

Bayes Network

Demo 1

Adapted from: <https://www.r-bloggers.com/bayesian-network-in-r-introduction/>

Bayesian networks (BN) are graphical models that encode the conditional probability between predictors in a directed acyclic graph.

BN advantages over other unsupervised algorithms:

- easy to exploit expert knowledge in BN models
- BN models are robust to noisy, missing, sparse data
- parameters are understandable, interpretable

```
library(bnlearn)
```

```
##  
## Attaching package: 'bnlearn'  
## The following object is masked from 'package:stats':  
##  
##      sigma  
data(coronary)  
names(coronary)  
  
## [1] "Smoking" "M. Work" "P. Work" "Pressure" "Proteins" "Family"
```

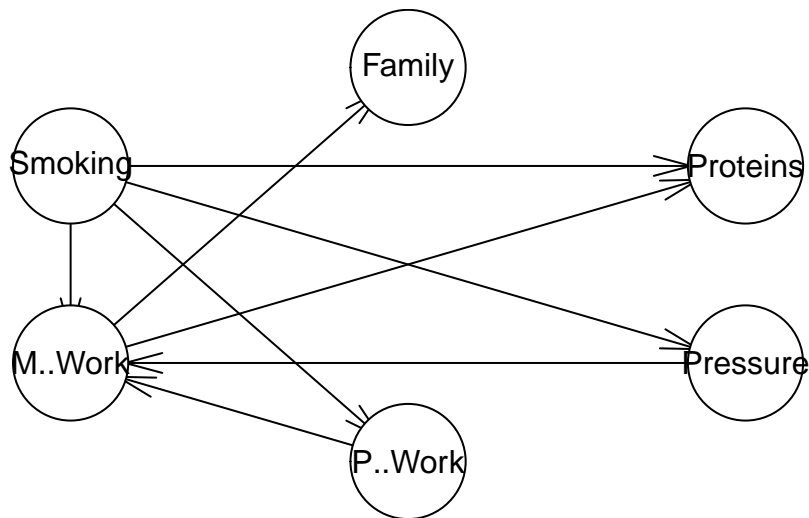
The coronary data contains the following information:

- smoking - yes/no
- M. Work - strenuous mental work yes/no
- P.Work - strenuous physical work yes/no
- Pressure - yes/no <140
- Proteins - 2-level factor with levels (ratio of beta and alpha lipoproteins)
- Family - two-level factor neg/pos for family anamnesis of coronary heart disease

Create the network

In this example we use the “max-min” hill climbing algorithm.

```
bn_df <- data.frame(coronary)  
res <- hc(bn_df)  
plot(res)
```



The causality between some nodes is intuitive and some doesn't seem to make sense. Ex: M.Work and Family. Let's remove that link.

```
res$arcs <- res$arcs[-which((res$arcs[, 'from'] == "M..Work" & res$arcs[, 'to'] == "Family")),]
```

CPT

We need to learn the conditional probability tables (CPT) at each node. The `bn.fit` function runs the EM algorithm to learn CPT for different nodes in the above graph. We can print them for different nodes.

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

```
##
## Parameters of node Proteins (multinomial distribution)
##
## Conditional probability table:
##
## , , M..Work = no
##
## Smoking
## Proteins    no    yes
## <3 0.6685824 0.6167763
## >3 0.3314176 0.3832237
##
## , , M..Work = yes
##
## Smoking
## Proteins    no    yes
## <3 0.5671982 0.3235294
## >3 0.4328018 0.6764706
```

Inference

The BN is ready and we can start inferring from the network.

Although Proteins variable is conditioned on 2 variables, we did the query based on the available evidence on only one variables.

Next we query the network. What is the chance that a non-smoker has a Proteins level less than 3?

The result is 0.61.

```
cpquery(fittedbn, event = (Proteins=="<3"), evidence = ( Smoking=="no" ) )
```

```
## [1] 0.6210046
```

What is the chance that a non-smoker with blood pressure greater than 140 has a Proteins level less than 3?

The result is 0.62.

```
cpquery(fittedbn, event = (Proteins=="<3"), evidence = ( Smoking=="no" & Pressure==">140" ) )
```

```
## [1] 0.6437346
```

We can also move in the opposite direction of an arc between two nodes. Let's see if a person's Proteins level is greater than 3, then what is the chance that his or her Pressure level is greater than 140?

The result is the Pressure is greater than 140 with probability 0.41.

```
cpquery(fittedbn, event = (Pressure==">140"), evidence = ( Proteins=="<3" ) )
```

```
## [1] 0.4253076
```