Artificial Intelligence Laboratory 3: Bayesian Network
DT8042 HT23, Halmstad University

Fabrice Bodson (fabbod23student.hh.se) ; Mery Ensay (merens23student.hh.se)

November 2023

Introduction

Q1: What types of algorithm have you used or implemented in this lab?

In this 3rd lab of the AI course we had the opportunity to implement basic Bayesian algorithms and networks. The first task focused on the Bayesian networks, which is a directed graph in which each node has a quantitative probability information. The second task focused on the Bayes theorem with some probability exercises and the implementaiton of the Naïve Bayes classifier.

Task 1: Bayesian Network

For task 1...

Q2: Learn a Bayesian Network from the given smart grid data. Illustrate the learned network.

Q3: Compute the following

1. p(Outage duration=$Otg \leq 1$ | Time = Morning, Demand Factor=Medium)
2. p(Demand Factor=High | Overload=Yes, Time=Afternoon)
3. p(Number of Customers=Low | Demand Factor=High)

	Outage_Duration	p
0	Less_than_1H	0.482832

	Demand_Factor	p
0	High	0.367494

	Number_of_Customers	p
1	Low	0.538824

Q4: What have you explored further? Write down your experiment and observation.

For this first task, we used a dataset and set up our first neural network.

The first step was to set up the dataset and then create the model with the following command line: model = bn.structure$_learning.fit(df)$ Fit is used to learn the structure of a Bayesian network from a dataset. Fit uses an algorithm to find the Bayesian network structure most compatible with the training data.

Model is therefore an object of type bn, which is the neural network. Now that we have the neural network, we can use it to make inferences.

We then created three different inferences, which can be found in question 3.

Task 1b : Finally, we have implemented our own inferences

We have tested inference with several variables

1. p(Number_of_Customers | Day = Weekdays) : Represents the probability of having a high or low number of customers during the week.

	Number_of_Customers	p
0	High	0.461176
1	Low	0.538824

2. p(Season,Weather | Time = Morning) : Represents the probability of cold or hot weather depending on the season.

	Season	Weather	p
0	Autumn	Cold	0.145356
1	Autumn	Warm	0.10818
2	Spring	Cold	0.134175
3	Spring	Warm	0.112296
4	Summer	Cold	0.0971001
5	Summer	Warm	0.153451
6	Winter	Cold	0.168307
7	Winter	Warm	0.0811352

3. p(Weather,Number_of_Customers | Time = Morning) : Represents the probability of having a high or low number of customers mixed with the season for mornings

	Weather	Number_of_Customers	p
0	Cold	High	0.250658
1	Cold	Low	0.29428
2	Warm	High	0.210519
3	Warm	Low	0.244544

Task 2: Naïve Bayes

Q5: Compute and fill in the likelihood table (task 2a).

likelihood	-	*Has_pet*	-	-
-	-	Yes	No	P(Gender)
Gender	Male	0.67	0.6	0.625
-	Female	0.33	0.4	0.375
-	P(*Has_pet*)	0.375	0.625	-

Table 1: Likelihood table Gender

likelihood	-	*Has_pet*	-	-
-	-	Yes	No	P(Gender)
Gender	Male	0.67	0.4	0.5
-	Female	0.33	0.6	0.5
-	P(*Has_pet*)	0.375	0.625	-

Table 2: Likelihood table Education

Q6: Compute the posterior probabilities (task 2b).

Posterior probability for P(No—Male): 0.6 Posterior probability for P(Yes—Female): 0.33 Posterior probability for P(No—HighSchool): 0.75 Posterior probability for P(Yes—University): 0.5

Q7: Compute the likelihood of having pets for numerical features (task 2c).

Have pets = Mean: 0.375 ; Standard deviation: 0.518
Normal Distribution for having a pet: 0.372
Normal Distribution for not having a pet: 0.593

L(Income = 90000 — Yes): 0.3333333333333333
L(Income = 90000 — No): 0.2

Q8: Make inference with Naïve Bayes (task 2d).

-	-	X1	X2
-	-	University	HighSchool
-	-	Female	Male
-	-	-	-
Has_pet	Yes	1.065450647621758e-06	6.948597222460778e-07
-	No	1.2850301795528517e-06	1.8856437510069952e-06

Table 3: Inference with Naive Bayes

Q9 (extra credit): Implement Naïve Bayes (task 2e) classifier and perform classification on the Iris dataset.

Please include the pseudo-code of your implementation in the report.
/ Not done /

Conclusion and Summary

This lab was a little bit less fun than the previous one as it consisted in implementing mathematical theories into code without having a game or context to see a fun result. Otherwise it helped us to have a better understanding of Bayesian networks.