Statistical Machine Learning
Fall 2016, Homework 3
(due on Nov 22, 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 function **createlinregdata.m** to create some synthetic linear regression 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 scalar values, with n rows (samples), 1 column
%             y(i) is the scalar value of the i-th sample
% Example on how to call the function: [X y] = createlinregdata(10,2);
function [X y] = createlinregdata(n,d)

w = 2*rand(d,1)-1;
w = w/norm(w);
X = randn(n,d);
y = X*w + 0.25*randn(n,1);
```

Additionally, use the following way to solve the linear regression problem, with training data $x_t \in \mathbb{R}^d$, $y_t \in \mathbb{R}$ for $t = 1, \ldots, n$.

$$\widehat{\theta} \leftarrow \arg\min_{\beta \in \mathbb{R}^d} \frac{1}{n} \sum_{t=1}^{n} (y_t - \beta \cdot x_t)^2$$

If $n > d$, a solution to the above is given by the following function **linreg.m**:

```
% 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 scalar values, with n rows (samples), 1 column
%            y(i) is the scalar value of the i-th sample
% Output: vector theta, with d rows, 1 column
function theta = linreg(X,y)

theta = pinv(X)*y;
```

Here are the questions:

1) [3 points] Implement k-fold cross validation (Lecture 15) with linear regression. (The function $\lfloor w \rfloor$ denotes the largest integer less than or equal to $w \in \mathbb{R}$, i.e., the "floor" function.)

Input: number of folds k, data $x_t \in \mathbb{R}^d$, $y_t \in \mathbb{R}$ for $t = 1, \ldots, n$
Output: mean square error $z \in \mathbb{R}^k$
for $i = 1, \ldots, k$ **do**
$T \leftarrow \{\lfloor n(i-1)/k \rfloor + 1, \ldots, \lfloor n\, i/k \rfloor\}$
$S \leftarrow \{1, \ldots, n\} - T$
$\widehat{\theta} \leftarrow \arg\min_{\beta \in \mathbb{R}^d} \frac{1}{|S|} \sum_{t \in S} (y_t - \beta \cdot x_t)^2$
$z_i \leftarrow \frac{1}{|T|} \sum_{t \in T} (y_t - \widehat{\theta} \cdot x_t)^2$
end for

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

```
% Input: number of folds k
%        matrix X of features, with n rows (samples), d columns (features)
%        vector y of scalar values, with n rows (samples), 1 column
% Output: vector z of k rows, 1 column
function z = kfoldcv(k,X,y)
```

2) [3 points] Implement bootstrapping (Lecture 15) with linear regression.

Input: number of bootstraps B, data $x_t \in \mathbb{R}^d$, $y_t \in \mathbb{R}$ for $t = 1, \ldots, n$
Output: mean square error $z \in \mathbb{R}^B$
for $i = 1, \ldots, B$ **do**
$u \leftarrow (0, \ldots, 0)$ (an array of n zeros)
$S \leftarrow$ emptyset
for $j = 1, \ldots, n$ **do**
choose k uniformly at random from $\{1, \ldots, n\}$
$u_j \leftarrow k$ (repeated elements are allowed in the array u)
$S \leftarrow S \cup \{k\}$ (repeated elements are not allowed in the set S)
end for
$T \leftarrow \{1, \ldots, n\} - S$ (repeated elements are not allowed in the set T)
$\widehat{\theta} \leftarrow \arg\min_{\beta \in \mathbb{R}^d} \frac{1}{n} \sum_{j=1}^{n} (y_{u_j} - \beta \cdot x_{u_j})^2$
$z_i \leftarrow \frac{1}{|T|} \sum_{t \in T} (y_t - \widehat{\theta} \cdot x_t)^2$
end for

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

```
% Input: number of bootstraps B
%        matrix X of features, with n rows (samples), d columns (features)
%        vector y of scalar values, with n rows (samples), 1 column
% Output: vector z of B rows, 1 column
function z = bootstrapping(B,X,y)
```

3) [3 points] Implement the learning part of principal component analysis (PCA), introduced in Lecture 16. Let $X \in \mathbb{R}^{n \times d}$ be the 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, 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., $W \in \mathbb{R}^{d \times F}$
$Z \leftarrow \sqrt{n}\, WE^{-1}$

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 point] Implement the projection part of principal component analysis (PCA), introduced in Lecture 16.

Input: 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 (**kfoldcv.m**, **bootstrapping.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**