Statistical Machine Learning
Spring 2020, Homework 4
(due on March 31, 11.59pm EST)

Jean Honorio jhonorio@purdue.edu

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

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) [3.5 points] Implement the learning part of the following probabilistic classifier, introduced in Lecture 12.

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

Output: proportion of positive samples $q \in \mathbb{R}$, mean vector of the positive class $\mu_{+1} \in \mathbb{R}^d$, mean vector of the negative class $\mu_{-1} \in \mathbb{R}^d$, variance of the

positive class $\sigma^2_{+1} \in \mathbb{R}$, variance of the negative class $\sigma^2_{-1} \in \mathbb{R}$

(* Comment: In the following pseudocode, k_{+1} will be the number of positive samples, k_{-1} will be the number of negative samples *)

$k_{+1} \leftarrow 0$
$k_{-1} \leftarrow 0$
$\mu_{+1} \leftarrow 0$
$\mu_{-1} \leftarrow 0$
for $t = 1, \ldots, n$ **do**
  if $y_t = +1$ **then**
    $k_{+1} \leftarrow k_{+1} + 1$
    $\mu_{+1} \leftarrow \mu_{+1} + x_t$
  else
    $k_{-1} \leftarrow k_{-1} + 1$
    $\mu_{-1} \leftarrow \mu_{-1} + x_t$
  end if
end for
$q \leftarrow k_{+1}/n$
$\mu_{+1} \leftarrow (1/k_{+1})\,\mu_{+1}$
$\mu_{-1} \leftarrow (1/k_{-1})\,\mu_{-1}$
$\sigma^2_{+1} \leftarrow 0$
$\sigma^2_{-1} \leftarrow 0$
for $t = 1, \ldots, n$ **do**
  if $y_t = +1$ **then**
    $\sigma^2_{+1} \leftarrow \sigma^2_{+1} + \left\|x_t - \mu_{+1}\right\|^2$
  else
    $\sigma^2_{-1} \leftarrow \sigma^2_{-1} + \left\|x_t - \mu_{-1}\right\|^2$
  end if
end for
$\sigma^2_{+1} \leftarrow (1/(d\,k_{+1}))\,\sigma^2_{+1}$
$\sigma^2_{-1} \leftarrow (1/(d\,k_{-1}))\,\sigma^2_{-1}$

The header of your **MATLAB function probclearn.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: scalar q
%         vector mu_pos of d rows, 1 column
%         vector mu_neg of d rows, 1 column
%         scalar sigma2_pos
%         scalar sigma2_neg
function [q, mu_pos, mu_neg, sigma2_pos, sigma2_neg] = probclearn(X,y)
```

2) [1.5 points] Implement the prediction part of the following probabilistic classifier, introduced in Lecture 12.

Input: proportion of positive samples $q \in \mathbb{R}$, mean vector of the positive class $\mu_{+1} \in \mathbb{R}^d$, mean vector of the negative class $\mu_{-1} \in \mathbb{R}^d$, variance of the positive class $\sigma^2_{+1} \in \mathbb{R}$, variance of the negative class $\sigma^2_{-1} \in \mathbb{R}$, testing point $x \in \mathbb{R}^d$

Output: label $\in \{+1, -1\}$

if $\log\left(\frac{q}{1-q}\right) - \frac{d}{2}\log\left(\frac{\sigma^2_{+1}}{\sigma^2_{-1}}\right) - \frac{1}{2\sigma^2_{+1}}\left\|x - \mu_{+1}\right\|^2 + \frac{1}{2\sigma^2_{-1}}\left\|x - \mu_{-1}\right\|^2 > 0$ **then**

  label $\leftarrow +1$

else

  label $\leftarrow -1$

end if

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

```
% Input: scalar q
%        vector mu_pos of d rows, 1 column
%        vector mu_neg of d rows, 1 column
%        scalar sigma2_pos
%        scalar sigma2_neg
%        vector x of d rows, 1 column
% Output: label (+1 or -1)
function label = probcpredict(q,mu_pos,mu_neg,sigma2_pos,sigma2_neg,x)
```

3) [3.5 points] Implement the learning part of principal component analysis (PCA), introduced in Lecture 15. Let $X \in \mathbb{R}^{n \times d}$ be the *training* data matrix for n samples and d features. PCA maps each sample from d dimensions to $F \in \{1, \ldots, \min(n, d)\}$ dimensions, thus we can express the projection as a matrix $Z \in \mathbb{R}^{d \times F}$.

Input: number of features F, *training* data matrix $X \in \mathbb{R}^{n \times d}$

Output: average $\mu \in \mathbb{R}^d$, principal components $Z \in \mathbb{R}^{d \times F}$

for $i = 1, \ldots, d$ **do**

  $\mu_i \leftarrow \frac{1}{n}\sum_{t=1}^{n} x_{ti}$

end for

for $t = 1, \ldots, n$ **do**

  for $i = 1, \ldots, d$ **do**

    $x_{ti} \leftarrow x_{ti} - \mu_i$

  end for

end for

Let $U \in \mathbb{R}^{n \times \min(n,d)}$, $D \in \mathbb{R}^{\min(n,d) \times \min(n,d)}$, $V \in \mathbb{R}^{d \times \min(n,d)}$ be the singular value decomposition of X, i.e., $X = UDV^{\mathrm{T}}$ where $U^{\mathrm{T}}U = I$, $V^{\mathrm{T}}V = I$ and D is a diagonal matrix

$E \leftarrow$ first F rows and columns of D, i.e., $E \in \mathbb{R}^{F \times F}$

$W \leftarrow$ first F columns of V, i.e., $W \in \mathbb{R}^{d \times F}$
$Z \leftarrow \sqrt{n}\, W E^{-1}$

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

```
% Input: number of features F
%        data matrix X, with n rows (samples), d columns (features)
% Output: average mu, with d rows, 1 column
%         principal component matrix Z, with d rows, F columns
function [mu Z] = pcalearn(F,X)
```

4) [1.5 points] Implement the projection part of principal component analysis (PCA), introduced in Lecture 15.

Input: *test* data matrix $X \in \mathbb{R}^{n \times d}$, average $\mu \in \mathbb{R}^d$, principal components $Z \in \mathbb{R}^{d \times F}$
Output: projected data matrix $P \in \mathbb{R}^{n \times F}$
for $t = 1, \ldots, n$ **do**
  for $i = 1, \ldots, d$ **do**
    $x_{ti} \leftarrow x_{ti} - \mu_i$
  end for
end for
$P \leftarrow XZ$

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

```
% Input: number of features F
%        data matrix X, with n rows (samples), d columns (features)
%        average mu, with d rows, 1 column
%        principal component matrix Z, with d rows, F columns
% Output: projected data matrix P, with n rows, F columns
function P = pcaproj(X,mu,Z)
```

Submission: Please, submit a single ZIP file **through Blackboard**. Your MATLAB code (**probclearn.m**, **probcpredict.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**