

Bayesian networks – assignment part 1

The goal of this assignment is to get a thorough understanding of inference in Bayesian networks. For this you have to implement the Variable Elimination algorithm for inference (computing marginal and posterior probabilities) and apply it on inference queries in different Bayesian networks. Keep a log (textfile) of the steps your algorithm does on a particular network with a particular elimination ordering according to the steps on the slides, e.g., what are the query/evidence variables, what are the factors, what is the elimination ordering, which factors are processed, etc.

There is code available in Java and Python for reading in a network from a file, and some structure to help you focus on the core of this assignment: efficiently implementing the algorithm using suitable data structures. The Bayesian Network repository (<http://www.bnlearn.com/bnrepository/>) gives you many examples of real-world (and less-real-world, but known benchmarks) networks. You can also construct example networks yourself. The example code uses the .bif format. We also attach a .bif of the Endorisk model.

This part of the programming assignment comes in two parts:

- In **1a** we ask you to implement multi-dimensional factors and do the necessary calculations on them (reduction, product, marginalization)
- In **1b** you build on this to implement variable elimination.

Bonus points are offered if your algorithm can process non-binary variables (1 bonus point extra) and if you experiment with different heuristics for the elimination order (e.g., least-incoming-arcs first, contained-in-fewest-factors-first, or others; also 1 bonus point). This can help you to achieve a high grade for this course.

To see if your algorithm's results are correct, you can compare them to the outcomes in the Jupyter-based AISpace2 (<https://aispace2.github.io/AISpace2/index.html>), or use a package such as pgmpy or similar for Java. The **programming language** for the task is:

1. Java (version 11 or later), or
2. Python (version 3.6 or later)

Choose informative names for the used variables, constants and functions. Use comments to make your code more accessible. Take care that your output clearly informs on the working of your algorithm. You may use libraries and pre-defined data structures etc. of course, as long as you implement the VE algorithm yourself. You may use (as example or as part of your program) the code for reading in Bayesian networks, but document where and how you used it and clearly comment chances to the source code.

Formative feedback

You are most strongly advised to hand in the code, and an accompanying readme document, before the deadline on Brightspace in order to get formative feedback on this part of the assignment. This will give you an opportunity to improve your work prior to formal assessment and get an indication of what is expected for a passing (or high) grade and where you are now. Instructions for the end product (combining parts 1, 2, and 3 of the assignment) will follow.

