605.649 — Introduction to Machine Learning

Programming Project #1

The purpose of this assignment is to give you an introduction to machine learning by implementing two fairly simple learning algorithms. These two algorithms are called Winnow-2 (introduced in Module 01) and Naïve Bayes (introduced in Module 02). Feel free to read ahead in the book if you want to get started on Naïve Bayes early. For this assignment, you will use the following five datasets that you will download from the UCI Machine Learning Repository[1], 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. Be careful with this data set since "?" does not indicate a missing attribute value. It actually means "abstain."

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 problem. But for Winnow-2, you will need to use one classifier for each class.

2. Some of the data sets have missing attribute values, which is usually indicated by "?". When this occurs in low numbers, you may simply edit the corresponding data items 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. A naïve approach is to impute the missing value with a random number or the attribute's mean (or median). A better approach is to sample according to the conditional probability of the values occurring, given the underlying class for that example. The choice of strategy 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 discrete situation, Winnow-2 in particular 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 will need to discretize them in some way for both algorithms and then proceed as in the multi-valued categorical case. A common approach is to bin into a small set of groups. Common methods include equal-width binning (divide data into equal-sized intervals) and equal frequency binning (divide bins such that each contains the same number of points). Again, the choice is yours, but explain your choice in your report.

[1] Note that all of the data sets are also available in the content area within Blackboard

4. Let's talk about tuning. You should take the following advice for all of your assignments (except Project 6 ... you'll see why when you get to it). Note that this strategy is not ideal but is recommended due to the small data sets combined with the fact this part should be kept relatively simple. First, extract 10% of the data to be used for tuning. For your training set, test against this 10% with different parameter values and pick the best model. Then apply the model that goes with those tuned values against your test set. To be specific, as you will see below, you will be working with a 2/3–1/3 split of the data. So first take out 10% for tuning. Then from the remaining 90% split 2/3–1/3. This means 60% will be used for training and 30% will be used for testing. Train on the 60% while tuning with the 10%. Take the result and evaluate generalization ability on the 30%.

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. Again, all of the data sets are available in Blackboard as well.
- Pre-process each data set as necessary to handle missing data and non-Boolean data (both classes and attributes).
- Set up your test and training sets from the provided data. Specifically, split the data into two groups randomly where 2/3 of the data will be used for training and 1/3 will be used for testing. If you are more ambitious, you may set up a cross-validation experiment. In that case, I recommend 5-fold cross-validation since that is what you will be doing in future assignments. If you don't know what this means, don't worry about it for now.
- Implement both Naïve Bayes and Winnow-2. Note that you will need to tune α and θ for Winnow-2 as well as p and m for Naïve Bayes.
- Run your algorithms on each of the five 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. Your paper is required to be at least 5 pages and no more than 10 pages using the JMLR format You can find templates for this format at `http://www.jmlr.org/format/format.html`. The format is also available within Overleaf.
  1. Title and author name
  2. Problem statement, including hypothesis, projecting how you expect each algorithm to perform
  3. Brief description of your experimental approach, including any assumptions made with your algorithms
  4. Presentation of the results of your experiments
  5. A discussion of the behavior of your algorithms, combined with any conclusions you can draw
  6. Summary
  7. References (Only required if you use a resource other than the course content.)
- Submit your fully documented code, the outputs from running your programs, and your paper.
- For the video, the following constitute minimal requirements that must be satisfied:
  - The video is to be no longer than 5 minutes long.
  - The video should be provided in mp4 format. Alternatively, it can be uploaded to a streaming service such as YouTube with a link provided.
  - Fast forwarding is permitted through long computational cycles. Fast forwarding is *not permitted* whenever there is a voice-over or when results are being presented.

  - Be sure to provide verbal commentary or explanation on all of the elements you are demonstrating.
  - Demonstrate the mapping of a non-binary categorical variable to one-hot coding.
  - Demonstrate your discretization method for the real-valued features.
  - Provide sample outputs on the hold-out set for one of the data sets showing classification performance on both Winnow-2 and Naïve Bayes.
  - Show a sample trained Winnow-2 model. This will consist of a table of weights associated with each of the features, as well as the associated decision threshold θ.
  - Show a sample trained Naïve Bayes mode. This will consist of the set of class probabilities as well as the class-conditional feature probabilities.
  - Demonstrate both promotion and demotion for Winnow-2.
  - Demonstrate the counting process for Naïve Bayes by showing the constitution counts for a class probability as well as for a class-conditional feature probability.
  - Show the performance on your hold-out set for both Winnow-2 and Naïve Bayes on one of the data sets.

Your grade will be broken down as follows:

- Code structure – 10%
- Code documentation/commenting – 10%
- Proper functioning of your code, as illustrated by a 5 minute video – 30%
- Summary paper – 50%