CS 4375 Introduction to Machine Learning
Fall 2021
Assignment 2: Bayesian Learning
Part I: *Due electronically by Thursday, September 23, 11:59 p.m.*
Part II: *Due electronically by Thursday, September 30, 11:59 p.m.*

Note: 1. Your solution to this assignment must be submitted via eLearning.
2. Whenever possible, you should provide brief justifications for your solution.
3. You may work in a group of two or individually.

Part I: Written Problems (70 points)

1. **Probability I (14 points)**

   Suppose that A and B are binary-valued random variables. We know that $P(A)$ = 0.6, $P(B)$ = 0.7, and $P(B|A)$ = 0.8.

   (a) What is $P(A|B)$?

   (b) What is $P(B| \sim A)$?

   Show and explain your work for both (a) and (b).

2. **Probability II (14 points)**

   Consider the following events:

   - E: a family with three children has children of both sexes
   - F: a family with three children has at most one boy

   Assuming that the eight ways a family can have three children are equally likely, are E and F independent? Show your work.

3. **Probability III (10 points)**

   What is the minimum number of people who need to be in a room so that the probability that at least two of them have the same birthday is greater than 0.5? Assume that (1) the birthdays of the people in the room are independent; (2) each birthday is equally likely; and (3) there are 366 days in the year. Show your work.

4. **Hypothesis Selection (16 points)**

   Consider a medical diagnosis problem in which there are two alternative hypotheses: (1) that the patient has a particular form of cancer, and (2) that the patient does not. The available data is from a particular laboratory test with two possible outcomes: *positive* and *negative*. We have prior knowledge that over the entire population of people only 0.008 have this disease. Furthermore, the lab test is only an imperfect indicator of the disease. The test returns a correct positive result in only 98% of the cases in which the disease is actually

present and a correct negative result in only 97% of the cases in which the disease is not present. In other cases, the test returns the opposite result.

(a) Suppose we now observe a new patient for whom the lab test returns a positive result.

(i) What is the MAP hypothesis? Show your work.

(ii) What is the maximum likelihood hypothesis? Show your work.

(b) Suppose the doctor decides to order a second lab test for the same patient, and suppose the second test returns a positive result as well. Assume that the two tests are independent.

(i) What is the MAP hypothesis? Show your work.

(ii) What is the maximum likelihood hypothesis? Show your work.

5. **Bayesian Classifiers (16 points)**

We want to learn the concept of `graduate student`. Consider the following set of training examples.

Class	good hacker?	has publications	hobby
positive	yes	yes	sci-fi
negative	yes	no	music
positive	no	yes	sci-fi
positive	yes	no	tennis
negative	no	yes	music
positive	no	no	tennis
positive	yes	yes	sci-fi
negative	yes	yes	tennis
positive	no	no	sci-fi
negative	no	yes	tennis

(a) How would a naive Bayes classifier trained on this training set classify the following instances? Show your work.

(i) `hobby`=tennis $\wedge$ `good-hacker`=yes $\wedge$ `has-publication`=yes

(ii) `hobby`=tennis $\wedge$ `good-hacker`=yes $\wedge$ `has-publication`=no

(b) How would a Bayes classifier trained on this training set classify the following instances? Show your work.

(i) `hobby`=tennis $\wedge$ `good-hacker`=yes $\wedge$ `has-publication`=yes

(ii) `hobby`=tennis $\wedge$ `good-hacker`=yes $\wedge$ `has-publication`=no

Part II: Programming (30 points)

In this problem, you will implement the Naive Bayes learning algorithm for binary classification tasks (i.e. each instance will have a class value of 0 or 1). To simplify the implementation, you may assume that all attributes are binary-valued (i.e. the only possible attribute values are 0 and 1) and that there are no missing values in the training or test data.

A sample training file (`train.dat`) and test file (`test.dat`) are available from the assignment page of the course website. In these files, only lines containing non-space characters are relevant. The first relevant line holds the attribute names. Each following relevant line defines a single example. Each column holds this example's value for the attribute named at the head of the column. The last column (labeled "class") holds the class label for the examples.

IMPORTANT:

- To implement the Naive Bayes classifier, you may use Python, C++, or Java to implement the Naive Bayes algorithm. Do **not** use any non-standard libraries in your code, and make sure your program can compile on UTD machines, not just your own machines. If you are using:
  - **Python**
    - Make sure that your primary source file is `main.py` and that your code runs successfully after executing `python main.py <args>`.
  - **C++**
    - Make sure that your primary source file is `main.cpp` and that your code runs successfully after executing `g++ main.cpp` and `./a.out <args>`.
  - **Java**
    - Make sure that your primary source file is `Main.java` and that your code runs successfully after executing `javac Main.java` and `java Main <args>`.
- Your program should be able to handle **any** binary classification task with **any** number of binary-valued attributes. Consequently, both the number and names of the attributes, as well as the number of training and test instances, should be determined at runtime. In other words, these values should **not** be hard-coded in your program.
- Your program should allow exactly two arguments to be specified in the command line invocation of your program: a training file and a test file. Your program should take these two arguments in the same order as they are listed above. There should be no graphical user interface (GUI). Any program that does not conform to the above specification will receive no credit.
- In the input files, only lines containing non-space characters are relevant, as mentioned previously. In particular, empty lines may appear anywhere in an input file, including the beginning and the end of the file. Care should be taken to skip over these empty lines.

What to Do

1. Train the Naive Bayes learning algorithm on the training instances. Print to `stdout` the parameters of the classifier that were estimated from the training data (i.e., the class priors and the class-conditional probabilities). To exemplify, assume that the class attribute is named C and has two possible values, c1 and c2; furthermore, assume that there are three attributes A1, A2, and A3, where A1 has two possible values (x1 and x2), A2 has two possible values (y1 and y2), and A3 has three possible values (z1, z2, and z3). The learned parameters should be printed according to the following format:

   P(C=c1)=0.32 P(A1=x1|c1)=0.33 P(A1=x2|c1)=0.67 P(A2=y1|c1)=0.25 P(A2=y2|c1)=0.75 P(A3=z1|c1)=0.26 P(A3=z2|c1)=0.49 P(A3=z3|c1)=0.25
   P(C=c2)=0.68 P(A1=x1|c2)=0.22 P(A1=x2|c2)=0.78 P(A2=y1|c2)=0.91 P(A2=y2|c2)=0.09 P(A3=z1|c2)=0.16 P(A3=z2|c2)=0.74 P(A3=z3|c2)=0.10

   In other words, all the probabilities associated with a particular class should appear in the same line.

2. Use the learned classifier to classify the **training** instances. Print to `stdout` the accuracy of the classifier. The accuracy should be computed as the percentage of examples that were correctly classified. For example, if 86 of 90 examples are classified correctly, then the accuracy of the classifier would be 95.6%.

   ```
   Accuracy on training set (90 instances):  95.6%
   ```

3. Use the learned classifier to classify the **test** instances. Print to `stdout` the accuracy of the classifier.

   ```
   Accuracy on test set (10 instances):  60.0%
   ```

What to Submit

Submit **via eLearning** (i) your source code, and (ii) a README file that contains instructions for **compiling** and **running** your program (as well as the platform (Windows/Linux/Solaris) on which you developed your program). Each group should hand in a single copy of the code. The names of all the members of the group should appear in the README file and in the Comments box on the eLearning submission page. Again, you will receive **zero credit** for your program if (1) we cannot figure out how to run your program from your README file or (2) your program takes more or less than two input arguments. We will likely run your program on a new data set (with a different number of attributes) to test your code, so we encourage you to do the same!