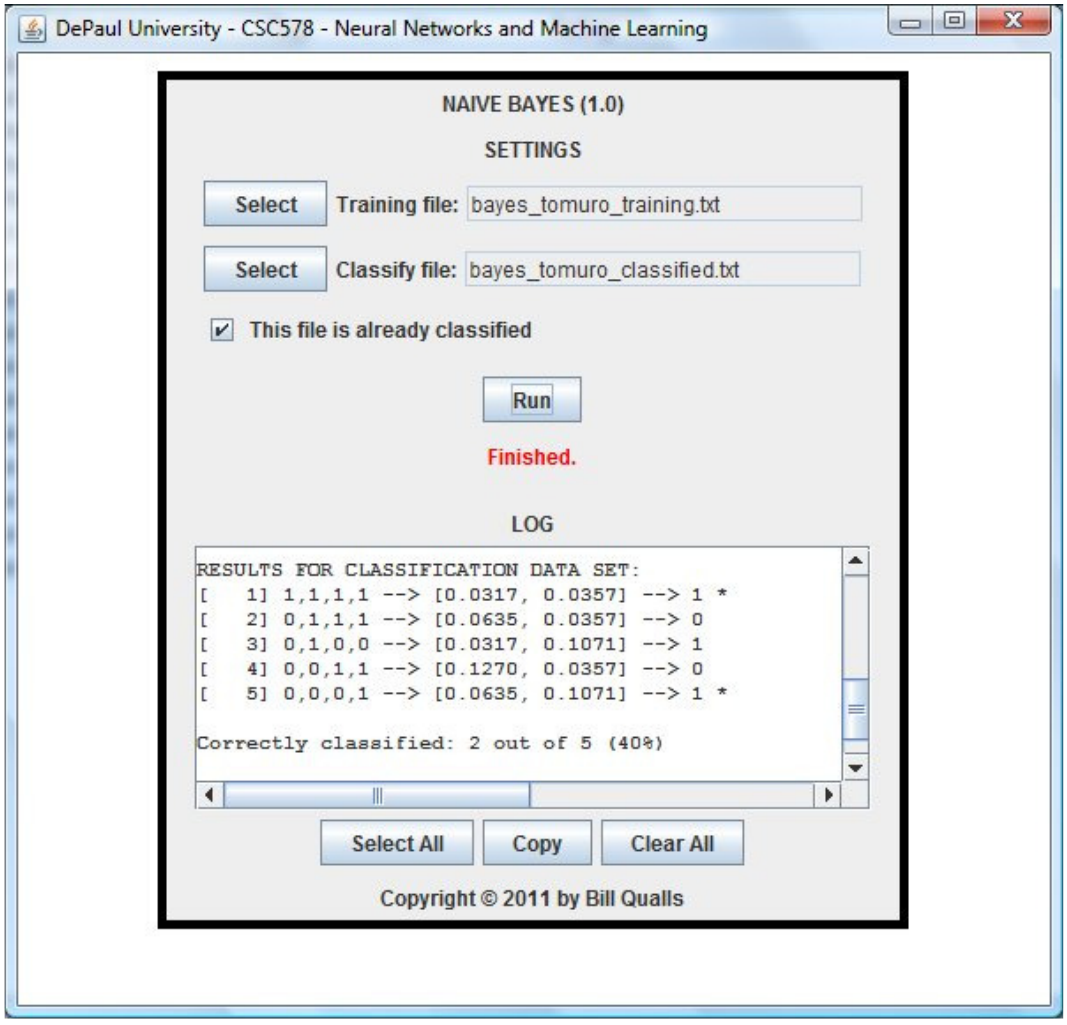
Bill Qualls

Installation

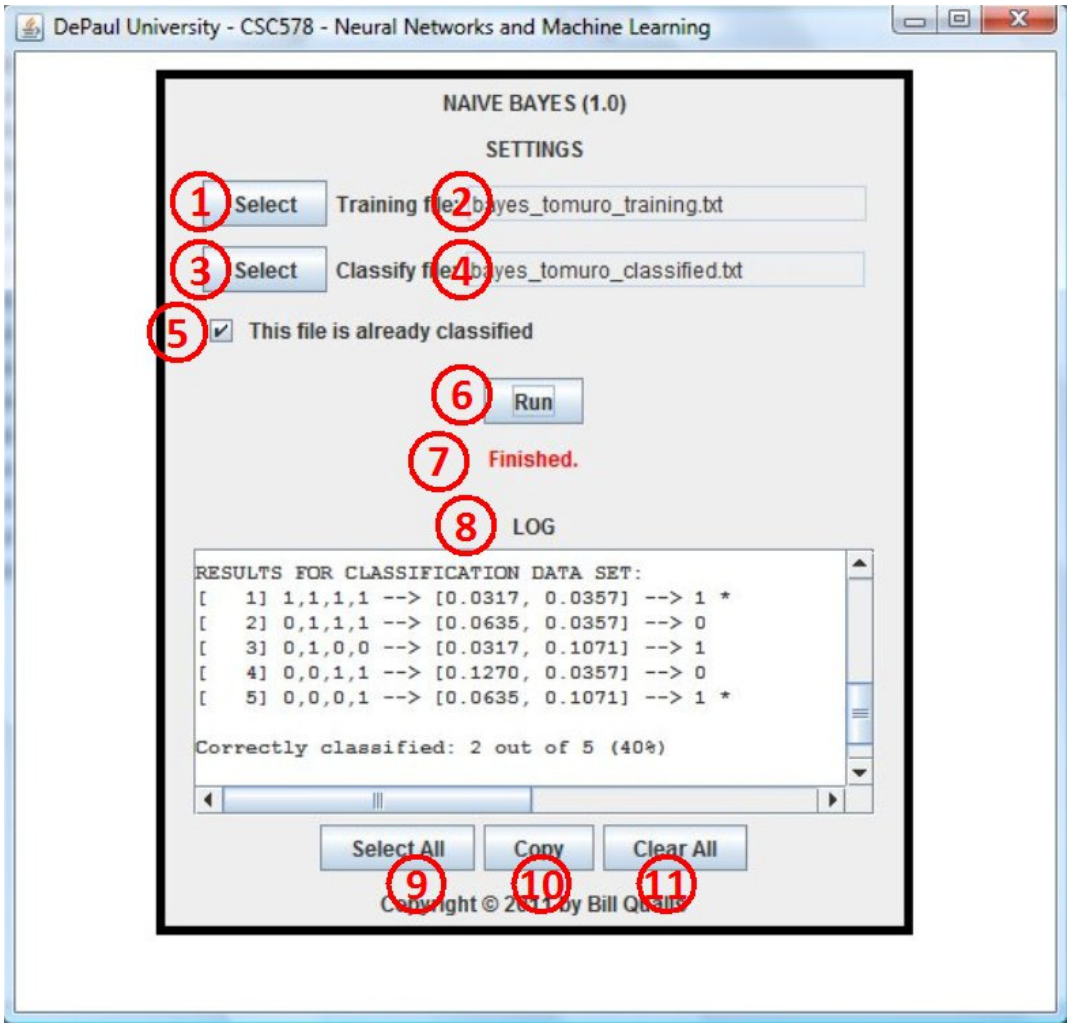
My program was written in Java and developed using Eclipse running under the Vista operating system. The accompanying zip file contains an executable jar file (**qualls_naive_bayes.jar**), or you can use the three source files (**NaiveBayes.java**, **NaiveBayesApp.java**, and **NaiveBayesInterface.java**. NaiveBayesApp.java is a Java application: run it.)

Description of the system

The interface appears as follows:



An annotated interface follows:



Item	Description
1	Click here to choose the training file via a file chooser dialog.
2	Name of the chosen training file shows here.
3	Click here to choose the testing file via a file chooser dialog.
4	Name of the chosen testing file shows here.
5	Select the check box if the testing file is already classified.
6	Click the Run button to run the program through to completion.
7	Status and error messages appear here.
8	Scroll pane containing a log of inputs and outputs.
9	Click the Select All button to select the entire scroll pane.
10	Click the Copy button to copy the selected area within the scroll pane to the paste buffer. This allows the user to copy the scroll pane to Notepad, Word, etc.
11	Click here to clear the entire scroll pane. You might choose to do so after each run to make better use of the Select All (item 9) and Copy (item 10) buttons.

Sample Inputs

This is an example of an input **training** file. Note that if the first column is an asterisk, the like is treated as a comment:

```
*  BAYES EXAMPLE
*  Using data from Dr. Tomuro's homework #3, problem #2
*
1,0,0,0
0,1,1,0
0,0,1,0
1,0,0,1
0,0,1,1
0,1,0,1
1,1,0,1
```

This is an example of an input **testing** file which is *not* already classified:

```
1,1,1
0,1,1
0,1,0
0,0,1
0,0,0
```

This is an example of an input **testing** file which *is* already classified. The class is in the last position on each line:

```
1,1,1,1
0,1,1,1
0,1,0,0
0,0,1,1
0,0,0,1
```

Sample Outputs

This is an example of the listing which appears in the log. In this example the testing file is *not* already classified:

```

*** Begin list of training file bayes_tomuro_training.txt ***
*  BAYES EXAMPLE
*  Using data from Dr. Tomuro's homework #3, problem #2
*
[   1] 1,0,0,0
[   2] 0,1,1,0
[   3] 0,0,1,0
[   4] 1,0,0,1
[   5] 0,0,1,1
[   6] 0,1,0,1
[   7] 1,1,0,1
```

```
*** End list of training file bayes_tomuro_training.txt ***

Training file has 7 rows and 2 attributes.

RESULTS FOR CLASSIFICATION DATA SET:
[   1] 1,1,1 --> [0.0317, 0.0357] --> 1
[   2] 0,1,1 --> [0.0635, 0.0357] --> 0
[   3] 0,1,0 --> [0.0317, 0.1071] --> 1
[   4] 0,0,1 --> [0.1270, 0.0357] --> 0
[   5] 0,0,0 --> [0.0635, 0.1071] --> 1
```

This is an example of the listing which appears in the log. In this example the testing file *is* already classified. Lines with the correct classification are marked with an asterisk. The number and percent correctly classified is also shown:

```
*** Begin list of training file bayes_tomuro_training.txt ***
*  BAYES EXAMPLE
*  Using data from Dr. Tomuro's homework #3, problem #2
*
[   1] 1,0,0,0
[   2] 0,1,1,0
[   3] 0,0,1,0
[   4] 1,0,0,1
[   5] 0,0,1,1
[   6] 0,1,0,1
[   7] 1,1,0,1
*** End list of training file bayes_tomuro_training.txt ***

Training file has 7 rows and 2 attributes.

RESULTS FOR CLASSIFICATION DATA SET:
[   1] 1,1,1,1 --> [0.0317, 0.0357] --> 1 *
[   2] 0,1,1,1 --> [0.0635, 0.0357] --> 0
[   3] 0,1,0,0 --> [0.0317, 0.1071] --> 1
[   4] 0,0,1,1 --> [0.1270, 0.0357] --> 0
[   5] 0,0,0,1 --> [0.0635, 0.1071] --> 1 *

Correctly classified: 2 out of 5 (40%)
```

Other datasets: Meisner

The dataset in the following example was taken from "Naïve Bayes Classifier Example" by Eric Meisner, retrieved from http://www.inf.u-szeged.hu/~ormandi/ai2/06-naiveBayes-example.pdf on November 10, 2011. My calculated values differ from his because he included another parameter in his formula for V_{nb}; specifically, *m* for equivalent sample size. Nevertheless, we had the same conclusion.

```
*** Begin list of training file bayes_meisner_training.txt ***
* Source: http://www.inf.u-szeged.hu/~ormandi/ai2/06-naiveBayes-example.pdf
*
* Variable 1 = Color
*   0 = Red
*   1 = Yellow
* Variable 2 = Type
*   0 = Sports
*   1 = SUV
* Variable 3 = Origin
*   0 = Domestic
*   1 = Imported
* Objective = Stolen
*   0 = No
*   1 = Yes
*
* Given data:
* 1 Red Sports Domestic Yes
* 2 Red Sports Domestic No
* 3 Red Sports Domestic Yes
* 4 Yellow Sports Domestic No
* 5 Yellow Sports Imported Yes
* 6 Yellow SUV Imported No
* 7 Yellow SUV Imported Yes
* 8 Yellow SUV Domestic No
* 9 Red SUV Imported No
* 10 Red Sports Imported Yes
*
* Test case:
* "We want to classify a Red Domestic SUV"
*
* Answer:
* "Since 0.069 > 0.037, our example gets classified as 'NO'"
*
[   1] 0,0,0,1
[   2] 0,0,0,0
[   3] 0,0,0,1
[   4] 1,0,0,0
[   5] 1,0,1,1
[   6] 1,1,1,0
[   7] 1,1,1,1
[   8] 1,1,0,0
[   9] 0,1,1,0
[  10] 0,0,1,1
*** End list of training file bayes_meisner_training.txt ***

Training file has 10 rows and 2 attributes.

RESULTS FOR CLASSIFICATION DATA SET:
[   1] 0,1,0,0 --> [0.0720, 0.0240] --> 0 *

Correctly classified: 1 out of 1 (100%)
```

Other datasets: My own spam detector

This dataset is my own creation and implements a simple spam detector:

```
*** Begin list of training file bayes_spam_training.txt ***
* SPAM DETECTION
* Fictitious dataset by Bill Qualls
*
* ATTRIBUTES (WORD FOUND IN LETTER)
*  - Please
*  - Provide
*  - Pay
*  - Penis
*  - Pleasure
*
* CLASSIFICATION
*  - Spam
*
[   1] 1,0,0,0,0,0
[   2] 1,0,1,0,0,0
[   3] 0,0,0,0,1,0
[   4] 0,1,1,0,0,0
[   5] 1,0,0,0,1,0
[   6] 1,1,1,0,0,1
[   7] 1,0,0,1,1,1
[   8] 1,0,1,0,1,1
[   9] 0,1,0,0,1,1
[  10] 0,1,0,1,1,1
*** End list of training file bayes_spam_training.txt ***

Training file has 10 rows and 4 attributes.

RESULTS FOR CLASSIFICATION DATA SET:
[   1] 1,1,0,1,0,1 --> [0.0000, 0.0086] --> 1 *
[   2] 1,0,0,0,1,0 --> [0.0576, 0.0346] --> 0 *
[   3] 0,0,0,0,0,0 --> [0.0576, 0.0058] --> 0 *
[   4] 1,1,1,1,1,1 --> [0.0000, 0.0230] --> 1 *
[   5] 0,1,1,0,1,1 --> [0.0064, 0.0230] --> 1 *

Correctly classified: 5 out of 5 (100%)
```

Ideas for enhancement

- The current program allows for boolean attributes only. I would like to modify the program to allow for continuous attributes as well.

Attachments

- Source code listings for NaiveBayes.java, NaiveBayesApp.java, and NaiveBayesInterface.java