Kyle Decker

Machine Learning

HW #4

1/ a) $\mathcal{L}(f, \theta_0) = \sum_{i=1}^{n} \ln\left(1 + \exp(-y_i(f(x_i) + \theta_0))\right) + \lambda \|f\|_H^2$

where $y_i \in \{-1, 1\}$, $f(x) = \theta^T x$ and $\|f\|_H^2 = \theta^T\theta$

$$\|f\|_{HK} = \sqrt{\langle f, f\rangle_{HK}}$$

$$K(x, z) = (x^T z + c)$$

$$K(\theta, x) = (\theta^T x + \theta_0) = \sum_{j=1}^{p} \theta^{(j)} x^{(j)} + \theta_0$$

$$\Phi(\theta) = [\theta^{(1)}, \theta^{(2)}, \theta^{(3)}, \dots, \theta_0$$

$$f(\cdot) = \sum_{i=1}^{m} \alpha_i K(\cdot, x_i)$$

$$\text{spam}\{\{\Phi(x) : x \in \mathcal{X}\}\} = \{f(\cdot) = \sum_{i=1}^{m} \alpha_i K(\cdot\, x_i) : m \in N, \theta_i \in \mathcal{X}, \alpha_i \in \mathbb{R}\}$$

$$\langle f, g\rangle_{HK} = \sum_{i=1}^{m} \sum_{j=1}^{m'} \alpha_i \beta_j K(x_i, x_j')$$

$$\langle K(\cdot, x), f\rangle_{HK} = f(x)$$

$$\langle K(\cdot, \cancel{x}), K(\cdot, \cancel{x}')\rangle_{HK} = K(x, x') \quad (\text{RKHS})$$

$$|f(x)|^2 = |\langle K(\cdot, x), f\rangle_{HK}|^2 \leq \langle K(\cdot, x), K(\cdot, x)\rangle_{HK} \cdot \langle f, f\rangle_{HK}$$
$$= K(x, x)\langle f, f\rangle_{HK}$$

$$\|f\|_{HK} = \sqrt{\langle f, f\rangle_{HK}}$$

$$H_K = \{f : f = \sum_i \alpha_i K(\cdot, x_i)\}$$

Using the representer theorem, we only need to solve for $\alpha_i \Rightarrow f^* = \sum_{i=1}^{n} \alpha_i K(x_i, \cdot)$

Table of Contents

Define x-axis

```
chi = [-2:0.01:2];
```

Logistic Loss

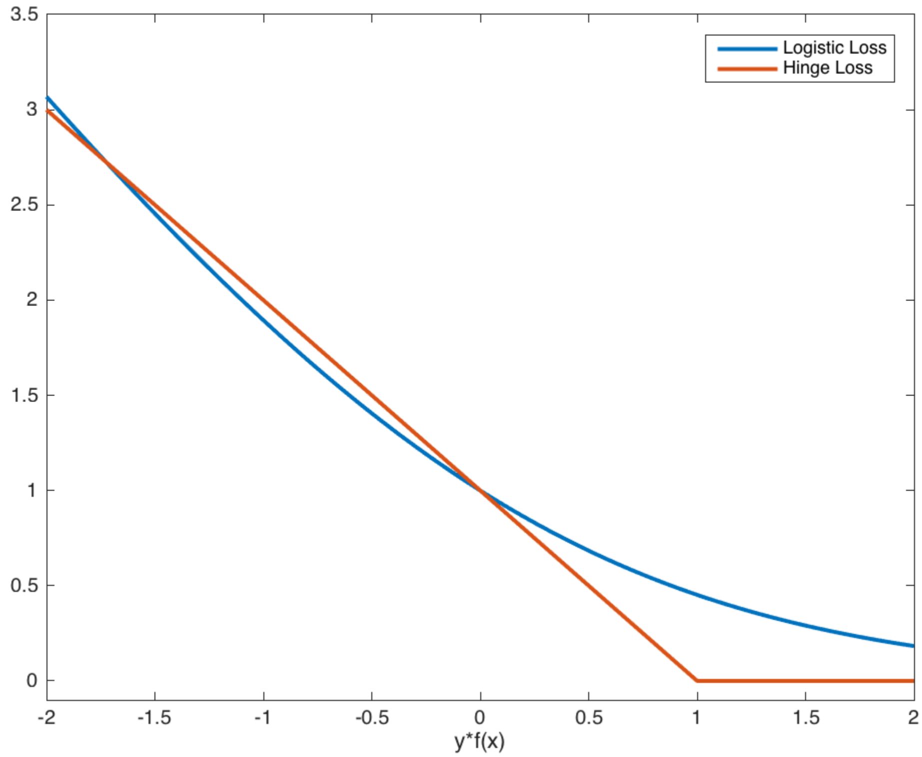
```
logistic_loss = log(1+exp(-1*chi));
```

Hinge Loss

```
hinge_loss = (1-chi);
index = (hinge_loss<0);
hinge_loss(index) = 0;
```

Plot the 2 loss functions and compare

```
figure;
plot(chi,1/log(2)*logistic_loss,'LineWidth',2);
hold on
plot(chi,hinge_loss,'LineWidth',2);
xlabel('y*f(x)')
legend('Logistic Loss', 'Hinge Loss')
axis([-2,2,-0.1,3.5])
```



Published with MATLAB® R2015b

1/c) $\min_{\theta,\theta_0,\xi} \frac{1}{2}\|\theta\|^2 + C\sum_{i=1}^{n} \ln(1+e^{-\xi_i})$ s.t. $y_i(\theta^T x_i + \theta_0) \geq \xi_i \ \forall i$

$$\mathcal{L}(\theta,\theta_0,\xi,\alpha,r) = \frac{1}{2}\|\theta\|_2^2 + C\sum_{i=1}^{n} \ln(1+e^{-\xi_i}) - \sum_{i=1}^{n}\alpha_i[y_i(\theta^T x_i+\theta_0)-\xi_i] - \sum_{i=1}^{n} r_i\xi_i$$

Lagrangian stationarity:

$$\nabla_\theta \mathcal{L}(\theta,\theta_0,\xi,\alpha,r) = \theta - \sum_{i=1}^{n}\alpha_i y_i x_i = 0$$

$$\Rightarrow \theta = \sum_{i=1}^{n}\alpha_i y_i x_i$$

$$\frac{\partial}{\partial\theta_0}\mathcal{L}(\theta,\theta_0,\xi,\alpha,r) = -\sum_{i=1}^{n}\alpha_i y_i = 0 \quad \Rightarrow \sum_{i=1}^{n}\alpha_i y_i = 0$$

$$\frac{\partial}{\partial\xi_i}\mathcal{L}(\theta,\theta_0,\xi,\alpha,r) = \frac{-Ce^{-\xi_i}}{1+e^{-\xi_i}} - \sum_{i=1}^{n} r_i - \sum_{i=1}^{n}\alpha_i = 0$$

$$r_i + \alpha_i = -\frac{Ce^{-\xi_i}}{1+e^{-\xi_i}}$$

$\alpha_i \geq 0 \ \forall i$ (dual feasability)
$r_i \geq 0 \ \forall i$

$\alpha_i[y_i(\theta^T x_i + \theta_0) - \xi_i] = 0 \ \forall i$ (complementary slackness)
$y_i(\theta^T x_i + \theta_0) - \xi_i \leq 0$ (primal feasability)

since $r_i \geq 0$ this gives new constraint

$$0 \leq \alpha_i \leq -\frac{Ce^{-\xi_i}}{1+e^{\xi_i}}$$

This becomes: $\max_{\alpha} \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,k=1}^{n}\alpha_i\alpha_k y_i y_k x_i^T x_k$ s.t. $\begin{cases} 0 \leq \alpha_i \leq \frac{-Ce^{-\xi_i}}{1+e^{\xi_i}} \\ \sum_{i=1}^{n}\alpha_i y_i = 0 \end{cases}$

2/ a) Data set has two data points: $\vec{x}_1, y_1$ where $y_1 = 1$
$\vec{x}_2, y_2$ $\quad y_2 = -1$

Maximum-margin hyper plane will be orthogonal to the shortest line connecting points $\vec{x}_1$ and $\vec{x}_2$ and will intersect the line at the half-way point such that the margin for $\vec{x}_1$ and $\vec{x}_2$ are equal as both will be support vectors.

Maximize the minimum margin:

$$\max_f \max_\gamma \gamma \quad \text{s.t. } y_i f(x_i) \geq \gamma \quad i = 1,2$$

$$\max_{\gamma,\lambda,\lambda_0} \gamma \quad \text{s.t. } y_i \frac{\lambda^T x_i + \lambda_0}{\|\lambda\|_2} \geq \gamma \quad i=1,2$$

$$\max_{\gamma,\lambda,\lambda_0} \gamma \quad \text{s.t. } y_i(\lambda^T x_i + \lambda_0) \geq \gamma \|\lambda\|_2 \quad i=1,2$$

$$\text{set } \|\lambda\|_2 = \frac{1}{\gamma}$$

$$\max_{\lambda,\lambda_0} \frac{1}{\|\lambda\|_2} \quad \text{s.t. } y_i(\lambda^T x_i + \lambda_0) \geq 1 \quad i=1,2$$

since $y_1 = 1$, $y_2 = -1$ $\Rightarrow$ constaints become $\begin{matrix} \lambda^T x_1 + \lambda_0 \geq 1 \\ \lambda^T x_2 + \lambda_0 \geq -1 \end{matrix}$

$$\min_{\lambda,\lambda_0} \frac{1}{2}\|\lambda\|_2^2 \quad \text{s.t. } \lambda^T x_1 + \lambda_0 - 1 \quad + \quad \lambda^T x_2 + \lambda_0 + 1 \geq 0$$

Lagrangian: $\mathcal{L}([\lambda,\lambda_0],\alpha) = \frac{1}{2}\sum_{j=1}^{2} \lambda_j^2 + \alpha_1(\lambda^T x_1 + \lambda_0 - 1) + \alpha_2(\lambda^T x_2 + \lambda_0 + 1)$

KKT $\Rightarrow$ $\nabla_\lambda \mathcal{L}([\lambda,\lambda_0],\alpha) = 0 = \vec{\lambda} + \alpha_1 x_1 + \alpha_2 x_2$

$\frac{\partial}{\partial \lambda_0} \mathcal{L}([\lambda,\lambda_0],\alpha) = 0 = \alpha_1 + \alpha_2 \Rightarrow \alpha_1 = -\alpha_2$ (Lagrangian Stationarity)

$\alpha_i \geq 0 \quad \forall i$ (dual feasibility)

$$0 = \vec{\lambda} + \alpha_1 x_1 - \alpha_1 x_2$$
$$= \vec{\lambda} + \alpha_1(x_1 - x_2)$$
$$\vec{\lambda} = -\alpha_1(x_1 - x_2)$$
$$\vec{\lambda}_0 = 1 - \lambda^T \vec{x}_1 = 1 - \lambda^T \vec{x}_2$$

2) b) Finding max-margin hyperplane:
Primal problem:

$$\min_{\lambda,\lambda_0} \frac{1}{2}\|\lambda\|_2^2 \quad \text{s.t. } y_i(\lambda^T x_i + \lambda_0) - 1 \geq 0 \quad i = 1 \cdots n$$

For this to be a convex optimization problem the objective function $\left(\min_{\lambda,\lambda_0} \frac{1}{2}\|\lambda\|_2^2\right)$ must be convex as well as the constraint $\left(y_i(\lambda^T x_i + \lambda_0) - 1 \geq 0 \quad i = 1 \cdots n\right)$.

Objective $f = \frac{1}{2}\|\lambda\|_2^2$ is convex if for any $x, z \in F$ and $\theta \in [0,1]$ we have:

$$f(\theta x + (1-\theta)z) \leq \theta f(x) + (1-\theta)g(z)$$
$$\theta f(x) + (1-\theta)g(z) \geq f(\theta x + (1-\theta)z)$$
$$\theta \cancel{\tfrac{1}{2}}\|x\|_2^2 + (1-\theta)\cancel{\tfrac{1}{2}}\|z\|_2^2 \geq \cancel{\tfrac{1}{2}}\|\theta x + (1-\theta)z\|_2^2$$
$$\sum_{i=1}^n \theta x_i^2 + \sum_{i=1}^n (1-\theta)z_i^2 \geq \sum_{i=1}^n (\theta x_i + (1-\theta)z_i)^2$$
$$\theta x^2 + (1-\theta)z^2 \geq ((\theta x) + (1-\theta)z)^2$$
$$\theta x^2 + z^2 - \theta z^2 \geq \theta^2 x^2 + 2\theta(1-\theta)xz + (1-\theta)^2 z^2$$
$$\theta x^2 + z^2 - \theta z^2 \geq \theta^2 x^2 + 2\theta(1-\theta)xz + z^2 - 2\theta z^2 + \theta^2 z^2$$
$$0 \geq (\theta^2 - \theta)x^2 + 2\theta(1-\theta)xz - \theta z^2 + \theta^2 z^2$$
$$0 \geq (\theta^2 - \theta)x^2 + 2\theta(1-\theta)xz + (\theta^2 - \theta)z^2$$
$$0 \geq (\theta^2 - \theta)x^2 + 2(\theta - \theta^2)xz + (\theta^2 - \theta)z^2$$
$$0 \geq (\theta^2 - \theta)x^2 - 2(\theta^2 - \theta)xz + (\theta^2 - \theta)z^2$$
$$0 \geq (\theta^2 - \theta)(x^2 - 2xz + z^2)$$
$$0 \leq (\theta - \theta^2)(x - z)^2$$
$$0 \leq \theta(1-\theta)(x-z)^2$$

$\theta(1-\theta) \geq 0$ because $\theta \in [0,1]$

$(x-z)^2 \geq 0$ So $\frac{1}{2}\|\lambda\|_2^2$ is convex.

2/B) • Also the constraint $y_i(\lambda^T x_i + \lambda_0) - 1 \geq 0 \quad i = 1 \cdots n$ is simply an affine function. Affine functions are convex and concave.

3/ Please see matlab code and outputs for Accuracy, ROC curves, and AUC.

Hard Margin Matlab SVM functions created:

```
function [ lambda, lambda_0 ] = train_svm( x_train, y_train )
%Train Hard-margin SVM train fxn
% Set up Solver
n = size(x_train,1);
X = zeros(size(n));
for i = 1:n
    for j = 1:n
        X(i,j) = x_train(i,:)*x_train(j,:)';
    end
end

Y = y_train*y_train';
I = eye(n);
zero = zeros(1,n);
D = Y.*X;

% Solve for alpha
H = D;
f = ones(size(zero));
A = I;
b = zero;
Aeq = Y;
beq = zero;
lb = zero;
ub = f*10;
alpha = quadprog(H,-f,-A,b,Aeq,beq);

% Find Support Vectors and Determine Lambda and Lambda_0
sv_index = alpha>1e-5;
sv_index1 = find(sv_index);
alpha = alpha.*sv_index;

lambda = zeros(1,size(x_train,2));
for i = 1:size(y_train,1)
    lambda = lambda + (alpha(i)*y_train(i)*x_train(i,:));
end
lambda_0 = 1 - lambda*x_train(sv_index1(1),:)';



end
```

```
function [ y_pred ] = predict_svm( lambda,lambda_0,x_test )
%Predict_svm Predict the output of hard-margin SVM

y_pred = (lambda*x_test' + lambda_0)';
index = y_pred>0;
y_pred(index) = 1;
index = y_pred<0;
y_pred(index) = -1;


end
```

Table of Contents

Part A

Hard-margin SVM

```
clear; clc; close all;
% Load data
load('fisheriris.mat')
x = meas(1:100,:);
y = zeros(100,1);
y(1:50) = 1;
y(51:end) = -1;

% Random Permutation then split into tranining and testing
rand_index = randperm(size(y,1));
x_shuffle = x(rand_index,:);
y_shuffle = y(rand_index);

train_size = 9*round(size(y,1)/10);
x_train = x_shuffle(1:train_size,:);
x_test = x_shuffle(train_size+1:end,:);
y_train = y_shuffle(1:train_size);
y_test = y_shuffle(train_size+1:end);

% Train SVM
[lambda, lambda_0] = train_svm(x_train,y_train);
% Predict SVM
[ y_pred ] = predict_svm( lambda,lambda_0,x_test );
```

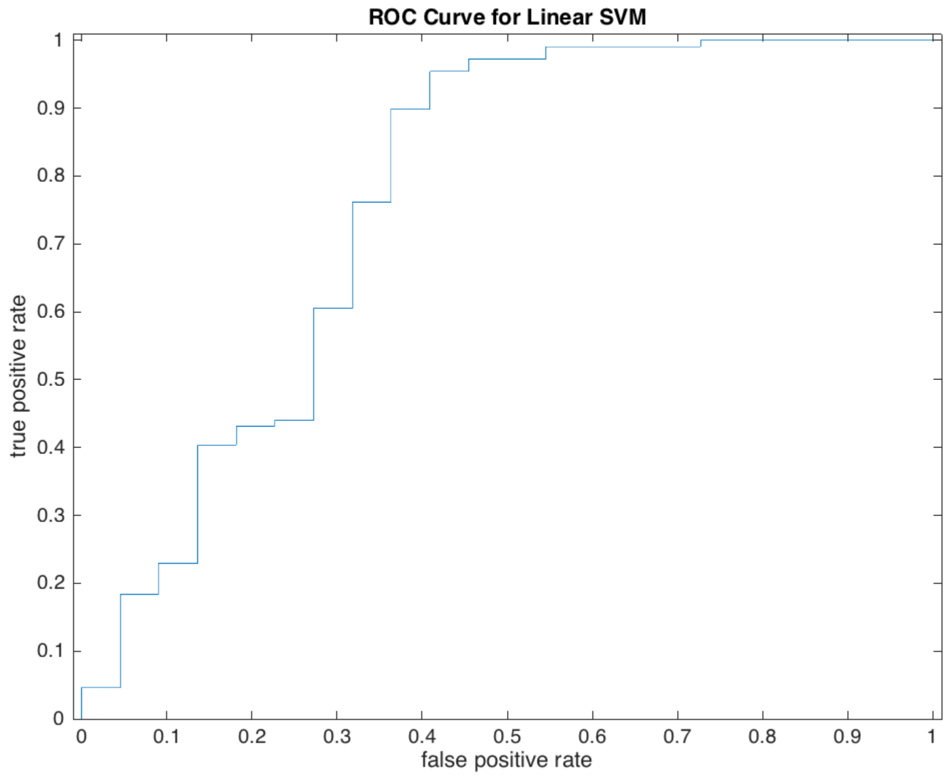
```
Minimum found that satisfies the constraints.

Optimization completed because the objective function is non-
decreasing in
feasible directions, to within the default value of the function
 tolerance,
and constraints are satisfied to within the default value of the
 constraint tolerance.
```

Part B SVM w/ k(x,z) = x'z on credit card data

```
clear; clc;
```

```
load('creditCard.mat')
data = double(creditCard);
x = data(:,1:9);
y = data(:,10);
index = (y==0);
y(index) = -1;

% Random Permutation then split into tranining and testing
rand_index = randperm(size(y,1));
x_shuffle = x(rand_index,:);
y_shuffle = y(rand_index);

train_size = 9*round(size(y,1)/10);
x_train = x_shuffle(1:train_size,:);
x_test = x_shuffle(train_size+1:end,:);
y_train = y_shuffle(1:train_size);
y_test = y_shuffle(train_size+1:end);

% Train Linear SVM
K = 'linear';
svmModel_linear = fitcsvm(x_train, y_train, 'Standardize',
 true, 'KernelFunction', K);
svmModel_linear = fitPosterior(svmModel_linear);
[~, Yscores] = predict(svmModel_linear, x_test);

y_predict = double(Yscores(:,2)>0.5);
index = (y_predict==0);
y_predict(index) = -1;
accuracy = (sum(abs(y_predict == y_test))/length(y_test))*100;

% Compute the standard ROC curve and the AUROC
[Xsvm, Ysvm, Tsvm, AUCsvm] = perfcurve(y_test, Yscores(:, 2), 1);

fprintf('Classifcation Results for Linear SVM... \n')
figure;
plot(Xsvm, Ysvm,'-')
xlabel('false positive rate');
ylabel('true positive rate');
axis([-0.01,1.01,0,1.01])
title('ROC Curve for Linear SVM')
fprintf('Classifcation Accuracy on Test Set: %f %% \n', accuracy);
fprintf('AUC : %f \n', AUCsvm);
```

```
Classifcation Results for Linear SVM...
Classifcation Accuracy on Test Set: 90.076336 %
AUC : 0.766472
```

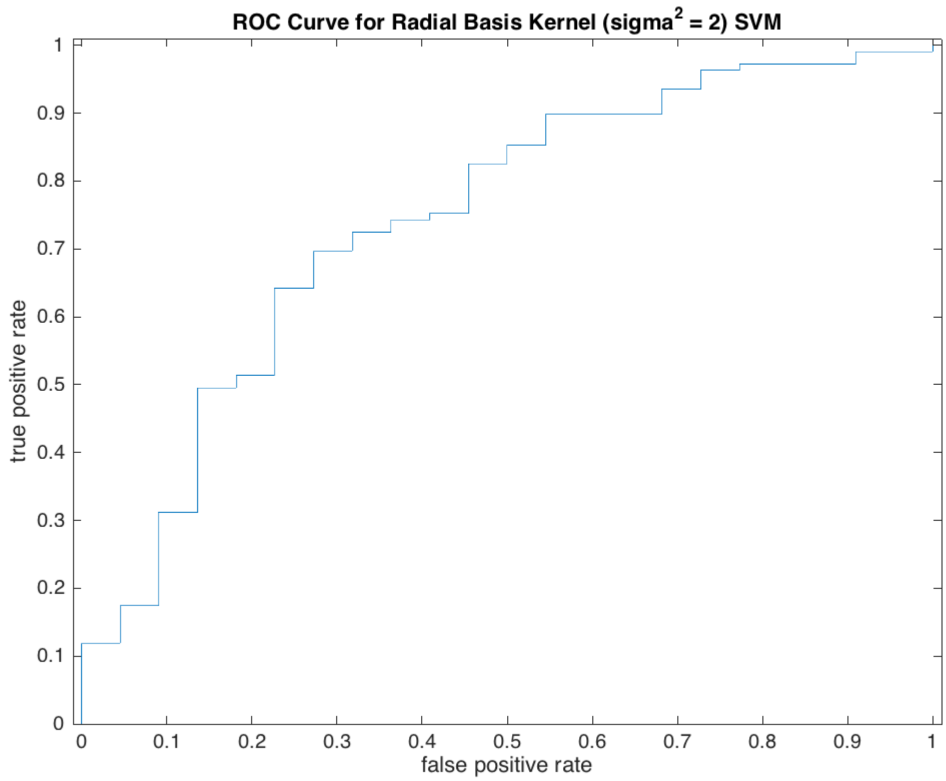


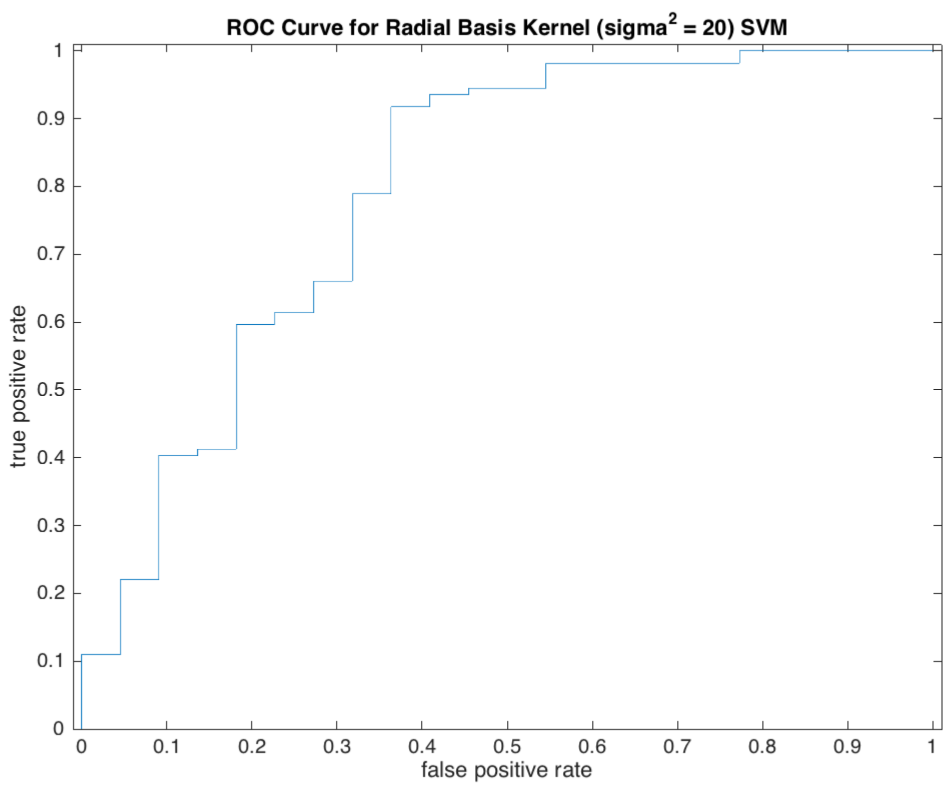
Part C SVM w/ radial basis kernel on credit card data

```
% Sigma^2 = 2

% Train RBF SVM
K = 'rbf';
svmModel_rbf_2 = fitcsvm(x_train, y_train, 'Standardize',
 true, 'KernelFunction', K,'KernelScale',sqrt(2));
svmModel_rbf_2 = fitPosterior(svmModel_rbf_2);
[~, Yscores] = predict(svmModel_rbf_2, x_test);

y_predict = double(Yscores(:,2)>0.5);
index = (y_predict==0);
y_predict(index) = -1;
accuracy = (sum(abs(y_predict == y_test))/length(y_test))*100;

% Compute the standard ROC curve and the AUROC
[Xsvm, Ysvm, Tsvm, AUCsvm] = perfcurve(y_test, Yscores(:, 2), 1);

fprintf('Classifcation Results for Radial Basis Kernel (sigma^2 = 2)
 SVM... \n')
figure;
plot(Xsvm, Ysvm,'-')
```

```
xlabel('false positive rate');
ylabel('true positive rate');
axis([-0.01,1.01,0,1.01])
title('ROC Curve for Radial Basis Kernel (sigma^2 = 2) SVM')
fprintf('Classifcation Accuracy on Test Set: %f %% \n', accuracy);
fprintf('AUC : %f \n', AUCsvm);

% Sigma^2 = 20

% Train RBF SVM
K = 'rbf';
svmModel_rbf_20 = fitcsvm(x_train, y_train, 'Standardize',
 true, 'KernelFunction', K,'KernelScale',sqrt(20));
svmModel_rbf_20 = fitPosterior(svmModel_rbf_20);
[~, Yscores] = predict(svmModel_rbf_20, x_test);

y_predict = double(Yscores(:,2)>0.5);
index = (y_predict==0);
y_predict(index) = -1;
accuracy = (sum(abs(y_predict == y_test))/length(y_test))*100;

% Compute the standard ROC curve and the AUROC
[Xsvm, Ysvm, Tsvm, AUCsvm] = perfcurve(y_test, Yscores(:, 2), 1);

fprintf('Classifcation Results for Radial Basis Kernel (sigma^2 = 20)
 SVM... \n')
figure;
plot(Xsvm, Ysvm,'-')
xlabel('false positive rate');
ylabel('true positive rate');
axis([-0.01,1.01,0,1.01])
title('ROC Curve for Radial Basis Kernel (sigma^2 = 20) SVM')
fprintf('Classifcation Accuracy on Test Set: %f %% \n', accuracy);
fprintf('AUC : %f \n', AUCsvm);
```

```
Classifcation Results for Radial Basis Kernel (sigma^2 = 2) SVM...
Classifcation Accuracy on Test Set: 83.206107 %
AUC : 0.743119
Classifcation Results for Radial Basis Kernel (sigma^2 = 20) SVM...
Classifcation Accuracy on Test Set: 88.549618 %
AUC : 0.793578
```


ROC Curve for Radial Basis Kernel (sigma^2 = 2) SVM
true positive rate
false positive rate
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


ROC Curve for Radial Basis Kernel (sigma^2 = 20) SVM
true positive rate
false positive rate
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

Published with MATLAB® R2015b