Artificial Intelligence

Assignment 3

Learning Bayesian Networks

Cameron Darragh - 42934936

Addison Gourluck - 42906380

**Part I [50 points]: Learning Conditional Probability Tables**

**Task 1 [25 points]**

Run the main method of ‘solution’, with the first parameter as the filename stored under ./data/ and the second parameter as “Task1”. The output file is located under: ./solutions/.

**Task 2 [10 points]**

Run the main method of ‘solution’, with the first parameter as the filename stored under ./data / and the second parameter as “Task2”. This outputs the likelihood and log-likelihood to standard error.

**Task 3 [15 points]**

*Please run the program you've written for Task-1 and Task-2 on each training data set.*

Files are stored under: ./solutions/.

*Please write the likelihood and log-likelihood of the CPT for each training data set.*

|  |  |  |
| --- | --- | --- |
| **Filename** | **Likelihood** | **Log Likelihood** |
| cpt-CPTNoMissingData-d1 | 2.871 E -11 | -24.274 |
| cpt-CPTNoMissingData-d2 | 8.626 E -58 | -131.395 |
| cpt-CPTNoMissingData-d3 | 0.0 | -Infinity |

Note: Because of our implementation, we could not express d3. Therefore, it is always –Infinity, or 0. Please forgive our incompetence.

*Please explain how the likelihood and log-likelihood measure of the Bayesian Network differs as the number of training data set increases.*

From the trend seen in the training data, and from tests with other sets of data, it can be said that the size of the data set and the log-likelihood are directly proportional to each other, irrelevant of the number of nodes in the network. For example; doubling the data set will double the magnitude of the log-likelihood. Naturally, to follow this trend, the likelihood approaches zero as the data set grows in size, expanding exponentially.

*Please explain how the likelihood and log-likelihood measure of the Bayesian Network differs as the number of variables (nodes) increases.*

The likelihood and log-likelihood seem to be completely independent of the number of nodes, barring the case when there are so few nodes that the number of possible different combinations (2^nodes) is significantly less than the amount of data, or if very few combinations of data are used (e.g: all data either 0000 or 1111).

*Please write a short discussion on how the likelihood and log-likelihood measure will differ when the possible values of each variable increases.*

As the number of variables increases, so too will the number of possible combinations of data, as mentioned previously. The formula for combinations would become (V^nodes), where ‘V’ is the number of possible values.

It is not possible for us to test with our current implementation, but logically, it wouldn’t make sense for the likelihoods to change. The dataset provided would have more possible datasets to choose from, but assuming we continue to use the same data set, none would be used, and the likelihoods would remain the same. If the data was re-generated, and contained the new variables, then the likelihood would have to change. This is because not only will there be a higher chance for more unique data sets, but there will be more combinations that were not chosen, and thus can be ruled out as having no relation. I predict that the likelihood would decrease drastically, approaching zero even faster, and likewise, the log-likelihood would approach negative infinity faster.

**Part II [50 points]: Learning Structure and Conditional Probability Tables**

**Task 4 [25 points]**

Run the main method of ‘solution’, with the first parameter as the filename stored under ./data/ and the second parameter as “Task4”. The output file is located under: ./solutions/.

**Task 5 [7 points]**

*Please experiment with the scoring function by changing the constant parameter. For each parameter, please run the program you’ve written for Task-4 on each data set. For each data set and each parameter, please write the score function of the final Bayesian Network. Please explain how the final Bayesian Network changes as the parameter increases/decreases.*

The function used is: Score = Log-Likelihood – C \* Data-Size

By using a negative constant in the network, its score will increase proportionately to its complexity. This is because the constant is serving the opposite of its original purpose, and adding the linearly magnified result of the constant and complexity rather than subtracting it. There probably won’t be many situations when a negative constant is used.

A constant of 0 (obviously) results in a score equal to the log likelihood.

|  |  |  |  |
| --- | --- | --- | --- |
| **C** | **noMissingData-d1** | **noMissingData-d2** | **noMissingData-d3** |
| -1 | -12.618 | -82.536 | -Infinity |
| 0 | -24.618 | -182.536 | -Infinity |
| 0.01 | -24.738 | -183.536 | -Infinity |
| 0.1 | -25.818 | -192.536 | -Infinity |
| 1 | -36.618 | -282.536 | -Infinity |
| 10 | -144.618 | -1182.536 | -Infinity |
| 100 | -1224.618 | -10188.064 | -Infinity |

For dataset 1:

C <= 206.29 results in A being the parent of B and C.

C >= 206.30 results in no parents for any nodes.

For dataset 2:

C <= 23.45 results in B being a parent of L, M and G; L being a parent of M and G; and G being a parent of M.

C >= 23.46 results in no parents for any nodes.

The trend seen in the first two columns of results is obvious – a variable equivalent to the size of their data is simply being subtracted from the log likelihood, after being multiplied by the constant.

**Task 6 [8 points]**

*Please implement “no edge” and “random chain” to initialize the structure. Please run the Bayesian Network generation program (Task-4) with these two initialization methods on each data set and compare the final Bayesian Network (in terms of the scoring function and structural complexity) after 3 minutes searching time.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **noMissingData-d1** | **noMissingData-d2** | **noMissingData-d3** |
| No Edge | -36.618 | -282.536 | -Infinity |
| Random Chain | -36.614 | -282.668 | -Infinity |

For dataset 1, Random Chain results in:

|  |  |
| --- | --- |
| **Node** | **Parents** |
| A | - |
| B | A |
| C | A, B |

For dataset 1, No Edge results in:

|  |  |
| --- | --- |
| **Node** | **Parents** |
| A | - |
| B | A |
| C | A |

For some reason, here No Edge didn’t connect for the ideal structure, which was found by Random Chain. It could be the result of a bug, or just a quirk of the method.

For dataset 2, Random Chain results in:

|  |  |
| --- | --- |
| **Node** | **Parents** |
| B | - |
| L | B, G |
| M | L, B |
| G | - |

For dataset 2, No Edge results in:

|  |  |
| --- | --- |
| **Node** | **Parents** |
| B | - |
| L | B |
| M | L, B, G |
| G | B, L |

The second dataset gave a slightly worse score, and slightly different parent relationships.

**Task 7 [10 points]**

*Please implement the “best tree network” to initialize the structure, and compare the final Bayesian Networks results with the Bayesian Networks generated with “no edge” and “random chain” initialization method (Task-6), in a similar manner as in Task-6.*

For dataset 1:

|  |  |
| --- | --- |
| **Node** | **Parents** |
| A | - |
| B | A |
| C | A, B |

Score: -36.61350042656724

The result here is the same as what was found by the random chain method previously. No surprises there.

For dataset 2:

|  |  |
| --- | --- |
| **Node** | **Parents** |
| B | L, G |
| L | - |
| M | L, B, G |
| G | L |

Score: -267.4716932888239

This is significantly better than the previous solutions, which yielded far poorer scores.

For dataset 3:

|  |  |
| --- | --- |
| **Node** | **Parents** |
| A | - |
| B | - |
| C | - |
| D | - |
| E | - |
| F | - |
| G | - |
| H | - |

Score: -Infinity