Machine Learning, Spring 2018
Project 2: Classification
Due Date: April 25, 2018

Overview.
The goal of this project is to do Classification of Patterns. You will use the following kernel-based algorithms:

- Relevance Vector Machines (RVM)
- Support Vector Machines (SVM)
- Gaussian Process Regression (GPR)

SVM is a two-class classifier that can be trained using the one-vs-all or all-pairs methods. I recommend the latter method. For a fair comparison, train the RVM and GPR using the same method.

Definitions:

Samples:	Feature Vectors of dimension D. In MATLAB, an N x D array of doubles. In Python, and N x D NumPy array of doubles.
N	Number of Samples
Nc:	Number of classes
ClassLabels:	An (Nc+1) x 1 vector y with values yc \in [0,1] representing the possibility of class c
Training Set:	A set of Samples used to estimate parameters and hyper-parameters of classifiers
Estimation Set:	A set of Samples used to estimate parameters of classifiers
Validation Set:	A set of Samples used to estimate hyper-parameters of classifiers and determine when to stop training
Fold	A Partition of a Training Set into an Estimation and Validation Set
N-Fold Cross Validation	See (2) in the description of the programs to be written
One-vs-all training	Denote the classes by C1, C2, …,CG. For each k = 1,2,…,G, train a two-class classifier using the two classes Ck and $\cup_{m \neq k} C_m$
All-pairs training	Train one two-class classifier, F_{km} for each pair of classes, (Ck,Cm), yielding G choose 2, or $\frac{G!}{(G-2)!2!}$, classifiers.
All-pairs testing	Calculate $y_{km} = F_{km}(\mathbf{x})$ for each pair of classes. Calculate $z_k = \frac{1}{G}\sum_{m \neq k} y_{km}$. Calculate z_{G+1} however you wish. The ClassLabels are the vector $(z_1, z_2, \ldots, z_G, z_{G+1})^t$
Test Set:	A set of Samples used to test classifiers
Confusion Matrix	A matrix A =(a_{mn}) where $a_{mn} = \frac{(\text{\# of samples from class m classified as class n})}{(\text{\# of samples from class m})}$

Data.
Data Set 1:
25,000 samples of 60-dimensional feature vectors from 5 classes, 5000 samples per class.
25,000 Target Outputs, which are 5 dimensional vectors containing exactly one 1 and four -1's.

Code to be written.

(1) Write a Program called "TrainMyClassifier" that has inputs

XEstimate
XValidate
Parameters

and estimates parameters and hyper-parameters using XEstimate and XValidate; and produces outputs:

Yvalidate	Class Labels on the Validation Set
EstParameters	All the estimated Parameters and Hyper-Parameters

(2) Write a program called "MyCrossValidate" that has inputs

XTrain	
Nf	The Number of Folds

that does the following:

Step 1: Randomly partition the data into Nf pairs of Estimation and Validation Sets, E_n and V_n, with $E_n \cap V_n = \phi,\ E_n \cup V_n = XTrain,\ \bigcup_{n=1}^{Nf} V_n = XTrain,$ and $V_n \cap V_m = \phi$ if $n \neq m$.

Step 2: Estimate parameters and hyper-parameters using E_n and V_n.
Step 3. Produce a confusion matrix, C_n, for each V_n.
Step 4: Produce a confusion matrix for all of XTrain using all the class labels.

The outputs should be:

Ytrain	The class labels for each validation sample
EstParameters:	An array of Estimated Parameters for each Vn
EstConfMatrices:	An array of Confusion Matrices for each Vn
ConfMatrix:	The overall Confusion Matrix

(3) Write a program called "TestMyClassifier" that has inputs (the latter two from the above programs)

XTest
Parameters
EstParameters

and produces an output

Ytest Class Labels for each sample in XTest

(4) Finally, write a program called "MyConfusionMatrix" that has inputs

Y
ClassNames

that prints a confusion matrix to the screen and returns

Confusion matrix	(with no names, just numbers)
Average accuracy	1 number

Submission:

You should submit a report showing your best results using 5-fold cross-validation. You should include overall confusion matrices for each classifier. You should include the number of Support Vectors and the number of Relevance Vectors for each fold. You may include up to one page of observations using 12 pt, double-spaced font.

You should submit your code and report by emailing them to me with the exact phrase "CAP6610Project2TeamXXX" as the subject line, where XXX is your team number. The code should be zipped into a file called "CAP6610Project2TeamXXXCode.zip". A readme.txt file should be included with the code.

The report should be a .pdf file called "CAP6610Project2TeamXXXReport.pdf".

You should hand in a report showing your best results.

Evaluation:

You will be evaluated based on

A. Your code should run.
B. Your code should get comparable results to mine on given data
C. Your code should get comparable results to mine on unseen data
D. Your code should be able to accurately identify input samples that are not from any class
E. Your code should be well-documented
F. Your report should be concise, insightful, and meet the requirements.