T-61.6040 Assignment-02/2011

Santosh Tirunagari (245577)
santosh.tirunagari@aalto.fi

October 6, 2011

AIM

Implemention of support vector regression algorithm discussed in the lecture using Gaussian kernel for a motorcycle dataset.

$$k_{GAU} = (x_i, x_j) = exp(-\|x_i x_j\|_2^2/s^2),\ C = 100,\ and\ s = \{0.0625, 0.25, 1, 4, 16\}; and\ \epsilon = 16. \quad (1)$$

$$k_{GAU} = (x_i, x_j) = exp(-\|x_i x_j\|_2^2/s^2),\ C = 100,\ and\ s = 1\ and\ \epsilon = \{1, 4, 16, 64, 256\} \quad (2)$$

Datasets

Motorcycle dataset for regression analysis.

Implementation

Load and zscore normalize both the X and Y data using zscore matlab command. Zscore normalization can also be done using the method described in the lecture.

```
X = zscore(dlmread('motorcycle_X.txt'));
y = dlmread('motorcycle_y.txt');
```

Calculate Gaussian kernel function.

```
K = exp(-bsxfun(@plus,sum(X.^2,2),...
    bsxfun(@plus,sum(X.^2,2)',-2.*X*X'))./(S(k).^2));
```

Solve SVR optimization problem in standard QP form using quadprog function in matlab and find alpha values.

```
H = [+K, -K; -K, +K];
A = [];
Aeq = [+ones(N, 1); -ones(N, 1)]';
l = zeros(2 * N, 1);
c = [epsilon * ones(N, 1) - y; epsilon * ones(N, 1) + y];
b = [];
beq = 0;
u = C * ones(2 * N, 1);
options = optimset('Algorithm','interior-point-convex','MaxIter',1500);
% solve quadprog to find alpha values
alpha = quadprog(H, c, A, b, Aeq, beq, l, u,[],options);
% calculate near boundary coefficients
```

Compute the near boundary coefficients and move near boundary alpha values to boundaries. And also seperate the alpha positives and alpha negatives.

```
alpha(alpha < C * (eps)) = 0;
alpha(alpha > C * (1-eps)) = C;
alphap = alpha(1:133,1);
alphan = alpha(134:266,1);
```

Compute value for bias parameter b

```
svp = find(alphap>0 & alphap<C);
svp_one = zeros(N,1);
svp_one(sv,1) = 1;%(alpha(alphap >0 & alphap<C));
b=(svp_one'*(Y-(((alphap-alphan)'*K')-epsilon)'))/(sum(svp_one));
```

Compute the decision function on training set

```
Ki = X(sv,:)*X';
temp = bsxfun(@plus,Ki'*(alpha(sv,:).*Y(sv,:)),b);
```

Compute the decision function on test set

```
Ki = exp(-bsxfun(@plus,sum(X(sv,:).^2,2),...
bsxfun(@plus,sum(X.^2,2)',-2.*X(sv,:)*X'))./(S(k).^2));
yi = bsxfun(@plus,(Ki'*(alphap(sv,:)-alphan(sv,:))),b);
```

Plot the graphs showing different hyperplanes.

```
figure(k);
plot(X,Y,'+','linewidth',1); hold on
plot(X(sv,1),Y(sv,1),'o','linewidth',1);
[X_r idx] = sort(X);
plot(X_r,yi(idx,:),'k-','linewidth',2);
title(['epsilon = 16, S = ',num2str(S(k))]);
%---print figure---
fnout = ['S=', num2str(k,'%.2d'), '.jpg'];
print('-djpeg','-r150',fnout);
```

The Gaussian kernel implementation is carried out for all the values of $s = \{0.0625, 0.25, 1, 4, 16\}; and\ \epsilon = 16$ and these plots are shown in figure 1.

similarly the same implementation is carried out for all the values of $s = 1\ and\ \epsilon = \{1, 4, 16, 64, 256\}$ and these plots are shown in figure 2.

Running of code for Gaussian kernel in matlab for case1 and case2 are as svr1 ad svr2 respectively.

Results

Figure 1 shows that the S value increases the guassian kernel behaves more like a linear kernal. With the smaller S value the small Gaussian distribution is actually seen properly. As the S increases the width of the Gaussian distribution tends to increase.

Figure 1 shows that the ϵ value is significantly large the guassian kernel behaves more like a linear kernal. With the smaller $\epsilon valuethethereisnotmuchchangeinthehyperplanebutlargesignific$

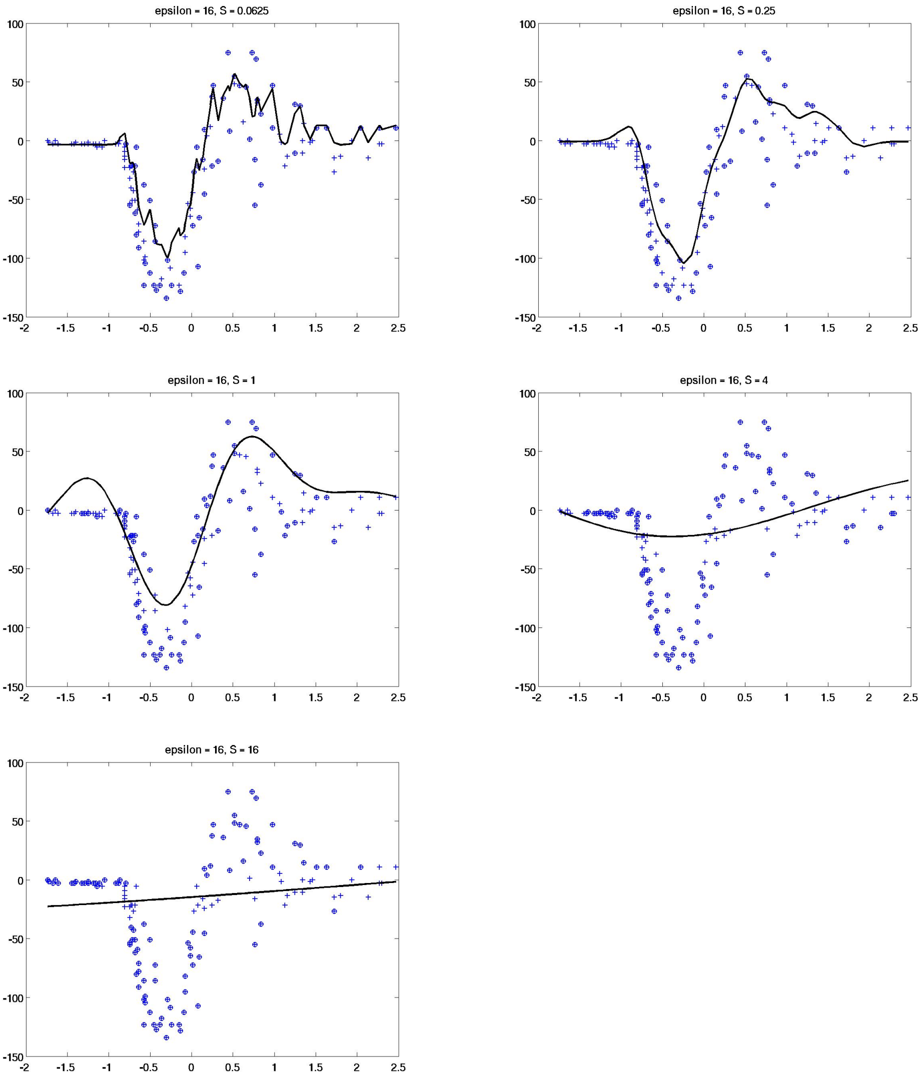
References

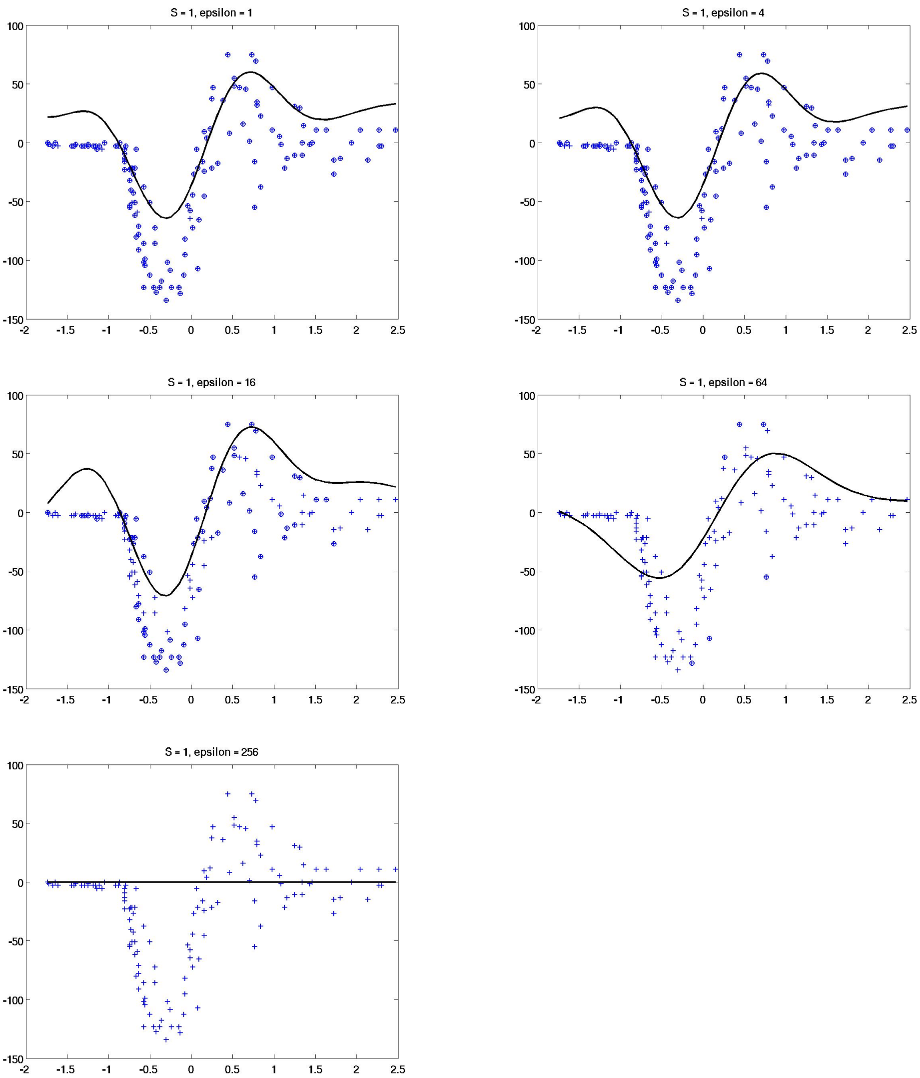
[1] Chapter 9 Scholkopf and Smola (2002)
[2] http://en.wikipedia.org/wiki/Support_vector_machine
[3] http://www.mathworks.se/help/toolbox/optim/ug/quadprog.html

Attachments

Matlab code svr1.m and svr2.m and datasets have been archived along with this report.

Figure 1: Plots showing the hyperlane seperations for different cases of S.




S = 1, epsilon = 1
S = 1, epsilon = 4
S = 1, epsilon = 16
S = 1, epsilon = 64
S = 1, epsilon = 256
100
50
0
-50
-100
-150
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5