Due on Jan 31 2017, beginning of class (Online students can email me a PDF/MS Word file, by that time).

Note: We can measure *accuracy-rate* or *error rate*. To convert between them we just subtract from 1.0. You can use either one in this report, just be clear and consistent.

1. Review the Nearest\_Neigbour\_Matlab.pptx briefing.
2. Visit the UCR time series Archive, download the data, and unzip it. The pass is attempttoclassify
3. Randomly select four datasets, two of them should have small *training* sets (less than 80 objects) and two should have large *training* sets (300 or more).
4. Compute the leave-one-out classification accuracy on your four datasets. To be clear, for this you are *only* using the TRAIN data. This is your model’s accuracy prediction, that is to say, you can tell your boss something like: “*I think when we classify MoteStrain data next week with this model, we should be able to get about 0.121 error rate, or 0.879 accuracy*”.
5. Compute the holdout accuracy. That is, use the TRAIN data to classify the TEST data.
6. Fill out a table like the one below. Write a paragraph explaining what you did, and any observations.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Num Classes | Size of Training data | Size of Test data | Num Features (time series length) | Default Rate | Leave-one-out Error-Rate | Hold-Out  Error-Rate |
| **Yoga** | 2 | 300 | 3,000 | 426 | etc | etc | etc |
| etc |  |  |  |  |  |  |  |
| etc |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

1. Visit the UCI Data Archive [b]. Grab the leaf dataset [c]. Read the *ReadMe.pdf* file. Here someone has done a good job at explaining how they took images of leaves, and extracted features.
2. Compute the leave-one-out error-rate on this dataset.
   1. You may have to slightly reformat data. For example, I think that column two in the csv file is just the item number, and therefore must be removed. Mention any such processing in your report.
3. Compute the leave-one-out error-rate on this dataset, first on the original data.
   1. I noticed that some features range from about 0 to 20, and others only 0 to 0.2. This suggests that it *might* be helpful to normalize the data. We will try two different ways.
   2. **Zero-One normalization:** For each column. Subtract the minimum value from everything in the column, then divide everything in the column by the largest value. Something like this..

>> VarName12 = VarName12-min(VarName12);

>> VarName12 = VarName12/max(VarName12);

As a sanity check, plot and inspect a histogram: hist(VarName12)

* 1. **Z normalization:** For each column. Subtract the mean value from everything in the column, then divide everything in the column by the standard deviation. Something like this..

>> VarName12 = (VarName12-mean(VarName12))/std(VarName12);

Matlab has a built-in function, zscore that does this.

1. Write a short report comparing the leave-one-out error-rate results you found on, original, z-normalized and Zero-one normalized features.
2. Visit the UCI data Archive [b]. Click on “Classification” Task, there will be about 260 datasets. Some of these datasets are not in the nice clean table format that we like, but many are. Find one more dataset with Multivariate features (but not a time series) and repeat steps 8 to 9 above. You may have to do some data cleaning, reformatting etc. Document anything you had to do.

Hand in a one to two page report, with tables, and any figures your think might be useful for me to see (if any).

[a] http://www.cs.ucr.edu/~eamonn/time\_series\_data/

[b] https://archive.ics.uci.edu/ml/datasets.html

[c] https://archive.ics.uci.edu/ml/machine-learning-databases/00288/