Data Mining 2018
Graphical Models for Discrete Data
Part 1: Undirected Graphs (2)

Ad Feelders

Universiteit Utrecht

September 27, 2018

Overview of Coming Two Lectures

- (Conditional) Independence
- Graphical Representation
- Log-linear Models
  - Hierarchical
  - Graphical
  - Decomposable
- Maximum Likelihood Estimation
- Model Testing
- Model Selection (Data Mining)

Why the log-linear representation?

Why do we use the log-linear representation of the probability table?

1. We are interested in expressing conditional independence constraints.
2. There is a straightforward correspondence between such constraints being satisfied, and the elimination of certain collections of u-terms from the log-linear expansion.
3. This correspondence is established by applying the factorisation criterion: $X \perp\!\!\!\perp Y \mid Z$ if and only if there exist functions g and h such that

$$\boxed{\log P(x, y, z) = g(x, z) + h(y, z)}$$

Three Dimensional Bernoulli

The joint distribution of three binary variables can be written:

$$P(x_1, x_2, x_3) = p(0,0,0)^{(1-x_1)(1-x_2)(1-x_3)} \cdots p(1,1,1)^{x_1 x_2 x_3}$$

Log-linear expansion

$$\begin{aligned} \log P(x_1, x_2, x_3) &= u_\emptyset + u_1 x_1 + u_2 x_2 + u_3 x_3 + u_{12} x_1 x_2 + \\ &\quad u_{13} x_1 x_3 + u_{23} x_2 x_3 + u_{123} x_1 x_2 x_3 \end{aligned}$$

With

$$\begin{aligned} u_{123} &= \log \frac{p(1,1,1)p(1,0,0)}{p(1,1,0)p(1,0,1)} - \log \frac{p(0,1,1)p(0,0,0)}{p(0,1,0)p(0,0,1)} \\ &= \log \frac{\text{cpr}(X_2, X_3 | X_1 = 1)}{\text{cpr}(X_2, X_3 | X_1 = 0)} \end{aligned}$$

Independence and the u-terms

Observation:

$$X_2 \perp\!\!\!\perp X_3 | X_1 \Leftrightarrow u_{23} = 0 \text{ and } u_{123} = 0$$

Proof: use factorisation criterion.

$X_2 \perp\!\!\!\perp X_3 | X_1 \Leftrightarrow$ there are functions $g(x_1, x_2)$ and $h(x_1, x_3)$ such that

$$\log P(x_1, x_2, x_3) = g(x_1, x_2) + h(x_1, x_3)$$

This is only possible when $u_{23} = 0$ (so the term $x_2 x_3$ drops out), and $u_{123} = 0$ (so the term $x_1 x_2 x_3$ drops out).

Log-linear expansion: non-binary variables

For a 2×2 table the log-linear expansion is given by:

$$\log P(x_1, x_2) = u_\emptyset + u_1 x_1 + u_2 x_2 + u_{12} x_1 x_2$$

for $x \in \{0, 1\}^2$.

What if the x_i have more than two levels? In that case the u terms become functions of x rather than constants:

$$\log P(x_1, x_2) = u_\emptyset + u_1(x_1) + u_2(x_2) + u_{12}(x_1, x_2)$$

Log-linear expansion: non-binary variables

Suppose $x \in \{0, 1, 2\}$. We can write

$$P(x) = p(1)^{\delta_{x=1}} p(2)^{\delta_{x=2}} p(0)^{(1-\delta_{x=1}-\delta_{x=2})},$$

where δ_A is the indicator function, that is,

$$\delta_A = \begin{cases} 1 & \text{if } A \text{ is true} \\ 0 & \text{otherwise} \end{cases}$$

Taking logarithms left and right, we get

$$\begin{aligned} \log P(x) &= \delta_{x=1} \log p(1) + \delta_{x=2} \log p(2) + (1 - \delta_{x=1} - \delta_{x=2}) \log p(0) \\ &= \delta_{x=1} \log p(1) + \delta_{x=2} \log p(2) + \log p(0) - \delta_{x=1} \log p(0) - \delta_{x=2} \log p(0) \\ &= \log p(0) + \log \frac{p(1)}{p(0)} \delta_{x=1} + \log \frac{p(2)}{p(0)} \delta_{x=2} \\ &= u_\emptyset + u(x) \end{aligned}$$

Log-linear expansion: non-binary variables

Where $u_\emptyset = \log p(0)$ and

$$u(x) = \begin{cases} \log \frac{p(1)}{p(0)} & \text{if } x = 1 \\ \log \frac{p(2)}{p(0)} & \text{if } x = 2 \\ 0 & \text{if } x = 0 \end{cases}$$

Similar rules apply to the case of multiple non-binary variables.

Log-linear expansion: general

The log-linear expansion of the probability distribution P_K is

$$\log P_K(x) = \sum_{a \subseteq K} u_a(x_a)$$

where the sum is taken over all possible subsets a of $K = \{1, 2, \ldots, k\}$.

- We get a (set of) u-term(s) for each subset of the variables.
- Code the values of X_i as $\{0, 1, \ldots, d_i - 1\}$, where d_i is size of the domain of X_i.
- Set $u_a(x_a) = 0$ whenever $x_i = 0$ for any X_i with $i \in a$.
- This is analogous to the case where X is binary.
- There are as many u-terms in the full log-linear expansion as there are cells in the contingency table.

Independence and the u-terms

If (X_a, X_b, X_c) is a partitioned random vector, then

$$X_b \perp\!\!\!\perp X_c \mid X_a$$

if and only if all u-terms in the log-linear expansion with coordinates in both b and c, are equal to zero.

Proof: Let s be an arbitrary subset of $a \cup b \cup c = \{1, 2, \ldots, k\}$.

If all u-terms, u_s, are zero whenever $s \not\subseteq a \cup b$ $(s \cap c \neq \emptyset)$ and $s \not\subseteq a \cup c$ $(s \cap b \neq \emptyset)$ (i.e. whenever s contains coordinates from both b and c) then we can write

$$\log P_K(x) = \sum_{s \subseteq a \cup b} u_s(x_s) + \sum_{s \subseteq a \cup c} u_s(x_s) - \sum_{s \subseteq a} u_s(x_s)$$

But this function is of the form $g(x_a, x_b) + h(x_a, x_c)$ and hence $X_b \perp\!\!\!\perp X_c \mid X_a$ by the factorisation criterion.

Independence and the u-terms: Example

Let $X = (X_1, \ldots, X_5)$, and $a = \{1, 3\}, b = \{4\}, c = \{2, 5\}$. According to the factorization criterion, the conditional independence

$$X_4 \perp\!\!\!\perp (X_2, X_5) \mid (X_1, X_3)$$

holds if, and only if, there are functions g and h such that

$$\log P(x_1, \ldots, x_5) = g(x_1, x_3, x_4) + h(x_1, x_2, x_3, x_5)$$

For this to be possible, the u-terms that contain elements from both the sets $\{4\}$ and $\{2, 5\}$ have to be zero. So all u-terms $u_{24}, u_{45}, u_{124}, u_{145}, \ldots, u_{12345}$ have to be zero.

Hierarchical Log-Linear Models

- In most applications, it does not make sense to include the three-way association u_{123} unless the two-way associations u_{12}, u_{13} and u_{23} are all present as well.
- A log-linear model is said to be *hierarchical* if the presence of a term implies that all lower-order terms are also present. That is, if $u_A(x_A)$ is present, then for all $a \subseteq A$, $u_a(x_a)$ must be present as well.
- Hence, a hierarchical model is uniquely identified by listing its highest order interaction terms.

Hierarchical Models for three dimensions

Model	Omitted	Interpretation
123	none	saturated
12,13,23	u_{123}	homogeneous association
12,13	u_{123}, u_{23}	$X_2 \perp\!\!\!\perp X_3 \mid X_1$
12,23	u_{123}, u_{13}	$X_1 \perp\!\!\!\perp X_3 \mid X_2$
13,23	u_{123}, u_{12}	$X_1 \perp\!\!\!\perp X_2 \mid X_3$
12,3	u_{123}, u_{13}, u_{23}	$(X_1, X_2) \perp\!\!\!\perp X_3$
13,2	u_{123}, u_{12}, u_{23}	$(X_1, X_3) \perp\!\!\!\perp X_2$
23,1	u_{123}, u_{12}, u_{13}	$(X_2, X_3) \perp\!\!\!\perp X_1$
1,2,3	$u_{123}, u_{12}, u_{13}, u_{23}$	mutual independence

Graphical Log-Linear Model

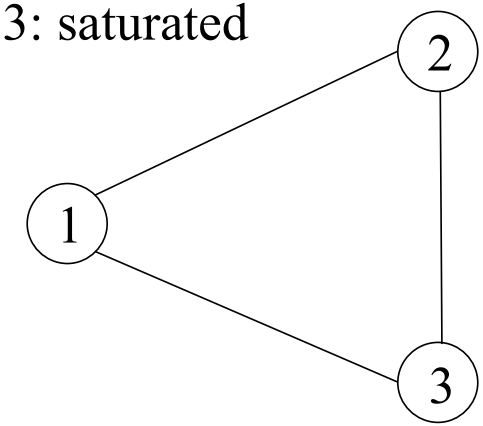
Given its independence graph $G = (K, E)$, the log-linear model for the random vector X is a *graphical model* for X if the distribution of X is *arbitrary* apart from constraints of the form that for all pairs of coordinates not in the edge set E, the u-terms containing the selected coordinates are equal to zero.

All constraints of a graphical model can be read from the independence graph.

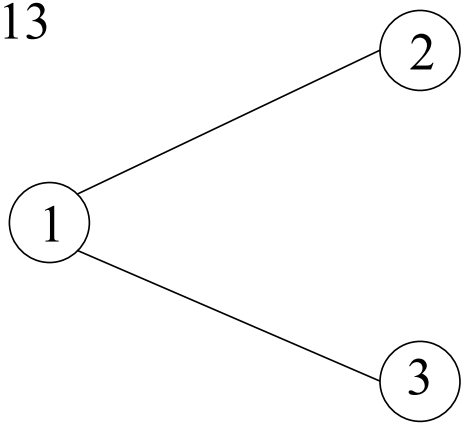
A graphical model is a hierarchical model in which the highest order interaction terms correspond to the cliques in the graph.

Hierarchical models and their independence graphs

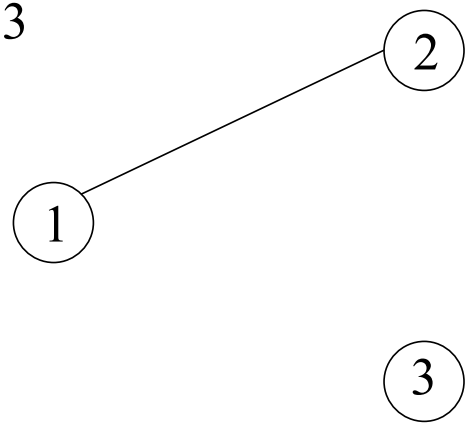
123: saturated



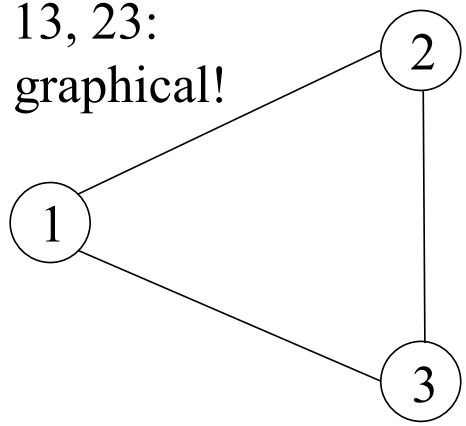
12,13



12,3



12, 13, 23:
not graphical!



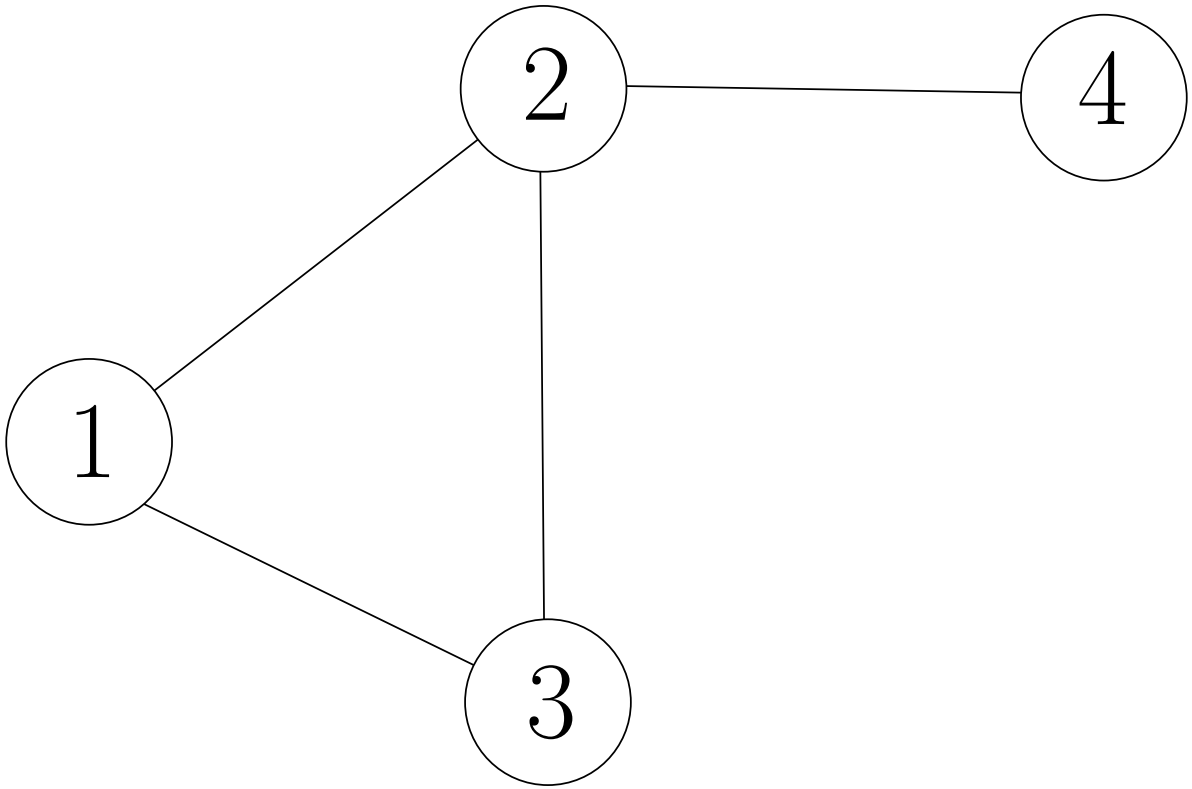
Maximum Likelihood Estimation

- ML estimator of graphical log-linear model M returns estimates of the cell probabilities that maximize the probability of the observed data, subject to the constraint that the conditional independencies of M are satisfied by the estimates.
- ML estimator of graphical log-linear model M satisfies the likelihood equations

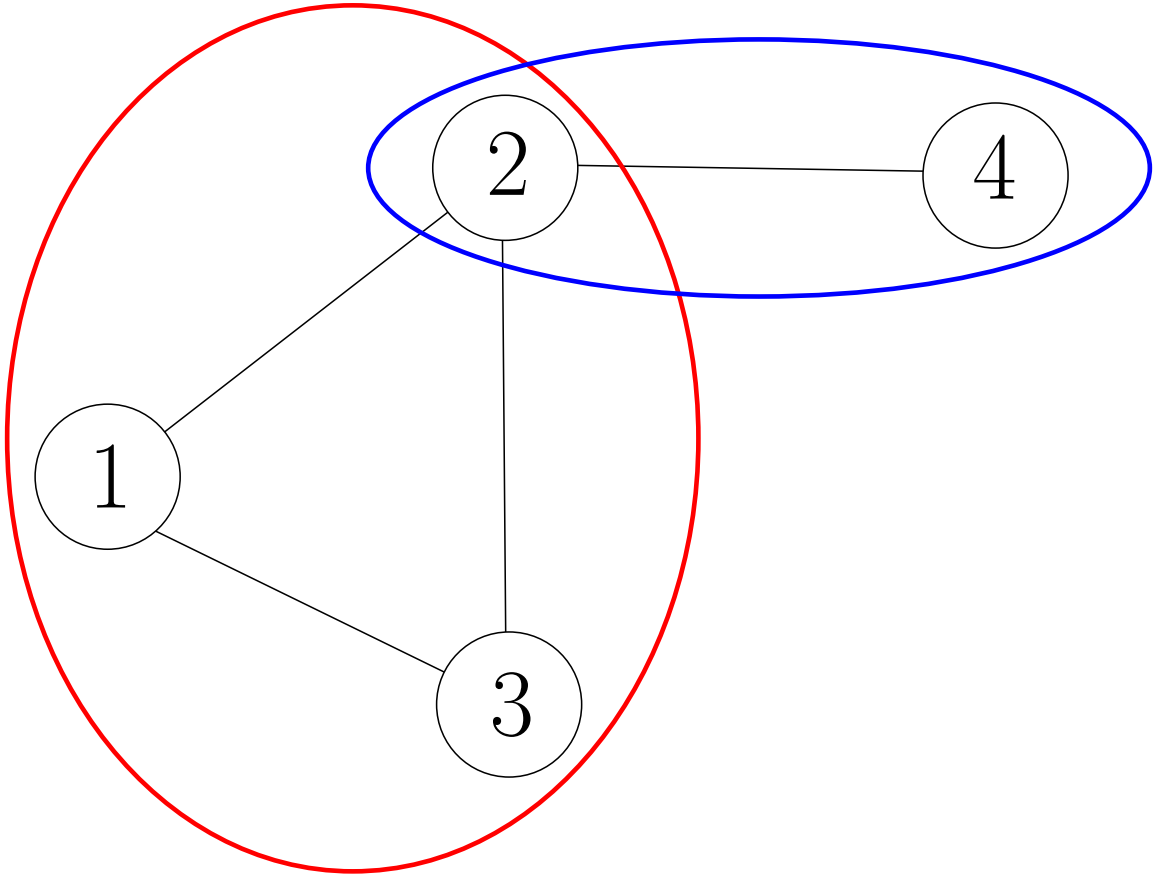
$$\hat{n}_a^M = N\hat{P}_a^M = n_a$$

  whenever the subset of vertices a in the graph form a clique.
- Slogan: Observed = Fitted for every marginal table corresponding to a complete subgraph.
- The same likelihood equations hold for all hierarchical models, where the margins a correspond to the highest order interaction terms in the model.

ML: Determine the cliques



ML: Observed=Fitted for margins corresponding to cliques



$$\hat{n}(x_1, x_2, x_3) = n(x_1, x_2, x_3)$$
$$\hat{n}(x_2, x_4) = n(x_2, x_4)$$

ML: Observed=Fitted for margins corresponding to cliques

We can see as follows why this has to be the case:

1. If there are no constraints to fit an observed table of counts, then the parameter estimates that yield fitted counts equal to the observed counts maximize the likelihood function. For example, the saturated model will yield fitted counts identical to the observed counts.
2. By definition, a graphical model is arbitrary (has no constraints) except for the constraints that can be read from the independence graph.
3. Suppose a forms a clique in the independence graph. Now consider the partitioning $X = (X_a, X_b)$ where b contains all variables not in a.
   We can write (product rule):

$$P(X) = P(X_a)P(X_b \mid X_a)$$

   Since $P(X_a)$ is not constrained by the model (complete graph), all model constraints apply only to $P(X_b \mid X_a)$. Therefore, the maximum likelihood estimates will yield $\hat{n}_a = n_a$.

ML: Example

$$
\begin{aligned}
\hat{P}(x_1, x_2, x_3, x_4) &= \hat{P}(x_1, x_3, x_4 | x_2)\hat{P}(x_2) && \text{(product rule)} \\
&= \hat{P}(x_1, x_3 | x_2)\hat{P}(x_4 | x_2)\hat{P}(x_2) && (X_4 \perp\!\!\!\perp (X_1, X_3) \mid X_2) \\
&= \hat{P}(x_1, x_3 | x_2)\hat{P}(x_2, x_4) && \text{(product rule)} \\
&= \frac{\hat{P}(x_1, x_2, x_3)\hat{P}(x_2, x_4)}{\hat{P}(x_2)} && \text{(product rule)}
\end{aligned}
$$

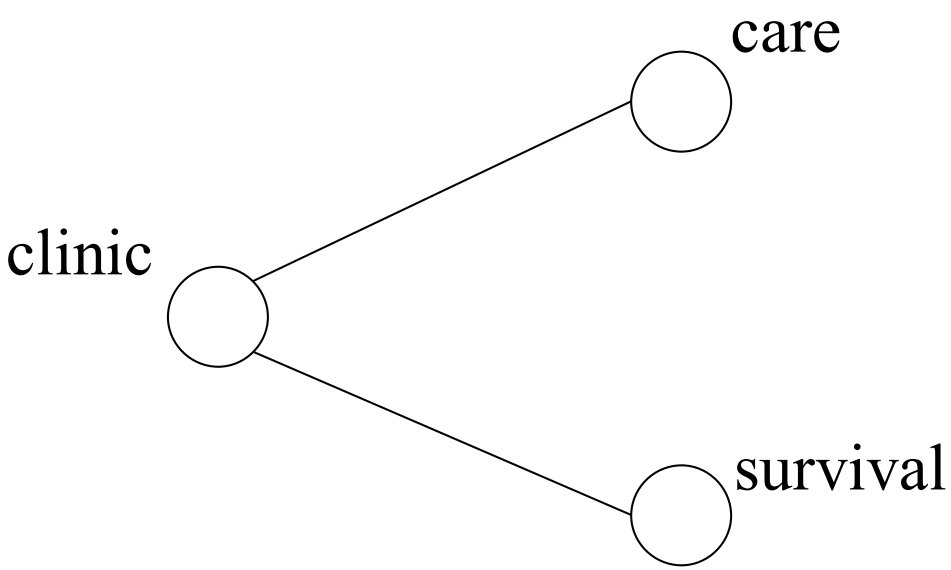
In terms of counts we have:

$$
\begin{aligned}
\hat{n}(x_1, x_2, x_3, x_4) &= \frac{\hat{n}(x_1, x_2, x_3)\hat{n}(x_2, x_4)}{\hat{n}(x_2)} \\
&= \frac{n(x_1, x_2, x_3)n(x_2, x_4)}{n(x_2)} && \text{(fitted = observed for complete subgraph)}
\end{aligned}
$$

In this case we have a closed form solution for the maximum likelihood fitted counts.

ML Estimation: Example

n_{123}		survival	
clinic	care	no	yes
clinic 1	less	3	176
	more	4	293
clinic 2	less	17	197
	more	2	23



Sufficient Statistics

n_{12}	care	
clinic	less	more
clinic 1	179	297
clinic 2	214	25

n_{13}	survival	
clinic	no	yes
clinic 1	7	469
clinic 2	19	220

Fitted values

$$\hat{n}_{123}(x) = \frac{n_{12}(x_1, x_2) n_{13}(x_1, x_3)}{n_1(x_1)}$$

$\hat{n}_{123}$		survival	
clinic	care	no	yes
clinic 1	less	2.63	176.37
	more	4.37	292.63
clinic 2	less	17.01	196.99
	more	1.99	23.01

Model seems to fit very well!

Iterative Proportional Fitting (IPF)

IPF is an algorithm to compute the maximum likelihood fitted counts for hierarchical log-linear models.

Fit independence model to

$n(x_1, x_2)$		$x_2 = 0$	$x_2 = 1$	$n_1(x_1)$
	$x_1 = 0$	30	10	40
	$x_1 = 1$	30	30	60
	$n_2(x_2)$	60	40	100

Sufficient statistics are row totals $n_1(x_1)$ and column totals $n_2(x_2)$.

Iterative Proportional Fitting

We begin with a table $\hat{n}^{(0)}$ of uniform counts

	0	1	
0	1	1	2
1	1	1	2
	2	2	

First step: fit to row margin

$$\hat{n}(x_1, x_2)^{(1)} = n_1(x_1) \times \frac{\hat{n}(x_1, x_2)^{(0)}}{\hat{n}_1(x_1)^{(0)}}$$

We compute (row 1):

$$\hat{n}(0,0)^{(1)} = 40 \times \frac{1}{2} = 20 \qquad \hat{n}(0,1)^{(1)} = 40 \times \frac{1}{2} = 20$$

Iterative Proportional Fitting

First step continued (row 2):

$$\hat{n}(1,0)^{(1)} = 60 \times \frac{1}{2} = 30 \qquad \hat{n}(1,1)^{(1)} = 60 \times \frac{1}{2} = 30$$

which yields $\hat{n}^{(1)}$:

	0	1	
0	20	20	40
1	30	30	60
	50	50	

Iterative Proportional Fitting

Second step: fit to column margin

$$\hat{n}(x_1, x_2)^{(2)} = n_2(x_2) \times \frac{\hat{n}(x_1, x_2)^{(1)}}{\hat{n}_2(x_2)^{(1)}}$$

Which gives (first column):

$$\hat{n}(0,0)^{(2)} = 60 \times \frac{20}{50} = 24 \qquad \hat{n}(1,0)^{(2)} = 60 \times \frac{30}{50} = 36$$

and (second column):

$$\hat{n}(0,1)^{(2)} = 40 \times \frac{20}{50} = 16 \qquad \hat{n}(1,1)^{(2)} = 40 \times \frac{30}{50} = 24$$

IPF

This yields $\hat{n}^{(2)}$:

	0	1	
0	24	16	40
1	36	24	60
	60	40	

Notice that the row totals are still 40 and 60, so we have simultaneously satisfied the conditions

$$\hat{n}_1(x_1) = n_1(x_1) \text{ and } \hat{n}_2(x_2) = n_2(x_2)$$

so we have converged.

IPF: Homogeneous association

Fit the model: 12,13,23

IPF proportionally adjusts the estimated expected frequencies $\hat{n}_{123}(x)$ to satisfy the constraints

1. $\hat{n}_{12}(x_1, x_2) = n_{12}(x_1, x_2)$
2. $\hat{n}_{13}(x_1, x_3) = n_{13}(x_1, x_3)$
3. $\hat{n}_{23}(x_2, x_3) = n_{23}(x_2, x_3)$

IPF: One iteration

Fit to 12 margin:

$$\hat{n}_{123}(x)^{(t+1)} = n_{12}(x_1, x_2) \left(\frac{\hat{n}_{123}(x)^{(t)}}{\hat{n}_{12}(x_1, x_2)^{(t)}} \right)$$

Fit to 13 margin:

$$\hat{n}_{123}(x)^{(t+2)} = n_{13}(x_1, x_3) \left(\frac{\hat{n}_{123}(x)^{(t+1)}}{\hat{n}_{13}(x_1, x_3)^{(t+1)}} \right)$$

Fit to 23 margin:

$$\hat{n}_{123}(x)^{(t+3)} = n_{23}(x_2, x_3) \left(\frac{\hat{n}_{123}(x)^{(t+2)}}{\hat{n}_{23}(x_2, x_3)^{(t+2)}} \right)$$

IPF: General Algorithm Sketch

Say we have m margins $\{a_1, a_2, \ldots, a_m\}$ to be fitted $(\cup_i a_i = K)$.

We have to find a table $\hat{n}(x)$ that agrees with the observed table $n(x)$ on the m margins corresponding to the subsets a_i.

The algorithm cycles through the list of subsets

$$a = a_i, \qquad i = 1, 2, \ldots, m$$

fitting $\hat{n}(x)$ to each margin in turn.

This is repeated until convergence is reached, i.e. all margin constraints are (approximately) satisfied simultaneously.

IPF updating rule

To fit to the margin a, the observed count $n_a(x_a)$ on x_a is distributed over $\hat{n}_{ab}(x_a, x_b)^{(t+1)}$ according to

$$\hat{n}_{ab}(x_a, x_b)^{(t+1)} = n_a(x_a)\hat{P}(x_b|x_a)^{(t)}$$

where b is the complement of a, and

$$\hat{P}(x_b|x_a)^{(t)} = \frac{\hat{n}_{ab}(x_a, x_b)^{(t)}}{\hat{n}_a(x_a)^{(t)}},$$

i.e., the current estimate of $P(X_b = x_b|X_a = x_a)$.

IPF Pseudocode

Algorithm 1 IPF($n(x)$, $\mathcal{A}$)

1: $t \leftarrow 0$
2: **for all** values x of X **do**
   $\hat{n}(x)^{(t)} \leftarrow 1$
3: **end for**
4: **repeat**
5: **for all** margins $a \in \mathcal{A}$ **do**
6: **for all** values x_a of X_a **do**
7: **for all** values x_b of X_b **do**
$\hat{n}_{ab}(x_a, x_b)^{(t+1)} \leftarrow n_a(x_a) \left(\frac{\hat{n}_{ab}(x_a, x_b)^{(t)}}{\hat{n}_a(x_a)^{(t)}} \right)$
8: **end for**
9: **end for**
10: $t \leftarrow t + 1$
11: **end for**
12: **until** convergence

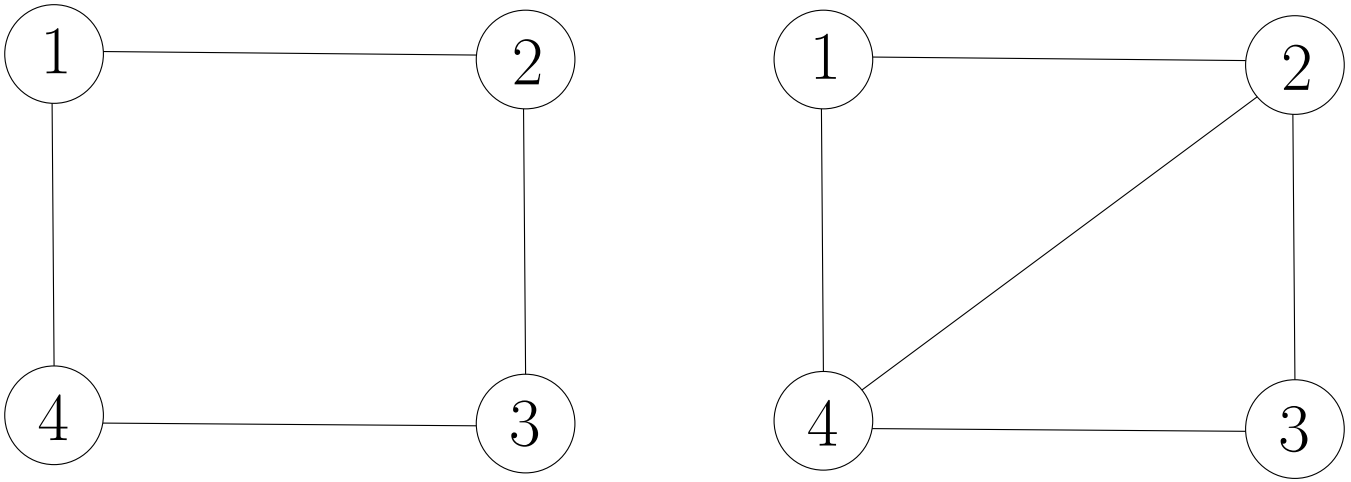
Decomposable Graphical Models

Decomposable models have explicit formulas for the MLE's.

Decomposable models have *triangulated* independence graphs, i.e. have no *chordless cycles* of length greater than three.

A cycle is chordless if only the *successive* pairs of vertices in the cycle are connected by an edge. (There is no "shortcut").

Example



- The left graph is *not* decomposable because it contains the chordless 4-cycle $1 - 2 - 3 - 4 - 1$.
- The graph on the right *is* decomposable.
  The cycle $1 - 2 - 3 - 4 - 1$ is no longer chordless because 2 and 4 are adjacent in the graph but not successive in the cycle.

ML Estimation for Decomposable Models

An ordering $C_1, \ldots, C_m$ of the cliques of the graph has the running intersection property (RIP) iff

$$C_j \cap (C_1 \cup \ldots \cup C_{j-1}) \subseteq C_i,$$

for some $i < j$, and for $j = 2, \ldots, m$.

We define the corresponding separator sets

$$S_j = C_j \cap (C_1 \cup \ldots \cup C_{j-1}),$$

with $S_1 = \emptyset$.

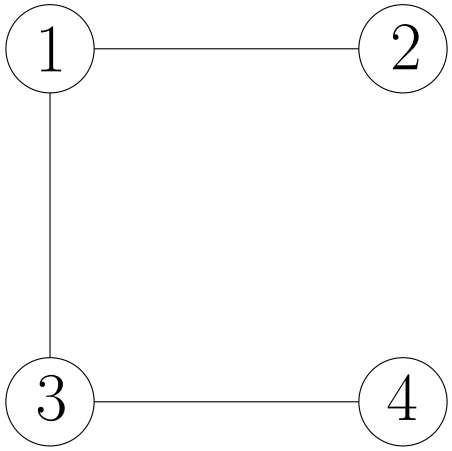
ML Estimation for Decomposable Models

The maximum likelihood fitted counts are given by

$$\hat{n}(x) = \frac{\prod_{j=1}^{m} n(x_{C_j})}{\prod_{j=2}^{m} n(x_{S_j})}$$

where $n(x_\emptyset) = N$.

Example



The clique ordering $\{1,2\},\{3,4\},\{1,3\}$ does not have the running intersection property and does therefore not produce correct estimates.

The RIP clique ordering $\{1,2\},\{1,3\},\{3,4\}$ gives ML estimates

$$\hat{n}(x_1,x_2,x_3,x_4) = \frac{n(x_1,x_2)n(x_1,x_3)n(x_3,x_4)}{n(x_1)n(x_3)}$$

Decomposable Models and IPF

- If the cliques of a decomposable model are presented in RIP order to IPF, then the algorithm will converge in one iteration (one cycle through all cliques).
- Otherwise IPF will converge in two iterations.

Likelihood score of a model

The likelihood score of a model M is

$$L^M = \prod_x \hat{P}^M(x)^{n(x)},$$

where $\hat{P}^M(x)$ is the fitted probability of cell x according to model M.

Hence, the likelihood score of model M is the probability of the observed data using the fitted cell probabilities according to model M.

Likewise, the log-likelihood score of a model M is

$$\mathcal{L}^M = \sum_x n(x) \log \hat{P}^M(x)$$

Model Deviance

Since for the saturated model

$$\hat{P}(x) = \frac{n(x)}{N},$$

the log-likelihood score of the saturated model is

$$\mathcal{L}^{\text{sat}} = \sum_x n(x) \log \frac{n(x)}{N}$$

The deviance of a fitted model compares the log-likelihood score of the fitted model to that of the saturated model.

The larger the model deviance, the poorer the fit.

Example

Suppose we have data

$n(x)$	$x_2 = 0$	$x_2 = 1$	$n(x_1)$
$x_1 = 0$	30	10	40
$x_1 = 1$	30	30	60
$n(x_2)$	60	40	100

The independence model gives probability estimates: $\hat{P}(0,0) = 0.24$, $\hat{P}(0,1) = 0.16$, $\hat{P}(1,0) = 0.36$, $\hat{P}(1,1) = 0.24$.

The probability of the observed data according to this model is

$$0.24^{30} \times 0.16^{10} \times 0.36^{30} \times 0.24^{30}$$

This is the likelihood score of the model given the data.
The log-likelihood score is

$$\mathcal{L} = 30 \log 0.24 + 10 \log 0.16 + 30 \log 0.36 + 30 \log 0.24 \approx -134.6$$

Example (continued)

$n(x)$	$x_2 = 0$	$x_2 = 1$	$n(x_1)$
$x_1 = 0$	30	10	40
$x_1 = 1$	30	30	60
$n(x_2)$	60	40	100

The saturated model gives probability estimates: $\hat{P}(0,0) = 0.3$, $\hat{P}(0,1) = 0.1$, $\hat{P}(1,0) = 0.3$, $\hat{P}(1,1) = 0.3$.

The probability of the observed data according to this model is

$$0.3^{30} \times 0.1^{10} \times 0.3^{30} \times 0.3^{30}$$

The log-likelihood score is

$$\mathcal{L} = 30 \log 0.3 + 10 \log 0.1 + 30 \log 0.3 + 30 \log 0.3 \approx -131.4$$

Of course this is better than the independence model.

Model Deviance

Deviance of M is 2 (log-likelihood of the saturated model $-$ log-likelihood of M):

$$\begin{aligned}
\text{dev}(M) &= 2\left(\sum_x n(x)\log\frac{n(x)}{N} - \sum_x n(x)\log\hat{P}^M(x)\right) \\
&= 2\left(\sum_x n(x)\left(\log\frac{n(x)}{N} - \log\hat{P}^M(x)\right)\right) \\
&= 2\sum_x n(x)\log\frac{n(x)}{N\hat{P}^M(x)}
\end{aligned}$$

which can be summarised by the *slogan*

$$2\sum_{\text{cells}} \text{observed} \times \log\frac{\text{observed}}{\text{fitted}}$$

Deviance difference

Let $M_0 \subseteq M_1$, that is M_0 is the simpler model
(the u-terms present in M_0 are a subset of the u-terms present in M_1).

The *deviance difference* between M_0 and M_1 is

$$\text{dev}(M_0) - \text{dev}(M_1) = -2\mathcal{L}^{M_0} + 2\mathcal{L}^{M_1} = 2(\mathcal{L}^{M_1} - \mathcal{L}^{M_0})$$

For large N we have that:

$$2(\mathcal{L}^{M_1} - \mathcal{L}^{M_0}) \approx_{M_0} \chi^2_\nu$$

ν: number of *additional* restrictions (zero u-terms) of M_0 compared to M_1.
(ν is called the degrees of freedom)

Likelihood Ratio Test

We reject the null hypothesis that M_0 is the true model when

$$2(\mathcal{L}^{M_1} - \mathcal{L}^{M_0}) > \chi^2_{\nu;\alpha},$$

where α is the significance level of the test, and $P(X^2 > \chi^2_{\nu;\alpha}) = \alpha$.

The test is called a likelihood ratio test because

$$\log \frac{L^{M_1}}{L^{M_0}} = \log L^{M_1} - \log L^{M_0} = \mathcal{L}^{M_1} - \mathcal{L}^{M_0}$$

Model Testing: example

Does

$$\text{survival} \perp\!\!\!\perp \text{care} \mid \text{clinic}$$

give a good fit of the observed table? Test against the saturated model.

Compute the deviance

$$2 \sum_{\text{cells}} \text{observed} \times \log \frac{\text{observed}}{\text{fitted}} \approx 0.082$$

$$\chi^2_{2;0.05} \approx 6$$

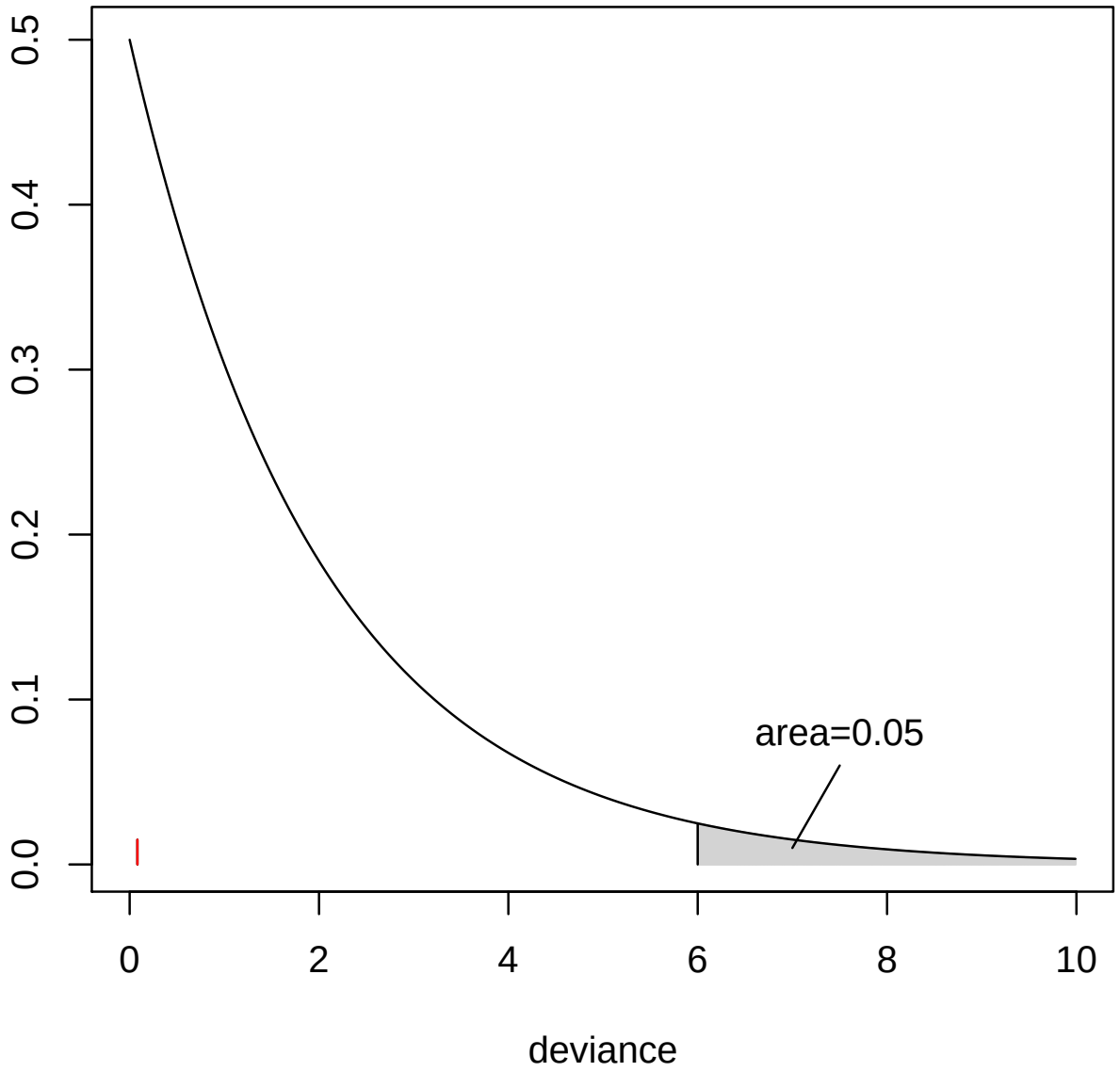
So we "accept" the model.

Fitted Counts and Observed Counts

$\hat{n}$(clinic, care, survival)		survival	
clinic	care	no	yes
clinic 1	less	2.63	176.37
	more	4.37	292.63
clinic 2	less	17.01	196.99
	more	1.99	23.01

n(clinic, care, survival)		survival	
clinic	care	no	yes
clinic 1	less	3	176
	more	4	293
clinic 2	less	17	197
	more	2	23

Test of survival $\perp\!\!\!\perp$ care|clinic; χ^2_2 distribution.



Model Testing: example

Does the mutual independence model give a good fit of the observed table? Test against the saturated model.

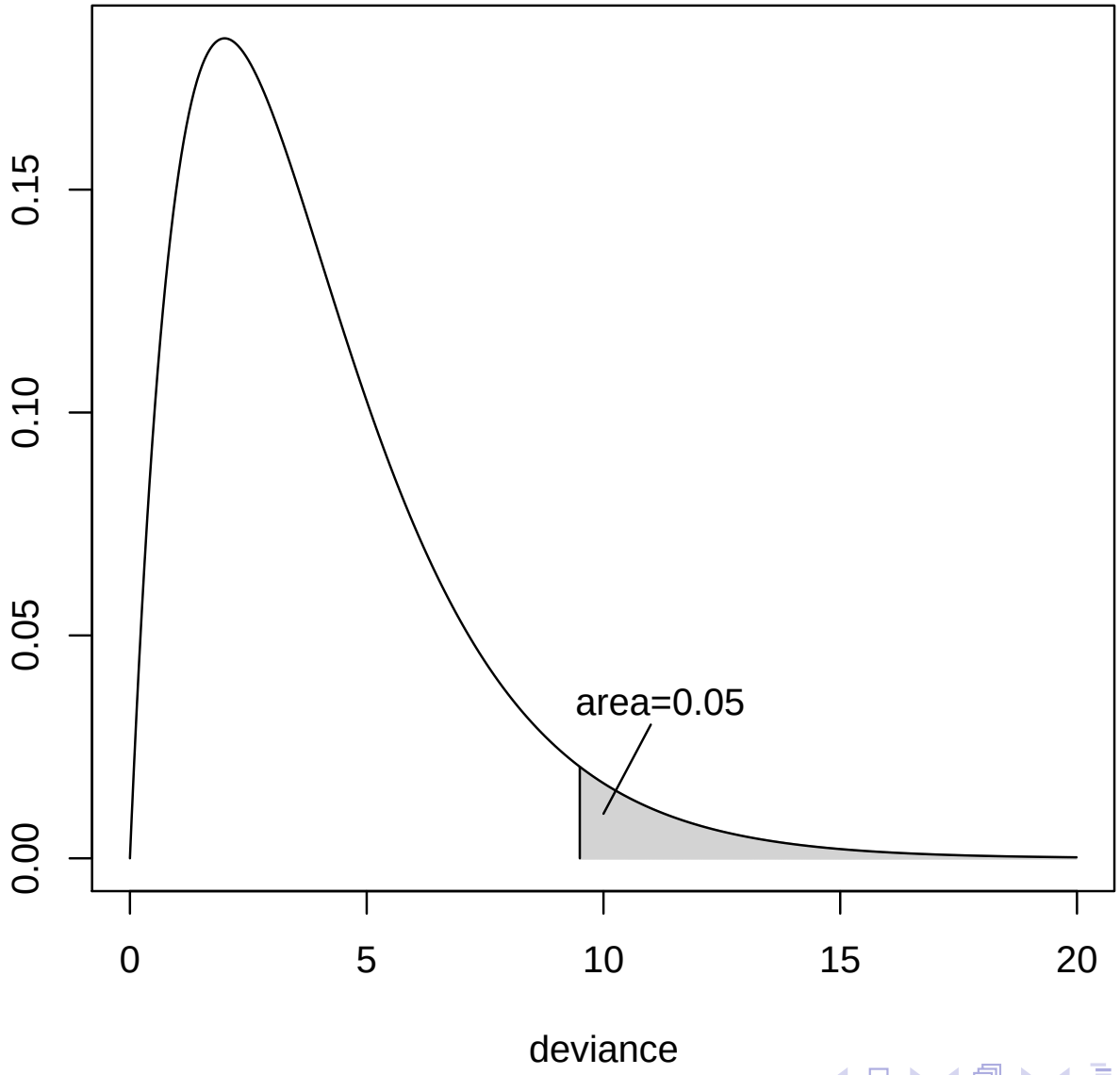
Compute the deviance

$$2 \sum_{\text{cells}} \text{observed} \times \log \frac{\text{observed}}{\text{fitted}} \approx 211$$

$$\chi^2_{4;0.05} \approx 9.5$$

So we reject the mutual independence model.

Test of Independence Model; χ^2_4 distribution.



Model Selection

The Problem: find a good model for a high-dimensional table when little prior knowledge is available.

Solution: Search the space of possible models.
Two approaches:

- Use significance testing
- Use a quality function

Quality Functions: AIC and BIC

Two components:

- the lack-of-fit of the model
- complexity of the model

Akaike's Information Criterion assigns quality AIC(M) to model M as follows

$$\mathrm{AIC}(M) = \mathrm{dev}(M) + 2\,\dim(M)$$

where $\dim(M)$ is the number of parameters (the number of u-terms) of the model.

Bayesian Information Criterion assigns quality BIC(M) to model M as follows

$$\mathrm{BIC}(M) = \mathrm{dev}(M) + \log(N)\,\dim(M)$$

Search

Exhaustive search is usually not feasible.

A straightforward approach is hill climbing:

1. pick some initial model
2. consider the quality of all neighbors of the current model
3. if they all have lower quality, stop and return the current model.
4. otherwise move to the neighbor with highest quality and return to 2.

Decomposable Graphical Models

1. pick an initial model, e.g. the empty graph
2. neighbors
   - add an edge
   - delete an edge

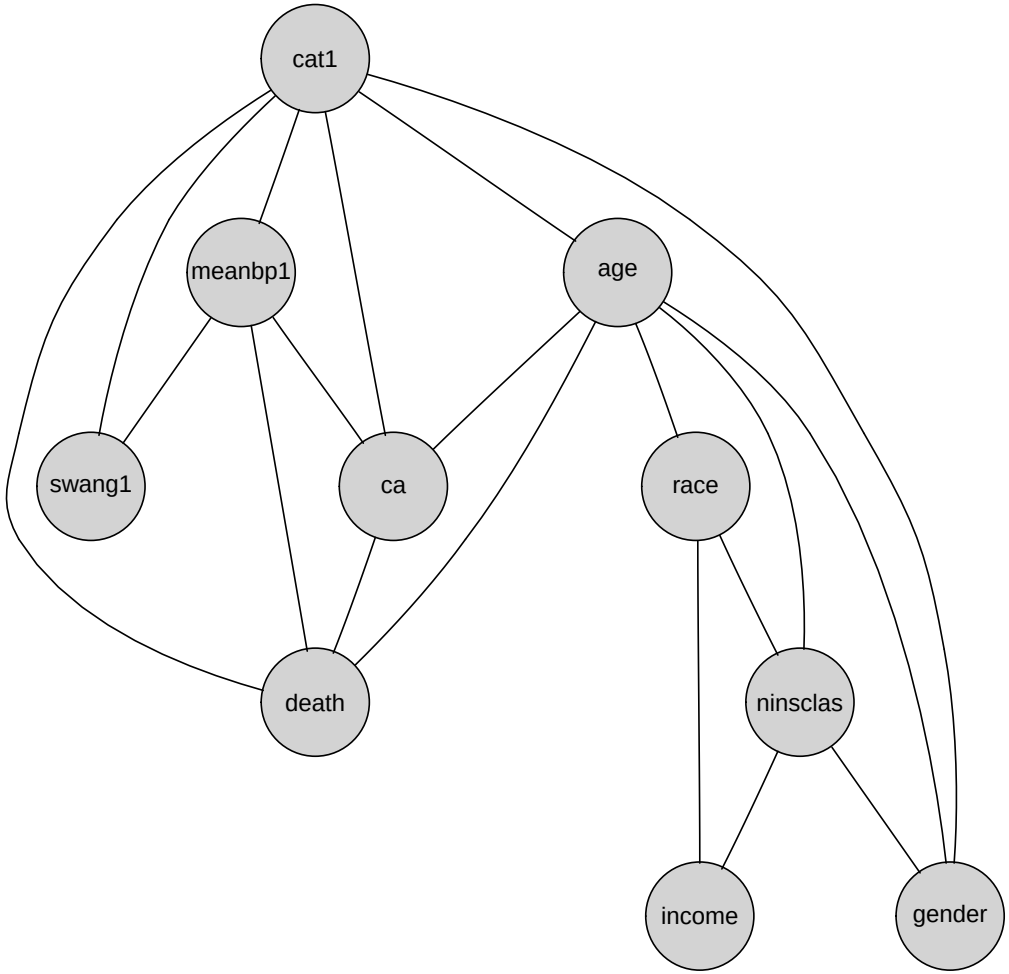
   (without creating a chordless cycle of length > 3)
3. if all neighbors have higher AIC, stop and return the current model.
4. otherwise move to the neighbor with lowest AIC and return to 2.

Local search in gRim

```
# fit initial model (empty graph)
> rhc.init <- dmod(~.^1,data=rhc.dat)
# display some info about this model
> summary(rhc.init)
is graphical=TRUE; is decomposable=TRUE
generators (glist):
  :"cat1"
  :"death"
  :"swang1"
  :"gender"
  :"race"
  :"ninsclas"
  :"income"
  :"ca"
  :"age"
  :"meanbp1"
# perform stepwise search of decomposable models using AIC (only add edges)
> rhc.step1 <- stepwise(rhc.init,direction="forward")
```

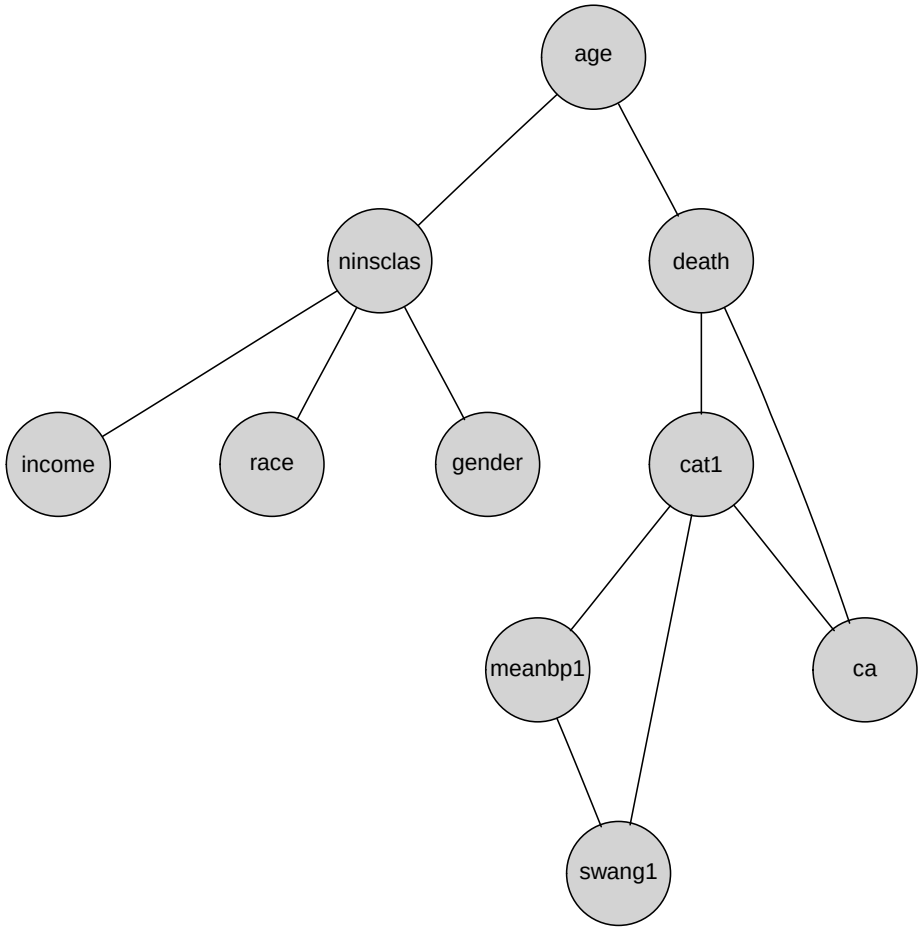
Result of Search (AIC, decomposable)

```
> plot(rhc.step1)
```



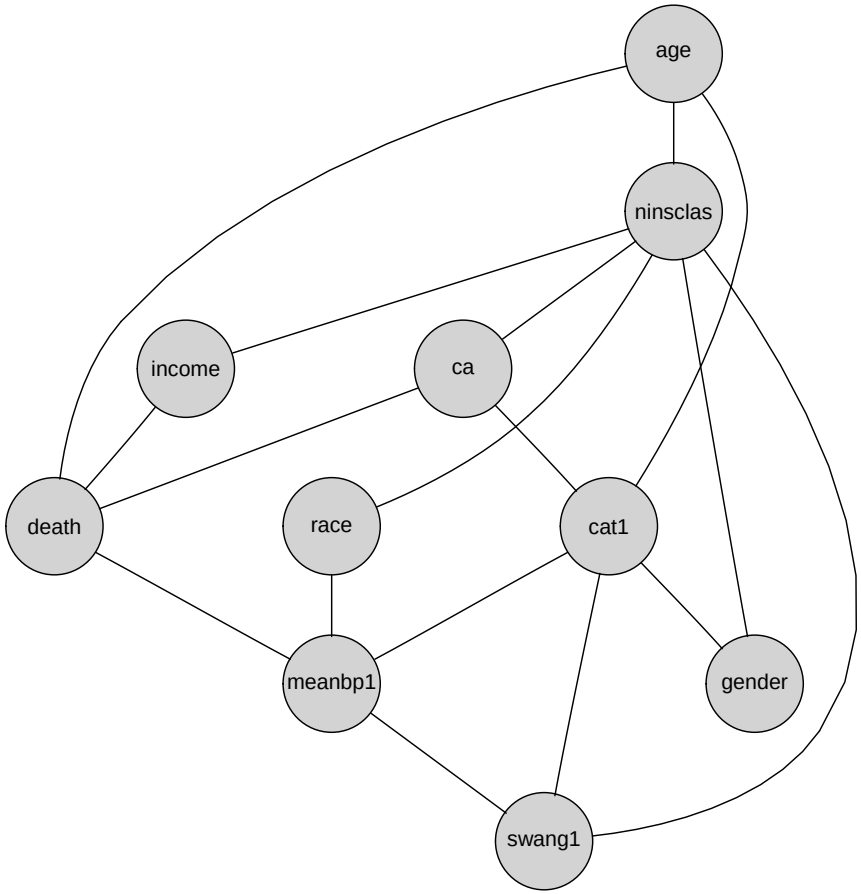
Result of Search (BIC, decomposable)

```
> rhc.step2 <- stepwise(rhc.init,direction="forward",k=log(nrow(rhc.dat)))
> plot(rhc.step2)
```



Result of Search (BIC, unrestricted)

```
> rhc.step3 <- stepwise(rhc.init,direction="forward",k=log(nrow(rhc.dat)),
                        type="unrestricted")
> plot(rhc.step3)
```



Result of Search (AIC, unrestricted)

```
> rhc.step4 <- stepwise(rhc.init,direction="both",
                        type="unrestricted",details=1)
STEPWISE:
 criterion: aic ( k = 2 )
 direction: both
 type     : unrestricted
 search   : all
 steps    : 1000
. FORWARD: type=unrestricted search=all, criterion=aic(2.00), alpha=0.00
. Initial model: is graphical=TRUE is decomposable=TRUE
  change.AIC -3061.4959 Edge added: ninsclas age
  change.AIC -1685.0212 Edge added: cat1 ca
  change.AIC -1347.0960 Edge added: income ninsclas
  change.AIC -420.0886 Edge added: swang1 cat1
  change.AIC -306.3228 Edge added: race ninsclas
  change.AIC -285.8656 Edge added: age cat1
  change.AIC -253.7602 Edge added: death ca
  change.AIC -497.6596 Edge added: age death
  ...
```

Model Use: Inference and Prediction with gRain

```
# prepare selected model for use in gRain
> rhc.mod2 <- grain(as(rhc.step2,"graphNEL"),rhc.dat)
# perform inference on "death" from evidence on "gender","ca", and "swang1"
> predict(rhc.mod2, c("death"), c("gender","ca","swang1"),
  data.frame(gender="Male",ca="Yes",swang1="No RHC"), type = "dist")
$pred
$pred$death
            No       Yes
[1,] 0.2235305 0.7764695

# change cancer to "Metastatic"
> predict(rhc.mod2, c("death"), c("gender","ca","swang1"),
  data.frame(gender="Male",ca="Metastatic",swang1="No RHC"), type = "dist")
$pred
$pred$death
             No       Yes
[1,] 0.09421555 0.9057845
```

Model Use: Inference and Prediction with gRain

```
# predict death (in-sample) from its Markov blanket
> death.pred <- predict(rhc.mod2, c("death"), c("ca","cat1","age"),
                        rhc.dat, type = "class")
> table(rhc.dat$death,death.pred$pred$death)

       No  Yes
  No   730 1283
  Yes  524 3198

> (730+3198)/nrow(rhc.dat)
[1] 0.6849172

# model is a little better than just predicting the majority class
> summary(rhc.dat$death)
  No  Yes
2013 3722
> 3722/nrow(rhc.dat)
[1] 0.6489974
```

Literature

- Y. Bishop, S.E. Fienberg, P.W. Holland, Discrete Multivariate Analysis, MIT Press, 1975.
- J. Whittaker, Graphical Models in Applied Multivariate Statistics, Wiley, 1990.
- D. Edwards, Introduction to Graphical Modelling (2nd edition), Springer, 2000.
- S. Højsgaard, D. Edwards and S. Lauritzen, Graphical Models with R, Springer, 2012.