EECE 6036: Intelligent Systems
Homework 2: Given 3/2/16; Due 3/15/16

1. (100 points) A team of scientists has been studying patients afflicted with a metabolic disorder. Medication is available for the disease, but has serious side-effects. Thus, it is important to screen patients so that only those who really need it get it. The only known test, unfortunately, is very expensive, but it is thought that some of the genes implicated in the disease are also involved in the regulation of blood pressure and blood sodium levels, so there is hope that it may be possible to screen subjects based on these two easily measured quantities. A database of sodium levels, (L) and blood pressure (P) has been compiled, with patients labeled as positive (D=1) or negative (D=0) for the disease based on the expensive test. Your task is to design a simple classifier that can classify future subjects with unknown labels correctly.

You will try two different approaches to the task:

1. An aumented k-nearest neighbors classifier.
2. A single perceptron.

The augmented k-NN classifier will use one or more heuristics, such as using a customized k value for each point based on some criterion (e.g., at least 60% of neighbors in the winning class), or using distance weights. The choice of augmentation is up to you. You can think of it yourself or search in the literature for options.

For the perceptron, you will need to specify a learning rate, choose a policy for initializing the weights, and decide how long you will train (i.e., how many epochs).

You should also think about the numerical ranges of L and P data to see whether any prior scaling is needed.

After some trial runs to establish your initialization policies and parameters (e.g., k, learning rate, number of training steps, etc.), you will do the following:

Keeping the parameters the same, run 9 independent with each algorithm. For each trial, you will randomly select 20% of data from each class as the test set and use the remaining as the training set. For the perceptron, all trials must have the same number of training steps and the same learning rate. The initial weights will be chosen randomly for each trial, but using the same policy (e.g., if you are choosing between 0 and 1, that must be true for all trials).

At the end of each trial, you will calculate the four metrics we discussed in class: Sensitivity, specificity, PPv and NPV. This will be done for both the training set and the test set. Thus, you will get 18 values of each metric for the perceptron (9 each for training and testing), and 18 values for k-NN.

For the perceptron, you should also calculate the fraction of training set points classified correctly at the beginning (i.e., after weight initialization but before training) and in every tenth epoch during training.

You should present your results in a brief report providing the following information. Each item required below should be placed in a separate section with the heading given at the beginning of the item.:

- **Approach:** A brief description and justification of your overall strategy (i.e., how chose the parameters that you did, and why).
- **k-NN Augmentation Strategy:** What augmentation(s) did you choose for k-NN, and why.
- **Performance on Individual Trials:** For each algorithm, four bar plots with nine groups of two bars each. Each plot will be for one of the metrics (sensitivity, specificity, PPV and NPV) , with each pair of bars showing the value of the metric for training and testing on one trial. The training bar should be on the left, the testing on the right. Thus, you will have eight total plots - four for the perceptron and four for k-NN.

- **Average Performance:** For each algorithm, a table giving the mean values of sensitivity, specificity, PPV and NPV for both training and testing data averaged over all 9 trials, along with the standard deviation for each case. The standard deviations will be indicated as $\pm$ values after the mean, e.g., 0.93 ± 0.03. Each table will have four data rows – one for each metric – and two data columns - the left one for training results and the right one for testing results.

- **Trial-Wise Training Error for the Perceptrons:** *For the perceptron case only*: graphs showing the training error plotted against time over the duration of training for the nine trials . Each plot will start with the initial error and plot the error at every tenth epoch. You can either plot all nine curves on the same graph (preferred), or if this gets too cluttered, 9 separate graphs. You should choose graph properties (line thicknesses, colors, etc.) for maximum clarity.

- **Mean Training Error for the Perceptron:** *For the perceptron case only*: a graph showing the mean training error averaged over the 9 trials plotted against time. This graph will have only one plotted curve, where each point of the curve is the average value of the 9 points at the corresponding time in the graph above. At each plotted point, put error bars indicating the $\pm$ standard deviation over the 9 trials.

- **Best k-NN Decision Boundary:** *For k-NN only*: a plot of the data in the feature space indicating the decision boundaries found by the classifier in the best trial. This will involve sampling the feature space, and will only give an approximate boundary.

- **Analysis of Results:** This will have three parts: a) Your analysis of what the results over the 9 trials for each classifier indicate about the suitability of the classifiers for the problem; b) Your opinion of the pros and cons of the classfiers; and c) Your recommendation for one of the classifiers, with a brief justification.

- **Appendix: Programs:** Printouts of your program(s). You may use any programming language, but you *cannot* use neural network or other simulators that provide pre-programmed versions of k-NN or perceptron algorithms. You must write full programs yourself.

The data for the problem is included in the .zip file in plain text. If you cannot access it, please send me mail at. Ali.Minai@uc.edu.

The report should be very brief — no more than 3 pages typed double-spaced, 12 point type — plus the graphs, tables and code. Points will be awarded for: 1) Correctness (i.e., do the classifiers work correctly?); 2) Clarity of description; 3) Quality of the strategy (i.e., does it provide useful information about the classifiers' quality?); and 4) Clarity of arguments and presentation.

As in Homework 1, the report should not be mixed in with the program. It should be a stand-alone document with text, tables, figures, etc., with the program as an appendix. *None of the information required in the report should be given as a comment or note in the program. It must all be in the report.*

You may consult your colleagues for ideas, but *please write your own programs and come to your own conclusions.*