732A96 Advanced Machine Learning

LAB 1: GRAPHICAL MODELS

Johannes Hedström

STIMA
Department of Computer and Information Science
Linköpings universitet

2024-09-05

Contents

```r
# packages

library(bnlearn)
library(gRain)
```

1 Question 1

Show that multiple runs of the hill-climbing algorithm can return non-equivalent Bayesian network (BN) structures. Explain why this happens. Use the Asia dataset which is included in the bnlearn package. To load the data, run data("asia"). Recall from the lectures that the concept of non-equivalent BN structures has a precise meaning. Hint: Check the function hc in the bnlearn package. Note that you can specify the initial structure, the number of random restarts, the score, and the equivalent sample size (a.k.a imaginary sample size) in the BDeu score. You may want to use these options to answer the question. You may also want to use the functions plot, arcs, vstructs, cpdag and all.equal.

```r
# loading data
data("asia")

print(head(asia,5)) # table of the data to get a view of it
```

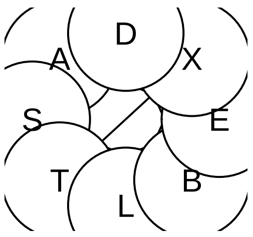
```
##    A   S   T  L   B   E   X   D
## 1 no yes  no no yes  no  no yes
## 2 no yes  no no  no  no  no  no
## 3 no  no yes no  no yes yes yes
## 4 no  no  no no yes  no  no yes
## 5 no  no  no no  no  no  no yes
```

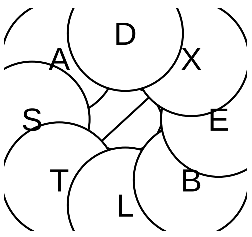
```r
# hill climbing with score = bde for Baysian drichlet equivalent
# iss for imaginary sample size
set.seed(12345) # seed to get same results

r1 <- hc(asia,restart = 10 ,score='bde', iss=5)

r2 <- hc(asia,restart = 10 ,score='bde', iss=5)

par(mfrow= c(1,2))
plot(cpdag(r1))
plot(cpdag(r2))
```





Hill climbing algorithm is a greedy search that relies on the initial conditions, which means that different starting positions will most likely lead to different local optimas. And thats why the algortihm produces non indentical structures

2 Question 2

Learn a BN from 80 % of the Asia dataset. The dataset is included in the bnlearn package. To load the data, run data("asia"). Learn both the structure and the parameters. Use any learning algorithm and settings that you consider appropriate. Use the BN learned to classify the remaining 20 % of the Asia dataset in two classes: S = yes and S = no. In other words, compute the posterior probability distribution of S for each case and classify it in the most likely class. To do so, you have to use exact or approximate inference with the help of the bnlearn and gRain packages, i.e. you are not allowed to use functions such as predict. Report the confusion matrix, i.e. true/false positives/negatives. Compare your results with those of the true Asia BN, which can be obtained by running $dag = model2network("[A][S][T|A][L|S][B|S][D|B:E][E|T:L][X|E]")$.

3 Question 3

In the previous exercise, you classified the variable S given observations for all the rest of the variables. Now, you are asked to classify S given observations only for the so-called Markov blanket of S, i.e. its parents plus its children plus the parents of its children minus S itself. Report again the confusion matrix. Hint: You may want to use the function mb from the bnlearn package.

4 Question 4

Repeat the exercise (2) using a naive Bayes classifier, i.e. the predictive variables are independent given the class variable. See p. 380 in Bishop's book or Wikipedia for more information on the naive Bayes classifier.

Model the naive Bayes classifier as a BN. You have to create the BN by hand, i.e. you are not allowed to use the function naive.bayes from the bnlearn package. Hint: Check http://www.bnlearn.com/examples/dag/ to see how to create a BN by hand.

5 Question 5

Explain why you obtain the same or different results in the exercises (2-4)