6 - Regression problems

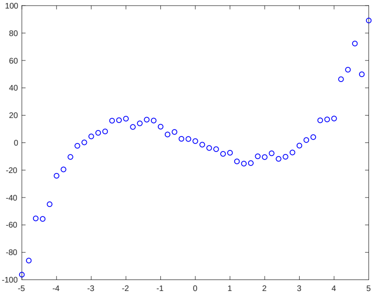
G. Mastroeni and M. Passacantando

Department of Computer Science, University of Pisa

Optimization Methods and Game Theory
Master of Science in Artificial Intelligence and Data Engineering
University of Pisa – A.Y. 2024/25

Polynomial regression

We have ℓ experimental data $y_1, y_2, \dots, y_\ell \in \mathbb{R}$ corresponding to the observations related to the points $x_1, x_2, \dots, x_\ell \in \mathbb{R}$.



We want to find the best approximation of experimental data with a polynomial p of degree $n-1$, with $n \leq \ell$.
Polynomial p has coefficients $z_1, \dots, z_n$:

$$p(x) = z_1 + z_2\,x + z_3\,x^2 + \cdots + z_n\,x^{n-1}$$

The residual is the vector $r \in \mathbb{R}^\ell$ such that $r_i = p(x_i) - y_i$, with $i = 1, \dots, \ell$.

We want to find coefficients $z := (z_1, z_2, .., z_n)$ of polynomial p such that $\|r\|$ is minimum, which amounts to solve the following unconstrained problem

$$\begin{cases} \min \ \|A\,z - y\| \\ z \in \mathbb{R}^n \end{cases}$$

where

$$A = \begin{pmatrix} 1 & x_1 & x_1^2 & \dots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \dots & x_2^{n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_\ell & x_\ell^2 & \dots & x_\ell^{n-1} \end{pmatrix} \in \mathbb{R}^{\ell \times n} \qquad y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_\ell \end{pmatrix}$$

For any norm, the objective function $f(z) = \|Az - y\|$ is convex.

We will consider three special norms: $\|\cdot\|_2$, $\|\cdot\|_1$ and $\|\cdot\|_\infty$.

Polynomial regression with $\|\cdot\|_2$

Euclidean norm $\|\cdot\|_2$ (least squares approximation)
$\rightarrow$ unconstrained quadratic programming problem:

$$\begin{cases} \min \ \frac{1}{2}\|Az - y\|_2^2 = \frac{1}{2}(Az-y)^{\mathsf{T}}(Az-y) = \frac{1}{2}z^{\mathsf{T}}A^{\mathsf{T}}Az - z^{\mathsf{T}}A^{\mathsf{T}}y + \frac{1}{2}y^{\mathsf{T}}y \\ z \in \mathbb{R}^n \end{cases}$$

It can be proved that $\operatorname{rank}(A) = n$, thus $A^{\mathsf{T}}A$ is positive definite.

Hence, the unique optimal solution is the stationary point of the objective function, i.e., the solution of the system of linear equations:

$$A^{\mathsf{T}}Az = A^{\mathsf{T}}y \tag{1}$$

Polynomial regression with $\|\cdot\|_1$

Norm $\|\cdot\|_1$ $\rightarrow$ linear programming problem:

$$\begin{cases} \min \ \|Az - y\|_1 = \min \ \sum_{i=1}^{\ell} |A_i z - y_i| \\ z \in \mathbb{R}^n \end{cases}$$

is equivalent to

$$\begin{cases} \min\limits_{z,u} \ \sum_{i=1}^{\ell} u_i \\ u_i = |A_i z - y_i| \\ \quad = \max\{A_i z - y_i, y_i - A_i z\} \end{cases} \rightarrow \begin{cases} \min\limits_{z,u} \ \sum_{i=1}^{\ell} u_i \\ u_i \geq \max\{A_i z - y_i, y_i - A_i z\} \end{cases}$$

$$\rightarrow \begin{cases} \min\limits_{z,u} \ \sum_{i=1}^{\ell} u_i \\ u_i \geq A_i z - y_i & \forall \ i = 1, \dots, \ell \\ u_i \geq y_i - A_i z & \forall \ i = 1, \dots, \ell \end{cases} \tag{2}$$

In matrix form (2) can be written as

$$\begin{cases} \min\limits_{z,u} \ e_\ell^T u \\ Az - u \leq y \\ -Az - u \leq -y \end{cases}$$

where $e^T = (1, ..., 1) \in \mathbb{R}^\ell$.
Set

$$D = \begin{pmatrix} A & -I_\ell \\ -A & -I_\ell \end{pmatrix} \quad d = \begin{pmatrix} y \\ -y \end{pmatrix}$$

where I_ℓ is the identity matrix of order ℓ, then we obtain

$$\begin{cases} \min\limits_{z,u} \ (0_n^T, e_\ell^T) \begin{pmatrix} z \\ u \end{pmatrix} \\ D \begin{pmatrix} z \\ u \end{pmatrix} \leq d \end{cases}$$

Norm $\|\cdot\|_\infty \to$ linear programming problem:

$$\begin{cases} \min \ \|Az - y\|_\infty = \min \ \max\limits_{i=1,\dots,\ell} |A_i z - y_i| \\ z \in \mathbb{R}^n \end{cases}$$

is equivalent to

$$\begin{cases} \min\limits_{z,u} \ u \\ u = \max\limits_{i=1,\dots,\ell} |A_i z - y_i| \end{cases} \to \begin{cases} \min\limits_{z,u} \ u \\ u \geq A_i z - y_i & \forall \ i = 1, \dots, \ell \\ u \geq y_i - A_i z & \forall \ i = 1, \dots, \ell \end{cases} \tag{3}$$

Set

$$D = \begin{pmatrix} A & -e_\ell \\ -A & -e_\ell \end{pmatrix} \quad d = \begin{pmatrix} y \\ -y \end{pmatrix}$$

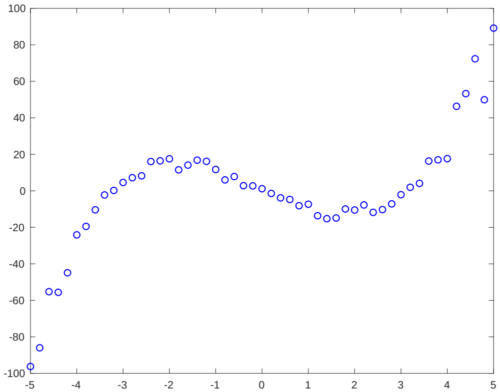
where $e_\ell = (1, \dots, 1) \in \mathbb{R}^\ell$, in matrix form (3) becomes:

$$\begin{cases} \min\limits_{z,u} \ (0, 0, \dots, 0, 1) \begin{pmatrix} z \\ u \end{pmatrix} \\ D \begin{pmatrix} z \\ u \end{pmatrix} \leq d \end{cases}$$

Exercise 6.1. Consider the experimental data (x_i, y_i), $i = 1, .., \ell$, given below:

$$\begin{pmatrix} -5.0000 & -96.2607 \\ -4.8000 & -85.9893 \\ -4.6000 & -55.2451 \\ -4.4000 & -55.6153 \\ -4.2000 & -44.8827 \\ -4.0000 & -24.1306 \\ -3.8000 & -19.4970 \\ -3.6000 & -10.3972 \\ -3.4000 & -2.2633 \\ -3.2000 & 0.2196 \\ -3.0000 & 4.5852 \\ -2.8000 & 7.1974 \\ -2.6000 & 8.2207 \\ -2.4000 & 16.0614 \\ -2.2000 & 16.4224 \\ -2.0000 & 17.5381 \\ -1.8000 & 11.4895 \\ -1.6000 & 14.1065 \\ -1.4000 & 16.8220 \\ -1.2000 & 16.1584 \\ -1.0000 & 11.6846 \\ -0.8000 & 5.9991 \\ -0.6000 & 7.8277 \\ -0.4000 & 2.8236 \\ -0.2000 & 2.7129 \\ 0 & 1.1669 \end{pmatrix} \qquad \begin{pmatrix} 0.2000 & -1.4223 \\ 0.4000 & -3.8489 \\ 0.6000 & -4.7101 \\ 0.8000 & -8.1538 \\ 1.0000 & -7.3364 \\ 1.2000 & -13.6464 \\ 1.4000 & -15.2607 \\ 1.6000 & -14.8747 \\ 1.8000 & -9.9271 \\ 2.0000 & -10.5022 \\ 2.2000 & -7.7297 \\ 2.4000 & -11.7859 \\ 2.6000 & -10.2662 \\ 2.8000 & -7.1364 \\ 3.0000 & -2.1166 \\ 3.2000 & 1.9428 \\ 3.4000 & 4.0905 \\ 3.6000 & 16.3151 \\ 3.8000 & 16.9854 \\ 4.0000 & 17.6418 \\ 4.2000 & 46.3117 \\ 4.4000 & 53.2609 \\ 4.6000 & 72.3538 \\ 4.8000 & 49.9166 \\ 5.0000 & 89.1652 \end{pmatrix}$$

Find the best approximating polynomials of degree 3 with respect to the norms $\| \cdot \|_2$, $\| \cdot \|_1$, $\| \cdot \|_\infty$.



- For the case of norm $\|\cdot\|_2$, the solution is given by (1).
- For the cases of norms $\|\cdot\|_2$, $\|\cdot\|_1$, $\|\cdot\|_\infty$, we have to solve problems (2) and (3) written in matrix form.

We will use the Matlab function "linprog".

From the Matlab help

- X = linprog(f,A,b) attempts to solve the linear programming problem:

$$\min_x f' * x \text{ subject to: } A * x <= b$$

- X = linprog(f,A,b,Aeq,beq) solves the problem above while additionally satisfying the equality constraints $Aeq * x = beq$.
  (Set A=[] and B=[] if no inequalities exist.)
- X = linprog(f,A,b,Aeq,beq,LB,UB) defines a set of lower and upper bounds on the design variables, X, so that the solution is in the range LB <= X <= UB. Use empty matrices for LB and UB if no bounds exist. Set LB(i) = -Inf if X(i) is unbounded below; set UB(i) = Inf if X(i) is unbounded above.

Exercise 6.1: Matlab commands

```
data =[....];              %Matrix of given data

x = data(:,1) ;
y = data(:,2) ;
l = length(x) ;
n = 4 ;                          % number of coefficients of polynomial
A = [ ones(l,1) x x.^ 2 x.^ 3 ];    % Vandermonde matrix
```

% 2-norm problem

```
z2 = inv(A'*A)*(A'*y)
p2 = A*z2;               % regression values at the data
```

% 1-norm problem

```
c = [ zeros(n,1); ones(l,1) ];
D = [ A -eye(l); -A -eye(l) ];
d = [ y; -y ];
sol1 = linprog(c,D,d) ;
z1 = sol1(1:n)
p1 = A*z1;               % regression values at the data
```

% inf-norm problem

```
c = [ zeros(n,1); 1 ];
D = [ A -ones(l,1); -A -ones(l,1) ];
solinf = linprog(c,D,d) ;
zinf = solinf(1:n)
pinf = A*zinf;               % regression values at the data
```

% plot the solutions

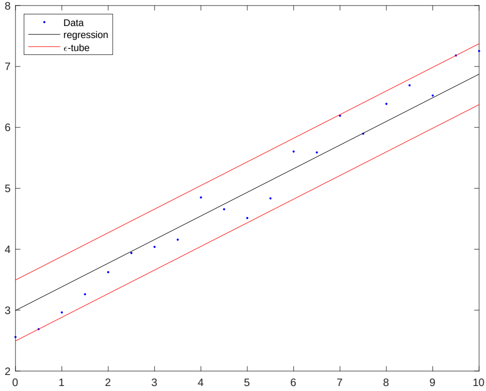
```
plot(x,y,'b.',x,p2,'r-',x,p1,'k-',x,pinf,'g-')
legend('Data','2-norm','1-norm','inf-norm','Location','NorthWest');
```

ε-SV regression

Given a set of training data $\{(x_1, y_1), \ldots, (x_\ell, y_\ell)\}$, where $x_i \in \mathbb{R}^n$, $y_i \in \mathbb{R}$, and a tolerance $\varepsilon > 0$, in ε-SV regression we aim at finding a function f such that

$$|f(x_i) - y_i| \leq \varepsilon, \ \ i = 1, .., \ell$$

and it fulfills suitable properties (for example, it is as flat as possible)



In a linear regression, we consider an affine function $f(x) = w^{\mathsf{T}}x + b$ and set a tolerance parameter $\varepsilon > 0$.

If we want f to be flat, then we must seek for a small w, which leads us to solve the convex quadratic optimization problem

$$\begin{cases} \min\limits_{w,b} \ \dfrac{1}{2}\|w\|^2 \\ y_i \leq w^{\mathsf{T}}x_i + b + \varepsilon \qquad \forall\ i = 1, \dots, \ell \\ y_i \geq w^{\mathsf{T}}x_i + b - \varepsilon \qquad \forall\ i = 1, \dots, \ell \end{cases} \tag{4}$$

In matrix form (4) becomes:

$$\begin{cases} \min\limits_{w,b} \ \frac{1}{2}(w^T, b)Q\begin{pmatrix} w \\ b \end{pmatrix} \\ D\begin{pmatrix} w \\ b \end{pmatrix} \leq d \end{cases}$$

where

$$Q = \begin{pmatrix} I_n & 0_n \\ 0_n^T & 0 \end{pmatrix} \quad D = \begin{pmatrix} -X & -e_\ell \\ X & e_\ell \end{pmatrix} \quad d = \begin{pmatrix} \varepsilon e_\ell - y \\ \varepsilon e_\ell + y \end{pmatrix}$$

and X is a $\ell \times n$ matrix with row i given by x_i^T, $i = 1, .., \ell$.

Remark

If $x_i \in \mathbb{R}$, then $X = x := (x_1, x_2, ..., x_\ell)^T$ so that

$$D = \begin{pmatrix} -x & -e_\ell \\ x & e_\ell \end{pmatrix}$$

Exercise 6.2. Apply the linear ε-SV regression model with $\varepsilon = 0.5$ to the following training data

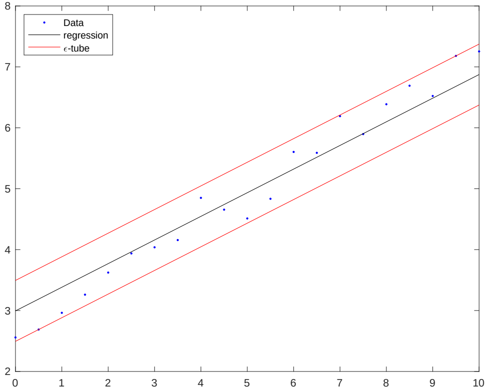
$$\begin{pmatrix} 0 & 2.5584 \\ 0.5000 & 2.6882 \\ 1.0000 & 2.9627 \\ 1.5000 & 3.2608 \\ 2.0000 & 3.6235 \\ 2.5000 & 3.9376 \\ 3.0000 & 4.0383 \\ 3.5000 & 4.1570 \\ 4.0000 & 4.8498 \\ 4.5000 & 4.6561 \\ 5.0000 & 4.5119 \\ 5.5000 & 4.8346 \\ 6.0000 & 5.6039 \\ 6.5000 & 5.5890 \\ 7.0000 & 6.1914 \\ 7.5000 & 5.8966 \\ 8.0000 & 6.3866 \\ 8.5000 & 6.6909 \\ 9.0000 & 6.5224 \\ 9.5000 & 7.1803 \\ 10.0000 & 7.2537 \end{pmatrix}$$

Matlab commands

```
data=[ ]
x = data(:,1) ;
y = data(:,2) ;
l = length(x) ; % number of points
epsilon = 0.5 ;

Q = [ 1 0; 0 0 ];
c = [0;0];
D = [-x -ones(l,1); x ones(l,1)];
d = epsilon*ones(2*l,1) + [-y;y];

sol = quadprog(Q,c,D,d);
w = sol(1);
b = sol(2);
z = w.*x + b ;
zp = w.*x + b + epsilon ;
zm = w.*x + b - epsilon ;

% plot the solution

plot(x,y,'b.',x,z,'k-',x,zp,'r-',x,zm,'r-');
legend('Data','regression','ϵ-tube',... 'Location','NorthWest')
```

w =
0.3880

b =
2.9942



If ε is too small, the model (4) might not be feasible.

The linear ε-SV regression model can be extended by introducing slack variables ξ^+ and ξ^- to relax the constraints of problem (4):

$$\begin{cases} \min\limits_{w,b,\xi^+,\xi^-} \ \dfrac{1}{2}\|w\|^2 + C\displaystyle\sum_{i=1}^{\ell}(\xi_i^+ + \xi_i^-) \\ y_i \le w^{\mathsf{T}}x_i + b + \varepsilon + \xi_i^+ \qquad \forall\ i = 1,\dots,\ell \\ y_i \ge w^{\mathsf{T}}x_i + b - \varepsilon - \xi_i^- \qquad \forall\ i = 1,\dots,\ell \\ \xi^+ \ge 0 \\ \xi^- \ge 0 \end{cases} \tag{5}$$

where parameter C gives the trade-off between the flatness of f and tolerance to deviations larger than ε.

In matrix form (5) becomes:

$$\begin{cases} \min\limits_{w,b} \ \frac{1}{2}(w^T, b, (\xi^+)^\mathsf{T}, (\xi^-)^\mathsf{T}) Q1 \begin{pmatrix} w \\ b \\ \xi^+ \\ \xi^- \end{pmatrix} + c^\mathsf{T} \begin{pmatrix} w \\ b \\ \xi^+ \\ \xi^- \end{pmatrix} \\ D1 \begin{pmatrix} w \\ b \\ \xi^+ \\ \xi^- \end{pmatrix} \leq d1 \\ \xi^+ \geq 0, \ \xi^- \geq 0 \end{cases}$$

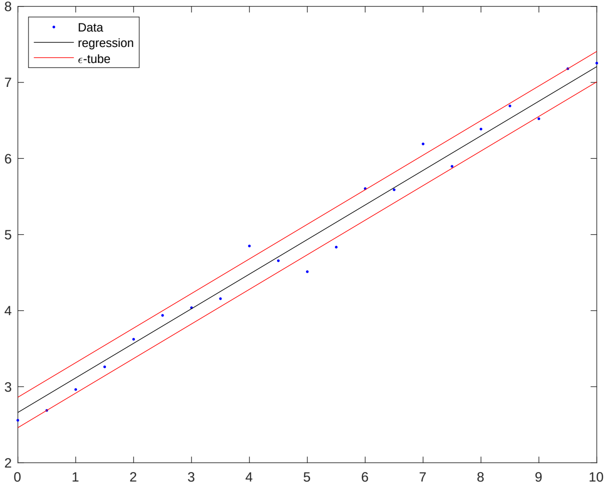
where

$$Q1 = \begin{pmatrix} I_n & 0_n & O_{n\times 2\ell} \\ 0_n^T & 0 & 0_{2\ell}^T \\ O_{2\ell\times n} & 0_{2\ell} & O_{2\ell\times 2\ell} \end{pmatrix} \quad c^T = (O_n^T, 0, C * e_\ell^T, C * e_\ell^T)$$

$$D1 = \begin{pmatrix} -X & -e_\ell & -I_\ell & O_{\ell\times\ell} \\ X & e_\ell & O_{\ell\times\ell} & -I_\ell \end{pmatrix} \quad d = \begin{pmatrix} \varepsilon e_\ell - y \\ \varepsilon e_\ell + y \end{pmatrix}$$

and X is a $\ell \times n$ matrix with row i given by x_i^T, $i = 1, .., \ell$.

Exercise 6.3. Apply the linear ε-SV regression model with slack variables (set $\varepsilon = 0.2$ and $C = 10$) to the training data given in Exercise 6.2.



Linear ε-SV regression with slack variables - dual problem

Let us compute the dual of problem (5). The Lagrangian function is

$$L(\underbrace{w, b, \xi^+, \xi^-}_{\text{primal var.}}, \underbrace{\lambda^+, \lambda^-, \eta^+, \eta^-}_{\text{dual var.}}) = \frac{1}{2}\|w\|^2 - w^\mathsf{T}\left[\sum_{i=1}^{\ell}(\lambda_i^+ - \lambda_i^-)x_i\right] - b\sum_{i=1}^{\ell}(\lambda_i^+ - \lambda_i^-)$$

$$+\sum_{i=1}^{\ell}\xi_i^+(C - \lambda_i^+ - \eta_i^+) + \sum_{i=1}^{\ell}\xi_i^-(C - \lambda_i^- - \eta_i^-) - \varepsilon\sum_{i=1}^{\ell}(\lambda_i^+ + \lambda_i^-) + \sum_{i=1}^{\ell}y_i(\lambda_i^+ - \lambda_i^-)$$

If $\sum_{i=1}^{\ell}(\lambda_i^+ - \lambda_i^-) \neq 0$ or $C - \lambda_i^+ - \eta_i^+ \neq 0$ for some i or $C - \lambda_i^- - \eta_i^- \neq 0$ for some i, then $\min_{w,b,\xi^+,\xi^-} L = -\infty$. Otherwise,

$$\nabla_w L = w - \sum_{i=1}^{\ell}(\lambda_i^+ - \lambda_i^-)x_i = 0.$$

Operating as in previously made calculations, we substitute $w = \sum_{i=1}^{\ell}(\lambda_i^+ - \lambda_i^-)x_i$ in the Lagrangian and we consider the above mentioned constraints.

The dual problem of (5) is

$$
\begin{cases}
\max\limits_{\lambda^+,\lambda^-,\eta^+,\eta^-\geq 0} \; -\dfrac{1}{2}\sum\limits_{i=1}^{\ell}\sum\limits_{j=1}^{\ell}(\lambda_i^+-\lambda_i^-)(\lambda_j^+-\lambda_j^-)(x_i)^{\mathsf{T}}x_j \\
\qquad\qquad -\varepsilon\sum\limits_{i=1}^{\ell}(\lambda_i^+ + \lambda_i^-) + \sum\limits_{i=1}^{\ell} y_i(\lambda_i^+-\lambda_i^-) \\
\sum\limits_{i=1}^{\ell}(\lambda_i^+-\lambda_i^-) = 0 \\
C-\lambda_i^+-\eta_i^+ = 0, \;\; i=1,..,\ell \\
C-\lambda_i^--\eta_i^- = 0, \;\; i=1,..,\ell
\end{cases}
$$

Finally, eliminating the variables η_i^+, η_i^- we obtain:

$$
\begin{cases}
\max\limits_{\lambda^+,\lambda^-} \; -\dfrac{1}{2}\sum\limits_{i=1}^{\ell}\sum\limits_{j=1}^{\ell}(\lambda_i^+-\lambda_i^-)(\lambda_j^+-\lambda_j^-)(x_i)^{\mathsf{T}}x_j \\
\qquad\qquad -\varepsilon\sum\limits_{i=1}^{\ell}(\lambda_i^+ + \lambda_i^-) + \sum\limits_{i=1}^{\ell} y_i(\lambda_i^+-\lambda_i^-) \\
\sum\limits_{i=1}^{\ell}(\lambda_i^+-\lambda_i^-) = 0 \\
\lambda_i^+ \in [0, C], \;\; i=1,..,\ell \\
\lambda_i^- \in [0, C], \;\; i=1,..,\ell
\end{cases}
$$

Let us write the dual problem in matrix form.

Consider first the quadratic part of the objective function.

It is possible to show that, setting

$$X = [(x_i)^\mathsf{T} x_j] \quad i,j = 1,..,\ell, \qquad Q = \begin{pmatrix} X & -X \\ -X & X \end{pmatrix}, \tag{6}$$

then, we have:

$$-\frac{1}{2}\sum_{i=1}^{\ell}\sum_{j=1}^{\ell}(\lambda_i^+ - \lambda_i^-)(\lambda_j^+ - \lambda_j^-)(x_i)^\mathsf{T} x_j = -\frac{1}{2}((\lambda^+)^\mathsf{T}, (\lambda^-)^\mathsf{T}) Q \begin{pmatrix} \lambda^+ \\ \lambda^- \end{pmatrix}$$

Indeed, note that

$$\sum_{i=1}^{\ell}\sum_{j=1}^{\ell}(\lambda_i^+ - \lambda_i^-)(\lambda_j^+ - \lambda_j^-)(x_i)^\mathsf{T} x_j$$

$$= \sum_{i=1}^{\ell}\sum_{j=1}^{\ell}(\lambda_i^+\lambda_j^+ - \lambda_i^+\lambda_j^- - \lambda_i^-\lambda_j^+ + \lambda_i^-\lambda_j^-)(x_i)^\mathsf{T} x_j \tag{7}$$

Moreover,

$$((\lambda^+)^\mathsf{T}, (\lambda^-)^\mathsf{T}) Q \begin{pmatrix} \lambda^+ \\ \lambda^- \end{pmatrix} = ((\lambda^+)^\mathsf{T}, (\lambda^-)^\mathsf{T}) \begin{pmatrix} X & -X \\ -X & X \end{pmatrix} \begin{pmatrix} \lambda^+ \\ \lambda^- \end{pmatrix} =$$

$$= [(\lambda^+)^\mathsf{T} X - (\lambda^-)^\mathsf{T} X]\lambda^+ + [(\lambda^+)^\mathsf{T}(-X) + (\lambda^-)^\mathsf{T} X]\lambda^-$$

$$= (\lambda^+)^\mathsf{T} X \lambda^+ - (\lambda^-)^\mathsf{T} X \lambda^+ - (\lambda^+)^\mathsf{T} X \lambda^- + (\lambda^-)^\mathsf{T} X \lambda^-$$

which equals (7).

Finally, we obtain:

Dual problem in matrix form

$$\begin{cases} \max\limits_{\lambda^+, \lambda^-} \; -\dfrac{1}{2}((\lambda^+)^\mathsf{T}, (\lambda^-)^\mathsf{T}) Q \begin{pmatrix} \lambda^+ \\ \lambda^- \end{pmatrix} + [-\epsilon(e_\ell^\mathsf{T}, e_\ell^\mathsf{T}) + (y^\mathsf{T}, -y^\mathsf{T})] \begin{pmatrix} \lambda^+ \\ \lambda^- \end{pmatrix} \\ (e_\ell^\mathsf{T}, -e_\ell^\mathsf{T}) \begin{pmatrix} \lambda^+ \\ \lambda^- \end{pmatrix} = 0 \\ \lambda_i^+ \in [0, C], \;\; i = 1, .., \ell \\ \lambda_i^- \in [0, C], \;\; i = 1, .., \ell \end{cases}$$

where Q is defined by (6).

- Dual problem is a convex quadratic programming problem
- If $\lambda_i^+ > 0$ or $\lambda_i^- > 0$, then x_i is said support vector
- If (λ^+, λ^-) is a dual optimum, then

$$w = \sum_{i=1}^{\ell} (\lambda_i^+ - \lambda_i^-) x_i, \tag{8}$$

- b is obtained using the complementarity conditions:

$$\begin{array}{l} \lambda_i^+ \left[\varepsilon + \xi_i^+ - y_i + w^\mathsf{T} x_i + b \right] = 0 \\ \lambda_i^- \left[\varepsilon + \xi_i^- + y_i - w^\mathsf{T} x_i - b \right] = 0 \\ \xi_i^+ (C - \lambda_i^+) = 0 \\ \xi_i^- (C - \lambda_i^-) = 0 \end{array}$$

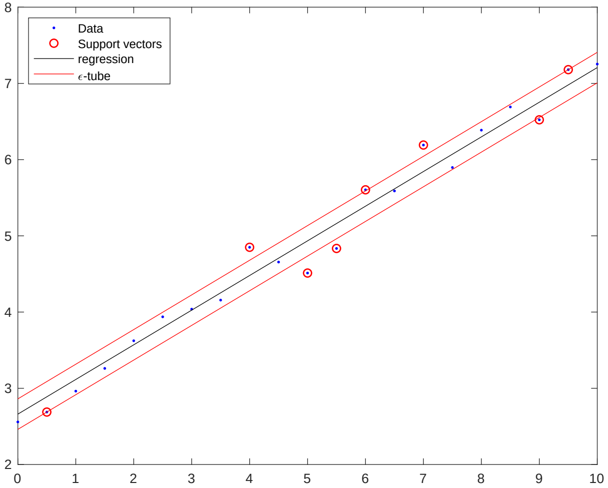
Hence, if there is some i s.t. $0 < \lambda_i^+ < C$, then

$$b = y_i - w^\mathsf{T} x_i - \varepsilon; \tag{9}$$

while, if there is some i s.t. $0 < \lambda_i^- < C$, then

$$b = y_i - w^\mathsf{T} x_i + \varepsilon. \tag{10}$$

Exercise 6.4. Solve the dual problem of the linear ε-SV regression model with slack variables (set $\varepsilon = 0.2$ and $C = 10$) applied to the training data given in Exercise 6.2.



Matlab commands

```
data=[ .....]

x = data(:,1) ; y = data(:,2) ; l = length(x) ; epsilon = 0.2 ; C = 10;

X = zeros(l,l);
for i = 1 : l
    for j = 1 : l
        X(i,j) = x(i)*x(j);
    end
end
Q = [ X -X ; -X X ];
c = epsilon*ones(2*l,1) + [-y;y];

% solve the problem

sol = quadprog(Q,c,[],[],[ones(1,l) -ones(1,l)],0,zeros(2*l,1),C*ones(2*l,1));
lap = sol(1:l);
lam = sol(l+1:2*l);

% compute w

w = (lap-lam)'*x ;
```

Matlab commands (continued)

```
% compute b

ind = find(lap > 0.001 & lap < C- 0.001);
if isempty(ind)==0
i = ind(1);
b = y(i) - w*x(i) - epsilon ;
else
ind = find(lam > 0.001 & lam < C- 0.001);
i = ind(1);
b = y(i) - w*x(i) + epsilon ;
end

% find regression and epsilon-tube

z = w.*x + b ;
zp = w.*x + b + epsilon ;
zm = w.*x + b - epsilon ;

% find support vectors and plot the solution

sv = [find(lap > 1e-3);find(lam > 1e-3)];
sv = sort(sv);

plot(x,y,'b.',x(sv),y(sv), 'ro',x,z,'k-',x,zp,'r-',x,zm,'r-');
```

Nonlinear ε-SV regression

In order to generate a nonlinear regression function f, we can use the kernel functions (see part 5 related to classification problems).

Consider a map $\phi : \mathbb{R}^n \to \mathcal{H}$ (associated to a kernel function), where $\mathcal{H}$ (features space) is a higher dimensional (maybe infinite) space and find a linear regression for the points $\{(\phi(x_i), y_i)\}$ in the space $\mathcal{H} \times \mathbb{R}$.

The primal problem becomes:

$$\begin{cases} \min \ \dfrac{1}{2}\|w\|^2 + C \displaystyle\sum_{i=1}^{\ell} (\xi_i^+ + \xi_i^-) \\ y_i \leq w^{\mathsf{T}} \phi(x_i) + b + \varepsilon + \xi_i^+ \qquad \forall\ i = 1, \dots, \ell \\ y_i \geq w^{\mathsf{T}} \phi(x_i) + b - \varepsilon - \xi_i^- \qquad \forall\ i = 1, \dots, \ell \end{cases}$$

where w is a vector in a high (possibly infinite) dimensional space called the dual space of $\mathcal{H}$.

The dual problem is:

$$\begin{cases} \max\limits_{(\lambda^+,\lambda^-)} & -\dfrac{1}{2}\sum\limits_{i=1}^{\ell}\sum\limits_{j=1}^{\ell}(\lambda_i^+ - \lambda_i^-)(\lambda_j^+ - \lambda_j^-)\phi(x_i)^\mathsf{T}\phi(x_j) \\ & \quad -\varepsilon\sum\limits_{i=1}^{\ell}(\lambda_i^+ + \lambda_i^-) + \sum\limits_{i=1}^{\ell} y_i(\lambda_i^+ - \lambda_i^-) \\ \sum\limits_{i=1}^{\ell}(\lambda_i^+ - \lambda_i^-) = 0 \\ \lambda_i^+, \lambda_i^- \in [0, C] \end{cases}$$

or,

$$\begin{cases} \max\limits_{(\lambda^+,\lambda^-)} & -\dfrac{1}{2}\sum\limits_{i=1}^{\ell}\sum\limits_{j=1}^{\ell}(\lambda_i^+ - \lambda_i^-)(\lambda_j^+ - \lambda_j^-)k(x_i, x_j) \\ & \quad -\varepsilon\sum\limits_{i=1}^{\ell}(\lambda_i^+ + \lambda_i^-) + \sum\limits_{i=1}^{\ell} y_i(\lambda_i^+ - \lambda_i^-) \\ \sum\limits_{i=1}^{\ell}(\lambda_i^+ - \lambda_i^-) = 0 \\ \lambda_i^+, \lambda_i^- \in [0, C] \end{cases}$$

a finite dimensional problem with 2ℓ variables

- choose a kernel function $k(x, y)$
- solve the dual and find (λ^+, λ^-)
- Recall that, by (8) now we have:

$$w = \sum_{i=1}^{\ell} (\lambda_i^+ - \lambda_i^-)\phi(x_i), \tag{11}$$

and, consequently,

$$w^T\phi(x) = \sum_{i=1}^{\ell} (\lambda_i^+ - \lambda_i^-)\phi(x_i)\phi(x) = \sum_{i=1}^{\ell} (\lambda_i^+ - \lambda_i^-)k(x_i, x).$$

By (9) and (10) we compute b:

$$b = y_i - \varepsilon - \sum_{j=1}^{\ell} (\lambda_j^+ - \lambda_j^-)k(x_i, x_j), \qquad \text{for some } i \text{ s.t. } 0 < \lambda_i^+ < C$$

or

$$b = y_i + \varepsilon - \sum_{j=1}^{\ell} (\lambda_j^+ - \lambda_j^-)k(x_i, x_j), \qquad \text{for some } i \text{ s.t. } 0 < \lambda_i^- < C$$

(Recall that $k(x_i, x_j) = k(x_j, x_i)$.)

Finally, the regression function is:

$$f(x) = w^\top \phi(x) + b = \sum_{i=1}^{\ell} (\lambda_i^+ - \lambda_i^-) k(x_i, x) + b$$

Remark

The regression function is

- linear in the features space
- nonlinear in the input space

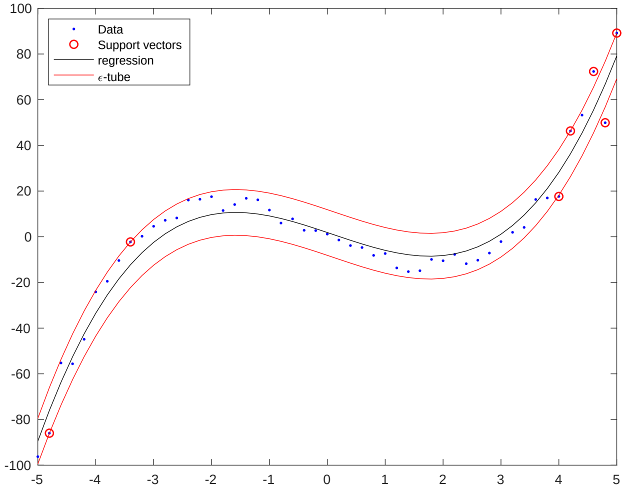
Exercise 6.5. Consider the training data given in Exercise 6.1. Apply the nonlinear ε-SV regression using a polynomial kernel with degree $p = 4$ and parameters $\varepsilon = 10$, $C = 10$. Moreover, find the support vectors.

Matlab commands

```
data=[ .....]

x = data(:,1) ; y = data(:,2) ; l = length(x) ; epsilon = 10 ; C = 10;

X = zeros(l,l); for i = 1 : l
    for j = 1 : l
        X(i,j) = kernel(x(i),x(j));
    end
end
Q = [ X -X ; -X X ];
c = epsilon*ones(2*l,1) + [-y;y];

% solve the problem

sol = quadprog(Q,c,[],[],[ones(1,l) -ones(1,l)],0,zeros(2*l,1),C*ones(2*l,1));
lap = sol(1:l);
lam = sol(l+1:2*l);

% compute b

ind = find(lap > 1e-3 & lap < C-1e-3);
if isempty(ind)==0
i = ind(1);
```

```
b = y(i) - epsilon;
for j = 1 : l
b = b - (lap(j)-lam(j))*kernel(x(i),x(j));
end
else
ind = find(lam > 1e-3 & lam < C-1e-3);
i = ind(1);
b = y(i) + epsilon ;
for j = 1 : l
b = b - (lap(j)-lam(j))*kernel(x(i),x(j));
end
end

% find regression and epsilon-tube evaluating f(x) at x(i), i=1,..,l

z = zeros(l,1);
for i = 1 : l
z(i) = b ;
for j = 1 : l
z(i) = z(i) + (lap(j)-lam(j))*kernel(x(i),x(j));
end
end
zp = z + epsilon ;
zm = z - epsilon ;
```

```
% find support vectors and plot the solution

sv = [find(lap > 1e-3);find(lam > 1e-3)];
sv = sort(sv);
plot(x,y,'b.',x(sv),y(sv), 'ro',x,z,'k-',x,zp,'r-',x,zm,'r-');

% kernel function

function v = kernel(x,y)

p = 4 ;

v = (x'*y + 1)^ p;

end
```


Data
Support vectors
regression
ϵ-tube
100
80
60
40
20
0
-20
-40
-60
-80
-100
-5
-4
-3
-2
-1
0
1
2
3
4
5