605.449 — Introduction to Machine Learning

Programming Project #2

Due: September 24, 2017

The purpose of this assignment is to give you an introduction to unsupervised learning by implementing two feature selection algorithms and two clustering algorithms. The two feature selection algorithms are StepwiseForwardSelection, or SFS (introduced in Module 03), and GeneticAlgorithmSelection, or GAS (also introduced in Module 03). Since these are wrapper methods, you will need to test these algorithms using another algorithm as well. Normally, these two feature selection methods are used with a classifier; however, in this assignment, we will use the results of clustering to evaluate the features. The two clustering algorithms being implemented are k-Means and HAC, both of which were introduced in Module 04. To evaluate HAC, cut the tree to yield the same number of clusters as in k-means (i.e., k). You should evaluate these algorithms for $k =$ the number of classes in the data set. Specifically, the measure based on Fisher's LDA should be used for your evaluation:

$$\mathcal{L} = \frac{\sum_{i=1}^{k-1} \sum_{j=i+1}^{k} (\mathbf{m}_i - \mathbf{m}_j)^2}{\sum_{i=1}^{k} s_i^2}$$

For this assignment, you will use three datasets that you will download from the UCI Machine Learning Repository, namely:

1. Glass — `https://archive.ics.uci.edu/ml/datasets/Glass+Identification`

   The study of classification of types of glass was motivated by criminological investigation.

2. Iris — `https://archive.ics.uci.edu/ml/datasets/Iris`

   The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant.

3. Spambase — `https://archive.ics.uci.edu/ml/datasets/Spambase`

   This collection of spam e-mails came from a postmaster and individuals who had filed spam. This is a two-class problem with a large number of attributes and a large number of instances.

As with the prior assignment, some of the data sets have missing attribute values. When this occurs in low numbers, you may simply edit the corresponding values out of the data sets. For more occurrences, you should do some kind of "data imputation" where, basically, you generate a value of some kind. This can be purely random, or it can be sampled according to the conditional probability of the values occurring, given the underlying class for that example. The choice is yours, but be sure to document your choice.

Our ability to handle large data sets is becoming more and more important, and one of the data sets has been selected to get you thinking about how to handle such data. Specifically, for the Spambase data set, some attention should be given to optimizing the feature selection process. One approach is to apply the feature selection on a subset of the data, rather than the full data set. While there is some risk associated with missing cluster boundaries this way, it can greatly speed up the feature selection process.

For this project, the following steps are required:

- Download the three (3) data sets from the UCI Machine Learning repository. You can find this repository at `http://archive.ics.uci.edu/ml/`. All of the specific URLs are also provided above.
- Pre-process each data set as necessary to handle missing data.
- Implement k-means and HAC so you can use them for the wrapper feature selection methods.
- Implement SFS and GAS with the loss function defined above.
- Run your algorithms on each of the data sets. These runs should output the feature sets and best clusters in a way that can be interpreted by a human. There is no need to separate the data into training and test sets. Be sure to test your algorithms with the Fisher score given above, *not* classification accuracy!

- Write a very brief paper that incorporates the following elements, summarizing the results of your experiments. You should also output the summary statistics on clustering performance.

  1. Title and author name
  2. A brief, one paragraph abstract summarizing the results of the experiments
  3. Problem statement, including hypothesis, projecting how you expect each algorithm to perform
  4. Brief description of algorithms implemented
  5. Brief description of your experimental approach
  6. Presentation of the results of your experiments
  7. A discussion of the behavior of your algorithms, combined with any conclusions you can draw
  8. Summary
  9. References (you should have at least one reference related to each of the algorithms implemented, a reference to the data sources, and any other references you consider to be relevant)

- Submit your fully documented code, the outputs from running your programs, and your paper. Your grade will be broken down as follows:

  - Code structure – 10%
  - Code documentation/commenting – 10%
  - Proper functioning of your code, as illustrated by the code outputs – 30%
  - Summary paper – 50%