CS5011: Assignment 4 - Reasoning with Uncertainty - Bayesian Networks

(Words in the essay: 1900, Time spent: 20hrs) **Student ID:180029539**

Parts Implemented:

Part 1:

- I have created the Bayesian network for the given requirements. I have also included two new events as part of the second Bayesian network.
- For this part, I have taken the help of the bayes.jar and created the Bayesian networks and I have also created three queries one of each type (predictive, diagnostic, profiling).
- I have also extracted the XML files for the same with names bn1.xml, bn2.xml.

Part 2:

- I have replicated the two networks I created as part of part 1 using the encog java library and merged both of them to create a single network.
- For the merge, I used Feng et al process as instructed during the lectures.

Part 3:

- I have created a Bayesian network which estimates the chances of a particular disease from given information like continent and nature of place you were born in.

Functionalities:

Part 1:

- I have implemented all the requirements for part 1. I have created two Bayesian networks. Also, created the queries as per the requirement and they are working as intended.

Part 2:

- I have used the merge algorithm and tested if the nodes, dependencies and probabilities are correct by manually implementing the Feng at al process and they validated with my code output.

Literature Review:

Bayesian Network:

- Bayesian networks are being widely used in a wide range of verticals.
- These verticals include medicine, biomonitoring, system biology, etc..
- Let us look at some of the applications in detail.
- In Medicine, one of the places where Bayesian networks are used is to determine the chances of a disease given the symptoms. These are calculated based on

the general information available in the world regarding the disease and its symptoms. (Nikovski, 2000)

- It is also used in document classification. They along with classification have been used to assign a document to one or more classes which makes it easier to retrieve it at a later time.
- It is also used in spam filtering. It helps with the classification of emails which results in better filtering of the spam emails. (Niedermayer, 2008)
- Bayesian networks also play an important role in peer to peer communications. They are used to calculate the probability values of the trust levels of the network based on file quality, file type and download speed. (Wang, 2005)

Design, Testing and Evaluation:

Part 1:

Design:

- First, I created an 'email content' event as per the requirement and assumed the probabilities of the email being a personal email or a business email is 0.5 and 0.5 respectively.
- Then, I have understood that the detection system makes a prediction of high or low based on the email content. Hence, this becomes the child of the email content as it is an effect of email content.
- As per the requirement, it is stated that the probability of the detection system being wrong is 3% regardless of the email content. Hence, I assigned 0.03 for both (P(detection system =false | personal) and (P(detection system =false | business).
- Then, I have linked it to attack as it is the effect of a detection system.
- Next, as per the second requirement, I created an event 'firewall staus' and linked it to attack as its a cause of an attack.
- I have also created the event 'maintenance schedule' and assigned it values 'correct' and 'out of date'.
- For P(maintenance schedule = out of date), I set 0.02 as per the given requirement. This makes P(maintenance schedule = correct) = 0.98 according to $P(A^c) = 1 - P(A)$.
- Next, I have assigned P(Firewall Status = false | maintenance schedule = correct) and P(Firewall Status = false | maintenance schedule = out of date) as 0.05. As it is given that regardless of the fact maintenance is happening correctly or out of date, the probability of firewall being down is 5%.
- By using the $P(A^c) = 1 - P(A)$, I got the complement probabilities as 0.95.
- Then, I have created the event 'events(holidays/political) and linked it to attack as it's one of the causes of attacks. As per requirement assigned it the probability of 0.27778 as given 100 days are events in 360 days.
- Next, created an event 'alert' and made it a child of 'attack' event as its a cause of an alert.

- Then, set P(Alert =true | attack =true) and P(Alert = false | attack = false) as 0.8. Because, if an attack happens alert should be true and if an attack doesn't happen alert should be false and its given that accuracy of alert is 80%.
- Next, I have added the event ‘log’ to represent the logging system and assigned 0.3 for both (P(Log= anomalous| attack=true) and P(Log = anomalous | attack =false)) as the logging system is anomalous 30% of the time regardless of the attack.
- This concludes the construction of my first Bayesian network.
- For the second Bayesian network, I have added new requirements.
- Requirement 1: When an alert is raised, 50% of the management chose to receive SMS and other 50% chose for an email alert. In the case of normal situations (alert = false), only 20% chose to receive SMS whereas 80% opted for email.
- Requirement 2: As soon as an attack happens, 95% of the time a backup of the entire system is restored successfully, whereas in normal days (attack = false) backup is accidentally restored 1% of the total days.
- For these two requirements, I have added two events ‘Notification’ and ‘Backup Restore’ and added the ‘Notification’ event to ‘alert’ event and ‘Backup restore’ event to ‘attack’ as they are effects of those two events.
- For probability, assigned P(Notification = SMS | alert = true) as 0.5 and P(Notification = SMS | alert = false) as 0.2 as per requirement.
- Then, assigned P(Backup Restore = true | attack = true) as 0.95 and P(Backup Restore = true | attack = false) as 0.01 as per requirement.

Evaluation:

- I have assigned random values to the attack event probabilities to observe the difference in affected events.
- I have made three types of queries based on the requirement sheet.
- Predictive: What is the probability that an alert is triggered, given that it is a special event(political/holiday).
- The answer is 27% in both the networks as all the dependent event probabilities are the same in both the networks.
- Diagnostic: What is the probability that the firewall was active given an alert was triggered.
- It is 95% in both the networks as all the probabilities of dependent events are the same in both the networks.
- Profiling: What are the affected events when we change the status of an alert.
- All the dependent events probabilities change including the common parent nodes as the parent is not given. If the attack is also fixed to true then the common parent nodes are not affected.

  Critical Analysis:

  - From the first two queries, we understand that if all the internal and external nodes are same, then the probabilities will be unaffected.

- From the profiling query, we can understand the patterns of a casual chain, common parent and common child. We can see how fixing the values of some events affect the probabilities of other related events.
- We can understand that when a parent and one of its child is fixed, this blocks the changes in other children and vice versa.
- We can also observe when there is a common effect, fixing the common cause influences the other causes and vice versa.

Part 2:

Design:

- First, I have created the main class called ‘mainExecution’ for the execution starting point of my program.
- Then, I created our next class which is our agent class and named it ‘BayesianAgent’.
- Next, I instantiated an object in ‘mainExecution’ for ‘BayesianAgent’.
- Let us now see the construction of the Bayesian Agent.
- First, I created a ‘createNetwork1’ method for creating the events, dependencies and assign probabilities.
- Then, I have created a second method ‘createNetwork2’ and created the events, dependencies and assigned probabilities.
- Then, I have used the method extrctnodes1 and extractnodes2 to get the list of nodes from both the networks and store it in an array list.
- Next, I have created the method ‘extractAllNodes’ to get the union of nodes from both networks.
- Then, I have used generateRelation methods for both the networks to create a boolean array that contains the dependencies of child and parent in the networks.
- Then, I extracted the intersecting nodes between both networks and stored them in an array list.
- Next, I have stored the remaining nodes as non-intersecting nodes in a different array list using ‘extractNonIntersectNodes’ method.
- Then, I have used ‘splitIntersectionNodes’ method along with ‘getParentNodes’ method to split external and internal nodes into separate array lists.
- Next, I used the ‘deleteRule’ method to determine which relations to keep and which to discard. This gives a new boolean array with all the relations from both networks
- Then, I used the ‘treatNonIntersectionNodes’ method to determine what relations to keep from non-intersecting nodes.
- Next, I created a new network with new nodes and new boolean array using ‘createNewNetwork’ method.

- Finally, using the method ‘setProbability’, I set the individual probabilities using Feng et al method.

Examples, Testing and Evaluation:

1. After I finished my code, I ran the code to validate if all the nodes and dependencies were created correctly.
2. I printed the list of nodes from both the networks and dependencies from both networks and validated with my XML files. They validated perfectly.
3. Next, I have manually did the steps for the intersection, non-intersection, external and internal nodes and validated with the nodes and dependencies from the network generated by code.
4. **Evaluation :**
    - I have manually printed each step of the deleteRule and verified if each step is carried out correctly.
    - I have also calculated the probabilities of the nodes and all the probabilities match with the manually calculated values
    - I have also made sure the probabilities of true and false gets added to 1 in each CPT.

Part 3:

Design:

- I have implemented a Bayesian network about the chances of getting a disease based on the continent and nature of place born in.
- **Problem Statement:**
    - The population of world are divided between asia, africa and other parts of the world at 65%, 15%, 25% respectively.
    - The total urban population of the world is 68%, rest being rural.
    - The chances that you are tested positive for a disease given you are from urban Asia and rural Asia is 10% and 70%.
    - The chances that you are tested positive for a disease given you are from urban Africa and rural Africa is 20% and 90%.
    - The chances that you are tested positive for a disease given you are from other urban parts of the world is 30%.
    - The chances that you are tested positive for a disease given you are from other rural parts of the world is 50%.
    - The possibility of you getting flu or yellow fever given that you are tested positive for a disease is 70% and 80% respectively.
    - The possibility of you getting flu or yellow fever given that you are tested negative for a disease is 10% and 30% respectively.

Evaluation:

- **Queries:**
    - **Predictive:** What are the chances of you getting flu if you are from rural Asia?
    - It is 52%.
    - **Diagnostic:** What are the chances that if someone is tested positive for yellow fever and they have yellow fever that they are from a rural background?
    - It is 34%.

Running the code:

Part 2:

- Run the first part using " java -jar prob.jar ". It doesn't require any parameters.

Bibliography:

1. Cheng, J. and Greiner, R. (1999). Comparing Bayesian network classifiers. In Proceeding UAI'99 Proceedings of the Fifteenth Conference on Uncertainty in artificial intelligence.
2. Fenton, N. Independence and conditional independence. https://www.eecs.qmul.ac.uk/~norman/BBNs/Independence_ and_conditional_independence.htm. Accessed: 2017-03-06.
3. Friedman, N., Geiger, D., and Goldszmidt, M. (1997). Bayesian network classifiers. In Machine Learning Volume 29.
4. Heckerman, D. (2008). A Tutorial on Learning with Bayesian Networks, pages 33–82.
5. Russell, S. J., Norvig, P., and Davis, E. (2009). Artificial intelligence: a modern approach. Pearson, 3 edition.
6. Nikovski, D. (2000). Constructing Bayesian networks for medical diagnosis from incomplete and partially correct statistics. IEEE Transactions on Knowledge and Data Engineering, 12(4):509–516.
7. Niedermayer, D. (2008). An introduction to Bayesian network theory and their contemporary applications. In Innovations in Bayesian Networks Volume 156.
8. Wang, Y. and Vassileva, J. (2005). Bayesian network trust model in peer-to-peer networks. In Agents and Peer-to-Peer Computing.