Homework 9

Introduction to AI

For this homework, you can work on your own or with a partner.

You will be completing some Python code that implements Bayesian networks, including exact and approximate inference.

“TODO” tags in the code give instructions on the parts you are to complete. There is not a lot of code for you to write, but it will take substantial time to understand the existing code, the algorithms, and to think about the code you should write.

The code is commented with information about each class and method, but here is a brief overview of the major parts:

* bayesnets.py is where you will be writing code. It contains BayesianNode, BayesianNetwork, and GibbsSampler objects.
  + BayesianNode is complete, including member variables for the random variable’s name, domain (i.e., “values”), parents in the network, and the conditional probability table. The underscores in front of the \_\_cpt member variable indicate that it is private and should only be accessed from within the class. Thus, get CPT information using the get\_probability method.
  + BayesianNetwork is where you will be completing several methods. The first section of the class deals with exact inference. Then there is a comment  
    ##################### Exact inference is above

##################### Approximate inference is below

With the approximate inference methods below it.  
The TODO tags tell you what code to complete.

* + GibbsSampler is a small class. The constructor uses exact inference to compute gibbs\_tables, which are like CPTs, but instead have the distribution of a variable given its Markov blanket, not just its parents. Precomputing these enables quicker sampling of variables given their Markov blanket. You will implement the Gibbs sampling in the get\_next\_sample method.
* test\_bayesnet.py contains some unit tests for the exact inference part of the BayesianNetwork. Run these after you complete each part of exact inference code in BayesianNetwork, so you can ensure it is working correctly before proceeding.
* run\_experiments.py contains a function to load a BayesianNetwork from a file (build\_net) and functions to run experiments with the different approximate inference procedures. Before you run the experiments, (1) install pandas and plotly Python packages (e.g., with pip); (2) complete the approximate inference code in bayesnets.py. There is a Boolean variable at the top that you should set to True when initially testing the code, as it runs a shorter set of experiments. For the final analysis, remember to set it back to False to run the full set of experiments.

After you run run\_experiments.py, create a Microsoft Word document (or PDF) that includes answers to the following questions and the plots you used to answer those questions. You might need to zoom in on some of the plots that the code generates to see meaningful differences, as outliers can stretch the axes limits. In your report, use the zoom level that best supports your conclusion.

1. For common upstream evidence, which sampling method would you say is best? Briefly explain why (i.e., one or two phrases or sentences).
2. For rare upstream evidence, which sampling method would you say is best? Briefly explain why (i.e., one or two phrases or sentences).
3. For rare downstream evidence, which sampling method would you say is best? Briefly explain why (i.e., one or two phrases or sentences).
4. For common downstream evidence, which sampling method would you say is best? Briefly explain why (i.e., one or two phrases or sentences).

# Hints

* I strongly recommend the Mypy Type Checker extension for VSCode, or similar type checking in your preferred IDE. This lets you know if the variables you are building and passing to functions have the proper types.
* Think before you write. As mentioned above, there is not much code to write. The challenge in this assignment is understanding the algorithm and the existing code well enough to know what code you need to write.
* Some of the code blocks you are to complete are very similar to nearby chunks of code, so it is worth reading and understanding the surrounding code.
* Work on exact inference first and get that passing the test cases. Then move to approximate inference.
* The experiments code will take a while to run (approx. 15-25 minutes). Don’t wait until the last minute to finish the code, since you need time to run the experiments, then answer the analysis questions.
* Understand the code that you submit. Code from this project might appear on the final exam, so even if your partner writes some of the code, make sure you understand what it does and why it works.

# Handin

Turn in two files on D2L:

* Your complete bayesnet.py (with a comment at the top saying your name(s)) and
* the Microsoft Word document or PDF containing your answers to the analysis questions and the plots you used to arrive at those conclusions.