Midterm COMP-135 – **Due 03/29/2016**

The dataset for the exam is the Semeion handwritten digits dataset.

- Each instance represents a handwritten character in black and white.
- Each feature specifies whether to 1 of the 256 pixels was on/off in the original 16x16 image.
- Each class label is a digit, which are identified as 1/0 in the last 10 columns of the dataset

Link to dataset:
https://archive.ics.uci.edu/ml/datasets/Semeion+Handwritten+Digit

What to turn in:

- Your version of **this exam document**.
- You will need to write code to complete the answers to this exam. **Turn in code** along with your solutions. In the corresponding sections of this document, **note what files, and if necessary, what lines correspond to each answer.**
- Most questions specify that you create output files that contain your answers, **use the specified file names**.
- If you aren't confident in any particular answer, write down your reasoning process into this document to boost your odds of partial credit.
- ***PUT ALL EXAM FILES INTO A SINGLE ZIP FILE WHEN SUBMITTING YOUR EXAM, THEN EMAIL THE ZIP TO: kyle@eecs.tufts.edu***
    - ***Your zip file should contain:***
        - MidtermExam.docx – this file with answers, pointers to your code, and any comments to help with partial credit
        - problem1a.txt
        - problem2a.txt
        - problem2b.txt
        - problem2c.txt
        - problem3a.txt
        - problem3b.txt
        - problem4a.txt
        - problem5a.txt
        - problem5b.txt
        - problem5c.txt
        - Any code you wrote

Suggestions:

- Some of the work/code from problems 1 and 2 are used multiple times, so read ahead before writing/coding your solutions.
    - Problem 3 involves using your decision tree algorithm
    - Problem 4 involves both the decision tree algorithm and kNN

PROBLEM 1: K-NEAREST NEIGHBORS

Classify the 5 digits contained in unknownCharacters.txt with kNN using k=25, if there is a tie return all best matching digits.

a) What is the index of the closest matching instance for each unknown character (if there is a tie list the indices of all closest matching instances)? **[20 points]**
Save this as a text file called "problem1a.txt" with the index of each closest match stored as a separate row.

PROBLEM 2: DECISION TREES

Create a decision tree using ID3 on the complete dataset with a maximum depth of 5, using information gain to split nodes (and no pruning).

a) What is the tree? Return a list of feature indices that are used for splits, and the height of each split (root=0) **[10 points]**
Save this as a text file called "problem2a.txt" with the feature index of each split and corresponding height a separate row, e.g. for the root: "161 0".

b) What is the accuracy of the trained decision tree for each digit in the training set? **[10 points]**
Save this as a text file called "problem2b.txt" with the accuracy of each digit listed as a new line, starting from 0.

c) What are the decision tree predictions on unknownCharacters.txt? **[10 points]**
Save this as a text file called "problem2c.txt" with each class listed as a new line in the same order as unknownCharacters.txt.

I made a pdf of the results for part 2a

PROBLEM 3: FORWARD-SELECTION WRAPPER METHOD

Use the forward-selection wrapper method and your decision tree algorithm from problem 2 to do feature selection on the dataset. Run for 25 iterations.

a) What is the accuracy of the classifiers you train at each iteration feature selection? **[10 points]**
Save this as a text file called "problem3a.txt" with the accuracy after each iteration listed as a newline starting with the accuracy with 1 feature on the first line.

b) What are the indices of the features you end up with (use 0-based indexing)? **[10 points]**
Save this as a text file called "problem3b.txt" with each feature index listed on a new line.

PROBLEM 4: COMPARING CLASSIFIERS

Compare the performance of kNN with k=25, and ID3 decision tree with no pruning and a maximum depth of 5.

- Partition the dataset into 10 folds according to k-fold cross validation. Use the same partitions for both classifiers.
- Train and test both your decision tree algorithm and kNN algorithm with respect to each fold.

a) What is the accuracy of each classifier for each fold? **[10 points]**
Save this as a text file called "problem4a.txt" with the accuracy for both classifiers on a given fold listed on a new line in order: decision tree, kNN; i.e. "0.9 0.5".

b) With a confidence of 99% using a 2-tailed T-test, is there a difference in performance between the algorithms? **[10 points]**
The answer to this question should be written into this document.

Solution:

Since our Semeion dataset observations far exceed 30, we may evoke the Central Limit Theorem in which our t statistic is approximately normal(it converges to a normal distribution as $n \to \infty$)

$$u_{knn} = 0.886823547226$$
$$u_{ID3} = 0.662406709961$$

$$H_0: u_{knn} = u_{ID3}$$
$$H_\alpha: u_{knn} \neq u_{ID3}$$

$$reject\ H_0\ if\ |t_0| = \left| \frac{u_{knn} - u_{ID3}}{\sqrt{\left(\frac{\sigma^2_{\mathrm{knn}}}{n_{knn}} + \frac{\sigma^2_{\mathrm{ID3}}}{n_{ID3}}\right)}} \right| > Z_{\frac{\alpha}{2}}\ where\ SE = \sqrt{\left(\frac{\sigma^2_{\mathrm{knn}}}{n_{knn}} + \frac{\sigma^2_{\mathrm{ID3}}}{n_{ID3}}\right)}$$

$$|t_0| = \left| \frac{0.224416837265}{0.0121800447643} \right| = |18.42|$$

$$Reject\ H_0\ since\ z_{\frac{\alpha}{2}} = 2.576 < 18.42\ and - 18.42 < -2.576$$

Therefore, there is statistically significant with 99% confidence that there exist a difference in performance between the two algorithms

PROBLEM 5: K-MEANS CLUSTERING

Apply k-Means clustering on the Semeion dataset to group characters together. Using k = 10 and initializing centroids at random instances in the dataset, run k-Means for 25 iterations.

a) Which cluster corresponds to which digit, and how accurate is that correspondence? To do this, create a 2D table, and count how many times each cluster label coincides with each class label. Report the accuracy for the best matching cluster and the corresponding class label for all digits (note: every digit should be assigned a cluster, although multiple digits may be assigned the same cluster). **[20 points]**
Save this as a text file called "problem5a.txt" with each digit listed on a new row along with the best matching cluster identifier, and the accuracy of that match, i.e. "0 3 0.5".

b) Run k-Means 10 times with different initial conditions, and calculate the normalized mutual information (http://kephale.github.io/TuftsCOMP135_Spring2016/Lecture12/#/6/2) for each clustering. For each run of k-Means, what is the normalized mutual information between the cluster labels and the class labels? **[10 points]**
Save this as a text file called "problem5b.txt" with the normalized mutual information from each run of k-Means as a new line.

c) What is the accuracy of cluster labels for each digit for the clustering with the greatest normalized mutual information? **[10 points]**
Save this as a text file called "problem5c.txt" with each digit listed on a new row along with the best matching cluster identifier, and the accuracy of that match, i.e. "0 3 0.5".