Statistical Machine Learning
Fall 2016, Homework 1
(due on Sep 8, 11.59pm EST)

Jean Honorio jhonorio@purdue.edu

The homework is based on a total of 10 points. Please read submission instructions at the end. **Failure to comply to submission instructions will cause your grade to be reduced.**

In this homework, we will focus on classification for separable data. Your code **should be in MATLAB**. The part of your code that deals with kernelized versions of the problems, should call the following function **K.m**

```
% Input: vector x of d rows, 1 column
%        vector xp of d rows, 1 column
% Output: kernel K(x,xp) = exp(-1/2 * norm(x-xp)^2)
% Example on how to call the function: v = K([1; 4; 3],[2; 5; -1]);
function v = K(x,xp)

v = exp(-1/2 * sum((x-xp).^2));
```

You can use the following function **createsepdata.m** to create some synthetic separable data:

```
% Input: number of samples n
%        number of features d
% Output: matrix X of features, with n rows (samples), d columns (features)
%             X(i,j) is the j-th feature of the i-th sample
%         vector y of labels, with n rows (samples), 1 column
%             y(i) is the label (+1 or -1) of the i-th sample
% Example on how to call the function: [X y] = createsepdata(10,3);
function [X y] = createsepdata(n,d)

y = ones(n,1);
y(ceil(n/2)+1:end) = -1;
X = rand(n,d);
X(y==1,1) = 0.1+X(y==1,1);
X(y==-1,1) = -0.1-X(y==-1,1);
U = orth(rand(d));
X = X*U;
```

Here are the questions:

1) [2.5 points] Implement the following perceptron algorithm, introduced in Lecture 1.

Input: number of iterations L, training data $x_t \in \mathbb{R}^d$, $y_t \in \{+1, -1\}$ for $t = 1, \ldots, n$
Output: $\theta \in \mathbb{R}^d$
$\theta \leftarrow 0$
for iter $= 1, \ldots, L$ **do**
  for $t = 1, \ldots, n$ **do**
    if $y_t(\theta \cdot x_t) \leq 0$ **then**
      $\theta \leftarrow \theta + y_t x_t$
    end if
  end for
end for

The header of your **MATLAB function linperceptron.m** should be:

```
% Input: number of iterations L
%        matrix X of features, with n rows (samples), d columns (features)
%            X(i,j) is the j-th feature of the i-th sample
%        vector y of labels, with n rows (samples), 1 column
%            y(i) is the label (+1 or -1) of the i-th sample
% Output: vector theta of d rows, 1 column
function theta = linperceptron(L,X,y)
```

2) [1 point] Implement the following linear predictor function, introduced in Lecture 1.

Input: $\theta \in \mathbb{R}^d$, testing point $x \in \mathbb{R}^d$
Output: label $\in \{+1, -1\}$
if $\theta \cdot x > 0$ **then**
  label $\leftarrow +1$
else
  label $\leftarrow -1$
end if

The header of your **MATLAB function linpred.m** should be:

```
% Input: vector theta of d rows, 1 column
%        vector x of d rows, 1 column
% Output: label (+1 or -1)
function label = linpred(theta,x)
```

3) [2.5 points] Implement the following perceptron algorithm with kernels, introduced in Lecture 3.

Input: number of iterations L, training data $x_t \in \mathbb{R}^d$, $y_t \in \{+1, -1\}$ for $t = 1, \ldots, n$

Output: $\alpha \in \mathbb{R}^n$
$\alpha \leftarrow 0$
for iter $= 1, \ldots, L$ **do**
  for $t = 1, \ldots, n$ **do**
    if $y_t(\sum_{i=1}^n \alpha_i y_i K(x_i, x_t)) \leq 0$ **then**
      $\alpha_t \leftarrow \alpha_t + 1$
    end if
  end for
end for

The header of your **MATLAB function kerperceptron.m** should be:

```
% Input: number of iterations L
%        matrix X of features, with n rows (samples), d columns (features)
%            X(i,j) is the j-th feature of the i-th sample
%        vector y of labels, with n rows (samples), 1 column
%            y(i) is the label (+1 or -1) of the i-th sample
% Output: vector alpha of n rows, 1 column
function alpha = kerperceptron(L,X,y)
```

4) [1.5 points] Implement the following linear predictor function, introduced in Lecture 3.

Input: $\alpha \in \mathbb{R}^n$, training data $x_t \in \mathbb{R}^d$, $y_t \in \{+1, -1\}$ for $t = 1, \ldots, n$, testing point $x \in \mathbb{R}^d$
Output: label $\in \{+1, -1\}$
if $\sum_{i=1}^n \alpha_i y_i K(x_i, x) > 0$ **then**
  label $\leftarrow +1$
else
  label $\leftarrow -1$
end if

The header of your **MATLAB function kerpred.m** should be:

```
% Input: vector alpha of n rows, 1 column
%        matrix X of features, with n rows (samples), d columns (features)
%            X(i,j) is the j-th feature of the i-th sample
%        vector y of labels, with n rows (samples), 1 column
%            y(i) is the label (+1 or -1) of the i-th sample
%        vector x of d rows, 1 column
% Output: label (+1 or -1)
function label = kerpred(alpha,X,y,x)
```

5) [2.5 points] Now we ask you to implement the following dual support vector

machines (DSVM) problem, introduced in Lecture 4.

$$\begin{aligned} &\text{maximize } \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \\ &\text{subject to } \alpha_i \geq 0 \text{ for } i = 1, \dots, n \end{aligned}$$

Let $f = (1, 1, \dots, 1)^{\mathrm{T}} \in \mathbb{R}^n$ be an n-dimensional vector of ones. Let $z = (0, 0, \dots, 0)^{\mathrm{T}} \in \mathbb{R}^n$ be an n-dimensional vector of zeros. Let $H \in \mathbb{R}^{n \times n}$ be a matrix with n rows and n columns, where $h_{i,j} = y_i y_j K(x_i, x_j)$ for all $i, j = 1, \dots, n$. Since $\alpha \in \mathbb{R}^n$, we can rewrite the DSVM problem as:

$$\begin{aligned} &\text{minimize } \frac{1}{2} \alpha^{\mathrm{T}} H \alpha - f^{\mathrm{T}} \alpha \\ &\text{subject to } \alpha \geq z \end{aligned}$$

Fortunately, the standard MATLAB function **quadprog.m** can solve exactly the above problem by doing: **alpha = quadprog(H,-f,[ ],[ ],[ ],[ ],z);** The header of your **MATLAB function kerdualsvm.m** should be:

```
% Input: matrix X of features, with n rows (samples), d columns (features)
%            X(i,j) is the j-th feature of the i-th sample
%        vector y of labels, with n rows (samples), 1 column
%            y(i) is the label (+1 or -1) of the i-th sample
% Output: vector alpha of n rows, 1 column
function alpha = kerdualsvm(X,y)
```

Notice that for prediction you can reuse the **kerpred.m** function that you wrote for question 4.

Submission: Please, submit a single ZIP file **through Blackboard**. Your MATLAB code (**linperceptron.m**, **linpred.m**, etc.) should be directly inside the ZIP file. **There should not be any folder inside the ZIP file**, just MATLAB code. The ZIP file should be named by the first letter of your first name followed by your last name. For instance, for Jean Honorio, the ZIP file should be named **jhonorio.zip**