605.449 — Introduction to Machine Learning

Programming Project #5

Due: November 5, 2017

The purpose of this assignment is to give you a firm foundation in comparing a variety of linear classifiers. In this project, you will compare two different algorithms, one of which you have already implemented. These algorithms include Naïve Bayes and Logistic Regression. For extra credit, you also have the option of implementing the Adaline network. You will also use the same five datasets that you used from Project 1 from the UCI Machine Learning Repository, namely:

1. Breast Cancer — `https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29`

   This breast cancer databases was obtained from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg.

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

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

3. 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.

4. Soybean (small) — `https://archive.ics.uci.edu/ml/datasets/Soybean+%28Small%29`

   A small subset of the original soybean database.

5. Vote — `https://archive.ics.uci.edu/ml/datasets/Congressional+Voting+Records`

   This data set includes votes for each of the U.S. House of Representatives Congressmen on the 16 key votes identified by the Congressional Quarterly Almanac.

When using these data sets, be careful of some issues.

1. Not all of these data sets correspond to 2-class classification problems. For Naïve Bayes, that is not really a problem. A method for handling multi-class classification was described for Logistic Regression. For Adaline, it is suggested that you use what is called a "multi-net." This is where you train a single network with multiple outputs. Note that if you wish to apply a one-vs-one or one-vs-all strategy for the neural network, that is acceptable. Just be sure to explain your strategy in your report.

2. 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.

3. Most of attributes in the various data sets are either multi-value discrete (categorical) or real-valued. You will need to deal with this in some way. For the multi-value situation, once again Naïve Bayes should be fine, but the other methods will have a problem. In that case, you can apply what is called "one-hot coding" where you create a separate Boolean attribute for each value. Again, I recommend you go ahead and use this for Naïve Bayes, even though it is not really necessary. For the continuous attributes, you may use one-hot-coding if you wish, and this will be required for Naïve Bayes and Logistic Regression, but not the neural net. For the networks, there is actually a better way. Specifically, it is recommended that you normalize them first to be in the range -1 to $+1$. (If you want to normalize to be in the range 0 to 1, that's fine. Just be consistent.)

For this project, the following steps are required:

- Download the five (5) 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 and non-Boolean data (both classes and attributes).
- Implement Naïve Bayes and Logistic Regression.
- For extra credit (up to 20 points), implement Adaline.
- Run your algorithms on each of the data sets. These runs should be done with 5-fold cross-validation so you can compare your results statistically. You can use classification error or mean squared error (as appropriate) for your loss function.
- Run your algorithms on each of the data sets. These runs should output the learned models in a way that can be interpreted by a human, and they should output the classifications on all of the test examples. If you are doing cross-validation, just output classifications for one fold each.
- Write a very brief paper that incorporates the following elements, summarizing the results of your experiments. You should also output the summary statistics on classification accuracy.
    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%