Learning a Bayesian Network structure from data

Luca Domeniconi
Master's Degree in Artificial Intelligence, University of Bologna
luca.domeniconi5@studio.unibo.it

March 26, 2024

Abstract

The goal of this mini-project is to explore the TPDA algorithm (Cheng, Bell, and Liu 1997) used to learn a Bayesian network structure directly from the data.

I implemented the TPDA algorithm using Python and the NetworkX library to manage the graph, and then tested the implementation using datasets for classification and already known Bayesian networks to compare the obtained structure.

Introduction

Domain

This project is deeply inspired by the paper (Cheng et al. 2001), in which the authors explain how and why the TPDA algorithm works and is able to learn the structure of a Bayesian network starting from the data.

A central concept in this work is the *Mutual Information*, that is used to measure the volume of information flow between two variables A and B, and it is evaluated with the following formula:

$$I(A,B) = \sum_{a,b} P(a,b) \log \left(\frac{P(a,b)}{P(a)P(b)} \right)$$

To calculate the mutual information between two variables with respect to some evidence variables (*condition-set*) C, we use the following formula:

$$I(A,B \mid C) = \sum_{a,b,c} P(a,b,c) \log \left(\frac{P(a,b \mid c)}{P(a \mid c)P(b \mid c)} \right)$$

The mutual information is then used as a measure of how much two variables are correlated.

It should be noted that one could have used some other measures, such as the correlation coefficient given by $C(A,B) = Cov(A,B)/\sqrt{Var(A)Var(B)}$. The main advantage of mutual information is that it gives a more general tool to analyze both linear and non-linear correlations.

Aim

The purpose of this project is to implement the TPDA algorithm as described in (Cheng et al. 2001) and experiment with it using some dataset and test the accuracy of the resulting structure on classification tasks. I also used some already known Bayesian Network to sample a dataset associated with it and use the TPDA algorithm to reconstruct its topology.

Method

To implement the TPDA algorithm, I used the NetworkX library to manage the graph and the Pandas library to manage the data. I then used the pgmpy library to load a well-known Bayesian Network and sample data from it. The TPDA algorithm is composed of three main steps:

Drafting the algorithm computes mutual information of each pair of nodes as a measure of *closeness* and creates a draft based on this information. It considers only the pair of nodes that have mutual information greater than a parameter ϵ, connecting first the pair of nodes with the highest mutual information. The connection is made only if there are no paths between the two nodes, in this way the draft is a singly connected graph (a graph without loops).

Thickening the algorithm looks at all the pairs of nodes that are not connected and adds an edge between them if they cannot be *d-separated*.

Thinning each edge of the graph is examined using conditional indipendence tests and will be removed if the two nodes of the edge can be *d-separated*.

Results

Something I have learned is that TPDA heavily depends on the parameter ϵ used as a threshold throughout the entire process of the creation of the Bayesian network structure. With high values (e.g. $\epsilon > 0.2$) we might lose some correlation between variables, while with low values (e.g. $\epsilon < 10^{-4}$) we might connect two variables that are not really correlated. So for an optimal value of ϵ we have to do some experiments and see the results.

Analysis

Experimental setup

To evaluate the result, three criteria have been used:

1. **Log likelihood**[1]: measures how well the specified model describes the data. For this purpose two different known Bayesian networks have been used ("Asia" and "Cancer").
2. **Graph Edit Distance**[2]: measures how many edges have to be modified in order to match the reconstructed model with the original one. For this purpose, the same Bayesian Network for Log likelihood had been used. In this criteria the undirected version of the graphs have been considered, because the edge orientation in a Bayesian Network is usually not relevant.
3. **Accuracy**: scores how well the predicted model behaves in a classification task using standard datasets from `sklearn`. For this purpose three classification datasets ("Apple Quality", "Iris", "Wine Quality") and the two well-known Bayesian networks ("Asia" and "Cancer") have been used. In order to use a Bayesian network for a classification task, I did inference on the target, using all the other variables as evidence. Then, looking at the resulting CPT of the target variable, the classification is made using the value with the highest probability.

Each test has been repeated with different values of ϵ to study how does the reconstructed model react to that parameter. It is expected to have bad results for too small values or too big values for ϵ.

Results

The results show how the chosen value for ϵ relates to the various metrics used on the different datasets.

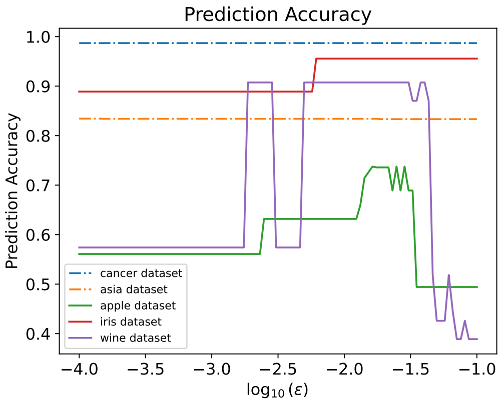

Figure 1: Prediction accuracy over ϵ values

According to expectation, in the accuracy results shown in Figure 1 the best outcomes are for values of ϵ that are neither too high nor too low ($\epsilon \approx 10^{-2}$). Examining the graph edit distance in Figure 3 the best values for ϵ lies at around 10^{-4} and the same is for the Log likelihood in Figure 2.

These results show that there is no perfect value for ϵ and the result may vary a lot depending on the dataset and the metrics used to evaluate the resulting network.

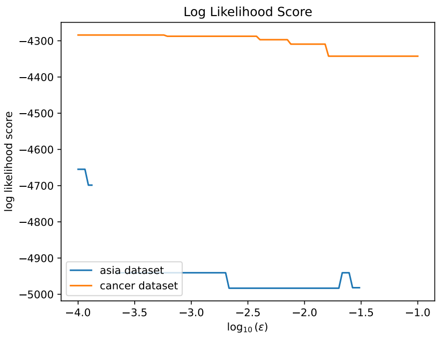

Figure 2: Log likelihood over ϵ values (Higher the better)

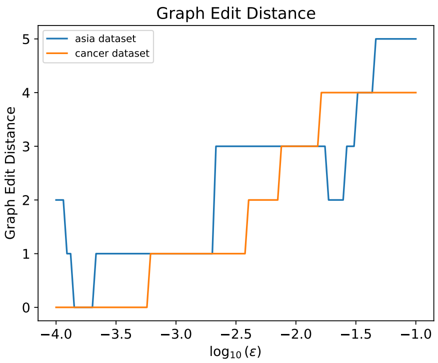

Figure 3: Graph edit distance over ϵ values (Lower the better)

Conclusion

During these experiments, I have learned the true capabilities of Bayesian networks and limitations, especially in terms of computational cost in building big networks. In fact, I had to discard the experiments on the breast cancer dataset from the `sklearn` package because it has 30 attributes and the execution times were unfeasible.

Learning Bayesian networks in an automated way directly from the data can be a very useful tool. Despite that, considering the fact that they are highly explainable, it is always important to analyze the resulting structure using expert knowledge to ensure its correctness. More importantly, it is critical to ensure that it does not contain any type of bias implicitly present in the data.

Links to external resources

- Asia Bayesian Network
- Cancer Bayesian Network
- Apple Quality Dataset
- Wine Quality Dataset
- Iris Dataset

[1] `https://pgmpy.org/metrics/metrics.html`

[2] `Graph edit distance documentation`

References

Cheng, J.; Bell, D. A.; and Liu, W. 1997. Learning belief networks from data: an information theory based approach. In *Proceedings of the Sixth International Conference on Information and Knowledge Management*, CIKM '97, 325–331. New York, NY, USA: Association for Computing Machinery.

Cheng, J.; Grainer, G.; Kelly, J.; Bell, D.; and Lius, W. 2001. Learning bayesian networks from data: An information-theory based approach, 2001. *URL citeseer. ist. psu. edu/628344. html*.