Policy on Collaboration among Students

These policies are the same as were used in Dr. Rosenfeld’s previous version of 2013. The purpose of student collaboration is to facilitate learning, not to circumvent it. Studying the material in groups is strongly encouraged. It is also allowed to seek help from other students in understanding the material needed to solve a particular homework problem, provided no written notes are shared, or are taken at that time, and provided learning is facilitated, not circumvented. The actual solution must be done by each student alone, and the student should be ready to reproduce their solution upon request. The presence or absence of any form of help or collaboration, whether given or received, must be explicitly stated and disclosed in full by all involved, on the first page of their assignment. Specifically, each assignment solution must start by answering the following questions in the report:

- Did you receive any help whatsoever from anyone in solving this assignment? Yes / No. If you answered ‘yes’, give full details: ____________ (e.g. “Jane explained to me what is asked in Question 3.4”)

- Did you give any help whatsoever to anyone in solving this assignment? Yes / No. If you answered ‘yes’, give full details: ____________ (e.g. “I pointed Joe to section 2.3 to help him with Question 2”)

Collaboration without full disclosure will be handled severely, in compliance with CMU’s Policy on Cheating and Plagiarism. As a related point, some of the homework assignments used in this class may have been used in prior versions of this class, or in classes at other institutions. Avoiding the use of heavily tested assignments will detract from the main purpose of these assignments, which is to reinforce the material and stimulate thinking. Because some of these assignments may have been used before, solutions to them may be
(or may have been) available online, or from other people. It is explicitly forbidden to use any such sources, or to consult people who have solved these problems before. You must solve the homework assignments completely on your own. I will mostly rely on your wisdom and honor to follow this rule, but if a violation is detected it will be dealt with harshly. Collaboration with other students who are currently taking the class is allowed, but only under the conditions stated below.

1 Scenario

After learning several machine learning algorithms, Kevin decided to work for a pet frog company "NothingButFrogs.com" as a modeler. Because of the increasing demand of pet frogs, the company wants to set up an automatic system to distinguish frogs from other animals. With a few samples, Kevin built up a classifier which outputs a binary result (frog or not frog). After doing cross-validation in this data set, it reaches over 95% accuracy. Then Kevin proposed to use this classifier.

However, few days after "NothingButFrogs.com" adopted this system, the company noticed the accuracy is only about 60% percent. Moreover, there was a surge of negative comments on the Web.

"Does my pet frog’s skin have some kind of mutation?? It has warts!"
"Thanks to NothingButFrogs.com, now my daughter loves a pet TOAD!!!!!! Where is our pet frog?!!!!!"
There are also several poisoning cases reported on the newspaper....

Unless Kevin can explain why the performance of classifier dropped so much, he is going to lose his job. Can smart 601ers help him out?

2 Confidence Interval

In this homework, you can use any classifier learners you implemented from your previous homework. Beside the accuracy, now you also have to output a confidence interval for the accuracy. You will have to test on two different sets in this part.

The confidence interval can be computed as

\[
[\text{Accuracy}_S(h) - Z_{N} \sqrt{\frac{\text{error}_S(h)(1-\text{Accuracy}_S(h))}{n}}, \text{Accuracy}_S(h) + Z_{N} \sqrt{\frac{\text{error}_S(h)(1-\text{Accuracy}_S(h))}{n}}]
\]

where \( \text{error}_S(h) \) is the error rate on sample set from hypothesis \( h \), \( Z_{N} \) is the appropriate
number of standard deviation corresponding to the interval level $N$ in Normal distribution which can be looked up in a table, and $n$ is the size of testing set.

## 2.1 Test on held-out set

One strong assumption for constructing confidence intervals is that the classifier should be independent from the testing set. (Otherwise it will almost always optimistically biased.)

One way to enforce independence is to draw a partition from training data as testing set and “hold out” this partition during training, so that it is not seen by the classifier learner. However, holding out too much data will weaken the accuracy of the classifier (there is less data in training), while drawing too little data will weaken the accuracy of the test (and lead to a wider confidence interval). Here, you should first create two equal-size partitions, using the last partition as the testing set, and report their 95% and 99% confidence interval respectively. You will then repeat this experiment with 10 equal-size partitions, using the first nine for training and the last one for test.

## 2.2 Test by cross-validation

In order to get more testing data, one way is to do cross-validation. For a 10-fold cross-validation, you train the model 10 times, each time using a different $\frac{1}{10}$ data as the testing set. In the end, every data point has one prediction (since it was used exactly once as a test case), and we can use whole training set to construct a confidence interval.

Note that for each model, though the classifier is independent from the testing data, those 10 models are actually dependent (due to overlap with the training set). However, in practice, this usually a good approximation.

For this section, try 2-fold and 10-fold cross-validation and report their 95% and 99% confidence intervals respectively.

## 3 Scenario II

After getting the help from 601ers, Kevin finally proposed a classifier which reaches over 95% accuracy consistently. However, his colleague, Amy, also proposed a classifier with similar accuracy. Though Kevin is very confident of his classifier, in testing, there are still some times that Amy’s classifier performs better than his. Based on his previous failure, it’s hard for him to persuade the company to use his classifier. Again, he seeks the help from 601ers to give him a systematic way to compare two classifiers.
4 Comparison of classifiers

For a systematic way to evaluate among two classifiers, we again divide the testing set into \( k \) disjoint subsets. For each subset \( T_i \), we compute the difference of error rates from two classifiers.

\[
Y_i = \text{error}_{T_i}(h_1) - \text{error}_{T_i}(h_2)
\]

Each \( Y_i \) is a random variable, so the average \( \bar{Y} = \frac{1}{k} \sum_{i=1}^{k} Y_i \) is also a random variable. When \( k \geq 30 \), the random variable \( \bar{Y} \) approximates a normal distribution. But if \( k < 30 \), it can be approximated as t-distribution with \( k - 1 \) degrees of freedom.

In the next step, we set a null-hypothesis that \( E[\bar{Y}] = 0 \). If we can reject this null-hypothesis by two-tail test, it means one classifier is better than the other (either \( h_1 \) or \( h_2 \) is better). Or, by one-tail test, we can test if the a specific \( h \) is better than the other.

For example, when a given \( k \), and null hypothesis \( E[\bar{Y}] = 0 \), t-value will be

\[
\frac{\bar{y} \sqrt{k}}{S_k}
\]

\[
S_k^2 = \frac{1}{k-1} \sum_{i=1}^{k} (y_i - \bar{y})^2
\]

Then we can see whether we can reject the null hypothesis under different significance levels.

Note: You should implement this test yourself, and not use matlab function \textit{ttest} directly.

4.1 Test on held-out set

Just as in the first part in this homework, we partition the data into training and test, and then partition the test data into \( k \) subsets. Testing a trained classifier on the subsets will give us error rates for each subset. For each subset \( i \), we compute \( y_i = \text{error}_{T_i}(h_1) - \text{error}_{T_i}(h_2) \) and finally \( \bar{y} = \frac{1}{k} \sum_{i=1}^{k} y_i \). Report the accuracy from both classifiers and, under the null-hypothesis \( E[\bar{Y}] = 0 \), report the p-value for a one-tailed test and a two-tailed test. In this experiment you will draw \( \frac{1}{3} \) of the instances as the testing set, and choose \( k \) to be 10.

4.2 Test by cross set

Suppose we don’t have enough data for testing. We can partition training data to \( k \) sets, and train the model \( k \) times, each time with \( k - 1 \) sets as training and one set as testing. Again, for each subset \( i \), compute \( y_i = \text{error}_{T_i}(h_1) - \text{error}_{T_i}(h_2) \) and finally \( \bar{y} = \frac{1}{k} \sum_{i=1}^{k} y_i \). Note that this time your size of training subset is larger than the previous approach. Report
the accuracy from both classifiers, and under the null-hypothesis $E[\tilde{Y}] = 0$, the p-value for one-tailed test and two-tailed test for $k = 10$.

5 Implementation Notes

For this assignment, we encourage you to use Naive Bayes and Logistic Regression as your classifier learner. For comparing classifiers, you can try to compare different parameters for the same classifier, or compare two different algorithms. For example, you might compare different smoothing methods in Naive Bayes, different regularized term in logistic regression, or different optimization methods (GD, SGD,...).

For first part of this assignment, you will need to implement following functions.

```plaintext
testInstanceLabel = PartitionHeldOut(size, k);
Ypredict = TrainHeldOut(Xtrain, Ytrain, testInstanceLabel);
crossSetLabel = PartitionCrossSet(size,k);
Ypredict = TrainCrossSet(Xtrain, Ytrain, crossSetLabel);
[Accuracy, lowerInterval, upperInterval] = ConstructInterval(Ypredict,Ytest, confLevel);
```

In `PartitionHeldOut` function, you randomly select $k$ equal size partition and use one of them to be your testing set. You should output a binary vector `testInstanceLabel`, indicating which instances are used as testing, it should have same length as $Xtrain$ and $Ytrain$ and has $\frac{n}{k}$ elements to be 1, where $n$ is the number of instance.

In `TrainHeldOut` function, `testInstanceLabel` is a binary vector to indicate the training set and testing set. You can use any classifier here. Your output `Ypredict` will only contain test part of instances.

In `PartitionCrossSet`, you randomly assign instances to one of $k$ sets, and every set should have (as close as possible to) the same number of instances. `crossSetLabel` is a vector contains number from 1 to $k$, which indicates the set the instances belong to.

In `TrainCrossSet`, `crossSetLabel` indicates how you segment the data to $k$ sets, you should train model $k$ times and every time one partition will be the testing set. Therefore, the size of `Ypredict` should be the same as $Ytrain$.

For the `ConstructInterval` function, $Ytest$ is the true label for your `Ypredict`, and you should at least support `confLevel` for 0.95 and 0.99.

For second part of this assignment, you will need to implement following functions.
testInstanceLabel = PartitionHeldOut(size, k1);
Ypredict1 = TrainHeldOut1(Xtrain, Ytrain, testInstanceLabel);
Ypredict2 = TrainHeldOut2(Xtrain, Ytrain, testInstanceLabel);

crossSetLabel = PartitionCrossSet(size,k2);
Ypredict1 = TrainCrossSet1(Xtrain, Ytrain, crossSetLabel);
Ypredict2 = TrainCrossSet2(Xtrain, Ytrain, crossSetLabel);

[accuracy1, accuracy2, p_value] =
CompareClassifier(Ypredict1,Ypredict2, Ytest, crossSetLabel,isTwoTail);

In TrainHeldOut1 and TrainHeldOut2 function, they have the same functionality as in part 1, but you have to use different classifiers in these two functions. Same rule applies to function TrainCrossSet1 and TrainCrossSet2

For the held-out part, after you made prediction on a partition, you have to segment them to $k_2$ subsets again, you should be able to use PartitionCrossSet from first part to do so.

For CompareClassifier, you should have same length for Ypredict1 and Ypredict2. crossSetLabel indicates the set each instances belong to. Ytest is the true label for Ypredict1 and Ypredict2. When IsTwoTail = 1, you should perform two-tailed test, otherwise you should perform one-tailed test. For accuracy, you should show the overall accuracy for both classifiers.

6 Data Set

For this assignment, you should download the handout data from [http://curtis.ml.cmu.edu/w/courses/images/a/ad/Assignment_5-1.mat](http://curtis.ml.cmu.edu/w/courses/images/a/ad/Assignment_5-1.mat) and [http://curtis.ml.cmu.edu/w/courses/images/1/1b/Assignment_5-2.mat](http://curtis.ml.cmu.edu/w/courses/images/1/1b/Assignment_5-2.mat) Both have ”Xtrain” and ”Ytrain”. In this assignment, you don’t have test data since you have to segment test data yourself in order to validate your selection of classifier. (You won’t have the real ”frog” dataset because Kevin is afraid of that smart 601ers will steal his job.)

7 Deliverables

Submit your codes in four parts via AutoLab. You can use the your codes from previous assignments as classifiers. You are NOT allowed to use any off-the-shelf optimizer. You should upload your codes (including all your function files) along with a report, which should solve
the following question:

1.1 Specify what classifier you use, and report their 95% and 99% confidence interval on number
of partition 2 and 10.
What do you observe in the result?

1.2 Specify what classifier you use, and report their 95% and 99% confidence interval for 2 and
10 cross sets.
What difference do you see from 1.1?

2.1 Specify what two classifiers you use, report their accuracy and p-value under one-tailed test
and two-tailed test for drawing \( \frac{1}{5} \) of instances as testing set and choose \( k \) to be 10.

2.2 Specify what two classifiers you use, report their accuracy and p-value under one-tailed test
and two-tailed test with \( k=10 \). Compare with the result you got from 2.1.

You should tar gzip the following items into hw5.tgz and submit to the homework 5 assign-
ment under Autolab:

- TrainHeldOut.m
- TrainCrossSet.m
- ConstructInterval.m
- TrainHeldOut1.m
- TrainHeldOut2.m
- TrainCrossSet1.m
- TrainCrossSet2.m
- CompareClassifier.m
- and all other auxiliary functions you have written
- report.pdf

Tar gzip the files directly using tar -cvf hw5.tgz *.m report.pdf. Do NOT put the above
files in a folder and then tar gzip the folder. You do not need to upload the saved predicted
labels (i.e. the .mat files). Please make sure your code is working fine under Octave before
you submit.
8 Submission

You must submit your homework through Autolab via the “Homework5-submission” link. In this homework, we provide an additional tool called “Homework5-validation”:

- Homework5-validation: You will be notified by Autolab if you can successfully finish your job on the Autolab virtual machines. Note that this is not the place you should debug or develop your algorithm. All development should be done on linux.andrew.cmu.edu machines. This is basically a Autolab debug mode. There will be NO feedback on your performance or score in this debugging mode. You have unlimited amount of submissions here (as in homework4), but do as much debugging as possible outside of autolab to avoid filling up the autolab queues.

- Homework5-submission: This is where you should submit your validated final submission. You have a total of 5 possible submissions. Your performance will be evaluated and feedback will be provided immediately.