υ SVMs
+
SMO

Contents

The ν parameter

In our earlier models , we had a 'c' parameter which we used to regulate between function complexity and training error.

What happens if we trade 'c' for something more intuitive?

Here comes ν …

This ν parameter will be used to weight the slacks and to encourage a wider margin.

ν SVC - primal

Using m samples …

SVC

$$\begin{array}{ll} \min_{w,\xi,b} & \frac{1}{2}w^T w + C\sum_i \xi_i \\ \text{s.t.} & y_i(w^T x_i + b) \geq 1 - \xi_i \\ & \xi_i \geq 0 \end{array}$$

ν SVC

$$\begin{array}{ll} \min_{w,\xi,b,\rho} & \frac{1}{2}w^T w - \nu\rho + \frac{1}{m}\sum_i \xi_i \\ \text{s.t.} & y_i(w^T x_i + b) \geq \rho - \xi_i \\ & \xi_i \geq 0, \rho \geq 0 \end{array}$$

We have a new variable ρ to optimize and a new parameter ν

So the two classes are now separated by a margin of width 2ρ

U SVC - dual

Using m samples …

SVM

$$\begin{aligned} \max_\alpha \quad & \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j x_i x_j \\ \text{s.t.} \quad & \sum_i \alpha_i y_i = 0 \\ & 0 \le \alpha_i \le C \end{aligned}$$

U SVC

$$\begin{aligned} \max_\alpha \quad & -\frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j x_i x_j \\ \text{s.t.} \quad & 0 \le \alpha_i \le \frac{1}{m} \\ & \sum_i \alpha_i y_i = 0 \\ & \sum_i \alpha_i \ge \nu \end{aligned}$$

SVR

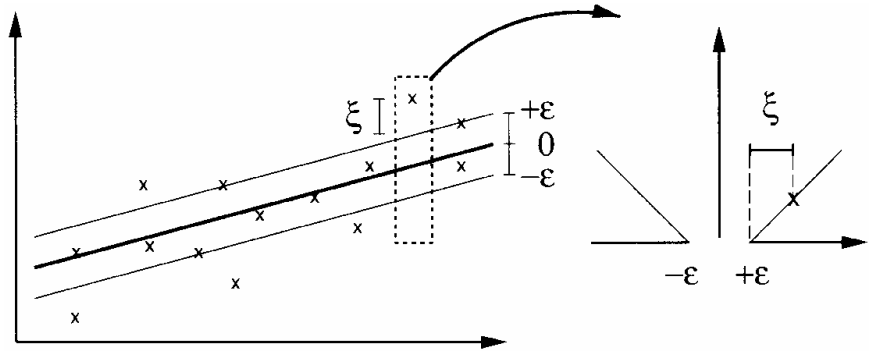

Figure 1: In SV regression, a desired accuracy ε is specified a priori. It is then attempted to fit a tube with radius ε to the data. The trade-off between model complexity and points lying outside the tube (with positive slack variables ξ) is determined by minimizing the expression 1.5.

U SVR - primal

Using m samples …

SVR

$$\begin{array}{ll} \min_{w,b,\xi,\hat{\xi}} & \frac{1}{2} \parallel w \parallel^2 + C \sum_i (\xi_i^2 + \hat{\xi}_i^2) \\ \text{s.t.} & (\langle w, \phi(x_i) \rangle + b) - y_i \leq \epsilon + \xi_i \\ & y_i - (\langle w, \phi(x_i) \rangle + b) \leq \epsilon + \hat{\xi}_i \\ & \xi_i, \hat{\xi}_i \geq 0 \end{array}$$

U SVR

$$\begin{array}{ll} \min_{w,b,\xi,\hat{\xi},\epsilon} & \frac{1}{2} \parallel w \parallel^2 + C[\nu\epsilon + \frac{1}{m} \sum_i^m (\xi_i^2 + \hat{\xi}_i^2)] \\ \text{s.t.} & (\langle w, \phi(x_i) \rangle + b) - y_i \leq \epsilon + \xi_i \\ & y_i - (\langle w, \phi(x_i) \rangle + b) \leq \epsilon + \hat{\xi}_i \\ & \xi_i, \hat{\xi}_i \geq 0 \end{array}$$

U SVR - dual

Using m samples …

SVR

$$\begin{array}{ll} \max & \sum_i (\hat{\alpha}_i - \alpha_i) y_i - \epsilon \sum_i (\hat{\alpha}_i + \alpha_i) \\ & -0.5 \sum_{i,j} (\hat{\alpha}_i - \alpha_i)(\hat{\alpha}_j - \alpha_j) \kappa(x_i, x_j) \\ \text{s.t.} & \sum_i (\hat{\alpha}_i - \alpha_i) = 0 \\ & 0 \leq \alpha_i, \hat{\alpha}_i \leq C \end{array}$$

U SVR

$$\begin{array}{ll} \max & \sum_i (\hat{\alpha}_i - \alpha_i) y_i - 0.5 \sum_{i,j} (\hat{\alpha}_i - \alpha_i)(\hat{\alpha}_j - \alpha_j) \kappa(x_i, x_j) \\ \text{s.t.} & \sum_i (\hat{\alpha}_i - \alpha_i) = 0 \\ & 0 \leq \alpha_i, \hat{\alpha}_i \leq \frac{C}{m} \\ & \sum_i (\hat{\alpha}_i + \alpha_i) \leq C\nu \end{array}$$

Properties of ν

Proposition 1 *Assume $\varepsilon > 0$. The following statements hold:*

(i) ν is an upper bound on the fraction of errors.

(ii) ν is a lower bound on the fraction of SVs.

(iii) Suppose the data (2) were generated iid from a distribution $P(\mathbf{x}, y) = P(\mathbf{x})P(y|\mathbf{x})$ with $P(y|\mathbf{x})$ continuous. With probability 1, asymptotically, ν equals both the fraction of SVs and the fraction of errors.

Intuition behind ν properties

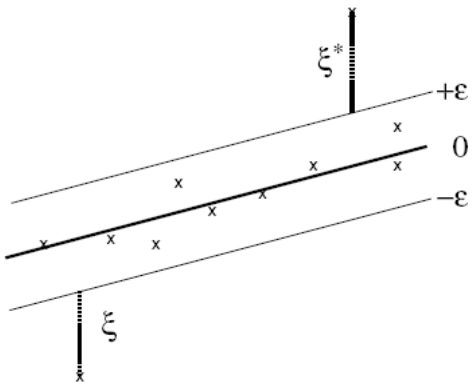

Figure 1: Graphical depiction of the ν-trick. Imagine increasing ε, starting from 0. The first term in $\nu\varepsilon + \frac{1}{\ell}\sum_{i=1}^{\ell}(\xi_i + \xi_i^*)$ (cf. (4)) will increase proportionally to ν, while the second term will decrease proportionally to the fraction of points outside of the tube. Hence, ε will grow as long as the latter fraction is larger than ν. At the optimum, it therefore must be $\leq \nu$ (Proposition 1, (i)). Next, imagine decreasing ε, starting from some large value. Again, the change in the first term is proportional to ν, but this time, the change in the second term is proportional to the fraction of SVs (even points *on* the edge of the tube will contribute). Hence, ε will shrink as long as the fraction of SVs is smaller than ν, eventually leading to Proposition 1, (ii).

Intuition behind ʋ properties

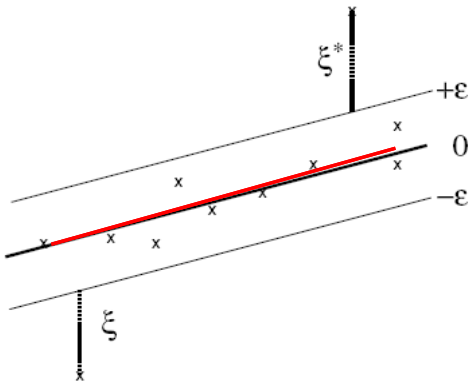

Figure 1: Graphical depiction of the ν-trick. Imagine increasing ε, starting from 0. The first term in $\nu\varepsilon+\frac{1}{\ell}\sum_{i=1}^{\ell}(\xi_i+\xi_i^*)$ (cf. (4)) will increase proportionally to ν, while the second term will decrease proportionally to the fraction of points outside of the tube. Hence, ε will grow as long as the latter fraction is larger than ν. At the optimum, it therefore must be $\leq \nu$ (Proposition 1, (i)). Next, imagine decreasing ε, starting from some large value. Again, the change in the first term is proportional to ν, but this time, the change in the second term is proportional to the fraction of SVs (even points *on* the edge of the tube will contribute). Hence, ε will shrink as long as the fraction of SVs is smaller than ν, eventually leading to Proposition 1, (ii).

Varying the ʋ parameter

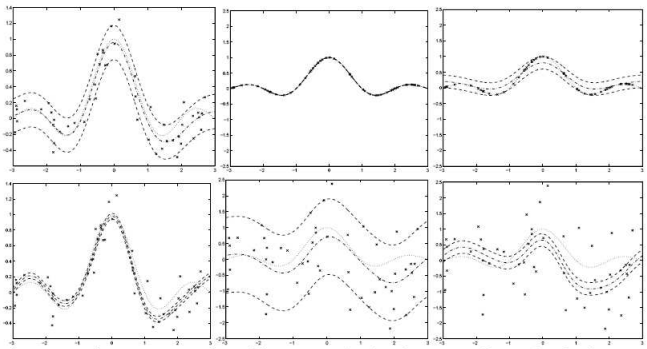

Figure 2: *Left:* ν-SV regression with $\nu = 0.2$ (top) and $\nu = 0.8$ (bottom). The larger ν allows more points to lie outside the tube (see Sec. 2). The algorithm automatically adjusts ε to 0.22 (top) and 0.04 (bottom). Shown are the sinc function (dotted), the regression f and the tube $f \pm \varepsilon$. *Middle:* ν-SV regression on data with noise $\sigma = 0$ (top) and $\sigma = 1$ (bottom). In both cases, $\nu = 0.2$. The tube width automatically adjusts to the noise (top: $\varepsilon = 0$, bottom: $\varepsilon = 1.19$). *Right:* ε-SV regression (Vapnik, 1995) on data with noise $\sigma = 0$ (top) and $\sigma = 1$ (bottom). In both cases, $\varepsilon = 0.2$ — this choice, which has to be specified a priori, is ideal for neither case: in the top figure, the regression estimate is biased; in the bottom figure, ε does not match the external noise (cf. Smola et al., 1998).

Results using Ʊ models

Table 1: Results for the Boston housing benchmark; *top:* ν-SVR, *bottom:* ε-SVR. MSE: Mean squared errors, STD: standard deviations thereof (100 trials), Errors: fraction of training points outside the tube, SVs: fraction of training points which are SVs.

ν	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
automatic ε	2.6	1.7	1.2	0.8	0.6	0.3	0.0	0.0	0.0	0.0
MSE	9.4	8.7	9.3	9.5	10.0	10.6	11.3	11.3	11.3	11.3
STD	6.4	6.8	7.6	7.9	8.4	9.0	9.6	9.5	9.5	9.5
Errors	0.0	0.1	0.2	0.2	0.3	0.4	0.5	0.5	0.5	0.5
SVs	0.3	0.4	0.6	0.7	0.8	0.9	1.0	1.0	1.0	1.0

ε	0	1	2	3	4	5	6	7	8	9	10
MSE	11.3	9.5	8.8	9.7	11.2	13.1	15.6	18.2	22.1	27.0	34.3
STD	9.5	7.7	6.8	6.2	6.3	6.0	6.1	6.2	6.6	7.3	8.4
Errors	0.5	0.2	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
SVs	1.0	0.6	0.4	0.3	0.2	0.1	0.1	0.1	0.1	0.1	0.1

Faster SVMs – related to SVMlight

(some of Kostja's talk)

- Chunking and decomposition
- SMO – Sequential Minimization Optimization

Using Analytic QP and Sparseness to Speed Training of Support Vector Machines

John C. Platt
Microsoft Research
1 Microsoft Way
Redmond, WA 98052
jplatt@microsoft.com

Chunking

Large datasets → large kernels …
storage ☹ computational costs ☹

But solution can be derived using only the support vectors!

Chunking

- Pick a chunk of data and train a SVM on it.
- Keep the SVs and toss the rest
- Identify M points that most violate the KKT conditions
- Update 'working set' with SVs and M 'bad boys'
- Terminate at some stopping criterion

Decomposition

Goal: Optimize the problem by acting on small subsets at a time (does not find ALL active constraints)

Every time you add a new point to the 'working set', a point is removed.
Keep working set fixed at a certain size (SVMlight)

Convergence of chunking and decomposition has not been proved.
But in practice they have been found to work really well.

What happens if we take decomposition to the extreme?
i.e. solve only two points at a time?!!!

SMO – Sequential Minimization Optimization

Break down the problem by solving it two points at a time!
No QP required!!!

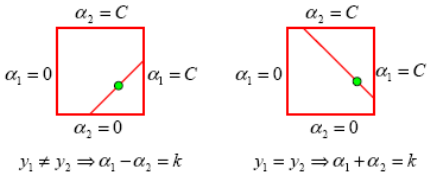

Figure 1: The Lagrange multipliers α_1 and α_2 must fulfill all of the constraints of the full problem. The inequality constraints cause the Lagrange multipliers to lie in the box. The linear equality constraint causes them to lie on a diagonal line.

SMO Performance

Experiment	SMO Time (sec)	SVM^{light} Time (sec)	Chunking Time (sec)	SMO Scaling Exponent	SVM^{light} Scaling Exponent	Chunking Scaling Exponent
AdultLin	**13.7**	217.9	20711.3	**1.8**	2.1	3.1
AdultLinD	**21.9**	n/a	21141.1	**1.0**	n/a	3.0
WebLin	**339.9**	3980.8	17164.7	**1.6**	2.2	2.5
WebLinD	**4589.1**	n/a	17332.8	**1.5**	n/a	2.5
AdultGaussK	442.4	**284.7**	11910.6	2.0	2.0	2.9
AdultGauss	**523.3**	737.5	n/a	2.0	2.0	n/a
AdultGaussKD	**1433.0**	n/a	14740.4	**2.5**	n/a	2.8
AdultGaussD	1810.2	n/a	n/a	2.0	n/a	n/a
WebGaussK	**2477.9**	2949.5	23877.6	**1.6**	2.0	2.0
WebGauss	**2538.0**	6923.5	n/a	**1.6**	1.8	n/a
WebGaussKD	**23365.3**	n/a	50371.9	2.6	n/a	**2.0**
WebGaussD	24758.0	n/a	n/a	1.6	n/a	n/a
MNIST	**19387.9**	38452.3	33109.0	n/a	n/a	n/a

Table 2: Timings of algorithms on various data sets.