

Homework 4 Coding

Due Sunday, April 3rd at 9pm ET

You are encouraged to discuss the assignment in general with your classmates, and may optionally collaborate with one other student. If you choose to do so, you must indicate with whom you worked. Multiple teams (or non-partnered students) submitting the same code will be considered plagiarism.

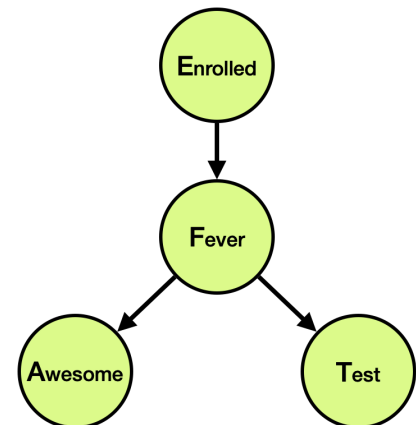
Code must be written in a reasonably current version of Python (>3.6), and be executable from a Unix command line. You are free to use Python's standard modules for data structures and utilities.

For this short coding assignment, you will estimate some of the probabilities associated with the 383 Fever Bayesian network. The bulk of the code for generating samples has been written for you and is contained in the file `bayes_fever.py`. The goal of the assignment is to help you get a feel for how these processes work, and understand how approximated answers compare to the exact ones.

383 Fever, Part III (co-starring Mr. T) (30 points)

Exposure to COMPSCI 383 is a known cause of “383 Fever”, whose main symptom is excessive awesomeness. Over the years, scientists have developed a simple blood test to help confirm clinical diagnoses.

*The causal Bayesian network DAG shown on the right describes the relationships between variables **Enrolled** (whether someone is taking 383), **Fever** (whether they have 383 Fever), **Awesome** (whether they are exhibiting an excessive amount of awesomeness), and **Test** (whether their blood sample tests positive). These are abbreviated **E**, **F**, **A**, and **T**, respectively.*



The above structure has been encoded in a list of `bayes_fever.BooleanVariableNode` objects called `FEVER_NODES`. However, the conditional probabilities have all been set to a value of 0.25. After you have completed the Primer exercises (or after the solutions have been posted), you should replace these values with the correct conditional probabilities. For this coding assignment, you must complete the code for calculating probabilities from samples and doing some plotting of results.

Step 1

Fill in the bodies of the three `get_prob()` member functions found in the `SimpleSampler`, `RejectionSampler`, and `LikelihoodWeightingSampler` classes. These methods should calculate an empirical probability estimate from a collection of samples generated by the associated `generate_samples()` member functions of each class (which have been written for you!). Be mindful of the different function parameters and return types.

Once you've filled those in correctly and run the script, the output will provide probability estimations for the quantities you've been dealing with in previous parts of the assignment. Take some time and see if your answers match, as well as the effect of changing the number of samples being generated by changing the value of `n` on line 207.

Step 2

Complete the `bayes_sample_size_plot()` function. This function should plot the probability estimates of $P(\text{enrolled} \mid \text{awesome}, \text{test})$ as a function of the number of samples, for both rejection sampling and likelihood weighting. You should call the provided `two_line_plot()` function to do the actual matplotlib plotting, but you'll need to fill in the code for creating the samples and collecting the x- and y-values to plot. Your script should produce a pdf called `bayes_fever.pdf`.

Step 3

There is no step 3.

What to Submit

You should submit your versions of `bayes_fever.py`, the plot file `bayes_fever.pdf`, and a `readme.txt` containing:

- Your name(s)
- Any noteworthy resources or people you consulting when doing your project
- Notes or warnings about what you got working, what is partially working, and what is broken