**Aim**

Our aim is to prove the functionality, purpose, reliability and applicability of Bayesian Networks. For that, we are performing an inference test through the likelihood weighting method using our own algorithm on a Bayes Network previously given, and an exact inference test using the open source program JavaBayes on a different Bayes Network given.We are also proving a property of a specific type of Bayes Network through mathematical calculations. Finally, we are creating an example of a Bayesian Network to address a generic problem.

This problem is important to understand better the properties and implications of a Bayesian Network, a method to determine probabilities widely used in the artificial intelligence field.

**Methods**

Bayesian networks are graphical models for representing the interaction between variables visually. The Bayesian network is a directed acyclic graph where each node corresponds to a random variable, X, and has a value corresponding to the probability of the random variable given it’s parents. The nodes and the arcs define the structure of the network. This graphical representation is visual and helps understanding. The network represents conditional independence statements and allows us to break down the problem of representing the joint distribution of many variables into local structures; this eases both analysis and computation.

Rejection sampling is a general method for producing samples of a given distribution. It can be used to compute conditional probabilities. Rejection sampling produces a consistent estimate of the true probability by repeating the heuristics many times enough to the result to converge to the actual result. The biggest problem with rejection sampling is that it rejects so many samples, that for complex problems this method is impossible to be used.

Likelihood weighting may be regarded as the optimization of the rejection sampling since it generates only events that are consistent with the evidence to compute the conditional probability.

Variable elimination is an exact inference method used in probabilistic models, such as Bayes Networks. It carries out summations of probabilities of the variables that are irrelevant for a certain inference query desired, then stores the immediate results to avoiding recalculations. By the end of the summations, the exact inference probability is extracted. Although the result is very accurate and it does not make recalculations, the processing time is too big and dependent of the order that the variables are eliminated, making it difficult to apply in huge networks.

**Results and Discussion**

This problem is important to understand better the properties and implications of a Bayesian Network, a method to determine probabilities widely used in the artificial intelligence field.

* Task 1
* Task 2

1. Given the following Bayes network based on a medical situation with the variables:

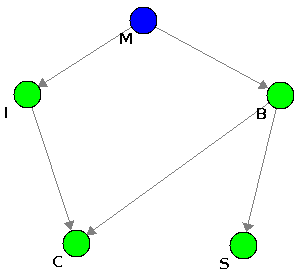
M: Patient has metastatic cancer.

I: Patient has increased total serum calcium.

B: Patient has brain tumor

C: Patient is in coma

S: Patient has severe headache



2) P(c) = 0.32 ????????????????????????to achando q eh aquele valor da primeira imagem q nao esta em highlight

3) P(m|s,¬c) = 0.2

4) Total Serum Calcium and Brain Tumor (B and I)

5) Consider

Which implies

?????????????????pq vc fez I e B dependendo d m?

We want to see if I and B are independent given C, for that we expect the equation as follows: . If they are conditionally dependent, the equation should be presented as

From:

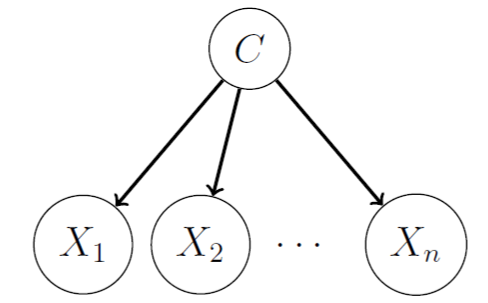
Using variable elimination:

Therefore, I and B are conditionally dependent given C.

6) P(c|m) = 0.68

* Task 3

Mathematically prove two properties of the given Bayesian Network.



* + Task 3 -1

We want to show that the factorisation follows the independence assumptions from the graph.

Since there is no in the function, the independence is proven.

* + Task 3 – 2

We want to show that if all variables are binary valued, then , where Xi=0 if X=¬x and 1 otherwise.

* Task 4

We created a Bayesian Network to solve the problem to decide to leave or stay at home depending of some conditions of the day.

The problem graph is:

SD

WD

T

WO

HS

Its full disjoint distribution is defined by:

The variables are:

T: the temperature of the day. It has three possible values with their probabilities: hot (1/3), normal (1/3) and cold (1/3).

SD: indicates if the current date is special date or not. It has three possible values with their probabilities: holiday (12/365 considering the NSW calendar), special birthday (10/365 considering that the average number of special birthdays that a normal person attends in a year is 10) and normal (343/365).

WD: the days of the week. It has seven possible values with their probabilities: sunday (1/7), monday (1/7), tuesday (1/7), wednesday (1/7), thursday (1/7), friday (1/7), and saturday (1/7).

WO: indicates if there is work to do in the current day or not. It is a binary variable with its probability tables depending on SD (since the work can be suspended in holidays, or the person decides to not work given a holiday or a special birthday) and WD (since it is less likely to be required to work on Sundays and Saturdays):

Work = 1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Sun | Mon | Tue | Wed | Thu | Fri | Sat |
| Holiday | 0.05 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.05 |
| Birthday | 0.05 | 0.2 | 0.8 | 0.8 | 0.8 | 0.2 | 0.05 |
| Normal | 0.05 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.5 |

Work = 0

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Sun | Mon | Tue | Wed | Thu | Fri | Sat |
| Holiday | 0.95 | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 | 0.95 |
| Birthday | 0.95 | 0.8 | 0.2 | 0.2 | 0.2 | 0.8 | 0.95 |
| Normal | 0.95 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.5 |

LH: indicates if the person should leave or stay home. It is a binary variable with its probability tables depending on T (since it is more likely to leave home in a hot day than in a cold one), SD (since it is more likely to leave home in holidays or special birthdays) and WO (since it is more likely to leave home in a work day):

Leave Home = 1

|  |  |  |  |
| --- | --- | --- | --- |
| Work = 1 | H | N | C |
| H | 0.98 | 0.95 | 0.25 |
| B | 0.98 | 0.98 | 0.75 |
| N | 0.95 | 0.95 | 0.75 |
| Work = 0 | H | N | C |
| H | 0.95 | 0.75 | 0.05 |
| B | 0.8 | 0.75 | 0.25 |
| N | 0.6 | 0.5 | 0.1 |

Leave Home = 0

|  |  |  |  |
| --- | --- | --- | --- |
| Work = 1 | H | N | C |
| H | 0.02 | 0.05 | 0.75 |
| B | 0.02 | 0.02 | 0.25 |
| N | 0.05 | 0.05 | 0.25 |
| Work = 0 | H | N | C |
| H | 0.05 | 0.25 | 0.95 |
| B | 0.1 | 0.25 | 0.75 |
| N | 0.4 | 0.5 | 0.9 |

**Conclusions**

In this work, we discovered how to sample from a Bayesian net, how to implement weighted randomness, how to specify Bayesian nets, how to design them in an open source program and how this may be helpful for visualization and querying of our data. We also observed that specific Bayesian networks may have very interesting properties, such as strong independency of the variables, and that we may find very interesting functions out of it.

We also noticed how hard it might be to design a Bayesian network and how expert knowledge is needed, since every single node probability must be defined, and also the interaction between the nodes. It was easy to notice that Bayesian networks may get very complex making it hard to define it’s probability, because some data is very hard to be acquired, or even impossible.

We suggest as future work:

1. To implement likelihood weighting and variable elimination on the designed network of the task 4;
2. Compare the results from this implementations in task 4 to the results acquired using an open source program;
3. To implement a generic query for the Bayesian network of Task 1 such that we can observe any node and run the likelihood weighting on any other;
5. Profit

**Reflection**

This experiment was helpful to have a better and deeper understanding of Bayesian networks since in computation the most efficient way to learn something is to implement that yourself.

In the programming part we learned how to implement Bayesian networks and to sample from them with good enough random algorithms. Then we had to learn how to use JavaBayes, which was sufficiently easy yet very interestingly because it is a very powerful tool to visualize Bayes Nets. In the mathematical proofs and the network creation, we learned more properties of the Bayesian networks and how they can be used in a great variety of problem situations.

**Bibliography**

Introduction to Machine Learning, Ethem Alpaydin

Artificial Intelligence - A Modern Approach  
Second Edition  
Stuart J. Russel and Peter Norvig