Automatic Lecture, Tutorial, and Laboratory University Venue Allocation for Varying Class Sizes

Litrature Review

Prepared by Uvir Bhagirathi

Supervised by: Ritesh Ajoodha

1141886

March 23, 2018

Introduction

Universities attract large amounts of students for which each faculty has to facilitate. Within each faculty, students are registered for various courses under different schools and require appropriate learning environments for each course. While these universities do offer these environments there is often difficulty in allocating students to these venues efficently, based on the needs of different schools. The goal of this research project is to construct a system that allocates venues to classes of various sizes by taking into consideration the distance between the respective faculty buildings to the venues being used, the capacity of the venue as well as the number of students in the course.

The task of venue allocation is a complex problem, however, this can be solved by using graphical modeling, more specifically, Bayesian networks. Each variable of the problem will be incorporated into a Bayesian network. In order to learn the parameters of each variable a parameter learning tool will need to be implemented. Thereafter, a scoring function will be used to evaluate and find the best scoring network structure which will output the optimum venue-class allocations.

Bayesian Network

A Bayesian network is a probabilistic graphical model in the form of a Directed Acyclic Graph (DAG) where each node is a variable and each edge is the conditional dependence between two variables. Bayesian networks map the relationship between events in terms of probability. It shows how the occurrence of certain events influence the probability of other events occurring. Below is an example of a Bayesian Network [Pearl 2014]:

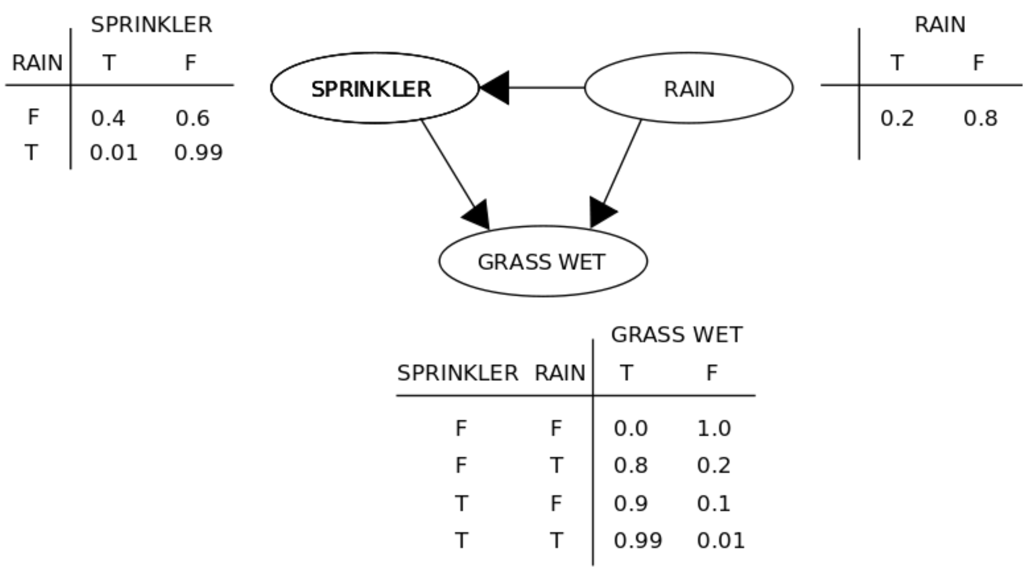

Figure 1: Sprinkler Problem

It can be seen in Figure 1, each node has a probability of its occurrence, and its probability of occurrence depending on the occurrence of other nodes. This kind of diagram is very useful when we have inter-dependent nodes, and want to model outcomes and make decisions.

Parameter Learning

When working with observable data it can be said that the simplest learning tool to use is Maximum Likelihood Estimation (MLE). MLE is a commonly used learning tool for observable data. Another popular parameter learning tool that can be used is Bayesian Estimation (BE) which is fairly similar to MLE [R. Ajoodha *et al.* 2018].

Maximum Likelihood Estimation

MLE attempts to find the parameter values that maximize the likelihood function, given the observation data. A likelihood function is simply a function of the parameters of a statistical model given data [R. Ajoodha *et al.* 2018]. MLE is used with large data sets. For example: Suppose that we want to measure the height of all the male students at wits, but we're unable to measure the height of every person due to the time and cost constraints. Assuming that the heights are normally distributed, the mean and variance can be estimated with MLE with only the heights of some sample of the overall population. MLE would accomplish that by taking the mean and variance as parameters and finding particular parametric values that make the observed results the most probable given the normal model. However, the optimal parameter value is not given by MLE. Guesses can be made and MLE will use a likelihood to compare which guess was better.

Bayesian Estimation

A Bayesian Estimator views any event that has uncertainty as a random variable with a distribution over it [R. Ajoodha *et al.* 2018]. BE uses Bayes' Theorem (Figure 2) and other functions to calculate the parameter values.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Figure 2: Bayes' Theorem

Where:

- $P(A)$ is the prior probability of A. It does not take B into account.
- $P(A|B)$ is the conditional probability of A, given B.
- $P(B|A)$ is the conditional probability of B given A.
- $P(B)$ is the prior probability of B.

Bayesian analysis is a convenient setting for many models, such as hierarchical models. However, it can get computationally intensive when models involve many variables.

Scoring Functions

A scoring function is used to evaluate the usability of a network model. Depending on the rules of the scoring function each network is allocated a 'score' which shows how well the model fits the observed training data [R. Ajoodha *et al.* 2018]. There are many scoring functions, however, the most prefered functions are the likelihood function and the Bayesian Information Criterion (BIC). In most cases the likelihood function tends to prefer the most complex networks over the simpler ones. This leads to overfitting problems but can be overcome by penalising the complexity of the structure. The BIC score adapts to a more complex model to fit the data as the size of the sample size grows, the more closer the model approaches its optimum form [Koller *et al.* 2009]. Greedy Hill-climbing is heuristsc search used for optimization problems [R. Ajoodha *et al.* 2018]. This algorithm can be used to find the best structured network but the solution that is found may not be the global optimal maximum.

Bayesian Information Criterion

BIC is a gauge for model selection among a finite set of models[Chickering et al. 1996].

$$BIC = ln(n)k - 2ln(L)$$

Figure 3: Bayesian Information Criterion

Where:

- L the maximized value of the likelihood function of the model.
- n the number of observations.
- k the number of parameters estimated by the model.

Greedy Hill Climbing

The Greedy Hill-Climbing algorithm is used in optimization problems where we need to minimize or maximize a function [Tsamardinos *et al.* 2006]. It is a heuristic search algorithm, which means that it might not find the optimum solution, however, it will provide a local minimum or maximum in reasonable time. The 'greedy' part of the algorithm implies that it moves in which ever direction optimizes the function. The problem with this algorithm is that it may not allows output the optimum network structure but this can be solved using backtracking.

Conclusion

A Bayesian network will be implemented with each node being a variable of our problem. i.e a venue, a course, etc. Bayesian Estimation will be used to learn the parameters between each node, as opposed to MLE. This is because MLE does not produce the optimal parameters. Even though BE can be computationally intensive when given models with many varibles, the output will be the optimal structured network. Thereafter, the Greedy-Hill-Climbing algorithm can be implemented to find the network with the best score. Random starts as well as backtracking methods can be used to overcome the issue of the algorithm obtaining a local maximum [Tsamardinos *et al.* 2006]. This will give us the optimum Bayesian network.

References

[Pearl 2014] Pearl, J., 2014. Probabilistic reasoning in intelligent systems: networks of plausible inference. Elsevier.

[R. Ajoodha *et al.* 2018] R. Ajoodha, B. Rosman. Learning the Influence Structure between Partially Observed Stochastic Processes using IoT Sensor Data. SmartIoT: AI Enhanced IoT Data Processing for Intelligent Applications at AAAI-18, 2018.

[R. Ajoodha *et al.* 2018] R. Ajoodha, B. Rosman. Tracking Influence between Naive Bayes Models using Score-Based Structure Learning. PRASA-RobMech International Conference, 2017.

[Chickering *et al.* 1996] Chickering, D.M. and Heckerman, D., 1996, August. Efficient approximations for the marginal likelihood of incomplete data given a Bayesian network. In Proceedings of the Twelfth inter-national conference on Uncertainty in artificial intelligence (pp. 158-168). Morgan Kaufmann Publishers Inc

[Koller *et al.* 2009] D. Koller and N. Friedman, Probabilistic graphical models: principles and techniques. MIT press, 2009. Chapter 18: section 18.3 - 18.4

[Tsamardinos *et al.* 2006] Tsamardinos, I., Brown, L.E. and Aliferis, C.F., 2006. The max-min hill-climbing Bayesian network structure learning algorithm. Machine learning, 65(1), pp.31-78.