VE414 Lecture 26

Jing Liu

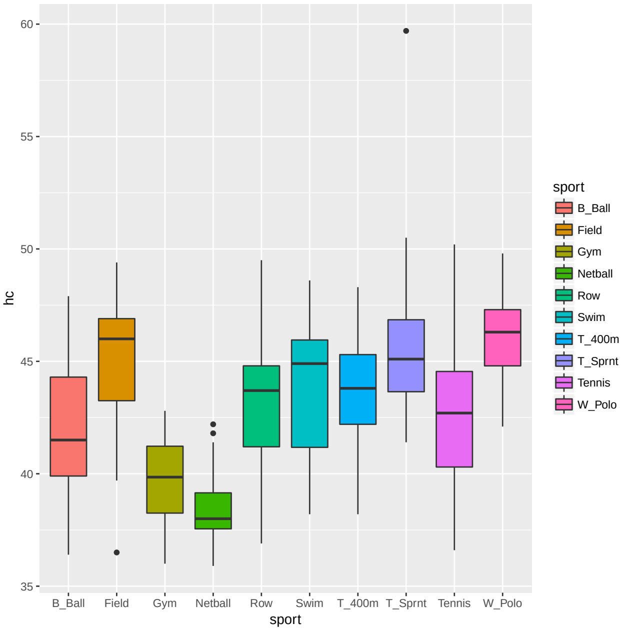
UM-SJTU Joint Institute

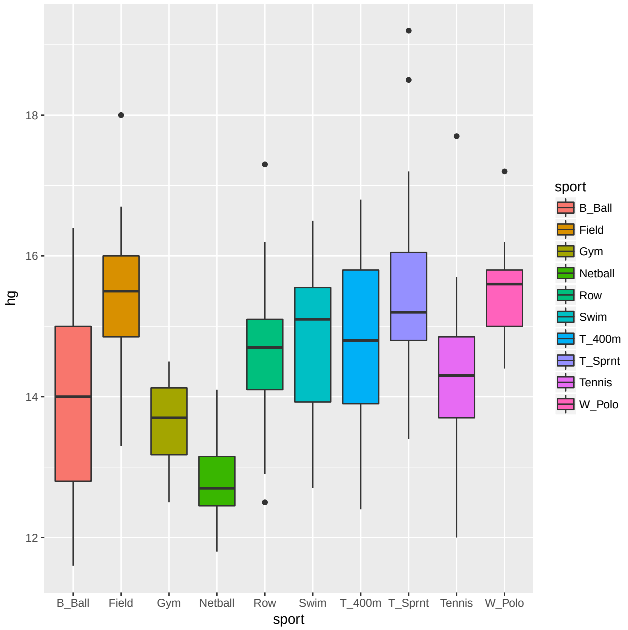
December 3, 2019

- Let us consider the following dataset to illustrate how to use R to build BN

`rcc`	Red blood cell count
`wcc`	White blood cell count
`hc`	Hematocrit
`hg`	Hemaglobin
`ferr`	Plasma ferritins
`bmi`	Body mass index
`pcBfat`	Percentage body fat
`lbm`	Lean body mass
`ht`	Height
`wt`	Weight
`sex`	Gender
`Sport`	Various types of sport that the athletes are in

- Suppose we are interested in how various characteristics of the blood varied with sport body size and sex of the athlete. It consists of 202 cases.
- We are particularly interested in `hc` and `hg` levels for various sports.


60
55
50
45
40
35
hc
B_Ball
Field
Gym
Netball
Row
Swim
T_400m
T_Sprnt
Tennis
W_Polo
sport
sport
B_Ball
Field
Gym
Netball
Row
Swim
T_400m
T_Sprnt
Tennis
W_Polo


hg
18
16
14
12
B_Ball
Field
Gym
Netball
Row
Swim
T_400m
T_Sprnt
Tennis
W_Polo
sport
sport
B_Ball
Field
Gym
Netball
Row
Swim
T_400m
T_Sprnt
Tennis
W_Polo

- To start with a simple network, we turn hc and hg into binary

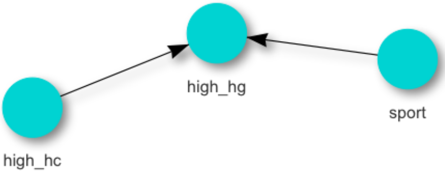
```
> ais$high_hc = as.factor(ais$hc > median(ais$hc))
> ais$high_hg = as.factor(ais$hg > median(ais$hg))
```

- Defining the nodes

```
> structure = empty.graph(
+   c("high_hc", "high_hg", "sport"))
```

- Defining the links

```
> modelstring(structure) =
+   "[high_hc][sport][high_hg|sport:high_hc]"
```



- Only the following three sports are included

```
> ais.sub = ais[
+   ais$sport %in% c("Netball", "Tennis", "W_Polo"),
+                c("high_hc", "high_hg", "sport")]
> head(ais.sub)
```

```
   high_hc high_hg   sport
36   FALSE   FALSE Netball
37   FALSE   FALSE Netball
38   FALSE   FALSE Netball
39   FALSE   FALSE Netball
40   FALSE   FALSE Netball
41   FALSE   FALSE Netball
```

- Asking R to estimate conditional probabilities using frequentist's approach

```
> ais.sub$sport = factor(ais.sub$sport)
>
> bn.mod = bn.fit(structure,
+                 method = "mle", data = ais.sub)
```

```
> cat("P(high hg) =",
+     cpquery(bn.mod, (high_hg == "TRUE"), TRUE),
+     "\n")
```

```
P(high hg) = 0.2092
```

```
> cat("P(high hg | water polo and high hc) =",
+     cpquery(bn.mod, (high_hg=="TRUE"),
+             (sport == "W_Polo" &
+                high_hc == "TRUE")), "\n")
```

```
P(high hg | water polo and high hc) = 0.9292929
```

```
> cat("P(water polo | high hg and have high hc) =",
+     cpquery(bn.mod, (sport=="W_Polo"),
+             (high_hg == "TRUE" &
+                high_hc == "TRUE")), "\n")
```

```
P(water polo | high hg and have high hc) = 0.6507937
```

- Now consider a hybrid Bayesian network, we use the original hc and hg.

```
> ais.sub = ais[
+   ais$sport %in% c("Netball", "Tennis", "W_Polo"),
+   c("hc", "hg", "sport")]
>
> ais.sub$sport = factor(ais.sub$sport)
>
> head(ais.sub)
```

```
     hc   hg   sport
36 42.2 13.6 Netball
37 38.0 12.7 Netball
38 37.5 12.3 Netball
39 37.7 12.3 Netball
40 38.7 12.8 Netball
41 36.6 11.8 Netball
```

- Defining the new nodes

```
> structure = empty.graph(c("hc", "hg", "sport"))
```

- Defining the links

```
> modelstring(structure) = "[hc][sport][hg|sport:hc]"
```

- The syntax for fitting is the same

```
> bn.mod = bn.fit(structure,
+                 method = "mle", data = ais.sub)
```

- Inference can be done using similar syntax

```
> cat("P(hg > 14 | water polo and high hc) =",
+     cpquery(bn.mod, (hg > 14),
+             (sport == "W_Polo" & hc > 42 )), "\n")
```

```
P(hg > 14 | water polo and high hc) = 0.9802891
```

- Notice this BN corresponds to linear regression

```
> bn.mod
```

```
  Parameters of node hg (conditional Gaussian distribution)

Conditional density: hg | hc + sport
```

```
Coefficients:
                     0           1           2
(Intercept)   1.5550754  -2.7611358  -0.1173597
hc            0.2929909   0.4019544   0.3398915
```

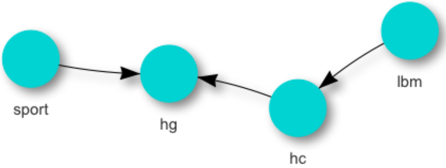
```
> fit = lm(hg~sport*hc, data = ais.sub)
> summary(fit)
```

```

Call:
lm(formula = hg ~ sport * hc, data = ais.sub)

Coefficients:
               Estimate Std. Error t value Pr(>t)
(Intercept)     1.55508    1.43580   1.083   0.2845
sportTennis    -4.31621    1.77311  -2.434   0.0190 *
sportW_Polo    -1.67244    2.29697  -0.728   0.4703
hc              0.29299    0.03732   7.851 5.68e-10 ***
sportTennis:hc  0.10896    0.04460   2.443   0.0185 *
sportW_Polo:hc  0.04690    0.05394   0.870   0.3892
```

- Bayesian Network is a pathway to build more complicated models.



- Defining the new nodes

```
> structure = empty.graph(
+   c("hc", "hg", "sport", "lbm"))
```

- Defining the links

```
> modelstring(structure) =
+   "[lbm][hc|lbm][sport][hg|sport:hc]"
```

```
> ais.sub = ais[
+   ais$sport %in% c("Netball", "Tennis", "W_Polo"),
+   c("hc", "hg", "sport", "lbm")]
> ais.sub$sport = factor(ais.sub$sport)
> lbm.mod = bn.fit(structure, data = ais.sub)
> lbm.mod
```

```
  Parameters of node hg (conditional Gaussian distribution)

Conditional density: hg | hc + sport
```

```
Coefficients:
                     0           1           2
(Intercept)   1.5550754  -2.7611358  -0.1173597
hc            0.2929909   0.4019544   0.3398915
```

```
> bn.mod
```

```
  Parameters of node hg (conditional Gaussian distribution)

Conditional density: hg | hc + sport
```

```
Coefficients:
                     0           1           2
(Intercept)   1.5550754  -2.7611358  -0.1173597
hc            0.2929909   0.4019544   0.3398915
```

Q: What is the difference?

- We have the following without lbm.

```
  Parameters of node sport (multinomial distribution)

Conditional probability table:
   Netball    Tennis    W_Polo
0.4509804 0.2156863 0.3333333
```

```
  Parameters of node hc (Gaussian distribution)

Conditional density: hc
Coefficients:
(Intercept)
   41.82353
Standard deviation of the residuals: 4.092363
```

- With lbm, we have

```
  Parameters of node lbm (Gaussian distribution)

Conditional density: lbm
Coefficients:
(Intercept)
   61.91667
Standard deviation of the residuals: 12.00722
```

- The marginal probabilities of sport are the same

```
  Parameters of node sport (multinomial distribution)

Conditional probability table:
   Netball    Tennis    W_Polo
0.4509804 0.2156863 0.3333333
```

- With lbm, we have the following in addition to the above

```
  Parameters of node hc (Gaussian distribution)

Conditional density: hc | lbm
Coefficients:
(Intercept)          lbm
 26.5212185    0.2471436
Standard deviation of the residuals: 2.846647
```

```
> cat("P(hg > 14 | water polo and LBM > 65 kg) =",
+       cpquery(hybrid.bn.lbm.mod, (hg > 14),
+                    (sport == "W_Polo" & lbm > 65 )),
+       "\n")
```

```
P(hg > 14 | water polo and LBM > 65 kg) = 0.8199181
```