Experiment 5 Graphical Models and Bayesian Networks: Inference and Classification in R

***Abstract*—This lab focuses on using graphical models and Bayesian inference in R to study how course grades are related by building Bayesian Networks. It also uses a naive Bayes classifier to make predictions. The dataset includes student grades from different courses and is used to create Conditional Probability Tables (CPTs) and predict whether a student qualifies for an internship program. The experiments divide the data into 70% for training and 30% for testing, and are repeated 20 times to measure how accurate the classifier is under both independent and dependent grade conditions.**

I. Introduction

In this lab, we explore how to use graphical models and Bayesian inference to handle uncertainty in data. The focus is on educational data, where we will build Bayesian Networks using R to model the relationships between student grades in different courses. We also explore classification, specifically using the naive Bayes method, which is a simple but effective way to classify data based on probability. The dataset includes student grades and their qualification status for an internship. The goal is to train and evaluate the naive Bayes classifier, both by assuming the grades are independent and by considering the dependencies between them. This lab serves as an introduction to using probabilistic models for data analysis and classification.

II. Fundamentals

A. *Bayesian Inference and Graphical Models*

Graphical models use simple diagrams to show how different variables are related, which helps us understand uncertainty better. Bayesian inference is a method that lets us change our beliefs about unknown things when we get new information. Together, these ideas help us make sense of complex data and make better decisions.

Before starting practical work, it's important to understand the basics of graphical models and Bayesian inference. This includes learning about conditional probability, Bayes' theorem, and how graphical models show relationships between variables. These ideas help us make better decisions when dealing with uncertainty.

B. *Constructing Bayesian Networks in R*

Bayesian Networks are types of graphical models that show how different variables are related using a directed acyclic graph (DAG), which is a diagram that connects the variables without any cycles. In R, there are packages like 'bnlearn' that make it easy to build, visualize, and analyze these Bayesian Networks.

Learners will understand how to create the structure of Bayesian Networks, set up Conditional Probability Tables (CPTs), and carry out tasks like asking probabilistic questions and learning from data.

C. *Learning Dependencies from Data*

Bayesian Networks often rely on expert insights to determine how variables are related, but these relationships can also be learned from data. Methods like structure learning and parameter learning help uncover these connections and build Conditional Probability Tables (CPTs) from observed data. Participants will learn how to use data to identify the structure and parameters of Bayesian Networks, exploring algorithms such as constraint-based methods, score-based methods, and hybrid approaches to discover these relationships

D. *Naive Bayes Classification*

Naive Bayes is a straightforward probabilistic model that applies Bayes' theorem while assuming that the features are independent of one another. Although it is a basic method, it is commonly employed in applications such as document classification, sentiment analysis, and email filtering.

Students will comprehend the underlying principles of naive Bayes classification, including the calculation of class probabilities using Bayes' theorem and the assumption of feature independence.

Bayes Theorem:

$$P(C|X) = \frac{P(X|C) \cdot P(C)}{P(X)}$$

Naive Bayes Classification Formula:

$$P(C|X) \propto P(C) \cdot P(x_1|C) \cdot P(x_2|C) \cdot \ldots \cdot P(x_n|C)$$

Where:

- $P(C|X)$ is the **posterior probability**: the probability of class C given the features X.
- $P(X|C)$ is the **likelihood**: the probability of the features X given the class C.
- $P(C)$ is the **prior probability** of the class C.
- $P(X)$ is the **marginal likelihood** or evidence: the total probability of the features X.

E. *Classifier Implementation and Evaluation*

Setting up a Naive Bayes classifier involves training the model on labeled data by calculating class priors (how often each outcome occurs) and conditional probabilities (how likely each feature is for each outcome). Once trained, the model can predict outcomes for new data based on these probabilities. To assess the model's performance, you use evaluation metrics like accuracy, precision, recall, and F1 score. Learners will gain practical experience using R to implement this classifier, covering data preparation, model training, prediction, and evaluation, ultimately applying these techniques to real-world educational data analysis tasks

III. Problem Statement

A. *Problem I*

IV. Problem Statement

The dataset *2020_bn_nb_data.txt* includes student grades from various courses. The objective is to model the relationships between these courses and learn the Conditional Probability Tables (CPTs). Furthermore, the aim is to predict a student's grade in PH100 based on their grades in other courses.

We will use the last column, which indicates internship qualification, to train a Naive Bayes classifier with 70% of the data. This classifier will predict if a student qualifies for an internship based on their grades, assuming that the courses are independent. We'll evaluate its accuracy on the remaining 30% of the data over 20 runs. Afterward, we will repeat the experiment, this time considering any potential relationships between the grades earned in different courses..

V. Implementation Method

1) **Data Preprocessing**:
   - Load the dataset (*2020_bn_nb_data.txt*) into R.
   - Analyze the data to gain insights into its structure and content.
   - Clean the data as needed by addressing any missing values or outliers.
2) **Bayesian Network Construction**:
   - Use the `bnlearn` package to construct a Bayesian Network.
   - Establish the network structure using domain knowledge or apply structure learning algorithms to derive it from the data.
   - Estimate Conditional Probability Tables (CPTs) for each node in the network.
3) **Grade Prediction in PH100**:
   - Given a student's grades in other courses, use the Bayesian Network to predict the grade in PH100.
   - Use the known grades to query the network and determine the most probable grade in PH100.
4) **Naive Bayes Classifier**:
   - Split the dataset into training and testing sets (70% training, 30% testing).
   - Train a naive Bayes classifier using the training data, assuming independence between course grades.
   - Evaluate the classifier's performance on the testing data using accuracy and other relevant metrics.
5) **Assessing Classifier Accuracy**:
   - Repeat the training and testing of the Naive Bayes classifier 20 times using randomly selected data.
   - Document the accuracy of the classifier during each iteration.
6) **Considering Dependencies**:
   - Adjust the Naive Bayes classifier to account for possible relationships between course grades.
   - Redo the training and testing process, evaluating the accuracy of the classifier based on this revised approach.

VI. Solution

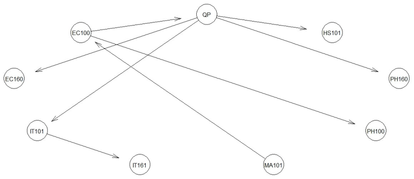
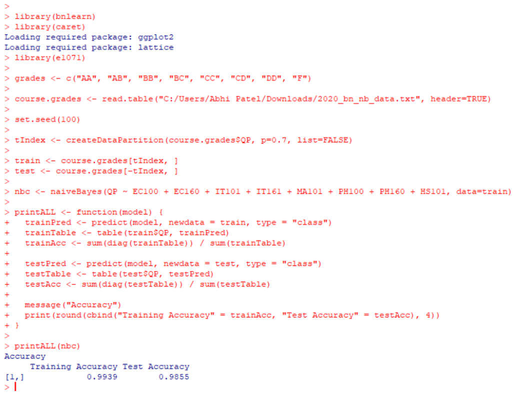


Figure 1: Dependencies



```
>
> library(bnlearn)
> library(caret)
Loading required package: ggplot2
Loading required package: lattice
> library(e1071)
>
> grades <- c("AA", "AB", "BB", "BC", "CC", "CD", "DD", "F")
>
> course.grades <- read.table("C:/Users/Abhi Patel/Downloads/2020_bn_nb_data.txt", header=TRUE)
>
> set.seed(100)
>
> tIndex <- createDataPartition(course.grades$QP, p=0.7, list=FALSE)
>
> train <- course.grades[tIndex, ]
> test <- course.grades[-tIndex, ]
>
> nbc <- naiveBayes(QP ~ EC100 + EC160 + IT101 + IT161 + MA101 + PH100 + PH160 + HS101, data=train)
>
> printALL <- function(model) {
+   trainPred <- predict(model, newdata = train, type = "class")
+   trainTable <- table(train$QP, trainPred)
+   trainAcc <- sum(diag(trainTable)) / sum(trainTable)
+
+   testPred <- predict(model, newdata = test, type = "class")
+   testTable <- table(test$QP, testPred)
+   testAcc <- sum(diag(testTable)) / sum(testTable)
+
+   message("Accuracy")
+   print(round(cbind("Training Accuracy" = trainAcc, "Test Accuracy" = testAcc), 4))
+ }
>
> printALL(nbc)
Accuracy
     Training Accuracy Test Accuracy
[1,]            0.9939        0.9855
> 
```

Figure 2: Final Solution

VII. Conclusion

In this lab, we learned about graphical models, Bayesian inference, and classification within the realm of educational data analysis. By building Bayesian Networks, we were able to model how student grades in different courses relate to one another and create Conditional Probability Tables (CPTs) from the data. We also used a Naive Bayes classifier to predict whether students qualify for a internships based on their grades, initially assuming that course grades are independent. Through our experiments, we found that while the classifier performed well without considering dependencies, taking these relationships into account improved its accuracy and better reflected real-world situations. Overall, this lab emphasized the importance of using graphical models and understanding dependencies in order to make more accurate predictions and informed decisions in educational contexts.