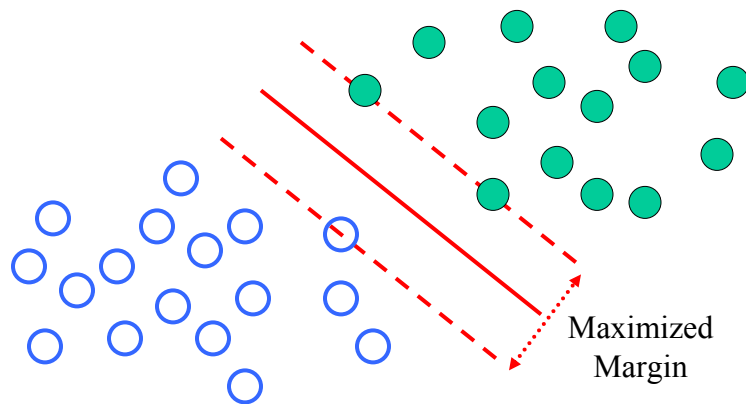


Support Vector Machine Regression

Greg Grudic

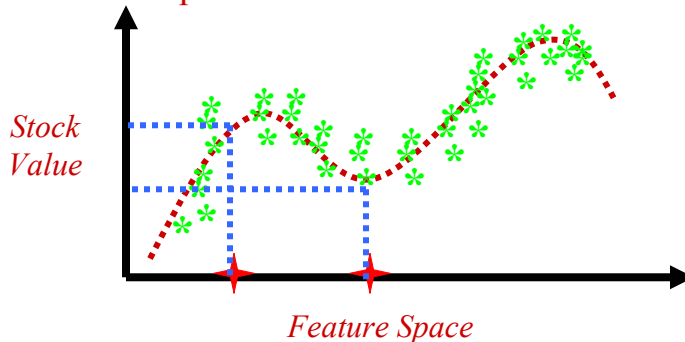
(Notes borrowed from Bernhard Schölkopf)

What are the Support Vectors in Classification?



Learning Regression Models

- Collect Training data
- Build Model: stock value = M (feature space)
- **Make a prediction**



Greg Grudic

Machine Learning

3

SV Regression: ε -Insensitive Loss

[64]

Goal: generalize SV pattern recognition to regression, preserving the following properties:

- formulate the algorithm for the linear case, and then use kernel trick
- sparse representation of the solution in terms of SVs

ε -Insensitive Loss:

$$|y - f(\mathbf{x})|_{\varepsilon} := \max\{0, |y - f(\mathbf{x})| - \varepsilon\}$$

Estimate a linear regression $f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$ by minimizing

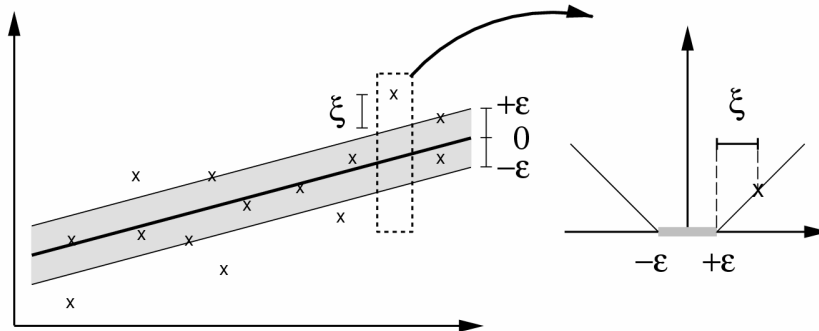
$$\frac{1}{2} \|\mathbf{w}\|^2 + \frac{C}{m} \sum_{i=1}^m |y_i - f(\mathbf{x}_i)|_{\varepsilon}.$$

B. Schölkopf, Canberra, February 2002

Greg Grudic

Machine Learning

4



Formulation as an Optimization Problem

Estimate a linear regression

$$f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$$

with precision ε by minimizing

$$\begin{aligned} \text{minimize} \quad & \tau(\mathbf{w}, \boldsymbol{\xi}, \boldsymbol{\xi}^*) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^m (\xi_i + \xi_i^*) \\ \text{subject to} \quad & (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) - y_i \leq \varepsilon + \xi_i \\ & y_i - (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \leq \varepsilon + \xi_i^* \\ & \xi_i, \xi_i^* \geq 0 \end{aligned}$$

for all $i = 1, \dots, m$.

Dual Problem, In Terms of Kernels

For $C > 0, \varepsilon \geq 0$ chosen a priori,

$$\begin{aligned} \text{maximize} \quad W(\boldsymbol{\alpha}, \boldsymbol{\alpha}^*) &= -\varepsilon \sum_{i=1}^m (\alpha_i^* + \alpha_i) + \sum_{i=1}^m (\alpha_i^* - \alpha_i) y_i \\ &\quad - \frac{1}{2} \sum_{i,j=1}^m (\alpha_i^* - \alpha_i) (\alpha_j^* - \alpha_j) k(\mathbf{x}_i, \mathbf{x}_j) \end{aligned}$$

$$\text{subject to} \quad 0 \leq \alpha_i, \alpha_i^* \leq C, \quad i = 1, \dots, m, \quad \text{and} \quad \sum_{i=1}^m (\alpha_i - \alpha_i^*) = 0.$$

The regression estimate takes the form

$$f(\mathbf{x}) = \sum_{i=1}^m (\alpha_i^* - \alpha_i) k(\mathbf{x}_i, \mathbf{x}) + b,$$

B. Schölkopf, Canberra, February 2002

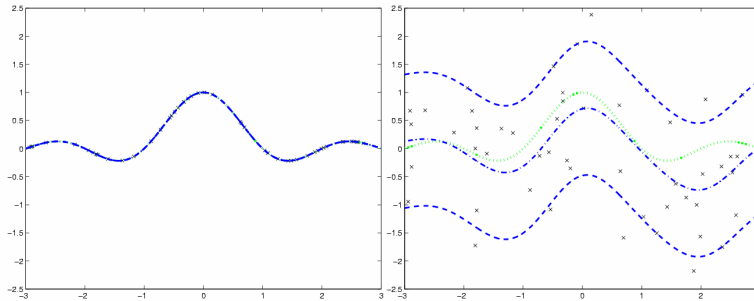
ν -SV Regression

Again, use ν to eliminate another parameter:
Estimate ε from the data s.t. the ν -property holds.

Primal problem: for $0 \leq \nu \leq 1$, minimize

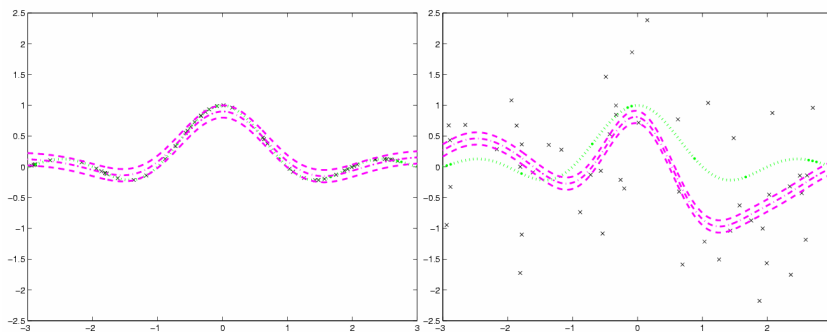
$$\tau(\mathbf{w}, \varepsilon) = \frac{1}{2} \|\mathbf{w}\|^2 + C \left(\nu \varepsilon + 1/m \sum_{i=1}^m |y_i - f(\mathbf{x}_i)| \varepsilon \right)$$

ν -SV-Regression: Automatic Tube Tuning



Identical machine parameters ($\nu = 0.2$), but different amounts of noise in the data.

ϵ -SV-Regression, Run on the Same Data



Identical machine parameters ($\epsilon = 0.2$), but different amounts of noise in the data.

Boston Housing Benchmark

- 506 examples, 13-dimensional.

Results (MSE):

- Bagging regression trees: 11.7 [8]
- ε -SV regression: 7.6 [59]

Mean Squared Error (MSE)

Results on test data:

$TestData : (y_1, \mathbf{x}_1), \dots, (y_K, \mathbf{x}_K)$

$$MSE = \frac{1}{K} \sum_{i=1}^K (y_i - f(\mathbf{x}_i))^2$$

- 100 runs, with 25 randomly selected test points.
- training set is split into actual training set and validation set (80 points) for selecting ε , C , and kernel parameters
- <ftp://ftp.ics.uci.com/pub/machine-learning-databases/housing>

B. Schölkopf, Canberra, February 2002

Comparison: ν vs. ε

ν -SVR	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	
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 -SVR	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
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

- RBF kernel, C and σ chosen as in [56]

B. Schölkopf, Canberra, February 2002