

Graphical Models with R

2nd talk: Graphical Models for Categorical Data

Dhafer Malouche

essai.academia.edu/DhaferMalouche

Table of contents

1. Introduction
2. Graphical Models are log-linear models
3. Estimating parameters
4. Hypothesis testing
5. Model selection

Introduction

Notations

- V is a finite set,
- (r. vector) $X_V = (X_v, v \in V) \sim \mathbf{p}$ (joint distribution)
 - $\forall v \in V$ and $\mathcal{P}(V) = \{A, A \subseteq V\} \cup \{\emptyset\}$ $X_v(\Omega) = \mathcal{I}_v = \{1, \dots, l_v\}$,
 $l_v \in \mathbb{N}^*$.
 - $\mathcal{I} = X_V(\Omega) = \bigotimes_{v \in V} \mathcal{I}_v = \bigotimes_{v \in V} \{1, \dots, l_v\}$
 - $\forall i \in \mathcal{I}, i = (i_v, v \in V)$
- (Marginalization) If $A \subseteq V$, $X_A = (X_v, v \in A)$ and $\mathcal{I}_A = \bigotimes_{v \in A} \mathcal{I}_v$

Notations

- (Data) (X_V^1, \dots, X_V^n) is an n -sample $\sim \mathbf{p}$ (same as X_V)
- (Contingency table)
 - $\mathcal{N} = (n(i), i \in \mathcal{I})$ such that

$$n(i) = \sum_{k=1}^n \mathbb{1}_{\{X_V^k=i\}}$$

- $X_A = (X_v, v \in V), \forall i_A \in \mathcal{I}_A$

$$n(i_A) = \sum_{k=1}^n \mathbb{1}_{\{X_A^k=i\}}$$

Question

- How can we provide an efficient visualization of the relationships between the variables $X_V = (X_v, v \in V)$?
- How can we provide a “good” estimation of the joint distribution, conditional distributions in X_V ?

Suggested solution

Solution

- Using an undirected graph
- Nodes=Variables of X_V
- Edges and separation statements = Conditional independences.

Graphical Models are log-linear models

Example with 3 variables

- $V = \{a, b, c\}$ and $X_a = a, X_b = b, X_c = c,$
- $i = (j, k, l) \in \mathcal{I} = \{1, \dots, l_a\} \times \{1, \dots, l_b\} \times \{1, \dots, l_c\}$
- $\{p_{jkl}, (j, k, l) \in \mathcal{I}\}$ s.t. $\sum_{j,k,l} p_{jkl} = 1$
- saturated *log-linear* model

$$\log(p_{jkl}) = u_{\emptyset} + u_j^a + u_k^b + u_l^c + u_{jk}^{ab} + u_{jl}^{ac} + u_{kl}^{bc} + u_{jkl}^{abc}$$

Example with 3 variables

$$\begin{aligned} a \perp\!\!\!\perp b \mid c &\iff p_{jkl} = p_{jk|l} \times p_{++l} \\ &= p_{j|l} \times p_{k|l} \times p_{++l} \\ &= h(j, l) \times g(k, l) \\ &\iff \log(p_{jkl}) = \log h(j, l) + \log g(k, l) \end{aligned}$$

Example with 3 variables

$$\begin{aligned} a \perp\!\!\!\perp b \mid c &\iff p_{jkl} &= p_{jk|l} \times p_{++l} \\ & &= p_{j|l} \times p_{k|l} \times p_{++l} \\ & &= h(j, l) \times g(k, l) \end{aligned}$$

$$\iff \log(p_{jkl}) = \log h(j, l) + \log g(k, l)$$

$$m : a \perp\!\!\!\perp b \mid c \iff \log(p_{jkl}) = u_\emptyset + u_j^a + u_k^b + u_l^c + u_{jl}^{ac} + u_{kl}^{bc}$$

Hierarchical log-linear models

- **Log-linear** Models $m = (m, \mathcal{P})$ where $\mathcal{P} \subseteq \mathcal{P}(V)$

$$\forall i \in \mathcal{I}, \log p_i = \sum_{A \in \mathcal{P}} u_{i_A}^A \quad (1)$$

where $i_A = (i_v, v \in A)$ and $u_{i_A}^A$ are unknown parameters.

- **Hierarchical** Log-linear Models $m = (m, \mathcal{P})$

$$\forall i \in \mathcal{I}, \text{ and } A \in \mathcal{P} \text{ If } u_{i_A}^A = 0 \Rightarrow \forall B \supseteq A \text{ and } B \in \mathcal{P} \ u_{i_B}^B = 0$$

- $\mathcal{C}(m) = \{A \in \mathcal{P} \text{ and } A \text{ is maximal}\}$ is called the set of *generators* of m .

Graphical Model

- $\mathcal{G} = (V, E)$, $X_V = (X_v, v \in V) \sim \mathbf{p}$ and $m = (m, P)$ is a hierarchical log linear model.
- m is called a **graphical** model if

$$u \not\sim v \iff \forall B \supseteq \{u, v\}, u_{i_B}^B = 0$$

- $m = (m, \mathcal{P}_G)$ where \mathcal{P}_G is the set of complete subsets of \mathcal{G} .
- We can read from \mathcal{G} :

$$u \not\sim v \iff u \perp\!\!\!\perp v \mid V \setminus \{u, v\}$$

and let $(A, B, S) \in \mathcal{P}(V)^3$

$$\text{If } S \text{ separates } A \text{ and } B \text{ in } \mathcal{G} \Rightarrow A \perp\!\!\!\perp B \mid S$$

- The set of generators of m are the set of cliques of $\mathcal{C}(m) = \mathcal{C}(\mathcal{G})$.

Example

- Data `lizard`; 409 lizards: species (S), perch diameter (D) and perch height (H).

```
> library(gRim)
> data("lizardRAW")
> head(lizardRAW,3)
  diam height species
1   >4   >4.75   dist
2   >4   >4.75   dist
3  <=4  <=4.75  anoli
> lizard=xtabs(~diam+height+species,data=lizardRAW)
> lizard
, , species = anoli

      height
diam  <=4.75 >4.75
  <=4      86    32
  >4       35    11

, , species = dist

      height
diam  <=4.75 >4.75
  <=4      73    61
  >4       70    41
```

Example

- Model 1:

$$\log(p_{jkl}) = u_{\emptyset} + u_j^a + u_k^b + u_l^c + u_{jk}^{ab} + u_{jl}^{ac} + u_{kl}^{bc}$$

```
> m1<- dmod(~species*height+species*diam+height*diam, data=lizard)
> summary(m1)
is graphical=FALSE; is decomposable=FALSE
generators (glist):
  : "species" "height"
  : "species" "diam"
  : "height" "diam"
> m1$glist
[[1]]
[1] "species" "height"

[[2]]
[1] "species" "diam"

[[3]]
[1] "height" "diam"
```

Example

- Model 2:

$$\log(p_{jkl}) = u_{\emptyset} + u_j^a + u_k^b + u_l^c + u_{jl}^{ac} + u_{kl}^{bc}$$

```
> m2<- dmod(~species*height+species*diam, data=lizard)
> summary(m2)
is graphical=TRUE; is decomposable=TRUE
generators (glist):
  : "species" "height"
  : "species" "diam"
> m2$glist
[[1]]
[1] "species" "height"

[[2]]
[1] "species" "diam"
```


Example

- Model 3: (saturated or complete model)

$$\log(p_{jkl}) = u_{\emptyset} + u_j^a + u_k^b + u_l^c + u_{jk}^{ab} + u_{jl}^{ac} + u_{kl}^{bc} + u_{jkl}^{abc}$$

```
> m3<- dmod(~species*height*diam, data=lizard)
> summary(m3)
is graphical=TRUE; is decomposable=TRUE
generators (glist):
  : "species" "height" "diam"
> m3$glist
[[1]]
[1] "species" "height" "diam"
```

Estimating parameters

Estimating the parameters

- Since $\forall k = 1, \dots, n$ and $\forall i \in \mathcal{I}$

$$\mathbb{P}(X_V^k = i) = p(i) \in [0, 1]$$

- Joint distribution of $\{n(i), i \in \mathcal{I}\}$ (Multinomial sampling):

$$\mathbb{P}(n(i), i \in \mathcal{I}) = \frac{N!}{\prod_{i \in \mathcal{I}} n(i)!} \prod_{i \in \mathcal{I}} p(i)^{n(i)}$$

- Likelihood:

$$L(p) \propto \prod_{i \in \mathcal{I}} p(i)^{n(i)}$$

Estimating the parameters

- With no constraints, the MLE of $p(i)$:

$$\hat{p}(i) = \frac{n(i)}{N}, \quad \forall i \in \mathcal{I}$$

and

$$\mathbb{E}(n(i)) = Np(i) \quad \text{and fitted by} \quad \hat{m}(i) = N\hat{p}(i)$$

- If \mathcal{G} is decomposable, $\{C_1, \dots, C_k\}$ set of cliques perfectly ordered (RIP) with Separators $\{S_1, \dots, S_k\}$, The ML is:

$$\hat{m}(i) = \frac{\prod_{j=1, \dots, k} n(C_j)}{\prod_{j=1, \dots, k} n(S_j)}$$

and

$$\hat{p}(i) = \frac{\hat{m}(i)}{N}$$

Estimating the parameters

- If \mathcal{G} is not decomposable, IPS-algorithm *Iterative proportional scaling*.
- $\mathcal{P}_{\mathcal{G}} = \{a_1, \dots, a_k\}$ the set of generators in the Log-linear models (m, \mathcal{G}) .
- **Algorithm:** $k = 0$, $[m(i)]_0 \leftarrow 1$, and at the Iteration k .

$$[m(i)]_{k+1} \leftarrow [m(i)]_k \frac{n(i_{a_k})}{[m(i_{a_k})]}, \quad \forall i \in \mathcal{I}$$

- Iteration continues until convergence when $m(i_{a_k}) = n(i_{a_k})$.

Example (IPS)

- Goal:

	30	35	65
	25	10	35
	55	45	

Example (IPS)

- Goal:

	30	35	65
	25	10	35
	55	45	

- IPS

$k = 0$

	1	1	2
	1	2	2
	2	2	

Example (IPS)

- Goal:

	30	35	65
	25	10	35

- IPS $\frac{65}{2} \times 1 = 32.5$
k and $\frac{35}{2} \times 1 = 17.5$

Example (IPS)

- Goal:

	30	35	65
	25	10	35
	55	45	

- IPS

$k = 0$

	1	1	2
	1	2	2
	2	2	

$\rightarrow k = 1$

	32.5	32.5	65
	17.5	17.5	35
	50	50	

Example (IPS)

- Goal:

	30	35	65
	25	10	35

- IPS $\frac{55}{50} \times 32.5 = 35.75$ $\frac{45}{50} \times 32.5 = 29.25$

k

$$\frac{55}{50} \times 17.5 = 19.25 \quad \frac{45}{50} \times 17.5 = 15.75$$

5	65
5	35

Example (IPS)

- Goal:

	30	35	65
	25	10	35
	55	45	

- IPS

$k = 0$

	1	1	2
	1	2	2
	2	2	

$\rightarrow k = 1$

	32.5	32.5	65
	17.5	17.5	35
	50	50	

$\rightarrow k = 2$

	35.75	29.25	65
	19.25	15.75	35
	55	45	

Hypothesis testing

Hypothesis testing

- $m = (m, \mathcal{G})$ is a Graphical model, and the *maximized log-likelihood* is

$$\hat{l}(m) = \sum_{i \in \mathcal{I}} n(i) \log \hat{p}(i)$$

- $\mathcal{G}_s = (V, V \times V)$ is the complete graph, and $\mathcal{G}_i = (V, \emptyset)$ is the independence graph,
- Let $m_s = (m_s, \mathcal{G}_s)$ and $m_i = (m_i, \mathcal{G}_i)$. For any $m = (m, \mathcal{G})$
- *Deviance*

$$D = 2(\hat{l}(m_s) - \hat{l}(m)) \sim \chi^2(k) \quad (n \rightarrow \infty)$$

- *iDeviance*

$$iD = 2(\hat{l}(m) - \hat{l}(m_i)) \sim \chi^2(l) \quad (n \rightarrow \infty)$$

k and l is the difference in the number of free parameters btw the two models (dimensions).

Example 1

```
> m<- dmod(~species*height+species*diam, data=lizard)
> m
Model: A dModel with 3 variables
graphical : TRUE decomposable : TRUE
-2logL : 1604.43 mdim : 5 aic : 1614.43
ideviance : 23.01 idf : 2 bic : 1634.49
deviance : 2.03 df : 2
> ms <- dmod(~.^., data=lizard)
> ms
Model: A dModel with 3 variables
graphical : TRUE decomposable : TRUE
-2logL : 1602.40 mdim : 7 aic : 1616.40
ideviance : 25.04 idf : 4 bic : 1644.50
deviance : 0.00 df : 0
> mi<- dmod(~.^1, data=lizard)
> mi
Model: A dModel with 3 variables
graphical : TRUE decomposable : TRUE
-2logL : 1627.44 mdim : 3 aic : 1633.44
ideviance : 0.00 idf : 0 bic : 1645.48
deviance : 25.04 df : 4
```

Example 2

```
> data(reinis)
> dim(reinis)
[1] 2 2 2 2 2 2
> names(dimnames(reinis))
[1] "smoke" "mental" "phys" "systol" "protein" "family"
> reinis[1,1,1,,]
, , family = y

      protein
systol y  n
      y 44 23
      n 35 24

, , family = n

      protein
systol y  n
      y  5  7
      n  4  4
```

Example 2

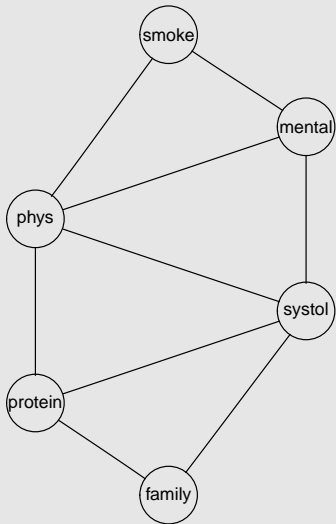
```
> m<- dmod(~smoke*mental*phys+mental*phys*systol+
+          phys*systol*protein+systol*protein*family, data=reinis)
> m
```

Model: A dModel with 6 variables

graphical :	TRUE	decomposable :	TRUE		
-2logL :	13372.25	mdim :	19	aic :	13410.25
ideviance :	757.97	idf :	13	bic :	13515.09
deviance :	85.98	df :	44		

Example 2

```
> m<- dmod(~sm
+
> m
Model: A dMode
graphical :
-2logL :
ideviance :
deviance :
```



data=reinis)

.25
.09

Example 2

Saturated and Independent models.

```
> ms <- dmod(~.^., data=reinis)
```

```
> ms
```

```
Model: A dModel with 6 variables
```

```
graphical : TRUE decomposable : TRUE  
-2logL    :      13286.27 mdim :    63 aic :      13412.27  
ideviance :      843.96 idf  :    57 bic :      13759.91  
deviance  :           0.00 df   :     0
```

```
Notice: Table is sparse
```

```
Asymptotic chi2 distribution may be questionable.
```

```
Degrees of freedom can not be trusted.
```

```
Model dimension adjusted for sparsity : 62
```

```
> mi <- dmod(~.^1, data=reinis)
```

```
> mi
```

```
Model: A dModel with 6 variables
```

```
graphical : TRUE decomposable : TRUE  
-2logL    :      14130.22 mdim :     6 aic :      14142.22  
ideviance :       -0.00 idf  :     0 bic :      14175.33  
deviance  :      843.96 df   :    57
```

Testing conditional independence

$$H_0 : u \perp\!\!\!\perp v \mid V \setminus \{u, v\}$$

$$H_1 : u \not\perp\!\!\!\perp v \mid V \setminus \{u, v\}$$

Proposition

If \mathcal{G} is decomposable, then

$$u \perp\!\!\!\perp v \mid V \setminus \{u, v\} \iff u \perp\!\!\!\perp v \mid C \setminus \{u, v\}$$

where the C is the clique containing uv .

Example (idea for the proof)

uv is in $C = \{u, v, w\}$, Assume

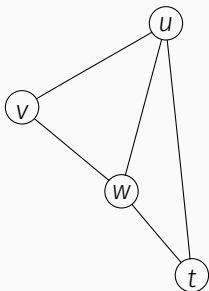
$u \perp\!\!\!\perp v \mid wt$. Then

$u \not\perp\!\!\!\perp v \Rightarrow w$ separates v and t

$\Rightarrow w \perp\!\!\!\perp v \mid t$

$$\begin{aligned} f(u, v, w) &= \int f(u, v, w, t) dt \\ &= \int f(v, w, t) f(u, w, t) dt \\ &= f(v, w) \int f(w, t) f(u, w, t) dt \\ &= f(v, w) f(u, w) \end{aligned}$$

$\Rightarrow u \perp\!\!\!\perp v \mid w$

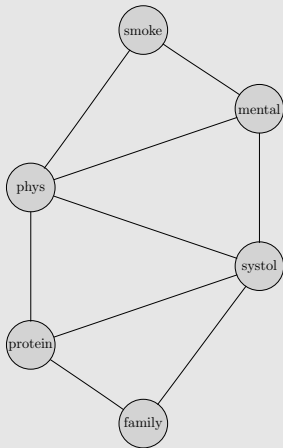


Example, \mathcal{G} decomposable

```
> m1<- dmod(~smoke*mental*phys+mental*phys*systol+
+          phys*systol*protein+systol*protein*family, data=reinis)
> m1
Model: A dModel with 6 variables
graphical : TRUE decomposable : TRUE
-2logL    :      13372.25 mdim :    19 aic :      13410.25
ideviance :       757.97 idf  :    13 bic :      13515.09
deviance  :         85.98 df   :     44
```

Example, \mathcal{G} decomposable

```
> m1<- dmod(~  
+  
> m1  
Model: A dMod  
graphical :  
-2logL :  
deviance :  
deviance :
```

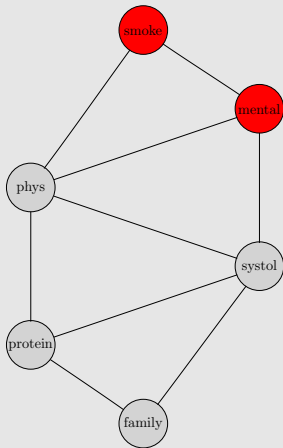


```
ily, data=reinis)
```

```
3410.25  
3515.09
```

Example, \mathcal{G} decomposable

```
> m1<- dmod(~  
+  
> m1  
Model: A dMod  
graphical :  
-2logL :  
ideviance :  
deviance :
```



```
ily, data=reinis)
```

```
3410.25  
3515.09
```


Example, \mathcal{G} decomposable

```
> testdelete(m1,c("smoke","mental"))
dev:      5.988 df:  2 p.value: 0.05010 AIC(k=2.0):      2.0 edge: smoke:mental
host:  phys mental smoke
Notice: Test performed in saturated marginal model
```

The Hypothesis testing on the clique only.

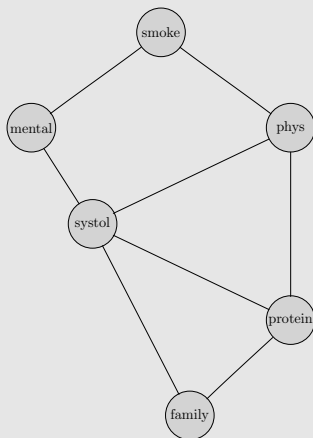
```
> m2<- dmod(~smoke*mental*phys, data=reinis)
> m2
Model: A dModel with 3 variables
graphical :  TRUE decomposable :  TRUE
-2logL    :      6889.11 mdim :      7 aic :      6903.11
ideviance :      719.44 idf  :      4 bic :      6941.74
deviance  :           0.00 df   :      0
> testdelete(m2,c("smoke","mental"))
dev:      5.988 df:  2 p.value: 0.05010 AIC(k=2.0):      2.0 edge: smoke:mental
host:  smoke mental phys
Notice: Test performed in saturated marginal model
```

Example, \mathcal{G} non-decomposable

```
> m<- dmod(~smoke*mental+ smoke*phys+phys*systol+mental*systol+
+          phys*systol*protein+systol*protein*family, data=reinis)
> m
Model: A dModel with 6 variables
graphical : TRUE decomposable : FALSE
-2logL    :      14055.27 mdim :    16 aic :      14087.27
ideviance :         74.95 idf  :    10 bic :      14175.56
deviance  :         769.01 df   :     47
Notice: Table is sparse
Asymptotic chi2 distribution may be questionable.
```

Example, \mathcal{G} non-decomposable

```
> m<- dmod(~sm
+
> m
Model: A dMode
graphical :
-2logL :
deviance :
deviance :
Notice: Table
Asymptotic c
```

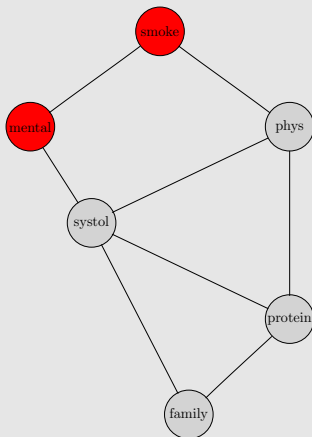


```
*systol+
data=reinis)
```

```
.27
.56
```

Example, \mathcal{G} non-decomposable

```
> m<- dmod(~sm
+
> m
Model: A dMode
graphical :
-2logL :
ideviance :
deviance :
Notice: Table
Asymptotic c
```



```
*systol+
data=reinis)
```

```
.27
.56
```

Example, \mathcal{G} non-decomposable

```
> testdelete(m,c("smoke","mental"))  
dev: 19.335 df: 1 p.value: 0.00001 AIC(k=2.0): 17.3 edge: smoke:mental  
Notice: Test performed by comparing likelihood ratios
```

It's a log-ratio between two graphical models

Example, \mathcal{G} non-decomposable

```
> m1<- dmod(~smoke*phys+phys*systol+mental*systol+
+          phys*systol*protein+systol*protein*family, data=reinis)
```

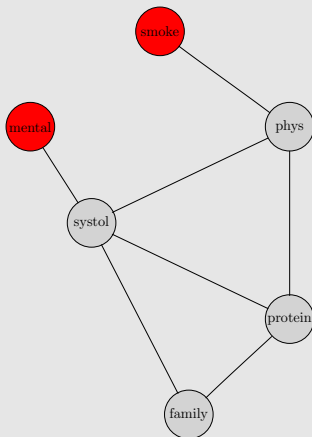
```
> m1
```

Model: A dModel with 6 variables

graphical :	TRUE	decomposable :	TRUE		
-2logL :	14064.94	mdim :	15	aic :	14094.94
ideviance :	65.28	idf :	9	bic :	14177.71
deviance :	778.67	df :	48		

Example, \mathcal{G} non-decomposable

```
> m1<- dmod(~s
+
> m1
Model: A dMode
graphical :
-2logL :
deviance :
deviance :
```



data=reinis)

.94
.71

Example, \mathcal{G} non-decomposable

```
> testdelete(m,c("smoke","mental"))
dev: 19.335 df: 1 p.value: 0.00001 AIC(k=2.0): 17.3 edge: smoke:mental
Notice: Test performed by comparing likelihood ratios
```

With the edge

```
> logLik(m)
'log Lik.' -7027.637 (df=47)
```

Without the edge

```
> logLik(m1)
'log Lik.' -7032.471 (df=48)
```

The log-ratio of the likelihood

```
> 4*(as.vector(logLik(m))-as.vector(logLik(m1)))
[1] 19.33531
```


Testing $u \perp\!\!\!\perp v \mid W, W \subset V$

$$u \perp\!\!\!\perp v \mid W$$

- `ciTest_table`: test the removing the edge uv from the saturated model with variables $\{u, v\} \cup W$.
- Testing `smoke` $\perp\!\!\!\perp$ `mental` given $\{\text{systol}, \text{family}\}$.

```
> cit<-ciTest_table(reinis,c("smoke","mental","systol","family"))
> cit
Testing smoke _|_ mental | systol family
Statistic (DEV): 12.582 df: 4 p-value: 0.0135 method: CHISQ
> names(cit)
[1] "statistic" "p.value"    "df"          "statname"   "method"     "adjust.df"
[7] "varNames"  "slice"
> cit$slice
  statistic    p.value df systol family
1 7.1727226 0.007402035 1      y      y
2 0.4125993 0.520653168 1      n      y
3 3.0934804 0.078606514 1      y      n
4 1.9036216 0.167673506 1      n      n
```

Model selection

Model Selection

- $2^{|V|(|V|-1)/2}$ possible graphical models with $|V|$ nodes.
- Select the model with the minimum value when it's based on the minimization of one selection criteria of penalized likelihood : *deviance, AIC, BIC*..
- penalized likelihood = $-2 \log L(m) + kp(m)$
 - $L(m)$ is the likelihood of m ,
 - k is the penalty parameter,
 - $p(m)$ is the number of free parameters.
 - If $k = 2 \rightarrow$ AIC, $k = \log n$ (sample size) \rightarrow BIC

Stepwise algorithm

Stepwise backward, with deviance

1. Start by the saturated model m_s , $k = 0$
2. $m_k = m_s$

3. Test the presence of every edge,

$$\forall u, v, \text{ test } H_0^{uv}(m_k) u \perp\!\!\!\perp v \mid V \setminus \{u, v\}$$

4. Let

$$p_{k,max} = \max_{u,v} p_{k,u,v}$$

where $p_{k,u,v}$ is the p-value of $H_0^{uv}(m_k)$

5. If $p_{k,max} < \alpha$ then Stop $\hat{m} = m_k$, else

$$(uv)_d = \operatorname{argmax}_{u,v} p_{k,u,v}$$

and let $m_s = m_k \setminus (uv)_d$, $k = k + 1$ and go to 2.

Stepwise backward, with AIC

1. Start by the saturated model m_s , $k = 0$

2. $m_k = m_s$

3. Fit $m_{uv} = m_k \setminus \{u, v\}$ for all $u, v \in V$

4. Let

$$\Delta(\text{AIC}_{uv}) = \text{AIC}_{m_{uv}} - \text{AIC}_{m_k}$$

5. If $\Delta(\text{AIC}) > 0$, stop and $\hat{m} = m_k$

6. Else, let

$$(uv)_d = \operatorname{argmin}_{u,v} \Delta(\text{AIC}_{uv})$$

7. let $m_s = m_k \setminus (uv)_d$, $k = k + 1$ and go to 2.

Stepwise forward, with deviance

1. Start by the independent model $m_\emptyset, k = 0$
2. for every $u, v \in V$ let $m_k(uv) = m_\emptyset \cup (uv)$

3. Test the presence of every edge in $m_k(uv)$,

$$\forall u, v, \text{ test } H_0(m_k(uv)) \ u \perp\!\!\!\perp v \mid V \setminus \{u, v\}$$

4. Let

$$p_{k,min} = \min_{u,v} p_{k,u,v}$$

where $p_{k,u,v}$ is the p-value of $H_0(m_k(uv))$

5. If $p_{k,min} \geq \alpha$ then Stop $\hat{m} = m_k$,
6. Else

$$(uv)_+ = \operatorname{argmin}_{u,v} p_{k,u,v}$$

and let $m_\emptyset = m_k \cup (uv)_+, k = k + 1$ and go to 2.

Stepwise backward, with AIC

1. Start by the independent model $m_\emptyset, k = 0$
2. $m_k = m_\emptyset$

3. Fit $m_{uv} = m_k \cup \{u, v\}$ for all $u, v \in V$

4. Let

$$\Delta(\text{AIC}_{uv}) = \text{AIC}_{m_{uv}} - \text{AIC}_{m_k}$$

5. If $\Delta(\text{AIC}) \leq 0$, stop and $\hat{m} = m_k$

6. Else, let

$$(uv)_d = \operatorname{argmax}_{u,v} \Delta(\text{AIC}_{uv})$$

7. let $m_s = m_k \cup (uv)_d, k = k + 1$ and go to 2.

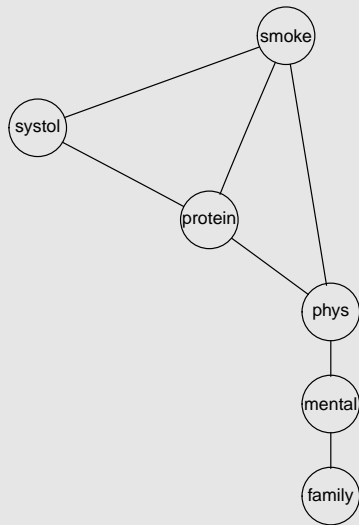
With 'R'

```
> ms<-dmod(~.^., data=reinis)
> mhat<-stepwise(ms,criterion = "test",details=1)
STEPWISE:
criterion: test
direction: backward
type      : decomposable
search    : all
steps     : 1000
. BACKWARD: type=decomposable search=all, criterion=test, alpha=0.05
. Initial model: is graphical=TRUE is decomposable=TRUE
p.value    0.7283 Edge deleted: mental systol
p.value    0.5207 Edge deleted: phys systol
p.value    0.1819 Edge deleted: mental protein
p.value    0.1733 Edge deleted: systol family
p.value    0.3335 Edge deleted: family protein
p.value    0.1349 Edge deleted: phys family
p.value    0.2647 Edge deleted: smoke family
p.value    0.0501 Edge deleted: smoke mental
```

```
> mhat
Model: A dModel with 6 variables
graphical : TRUE decomposable : TRUE
-2logL    : 13343.61 mdim : 15 aic : 13373.61
ideviance : 786.61 idf : 9 bic : 13456.38
deviance  : 57.35 df : 48
```

With R

```
> mhat  
Model: A dMode  
graphical :  
-2logL :  
ideviance :  
deviance :
```



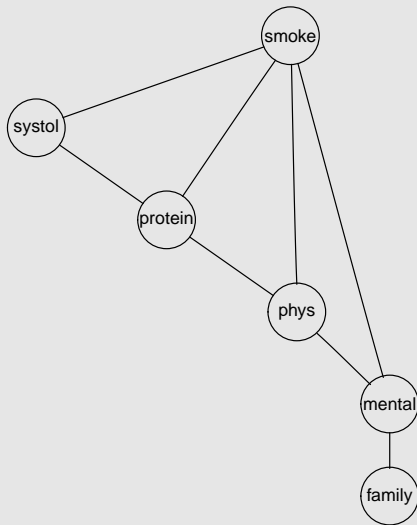
.61
.38

```
> ms<-dmod(~.^., data=reinis)
> mhat<-stepwise(ms,criterion = "aic",details=1)
STEPWISE:
  criterion: aic ( k = 2 )
  direction: backward
  type      : decomposable
  search    : all
  steps     : 1000
. BACKWARD: type=decomposable search=all, criterion=aic(2.00), alpha=0.00
. Initial model: is graphical=TRUE is decomposable=TRUE
  change.AIC  -19.7744 Edge deleted: mental systol
  change.AIC   -8.8511 Edge deleted: phys systol
  change.AIC   -4.6363 Edge deleted: mental protein
  change.AIC   -1.6324 Edge deleted: systol family
  change.AIC   -3.4233 Edge deleted: family protein
  change.AIC   -0.9819 Edge deleted: phys family
  change.AIC   -1.3419 Edge deleted: smoke family

> mhat
Model: A dModel with 6 variables
graphical : TRUE decomposable : TRUE
-2logL    :      13337.63 mdim :    17 aic :      13371.63
ideviance :      792.60 idf  :    11 bic :      13465.43
deviance  :      51.36 df   :     46
```

With R

```
> ms<-dmod(~.1
> mhat<-stepwi
STEPWISE:
criterion: ai
direction: ba
type      : de
search    : al
steps     : 10
. BACKWARD: ty
. Initial mode
change.AIC
change.AIC
change.AIC
change.AIC
change.AIC
change.AIC
change.AIC
change.AIC
> mhat
Model: A dMode
graphical :
-2logL    :
deviance  :
deviance  :
```



(2.00), alpha=0.00

.63
.43

Chow-Liu Algorithm

Chow–Liu Algorithm

- Chow and Liu (1968) showed that finding the maximum likelihood tree can be formulated as finding a maximum weight spanning tree
- - A forest is an acyclic undirected graph, A forest with n vertices and k components, has $n - k$ edges.
 - A tree is a connected acyclic undirected graph
 - forest = union of trees.
- spanning tree T of an undirected graph is a subgraph of \mathcal{G} that is a tree which includes all of the vertices of G , with minimum possible number of edges.

Maximum weight spanning tree

- Algorithm
 1. Starts with the null graph, $T = \emptyset$
 2. Successively selects edges e_1, \dots, e_r
 3. If edges e_1, \dots, e_k have been selected, the algorithm selects an edge e selected s.t
 - (a) $e \in \{e_1, \dots, e_k\}$ and $\{e_1, \dots, e_k, e\}$ is a forest and
 - (b) e has has maximum weight among all edges satisfying (a).
- Modifying the weights appropriately to find the minimal AIC or BIC forest

Quick check of a Survey data: Afrobarometer

- Source of data <http://www.afrobarometer.org>
- Wage 1: 1999-2000, 12 African Countries, 144 columns and 21531 observations.
- Keep only categorical variables.

Cleaning data

```
> X=afro_data[[1]]
> dim(X)
[1] 21531 144
> z=c()
> for(j in 1:ncol(X)){
+   z=c(z,is.factor(X[,j]))
+ }
> X=X[,z==T]
> dim(X)
[1] 21531 132
> x=apply(X,2,function(x)length(unique(x)))
> summary(x)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 4.00  6.75   7.00  164.20  8.00 19940.00
```

Cleaning the data

```
> j=which(x>1000)
> X=X[,-j]
> dim(X)
[1] 21531 131
> x=apply(X,2,function(x)length(unique(x)))
> summary(x)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 4.00   6.50   7.00  13.27   8.00  212.00
```

Estimating the tree model

```
> library(gRapHD)
> tree1<-minForest(X)
> tree1
gRapHD object
Number of edges      = 97
Number of vertices   = 131
Model                = discrete
Statistic (minForest) = BIC
Statistic (stepw)    =
Statistic (user def.) =
Edges (minForest)    = 1...97
Edges (stepw)        = 0...0
Edges (user def.)    = 1...97
```

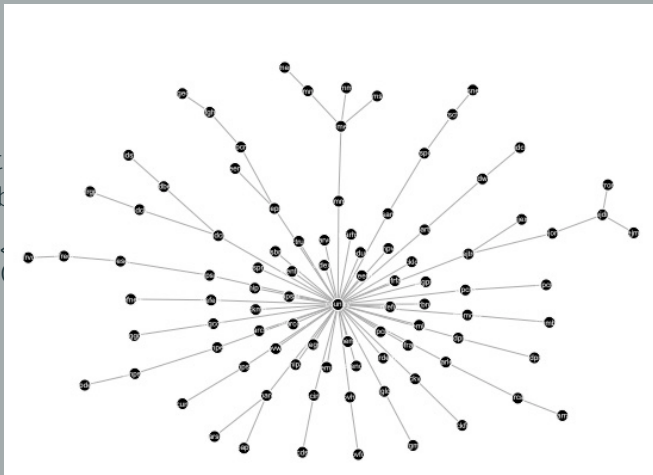
Extract the nodes of `tree1` whose path length ≤ 4 , and then display the subgraph. these nodes.

```
> nby <- neighbourhood(tree1, orig=1, rad=4)$v[,1]
> plot(tree1, vert=nby, numIter=1000)
```

Estimating the tree model

Extract
the sub

```
> nby  
> plot
```



display

Neighbor of the variable country

```
> adj(as(tree1,"graphNEL"),"country")
$country
 [1] "identity" "rejtrd" "dmpext" "dmpvot" "dmpsay" "defele"
 [7] "sparet" "sapsat" "educ" "unemp" "urbrur" "povwat"
[13] "povhth" "aidexp" "demmaj" "marhse" "marcom" "sctrust"
[19] "scint" "sckloc" "sckmp" "sckvp" "gidprd" "gidchl"
[25] "pfeprc" "pfenow" "pfefut" "pfeerd" "pfegrp" "pfrfai"
[31] "pfrall" "pfpsaf" "pfpcr3" "pfpcr5" "trspre" "trsarm"
[37] "trsbrd" "pfgpre" "pfgloc" "mip1" "mip2" "legcon"
[43] "memrel" "memdev" "memlab" "parvot" "parlet" "pardem"
[49] "parctg" "party"
```