ESANN'2007 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 25-27 April 2007, d-side publi., ISBN 2-930307-07-2.

Computing and stopping the solution paths for ν-SVR

G. Gasso, K. Zapien and S. Canu

LITIS, EA 4051, Rouen, France
{gilles.gasso, karina.zapien, stephane.canu}@insa-rouen.fr

Abstract. The paper describes the computation of the full paths of the well-known ν-SVR. In the classical method, the user provides two parameters: the regularization parameter λ and ν which settles the width of the tube of the ϵ-insensitive cost optimized by SVR. The paper proposes an efficient to way to get all the solutions by varying ν and λ. It analyzes also the stopping of the algorithm using the leave-one-out-criterion.

1 Introduction

The SVR algorithm is a popular method for dealing with regression problems [1]. 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 $\epsilon \geq 0$ of the tube. As the practical choice of ϵ is difficult, the ν-SVR method was proposed and permits to automatically determine the value of ϵ [1].

Many research works have been dedicated to ways of helping the setting of the parameters. Most rely on measures such as cross-validation [2] or on measures derived from bounds [3] to guide the selection. Recently, novel approaches based on the computation of the regularization paths have been widely studied [4, 5] since they provide a smart and fast way to get all the optimal solutions. Indeed, given an initial regression function, the next solution (wrt the hyperparameters) is obtained by solving a linear system with very few equations. In the paper, we propose to explore the two-dimensional space of the hyperparameters for ν-SVR by following two solution paths (λ-path and ν-path).

Having the whole regularization path is not enough. Indeed, the user still needs to retrieve from it the best values for the hyperparameters. This can be done using a generalization error or an estimate of this error. We propose to include the leave-one-out (LOO) estimator inside the solution paths in order to have an idea of the generalization error at each step. The LOO is known to be unbiased but has a huge computational cost. However, by exploiting the warm-start property of the support vector algorithm [6] this computation can be carried up. After the formulation of the ν-SVR problem, we describe the mechanisms of the solution paths. The next section analyzes the criterion to stop the algorithm whereas the last section presents experimental results.

2 The ν-SVR setting

Assuming m training points $\{(x_i, y_i) \in \mathcal{X} \times \mathbb{R}\}$, the ν-SVR algorithm optimizes the ϵ-insensitive cost $L(y, f(x)) = max(0, |y - f(x)| - \epsilon)$ and allows the automatic computation of the ϵ-tube [1]. Its primal formulation is:

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

According to this formulation, the parameter ν varies in the interval $[0, m]$. It gives an upper bound on the number of points allowed to be outside the tube and a lower bound on the number of support vectors. The regression function: $f(x) = \frac{1}{\lambda}\left(\sum_{i=1}^m (\alpha_i^* - \alpha_i)k(x_i, x) + \beta_0\right)$ with $\beta_0 = \lambda b$ is the solution of the previous problem. Here, $k(.,.)$ represents the kernel and the Lagrange multipliers $\alpha_i^{(*)}$ are solutions of the dual problem (with K the Gram matrix):

$$\begin{cases} max_{\boldsymbol{\alpha}^*\in\mathbb{R}^m,\boldsymbol{\alpha}\in\mathbb{R}^m} -\dfrac{1}{2\lambda}(\boldsymbol{\alpha}^* - \boldsymbol{\alpha})^\top K(\boldsymbol{\alpha}^* - \boldsymbol{\alpha}) + (\boldsymbol{\alpha}^* - \boldsymbol{\alpha})^\top \mathbf{y} \qquad s.t. \\ 0 \leq \alpha_i,\ \alpha_i^* \leq 1,\ \forall i \in \{1,\dots,m\} \text{ and } (\boldsymbol{\alpha}^* - \boldsymbol{\alpha})^\top \mathbf{1}_m = 0,\ (\boldsymbol{\alpha}^* + \boldsymbol{\alpha})^\top \mathbf{1}_m \leq \nu \end{cases}$$

3 Formulation of the ν-SVR solution paths

Given fixed values of the regularization parameter λ and of ν, the KKT conditions permit the automatic determination of b and ϵ [1]. 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 ν fixed: the λ-path. Conversely, keeping λ fixed at a specified value, the regression function can be analyzed with respect to ν: the ν-path. Following the original idea in [5], it can be shown that these paths are piecewise linear. The initial regularization path for SVR [5] was extended to the double path (λ-path and ϵ-path) in [7]. In this paper, we give another formulation of the double path as we compute the ν-path instead of the ϵ-path. Moreover this formulation is based on more intuitive hyperparameter ν. Let define the following sets:

$\mathcal{L}$:	$y_i - f(x_i) < -\epsilon$,	$\forall i \in \mathcal{L}$,	$\alpha_i = 1, \alpha_i^* = 0$	bounded points
$\mathcal{R}$:	$y_i - f(x_i) > \epsilon$,	$\forall i \in \mathcal{R}$,	$\alpha_i = 0, \alpha_i^* = 1$	bounded points
$\mathcal{C}$:	$\|y_i - f(x_i)\| < \epsilon$,	$\forall i \in \mathcal{C}$,	$\alpha_i = 0, \alpha_i^* = 0$	useless points
$\mathcal{E_L}$:	$y_i - f(x_i) = -\epsilon$,	$\forall i \in \mathcal{E_L}$,	$0 \leq \alpha_i \leq 1, \alpha_i^* = 0$	useful points
$\mathcal{E_R}$:	$y_i - f(x_i) = \epsilon$,	$\forall i \in \mathcal{E_R}$,	$\alpha_i = 0, 0 \leq \alpha_i^* \leq 1$	useful points

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 $\mathcal{E_L}$ and $\mathcal{E_R}$ lie on the left and right elbows. The elements of $\mathcal{C}$ are the points in the tube.

3.1 Computation of the λ-path

We suppose the value of ν is constant. Let $f^t(x)$, the solution obtained for λ^t. The corresponding sets are $\mathcal{L}^t$, $\mathcal{R}^t$, $\mathcal{C}^t$, $\mathcal{E_L}^t$, $\mathcal{E_R}^t$. The parameters of the model are piecewise-linear in λ as long as the sets are not modified. The key point is to determine the values of λ for which a point moves from a set to another one. The modification of the sets arrives if one of these eight moves occurs: from $\mathcal{L}^t$

to ${\mathcal{E}_\mathcal{L}}^t$ $(in(\ell))$, from $\mathcal{R}^t$ to ${\mathcal{E}_\mathcal{R}}^t$ $(in(r))$, from $\mathcal{C}^t$ to ${\mathcal{E}_\mathcal{L}}^t$ or from $\mathcal{C}^t$ to ${\mathcal{E}_\mathcal{R}}^t$ $(in(c))$; from ${\mathcal{E}_\mathcal{L}}^t$ to $\mathcal{L}^t$ $(out(\ell))$, from ${\mathcal{E}_\mathcal{R}}^t$ to $\mathcal{R}^t$ $(out(r))$ and finally from ${\mathcal{E}_\mathcal{L}}^t$ or ${\mathcal{E}_\mathcal{R}}^t$ to $\mathcal{C}^t$ $(out(c))$. Before be interested by these moves of the training points, let write for $\lambda^{t+1} < \lambda < \lambda^t$, the regression function as $\lambda f(x) = \lambda f(x) - \lambda^t f^t(x) + \lambda^t f^t(x)$. Hence we have:

$$\lambda f(x) = \sum_{i \in {\mathcal{E}_\mathcal{L}}^t \cup {\mathcal{E}_\mathcal{R}}^t} (\delta\alpha_i^* - \delta\alpha_i)\, k(x_i, x) + \delta\beta_0 + \lambda^t f^t(x) \tag{1}$$

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, 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:

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

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

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

Using the constraints $(\boldsymbol{\alpha}^* - \boldsymbol{\alpha})^\top \mathbf{1} = 0$ (which leads to $(\boldsymbol{\delta\alpha}^* - \boldsymbol{\delta\alpha})^\top \mathbf{1} = 0$) and $(\boldsymbol{\alpha}^* + \boldsymbol{\alpha})^\top \mathbf{1} \le \nu$ (hence $(\boldsymbol{\delta\alpha}^* + \boldsymbol{\delta\alpha})^\top \mathbf{1} \le 0$) of the dual problem, we obtain the linear system of $|\mathcal{E}_\mathcal{L}| + |\mathcal{E}_\mathcal{R}| + 2$ equations: $A\boldsymbol{\delta} = (\lambda - \lambda^t)\mathbf{z}$ where:

$$A = \begin{bmatrix} -K(\mathcal{E}_\mathcal{L}, \mathcal{E}_\mathcal{L}) & K(\mathcal{E}_\mathcal{L}, \mathcal{E}_\mathcal{R}) & \mathbf{1} & -\mathbf{1} \\ -K(\mathcal{E}_\mathcal{L}, \mathcal{E}_\mathcal{R})^\top & K(\mathcal{E}_\mathcal{R}, \mathcal{E}_\mathcal{R}) & \mathbf{1} & \mathbf{1} \\ -\mathbf{1}^\top & \mathbf{1}^\top & 0 & 0 \\ \mathbf{1}^\top & \mathbf{1}^\top & 0 & 0 \end{bmatrix}, \boldsymbol{\delta} = \begin{bmatrix} \boldsymbol{\delta\alpha} \\ \boldsymbol{\delta\alpha}^* \\ \delta\beta_0 \\ \delta d \end{bmatrix}, \mathbf{z} = \begin{bmatrix} \mathbf{y}_{{\mathcal{E}_\mathcal{L}}^t} \\ \mathbf{y}_{{\mathcal{E}_\mathcal{R}}^t} \\ 0 \\ 0 \end{bmatrix}$$

Let $\boldsymbol{\eta} = A^{-1}\mathbf{z}$, the parameters are given by these equations linear in λ:

$$\boldsymbol{\alpha}^{t+1} = \boldsymbol{\alpha}^t + (\lambda - \lambda^t)\boldsymbol{\eta}_{\boldsymbol{\alpha}}, \quad \boldsymbol{\alpha}^{*t+1} = \boldsymbol{\alpha}^{*t} + (\lambda - \lambda^t)\boldsymbol{\eta}_{\boldsymbol{\alpha}^*} \tag{2}$$

$$\beta_0^{t+1} = \beta_0^t + (\lambda - \lambda^t)\eta_{\beta_0} \tag{3}$$

$$d^{t+1} = d^t + (\lambda - \lambda^t)\eta_d \tag{4}$$

3.1.1 *Points in $\mathcal{E}_\mathcal{L}$ or $\mathcal{E}_\mathcal{R}$ and detection of the events $out(\ell)$, $out(r)$ and $out(c)$*

These events occur respectively (due to the definition of the sets and the events) when the parameters α hint the boundary 1, the parameters α^* reach their boundary 1 and both parameters attain the value 0. According to 2, we get:

$$\lambda_{out(\ell)}^{t+1} = \frac{1 - \alpha_i^t}{\eta_{\alpha_i}} + \lambda^t, \quad i \in {\mathcal{E}_\mathcal{L}}^t; \qquad \lambda_{out(r)}^{t+1} = \frac{1 - \alpha_i^{*t}}{\eta_{\alpha_i^*}} + \lambda^t, \quad i \in {\mathcal{E}_\mathcal{R}}^t$$

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

3.1.2 Points in $\mathcal{L}$, $\mathcal{R}$, $\mathcal{C}$ and detection of $in(\ell)$, $in(r)$ and $in(c)$

These events occur respectively when $y_i - f(x_i) = -\epsilon^{t+1}$ for previously $i \in \mathcal{L}^t$, $y_i - f(x_i) = \epsilon^{t+1}$ for previously $i \in \mathcal{R}^t$ and $|y_i - f(x_i)| = \epsilon^{t+1}$, for $i \in \mathcal{C}^t$.

By substituting equations 2 - 3 in (1) and after some algebras, we get: $f(x) = \frac{\lambda^t}{\lambda}\left[f^t(x) - h^t(x)\right] + h^t(x)$ with $h^t(x) = \sum_{i \in {\mathcal{E}_\mathcal{R}}^t \cup {\mathcal{E}_\mathcal{L}}^t} (\delta\alpha_i^* - \delta\alpha_i)k(x_i, x_j) + \delta\beta_0$. Using (4) and the latest relation, the values of λ associated to these events are:

$$\lambda_{in(\ell)}^{t+1} = \frac{\lambda^t\left(f^t(x_i) - h^t(x_i) - \epsilon^t + \eta_d\right)}{y_i - h^t(x_i) + \eta_d}, i \in \mathcal{L}^t \quad \lambda_{in(r)}^{t+1} = \frac{\lambda^t\left(f^t(x_i) - h^t(x_i) + \epsilon^t - \eta_d\right)}{y_i - h^t(x_i) - \eta_d}, i \in \mathcal{R}^t$$
$$\lambda_{in(c)}^{t+1} = \left\{\lambda_{in(\ell)}^{t+1}, \quad i \in \mathcal{C}\right\} \cup \left\{\lambda_{in(r)}^{t+1}, \quad i \in \mathcal{C}\right\}$$

3.1.3 λ-path algorithm

The next value λ^{t+1} is the largest positive 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 (see [5, 7]). We initialize our algorithm by solving a QP problem.Thus the algorithm proceeds until one elbow becomes empty or the value of λ becomes small. At each step, solving the linear system has a complexity of $\mathcal{O}(p^3)$ with $p = |\mathcal{E}_\mathcal{L}| + |\mathcal{E}_\mathcal{R}| + 2$. The calculation of $h(x)$ requires $\mathcal{O}(m(|\mathcal{E}_\mathcal{L}| + |\mathcal{E}_\mathcal{R}|))$ operations and the detection of the next event induces a complexity about $\mathcal{O}(m)$.

3.2 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 $\nu^t < \nu < \nu^{t+1}$ such as the sets obtained at the step t are not modified. As λ is constant, from (1), we obtain: $\lambda\left(f(x) - f^t(x)\right) = \sum_{i \in {\mathcal{E}_\mathcal{R}}^t \cup {\mathcal{E}_\mathcal{L}}^t} (\delta\alpha_i^* - \delta\alpha_i)k(x_i, x)) + \delta\beta_0$. 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:

$$\sum_{i \in {\mathcal{E}_\mathcal{R}}^t \cup {\mathcal{E}_\mathcal{L}}^t} (\delta\alpha_i^* - \delta\alpha_i)k(x_i, x_j) + \delta\beta_0 - \delta d = 0 \qquad \forall j \in {\mathcal{E}_\mathcal{L}}^t \tag{5}$$

$$\sum_{i \in {\mathcal{E}_\mathcal{R}}^t \cup {\mathcal{E}_\mathcal{L}}^t} (\delta\alpha_i^* - \delta\alpha_i)k(x_i, x_j) + \delta\beta_0 + \delta d = 0 \qquad \forall j \in {\mathcal{E}_\mathcal{R}}^t \tag{6}$$

with $\delta d = \lambda(\epsilon - \epsilon^t)$. Also here, the condition $(\boldsymbol{\alpha}^* - \boldsymbol{\alpha})^T\mathbf{1} = 0$ holds and leads to $(\boldsymbol{\delta\alpha}^* - \boldsymbol{\delta\alpha})^\top\mathbf{1} = 0$ whereas the inequality $(\boldsymbol{\alpha}^* + \boldsymbol{\alpha})^\top\mathbf{1} \le \nu$ yields the constraint $(\boldsymbol{\delta\alpha}^* + \boldsymbol{\delta\alpha})^\top\mathbf{1} \le \nu - \nu^t$. Grouping all these equations, we obtain a linear system: $A\boldsymbol{\delta} = (\nu - \nu^t)\mathbf{z}$ with $\mathbf{z} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & 0 & 1 \end{bmatrix}^\top$. The values of ν corresponding to the events are computed by applying the previous mechanism. The events $in(\ell)$, $in(r)$ and $in(c)$ can be checked 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 complexity of each step is equal to the complexity of a step of λ-path.

The ν-path is similar to the λ-path. Here the initialization of the algorithm is easy as we can choose ν as one of its extreme value: choosing ν small means that no training point is allowed to be outside the tube (ν is an upper bound on the number of points outside the tube). The initial solution is very sparse and the tube is width. Therefore the LOO error can be computed very quickly. A choice of $\nu \approx m$ leads to a tiny tube. The obtained solution is less sparse and induces a high computational cost of the LOO error.

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 as long as each elbow contains at least one point. Remark that the parameters λ and ν can be increased or decreased along the paths.

4 Evaluation of the path

At each step, we quantify the generalization ability of the obtained regression function. Here, we study the application of two types of generalization error: the cross-validation error $CV = \frac{1}{N_v} \sum_{i=1}^{N_v} |(y_i - f(x_i)|$ and the Leave-One-Out error $LOO = \frac{1}{m} \sum_{i=1, i \neq k}^{m} |y_i - f_k(x_i)|$. $f_k(x)$ is the solution obtained with the point (x_k, y_k) out of the training set. The LOO is known to be unbiased but is very time consuming. To circumvent this drawback, a solution is the use of the warm-start procedure (starting from the current solution as an *a priori* on the next solution) of the support vector machine [6].

5 Simulation results

To evaluate the algorithm, we realize simulations on a toy problem. Let the non-linear function $y = sin(exp(3 * x))$. The variable x is distributed uniformly in the interval [0, 1]. 150 points are used for training and 100 for cross-validation. A gaussian kernel with bandwidth 0.1 was used. The results concerning the application of the λ-path are reported on figure 1. We remark that 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 thus saving much effort. Therefore, the earlier stopping of the algorithm gives sparse solution and there is no need to explore all the path. Notice that, for a lack of place, the figures related to the CV error are not presented by the obtained results are quite similar to those of the LOO.

The illustration of the ν-path for different values of λ is displayed on figure 2. As ν decreases, the tube vanishes and the LOO error decreases. These results are coherent with those of the λ-path.

6 Conclusion

This paper presents an algorithm to compute efficiently the solution paths of the ν-SVR (based on two hyperparameters λ and ν). From an initial solution,

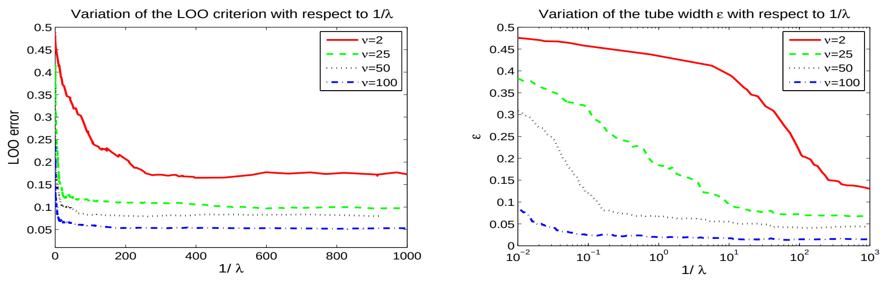

Fig. 1: Illustration of the λ-path for different values of ν. Remark that the x-axis of the second figure is in a logarithmic scale.

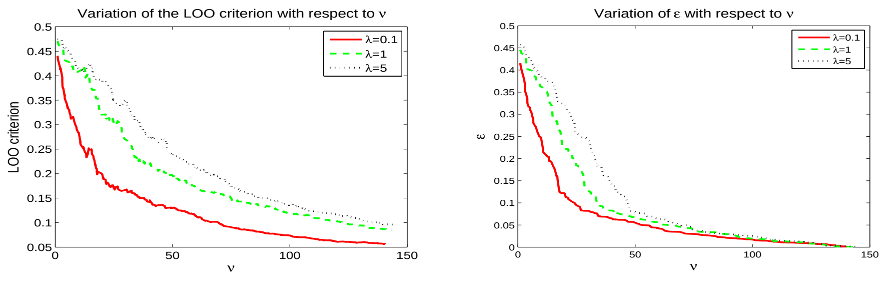

Fig. 2: Illustration of the ν-path for different values of λ.

it provides an easy and automatic way to calculate all the solutions for an hyperparameter fixed and vice versa. The main difficulty is to find in the tricky way these hyperparameters without exploring entirely the two-dimensional path. The development of this part is currently under study. However some simulations show that a good strategy consists to fix ν at a small value and then to run the λ-path. The optimal value obtained from this path is fed to the ν-path.

References

[1] B. Schölkopf and A. Smola. *Leaning with Kernels.* MIT Press, 2001.

[2] K. Kobayashi, D. Kitakoshi, and R. Nakano. Yet faster method to optimize svr hyperparameters based on minimizing cross-validation error. In *Proc. of the IJCNN05*, 2005.

[3] M.-W. Chang and C.-J. Lin. Leave-one-out bounds for support vector regression model selection. *Neural Computation*, 17:1188–1222, 2005.

[4] Trevor Hastie, Saharon Rosset, Robert Tibshirani, and Ji Zhu. The entire regularization path for the support vector machine. *JMLR*, 5:1391–1415, 2004.

[5] Lacey Gunter and Ji Zhu. Computing the solution path for the regularized support vector regression. In *NIPS*, 2005.

[6] M. L., S. Keerthi, C.-J. Ong, and D. DeCoste. An efficient method for computing leave-one-out error in support vector machines with gaussian kernels. *Neural Networks, IEEE Transactions*, 15:750– 757, 2004.

[7] Gang Wang, Dit-Yan Yeung, and Frederick Lochovsky. Two-dimensional solution path for support vector regression. In *Proc. of ICML'06*, 2006.