An Introduction to Graphical Models

Pratik Shah

29 May 2020

Graphical Models

1. Directed

a. Bayesian Network (Belief Network)
b. Hidden Markov Model
c. Kalman Filter
d. etc…

2. Undirected

a. Markov Network (Markov Random Field)
b. Markov Logic Network

Essence of GM

1. Capture the variable dependency
2. Reduce parameters
3. Compact representation
4. Factorization of Joint Probability

DATA: Three Random Variables

```r
df<-read.table("abcSampleTable.txt",head=TRUE)
N<-nrow(df)
df
```

```
##      A   B   C
## 1  yes  no yes
## 2   no  no yes
## 3  yes yes  no
## 4  yes yes  no
## 5  yes  no  no
## 6   no  no  no
## 7  yes yes  no
## 8  yes yes  no
## 9   no  no yes
## 10 yes  no  no
## 11  no  no yes
## 12 yes  no  no
## 13  no  no yes
## 14 yes yes  no
## 15 yes yes yes
## 16 yes yes  no
## 17  no  no  no
## 18  no  no yes
## 19 yes yes  no
## 20  no  no  no
```

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(bnlearn)
```

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

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

Random Variables and Probability Table

Answer the following probability queries:

1. $P(A = yes, B = yes)$:

```
## [1] 0.4
```

2. $P(A = yes|B = yes)$

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

3. $P(A = yes)$

```
## [1] 0.6
```

4. $P(B = yes)$

```
## [1] 0.4
```

5. $P(B = yes|A = yes)$

```
## [1] 0.6666667
```

Recollect Bayes' Rule

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

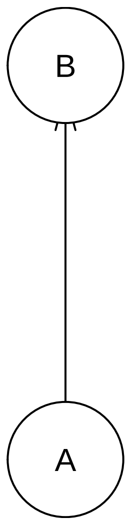
where,

$P(A|B)$ is *posterior*, $P(B|A)$ is *likelihood*, $P(A)$ is *prior* and, $P(B)$ is *evidence*

Joint Probability

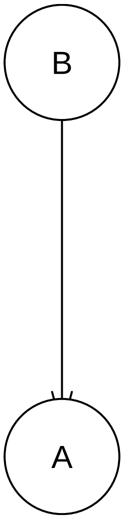
$$P(AB) = P(B|A)P(A)$$
$$P(AB) = P(A|B)P(B)$$

Interpretation-1



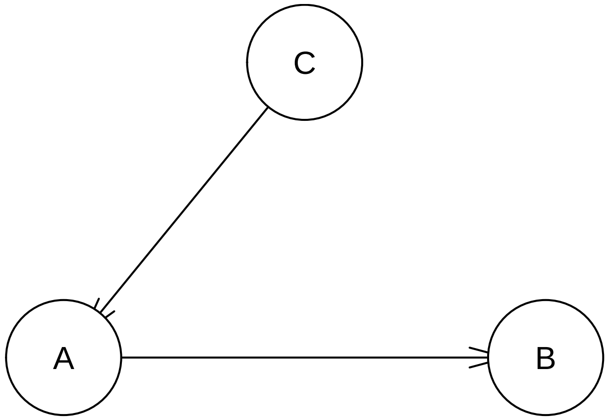

```
##
##   Bayesian network parameters
##
##   Parameters of node A (multinomial distribution)
##
## Conditional probability table:
##   no yes
## 0.4 0.6
##
##   Parameters of node B (multinomial distribution)
##
## Conditional probability table:
##
##      A
## B            no       yes
##   no  1.0000000 0.3333333
##   yes 0.0000000 0.6666667
```

Interpretation-2

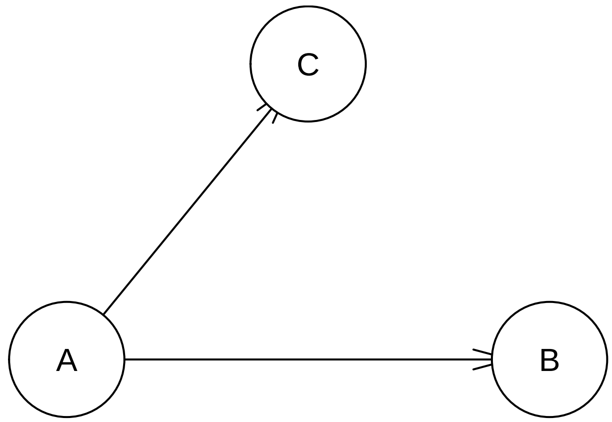



```
##
##   Bayesian network parameters
##
##   Parameters of node A (multinomial distribution)
##
## Conditional probability table:
##
##      B
## A            no       yes
##   no  0.6666667 0.0000000
##   yes 0.3333333 1.0000000
##
##   Parameters of node B (multinomial distribution)
##
## Conditional probability table:
##   no yes
## 0.6 0.4
```

What about a larger set of variables?




```
##
##   Bayesian network parameters
##
##   Parameters of node A (multinomial distribution)
##
## Conditional probability table:
##
##      C
## A            no       yes
##   no  0.2307692 0.7142857
##   yes 0.7692308 0.2857143
##
##   Parameters of node B (multinomial distribution)
##
## Conditional probability table:
##
##      A
## B            no       yes
##   no  1.0000000 0.3333333
##   yes 0.0000000 0.6666667
##
##   Parameters of node C (multinomial distribution)
##
## Conditional probability table:
##    no  yes
## 0.65 0.35
```




```
##
##   Bayesian network parameters
##
##   Parameters of node A (multinomial distribution)
##
## Conditional probability table:
##   no yes
## 0.4 0.6
##
##   Parameters of node B (multinomial distribution)
##
## Conditional probability table:
##
##      A
## B            no       yes
##   no  1.0000000 0.3333333
##   yes 0.0000000 0.6666667
##
##   Parameters of node C (multinomial distribution)
##
## Conditional probability table:
##
##      A
## C            no       yes
##   no  0.3750000 0.8333333
##   yes 0.6250000 0.1666667
```

Let's query these networks

```r
set.seed(0)
ep1 <- cpquery(bayes_bn31.net, event = (A == "yes" & B == "yes"),
evidence = TRUE, n = 1000)
ep2 <- cpquery(bayes_bn32.net, event = (B == "yes"  ),
evidence = (A == "yes"), n = 1000)
ep1
```

```
## [1] 0.394
```

```r
ep2
```

```
## [1] 0.6484245
```

Joint Probability

```
bayes_bn31
```

```
##
##   Bayesian network learned via Score-based methods
##
##   model:
##    [C][A|C][B|A]
##   nodes:                                 3
##   arcs:                                  2
##     undirected arcs:                     0
##     directed arcs:                       2
##   average markov blanket size:           1.33
##   average neighbourhood size:            1.33
##   average branching factor:              0.67
##
##   learning algorithm:                    Hill-Climbing
##   score:                                 Cooper & Herskovits' K2
##   tests used in the learning procedure:  10
##   optimized:                             TRUE
```

```
bayes_bn32
```

```
##
##   Bayesian network learned via Score-based methods
##
##   model:
##    [A][B|A][C|A]
##   nodes:                                 3
##   arcs:                                  2
##     undirected arcs:                     0
##     directed arcs:                       2
##   average markov blanket size:           1.33
##   average neighbourhood size:            1.33
##   average branching factor:              0.67
##
##   learning algorithm:                    Hill-Climbing
##   score:                                 BIC (disc.)
##   penalization coefficient:              1.497866
##   tests used in the learning procedure:  7
##   optimized:                             TRUE
```

$$P(ABC) = P(C)P(A|C)P(B|A)$$
$$P(ABC) = P(A)P(B|A)P(C|A)$$

Aha... you have seen a Graphical Model!

```r
nbr(bayes_bn32,node="A")
```

```
## [1] "B" "C"
```

```r
parents(bayes_bn32,node="A")
```

```
## character(0)
```

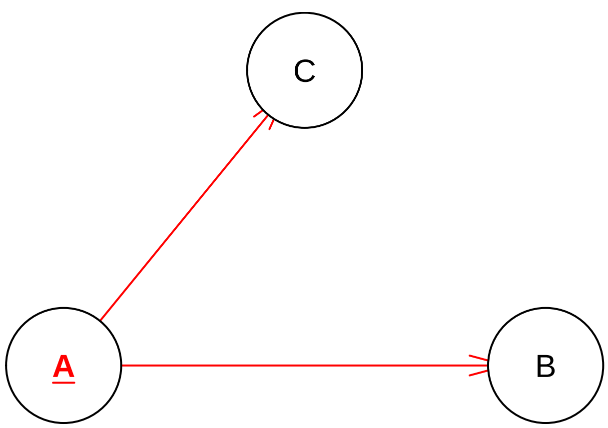
```r
children(bayes_bn32,node="A")
```

```
## [1] "B" "C"
```

```r
mb(bayes_bn32,node="A")
```

```
## [1] "B" "C"
```

```r
plot(bayes_bn32,highlight=list(nodes="A"))
```




```r
arcs(bayes_bn32)
```

```
##      from to
## [1,] "A"  "B"
## [2,] "A"  "C"
```

```r
path(bayes_bn32, from="B", to="C")
```

```
## [1] FALSE
```

```r
dsep(bayes_bn32, "B","C")
```

```
## [1] FALSE
```

```r
dsep(bayes_bn32, "B","C","A")
```

```
## [1] TRUE
```

Can we go back to Sample!

```
set.seed(0)
samples.bn32<-cpdist(bayes_bn32.net, nodes=nodes(bayes_bn32.net), evidence=TRUE, n=20)
samples.bn32
```

```
##      A   B   C
## 1   no  no  no
## 2  yes  no yes
## 3  yes yes  no
## 4  yes  no  no
## 5   no  no yes
## 6  yes yes  no
## 7   no  no yes
## 8   no  no yes
## 9   no  no yes
## 10  no  no  no
## 11 yes  no  no
## 12 yes yes  no
## 13 yes  no  no
## 14  no  no yes
## 15 yes yes  no
## 16  no  no  no
## 17 yes yes  no
## 18  no  no  no
## 19  no  no yes
## 20 yes yes  no
```

How many three node Directed Acyclic Graphs?

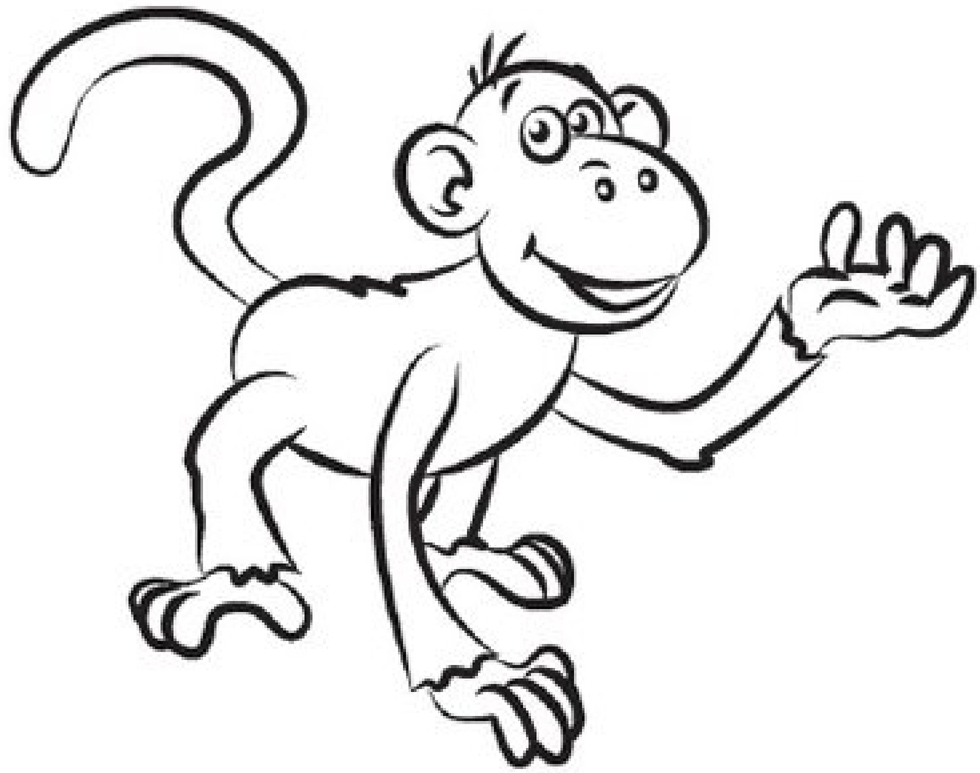


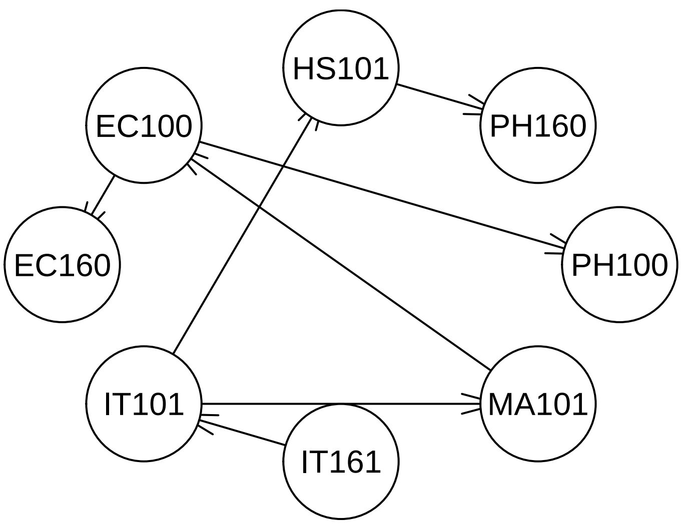
Figure 1: Guesses?

Need a Larger One?

```
course.grades<-read.table("2020_bn_nb_data.txt",head=TRUE)
head(course.grades)
```

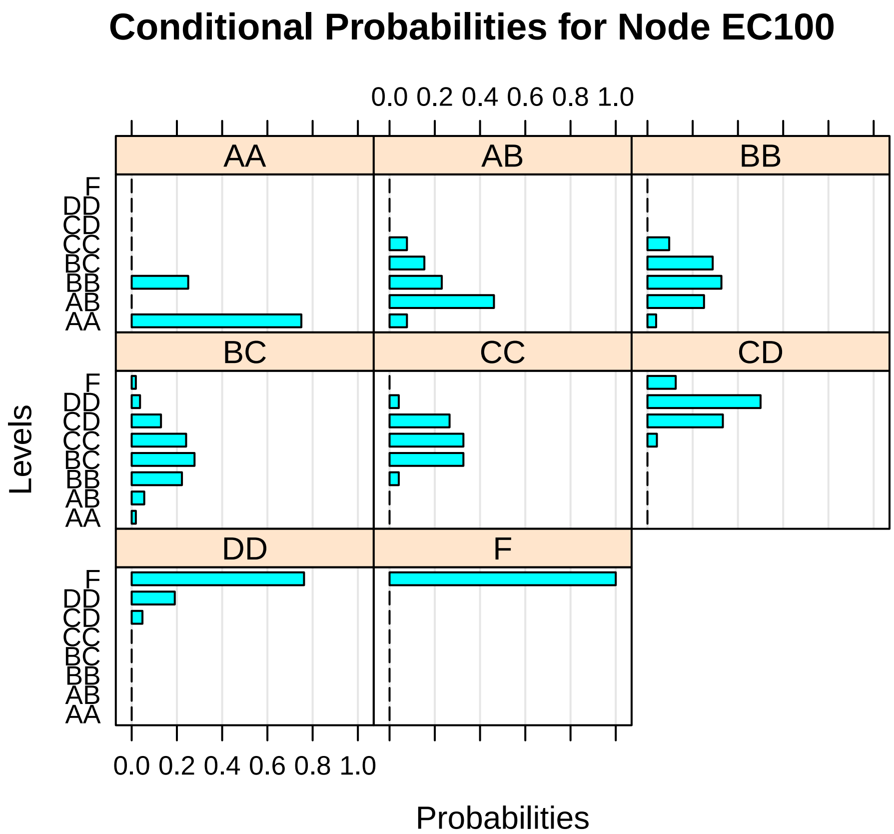
```
##   EC100 EC160 IT101 IT161 MA101 PH100 PH160 HS101 QP
## 1    BC    CC    BB    BC    CC    BC    AA    BB  y
## 2    CC    BC    BB    BB    CC    BC    AB    BB  y
## 3    AB    BB    AB    AB    BB    CC    BC    AB  y
## 4    BC    CC    BB    BB    BB    BB    BC    BB  y
## 5    BC    AB    CD    BC    BC    BC    BC    CD  y
## 6    DD    CC    DD    CD    CD    CC    BC    BC  n
```

```
course.grades.net<-hc(course.grades[,-9],score="k2")
plot(course.grades.net)
```




```
course.grades.net.fit<-bn.fit(course.grades.net, course.grades[,-9])
bn.fit.barchart(course.grades.net.fit$EC100)
```

```
## Loading required namespace: lattice
```



Conditional Probabilities for Node EC100
AA
AB
BB
BC
CC
CD
DD
F
Levels
Probabilities
0.0 0.2 0.4 0.6 0.8 1.0

Naive Bayes Classifier

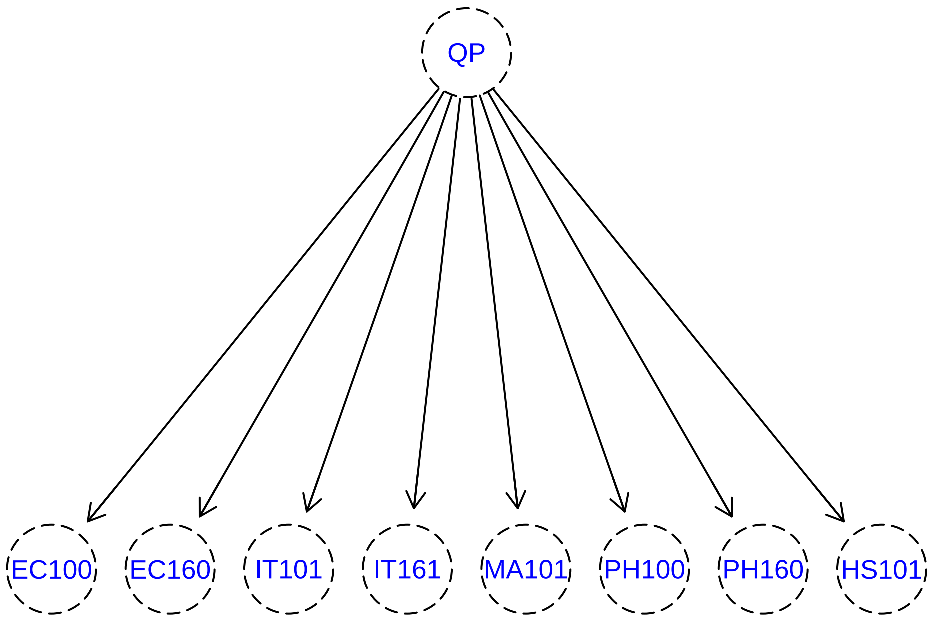
```r
library(bnclassify)
```

```
##
## Attaching package: 'bnclassify'

## The following objects are masked from 'package:bnlearn':
##
##     modelstring, narcs, nparams

## The following object is masked from 'package:dplyr':
##
##     vars
```

```r
course.grades<-read.table("2020_bn_nb_data.txt",head=TRUE)
nb.grades<-nb(class="QP",dataset=course.grades)
plot(nb.grades)
```

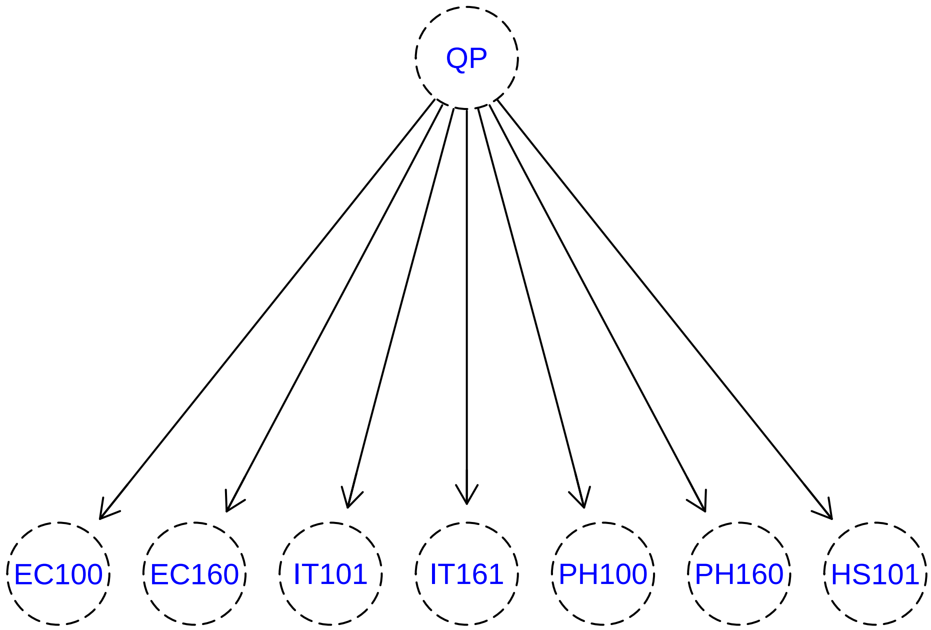



```r
nb.grades<-lp(nb.grades, course.grades, smooth=0)
p<-predict(nb.grades, course.grades)
cm<-table(predicted=p, true=course.grades$QP)
cm
```

```
##          true
## predicted   n   y
##         n  70   1
##         y   2 159
```

Remove MA101

```r
nb.grades<-nb(class="QP",dataset=course.grades[,-5])
plot(nb.grades)
```




```r
nb.grades<-lp(nb.grades, course.grades[,-5], smooth=1)
p<-predict(nb.grades, course.grades[,-5])
cm<-table(predicted=p, true=course.grades$QP)
cm
```

```
##          true
## predicted   n   y
##         n  69   0
##         y   3 160
```

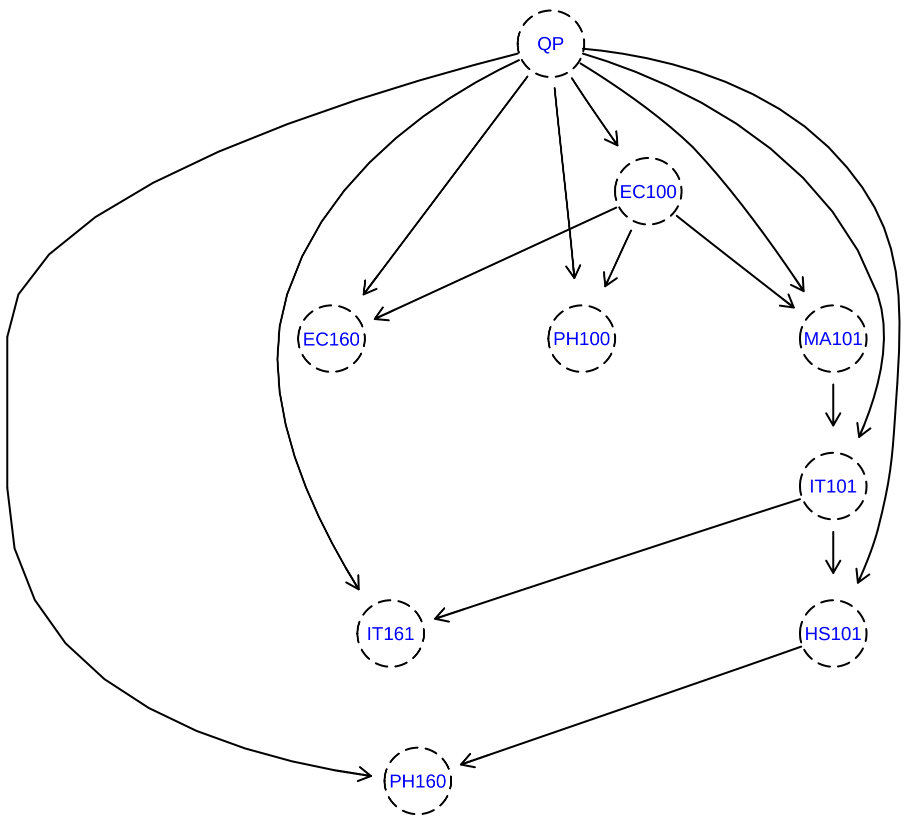
Conditional Independence Check

```r
ci.test("MA101","IT101","QP",course.grades)
```

```
##
##  Mutual Information (disc.)
##
## data:  MA101 ~ IT101 | QP
## mi = 105.55, df = 98, p-value = 0.2832
## alternative hypothesis: true value is greater than 0
```

Something More Interesting

```
tn <- tan_cl("QP", course.grades)
tn <- lp(tn, course.grades, smooth = 1)
plot(tn)
```




```
tn <- lp(tn, course.grades, smooth = 1)
p <- predict(tn, course.grades)
cm1<-table(predicted=p, true=course.grades$QP)
cm1
```

```
##          true
## predicted   n   y
##         n  71   0
##         y   1 160
```

References

1. Bayesian Network without Tears by Eugene Charniak
2. Bayesian Networks with R by Bojan Mihaljevic
3. bnstruct: an R package for Bayesian Network Structure Learning with missing data by Francesco Sambo and Alberto Franzin
4. Introduction to Artificial Intelligence by Stuart Russell and Peter Norvig