Computer Science 3113: Artificial Intelligence
University of Arkansas – Fort Smith
Spring 2023

Problem Set 2: Naive Bayes Algorithm

50 points

Version 1.0

due Thursday, 23 February 2023, by 11:00 PM CT

Review the following assignment in its entirety prior to beginning. All submissions will be managed from within the course website.

Application Development

For this problem set, you are going to implement a classification algorithm in a supervised machine learning task. Begin by creating a class named `UASimpleClassifier.java`. Two files named `train.txt` and `test.txt` will be provided that are of the following form:

```
feature1,feature2,feature3,feature4,feature5,class
```

The first three features are discrete data; `feature4` and `feature5` data are continuous, and the `class` field will be a string. Your goal is to train a classification algorithm based on the `train.txt` (use the frequencies or values to determine probabilities). Next, you will need to use the following formula for the purpose of classification:

$$\hat{c} = \underset{j \in i \ldots k}{\arg\max} \frac{\Pr(C_j) \prod_{i=1}^{n} \Pr(f_i \mid C_j)}{\sum_{k}^{|C|} \Pr(C_k) \prod_{i=1}^{n} \Pr(f_i \mid C_k)}$$

For some feature X that follows some normal distribution $X \sim \mathcal{N}(\mu, \sigma^2)$, it will have the following probability distribution function (p.d.f.):

$$\text{PDF}(X) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

After you have trained your algorithm based on the training data, you will then classify each of the records in the `test.txt` file. Since these records are also labeled, you will calculate the accuracy of your results.

You will need to implement the following functions in your class:

1. `void train(String filename)` – this will read the contents of the `filename` file and perform any processing that is necessary to train your model. The file will be in the form of as described above (do not hard code the filename and/or path).

2. `void test(String filename)` – this will read the contents of the `filename` file. For each line in the file, the algorithm with output the features, the correct class, and the predicted class along with `CORRECT` or `INCORRECT` to determine whether the classification effort was successful. The algorithm will also report the accuracy (e.g. 88.12%) at the conclusion of the file processing. The file will be in the form of as described above (do not hard code the filename and/or path). The output should be formatted using fixed width fields using the `printf()` method; each field should be fixed width of 10 characters while floating a decimal points to three digits of precision (0.###). A sample output should appear as follows.

```
F1   F2   F3   F4       F5       CLASS    PREDICT  PROB    RESULT
---  ---  ---  -------  -------  -------  -------  -----   ----------
a    T    Z    1.000    100.000  class1   class1   81.2%   CORRECT
b    F    Z    2.000    200.000  class1   class2   35.2%   INCORRECT


Total Accuracy:    1 correct / 2 total  =   50.00% Accuracy
```

3. `String classify(String f1, String f2, String f3, double f4, double f5)` – this will accept five features as provided in the files and classify the result based on the trained model. The correct class should be returned.

4. Your `main()` method should accept three command line arguments:

   a) $\texttt{arg}_1$ – full path to a file containing the training data

   b) $\texttt{arg}_2$ – full path to a file containing the testing data

   c) $\texttt{arg}_3$ – full path to a file containing the values to predict (each line contains f_1, f_2, f_3, f_4, and f_5, comma-delimited)

   Your main method should train your algorithm, run the testing function, and finally predict each entry in the file provided for arg_3.

For each phase, you will need to call the appropriate function and output the following data based on the input from the files. The following should be executed from the command line:

```
java UASimpleClassifier /path/train.txt /path/test.txt /path/predict.txt
```

The following represents the output from the command line arguments:

```
************************************************************
Problem Set:  Problem Set 3:  Naive Bayes Algorithm
Name:         Andrew Mackey
Synax:        java UASimpleClassifier  arg1  arg2  arg3
************************************************************

Training Phase:  /path/to/training/file  <-- use real filename
------------------------------------------------------------
    => Number of Entries (n):              10000
    => Number of Features (p):             5
    => Number of Distinct Classes (y):     2

Testing Phase:
------------------------------------------------------------

F1   F2   F3   F4       F5       CLASS    PREDICT  PROB    RESULT
---  ---  ---  -------  -------  -------  -------  -----   ----------
a    T    Z    1.000    100.000  class1   class1   81.2%   CORRECT
b    F    Z    2.000    200.000  class1   class2   35.2%   INCORRECT

    Total Accuracy:    1 correct / 2 total  =   50.00% Accuracy

    => Number of Entries (n):              5000

Prediction Phase:
------------------------------------------------------------

     F1   F2   F3   F4       F5          PREDICT    PROB
     ---  ---  ---  -------  ---------   ---------  -----
     a    T    Z    1.000    100.000     class1     72.5%

    => Number of Entries (n):              10
```

5. A presentation will be conducted to demonstrate your results. Construct the report template provided on the course website. It should have the appropriate detail and supporting information regarding the implementation of your algorithm. You will need to find two citations from a top-tier journal (feel free to ask for recommendations if needed) that uses a Naive Bayes algorithm in their experiment. You should report both the experiment and the results in your work. The papers should be cited in your report using the IEEE format and a bibliography in LaTeX. (*Note: this is actually very simple. If you are not familiar with the tool, simply ask and I will be happy to assist you.*)

Deliverables

You will be responsible for delivering the following items:

1. Latex Documents – you will need to submit your report as the first section of the document. The next section should include the output of a successful execution of your program (based on the example provided).

2. Application Code – be sure to submit your source code as indicated above to the `code` server. A copy of this should also be uploaded to the course website. All files should be contained in a directory named `ps#` in your home directory (all lower case). All submitted code must have your 1) name 2) username (code server) 3) problem set number and 4) due date as a comment at the top of each class.

   ```
   /*******************************
   Name:           Andrew Mackey
   Username:       ua12345
   Problem Set:    PS1
   Due Date:       Month day, YEAR
   *******************************/
   ```

   Be sure to remove any extraneous code that is unnecessary prior to submission.

3. In your submission to the course website, be sure that you upload two files: a PDF and a zip file named `PS3LastNameFirstname.zip` (e.g. `PS3MackeyAndrew.zip`). The zip file should contain a single folder with the same name (e.g. `PS3MackeyAndrew`). Within this folder, add `UASimpleClassifier.java` file, `train.txt`, `test.txt`, and `predict.txt` files. The `predict.txt` file you will have to build yourself and it should contain 5 entries similar to the testing data (representing more than one target class).