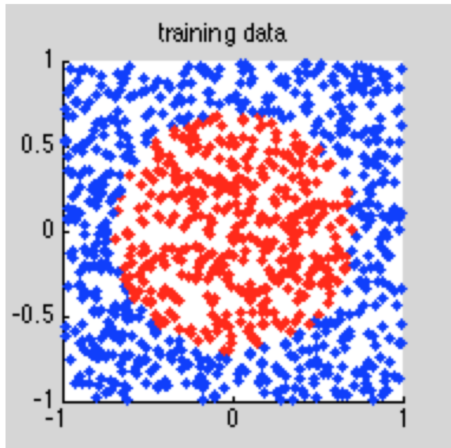
CS/ECE/ME 532
Homework 7: Classification and Kernel Methods

Consider the classification problem discussed in class, where the goal is to design a classifier based on the training data shown in the plot below. The Matlab file `training_data.m` generates this dataset and also computes the standard least squares solution.



1. Express the standard least squares solution using the duel form using regularization parameter $\lambda = 10^{-5}$. Verify that it generates the same results as the primal solution.

2. Design a classifier using the Gaussian kernel

$$k(\boldsymbol{a}_i, \boldsymbol{a}_j) = \exp(-\frac{1}{2}\|\boldsymbol{a}_i - \boldsymbol{a}_j\|_2^2)$$

and regularization parameter $\lambda = 10^{-5}$. Compare its classification of the training data to the least squares classification.

3. Design a classifier using the polynomial kernel

$$k(\boldsymbol{a}_i, \boldsymbol{a}_j) = (\boldsymbol{a}_i^T \boldsymbol{a}_j + 1)^2$$

and regularization parameter $\lambda = 10^{-5}$. Compare its classification of the training data to the least squares classification and Gaussian kernel classifier.

4. Now experiment with these methods in the following way. Generate different sized sets of training data: $m = 10, 100, 1000$. Design the three types of classifiers using these data. Test the performance of the classifiers using independent sets of data of size 100. For each training data size, repeat the training and testing 100 times and average the results. This will provide a good indication of how the different approaches perform with different amounts of training data. Summarize your results by reporting the average test error for each of the three methods and each of the three training set sizes.

```
% kernel classifier

clear
close all

% generate data
m = 1000;
n = 2;
b = zeros(m,1);
figure(1);
subplot(221); hold on;
for i=1:m
    a = 2*rand(2,1)-1;
    A(i,:)=a';
    b(i) = sign(a(1)^2+a(2)^2-.5);
    if b(i)==1
        plot(a(1),a(2),'b.');
    else
        plot(a(1),a(2),'r.');
    end
end
axis('square')
title('training data')

% least squares
subplot(222); hold on;
xLS = pinv(A)*b;
bLS = sign(A*xLS);



% or using duel solution
lam = 10e-5;
K=A*A';
bLS = sign(K*inv(K+lam*eye(size(K)))*b);

for i=1:m
    a = A(i,:);
    if bLS(i)==1
        plot(a(1),a(2),'b.');
    else
        plot(a(1),a(2),'r.');
    end
end
axis('square')
title('least squares')



% Gaussian kernel predition
subplot(223); hold on;
sig = 1;
lam = 10e-5;
for i=1:m
     na(i) = norm(A(i,:))^2;
end
kk = repmat(na,length(na),1)+repmat(na',1,length(na))-2*A*A';
Kg = exp(-kk/(2*sig^2));
alpha = inv(Kg+lam*eye(size(Kg)))*b;
bG = sign(Kg*alpha);
for i=1:m
    a = A(i,:);
    if bG(i)==1
        plot(a(1),a(2),'b.');
```

```
    else
        plot(a(1),a(2),'r.');
    end
end
axis('square')
title('Gaussian kernel')


% poly kernel predition
subplot(224); hold on
d = 2;
lam = 10e-5;
Kp = (A*A'+1).^d;
alpha = pinv(Kp+lam*eye(size(Kp)))*b;
bp = Kp*alpha;
bp = sign(bp);
for i=1:m
    a = A(i,:);
    if bp(i)==1
        plot(a(1),a(2),'b.');
    else
        plot(a(1),a(2),'r.');
    end
end
axis('square')
title('poly kernel')

break

% visualized data in 3-d space a(1)^2, a(1)*a(2),a(2)^2
figure(2);
for i=1:m
    a = A(i,:);
    q = [a(1)^2 a(1)*a(2) a(2)^2];
    Q(i,:) = q;
end
ip = find(b==1);
in = find(b==-1);
scatter3(Q(ip,1),Q(ip,2),Q(ip,3),'b.');
hold on
scatter3(Q(in,1),Q(in,2),Q(in,3),'r.');
```