EECE 5644 Machine Learning and Pattern Recognition Fall, 2019
Exam 2

Directions: This exam consists of 3 problems. The first two problems are required and the third problem is optional. The first two problems have equal weight, divided equally among each subproblem. The score for the third problem will be added to your score for the midterm exam and does not count towards the score for exam 2. Please follow directions carefully and submit your solutions as an unzipped PDF before the due date. Do not submit code, but submit a listing carefully included in the PDF.

NEU's academic honesty policy will be strictly enforced for this take-home exam. Submit only your own work for this exam and do not discuss the exam with other students. Direct your questions to the instructor only during the announced office hours.

Question 1
The data in file Q1.csv contains 2 real-valued measurements x_i, i=1,2 and a class label y=+-1 as rows in comma separated values format. In this problem you will use decision trees with bagging and boosting to create classifiers.

a. Provide a scatter plot of the entire data set, using the marker 'o' and color 'red' for class label -1 and marker 'x' and color 'black' for class label 1. Carefully label the axes and provide a grid.

b. Set aside the first 10 percent of the available data for testing, and use the remainder for training. Create a decision tree with less than 12 nodes (11 splits) using the ID3 algorithm as presented in class. Use either the Gini or entropy metric for a measure of subpopulation purity. You may stop extending the tree when every leaf subpopulation is either empty or has an impurity of 1% or less, or when a total of 11 splits have been performed. Carefully present your tree structure labelling questions in each split box, 'True' branches to the left and 'False' branches to the right. Evaluate and present the confusion matrix for the decision tree on the test data. On a copy of your plot in 1.a, carefully place the decision boundaries for the decision tree.

c. Create a bagging decision tree solution, using 7 trees. Construct each tree using training data obtained as follows: set aside the first 10% of the data file for testing, randomly sample with replacement the remaining data to obtain a training population size equal to that in 1.b. Create each tree using the method and code from 1.b.

   Apply the test data to each of the 7 classifiers, and use the majority vote for the final decision. Evaluate and present the confusion matrix. Carefully present the decision boundaries of your bagging classifier on a copy of your plot from 1.a. Note: this will require entering artificial values of x_1 and x_2 to your classifier to determine the decision boundaries really well. Please do this.

d. Use the Adaboost algorithm with 7 levels of 12-node classifiers. Set aside the first 10% of the data for testing. Nonuniformly select with replacement from the training data as per the Adaboost adaptive weights, using a training sample size equal to that in 1.b for each level. Design the classifier for each level according to the method and code in 1.b. Your final classifier weighs each of the level decisions. Present and comment on the initial and final sample selection weights and the level weights concisely and clearly. Carefully state any assumptions you made. (Note: Slide 13 in Lecture 11.4 has been updated. Please download the latest.)

   Evaluate the boosted classifier on the test data and present the confusion matrix. Carefully present the decision boundaries of your Adaboost classifier on a copy of your plot from 1.a. Note: this will require entering artificial values of x_1 and x_2 to your classifier to accurately determine the decision boundaries.

e. Carefully and concisely compare the performance of the classifiers in 1.b, 1.c, and 1.d.

Question 2

The motion of an object in two dimensions will be tracked in discrete time for this problem using a Kalman filter. The observations will be noisy values of position.

Let the motion of an object at time nT be denoted by the column vector **x**[n]= [h(nT), v_h(nT), a_h(nT), b(nT), v_b(nT), a_b(nT)]^T. Here, h denotes longitude position and b denotes latitude position. Also, variables v_q and a_q denote velocity and acceleration in the component q. In this problem, you will estimate the vector sequence {**x**[n]}.

In the absence of model noise, h evolves as: h((n+1)T) = h(nT)+v_h(nT)T+1/2 a_h T^2. Further, the velocity evolves as v_h((n+1)T)=v_h(nT)+a_hT. Variables b(nT) and v_b(nT) evolve similarly. Note that this is a (noiseless) constant acceleration model for each component, as presented in lecture. Including model noise, the vector **x**[n] evolves as **x**[n+1]=**Ax**[n]+**w**[n] for some matrix **A**. Further, vector sequence {**w**[n]} is an iid sequence of Gaussian vectors with zero mean vector and covariance matrix K **I**, where **I** is the identity matrix and K is a positive real number.

The measurement vector sequence {**y**[n]} before index n+1 will be used to estimate **x**[n]. With measurement noise, **y**[n]=[h(nT),b(nT)]^T +**m**[n]=**Cx**[n]+**m**[n], for some matrix **C**. The measurement noise sequence {**m**[n]} is an iid sequence of Gaussian vectors with zero mean vector and covariance matrix S **I**, for some positive number S. For this problem, the measurement sequence is contained as rows in the files Q2train.csv and Q2test.csv, with the nth row containing the sampling time t (column 1) and the components of the nth measurement **y**[n]^T (columns 2 and 3). Note that the measurement times in each file are equally spaced, with the testing times offset from the training times by T/2.

1. Completely specify the matrices **A** and **C** for this problem. Let T=2.

2. Display the measurement sequence from Q2train.csv (not the measurement time) as a scatter plot in two dimensions. Carefully label the axes. Connect adjacent measurements using dotted lines so that the noisy trajectory is visualized.

3. Write Kalman filter code in the language of your choice, using the notation in this problem. Follow the description in the class notes. Comment your code clearly. Let the input to your code include the matrices from 2.1 as well as (general, invertible) noise covariance matrices. Also, allow an initial state estimate **x**[0] as an input. Make sure that an output of your code includes the estimated sequence {$\mathbf{x}_e$[n]}.

4. Write code which takes the sequence of $\mathbf{x}_e$[n], for time nT, and return an estimate the sequence { [h(t),b(t)]^T} ={[h_e(t),b_e(t)]^T}for any ordered sequence of times {t}. For each time, you may do this by "connecting the dots" given by latitude and longitude position estimates found in $\mathbf{x}_e$[n] and $\mathbf{x}_e$[n+1], where a time falls between nT and (n+1)T. You may also use other interpolation methods, but this is unnecessary.

5. Consider the pair (K,S) as hyperparameters for this problem. For each pair, evaluate your result of 2.3 on data found in Q2train.csv. Use T=2 and x[0]=[0,…0]^T. Next, create a time sequence from Q2test.csv and run your solution to 2.4 using this time sequence. For each (K,S) pair, evaluate a cross-validation metric given by the sample average of ||[h_e(t),b_e(t)]^T-**y**[n]||^2, where **y**[n] and t both come from the same row of Q2test.csv. Carefully and clearly plot and label the cross-validation metric using a contour plot. Identify the pair (K_o,S_o) which minimizes the cross-validation metric.

6. Present a display showing all (testing and training), time-ordered measurements without connections. On this display, assemble the corresponding Kalman filter outputs and {[h_e(t),b_e(t)]} with connected markers. Carefully label your axes and provide a helpful legend.

Question 3 (optional)
Return to the midterm exam, and re-do the question which had the greatest reduction in points for you. Carefully follow the instructions for that question as given in the exam.