution in iteration 18, and in these intermediate iterations the algorithm is only fine tuning the values of α_i.

Acknowledgements

We would like to kindly thanks Chih-Jen Lin for his useful comments and pointing out the weakness of our previous proofs.

References

Burges, C. J. C. (1998). A Tutorial on Support Vector Machines for Pattern Recognition. *Knowledge Discovery and Data Mining*, 2(2):121–167.

Mangasarian, O. L. and Musicant, D. R. (2000). Lagrangian support vector machines. *Journal of Machine Learning Research*, pages 161–177.

Nocedal, J. and Wright, S. J. (1999). *Numerical Optimization*. Springer.

Pérez-Cruz, F., Alarcón-Diana, P. L., Navia-Vázquez, A., and Artés-Rodríguez, A. (2000a). Fast training of support vector classifiers. In *Advances in Neural Information Processing Systems 13*, Cambridge, MA. M.I.T. Press.

Pérez-Cruz, F., Navia-Vázquez, A., Alarcón-Diana, P. L., and Artés-Rodríguez, A. (2000b). An IRWLS procedure for SVR. In *EUSIPCO'00*, Tampere, Finland.

Pérez-Cruz, F., Navia-Vázquez, A., Alarcón-Diana, P., and Artés-Rodríguez, A. (2001). SVC-based equalizer for burst TDMA transmissions. *Signal Processing*, 81(8):1681–1693.

Pérez-Cruz, F., Navia-Vázquez, A., Rojo-Álvarez, J. L., and Artés-Rodríguez, A. (1999). A new training algorithm for support vector machines. In *Fifth Bayona Workshop on Emerging Technologies in Telecommunications*, pages 116–120, Baiona, Spain.

Schölkopf, B. and Smola, A. (2001). *Learning with kernels*. M.I.T. Press.

Suykens, J. A. K., Van Gestel, T., De Brabanter, J., De Moor, B., and Vandewalle, J. (2003). *Least Squares Support Vector Machines*. World Scientific Pub Co.

Suykens, J. A. K. and Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural Processing Letters*, 9(3):293–300.

Van Gestel, T., Suykens, J. A. K., Baesens, B., Vanthienen, S., Dedene, G., De Moor, B., and Vandewalle, J. (2004). Benchmarking least squares support vector machines classifiers. *Machine Learning*, 54(1):5–32.

Vapnik, V. N. (1998). *Statistical Learning Theory*. John Wiley & Sons, New York.