Experiment 5
Graphical models for inference under uncertainity and Naive Bayes classification

Jagruti Piprade
Computer Science Engineering
202151067@iiitvadodara.ac.in

Makwana Harsh Maheshkumar
Computer Science Engineering
202151082@iiitvadodara.ac.in

Khushi Saxena
Computer Science Engineering
202151078@iiitvadodara.ac.in

Nitin Gautam
Computer Science Engineering
202151101@iiitvadodara.ac.in

Google collab notebook for this assignment can be found here.

***Abstract*—This report explores how we can use graphs to understand situations where things aren't certain. It discusses creating models called Bayesian Networks using R, a programming language. This report explores how we can use these ideas practically, using the R programming language and a package called bnlearn. The goal is to learn the structure and CPTs of a Bayesian Network from a dataset of student grades, and subsequently use this network for inference and classification tasks.**

I. Learning Objective

Understand the graphical models for inference under uncertainty, build Bayesian Network in R, Learn the structure and CPTs from Data, naive Bayes classification with dependency between features.

II. Introduction

Graphical models are great for handling uncertainty because they offer a structured way to represent and work with complicated probability information. Among them, Bayesian Networks are particularly good at showing how things relate to each other. It's like a flowchart where each point (or node) is a random variable, and each arrow (or edge) shows how one variable affects another. Each node has a probability function that tells us the likelihood of different outcomes. Bayesian networks help us see how different things are connected.

III. Probelm I: Dependencies Between Courses

Problem Statement: Consider grades earned in each of the courses as random variables and learn the dependencies between courses.

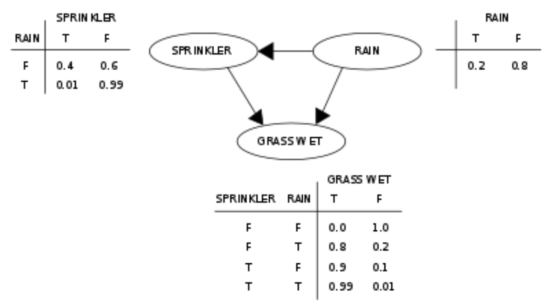

Fig. 1. Example of a Bayesian network

A. Bayesian Network

Bayesian networks are a way of modeling relationships between different events or variables through a directed acyclic graph (DAG).

Key Components:

- Nodes: Random variable
- Edges: Conditional dependencies among variables
- Conditional Probability Tables (CPTs): Associated with each node is a conditional probability table, which specifies the probability distribution of a node given its parents in the graph. It quantifies how the variables are related.

$$\text{Cause} \rightarrow \text{Effect}$$

They are particularly useful for modeling and reasoning under uncertainty. These networks can be employed for tasks such as classification, prediction, diagnosis, and decision-making by utilizing probabilistic inferences.

Naive Bayes: Naive Bayes is a machine learning algorithm used for classification tasks. The **Naive** part comes from a simplifying assumption the algorithm makes: it

	EC100	EC160	IT101	IT161	MA101	PH100	PH160	HS101	QP
	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>
1	BC	CC	BB	BC	CC	BC	AA	BB	y
2	CC	BC	BB	BB	CC	BC	AB	BB	y
3	AB	BB	AB	AB	BB	CC	BC	AB	y
4	BC	CC	BB	BB	BB	BB	BC	BB	y
5	BC	AB	CD	BC	BC	BC	BC	CD	y
6	DD	CC	DD	CD	CD	CC	BC	BC	n

Fig. 2. Dependency between courses

assumes that all the features are independent of each other.

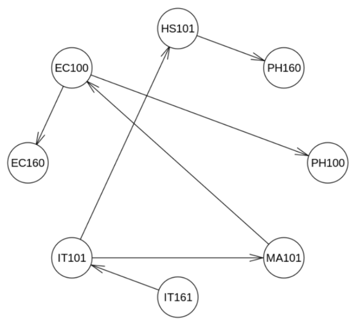

Fig. 3. Dependency Graph of different courses

The Bayesian network was created using the Hill-Climbing algorithm with Cooper and Herskovits' K2 scoring metric. It has 8 nodes and 7 arrows pointing in different directions, with an average Markov blanket size of 1.75 and an average branching factor of 0.88. The learning process was improved and tested 105 times. The network structure shows how variables like IT161, IT101, MA101, HS101, EC100, PH160, EC160, and PH100 are connected and affect each other's probabilities.

IV. Probelm II: Learning the CPTs for Each Course Node

Problem Statement: Using the data, learn the CPTs for each course node.

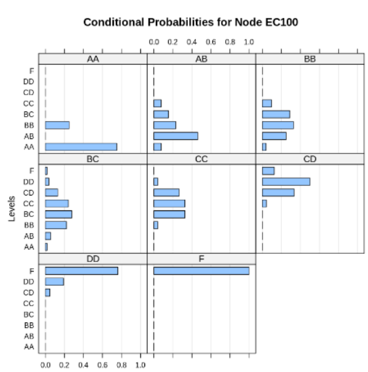

Fig. 4. Conditional probabilities for Node EC100

```
  Bayesian network parameters

  Parameters of node EC100 (multinomial distribution)

Conditional probability table:

     MA101
EC100         AA         AB         BB         BC         CC         CD
   AA 0.75000000 0.07692308 0.03846154 0.01851852 0.00000000 0.00000000
   AB 0.00000000 0.46153846 0.25000000 0.05555556 0.00000000 0.00000000
   BB 0.25000000 0.23076923 0.32692308 0.22222222 0.04081633 0.00000000
   BC 0.00000000 0.15384615 0.28846154 0.27777778 0.32653061 0.00000000
   CC 0.00000000 0.07692308 0.09615385 0.24074074 0.32653061 0.04166667
   CD 0.00000000 0.00000000 0.00000000 0.12962963 0.26530612 0.33333333
   DD 0.00000000 0.00000000 0.00000000 0.03703704 0.04081633 0.50000000
   F  0.00000000 0.00000000 0.00000000 0.01851852 0.00000000 0.12500000
     MA101
EC100         DD          F
   AA 0.00000000 0.00000000
   AB 0.00000000 0.00000000
   BB 0.00000000 0.00000000
   BC 0.00000000 0.00000000
   CC 0.00000000 0.00000000
   CD 0.04761905 0.00000000
...
   CD 0.13043478 0.08333333
   DD 0.52173913 0.58333333
   F  0.04347826 0.33333333
```

```
  Parameters of node EC160 (multinomial distribution)

Conditional probability table:

     EC100
EC160         AA         AB         BB         BC         CC         CD
   AA 0.42857143 0.22727273 0.05714286 0.04166667 0.00000000 0.00000000
   AB 0.42857143 0.22727273 0.08571429 0.04166667 0.08333333 0.00000000
   BB 0.14285714 0.31818182 0.20000000 0.22916667 0.08333333 0.03448276
   BC 0.00000000 0.22727273 0.42857143 0.43750000 0.36111111 0.17241379
   CC 0.00000000 0.00000000 0.22857143 0.25000000 0.30555556 0.34482759
   CD 0.00000000 0.00000000 0.00000000 0.00000000 0.11111111 0.27586207
   DD 0.00000000 0.00000000 0.00000000 0.00000000 0.05555556 0.17241379
   F  0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
     EC100
EC160         DD          F
   AA 0.00000000 0.00000000
   AB 0.00000000 0.00000000
   BB 0.05000000 0.00000000
   BC 0.00000000 0.00000000
   CC 0.25000000 0.02857143
   CD 0.55000000 0.40000000
   DD 0.15000000 0.34285714
   F  0.00000000 0.22857143
```

Fig. 5. Conditional Probability Tables

Similarly, we calculated Conditional Probability Tables (CPTs) for every course in the dataset. These tables show the likelihood of different outcomes for each course, based on the outcomes of its related courses in the Bayesian network. This helps us understand how the courses influence each other's outcomes, making it easier to make informed decisions based on the data.

V. Probelm III: Predicting Student Grades

Problem Statement: What grade will a student get in PH100 if he earns DD in EC100, CC in IT101 and CD in MA101.

With the Bayesian Network and its Conditional Probability Tables (CPTs) that we created, we can predict a student's grade in PH100. We do this by putting in the grades we already know and then figuring out the chances of getting different grades in PH100 based on that information.

The implemented code predicts the grade in PH100 based on the given evidence.

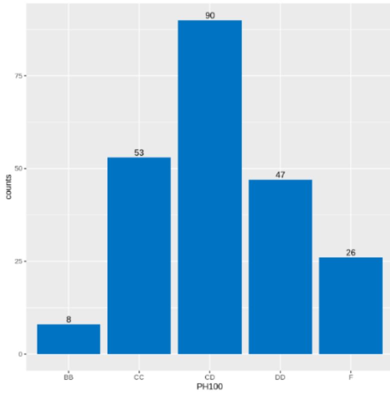

Fig. 6. Predicted Grade in PH100

VI. Probelm IV: Building a Naive Bayes Classifier

Problem Statement: The last column in the data file indicates whether a student qualifies for an internship program or not. From the given data, take 70 percent data for training and build a naive Bayes classifier (considering that the grades earned in different courses are independent of each other) which takes in the student's performance and returns the qualification status with a probability. Test your classifier on the remaining 30 percent data. Repeat this experiment for 20 random selection of training and testing data. Report results about the accuracy of your classifier.

A Naive Bayes classifier is created using 70% of the data for training. The classifier assumes that the grades earned in different courses are independent of each other. Then, the classifier is tested on the remaining 30% of the data. This process is repeated 20 times with randomly chosen training and testing data, and the accuracy of the classifier is recorded.

Achieved average accuracy = 96.93%

```
Accuracy for iteration 1 : 97.14 %
Accuracy for iteration 2 : 98.57 %
Accuracy for iteration 3 : 97.14 %
Accuracy for iteration 4 : 100 %
Accuracy for iteration 5 : 98.57 %
Accuracy for iteration 6 : 98.57 %
Accuracy for iteration 7 : 98.57 %
Accuracy for iteration 8 : 98.57 %
Accuracy for iteration 9 : 94.29 %
Accuracy for iteration 10 : 98.57 %
Accuracy for iteration 11 : 97.14 %
Accuracy for iteration 12 : 91.43 %
Accuracy for iteration 13 : 95.71 %
Accuracy for iteration 14 : 97.14 %
Accuracy for iteration 15 : 95.71 %
Accuracy for iteration 16 : 97.14 %
Accuracy for iteration 17 : 95.71 %
Accuracy for iteration 18 : 97.14 %
Accuracy for iteration 19 : 94.29 %
Accuracy for iteration 20 : 97.14 %
```

Fig. 7. Accuracy achieved in each iteration

VII. Problem V: Considering Dependencies in Naive Bayes Classification

Problem Statement: Repeat 4, considering that the grades earned in different courses may be dependent.

In this case, we adjust the Naive Bayes classifier to recognize that there could be connections between the grades in different courses. This means we change how the classifier calculates probabilities to consider these potential relationships.

Achieved average accuracy = 97.61%

```
Accuracy for iteration 1 : 97.14 %
Accuracy for iteration 2 : 98.57 %
Accuracy for iteration 3 : 97.14 %
Accuracy for iteration 4 : 100 %
Accuracy for iteration 5 : 98.57 %
Accuracy for iteration 6 : 98.57 %
Accuracy for iteration 7 : 98.57 %
Accuracy for iteration 8 : 98.57 %
Accuracy for iteration 9 : 94.29 %
Accuracy for iteration 10 : 98.57 %
Accuracy for iteration 11 : 97.14 %
Accuracy for iteration 12 : 91.43 %
Accuracy for iteration 13 : 95.71 %
Accuracy for iteration 14 : 97.14 %
Accuracy for iteration 15 : 95.71 %
Accuracy for iteration 16 : 97.14 %
Accuracy for iteration 17 : 95.71 %
Accuracy for iteration 18 : 97.14 %
Accuracy for iteration 19 : 94.29 %
Accuracy for iteration 20 : 97.14 %
```

Fig. 8. Accuracy achieved in each iteration(considering dependencies)

VIII. Conclusion

Studying graphical models, Bayesian Networks, and Naive Bayes classification helps us predict outcomes in uncertain situations. Understanding how things are connected and estimating these connections accurately in a Bayesian Network is crucial for reliable predictions. Also, deciding whether to assume independence or consider relationships between features makes a big difference in how well a Naive Bayes classifier works.

References

[1] http://gauss.inf.um.es/umur/xjurponencias/talleres/J3.pdf
[2] https://www.bnlearn.com/