Bayesian Network Learning

On the Default and Titanic data sets

Karen Mazidi

Default data set

Let's try bnlearn on some familiar data sets. First, the Default data set from ISLR.

```
library(bnlearn)
```

```
##
## Attaching package: 'bnlearn'
```

```
## The following object is masked from 'package:stats':
##
##     sigma
```

```
library(ISLR)
data("Default")
```

Build the model

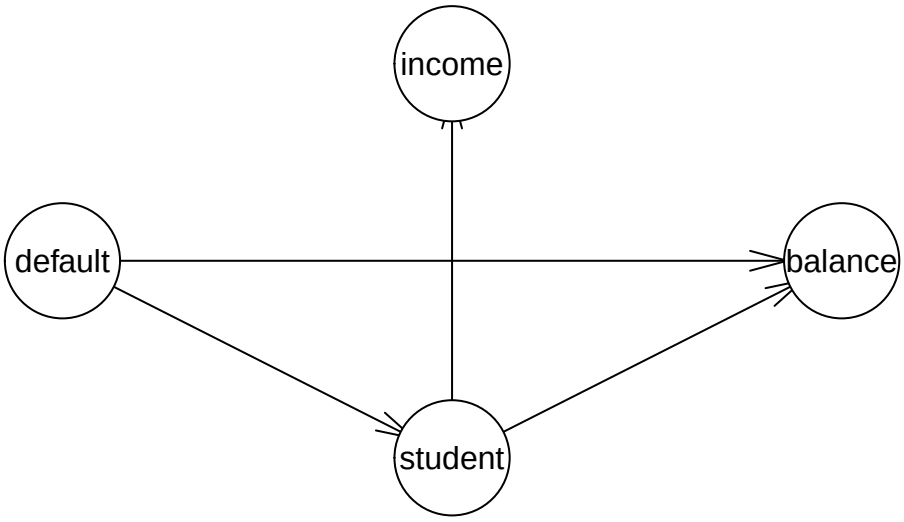
Use the hill-climbing algorithm to create the model, then display it.

The model seems to fit what we learned by doing ML on this data set before. Income is not a good predictor for default, notice there is no link between them.

```
bn_df <- data.frame(Default)
str(bn_df)
```

```
## 'data.frame':    10000 obs. of  4 variables:
##  $ default: Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ student: Factor w/ 2 levels "No","Yes": 1 2 1 1 1 2 1 2 1 1 ...
##  $ balance: num  730 817 1074 529 786 ...
##  $ income : num  44362 12106 31767 35704 38463 ...
```

```
res <- hc(bn_df)
plot(res)
```



Probabilities

This is a really unbalanced data set!

```
fittedbn <- bn.fit(res, data=bn_df)
print(fittedbn$default)
```

```
##
##   Parameters of node default (multinomial distribution)
##
## Conditional probability table:
##      No    Yes
## 0.9667 0.0333
```

Try some predictions

```
cpquery(fittedbn, event=(default=="Yes"), evidence = (student=="Yes"))
```

```
## [1] 0.04416839
```

```
cpquery(fittedbn, event=(default=="Yes"), evidence = (balance>1200))
```

```
## [1] 0.175713
```

```
cpquery(fittedbn, event=(default=="Yes"), evidence = (student=="Yes" & balance>1200))
```

```
## [1] 0.1038697
```

Titanic

Now let's try a subset of the Titanic data set. We had to a little data fiddling because the bnlearn didn't like NAs and ints.

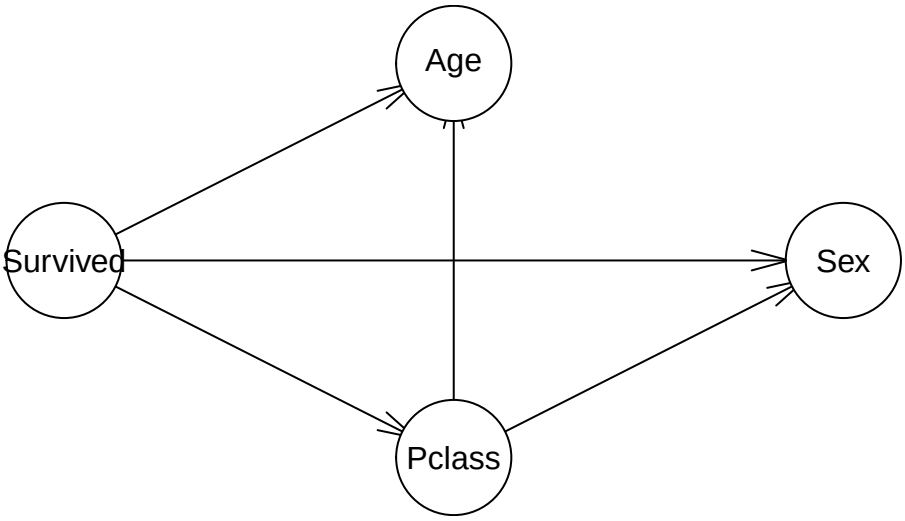
```
df <- read.csv('train.csv', header=T, na.strings=c(""))
df <- df[1:750, c(2,3,5,6)]  # Survived, Pclass, Sex, Age
bn_titan <- df[complete.cases(df), ]
bn_titan$Survived <- as.factor(bn_titan$Survived)
bn_titan$Pclass <- as.factor(bn_titan$Pclass)
bn_titan$Sex <- as.factor(bn_titan$Sex)
str(bn_titan)
```

```
## 'data.frame':    597 obs. of  4 variables:
##  $ Survived: Factor w/ 2 levels "0","1": 1 2 2 2 1 1 1 2 2 2 ...
##  $ Pclass  : Factor w/ 3 levels "1","2","3": 3 1 3 1 3 1 3 3 2 3 ...
##  $ Sex     : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 1 1 1 ...
##  $ Age     : num  22 38 26 35 35 54 2 27 14 4 ...
```

Build the net

This model is not surprising. Age and Sex and Pclass were found to be good predictors in previous experiments.

```
res_titan <- hc(bn_titan)
plot(res_titan)
```



Print the probabilities

This breaks down the probabilities nicely.

```
fitted_bn_titan <- bn.fit(res_titan, data=bn_titan)
print(fitted_bn_titan)
```

```
##
##   Bayesian network parameters
##
##   Parameters of node Survived (multinomial distribution)
##
## Conditional probability table:
##          0         1
## 0.5946399 0.4053601
##
##   Parameters of node Pclass (multinomial distribution)
##
## Conditional probability table:
##
##       Survived
## Pclass         0         1
##      1 0.1633803 0.4214876
##      2 0.2112676 0.2975207
##      3 0.6253521 0.2809917
##
##   Parameters of node Sex (multinomial distribution)
##
## Conditional probability table:
##
## , , Pclass = 1
##
##         Survived
## Sex               0          1
##   female 0.05172414 0.63725490
##   male   0.94827586 0.36274510
##
## , , Pclass = 2
```

```
##
##         Survived
## Sex                0          1
##   female 0.05333333 0.83333333
##   male   0.94666667 0.16666667
##
## , , Pclass = 3
##
##         Survived
## Sex                0          1
##   female 0.21171171 0.55882353
##   male   0.78828829 0.44117647
##
##
##   Parameters of node Age (conditional Gaussian distribution)
##
## Conditional density: Age | Survived + Pclass
## Coefficients:
##                      0         1         2         3         4         5
## (Intercept)  44.49138  34.81294  34.16000  26.22681  26.43919  21.63971
## Standard deviation of the residuals:
##        0         1         2         3         4         5
## 15.74425  13.58176  12.28907  14.09778  12.22321  12.20613
## Discrete parents' configurations:
##    Survived  Pclass
## 0         0       1
## 1         1       1
## 2         0       2
## 3         1       2
## 4         0       3
## 5         1       3
```

Predict

Try some predictions based on the net

```
cpquery(fitted_bn_titan, event = (Survived==1), evidence = (Pclass==1))
```

```
## [1] 0.6208178
```

```
cpquery(fitted_bn_titan, event = (Survived==1), evidence = (Age<9) )
```

```
## [1] 0.5246377
```

```
cpquery(fitted_bn_titan, event = (Survived==1), evidence = (Pclass==1 & Age<=9) )
```

```
## [1] 0.8148148
```

```
cpquery(fitted_bn_titan, event = (Survived==1), evidence = (Pclass==1 & Age>9) )
```

```
## [1] 0.6355556
```

```
cpquery(fitted_bn_titan, event = (Survived==1), evidence = (Sex=="female") )
```

```
## [1] 0.7788043
```

```
cpquery(fitted_bn_titan, event = (Survived==1), evidence = (Sex=="male" & Age>21) )
```

```
## [1] 0.1819322
```