School of Computer Science, University of St Andrews, CS5011, 2021-22

CS5011: A3 - Uncertainty - Bayesian Networks for Security

Assignment: A3 - Assignment 3

Deadline: 30th of March 2022

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 Objective

This practical aims to learn how to model and use Bayesian networks to reason with uncertainty.

2 Competencies

- Develop understanding of probabilistic inferences.
- Design and document a Bayesian network for expert diagnostic systems.
- Develop and evaluate methods for inferencing in Bayesian networks.

3 Practical Requirements

3.1 Introduction

Bayesian Belief networks (BNs) are a way of representing probabilistic relationships among a set of variables. They are often much more compact than the full joint probability distribution. They constitute an effective systematic way to represent uncertain representation of the world. Bayesian networks are probabilistic graphical models representing a set of random variables and their conditional dependencies via 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]. Bayesian networks have important applications in diagnostic expert systems particularly for medical applications, investigation of faulty systems, and legal domains. In recent years, Bayesian networks have also seen interesting applications in classification tasks, as basis for Bayesian network classifiers (See Chapter 20 of Russel & Norvig[3], CS5010 BN-L01). In this practical, you will gain familiarity with modelling and using Bayesian networks.

[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.

3.2 The task

In this part of the practical, we are tasked to design a Bayesian system for predicting the risk of digital security attacks for a company network CNX. The requirements are as follows:

> Design an alert system to help our CNX digital security officers to predict potential computer attacks. The alert system is intended to inform the officers about a potential risk with no distinction on the type of attack. The CNX network is a simple network that connects together a set of employees workstations, a database server, and additional web servers which in turn connect to the World Wide Web.

The Bayesian alert system relies on the following information in relation to a potential security attack:

1. A firewall is present to protect access to the database server. Normally, there is a 3% probability that the firewall is deactivated when maintenance is planned by the CNX network engineers. There is, however, a further 2% risk that the information held by the company regarding planned maintenance is out of date, causing vulnerabilities in the CNX network.
2. Access to unsafe websites is generally blocked successfully but there is a 15% chance that some websites are not identified as malicious.
3. World-wide Data Analytics about Distributed Denial of Service (DDoS) Attacks provide information about the intensity of attacks in different periods of the year. The analysis indicates that attacks are at peak during holidays. Assume that the CNX network is located in a place where there are 45 days of holiday in 360 days.
4. If we predict a risk of an illegal computer attack, there is only a 95% probability that the alert will be triggered due to some noise in the CNX network.
5. A logging system is also present in the CNX network, normally used for post-investigation in the event of an attack. The system differentiates logs by recording whether an activity was normal or anomalous. This logging system however has a bad fault tolerance and may give a wrong value with 30% probability.

In the core parts of the practical we assume that we intend to predict the risk of illegal attacks right now, without considering how the situation evolves in time.

3.3 Part 1

The first task is to design the Bayesian network for the system discussed above and develop a program that is able to construct and print this constructed Bayesian network. You must print variable names and their conditional probability tables (CPTs). To perform this task consider:

- Identify the events or variables required to model the alert system described above.
- Assume that each event has a binary domain.
- Decide the relationships between the domain variables.
- Then add the conditional probabilities for nodes that have parents, and the prior probabilities for nodes without parents.
- Use the information given to fill these probabilities, and for those probabilities that have not been given explicitly choose values that seem reasonable and explain why in your report.

- Construct the Bayesian network directly in your code, and ensure that this is accessible when invoking the program using the argument "CNX".

You may assume that the Bayesian network constructed is valid (i.e., it does not contain cycles and has correct CPTs). Three additional Bayesian networks, "BNA", "BNB" and "BNC", are given in a xml[4] and printed format for testing the next part of the practical. Please ensure that your code is also able to construct "BNA", "BNB", and "BNC".

3.4 Part 2 - Making simple inferences

In this part of the practical, you are tasked with developing an agent that can construct a Bayesian network and make inferences using the variable elimination algorithm. Details of the algorithm can be found in the slides, similar to the Russell & Norvig[5] algorithm, Elimination-ASK, page 528. The algorithm should be developed for events with binary domains (True or False). Queries for part 2 do not include evidence, and the order for elimination is provided as input. You may assume that the query and the order given are valid. For example, assume that we are querying the BNA network presented in Figure 1 and that we want to know the probability $p(D = True)$ or simply $p(D)$. For this task, your system should:

- Load the network constructed in part 1 (E.g., invoke the system with input argument '`BNA`')
- 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`')
- Read the order as user input: a list of variable names separated by a comma (E.g., '`A,B,C`')
- Calculate the output (E.g., $p(D) = 0.57050$)
- Print the output without further information. Outputs should be formatted as numbers with 5 decimal places. (E.g., print '`0.57050`' only)

In your report, discuss the output of at least one query on your CNX network. What does this indicate with respect to the alert system? In addition, discuss how different orders affect the results on your computation.

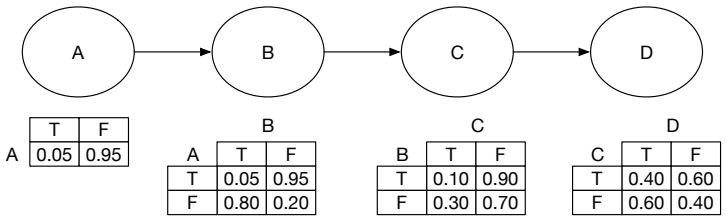

Figure 1: Example Network BNA

3.5 Part 3 - Adding Evidence

You are asked in this part to extend the variable elimination algorithm developed in part 2 with the possibility for a query to include evidence. You may assume that the query and the order given are valid. For example, assume that we are querying the BNA network presented in Figure 1 and that we want to know the probability $p(D = True|A = False, B = True)$ or simply $p(D|\neg A, B)$. For this task, your system should:

[4] readable with the AIspace Belief and Decision Networks tool (version 5.1.10, August 9th, 2016) available from http://aispace.org/

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

- Load the network constructed in part 1
- Read the query as user input as in part 2
- Read the evidence as user input (new part): a list of evidence variable name with their truth values (T/F) after a colon separated by a space (E.g., '`A:F B:T`')
- Read the order as user input as in part 2
- Calculate the output taking into consideration the evidence given (extended from part 2) (E.g. $p(D|\neg A, B) = 0.58000$)
- Print the output without further information. Outputs should be formatted as numbers with 5 decimal places.

In your report, discuss the output of at least two queries on your CNX network, one diagnostic and one predictive. What do they indicate with respect to the alert system?

3.6 Part 4 - Deciding on an order

You are asked in this part to extend the variable elimination algorithm developed in part 3 with an automatic approach to determine the order for elimination. You may choose any of the options discussed in the lectures. As previously, you may assume that the query is valid. For this task, your system should:

- Load the network constructed in part 1
- Read the query as user input as in part 2
- Read the evidence as user input as in part 3
- Identify an order using an algorithm (new part)
- Calculate the output taking into consideration the evidence given as in part 3
- Print the order of elimination
- Print the output without further information. Outputs should be formatted as numbers with 5 decimal places.

In your report, discuss how the order affects the computation of queries in your CNX network.

3.7 Part 5 - Advanced functionalities

It is strongly recommended that you ensure you have completed the requirements of part 1 to 4 before attempting any of these requirements.

You may consider one or two additional advanced functionalities. Please note that there is no benefit for implementing more than two. Examples of advanced functionalities include:

1. Define your own problem involving reasoning with evidence and uncertainty. Write down a text description of the problem, then model it using the code provided in Part 1. Make the problem sufficiently complex such that there are 5 nodes and some degree of correlation among variables. Use the query system provided in part 2 and 3 to investigate changes of probability distribution giving examples of one *predictive* and one *diagnostic* tasks. Present these examples and summarise your findings in your report.

2. Extend your application for an expert security agent assistant. The system should include a text-based interface that can be used to query the BN as an expert system. The user should be able to receive support for *diagnostic* or *predictive* tasks and the agent should show the results in a user friendly way.

3. Extend part 3 with additional algorithms to identify orders and compare the results with that of part 3.

4. Find out about approximate inference methods, such as a Gibbs sampler and compare the results with those obtained with the exact inference via your variable elimination algorithm.

5. Find out about dynamic Bayesian networks[6], and extend your CNX or a subset of CNX to model an alert system that varies over time. Demonstrate how your new model works using some queries.

6. Suggest and develop your own additional functionalities. A significant functionality should consider an extended design of a Bayesian network or an extended or alternative algorithm to make inferences.

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 11 – JDK11). 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 `A3main.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 is as that produced in the sample code.

4.3 Running

Your code should run using the following command:

```
java A3main <Pn> <NID> [<any other param>]
```

where `Pn` indicate the part of the practical `P1,P2,P3,P4,...` and `NID` indicates the network ID, for example `BNA,BNB,BNC,CNX`. For example to print BNA in part 1 we will use

```
java A3main P1 BNA
```

[6] see S. J. Russell and P. Norvig. Artificial Intelligence: A Modern Approach. Pearson Education, 3 ed., 2010. algorithm, Chapter 15, page 590

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

The output should include the variable name and its conditional probability table in any format.

To run the example in part 2 we will use:

```
java A3main P2 BNA
```

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 for part 2 and 3. They can be found at `/cs/studres/CS5011/Practicals/A3/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 working remotely, then change directory to your CS5011 A3 directory and execute the following command:

```
stacscheck /cs/studres/CS5011/Practicals/A3/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
- Examples and testing
- Insights and analysis demonstrated in the report

Some guideline descriptors for this assignment are given below:

- For a mark of 8 or higher: the submission implements part 1, adequately documented or reported.
- For a mark of 11 or higher: the submission implements part 1 and some attempt to part 2. The code submitted is of an acceptable standard, and the report describes clearly what was done, with good style.
- For a mark of 14 or higher: 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 design and evaluation of Bayesian reasoning.
- For a mark of 16 or higher: the submission implements fully parts 1 to 3. It contains clear and well-structured code and a clear report showing a good level of understanding of design and evaluation of Bayesian reasoning.
- For a mark up to 18: the submission implements fully parts 1 to 4. It contains clear, well-designed code, together with a clear, insightful and well-written report, showing in-depth understanding of design and evaluation of Bayesian reasoning.
- Above 18: the submission implements fully parts 1 to 4 and one or two advanced agent functionalities. The submission should demonstrate unusual clarity of implementation, together with an excellent and well-written report showing evidence of extensive understanding of design and evaluation of Bayesian reasoning.

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/`

Alice Toniolo, Mun See Chang, & Nguyen Dang

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

March 11, 2022