

Regularization paths for ν -SVM and ν -SVR

Gaëlle Loosli, Gilles Gasso, and Stéphane Canu

LITIS, EA 4051, Rouen, France
firstname.name@insa-rouen.fr

Abstract. This paper presents the ν -SVM and the ν -SVR full regularization paths along with a leave-one-out inspired stopping criterion and an efficient implementation. In the ν -SVR method, two parameters are provided by the user: the regularization parameter C and ν which settles the width of the ϵ -tube. In the classical ν -SVM method, parameter ν is an lower bound on the number of support vectors in the solution. Based on the previous works of [1, 2], extensions of regularization paths for SVM and SVR are proposed and permit to automatically compute the solution path by varying ν or the regularization parameter.

1 Introduction

The utilization of SVM by neophyte users is still hampered by the need to supply values for control parameters in order to get the best attainable results. Mainly, SVM's users must make three choices: the kernel, its bandwidth and the regularization parameter. Within the usual formulation of the soft margins SVM, the regularization parameter takes its value between 0 (random) and ∞ (hard-margins). The ν -SVM [3] technique reformulates the SVM problem so that C is replaced by a ν parameter taking values in $[0, 1]$. This re-normalized parameter has a more intuitive meaning: it represents the minimal proportion of points in the solution and the maximal proportion of misclassified points. The SVR algorithm is a popular method for dealing with regression problems [4]. The algorithm minimizes the ϵ -insensitive cost while preserving the smoothness of the regression function. The trade-off is realized via a regularization parameter λ set by the user. The user also provides the width $\epsilon \in [0, \infty[$ of the tube. As the practical choice of ϵ is difficult, the ν -SVR method [4] was proposed and permits to automatically determine the value of ϵ . Given the importance of this problem for reaping all the potential benefits of the use of SVM and SVR, many research work have been dedicated to ways of helping the setting of the parameters. Most rely on either *outer* measures, such as cross-validation, to guide the selection, or to measures embedded in the learning method itself (see [5, 6]). In place of these empirical approaches regularization paths have been widely studied recently [1, 2, 7] since they provide a smart and fast way to access all the optimal solutions of a problem according to all compromises between bias and regularity. However, having the whole regularization path is not enough. Indeed, the end user still needs to retrieve from it the best values for the regularization parameters. Instead of selecting these values by k -fold cross-validation or leave-one-out, or other

approximations [8], we propose to include the leave-one-out estimator inside the regularization path in order to have an idea of the generalization error at each step. We explain why it is less expansive than selecting the best parameter *a posteriori* and give a method to stop learning before attaining the end of the path to save useless efforts. This paper presents a low-cost (in term of computational time and memory) method for the auto-setting of the regularization parameter for the classification case. Contrarily to what is usually done for regularization path, our method does not start with all points as support vectors. Doing so we avoid the computation of the whole Gram matrix at the first step. Then, since the proposed method stops on the path, this extreme non-sparse solution is never attained and thus the whole Gram matrix never required. One of the main advantage of this is that it is possible to use this setting for large databases.

2 Regularization path for ν -SVM

We consider a binary classification problem with training patterns $x_1 \dots x_m \in \mathcal{X}$ and associated classes $y_1 \dots y_m \in \{+1, -1\}$. A ν -SVM classifies a pattern x according to the sign of the decision function $f(\cdot) = \frac{1}{m} \sum_{i=1}^m \alpha_i y_i k(x_i, \cdot) + b$. A pattern x_i is called “support vector” when the corresponding coefficient $\alpha_i \neq 0$. The hyper-parameter ν can be scaled by the size of the training database. In this case, we have $\lambda = m\nu$ and λ is a lower bound on the number of support vectors. Since we have at least a point in the solution, we can set $1 \leq \lambda \leq m$. The primal ν -SVM problem is written as follows:

$$\begin{cases} \min_{f,b,\rho,\xi_i} & \frac{m}{2} \|f\|^2 - \lambda\rho + \sum_{i=1}^m \xi_i \\ \text{s.t.} & y_i(f(x_i) + b) \geq \rho - \xi_i, \quad \xi_i \geq 0, \forall i \in [1, \dots, m] \quad \text{and} \quad \rho \geq 0 \end{cases}$$

with ρ being the margin to be optimized and the dual problem is:

$$\begin{cases} \max_{\alpha} & -\frac{1}{2} \alpha^\top G \alpha \\ \text{s.t.} & \alpha^\top \mathbf{1} \geq \lambda, \quad \alpha^\top \mathbf{y} = 0 \quad \text{and} \quad 0 \leq \alpha_i \leq 1 \quad \forall i \in [1, \dots, m] \end{cases} \quad (1)$$

where $G(i, j) = \frac{1}{m} y_i y_j k(x_i, x_j)$. Our aim is to compute the ν -SVM solution for all values of ν . As shown in [1], the path is piecewise linear. It means that the support vectors set does not change between two values of ν . Hence we only need to identify when a change occurs in the sets. Similarly to what is done in active set methods solving the SVM [9], the first steps consists in identifying the best direction to follow and the second step is to determine how far to follow it. Let note $g(x_i) = y_i(f(x_i) + b) - \rho$. Then we have:

$$\begin{cases} \mathcal{L} : g(x_i) < 0 \quad \forall i \in \mathcal{L} & \alpha_i = 1 & \text{bounded points} \\ \mathcal{E} : g(x_i) = 0 \quad \forall i \in \mathcal{E} & 0 < \alpha_i < 1 & \text{margins points} \\ \mathcal{R} : g(x_i) > 0 \quad \forall i \in \mathcal{R} & \alpha_i = 0 & \text{useless points} \end{cases}$$

The idea of the path is to provide an iterative method that follows the path by pieces, stopping at each change among the groups. Each point of the path

reflects the optimal solution of the ν -SVM according to a particular value of λ . We begin with the smallest value of λ and let it grow up to its larger value. Since the provided solutions are equivalent as long as the groups do not change, we only need to identify for which values of λ a point is going to move from one group to another. We identify four possible movements: from \mathcal{L} to \mathcal{E} (from the wrong side of the margin to the margin), from \mathcal{R} to \mathcal{E} (the point becomes support vector), from \mathcal{E} to \mathcal{L} (a support vector becomes bounded) and from \mathcal{E} to \mathcal{R} (a support vector is no longer one). We will denote those steps respectively as $in(\ell)$, $in(r)$, $out(\ell)$ and $out(r)$. Events $in(\ell)$ and $in(r)$ happen when $\exists i \in \mathcal{L}$ or $i \in \mathcal{R}$ such that $g(x_i) = 0$. Events $out(\ell)$ and $out(r)$ occur respectively when $\exists i \in \mathcal{E}$ such that $\alpha_i = 1$ or $\alpha_i = 0$. Hence we need to express $g(x)$ and α depending on steps t and $t + 1$ as:

$$\begin{aligned} g^{t+1}(x_i) &= g^{t+1}(x_i) - g^t(x_i) + g^t(x_i) \\ &= G(i, :) \alpha^{t+1} + b^{t+1} y_i - \rho^{t+1} - G(x_i, :) \alpha^t - b^t y_i + \rho^t + g^t(x_i) \quad (2) \\ &= G(i, :) \delta_\alpha + \delta_b y_i - \delta_\rho + g^t(x_i) \end{aligned}$$

with $\delta_\alpha = \alpha^{t+1} - \alpha^t$, $\delta_b = b^{t+1} - b^t$ and $\delta_\rho = \rho^{t+1} - \rho^t$. $G(i, :)$ designates the i^{th} row of the matrix. From equation 2 and depending on the concerned event, it is possible to find which value of λ to choose next. We summarize in table 1 the events and the algorithm mechanic.

| Step | $in(r)$ | $out(r)$ | $out(\ell)$ | $in(\ell)$ |
|---------|---|---|---|---|
| t | $i \in \mathcal{R}$ $g^t(x_i) > 0$ $\alpha_i = 0$ | $i \in \mathcal{E}$ $\star g^t(\mathbf{x}_i) = \mathbf{0}$ $0 < \alpha_i < 1$ | $i \in \mathcal{E}$ $\star g^t(\mathbf{x}_i) = \mathbf{0}$ $0 < \alpha_i < 1$ | $i \in \mathcal{L}$ $g^t(x_i) < 0$ $\alpha_i = 1$ |
| $t + 1$ | $i \in \mathcal{E}$ $\star g^{t+1}(\mathbf{x}_i) = \mathbf{0}$ $0 < \alpha_i < 1$ | $i \in \mathcal{R}$ $\star g^{t+1}(\mathbf{x}_i) \geq \mathbf{0}$ $\star \alpha_i = \mathbf{0}$ | $i \in \mathcal{L}$ $\star g^{t+1}(\mathbf{x}_i) \leq \mathbf{0}$ $\star \alpha_i = \mathbf{1}$ | $i \in \mathcal{E}$ $\star g^{t+1}(\mathbf{x}_i) = \mathbf{0}$ $0 < \alpha_i < 1$ |

Table 1. Summary of the events. Each column stands for a particular event. In blue starred are noted the properties that are used to compute the corresponding λ^{t+1}

Points in \mathcal{E} and detection of $out(\ell)$ and $out(r)$ Events $out(\ell)$ and $out(r)$ are detected using their values of α . Indeed, one condition to remain in \mathcal{E} is to keep $0 < \alpha < 1$. Retrieving λ^{t+1} for which one of these conditions is violated for each point requires to write α_i depending on λ^{t+1} . Remark that in \mathcal{E} , $g^t(x) = 0$ and $g^{t+1}(x) = 0$. Equation 2 together with the constraints from 1 ($\alpha^\top \mathbf{1} \geq \lambda$, hence $\delta_\alpha^\top \mathbf{1} \geq \lambda^{t+1} - \lambda^t$ and $\alpha^\top \mathbf{y} = 0$, hence $\delta_\alpha^\top \mathbf{y} = 0$) leads to a system of linear equations $A\delta = (\lambda^{t+1} - \lambda)\mathbf{c}$, where

$$A = \begin{bmatrix} G & \mathbf{y} - \mathbf{1} \\ \mathbf{y}^\top & 0 & 0 \\ \mathbf{1}^\top & 0 & 0 \end{bmatrix} \quad \delta = \begin{bmatrix} \delta_\alpha \\ \delta_b \\ \delta_\rho \end{bmatrix} \quad \mathbf{c} = \begin{bmatrix} \mathbf{0} \\ 0 \\ 1 \end{bmatrix}$$

This leads to $\delta = (\lambda^{t+1} - \lambda)A^{-1}\mathbf{c}$. So for points in the set \mathcal{E} there is:

$$\begin{cases} \alpha^{t+1} = \alpha^t + (\lambda^t - \lambda^{t+1})\eta_\alpha \\ b^{t+1} = b^t + (\lambda^t - \lambda^{t+1})\eta_b \\ \rho^{t+1} = \rho^t + (\lambda^t - \lambda^{t+1})\eta_\rho \end{cases}$$

where $\boldsymbol{\eta}$ denotes the vector $A^{-1}\mathbf{c}$. As mentioned earlier, a change in the groups will occur as soon as one of the α_i will meet one of the boundaries, *i.e.* when $\alpha_i = 0$ or $\alpha_i = 1$ for $i \in \mathcal{E}$. This gives two change conditions:

$$\lambda_{out(r)}^{t+1} = \frac{-\alpha_i^t}{\eta_i} + \lambda^t \quad \text{and} \quad \lambda_{out(\ell)}^{t+1} = \frac{1-\alpha_i^t}{\eta_i} + \lambda^t$$

Points in \mathcal{L} and \mathcal{R} and detection of $in(\ell)$ and $in(r)$ Events $in(\ell)$ and $in(r)$ are detected using their values of $g(x_i)$. Indeed, one condition to remain in \mathcal{R} is to keep $g(x_i) > 0$ and $g(x_i) < 0$ to stay in \mathcal{L} . Retrieving λ^{t+1} for which one of these conditions is violated for each point requires to write $g^{t+1}(x_i)$ depending on λ^{t+1} . For a point moving inside \mathcal{E} , the particularity is that $g^{t+1}(x_i)$ becomes nul. Defining $h(x) = G(i, :)\boldsymbol{\eta}\boldsymbol{\alpha} + \eta_b - \eta_\rho$ leads to

$$\begin{aligned} g^{t+1}(x_i) &= G(i, :)\boldsymbol{\delta}\boldsymbol{\alpha} + \delta_b y_i - \delta_\rho + g^t(x_i) \\ &= (\lambda^{t+1} - \lambda^t)(G(i, :)\boldsymbol{\eta}\boldsymbol{\alpha} + \eta_b - \eta_\rho) + g^t(x_i) \\ &= (\lambda^{t+1} - \lambda^t)h^t(x_i) + g^t(x_i) = 0 \end{aligned}$$

and thus

$$\lambda_{in(\ell)}^{t+1} \frac{-g^t(x_i)}{h^t(x_i)} + \lambda^t \quad i \in \mathcal{L} \quad \text{and} \quad \lambda_{in(r)}^{t+1} \frac{-g^t(x_i)}{h^t(x_i)} + \lambda^t \quad i \in \mathcal{R}$$

Note that it may happen that several points reach \mathcal{E} at the same time. Even though this does not change equations, it is a relevant remark for implementation. Indeed, if a point is missed, the path is left and the missed point will not be selected afterward.

Regularization path algorithm At each step we look for the smallest $\lambda^{t+1} > \lambda^t$ among $\{\lambda_{in(\ell)}^{t+1}, \lambda_{in(r)}^{t+1}, \lambda_{out(\ell)}^{t+1}, \lambda_{out(r)}^{t+1}\}$. We update the $\boldsymbol{\alpha}^{t+1}$ according to the chosen λ^{t+1} and then the groups. The process is stopped when $\lambda = m$.

3 Regularization path for the ν -SVR

Assuming m training points $\{(x_i, y_i) \in \mathcal{X} \times \mathbb{R}\}$, the ν -SVR algorithm optimizes an ϵ -insensitive cost ($L(y, f(x)) = \max(0, |y - f(x)| - \epsilon)$) and allows the automatic computation of the ϵ -tube [4]. Its primal formulation is:

$$\begin{cases} \min_{f, b, \epsilon, \xi, \xi^*} & \frac{\lambda}{2} \|f\|^2 + \nu\epsilon + \sum_{i=1}^m \xi_i + \xi_i^* & s.t. \\ -\epsilon - \xi_i \leq y_i - f(x_i) \leq \epsilon + \xi_i^*, & \xi_i \geq 0, \xi_i^* \geq 0, \forall i \in [1, \dots, m] \text{ and } \epsilon \geq 0 \end{cases}$$

where λ is the regularization parameter. According to this formulation, the parameter ν varies in the interval $[0, m]$.

3.1 The double path

The regression function: $f(x) = \frac{1}{\lambda} \sum_{i=1}^m (\alpha_i^* - \alpha_i)k(x_i, x) + b$ is a solution of the previous problem where $k(\cdot, \cdot)$ is the kernel function and the Lagrange multipliers $\alpha_i^{(*)}$ are solutions of the dual problem [4]. Given fixed values of the regularization

parameter λ and of ν , the KKT conditions permit the automatic determination of b and ϵ (see [4]). The quality of the regression depends on the chosen values. The aim of this section is to analyze the evolution of the regression function $f(x)$ according to the variations of λ for a fixed value ν : the λ -path. Conversely, keeping λ fixed at a specified value, the regression function can be studied with respect to ν : the ν -path. Following the original idea of [2] it can be shown that these paths are piecewise linear. The initial regularization path for SVR of Gunter and Zhu was extended to the double path (λ -path and ϵ -path) in [10]. In this paper, we give another formulation of the double path as we compute the ν -path instead of the ϵ -path. To do so, let define the following sets:

$$\begin{cases} \mathcal{L} : y_i - f(x_i) < -\epsilon, \forall i \in \mathcal{L}, & \alpha_i = 1, \alpha_i^* = 0 & \text{bounded points} \\ \mathcal{R} : y_i - f(x_i) > \epsilon, \forall i \in \mathcal{R}, & \alpha_i = 0, \alpha_i^* = 1 & \text{bounded points} \\ \mathcal{C} : |y_i - f(x_i)| < \epsilon, \forall i \in \mathcal{C}, & \alpha_i = 0, \alpha_i^* = 0 & \text{useless points} \\ \mathcal{E}_{\mathcal{L}} : y_i - f(x_i) = -\epsilon, \forall i \in \mathcal{E}_{\mathcal{L}}, & 0 \leq \alpha_i \leq 1, \alpha_i^* = 0 & \text{useful points} \\ \mathcal{E}_{\mathcal{R}} : y_i - f(x_i) = \epsilon, \forall i \in \mathcal{E}_{\mathcal{R}}, & \alpha_i = 0, 0 \leq \alpha_i^* \leq 1 & \text{useful points} \end{cases}$$

The sets \mathcal{L} and \mathcal{R} contain respectively the points with errors belonging to the left part and right part of the ϵ -tube whereas the points of $\mathcal{E}_{\mathcal{L}}$ and $\mathcal{E}_{\mathcal{R}}$ lie on the left and right elbows (on the tube). The elements of \mathcal{C} are the points in the tube. Compared to the ν -SVM, we remark that there is more groups of points to tract.

3.2 Computation of the λ -path

We suppose the value of ν is constant. The regression function can be written as: $f(x) = \frac{1}{\lambda} (\sum_{i=1}^m (\alpha_i^* - \alpha_i) k(x_i, x) + \beta_0)$ with $\beta_0 = \lambda b$. Let $f^t(x)$, the solution obtained for λ^t . The corresponding sets are $\mathcal{L}^t, \mathcal{R}^t, \mathcal{C}^t, \mathcal{E}_{\mathcal{L}}^t, \mathcal{E}_{\mathcal{R}}^t$. The solution is not modified as long as the sets are not modified. As previously, the key point is to determine the values of λ for which a point is moved from a set to another. The conditions for the occurring of these events are summarized in table 2. Let

| Step | $in(\ell)$ \mathcal{L}^t to $\mathcal{E}_{\mathcal{L}}^t$ | $in(r)$ \mathcal{R}^t to $\mathcal{E}_{\mathcal{R}}^t$ | $in(c)$ \mathcal{C}^t to $\mathcal{E}_{\mathcal{L}}^t$ or \mathcal{C}^t to $\mathcal{E}_{\mathcal{R}}^t$ | $out(\ell)$ $\mathcal{E}_{\mathcal{L}}^t$ to \mathcal{L}^t | $out(r)$ $\mathcal{E}_{\mathcal{R}}^t$ to \mathcal{R}^t | $out(c)$ $\mathcal{E}_{\mathcal{L}}^t$ to \mathcal{C}^t or $\mathcal{E}_{\mathcal{R}}^t$ to \mathcal{C}^t |
|-------|--|---|--|--|---|---|
| t | $i \in \mathcal{L}$ $r_i^t < -\epsilon^t$ $\alpha_i = 1$ $\alpha_i^* = 0$ | $i \in \mathcal{R}$ $r_i^t > \epsilon^t$ $\alpha_i = 0$ $\alpha_i^* = 1$ | $i \in \mathcal{C}$ $ r_i^t < \epsilon^t$ $\alpha_i = 0$ $\alpha_i^* = 0$ | $i \in \mathcal{E}_{\mathcal{L}}$ $r_i^t = -\epsilon^t$ $0 \leq \alpha_i \leq 1$ $\alpha_i^* = 0$ | $i \in \mathcal{E}_{\mathcal{R}}$ $r_i^t = \epsilon^t$ $\alpha_i = 0$ $0 \leq \alpha_i^* \leq 1$ | $i \in \mathcal{E}_{\mathcal{L}}$ or $\mathcal{E}_{\mathcal{R}}$ |
| $t+1$ | $i \in \mathcal{E}_{\mathcal{L}}$ $r_i^{t+1} = -\epsilon^{t+1}$ $0 \leq \alpha_i \leq 1$ $\alpha_i^* = 0$ | $i \in \mathcal{E}_{\mathcal{R}}$ $r_i^{t+1} = \epsilon^{t+1}$ $\alpha_i = 0$ $0 \leq \alpha_i^* \leq 1$ | $i \in \mathcal{E}_{\mathcal{L}}$ or $i \in \mathcal{E}_{\mathcal{R}}$ $r_i^{t+1} = -\epsilon^{t+1}$ or $r_i^{t+1} = \epsilon^{t+1}$ $0 \leq \alpha_i \leq 1, \alpha_i = 0$ $\alpha_i^* = 0, 0 \leq \alpha_i^* \leq 1$ | $i \in \mathcal{L}$ $r_i^{t+1} < -\epsilon^{t+1}$ $\alpha_i = 1$ $\alpha_i^* = 0$ | $i \in \mathcal{R}$ $r_i^{t+1} > \epsilon^{t+1}$ $\alpha_i = 0$ $\alpha_i^* = 1$ | $i \in \mathcal{C}$ $ r_i^{t+1} < \epsilon^{t+1}$ $\alpha_i = 0$ $\alpha_i^* = 0$ |

Table 2. Events in the doubly path of ν -SVR. The bold blue color denotes the conditions used to compute the parameters of the model and $r_i^k = y_i - f^k(x_i)$. Some elements of the last column are not specified as they are given in the previous columns.

write $\lambda f(x) = \lambda f(x) - \lambda^t f^t(x) + \lambda^t f^t(x)$. Hence we have:

$$\lambda f(x) = \sum_{i \in \mathcal{E}_{\mathcal{L}}^t \cup \mathcal{E}_{\mathcal{R}}^t} (\delta \alpha_i^* - \delta \alpha_i) k(x_i, x) + \delta \beta_0 + \lambda^t f^t(x) \quad (3)$$

with $\delta \alpha_i = \alpha_i - \alpha_i^t$, $\delta \alpha_i^* = \alpha_i^* - \alpha_i^{*t}$, $\delta \beta_0 = \beta_0 - \beta_0^t$. In the latest relation, the sum is carried only over $\mathcal{E}_{\mathcal{L}}$ and $\mathcal{E}_{\mathcal{R}}$ as the Lagrange parameters corresponding to the other sets are fixed (equal to 0 or 1). For $j \in \mathcal{E}_{\mathcal{L}}^t$, we have: $y_j - f^t(x_j) = -\epsilon^t$. Therefore, for $\lambda^{t+1} < \lambda < \lambda^t$, the following equation holds: $\lambda(y_j + \epsilon) = \sum_{i \in \mathcal{E}_{\mathcal{R}}^t \cup \mathcal{E}_{\mathcal{L}}^t} (\delta \alpha_i^* - \delta \alpha_i) k(x_i, x_j) + \delta \beta_0 + \lambda^t (y_j + \epsilon^t)$. By defining $d = \lambda \epsilon$ and $\delta d = \lambda \epsilon - \lambda^t \epsilon^t$, we get:

$$(\lambda - \lambda^t) y_j = \sum_{i \in \mathcal{E}_{\mathcal{R}}^t} \delta \alpha_i^* k(x_i, x_j) - \sum_{i \in \mathcal{E}_{\mathcal{L}}^t} \delta \alpha_i k(x_i, x_j) + \delta \beta_0 - \delta d, \quad \forall j \in \mathcal{E}_{\mathcal{L}}^t$$

Similarly, for $j \in \mathcal{E}_{\mathcal{R}}^t$, one can establish:

$$(\lambda - \lambda^t) y_j = \sum_{i \in \mathcal{E}_{\mathcal{R}}^t} \delta \alpha_i^* k(x_i, x_j) - \sum_{i \in \mathcal{E}_{\mathcal{L}}^t} \delta \alpha_i k(x_i, x_j) + \delta \beta_0 + \delta d, \quad \forall j \in \mathcal{E}_{\mathcal{R}}^t$$

Using the constraints $(\alpha^* - \alpha)^\top \mathbf{1} = 0$ (which leads to $(\delta \alpha^* - \delta \alpha)^\top \mathbf{1} = 0$) and $(\alpha^* + \alpha)^\top \mathbf{1} \leq \nu$ (hence $(\delta \alpha^* + \delta \alpha)^\top \mathbf{1} \leq 0$) of the dual problem, we obtain the linear system $A \delta = (\lambda - \lambda^t) \mathbf{c}$ where:

$$A = \begin{bmatrix} -K(\mathcal{E}_{\mathcal{L}}, \mathcal{E}_{\mathcal{L}}) & K(\mathcal{E}_{\mathcal{L}}, \mathcal{E}_{\mathcal{R}}) & 1 & -1 \\ -K(\mathcal{E}_{\mathcal{L}}, \mathcal{E}_{\mathcal{R}})^\top & K(\mathcal{E}_{\mathcal{R}}, \mathcal{E}_{\mathcal{R}}) & 1 & 1 \\ -\mathbf{1}^\top & \mathbf{1}^\top & 0 & 0 \\ \mathbf{1}^\top & \mathbf{1}^\top & 0 & 0 \end{bmatrix}, \delta = \begin{bmatrix} \delta \alpha \\ \delta \alpha^* \\ \delta \beta_0 \\ \delta d \end{bmatrix} \in \mathbb{R}^{|\mathcal{E}_{\mathcal{L}}| + |\mathcal{E}_{\mathcal{R}}| + 2}, \mathbf{c} = \begin{bmatrix} \mathbf{y}_{\mathcal{E}_{\mathcal{L}}} \\ \mathbf{y}_{\mathcal{E}_{\mathcal{R}}} \\ 0 \\ 0 \end{bmatrix}$$

and K the gram matrix. Let $\eta = A^{-1} \mathbf{c}$, the parameters are given by:

$$\alpha^{t+1} = \alpha^t + (\lambda - \lambda^t) \eta \alpha, \quad \alpha^{*t+1} = \alpha^{*t} + (\lambda - \lambda^t) \eta \alpha^* \quad (4)$$

$$\beta_0^{t+1} = \beta_0^t + (\lambda - \lambda^t) \eta \beta_0 \quad (5)$$

$$d^{t+1} = d^t + (\lambda - \lambda^t) \eta d \quad (6)$$

Points in $\mathcal{E}_{\mathcal{L}}$ or $\mathcal{E}_{\mathcal{R}}$ and detection of $out(\ell)$, $out(r)$ and $out(c)$ These events occur when the parameters α and α^* hints their boundaries 0 or 1 (see table 2). According to 4, we get:

$$\begin{aligned} \lambda_{out(\ell)}^{t+1} &= \frac{1 - \alpha_i^t}{\eta_{\alpha_i}} + \lambda^t, \quad i \in \mathcal{E}_{\mathcal{L}}; & \lambda_{out(r)}^{t+1} &= \frac{1 - \alpha_i^{*t}}{\eta_{\alpha_i^*}} + \lambda^t, \quad i \in \mathcal{E}_{\mathcal{R}} \\ \lambda_{out(c)}^{t+1} &= \left\{ \frac{-\alpha_i^t}{\eta_{\alpha_i}} + \lambda^t, \quad i \in \mathcal{E}_{\mathcal{L}} \right\} \cup \left\{ \frac{-\alpha_i^{*t}}{\eta_{\alpha_i^*}} + \lambda^t, \quad i \in \mathcal{E}_{\mathcal{R}} \right\} \end{aligned}$$

Points in \mathcal{L} , \mathcal{R} , \mathcal{C} and detection of $in(\ell)$, $in(r)$ and $in(c)$ By substituting equations 4 - 5 in 3 and after some algebras, we get: $f(x) = \frac{\lambda^t}{\lambda} [f^t(x) - h^t(x)] +$

$h^t(x)$ with $h^t(x) = \sum_{i \in \mathcal{E}_{\mathcal{R}}^t} \delta\alpha_i^* k(x_i, x_j) - \sum_{i \in \mathcal{E}_{\mathcal{L}}^t} \delta\alpha_i k(x_i, x_j) + \delta\beta_0$. Using 6, the latest relation and table 2, the values of λ associated to these events are:

$$\lambda_{in(\ell)}^{t+1} = \frac{\lambda^t (f^t(x_i) - h^t(x_i) - \epsilon^t + \eta_d)}{y_i - h^t(x_i) + \eta_d}, \quad \lambda_{in(r)}^{t+1} = \frac{\lambda^t (f^t(x_i) - h^t(x_i) + \epsilon^t - \eta_d)}{y_i - h^t(x_i) - \eta_d}$$

$$\lambda_{in(c)}^{t+1} = \left\{ \lambda_{in(\ell)}^{t+1}, \text{ evaluated for } i \in \mathcal{C} \right\} \cup \left\{ \lambda_{in(r)}^{t+1}, \text{ evaluated for } i \in \mathcal{C} \right\}$$

λ -path algorithm The next value λ^{t+1} is the largest value of λ less than λ^t . The difficult part of the method is to find an initial configuration of λ such as each elbow $\mathcal{E}_{\mathcal{R}}$ and $\mathcal{E}_{\mathcal{L}}$ contains at least one point. In [2], the authors suggest to choose $\lambda = \infty$ and to find the corresponding b . From this initial point, the value of λ is decreased in order to find two points in the elbows. Thus the algorithm is runned until one elbow becomes empty or the the value of λ becomes small.

3.3 Computation of the ν -path

In this case, the parameter λ is fixed and we examine the effect of ν on the regression solution. The proposed approach is closely similar to the derivation of the λ -path. Let the parameter ν^t corresponding to the solution $f^t(x)$. Let $\nu^{t+1} < \nu < \nu^t$ such as the sets obtained at the step t are not modified. As λ is constant, from 3, we obtain: $\lambda(f(x) - f^t(x)) = \sum_{i \in \mathcal{E}_{\mathcal{R}}^t \cup \mathcal{E}_{\mathcal{L}}^t} (\delta\alpha_i^* - \delta\alpha_i) k(x_i, x) + \delta\beta_0$. According to the conditions verified by the points belonging to $\mathcal{E}_{\mathcal{L}}$ and $\mathcal{E}_{\mathcal{R}}$ (respectively $y_i - f(x_i) = -\epsilon$ and $y_i - f(x_i) = \epsilon$) we obtain the set of equations:

$$\sum_{i \in \mathcal{E}_{\mathcal{R}}^t} \delta\alpha_i^* k(x_i, x_j) - \sum_{i \in \mathcal{E}_{\mathcal{L}}^t} \delta\alpha_i k(x_i, x_j) + \delta\beta_0 - \delta d = 0 \quad \forall j \in \mathcal{E}_{\mathcal{L}}^t \quad (7)$$

$$\sum_{i \in \mathcal{E}_{\mathcal{R}}^t} \delta\alpha_i^* k(x_i, x_j) - \sum_{i \in \mathcal{E}_{\mathcal{L}}^t} \delta\alpha_i k(x_i, x_j) + \delta\beta_0 + \delta d = 0 \quad \forall j \in \mathcal{E}_{\mathcal{R}}^t \quad (8)$$

with $\delta d = \lambda(\epsilon - \epsilon^t)$. Also here, the condition $(\alpha^* - \alpha)^T \mathbf{1} = 0$ (which leads to $(\delta\alpha^* - \delta\alpha)^T \mathbf{1} = 0$) holds whereas the inequality $(\alpha^* + \alpha)^T \mathbf{1} \leq \nu$ yields $(\delta\alpha^* + \delta\alpha)^T \mathbf{1} \leq \nu - \nu^t$. Grouping all these equations, we obtain a linear system: $A\delta = (\nu - \nu^t)\mathbf{c}$ with $\mathbf{c} = [\mathbf{0} \ \mathbf{0} \ 0 \ 1]^T$. The values of ν corresponding to the events are computed by applying the same mechanism as previously. The events $in(\ell)$, $in(r)$ and $in(c)$ can be monitored by using the relation $f(x) = f^t(x) + \frac{\nu - \nu^t}{\lambda} h^t(x)$ derived from the updating equations of the parameters with respect to ν . The ν -path is similar to the λ -path. Here the initialization is very easy as we can choose $\nu \approx 0$. In this case, all the points are inside the tube or in the margin and the initial solution is very sparse. As the elbows are initialized, the algorithm proceeds from this point. We have presented a doubly path algorithm. The question arises how to switch from a path to the other. As at each step of a path all the parameters are available, the switching is easily carried.

4 Stopping on the path using Leave One Out

We want to find a stopping criteria along the path. To do so the idea is to compute together with the regularization path a sequence of estimates of the

generalization error to stop when this sequence reaches a minimum. Among all possible estimations of the generalization error, the leave-one-out seems to be the one advocated by practitioners. The major drawback of the leave-one-out estimate is the time required to compute it. Solutions have been proposed to overcome this deficiency. [8] propose to use an approximation easily available called the GCV (Generalized Cross Validation). Others [11] propose to take advantage of efficient implementation of the SVM with warm-start (starting from the current solution as an *a priori* on the next solution) to derive acceptable procedures[12]. Following this idea, we show next how to integrate those estimators in the algorithm and we point out that this method is much cheaper than an external LOO method. The leave-one-out error is defined as the mean error done for the removed points. We also compute a second leave-one-out estimation to have an idea of the variance of the solution:

$$LOO1_{error} = \frac{1}{n} \sum_i 1 - \text{sign}(\hat{y}_i y_i) \quad LOO2_{error} = \frac{1}{n} \sum_i \max(0, \rho - \hat{y}_i y_i)$$

This second formula is very helpful to detect over-fitting. Indeed, outliers will be very penalized. The leave-one-out error rates are estimated at each step. Since no point from \mathcal{R} participate to the solution, they would necessarily have a zero error if once removed from the training set. Hence we only need to compute the LOO errors of each point t of \mathcal{E} and \mathcal{L} . The solution S_{-t} is close to the current solution S given along the path. Thus we obtain S_{-t} from S with Simple- ν -SVM warm start. The cost of the computation of the LOO at each step is $(|\mathcal{E}| + |\mathcal{L}|)$ update steps. If the generalization error is significantly better for some range of parameters λ , we expect to see it through the LOO error rates. Hence we monitor this value to detect when it starts to increase significantly. Then we can stop learning and go back to S_λ which has given the lowest LOO errors. Doing so, we avoid to compute the part of the path for which most of the points are bounded support vectors. Indeed those solution contradict the SVM goal: sparseness.

For the ν -SVR algorithm, the generalization ability is evaluated using the following LOO error (computed by exploiting also a warm start procedure) $LOO = \frac{1}{m} \sum_{i=1, i \neq k}^m |y_i - f_k(x_i)|$ with $f_k(x)$, the solution obtained with the point k out of the training set. This implies $|\mathcal{E}_{\mathcal{L}}| + |\mathcal{E}_{\mathcal{R}}| + |\mathcal{L}| + |\mathcal{R}|$ updates at each step of the path.

4.1 Experimental results

We have conducted experiments on artificial data-sets in order to illustrate how the LOO estimation can be a criteria to stop learning on the path before attaining non sparse solutions. Figure 1 give example of results on the mixture data-set, with an *rbf* kernel. Each LOO estimate provide useful information. The first one (based on the counting of the errors) gives a good approximation of the generalization error. The second one, based on the output value of the SVM represents the variance of the solution and we look for a great low variance solution. From a practical point of view, determining the correct moment to stop requires some heuristic and using a smoothed curved of the LOO error is useful. Our heuristic

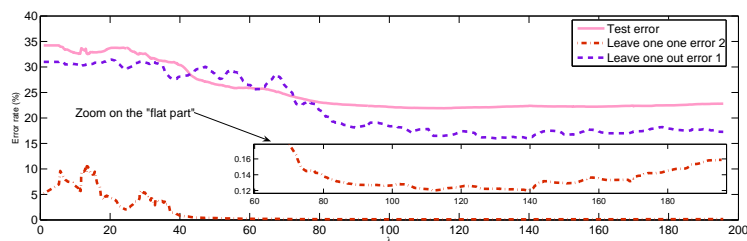


Fig. 1. Illustration on the mixture data-set of the LOO error rate evolution according to λ , reported with the test error

consists in choosing to stop when the smoothed LOO2 error has not been decreasing for a period lasting as long as it takes for λ to grow of 10. Then we choose backward the λ corresponding to the minimum achieved by smoothed LOO1. The test of the ν -SVR algorithm is realized on a toy problem which consists to approximate the nonlinear function $y = \sin(\exp(3 * x))$. A gaussian kernel with bandwidth 0.1 was used. The results obtained for the λ -path are depicted on figure 2. When λ decreases, the LOO error decreases quickly so the algorithm can be stopped earlier. The same remark holds for the width of the tube. For the small values of ν , as the initial solution is sparse, the LOO computation is very fast and stopping earlier the algorithm yields a sparse solution. There is no need to explore the overall path. The illustration of the ν -path for different values of λ is displayed on figure 3. As ν decreases, the tube vanishes and the LOO error decreases. However, it remains to find a suitable strategy to explore the

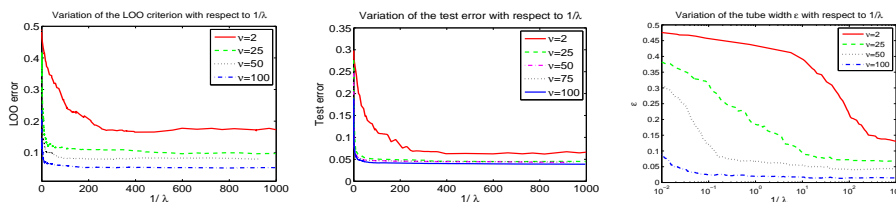


Fig. 2. Illustration of the λ -path for different values of ν . The plots show (from the left to the right) the LOO error, the test error and the width of the tube.

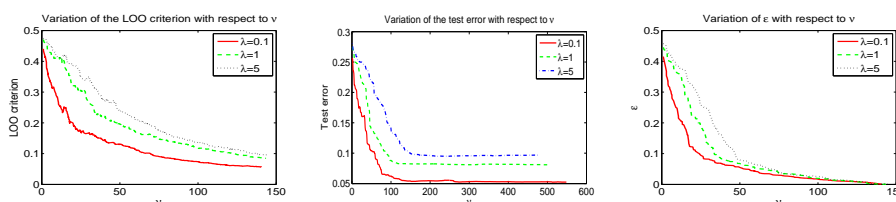


Fig. 3. Illustration of the ν -path for different values of λ .

hyper-parametric space (λ, ν) . It seems that an interesting methodology consists to run the λ -path for small values of ν . The "optimal" value of λ is plugged in the ν -path algorithm to find a final solution. This strategy has to be confirmed by intensive simulations.

5 Conclusion

This paper gathers the ν -SVM and the ν -SVR regularization paths. The motivation for using the ν derivations of the standard methods SVM and SVR is that they provide formulations with more intuitive and bounded hyper-parameters. We give details on the derivations of the regularization paths and we show how to include an estimation of the generalization error within the path so that learning can be stopped when the best solution is attained, without computing useless solutions. Applying this to the SVR is more tricky since we need to search on a surface instead of a line and we are currently developing this part.

Acknowledgments This work was supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views.

References

1. Hastie, T., Rosset, S., Tibshirani, R., Zhu, J.: The entire regularization path for the support vector machine. *Journal of Machine Learning Research* **5** (2004) 1391–1415
2. Gunter, L., Zhu, J.: Computing the solution path for the regularized support vector regression. In: NIPS. (2005)
3. Chen, P.H., Lin, C.J., Schölkopf, B.: A tutorial on ν -support vector machines. *Applied Stochastic Models in Business and Industry* **21** (2005) 111–136
4. Schölkopf, B., Smola, A.: *Learning with Kernels*. MIT Press (2001)
5. Argyriou, A., Hauser, R., Micchelli, C.A., Pontil, M.: A dc-programming algorithm for kernel selection. ICML (2006)
6. Micchelli, C.A., Pontil, M.: Learning the kernel function via regularization. *Journal of Machine Learning Research* **6** (2005) 1099–1125
7. Bach, F., Heckerman, D., Horvitz, E.: On the path to an ideal ROC curve: Considering cost asymmetry in learning classifiers. In Cowell, R.G., Ghahramani, Z., eds.: AISTATS, Society for Artificial Intelligence and Statistics (2005) 9–16
8. Wahba, G. In: *Support Vector Machines, Reproducing Kernel Hilbert spaces and the randomized GACV*. B. Scholkopf and C. Burges and A. Smola edn. MIT Press (1999) 69–88
9. Vishwanathan, S.V.N., Smola, A.J., Murty, M.N.: SimpleSVM. *Proceedings of the Twentieth International Conference on Machine Learning* (2003)
10. Wang, G., Yeung, D.Y., Lochofsky, F.: Two-dimensional solution path for support vector regression. In: *Proc. of the 23rd International Conference on Machine Learning, ICML*. (2006)
11. Lee, J.H., Lin, C.J.: Automatic model selection for support vector machines. Technical report, Dept. of Computer Science and Information Engineering, National Taiwan University (2000)
12. Lee, M., Keerthi, S., Ong, C.J., DeCoste, D.: An efficient method for computing leave-one-out error in support vector machines with gaussian kernels. *Neural Networks, IEEE Transactions* **15** (2004) 750–757