School of Computer Science, University of St Andrews, CS5011, 2022-23

CS5011: A2 - Uncertainty - Bayesian Networks

Assignment: A2 - Assignment 2

Deadline: 8th of March 2023

Weighting: 25% of module mark

Please note that MMS is the definitive source for deadline and credit details. You are expected to have read and understood all the information in this specification and any accompanying documents at least a week before the deadline. You must contact the lecturer regarding any queries well in advance of the deadline.

---

1 Introduction & Objective

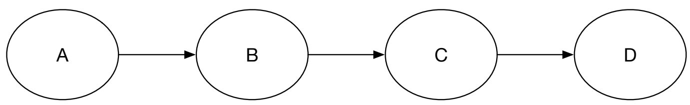
Bayesian networks (BNs) are a way of representing probabilistic relationships among a set of variables. BNs are probabilistic graphical models representing a set of random variables and their conditional dependencies by directed acyclic graphs. Seminal research in the context of Bayesian networks and causal reasoning has led Judea Pearl[1], one of the first pioneer of BNs, to being awarded the prestigious Turing Award in 2011[2]. In recent years, Bayesian networks have also seen interesting applications in machine learning (see Chapter 20 of Russel & Norvig[3]). In this practical, you will gain familiarity with modelling and making probabilistic inferences with Bayesian networks.

2 Competencies

- Develop understanding of probabilistic inferences.
- Develop and evaluate methods for inference in Bayesian networks.

3 Practical Requirements

3.1 Part 1 - Simple inference on a chain



	T	F
A	0.05	0.95

B

A	T	F
T	0.05	0.95
F	0.80	0.20

C

B	T	F
T	0.10	0.90
F	0.30	0.70

D

C	T	F
T	0.40	0.60
F	0.60	0.40

Figure 1: Example Network of a Chain structure

[1] `http://amturing.acm.org/bib/pearl_2658896.cfm`

[2] `http://amturing.acm.org/alphabetical.cfm`

[3] S. J. Russell and P. Norvig. Artificial Intelligence: A Modern Approach. Pearson Education, 3 ed., 2010.

The first task is to develop a program that is able to construct Bayesian networks of chain structure based on a given BN specification. An example of the BN is listed in Figure 1.

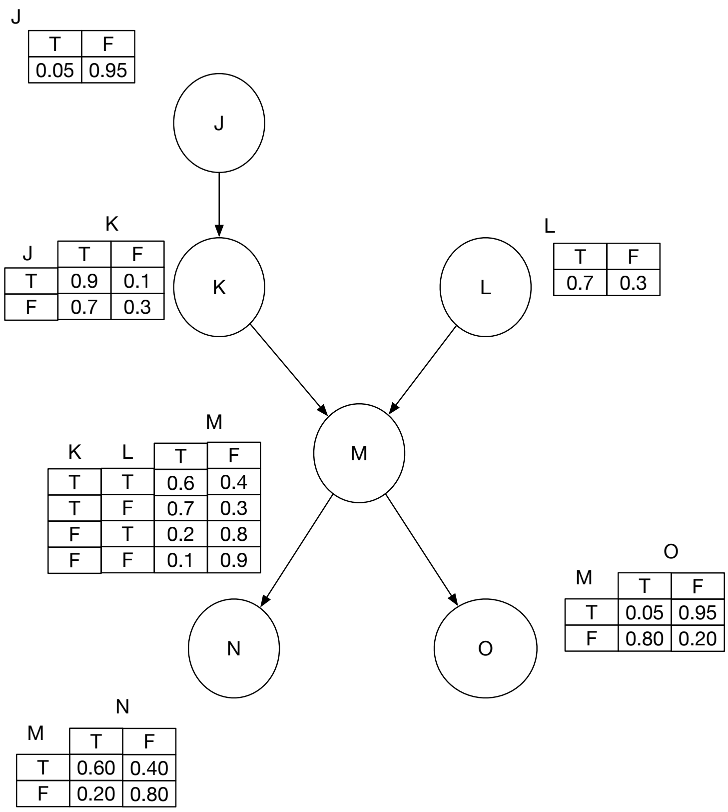
For this task, your system should:

- Construct a network based on a given BN's specification in an XML file (E.g., invoke the system with input argument '`BNA.xml`')
- Read the query as user input: a queried variable name followed by its truth value (T/F) separated by a colon (E.g., '`D:T`')
- Calculate the output (E.g., $P(D = true) = 0.57050$)
- Print the output without further information. Outputs should be formatted as numbers with 5 decimal places. (E.g., print '`0.57050`' only)

You should assume all variables are binary and the graph is a simple acyclic chain for this part. You may further assume the given Bayesian network is always valid (i.e., it does not contain cycles and has correct CPTs). You do not need to use Variable Elimination algorithm for this part if you choose not to.

3.2 Part 2 - Variable elimination without evidence

In this part, you are tasked with reading in BNs with a maximum branching factor of 2 (in other words, each node has at most two parents and/or two children) and making probabilistic inferences with variable elimination algorithm. An example of the BN is listed in Figure 2.



J

T	F
0.05	0.95

K

J	T	F
T	0.9	0.1
F	0.7	0.3

L

T	F
0.7	0.3

M

K	L	T	F
T	T	0.6	0.4
T	F	0.7	0.3
F	T	0.2	0.8
F	F	0.1	0.9

O

M	T	F
T	0.05	0.95
F	0.80	0.20

N

M	T	F
T	0.60	0.40
F	0.20	0.80

Figure 2: Example Network for Part 2

For this task, your system should:

- Construct a network based on the input specification (E.g., invoke the system with input argument '`BNB.xml`')
- Read the query as user input as in part 1
- Read the order of elimination as user input: a list of variable names separated by a comma (E.g., 'J,K,L')
- Calculate the output
- Print the output without further information. Outputs should be formatted as numbers with 5 decimal places.

You may assume again all variables are binary for this part.

3.3 Part 3 - General inference on a general BN

You are asked in this part to extend the variable elimination algorithm developed in part 2 to include evidence. For example, assume that we are querying the BNB network presented in Figure 2 and that we want to know the probability $p(J = True|N = False, O = True)$. The BNs considered in this section can be any directed acyclic graph (DAG) and the variables are not restricted to be binary. In other words, a general BN with the only restriction that the variables are discrete or categorical.

For this task, your system should:

- Read-in a network based on a given specification
- Read the query as user input as in part 1
- Read the evidence as user input (new part): a list of evidence variable names with their truth values (T/F) after a colon separated by a space (E.g., '`A:F B:T`')
- Calculate the output taking into consideration the evidence given
- Print the output without further information. Outputs should be formatted as numbers with 5 decimal places.

Note that Variable Elimination algorithm needs an elimination order. Your program needs to decide a proper order automatically.

3.4 Part 4 - Extension

It is strongly recommended that you ensure you have completed the requirements of part 1 to 3 before attempting any of these requirements. Acceptable extensions are:

- Validate a given BN. In other words, the given graph should be DAG and the given conditional probability tables are valid (non-negative and sum to one for each conditional distribution)
- Implement Gibbs sampling and Rejection sampling to make approximate inferences and compare the results with those obtained with the exact inference in the report

You need to test your extension extensively and provide testing evidence in the report. In other words, you need to convince and prove to the marker your extensions are correct and therefore worth the extra credit. Without sufficient evidence of testing, extensions will be ignored. No other extensions will be accepted unless they are agreed with the lecturer.

4 Code Specification

4.1 Code Submission

The program must be written in Java and your implementation must compile and run without the use of an IDE. Your system should be compatible with the version of Java available on the School Lab Machines (Amazon Corretto 17 – JDK17). Please note, you are not allowed to use libraries that implement Bayesian or graph algorithms, but other libraries that are secondary to the objectives of the practical can be used (e.g., Java Util). Your source code should be placed in a directory called **src/** and should include all non-standard external libraries. The code must run and produce outputs as described in the next sections.

Please note that code that does not adhere to these instructions may not be accepted.

4.2 Starter Code

For this practical, a simple starter class is provided in `A2main.java`. This class provides suggestions on how to read the required inputs and arguments. This class can be modified as required, however, please ensure that the final format of input and output matches that produced by the sample code.

4.3 Running

Your code should run using the following command:

```
java A2main <Pn> <NID> [<any other param>]
```

where `Pn` indicate the part of the practical `P1,P2,P3,P4` and `NID` indicates the network specification's file name, for example `BNA.xml, BNB.xml, BNC.xml`.

For example, for part 2 with BNA.xml we will use:

```
java A2main P2 BNA.xml
```

Then the user enters:

```
Query:
D:T
Order:
A,B,C
```

And the output should be:

```
0.57050
```

For part 2 to part 4, the output of the system must be as described as we will perform some automatic tests.

4.4 Automatic Testing

We provide some very basic unit tests to help you check that your input and output is structured with the required format. They can be found at
`/cs/studres/CS5011/Practicals/A2/Tests` and can be run with `stacscheck`.

In order to run the automated checker on your program, place your program in a directory in the School lab machines, ssh into one of the School computers running Linux if you are

[3] `https://aws.amazon.com/corretto/`

working remotely, then change directory to your CS5011 A2 directory and execute the following command:

```
stacscheck /cs/studres/CS5011/Practicals/A2/Tests
```

Please note that these tests are not sufficient to check that your program works correctly, please do test your system well and provide information on your own testing in the report.

5 Report

You are required to submit a report describing your submission in PDF with the structure and requirements presented in the additional document *CS5011_A_Reports* found on `studres`. The report includes 5 sections (Introduction, Design & Implementation, Testing, Evaluation, Bibliography) and has an advisory limit of 2000 words in total. The report should include clear instructions on how to run your code. For the evaluation, the description of each part of the practical discusses points to include.

6 Deliverables

A single ZIP file must be submitted electronically via MMS by the deadline. Submissions in any other format will be rejected.

Your ZIP file should contain:

1. A PDF report as discussed in Section 5
2. Your code as discussed in Section 4

7 Assessment Criteria

Marking will follow the guidelines given in the school student handbook. The following issues will be considered:

- Achieved requirements
- Quality of the solution provided
- Testing
- Insights and analysis demonstrated in the report

Some guideline descriptors for this assignment are given below:

- For a mark up to 8: the submission implements part 1, adequately documented or reported.
- For a mark up to 11: the submission implements part 1 and some attempt at part 2. The code submitted is of an acceptable standard, and the report describes clearly what was done, with good style.
- For a mark up to 14: the submission implements fully part 1 and part 2. It contains clear and well-structured code and a clear report showing a good level of understanding of Bayesian networks.

- For a mark 17 or higher: the submission implements fully parts 1 to 3. It contains clear, well-designed code, together with a clear, insightful and well-written report, showing in-depth understanding of design and inference of Bayesian networks.
- For a mark 19 and higher: the submission implements fully parts 1 to 3 and the extension(s). The submission should demonstrate unusual clarity of implementation, together with an excellent and well-written report showing evidence of extensive understanding of design and inference of Bayesian networks.

8 Policies and Guidelines

Marking: See the standard mark descriptors in the School Student Handbook

`https://info.cs.st-andrews.ac.uk/student-handbook/learning-teaching/feedback.html#Mark_-Descriptors`

Lateness Penalty: The standard penalty for late submission applies (Scheme B: 1 mark per 8 hour period, or part thereof):

`https://info.cs.st-andrews.ac.uk/student-handbook/learning-teaching/assessment.html#latenesspenalties`

Good Academic Practice: The University policy on Good Academic Practice applies:

`https://www.st-andrews.ac.uk/students/rules/academicpractice/`

Lei Fang, Fahrurrozi Rahman, Mun See Chang, Nguyen Dang

cs5011.lec@cs.st-andrews.ac.uk

February 14, 2023