CSCI 374 — Machine Learning and Data Mining
Oberlin College — Fall 2016

Homework #2: Naïve Bayes

Important Dates

Assigned: October 14

Snapshot 1: October 28 (11:59 PM)

Snapshot 2: November 4 (11:59 PM)

Final Due Date: November 7 (11:59 PM)

Assignment

In this assignment, you will practice:

1) acquiring real-world data to use as "experience" for machine learning,
2) preprocessing data before training,
3) implementing another machine learning algorithm from scratch,
4) experimenting with various algorithms on a variety of data sets,
5) analyzing the results of those experiments to evaluate the performance of the different implemented learning algorithms with respect to different data sets, and
6) writing a technical report detailing (i) how your implementation works, (ii) your experimental setup, (iii) the results of your experiments, and (iv) any implications or lessons learned from your implementation and results.

In particular, you will implement the Naïve Bayes algorithm discussed in class for learning probabilistic representations of a supervised learning classifier. Through implementing the algorithm (rather than re-using existing implementations), you will gain a better understanding of how Naïve Bayes learns to predict the probabilities of labels, how it can be used in different settings, as well as the differences between Naïve Bayes and two of the decision tree algorithms from the first homework — including their relative advantages and disadvantages.

This assignment has two parts, described below. You will write a separate program for each of the two parts (although you should feel free to share code between the two – the two parts just need different entry points into your programs.

Acceptable Programming Languages

You can use either the **Java** or **Python** programming languages to complete this assignment.

Part 1: Comparison with Decision Trees

In the first part this assignment, your goal is to train and test Naïve Bayes on the two nominal data sets considered in Homework 1, then compare the results from Naïve Bayes with the results from two of the algorithms you implemented in Homework 1 — ID3 and C4.5. Note: if you did not complete C4.5, you can once again use Weka to generate your C4.5 results (please note this in your final report).

Data Sets

1) **monks1.csv**: A data set describing two classes of robots using all nominal attributes and a binary label. This data set has a simple rule set for determining the label: `if head_shape = body_shape ∨ jacket_color = red, then` *`yes`*`, else` *`no`*. This data set is useful for debugging your implementations and verifying their correctness. Monks1 was one of the first machine learning challenge problems (http://www.mli.gmu.edu/papers/91-95/91-28.pdf). This data set comes from the UCI Machine Learning Repository: http://archive.ics.uci.edu/ml/datasets/MONK%27s+Problems
2) **opticalDigit.csv**: A data set of optical character recognition of numeric digits from processed pixel data. Each instance represents a different 32x32 pixel image of a handwritten numeric digit (from 0 through 9). Each image was partitioned into 64 4x4 pixel segments and the number of pixels with non-background color were counted in each segment. These 64 counts (ranging from 0-16) are the 64 attributes in the data set, and the label is the number from 0-9 that is represented by the image. This data set is more complex than the Monks1 data set, but still contains only nominal attributes and a nominal label. This data set comes from the UCI Machine Learning Repository: http://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits

Both data sets can still be downloaded from the "Course Content/Homework 1" folder on Blackboard. The file format for each of these data sets is described in the Homework 1 Assignment, in case you need to refer back to it.

Program Behavior

Your program for Part 1 should behave as follows, similar to your program for Homework 1:

1) Your program should be named NBPart1
2) It should take as input two parameters:
    a. The path to a file containing a data set (e.g., monks1.csv)
    b. A random seed as an integer
3) Next, the program should read in the data set as a set of instances
4) The instances should be split into training and test sets (using the random seed input to the program)

5) The training set should be fed into Naïve Bayes to learn the conditional probability relationships between attribute values and labels
6) The learned model should be evaluated using the test set created in Step 3.
7) The confusion matrix counted during Step 5 should be output as a file with its name following the pattern: results_<DataSet>_NaiveBayes_<Seed>.csv (e.g., results_monks1_NaiveBayes_12345.csv).

The file format for your output file should be exactly the same as in Homework 1. Please refer back to the Homework 1 Assignment for details, if necessary.

Experiment

The goal of the following experiment is to investigate how different types of supervised learning algorithms perform on different data sets with different properties. That is, our goal is to compare Bayesian Learning (Naïve Bayes) with Decision Trees (ID3 and C4.5).

For your experiment in Part 1, pick 30 random seeds (include them in your report), then calculate the average accuracy of Naïve Bayes, ID3, and C4.5 on each of the two data sets across the 30 runs (one run per seed). Afterwards, compare the average accuracies between the algorithms to evaluate their performances on the different data sets. In particular, evaluate:

1) How did each algorithm perform on Monks 1? Did one type of approach (Bayesian vs. Decision Tree) achieve significantly better performance on this data set? If so, why do you think this happened? If not, what does this say about the two approaches?
2) How did each algorithm perform on Optical Digit? Did one type of approach (Bayesian vs. Decision Tree) achieve significantly better performance on this data set? If so, why do you think this happened? If not, what does this say about the two approaches?

Confidence Interval Calculations

Different from Homework 1, we want to calculate confidence intervals around the average accuracies from the 30 runs for Homework 2 so that we can evaluate if one algorithm significantly outperformed another. To do so, we use a slightly different equation for the confidence interval.

Let Acc_{Alg} be the list of accuracy values calculated from the $m = |Acc_{Alg}|$ runs for a particular algorithm Alg. Let

$$\bar{x} = \frac{1}{m} \sum_{\hat{p} \in Acc_{Alg}} \hat{p}$$

represent the average accuracy of Alg across those m runs. Let n be the size of the testing set in each run. Finally, let

$$s_{Alg} = \sqrt{\frac{1}{m-1} \sum_{\hat{p} \in Acc_{Alg}} (\hat{p} - \bar{x})^2}$$

represent the standard deviation of the of Alg across those m runs. Then, the 95% confidence interval of $\bar{x}$ for Alg is approximately:

$$\bar{x} \pm 1.96 \frac{s_{Alg}}{\sqrt{mn}} = \left[\bar{x} - 1.96 \frac{s_{Alg}}{\sqrt{mn}}, \bar{x} + 1.96 \frac{s_{Alg}}{\sqrt{mn}}\right]$$

Although this approximation is not exact, it is close enough (and easy enough to calculate) for the purposes of this homework assignment.

Part 2: Let's Play Jeopardy!® (Text Classification)

In the second part this assignment, your goal is to write a program capable of competing in a simplified game of Jeopardy!® where all the questions ask for the author of a given passage from a famous writing. For your program, you will train and test Naïve Bayes as a text classifier using text downloaded from Project Gutenberg. In particular, you will download popular, famous books from 10 authors, train Naïve Bayes to learn the writing styles (indicated by word choices) of each author from those texts, then predict which author wrote 50 short passages (taken from different texts than those you used for training).

Data Sets

You need to download the TXT files of the following books from Project Gutenberg at https://www.gutenberg.org/wiki/Main_Page:

1. *Pride and Prejudice* by Jane Austen
2. *Alice's Adventures in Wonderland* by Lewis Carroll
3. *Great Expectations* by Charles Dickens
4. *The Adventures of Sherlock Holmes* by Arthur Doyle
5. *The Odyssey* by Homer
6. *The Trial* by Franz Kafka
7. *The Republic* by Plato
8. *Anna Karenina* by Leo Tolstoy
9. *The War of the Worlds* by H.G. Wells

Additionally, you should also find two texts by another author of your choice. Pick one of those two (preferably the larger of the two) as a tenth book to include with the nine listed above. These nine books listed above (plus the one you chose as a tenth) will serve as the training set for your machine learning with Naïve Bayes. The other book you chose will be part of your testing set (described below).

Note: from these ten books used for training, you will want to manually remove the additional text added by Project Gutenberg located at the beginning and end of each file so that you are only learning from the original text by the author (or its translation by another author).

Program Behavior

Your program for Part 2 should behave as follows:

1) Your program should be named NBPart2
2) It should take in one parameter:
    a. The path to a file containing the *test* set (downloaded from Blackboard)
3) Next, the program should read in the ten previously identified books to use for training
4) The text of each book should be preprocessed to make it appropriate for training with Naïve Bayes:
    a. The text should be split into a list of words (or lists of words, one per paragraph/sentence/however you wish)
    b. Each word should be converted to lower case so that capitalization is ignored
    c. Stop words should be removed (e.g., a, an, the). You can choose your own stop words (feel free to search the internet for a list, just remember to cite your source in your code and report)
    d. Remaining words should be "stemmed"
5) The text of the test set should also be read in and preprocessed the same as with the training data in Step 4.
6) The stemmed words from each book should be fed into Naïve Bayes to learn models of the writing styles of each author (where the label for your data is the author of the text).
7) The learned model should be evaluated using the provided test set (with your additions for your chosen tenth author).
8) The confusion matrix counted during Step 6 should be output as a file with its name following the pattern: results_TextClassification.csv

I will provide you with code to: (1) build a list of keywords from a String of text, and (2) create a list of stemmed words from a list of words. This code can be the only code you use that relies on external libraries.

The format of the test file (called TestSet_Passages.txt on Blackboard) is as follows:

```
################################################################################
Label
Passage
################################################################################
Label
Passage
################################################################################
etc.
```

You will need to be able to read in the test instances between the #### lines, where the first line is the actual author of the passage, and the second line is the passage to be tested. **You should add 5 passages from the second book (*not* the one used in training) written by your chosen author to this test set so that you can also evaluate the ability of your program to predict passages written by your chosen author**.

Experiment

For this experiment, you do not need to do anything with random seeds. Instead, you are given an explicit training set (the 10 books) and an explicit test set (of 50 passages, after your 5 passage are added). Your goal is to:

1) Calculate the overall predictive accuracy of your Naïve Bayes implementation on the 50 test passages.
2) Compare the recall and precision for each author. Which authors did your program best learn to predict correctly, and for whom did it have the most difficulty?
3) Investigate: for the authors for which your program made incorrect predictions, were there any trends that you observed? That is, did your program tend to confuse two or more authors, thinking that they were similar? If so, does this confusion make sense given what you know about those authors (e.g., their time period, their location, etc.)?

For your overall accuracy, please use the original confidence interval calculations used for Experiment 1 in Homework 1 (and not the one described above for Part 1 of this assignment). You do not need to find confidence intervals for the precision and recall measures (since their n will be much smaller than 30, $Z_{0.95} = 1.96$ will not be not a close approximation).

Snapshots

Since the homework assignment is multiple weeks long, there are two intermediate deadlines to help you make sure you complete the entire assignment on time:

Snapshot 1 (due Friday October 28 at 11:59 PM): you should have the program done for:

- The implementation for Part 1 of the assignment

Snapshot 2 (due Friday November 4 at 11:59 PM): you should have the program done for:

- The implementation for Part 2 of the assignment

For each snapshot, your code (and associated Makefile and README described below) should be organized in a ZIP file and turned in on Blackboard. Your zip file should be named:

<OCCSUserName>_SnapshotX.zip

For example, Alice Student's second snapshot would be named: astudent_Snapshot2.zip

Final Handin

Before the assignment due date (Monday November 7 at 11:59 PM), you will turn in:

1) A ZIP file (named as your OCCS username) containing:
    a. Your source code
    b. A Makefile for compiling your source code
    c. A README file
2) Your technical report as a PDF file, named the same as your ZIP file.

Your Makefile must be able to compile your source code into an executable program that behaves as described above. Your README file should describe the different source code files used by your program, as well as instructions for running your program and finding its output file(s).

Your technical report should contain:

- An introduction describing the assignment and the contents of the report (provide the reader with the background needed to understand the rest of the report)
- A description of your implementation for both parts (what did you create?)
- A description of your experimental setups for both parts (what did you run and for what purpose?)
- A discussion of the results from both parts (what did you find, why did you find that, and what are the implications?)
- A conclusion summarizing the report and assignment
- An estimate of the total time spent on this assignment (broken down into the two parts)

Grading

The homework will be graded as follows:

- Snapshot 1: 5%
- Snapshot 2: 5%
- Implementation Correctness and Documentation: 50%
- Report: 40%

Honor Code

Each student is to complete this assignment individually or with a single partner. Since the assignment is a mini-project in scope, students are encouraged to collaborate with one another to discuss the abstract design and processes of their implementations. For example, please feel free to discuss the pseudocode for each learning algorithm to help each other work through issues understanding exactly how the learning algorithms work. You might also want to discuss the processes used to generate the training and test sets from the read in data set. Or, you might need to discuss how to work with the input and output files.

At the same, since this is an individual or small group assignment, no code can be shared between small groups, nor can students look at other groups' code. All discussions between small groups should be limited to abstract details and not implementation-specific concerns. Furthermore, the source code of existing machine learning libraries (e.g., Weka for Java, scikit-learn for Python) must not be consulted. Any violation of the above will be considered an Honor Code violation.

If you have any questions about what is permissible and what is not, please discuss with the professor. Please also feel free to stop by office hours to discuss the homework assignment if you have any questions or concerns.