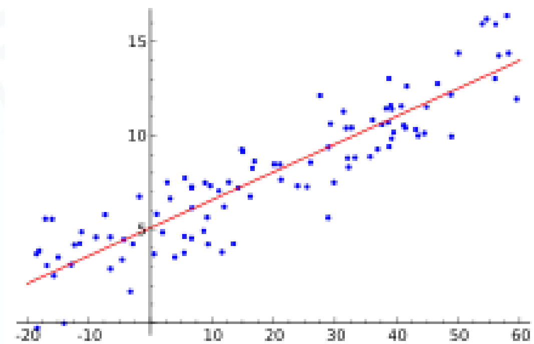
Regression

Dr. Fayyaz ul Amir Afsar Minhas
PIEAS Biomedical Informatics Research Lab
Department of Computer and Information Sciences
Pakistan Institute of Engineering & Applied Sciences
PO Nilore, Islamabad, Pakistan
http://faculty.pieas.edu.pk/fayyaz/

Regression

- Estimate the relationship among variables
  - Dependent variables
  - Independent variables
- Used in prediction and forecasting
- Mathematical formulation
  - **Model:** $\boldsymbol{Y} = f(\boldsymbol{X}; \boldsymbol{w}) + \boldsymbol{\epsilon}$
  - Linear
    - $f(\boldsymbol{x_i}) = b + w_1 x_i^{(1)} + w_2 x_i^{(2)} + \cdots + w_d x_i^{(d)}$
  - Objective is to estimate the parameters $\boldsymbol{w}$



x	y
1	3.5
2	4.75
3	7.05
4	9.5
…	…

Linear Regression

- $f(\boldsymbol{x_i}) = b + w_1 x_i^{(1)} + w_2 x_i^{(2)} + \cdots + w_d x_i^{(d)}$
- This implies
  - $f(\boldsymbol{x_i}) = [\boldsymbol{x_i^T} \quad 1]\boldsymbol{w}'$
  - Note that we do not have an explicit bias term anymore
- For N points with
  - $y_1 = f(\boldsymbol{x_1}) = [\boldsymbol{x_1^T} \quad 1]\boldsymbol{w}'$
  - $y_2 = f(\boldsymbol{x_2}) = [\boldsymbol{x_2^T} \quad 1]\boldsymbol{w}'$
  - …
  - $y_N = f(\boldsymbol{x_N}) = [\boldsymbol{x_N^T} \quad 1b]$
- In Matrix form
  - $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w}'$

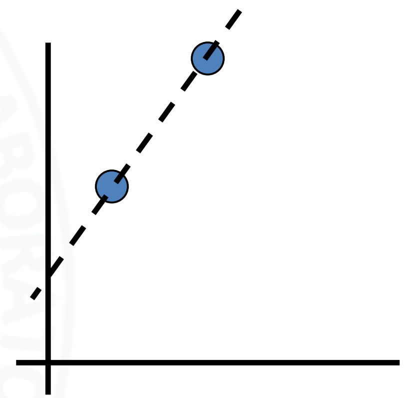
$$\boldsymbol{w}'_{((d+1)\times 1)} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \\ b \end{bmatrix}$$

$$\boldsymbol{X}_{(N\times(d+1))} = \begin{bmatrix} x_1^{(1)} & x_1^{(2)} & \cdots & x_1^{(d)} & 1 \\ x_2^{(1)} & x_2^{(2)} & \cdots & x_2^{(d)} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_N^{(1)} & x_N^{(2)} & \cdots & x_N^{(d)} & 1 \end{bmatrix}$$

$$\boldsymbol{y}_{(N\times 1)} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

Linear Regression

- It can also be written as:
  - $\boldsymbol{X}\boldsymbol{w}' = \boldsymbol{y}$
- If $\boldsymbol{X}$ is square (N=d+1) then the solution to the above equation is
  - $\boldsymbol{w}' = \boldsymbol{X}^{-1}\boldsymbol{y}$
- Example
  - $\boldsymbol{X} = \begin{bmatrix} \mathbf{1} & \mathbf{1} \\ \mathbf{2} & \mathbf{1} \end{bmatrix}, \boldsymbol{y} = \begin{bmatrix} \mathbf{3.5} \\ \mathbf{4.75} \end{bmatrix}$
  - Thus: $\boldsymbol{w}' = \begin{bmatrix} \mathbf{1.25} \\ \mathbf{2.25} \end{bmatrix}$



x	y
1	3.5
2	4.75

Linear Regression: Least Squares solution

- However, having a square $\boldsymbol{X}$ is very restrictive
- If X is not square, we can use a pseudo-inverse

$$\boldsymbol{X}\boldsymbol{w}' = \boldsymbol{y}$$

$$\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X}\boldsymbol{w}' = \boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{y}$$

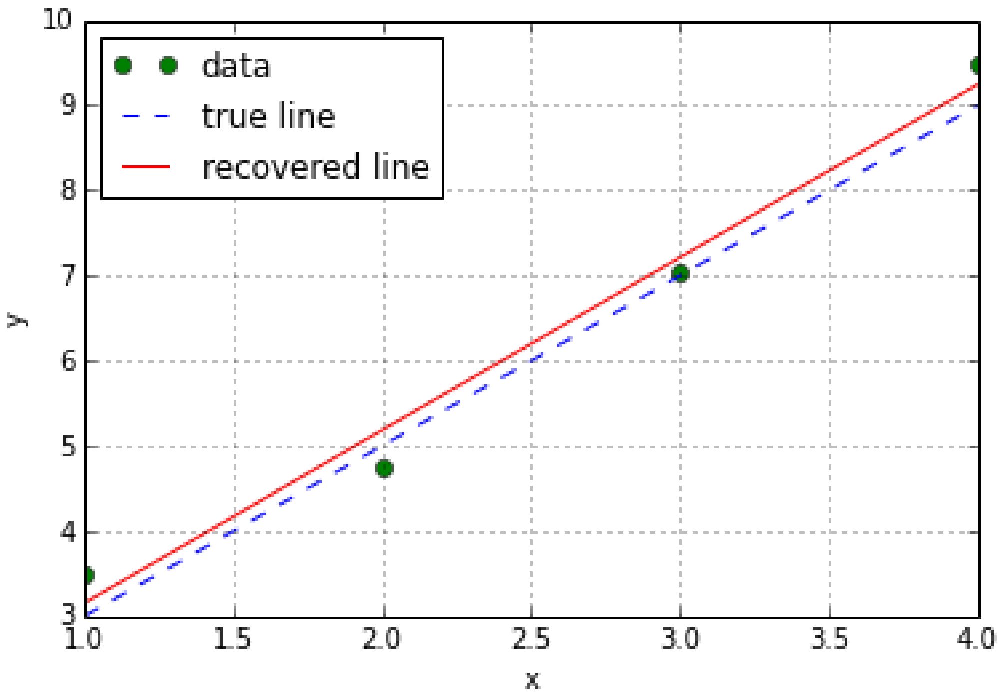
$$\boldsymbol{w}' = \left(\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X}\right)^{-\boldsymbol{1}}\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{y}$$

$$\boldsymbol{w}' = \boldsymbol{X}^{+}\boldsymbol{y}$$

- $\boldsymbol{X}^{+} = \left(\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X}\right)^{-\boldsymbol{1}}\boldsymbol{X}^{\boldsymbol{T}}$ is the pseudo-inverse
- Used to solve over-determined systems
  - More constraints than parameters

Linear Regression: Least Squares solution

- Compute the pseudo-inverse and plot
- Note that the line isn't passing through all points



x	y
1	3.5
2	4.75
3	7.05
4	9.5
…	…

Properties of the least-squares solution

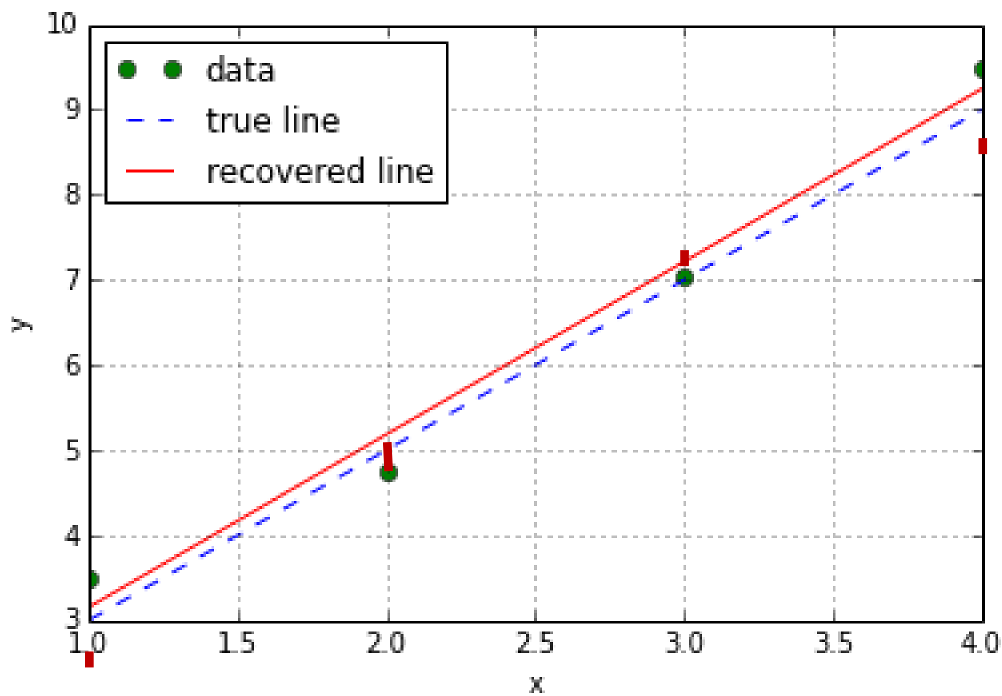
- The previous line represents a least squares (LS) solution to the regression problem
- It minimizes the mean square error of the prediction
- The mean square error resulting from a specific weight vector can be written as (ignoring bias):
  - $\boldsymbol{E}(\boldsymbol{w}) = \frac{1}{N}\sum_{\boldsymbol{i-1}}^{\boldsymbol{N}}(\boldsymbol{f}(\boldsymbol{x_i}) - y_i)^{\boldsymbol{2}} = \frac{1}{N}\sum_{\boldsymbol{i-1}}^{\boldsymbol{N}}\left(\boldsymbol{x_i^T w} - y_i\right)^{\boldsymbol{2}}$
  - Thus the LS learning problem can written as:

$$min_{\boldsymbol{w}}\boldsymbol{E}(\boldsymbol{w}) = \frac{1}{N}\sum_{\boldsymbol{i-1}}^{\boldsymbol{N}}\left(\boldsymbol{x_i^T w} - y_i\right)^{\boldsymbol{2}}$$

  - In Matrix form, $\boldsymbol{E}(\boldsymbol{w}) = \frac{1}{N}(\boldsymbol{Xw} - \boldsymbol{y})^{\boldsymbol{T}}(\boldsymbol{Xw} - \boldsymbol{y})$
  - If we differentiate $\boldsymbol{E}(\boldsymbol{w})$ with respect to $\boldsymbol{w}$ and substituting it to zero we get: $\boldsymbol{Xw} = \boldsymbol{y}$
    - We can now solve for $\boldsymbol{w}$ to get a closed form least square solution

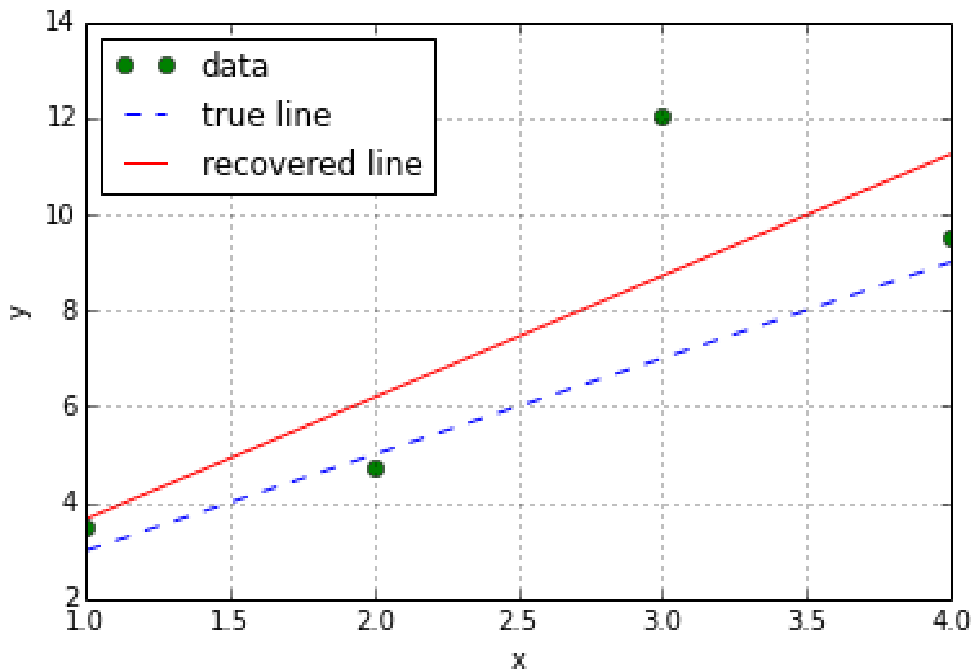
Least squares visualized

- The thick red lines indicate the error corresponding to each data point
- Least square solution minimizes the sum of the square lengths of all red lines



Problems with least squares solution

- Due to squaring of the error of each data point the least square solution is very sensitive to outliers
- It gives only a linear solution



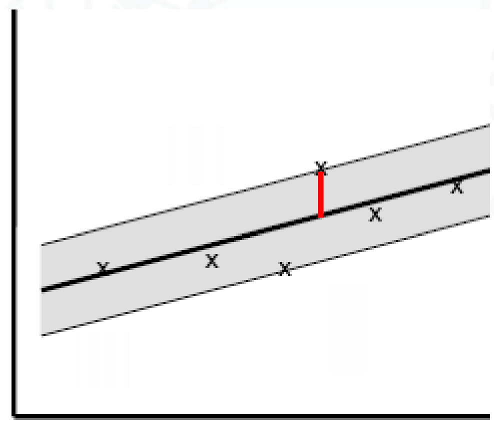
Support Vector Regression

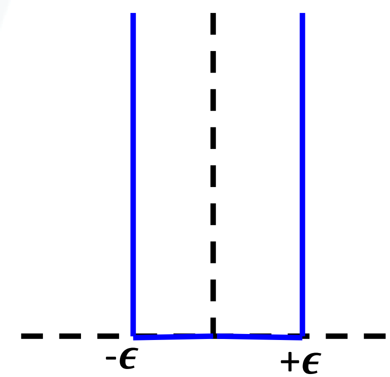
- Hard Form
  - All errors must be within a user specified threshold
  - Minimize the norm of the weight vector

$$\boldsymbol{min}_{w,b} ||\boldsymbol{w}||^2$$

Such that for all $\boldsymbol{i} = 1 \dots N$:

$$\left|\left(\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x}_{\boldsymbol{i}} + b\right) - y_{\boldsymbol{i}}\ \right| \leq \boldsymbol{\epsilon}$$





Penalty/loss function

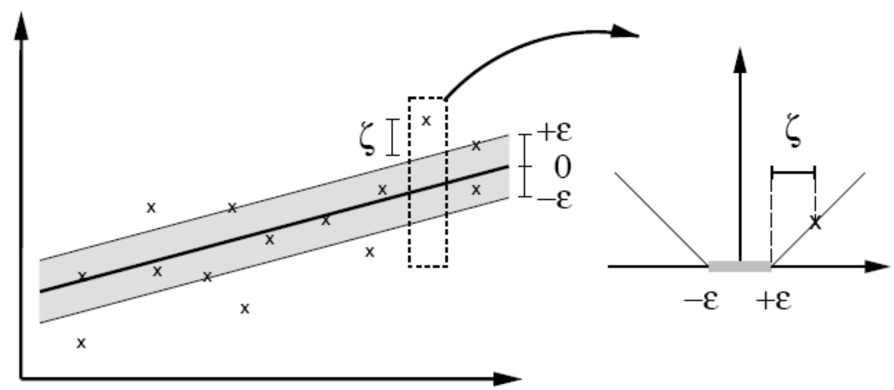
Support Vector Regression

- Soft Form
  - Errors must be within a user specified threshold or penalize them linearly (instead of quadratically as in the LS solution)
  - Minimize the norm of the weight vector
- More robust!

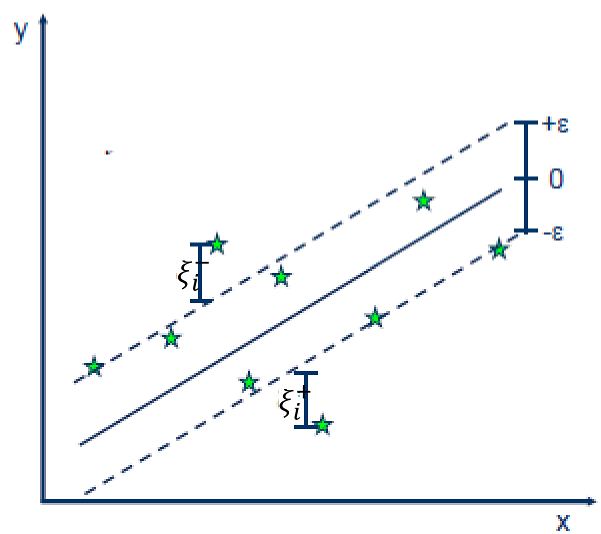
$$\boldsymbol{min}_{\boldsymbol{w},b,\boldsymbol{\xi}\geqslant 0}\|\boldsymbol{w}\|^{2}+C\sum_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{N}}\xi_{\boldsymbol{i}}$$

Such that for all $\boldsymbol{i}$:

$$\left|\left(\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x}_{\boldsymbol{i}}+b\right)-y_{\boldsymbol{i}}\ \right|\leq\boldsymbol{\epsilon}+\xi_{\boldsymbol{i}}$$



SVR: Primal Form



$$\boldsymbol{min}_{w,b,\xi\geqslant 0}\|\boldsymbol{w}\|^2 + C\sum_{\boldsymbol{i=1}}^{\boldsymbol{N}}(\xi_i^+ + \xi_i^-)$$

Such that for all $\boldsymbol{i}$:

$$y_{\boldsymbol{i}} - \left(\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x}_{\boldsymbol{i}} + b\right) \leq \boldsymbol{\epsilon} + \xi_i^+$$

$$\left(\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x}_{\boldsymbol{i}} + b\right) - y_{\boldsymbol{i}} \leq \boldsymbol{\epsilon} + \xi_i^-$$

$$\xi_i^+, \xi_i^- \geq 0$$

So what is min $||\boldsymbol{w}||^2$ doing here

- Geometric interpretation of margin maximization in regression

SVR: Dual Form

$$max_{\boldsymbol{\alpha}} -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i^+ - \alpha_i^-)(\alpha_j^+ - \alpha_j^-)\boldsymbol{x_i^T x_j} - \epsilon\sum_{i=1}^{N}(\alpha_i^+ + \alpha_i^-) - \sum_{i=1}^{N} y_i(\alpha_i^+ - \alpha_i^-)$$

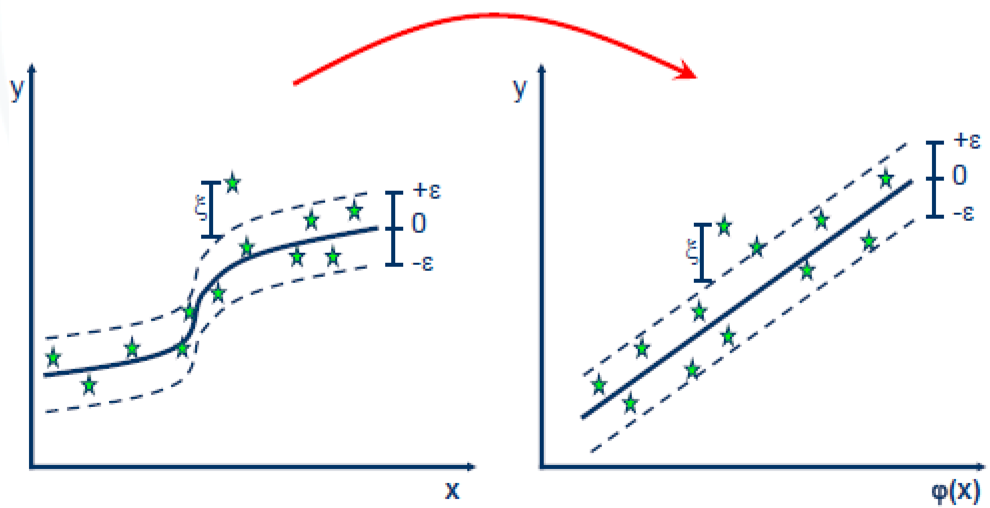
Subject to:

$$0 \le \alpha_i^+ \le C, 0 \le \alpha_i^- \le C$$
$$\sum_{i=1}^{N}(\alpha_i^+ - \alpha_i^-) = 0$$

- We can now apply the kernel trick and use it for non-linear regression

Kernels in SVR

- In the kernel space the SVR is fitting a line which corresponds to an arbitrary curve in the original feature space



SVR in action

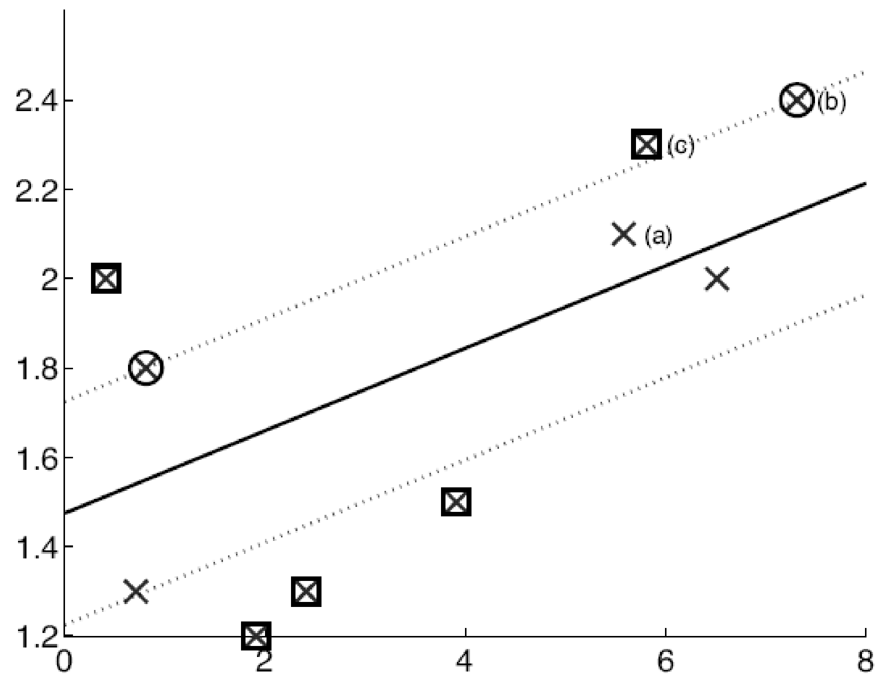


Figure 13.7 The fitted regression line to data points shown as crosses and the ϵ-tube are shown ($C = 10, \epsilon = 0.25$). There are three cases: In (a), the instance is in the tube; in (b), the instance is on the boundary of the tube (circled instances); in (c), it is outside the tube with a positive slack, that is, $\xi_+^t > 0$ (squared instances). (b) and (c) are support vectors. In terms of the dual variable, in (a), $\alpha_+^t = 0, \alpha_-^t = 0$, in (b), $\alpha_+^t < C$, and in (c), $\alpha_+^t = C$.

SVR with quadratic kernel

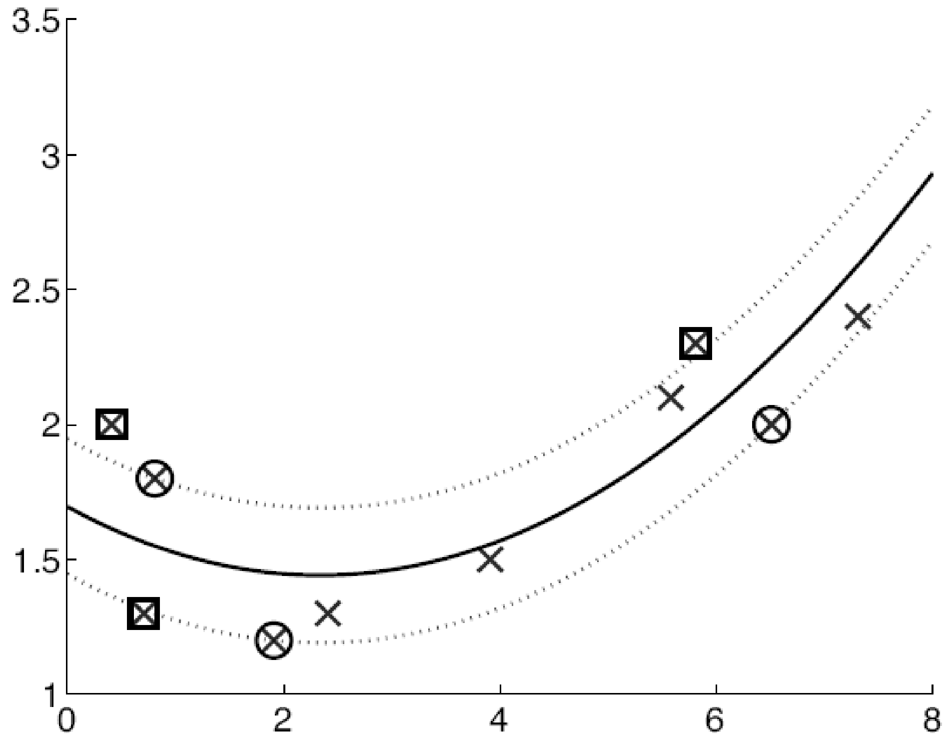


Figure 13.8 The fitted regression line and the ϵ-tube using a quadratic kernel are shown ($C = 10, \epsilon = 0.25$). Circled instances are the support vectors on the margins, squared instances are support vectors which are outliers.

SVR with RBF kernel

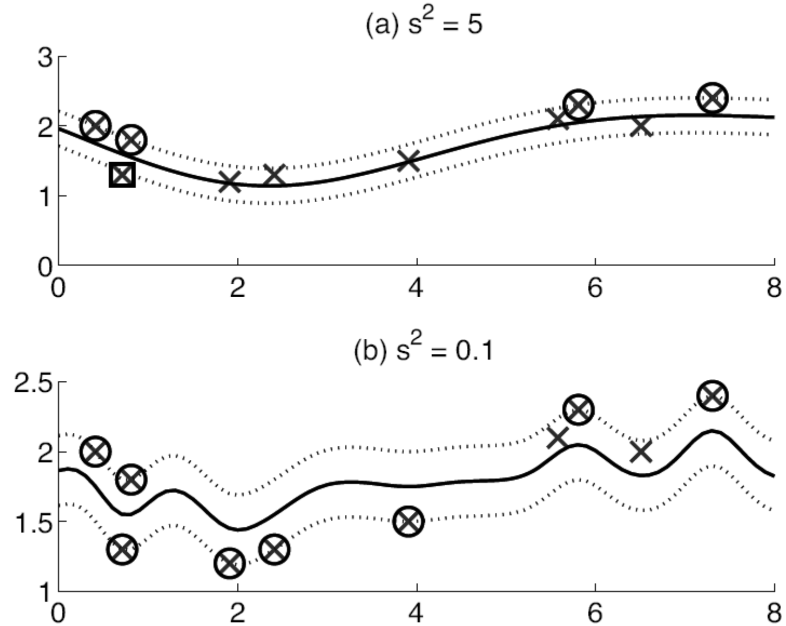
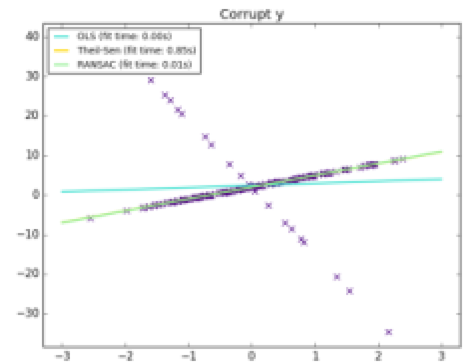
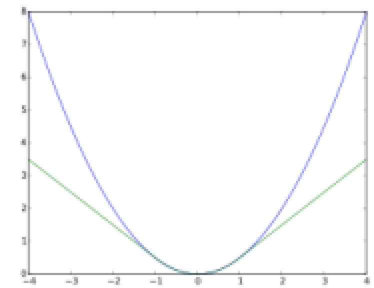


Figure 13.9 The fitted regression line and the ϵ-tube using a Gaussian kernel with two different spreads are shown ($C = 10, \epsilon = 0.25$). Circled instances are the support vectors on the margins, and squared instances are support vectors that are outliers.

Regression with linear models

- http://scikit-learn.org/stable/modules/linear_model.html

Model	Description
Ordinary Least Squares	$\min_{w} \|\|Xw - y\|\|_2{}^2$
Ridge Regression	$\min_{w} \|\|Xw - y\|\|_2{}^2 + \alpha \|\|w\|\|_2{}^2$
Lasso	$\min_{w} \frac{1}{2n_{samples}} \|\|Xw - y\|\|_2^2 + \alpha \|\|w\|\|_1$
Multitask Lasso	$\min_{w} \frac{1}{2n_{samples}} \|\|XW - Y\|\|_{Fro}^2 + \alpha \|\|W\|\|_{21}$ $\|\|A\|\|_{Fro} = \sqrt{\sum_{ij} a_{ij}^2}$
Elastic Net	$\min_{w} \frac{1}{2n_{samples}} \|\|Xw - y\|\|_2^2 + \alpha\rho \|\|w\|\|_1 + \frac{\alpha(1-\rho)}{2} \|\|w\|\|_2^2$
Multi-task Elastic Net	$\min_{W} \frac{1}{2n_{samples}} \|\|XW - Y\|\|_{Fro}^2 + \alpha\rho \|\|W\|\|_{21} + \frac{\alpha(1-\rho)}{2} \|\|W\|\|_{Fro}^2$
LARS and LARS Lasso	Can get the fulll regularization path
Orthogonal Matching Pursuit	$\arg\min \|\|y - X\gamma\|\|_2^2 \text{ subject to } \|\|\gamma\|\|_0 \leq n_{nonzero\_coefs}$ OR $\arg\min \|\|\gamma\|\|_0 \text{ subject to } \|\|y - X\gamma\|\|_2^2 \leq \text{tol}$
Logistic Regression (For classification)	$\min_{w,c} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \log(\exp(-y_i(X_i^T w + c)) + 1).$

Perceptron	
Passive Aggressive Algorithm	Regularized Perceptron
RANSAC	For Robust regression 
Thiel Sen	For robust regression
Huber Regressor	$$\min_{w,\sigma} \sum_{i=1}^{n} \left(\sigma + H_m\left(\frac{X_i w - y_i}{\sigma}\right)\sigma\right) + \alpha \lVert w \rVert_2{}^2$$ $$H_m(z) = \begin{cases} z^2, & \text{if } \lvert z \rvert < \epsilon, \\ 2\epsilon \lvert z \rvert - \epsilon^2, & \text{otherwise} \end{cases}$$, 

Least Square SVM

$$\min_{w,\beta_0,e} \frac{1}{2}w^T w + \gamma \frac{1}{2}\sum_{i=1}^{N} e_i^2 \text{ such that } y_i = y_{\mathbf{w}}(x_i) + e_i, i = 1, \cdots, N$$

$$L_\alpha(w, \beta_0, e) = \tfrac{1}{2}w^T w + \gamma \tfrac{1}{2}\textstyle\sum_{i=1}^{N} e_i^2 - \sum_{i=1}^{N} \alpha_i \{w^T \phi(x_i) + \beta_0 + e_i - y_i\}$$

$$w = \textstyle\sum_{i=1}^{N} \alpha_i \phi(x_i) \qquad \sum_{i=1}^{N} \alpha_i = 0, \; \alpha_i = \gamma e_i, i = 1, \cdots, N$$

$$w^T \phi(x_i) + \beta_0 + e_i - y_i = 0, i = 1, \cdots, N$$

$$\begin{bmatrix} 0 & 1_N^T \\ 1_N & Z^T Z + I/y \end{bmatrix} \begin{bmatrix} \beta_0 \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix} \text{ where } Z^T = \begin{bmatrix} \phi(x_1)^T y_1 \\ \phi(x_2)^T y_2 \\ \cdots \\ \phi(x_N)^T y_N \end{bmatrix}$$

LS SVM

- Direct Solution of the form Ax=b
- Can be used for regression or for classification

Multi-output regression

- When we need to predict more than one variable as the output
  - The output variables may not be independent of each other

$$h : \Omega_{X_1} \times \ldots \times \Omega_{X_m} \longrightarrow \Omega_{Y_1} \times \ldots \times \Omega_{Y_d}$$
$$\mathbf{x} = (x_1, \ldots, x_m) \longmapsto \mathbf{y} = (y_1, \ldots, y_d),$$

  - Solutions?
    - Apply a regression method for each variable
      - Shortcoming?
        - » Correlations are ignored

- Borchani, Hanen, Gherardo Varando, Concha Bielza, and Pedro Larrañaga. 2015. “A Survey on Multi-Output Regression.” *Wiley Int. Rev. Data Min. and Knowl. Disc.* 5 (5): 216–33. doi:10.1002/widm.1157.

Required

- Reading:
  - Section 13.10 in Alpaydin 2010
- Problems
  - Discuss least squares regression and SVR in terms of structural risk minimization (SRM)
  - Understand how we achieved the dual form
  - Understand how we got from $\left|(\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x}_{\boldsymbol{i}} + b) - y_{\boldsymbol{i}}\ \right| \leq \boldsymbol{\epsilon} + \xi_{\boldsymbol{i}}$

    to the following two constraints

$$y_{\boldsymbol{i}}\ - (\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x}_{\boldsymbol{i}} + b) \leq \boldsymbol{\epsilon} + \xi_i^+$$
$$(\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x}_{\boldsymbol{i}} + b) - y_{\boldsymbol{i}} \leq \boldsymbol{\epsilon} + \xi_i^-$$

We want to make a machine that will be proud of us.

- Danny Hillis