Instructions for Experimenter

**Introduction to Bayes Nets**

Bayes nets are a type of machine learning model that allow a user to make predictions about the probability of a particular event occurring based on previous events. In this tutorial, we will study the behavior of a student attending an 8 AM tutoring session while familiarizing ourselves with the *Bayes Net Debugger* program. The program allows you to build a Bayes Net graph to represent your machine learning model, then run examples through the graph to see how well your model classifies test data points.

**The Problem**

Alice is a freshman at MIT who needs extra tutoring for 8.01 (physics). However, the only time the tutor is available to meet is at 8 in the morning on Sundays. Attending sessions at 8 AM may be tough for Alice for several reasons: she often goes out partying Saturday nights and thus goes to bed really late, she is not a morning person and is often too lazy to get out of bed at an early time, she is taking a rigorous schedule at MIT so sometimes she would rather do other work than attend tutoring sessions. Our job is to help the tutor figure out the probability that Alice will attend the tutoring session because the tutor does not want to get out of bed at 8 AM if she doesn’t have to. We can figure this out by looking at how Alice behaved in past situations and use conditional probabilities based on other events to calculate the probability that Allice will attend the next session.

**Tutorial**

To solve this problem we will first build a graphical representation of the problem, and then run the test data through the graph. The goal is to achieve the highest test accuracy possible. Follow the next set of steps in order to familiarize yourself with *Bayes Net Debugger*:

1. **Upload the data to the program.** Click “choose file” at the bottom left on the program and select the appropriate data file (Alice.csv). Then click the “Upload Data” to upload the data to the program.
2. **Create a node** for the graph by dragging the oval-shaped node from the toolbar into the graph-building section (big square). Double click the node to rename it as a feature which may contribute to Alice not showing up to her tutoring session. **Let’s begin by renaming the node “Attended 8AM Session”**.
   1. **Let’s also create a second node and name it “Partied Night Before”.**
3. **Highlight a node by Ctrl+clicking the node.** You can perform many different operations with a highlighted node. Unhighlight a node by also pressing Ctrl+click.
4. In order to view how the data looks with this graphical model, we can animate the nodes. The nodes will change color (from white) if the feature is true at that data point. The nodes will turn the same color if all features are true. This is helpful because it will allow you to visualize the data. If one feature only seems to be true when another feature is true, then it is likely that there is some correlation between the two features. To animate nodes, highlight the chosen nodes and then click “Start Animation”. You can even select the speed at which the colors flash by sliding the bar below the “Stop Animation” button. **Animate the nodes “Attended 8 AM Session” and “Partied Night Before”. Do you suspect that the two events are correlated to one another?**
5. Connect nodes on the graph by clicking the “Connect Mode” button. Connect mode is ON when the “connect mode” button is gray. Once in connect mode, select the parent node by clicking once on any node. After the parent has been chosen, move the mouse to the child node (a line will be drawn as you move your mouse over). When the mouse enters the child node, the line is set and the parent-child relationship between the two nodes is established. **Connect the nodes “Partied Night Before” and “Attended 8AM Session”, with “Partied Night Before” being the parent node. Exit connect mode when finished.**
6. Disconnect nodesby highlighting the two nodes you want to disconnect, then pressing the “Disconnect Highlighted Nodes” button.
7. Let’s move our attention over to the testing modules to the far right. The first module allows you to specify values for nodes in the graph. It will then, for each node without an assigned value, calculate the probability that the node’s value is true and randomly assign a value to that node based on the probability. **Assign a value of true to the “Partied Night Before” feature and a value of “not given” to the “Attended 8AM Session” feature. Set the Result Node, which is the feature that we build this model for, to “Attended 8AM Session”. Click “Get Results” to step through the program and get a value for each node.** **Click “clear results: to clear all values from the nodes. Try using this module again except this time set “Partied Night Before” to false. Observe how the probability of being true changes for the “Attended 8AM Session”.**
8. Once your model is built to your satisfaction, **compute the test accuracy for your model.** Focus your attention to the “Test Accuracy” section at the bottom right of the program. Choose the “AliceTest.csv” file and click “Upload Test Data”. **Choose the result node which you want your model to make predictions for: “Attended 8AM Session”**. You can quickly run through all tests at once (and select the speed at which you do so by sliding the bar below) by clicking “Run All Tests”. Or you can step through the test points one at a time using the “Single Step Through Tests” button. What is the test accuracy? Do you think it can be improved?
9. If the test accuracy does not appear to your liking, you can also split and merge nodes. **Try splitting the node “Partied Night Before” into two nodes by highlighting the partying node and clicking “Split Highlighted Nodes” (you can also double click a highlighted node to split). Rename the split nodes to “Drinking” and “Not Drinking”. See if this improves your results.**
10. If splitting the nodes does not improve your results, merge the two nodes back together by highlighting them and clicking “Merge Two Highlighted Nodes”.
11. **Try adding more nodes to see if you can increase the test Accuracy.**

**The Second Problem**

Now that you have a feel for how the program works, we present you with a second problem:

There are many factors that determine whether a plane is able to land at its destination. Some factors are the weather, and the airline’s ability to stay organized. Imagine that you have a friend who is supposed to land at Boston Logan Airport. You need to pick your friend up from the airport, so you want to know ahead of time whether or not his/her plane will land. Unfortunately, the airline doesn’t give you the status of the plane, so you must pay attention to other factors in order to make your decision. To make your decision, you will build a Bayes Net. An initial Bayes net is built for you (shown below). Training data is provided that will allow you to compute the probability of the plane landing based on other events. Please look at the file, planeLandData.csv for training data and testData.csv for test data.