T-61.6040 Assignment-03/2011

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

October 13, 2011

AIM

Implemention of support vector data description algorithm discussed in the lecture using Gaussian and linear kernel for a toy dataset.

$$k_{GAU} = (x_i, x_j) = exp(-\|x_i x_j\|_2^2/s^2),\ C = 0.2,\ and\ s = \{1, 2, 4, 8\}. \quad (1)$$

$$k_{Lin} = (x_i, x_j) =< x_i, x_j > \ andC = 0.2. \quad (2)$$

Datasets

Toy dataset with two classes.

Implementation

Load and zscore normalize the X. Zscore normalization can also be done using the method described in the lecture. Consider X dataset with class = Y = 1;

```
    Y = dlmread('toy_data_train_y.txt');
data = zscore(dlmread('toy_data_train_X.txt'));
```

Calculate Gaussian kernel function.

```
function [ out ] = kro(x1,x2,S,k)
%out = exp(-bsxfun(@plus,sum(x2.^2,2),bsxfun(@plus,sum(x1.^2,2)',-2.*x2*x1'))./
out = exp(-pdist2(x2,x1).^2/S(k).^2);
end
```

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

```
K = kro_l(X,X);
H = 2.*K;
A = [];
Aeq = ones(1,size(X,1));
l = zeros(size(X,1),1);
c = -diag(K);
b = [];
beq = 1;
u = C*ones(size(X,1), 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);
```

Compute the near boundary coefficients and move near boundary alpha values to boundaries.

```
% calculate near boundary coefficients
alpha(alpha < C * 0.001) = 0;
alpha(alpha > C*0.999) = C;
% get support vector indices
sv = find(alpha >0 & alpha<C);
```

Compute value for radius parameter r

```
dia = diag(kro_l(X(sv,:),X(sv,:)));
r = mean((dia-(2.*alpha(sv,:)'*kro_l(X(sv,:),X(sv,:)))')...
+(alpha(sv,:)'*kro_l(X(sv,:),X(sv,:))*alpha(sv,:)));
```

Compute the decision function

```
des = (r-(diag(K)+((2*alpha(sv,:)'*kro_l(X,X(sv,:)))-...
(alpha(sv,:)'*kro_l(X(sv,:),X(sv,:))*alpha(sv,:)))')));
res = des;
res(res>=0) = 1;
res(res<0) = -1;
```

Calculate the hypersphere

```
X_g = -4:.1:4;
Y_g = -4:.1:4;
[X1_o, Y1_o] = meshgrid(X_g,Y_g);
X1 = reshape(X1_o,size(X1_o,1)^2,1);
Y1 = reshape(Y1_o,size(Y1_o,1)^2,1);
B = [X1 Y1];
bound = r-diag(kro_l(B,B))+( (2*alpha(sv,:)'...
*kro_l(B,X(sv,:)))-(alpha(sv,:)'*kro_l(X(sv,:),X(sv,:))*alpha(sv,:)))';
X1 = reshape(X1,81,81);
Y1 = reshape(X1,81,81);
bound = reshape(bound,81,81);
```

Plot the graphs showing different hyperspheres.

```
    scatter(X(:,1),X(:,2));
hold on
scatter(X_an(:,1),X_an(:,2));
scatter(X(sv1,1),X(sv1,2),'k*');
scatter(X_an(sv2,1),X_an(sv2,2),'k*');

contour(X1_o,Y1_o,bound1,'LevelList',[0,0]);
contour(X1_o,Y1_o,bound2,'LevelList',[0,0]);
contour(X1_o,Y1_o,bound2-bound1,'LevelList',[0,0]);
axis([-2 2 -2 2])
title(['K_Gau, 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 $s = \{1, 2, 4, 8\}$ and these plots are shown in figure 1.

similarly the same implementation is carried out for linear kernel for $C = 0.2$ and these plots are shown in figure 2.

Running of code for Gaussian kernel and linear kernel in matlab are svm_guass and svm_lin 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 2 shows the linear kernel which is speherized the whole data and is not advised for this kind of problems. Figure 1 the σ value is significantly large the guassian kernel behaves more like a linear kernal. With the smaller σ value the there is not much change in the hypersphere and it is perfect but large significance would make Gaussian tend to be a linear kernel.

References

[3] http://www.mathworks.se/help/toolbox/optim/ug/quadprog.html

Attachments

Matlab code svm_guass.m and svm_lin.m and other function files and datasets have been archived along with this report.

Figure 1: Plots showing the hyperspheres S.

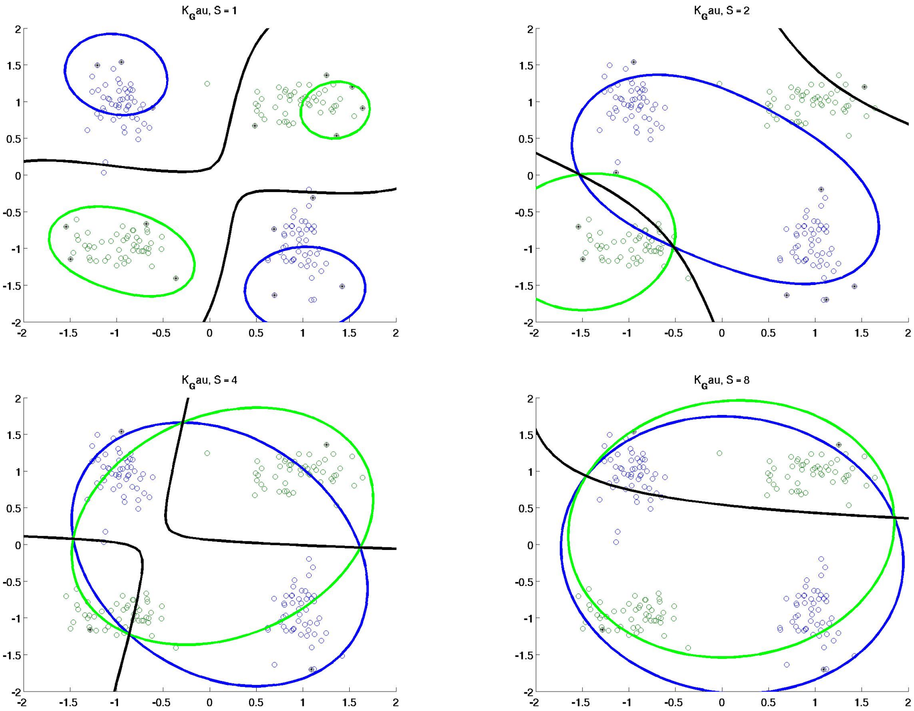

Figure 2: Plots showing the hyperspheres for linear kernel.