Graphical Models for Discrete Data

Part 1:

1 Introduction

In this chapter we consider

Undirected Graphs

models that aim to represent the associations between a number of discrete variables. In contrast to for example classification trees and bump hunting we don't select a particular variable as a target that is to be explained or predicted by the other variables. Instead, all variables are treated on an equal footing: we simply want to model the associations between them. We confine our attention to discrete variables, although similar ideas have been developed for continuous as well as mixed discrete and continuous variables.

After giving a motivating example, we give a short review of the notions of independence and conditional independence of random variables. These notions are central to the interpretation of the type of models we are going to discuss. Next we start with the so called log-linear representation of a multi-way contingency table. This representation is convenient for our purpose because it allows us to express (conditional) independence constraints by setting certain coefficients equal to zero. In fact we define subclasses of the log-linear model that can be fully interpreted in terms of conditional independence relations. These are in order of inclusion: hierarchical models, graphical models and finally decomposable models. We discuss how such models can be estimated from data, sometimes requiring an iterative algorithm called iterative proportional fitting. Finally, we discuss how we can test whether a model gives a reasonable fit, and how one can select a good model when little prior knowledge is available concerning the conditional independence relations between the variables. Most of the material in this chapter is based on the book of Whittaker [Whi90]. Other sources used in writing this chapter are [Edw00, Sch97, Chr97, BFH75].

2 Example

Although we assume familiarity with the basic rules of probability, almost all results we use can be inferred from two elementary properties that we list here for reference:

$$P(X) = \sum_{Y} P(X, Y) \qquad \text{(sum rule)}$$

$$P(X, Y) = P(X|Y)P(Y) \qquad \text{(product rule)}$$

Consider problems where we have a collection of discrete random variables whose joint probability distribution has to be estimated from data. Now suppose we have k random variables each of which can take on m possible values. To estimate the probability of each possible combination would require the estimation of m^k probabilities. For a relatively small problem with 10 variables with 5 possible values each, this is

$$5^{10} = 9,765,625$$

say 10 million probabilities. Typically we have far fewer observations than that, so it is clear we cannot estimate all these probabilities reliably from the limited amount of data we have. This is one of the many manifestations of what is called the *curse of dimensionality*.

How can we simplify such a problem? How can we reduce the number of probabilities we have to estimate in a natural way? One of the most natural ways to do this is to exploit (conditional) independences that might hold in the problem domain. To illustrate, consider a problem with just two ternary variables. There are $3 \times 3 = 9$ possible value combinations, so without making any simplifying assumptions we have to estimate 8 probabilities (we subtract 1, because we have the constraint that the probabilities must sum to one). Now suppose we observe the counts displayed in Table 1.

To estimate the joint distribution of X and Y, we use

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

that is, we just look up how many times a particular combination of values of X and Y occurs in the data and divide this number by the total number

$n(x,y)$	y			
x	1	2	3	$n(x)$
1	2	5	3	10
2	10	20	10	40
3	8	35	7	50
$n(y)$	20	60	20	100

Table 1: Cross-table of counts for two ternary variables

$\hat{P}(x,y)$	y			
x	1	2	3	$\hat{P}(x)$
1	0.02	0.05	0.03	0.10
2	0.10	0.20	0.10	0.40
3	0.08	0.35	0.07	0.50
$\hat{P}(y)$	0.20	0.60	0.20	1

Table 2: Estimated joint distribution for two ternary variables

of observations. Hence, we obtain the estimated joint distribution as given in Table 2.

Now suppose we assume that X and Y are independent random variables. In that case, we can write

$$P(X, Y) = P(X)P(Y), \tag{1}$$

that is, the joint distribution can be written as the product of the marginal distributions. Now we only need to estimate the marginal distributions $P(X)$ and $P(Y)$ and plug these estimates into equation (1) to obtain an estimate of the joint probability. This requires the estimation of 2 probabilities (remember the sum to one constraint) for $P(X)$ and the same number for $P(Y)$, hence a total of just 4 probabilities. These estimates can be read from the margins (hence the name marginal probability) of Table 2 and filling them in in equation (1) gives the estimates as displayed in Table 3.

Another way of expressing the result is to compute the "fitted counts" of this model as displayed in Table 4. These are simply obtained by multiplying the estimated probabilities with the total number of observations. To

determine whether the independence assumption is justified, we compare the

$\hat{P}(x)\hat{P}(y)$	y			
x	1	2	3	$\hat{P}(x)$
1	0.02	0.06	0.02	0.10
2	0.08	0.24	0.08	0.40
3	0.10	0.30	0.10	0.50
$\hat{P}(y)$	0.20	0.60	0.20	1

Table 3: Estimated joint distribution for two ternary variables using independence assumption

$\hat{n}(x,y)$	y			
x	1	2	3	$n(x)$
1	2	6	2	10
2	8	24	8	40
3	10	30	10	50
$n(y)$	20	60	20	100

Table 4: Fitted counts for two ternary variables using independence assumption

observed counts with the fitted counts of the independence model. We observe that the fitted counts are not that far off, and the independence model seems to give a reasonable fit. To decide in a more justified manner whether the independence assumption should be accepted, a statistical test can be performed. We don't discuss this here.

Next we discuss a somewhat more complicated example. The data set displayed in Table 5 has been made famous by the book of Bishop, Fienberg and Holland [BFH75]. The data gives information on the survival rate of 715

infants attending two clinics and the amount of care received by the mother, where the amount of care is classified as either *more* or *less*. Table 6 gives the probability estimates corresponding to the saturated model (making no independence assumptions). These estimates are obtained simply by dividing the count in each cell of the table by the total number of observations.

Now consider the model that assumes survival and care are independent within each clinic. This is called a conditional independence assumption because we condition on clinic: we don't state survival and care are inde-

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

Table 5: Three-way tabel relating clinic, care and survival

$\hat{P}$(clinic, care, survival)		survival	
clinic	care	no	yes
clinic 1	less	.004	.246
	more	.006	.410
clinic 2	less	.024	.276
	more	.003	.032

Table 6: Estimated joint distribution of clinic, care and survival without making any independence assumptions (the so-called saturated model)

pendent per se, but that they are independent given clinic. This assumption corresponds to the following factorization

$$P(\text{care, survival}|\text{clinic}) = P(\text{care}|\text{clinic})P(\text{survival}|\text{clinic})$$

Multiplying left and right by $P(\text{clinic})$ we get

$$P(\text{care, survival, clinic}) = P(\text{care,clinic})P(\text{survival}|\text{clinic})$$

$$= \frac{P(\text{care,clinic})P(\text{survival,clinic})}{P(\text{clinic})}$$

As you can see from this last expression we have to estimate the joint distribution of care and clinic, and the joint distribution of survival and clinic (the marginal distribution of clinic is obtained by summing out care from the joint of care and clinic, or alternatively, by summing out survival from the joint of survival and clinic). To obtain the necessary counts, we take Table 5 and sum out care respectively survival. The resulting tables are displayed below

n(clinic, care)	care	
clinic	less	more
clinic 1	179	297
clinic 2	214	25

n(clinic, survival)	survival	
clinic	no	yes
clinic 1	7	469

clinic 2	19	220

Writing $\hat{n}$ for $\hat{P}n$ we get

$$\hat{n}(\text{clinic, care, survival}) = \frac{n(\text{clinic,care})n(\text{clinic,survival})}{n(\text{clinic})}$$

which gives the fitted values:

$\hat{n}$(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

To give one example, the fitted count for clinic=clinic 1, care=more, survival=yes is computed as follows

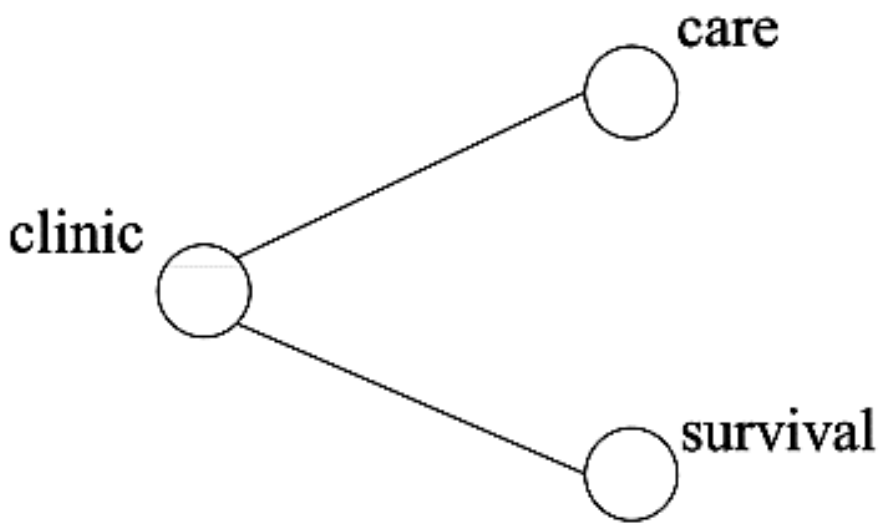
$$\hat{n}(\text{clinic 1, more, yes}) = \frac{n(\text{clinic 1,more})n(\text{clinic 1, yes})}{n(\text{clinic 1})}$$
$$= \frac{297 \times 469}{179 + 297} = 292.63$$

When we compare the fitted counts to the observed counts we see that they are very close. Hence the assumption that care and survival are independent within each clinic seems to be justified. Again, a rigorous statistical test will confirm this but is not discussed here. Within the first clinic the mortality rate for the less care group is practically the same as for the more care group; the same is true for the second clinic. In neither clinic is there a relationship between care and survival. In other words, given clinic, care and survival are independent. A graph that describes this structure is

	survival		
care	no	yes	(%)
less	20	373	5.1

more	6	316	1.9

Table 7: Cross-table of care and survival

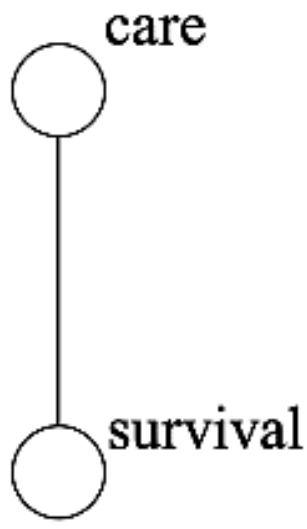


where the vertices correspond to the variables and lack of an edge between care and survival indicates that these variables are conditionally independent given clinic.

The reason this dataset has become well-known is that a strange phenomenon occurs when we sum out clinic, and then analyse the association

between care and survival. From table 7 one would conclude that the more maternal care received the lower the infant mortality rate, with the rate dropping by more than half.

Apparently, when the three-way table is collapsed over clinic a spurious association between care and survival is induced. The lack of independence suggests the graph



A lesson to learn is that it is dangerous to analyse a three-way table solely by inspecting its two way margins. Can you explain how the spurious association between care and survival comes about?

3 Independence and Conditional Independence

We give a short review of the concepts of independence and conditional independence of random vectors.

Random vectors X and Y are independent iff

$$P(x, y) = P(x)P(y) \text{ for all } (x, y),$$

and as a consequence $P(x \mid y) = P(x)$, and $P(y \mid x) = P(y)$. We also write $X \perp\!\!\!\perp Y$.

To establish independence, it is sufficient to show that the joint density function factorises rather than that it factorises into the product of the marginals. This gives us the factorisation criterion for independent random vectors: random vectors X and Y are independent iff there exist two functions g and h such that

$$P(x, y) = g(x)h(y) \text{ for all } (x, y)$$

We will often make use of the "log-version" of this criterion:

$$\log P(x, y) = g'(x) + h'(y) \text{ for all } (x, y)$$

Random vectors X and Y are independent given Z iff

$$P(x, y \mid z) = P(x \mid z)P(y \mid z)$$

for all (x, y) and for all z for which $P(z) > 0$. We also write $X \perp\!\!\!\perp Y \mid Z$.

An equivalent formulation is

$$P(x, y, z) = P(x, z)P(y, z)/P(z)$$

which shows that conditional independence can be rephrased entirely in terms of marginal densities.

Like with marginal independence we can state a simple factorisation criterion to establish conditional independence: random vectors X and Y are

conditionally independent given Z, $X \perp\!\!\!\perp Y \mid Z$ if and only if there exist functions g and h such that

$$P(x, y, z) = g(x, z)h(y, z)$$

for all (x, y) and for all z for which $P(z) > 0$. Again we will often use the "log-version"

$$\log P(x, y, z) = g'(x, z) + h'(y, z)$$

4 Independence Graphs

We can represent the conditional independence relations between a set of random variables in a so-called conditional independence graph. Let $X = (X_1, X_2, \ldots, X_k)$ be a k-dimensional random vector. The conditional independence graph of X is the undirected graph $G = (K, E)$, with $K = \{1, 2, \ldots, k\}$, and where $\{i, j\}$ is *not* in the edge set E iff

$$X_i \perp\!\!\!\perp X_j | \text{rest}$$

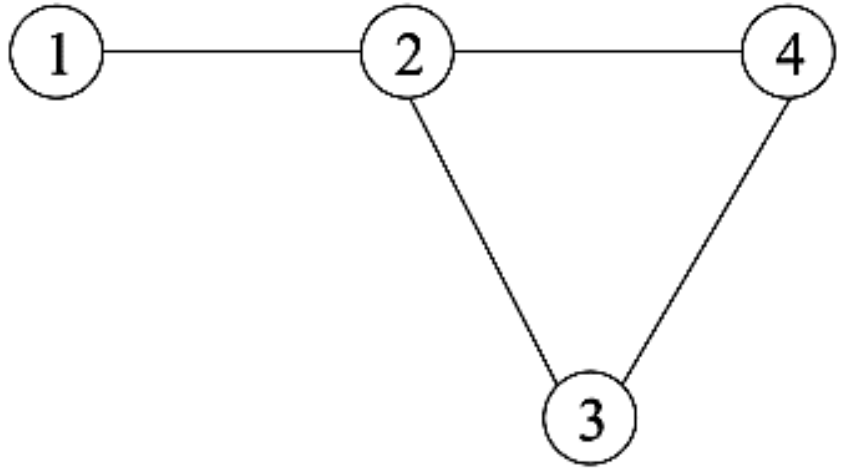
Example: Let $X = (X_1, X_2, X_3, X_4), 0 < x_i < 1$ with probability density

$$P(x) = \exp(u + x_1 + x_1 x_2 + x_2 x_3 x_4)$$

Application of the factorisation criterion gives

$$X_1 \perp\!\!\!\perp X_4 | (X_2, X_3) \text{ and } X_1 \perp\!\!\!\perp X_3 | (X_2, X_4)$$

Hence the conditional independence graph of X is



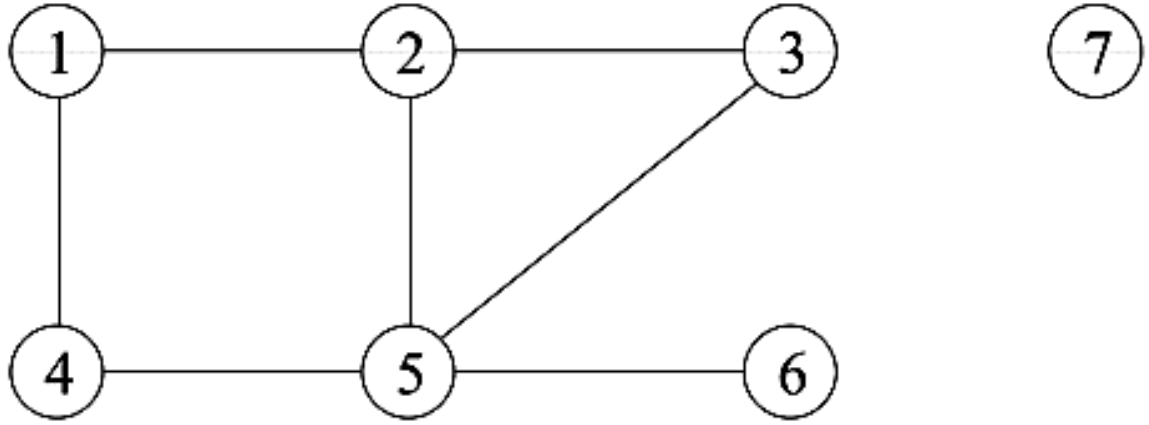


Figure 1: Example of a conditional independence graph

In fact we can reduce the conditioning set by using the concept of separation.

From the conditional independence graph in figure 1 we can read that

$$X_1 \perp\!\!\!\perp X_3 | (X_2, X_4, X_5, X_6, X_7)$$

However, since $\{2, 5\}$ separates 1 from 3 in the graph (i.e. every path from 1 to 3 must go through 2 or 5), we can make the stronger statement

$$X_1 \perp\!\!\!\perp X_3 | (X_2, X_5)$$

We defined the conditional independence graph using the rule that for all non-adjacent vertices i and j

$$X_i \perp\!\!\!\perp X_j | \text{rest}$$

This is called the pairwise Markov property. Perhaps surprisingly, the following properties turn out to be equivalent

Global Markov property: a separates b from c (a, b, c disjoint) iff

$$X_b \perp\!\!\!\perp X_c | X_a$$

where $X_a = (X_i; i \in a)$ and a separates b from c if for all $i \in b, j \in c$: a separates i from j.

Local Markov property:

$$X_i \perp\!\!\!\perp \text{ rest} \mid \text{boundary}(i)$$

where the boundary of a vertex i is simply the set of adjacent vertices.

The local Markov property is particularly relevant to prediction. For example, to predict X_2 in figure 1, we only need to know the values of X_1, X_3 and X_5.

5 Log-linear Models

In this section we introduce the class of log-linear models and its subclasses of hierarchical, graphical log-linear, and finally decomposable models. For ease of exposition we start with log-linear models for binary variables.

5.1 Log-linear models for binary data

A random experiment that only distinguishes between two possible outcomes is called a *Bernoulli* experiment. The outcomes are usually referred to as *success* and *failure* respectively. We define a random variable X that denotes the number of successes in a Bernoulli experiment; X consequently has possible values 0 and 1. The probability distribution of X is completely determined by the probability of success, which we denote by p, and is: $P(X = 0) = 1-p$ and $P(X = 1) = p$.

A Bernoulli random variable X, has the probability density function

$$P(x) = p^x(1-p)^{1-x} \quad \text{for } x = 0, 1 \text{ and } 0 \leq p \leq 1$$

This is a clever way of writing the probability density in one formula; check that indeed $P(1) = p$ and $P(0) = 1 - p$ as required.

Next we consider the analysis of a 2×2 table. The bivariate Bernoulli random vector (X_1, X_2), takes the values $(0,0)$, $(0,1)$,$(1,0)$ and $(1,1)$ in the Cartesian product $\{0,1\} \times \{0,1\}$. Its distribution is completely specified by the table of probabilities

$P(x_1, x_2)$		$x_2 = 0$	$x_2 = 1$	Total
	$x_1 = 0$	$p(0,0)$	$p(0,1)$	$p_1(0)$
	$x_1 = 1$	$p(1,0)$	$p(1,1)$	$p_1(1)$
	Total	$p_2(0)$	$p_2(1)$	1

Here's a clever way to write the probability distribution as one function:

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

for $x_1 = 0, 1$ and $x_2 = 0, 1$. Verify that $P(0,0) = p(0,0)$, $P(0,1) = p(0,1)$, $P(1,0) = p(1,0)$ and $P(1,1) = p(1,1)$ as required.

Taking logarithms of this identity for P, and collecting terms in x_1 and x_2 gives

$$\begin{aligned}\log P(x_1, x_2) &= \log p(0,0) + x_1 \log \frac{p(1,0)}{p(0,0)} + \\ &\quad x_2 \log \frac{p(0,1)}{p(0,0)} + x_1 x_2 \log \frac{p(1,1)p(0,0)}{p(0,1)p(1,0)}\end{aligned}$$

Exercise: Verify this result using the following properties of logarithms:

$$\log a^b = b \log a$$
$$\log ab = \log a + \log b$$
$$\log \frac{a}{b} = \log a - \log b$$

Reparameterizing the right hand side leads to the so-called *log-linear expansion*

$$\log P(x_1, x_2) = u_\emptyset + x_1 u_1 + x_2 u_2 + x_1 x_2 u_{12} \quad \text{for } (x_1, x_2) \text{ in } \{0,1\}^2$$

The coefficients, $u_\emptyset, u_1, u_2, u_{12}$ are known as the u-terms. For example

$$u_1 = \log \frac{p(1,0)}{p(0,0)}$$

which is just the log of the odds of the event $X_1 = 1$ to the event $X_1 = 0$ conditioned on $X_2 = 0$. The coefficient of the product $x_1 x_2$ is the logarithm of the cross product ratio

$$u_{12} = \log \frac{p(1,1)p(0,0)}{p(0,1)p(1,0)} = \log \operatorname{cpr}(X_1, X_2)$$

The random variables X_1 and X_2 are independent if and only if $u_{12} = 0$. The factorisation criterion states that X_1 and X_2 are independent iff there exist two functions g and h such that

$$\log P(x_1, x_2) = g(x_1) + h(x_2) \text{ for all } (x_1, x_2)$$

If $u_{12} = 0$, we can take $g(x_1) = u_\emptyset + x_1 u_1$ and $h(x_2) = u_2 x_2$. If $u_{12} \neq 0$ no such decomposition is possible.

To calculate systematically the u's from the given p's, substitute in the log-linear expansion for $(x_1, x_2) = (0,0), \ldots, (1,1)$ to get

$$
\begin{array}{lcl}
 & & 12 \\
\log p(0,0) & = & u_\emptyset \\
\log p(1,0) & = & u_\emptyset + u_1 \\
\log p(0,1) & = & u_\emptyset + u_2 \\
\log p(1,1) & = & u_\emptyset + u_1 + u_2 + u_{12}
\end{array}
$$

This is a simple set of linear equations to solve.

The log-linear expansion of a $2 \times 2 \times 2$ table (three dimensional Bernoulli) is obtained in a similar way. The density function can be written

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

The log-linear expansion is

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

Note that for example

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

In general, we can enforce (conditional) independence constraints, by setting the right u-terms to zero.

5.2 Extension to non-binary data

So far we assumed all variables are binary. In general we allow discrete variables with more than two values as well. To see how we can generalise the log-linear model to this case, consider again the 2×2 table

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

for $x \in \{0, 1\}^2$. What if the x_i have more than two levels? The trick is to make the u-terms *functions* of x rather than constants:

$$\log P(x_1, x_2) = u_\emptyset + u_1(x_1) + u_2(x_2) + u_{12}(x_1, x_2) \tag{2}$$

In fact we now have too many parameters, and in order to identify them we have to impose some extra constraints. To be consistent with the binary

case, we impose the constraint that $u_a(x_a) = 0$ whenever $x_i = 0$ and $i \in a$. Here we assume that if x_i has d_i possible values, these are numbered

$0, 1, \ldots, d_i - 1$. Note however that this numbering does not imply any ordering of the values.

So for example, suppose x_1 has two possible values (0,1) and x_2 has three possible values (0,1,2) then the following u-terms are constrained to be zero

$$u_1(0) = 0, u_2(0) = 0, u_{12}(0,1) = u_{12}(0,2) = u_{12}(1,0) = u_{12}(0,0) = 0$$

5.3 Hierarchical and Graphical Log-linear models

Definition 1 (Log-linear expansion) *The log-linear expansion of the cross-classified Multinomial density function 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 $K = \{1, 2, \ldots, k\}$ and where the u-terms $\{u_a\}$ are coordinate projection functions, so that $u_a(x) =$

$u_a(x_a)$, and also satisfy the constraint that $u_a(x) = 0$ whenever $x_i = 0$ and $i \in a$.

It is particularly easy to list the conditions for independence as conditions on u-terms.

Proposition 1 (Independence and the u-terms) *If (X_a, X_b, X_c) is a partitioned Multinomial random vector then $X_b \perp\!\!\!\perp X_c | X_a$ if and only if all u-terms in the log-linear expansion with one or more coordinate in b and one or more coordinate in c, are zero.*

The proof is a direct application of the factorisation theorem for conditional independence. Let t be an arbitrary subset of $a \cup b \cup c = \{1, 2, \ldots, k\}$. If all u-terms, u_t, are zero whenever $t \not\subseteq a \cup b$ and $t \not\subseteq a \cup c$ (i.e. whenever t contains coordinates from both b and c) then we can write

$$\log P_K = \sum_{t \subseteq a \cup b} u_t + \sum_{t \subseteq a \cup c} u_t - \sum_{t \subseteq a} u_t$$

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

The importance of the log-linear expansion rests in the fact that many

interesting hypotheses can be generated by setting u-terms to zero. Proposition 1 gives conditions on the u-terms for conditional independence.

Model	Omitted	Interpretation
123	none	saturated
12,13,23	u_{123}	
12,13	u_{123}, u_{23}	
12,23	u_{123}, u_{13}	
13,23	u_{123}, u_{12}	

12,3	u_{123}, u_{13}, u_{23}
13,2	u_{123}, u_{12}, u_{23}
23,1	u_{123}, u_{12}, u_{13}
1,2,3	$u_{123}, u_{12}, u_{13}, u_{23}$

homogeneous association

$$X_2 \perp\!\!\!\perp X_3 \mid X_1$$

$X_1 \perp\!\!\!\perp X_3 \mid X_2$
$X_1 \perp\!\!\!\perp X_2 \mid X_3$
$(X_1, X_2) \perp\!\!\!\perp X_3$
$(X_1, X_3) \perp\!\!\!\perp X_2$
$(X_2, X_3) \perp\!\!\!\perp X_1$
mutual independence

Table 8: All hierarchical models with 3 variables

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 also present. A log-linear model is said to be hierarchical if the presence of a term implies that all lower-order terms that are contained in it are also present. This implies that a hierarchical model is identified by listing its highest order interaction terms.

In table 8 we give all hierarchical models for three dimensions

Definition 2 (Graphical Model) *Given an independence graph $G = (K, E)$, the cross-classified Multinomial distribution 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 of G, the u-terms containing the selected coordinates are identically zero.*

More explicitly, the density of a Multinomial graphical model is

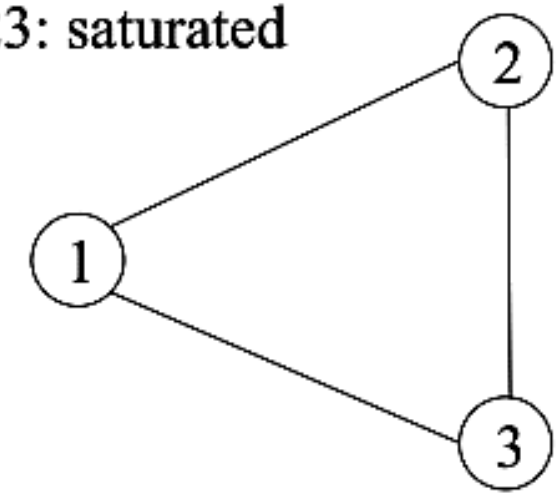
$$\log P_K(x) = \sum_{a \subseteq K} u_a(x_a)$$

subject to the constraints that $u_a = 0$ if $\{i, j\} \subseteq a$ and (i, j) is not in the edge set E. The parameters of the graphical model are the remaining u-terms that are not set to zero.

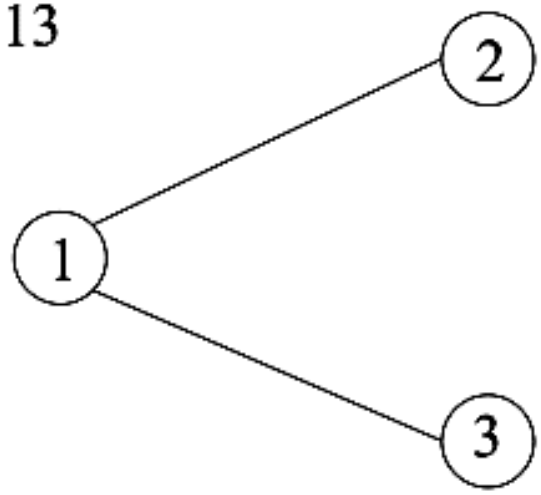
In figure 2 we show four hierarchical models and their independence graphs. Note that the saturated model and the homogeneous association model have the same independence graph. The homogeneous association

model is *not* a graphical model however, because it imposes the additional constraint that $u_{123} = 0$. In fact the homogeneous association model is the only hierarchical model in 3 dimensions that is not graphical.

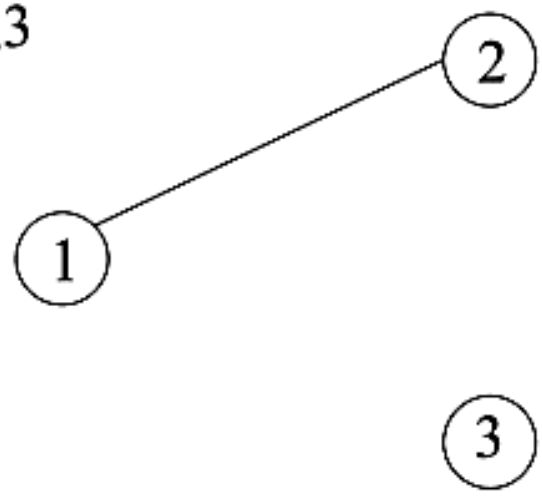
123: saturated



12,13



12,3



12, 13, 23:
not graphical!

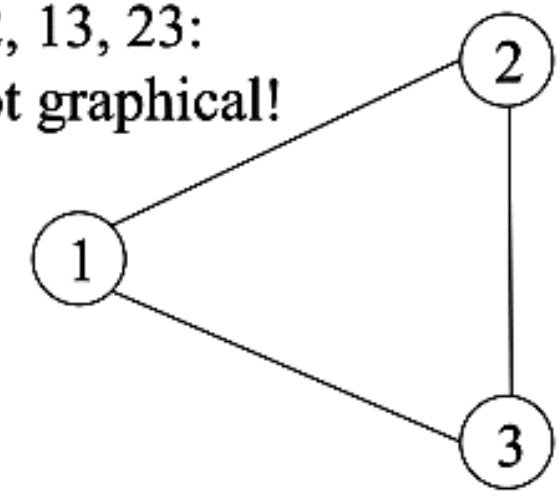


Figure 2: Four hierarchical models and their independence graphs

6 Maximum Likelihood Estimation of Hierarchical and Graphical Models

The maximum likelihood estimator of graphical log-linear model M satisfies the likelihood equations

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

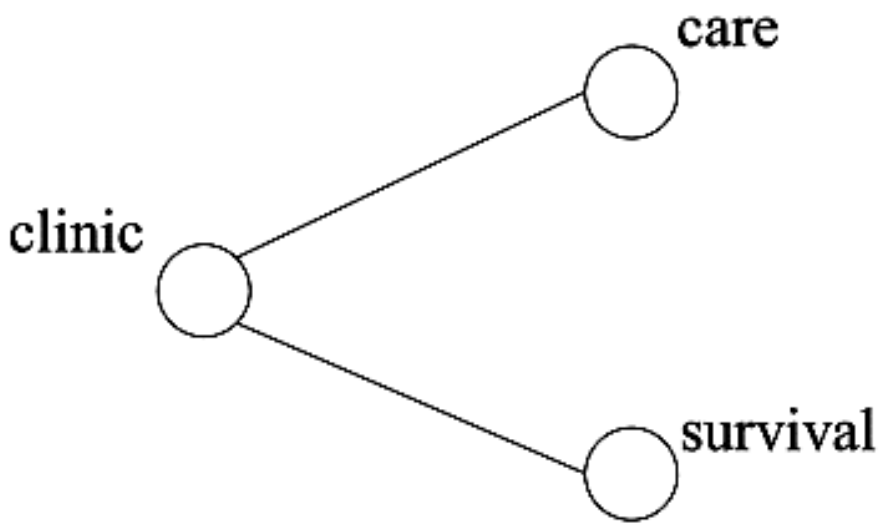
whenever the subset of vertices a in the graph form a clique. This is summarized by the slogan: "Observed = Fitted" for every marginal table corresponding to a complete subgraph.

Likewise, the maximum likelihood estimator of hierarchical log-linear model M satisfies the likelihood equations

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

whenever a belongs to the highest order interaction terms of M.

As an example, we return to the infant survival data. We saw that the model



seemed to give a pretty good representation of the data at first sight. Let's fit this model to the data

n_{123}		survival	
clinic	care	no	yes
clinic 1	less	3	176

	more	4	293
clinic 2	less	17	197
	more	2	23

We number the variables as follows: 1=clinic, 2=care, 3=survival. Then the cliques in the graph are 12 and 13, and so the sufficient statistics are n_{12} and n_{13}. Hence the maximum likelihood estimate satisfies the equations

$$\begin{aligned} \hat{n}_{12} = N\hat{P}_{12} &= n_{12} \\ \hat{n}_{13} = N\hat{P}_{13} &= n_{13} \end{aligned}$$

The relevant tables are given below

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

Writing $\hat{n}$ for $N\hat{P}$ we get

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

which gives the fitted values:

$\hat{n}_{123}$		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

The model seems to fit very well indeed.

6.1 Iterative Proportional Fitting

Not all (hierarchical) log-linear models have closed form maximum likelihood estimates as in the previous example. There is however a simple iterative algorithm called Iterative Proportional Fitting (IPF) that will converge to those estimates. We start by giving a simple example that actually does not require IPF. Suppose we want to fit the 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

The minimal sufficient statistics are row totals $n_1(x_1)$ and column totals $n_2(x_2)$. In other words, the ML estimates satisfy the equations

$$\begin{aligned} \hat{n}_1(x_1) &= n_1(x_1) \\ \hat{n}_2(x_2) &= n_2(x_2) \end{aligned}$$

This gives the closed form estimates

$$\hat{n}_{12}(x) = n_1(x_1)n_2(x_2)/N$$

Application of this formula gives the following table of fitted values

$\hat{n}(x_1, x_2)$		$x_2 = 0$	$x_2 = 1$	$n_1(x_1)$
	$x_1 = 0$	24	16	40
	$x_1 = 1$	36	24	60
	$n_2(x_2)$	60	40	100

We will now show how we arrive at this solution using IPF. We usually begin with a table $\hat{n}^{(0)}$ of uniform counts

1	1	2
1	1	2

In the first step we fit to the row margin

$$\hat{n}(x)^{(1)} = \hat{n}(x)^{(0)} \times$$

We compute

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

and

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

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

$$\frac{n_1(x_1)}{\hat{n}_1(x_1)^{(0)}}$$

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

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

$$\begin{array}{|cc|c} \hline 20 & 20 & 40 \\ 30 & 30 & 60 \\ \hline \end{array}$$

In the second step we fit to the column margin

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

Which gives

$$\hat{n}(0,0)^{(2)} = 20 \times \frac{60}{50} = 24 \qquad \hat{n}(0,1)^{(2)} = 20 \times \frac{40}{50} = 16$$

and

$$\hat{n}(1,0)^{(2)} = 30 \times \frac{60}{50} = 36 \qquad \hat{n}(1,1)^{(2)} = 30 \times \frac{40}{50} = 24$$

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

24	16
36	24
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 has the nice property that if there is an explicit formula for the ML estimates, then the algorithm will reach these values within one iteration, i.e. each margin has to be fit only once. In case there is no closed-form solution more iterations are required. Why did we start the procedure from a uniform table of counts? The point is we have to start with a table that satisfies all constraints imposed by the log-linear model. In our example, we were fitting the independence model

$$\log P(x_1, x_2) = u_0 + u_1 x_1 + u_2 x_2$$

The uniform table of counts satisfies this model with $u_1 = 0$, $u_2 = 0$ and $u_0 = \log 1/4$. In fact the uniform table sets all u terms to zero except for

u_0 which has the value $\log 1/N$. So as long as the model does not set u_0 to zero (and no acceptable model does), the uniform table satisfies the model constraints. Now if the log-linear model constrains a particular u-term to be zero, then the steps of the IPF algorithm will not violate this constraint. For example, in the independence model we set

$$u_{12} = \log \text{cpr}(X_1, X_2) = 0$$

In other words, $\text{cpr}(X_1, X_2) = 1$. Now the uniform table obviously satisfies this constraint (recall the definition of the cross-product ratio). A proportional adjustment of a row or column does not change the cross-product ratio since

$$\frac{\hat{n}(0,0)\hat{n}(1,1)}{\hat{n}(0,1)\hat{n}(1,0)} = \frac{c\,\hat{n}(0,0)\hat{n}(1,1)}{c\,\hat{n}(0,1)\hat{n}(1,0)}$$

for any value of $c \neq 0$. Hence we had to start with a table with $\text{cpr} = 1$, to get a solution for which this is also the case.

We now consider a slightly more complicated example in 3 dimensions. The only hierarchical model with 3 variables that does not have a closed from solution is the so called homogeneous association model with highest

order interaction terms: 12,13,23. IPF proportionally adjusts the estimated expected frequencies $\hat{n}_{123}(x)$ to in turn 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) $\hat{n}_{23}(x_2, x_3) = n_{23}(x_2, x_3)$

One iteration of IPF for this model looks like this.
Fit to 12 margin:

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

Fit to 13 margin:

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

Fit to 23 margin:

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

In the first step we make sure the fitted 12 margin is equal to the observed 12 margin. In the second step we do the same for the 13 margin. This may disrupt the result op the previous step. In the third step we fit to the 13 margin. These three steps are repeated until all three fitted margins are equal to the observed margins simultaneously.

Finally we give a sketch of the general IPF algorithm. 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

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

fitting $\hat{n}(x)$ to each margin in turn. For each margin a we apply the IPF updating rule

$$\hat{n}_{ab}(x_a, x_b)^{(t+1)} = n_a(x_a) \left(\frac{\hat{n}_{ab}(x_a, x_b)^{(t)}}{\hat{n}_a(x_a)^{(t)}} \right)$$

where b is the complement of a. We keep cycling through the margins until convergence is reached. It is easy to show that after fitting to margin a, we indeed have

$$\hat{n}_a(x_a)^{(t+1)} = n_a(x_a)$$

Proof:

$$\hat{n}_a(x_a)^{(t+1)} =$$

$$=$$

$$\sum_{x_b} \hat{n}_{ab}(x_a, x_b)^{(t+1)}$$

$$\sum_{x_b} \left(\frac{\hat{n}_{ab}(x_a, x_b)^{(t)}}{\hat{n}_a(x_a)^{(t)}} \right) n_a(x_a)$$

$$\sum_{x_b} \left(\frac{\hat{n}_{ab}(x_a, x_b)^{(t)}}{\sum_{x_b} \hat{n}_{ab}(x_a, x_b)^{(t)}} \right) n_a(x_a)$$

$n_a(x_a)$

7 Decomposable Graphical Models

Decomposable models are graphical models that have explicit formulas for the maximum likelihood estimates. This is a convenient property from a computational viewpoint. If we only have to fit one model this is perhaps not so important, but when we have little prior knowledge we typically have to search a potentially large space of possible models.

Decomposable models are very easy to characterize by their independence graphs. They have *triangulated* independence graphs: their independence graphs have no *chordless cycles* of length greater than three. A cycle is called chordless if no other than successive pairs of vertices in the cycle are adjacent. The graphs in figure 7 are *not* decomposable because they have chordless 4-cycles.

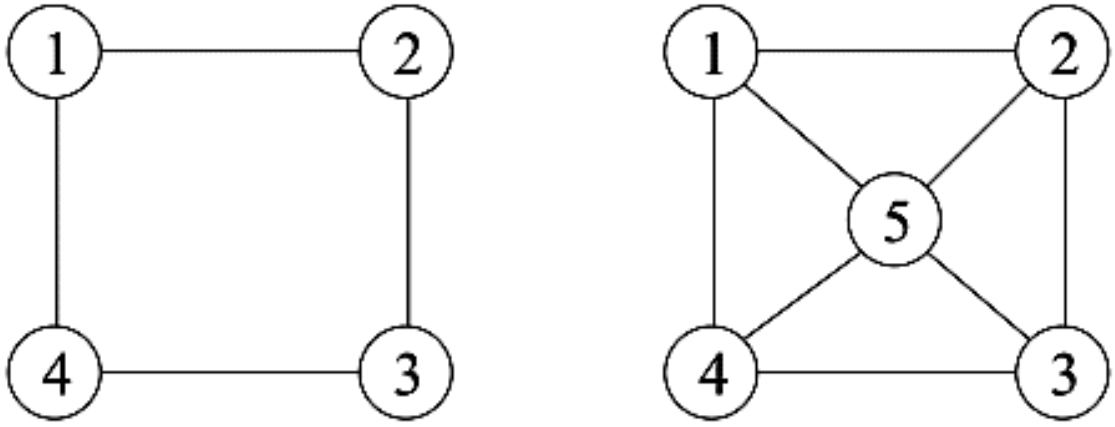


Figure 3: Two graphs with chordless 4-cycles

8 Deviance and Likelihood Ratio Test

The deviance of a fitted model compares the log-likelihood of the fitted model to the log-likelihood of the saturated model. The larger the model deviance, the poorer the fit. The likelihood of a model M is

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

where $\hat{P}(x)$ are the ML estimates of the cell probabilities for model M. This is of course just the probability of the data given $\hat{P}$.

Consequently, the log-likelihood of a model M is

$$\sum_x n(x) \log \hat{P}(x)$$

For example, suppose we have the following table of observed counts:

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

We have already seen that the independence model gives estimates

$$\hat{P}(0,0) = 0.24,\ \hat{P}(0,1) = 0.16,\ \hat{P}(1,0) = 0.36,\ \hat{P}(1,1) = 0.24$$

So the probability of the observed table for this model is

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

The log-likelihood is

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

Since for the saturated model

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

the log-likelihood of the saturated model is

$$\sum_x n(x) \log \frac{n(x)}{N}$$

So for the saturated model the log-likelihood value is

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

The log-likelihood value of the saturated model is of course always higher than for any other model. The saturated model gives the best possible fit.

The deviance of M is twice the difference between the log-likelihood of the saturated model and the log-likelihood of M, i.e.

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

which can be summarised by the *slogan*

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

The deviance of the independence model in the previous example is

$$\text{dev}(\text{independence model}) = 2(-131.4 + 134.6) = 6.4$$

Let

$$\mathcal{L}^i = \mathcal{L}(\hat{P}^{M_i})$$

be the value of the log-likelihood function evaluated at $\hat{P}^{M_i}$; the ML estimates of P under M_i. Let $M_0 \subseteq M_1$; i.e. M_0 can be obtained from M_1 by imposing

additional restrictions (setting additional u-terms to zero). The deviance difference between M_0 and M_1 is

$$\text{dev}(M_0) - \text{dev}(M_1) = -2\mathcal{L}^0 + 2\mathcal{L}^1 = 2(\mathcal{L}^1 - \mathcal{L}^0)$$

We state without proof that for large N

$$2(\mathcal{L}^1 - \mathcal{L}^0) \approx_{M_0} \chi^2_\nu$$

where the degrees of freedom ν is equal to the number of additional restrictions of M_0. This result will be the basis for subsequent model testing. We reject the null hypothesis that M_0 is the true model when

$$2(\mathcal{L}^1 - \mathcal{L}^0) > \chi^2_{\nu;\alpha}$$

Remark 1 *The test is called a likelihood ratio test because we are looking at logs, and*

$$\log \frac{L^1}{L^0} = \log L^1 - \log L^0 = \mathcal{L}^1 - \mathcal{L}^0$$

We show how the likelihood ratio test can be used to test whether a model gives an adequate fit of the data. Does

$$\text{survival} \perp\!\!\!\perp \text{care} \mid \text{clinic} \tag{3}$$

give a good fit of the observed table? To test this we perform a likelihood ratio test against the saturated model. We fit the model and compute the deviance:

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

Now we have to determine the appropriate degrees of freedom for the test. Since (3) imposes two additional constraints (two u-terms to zero) compared to the saturated model, we compute

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

Since the deviance difference is not significant at the 5% level, we accept model (3).

Does the mutual independence model give a good fit of the observed

table? Compute

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

Now, since

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

we reject the mutual independence model at the 5% level.

9 Fitting Hierarchical Loglinear Models in R

Our preferred data analysis system `R` contains a function called `loglin` for fitting hierarchical loglinear models. To specify the model you want to fit, you have to list the highest order interaction terms. Here's the clinic example in `R`:

```
> a <- array(c(3,17,4,2,176,197,293,23),dim=c(2,2,2),
      dimnames=list(c("clinic 1","clinic 2"),
                    c("less","more"),c("no","yes")))
```

```
> a <- as.table(a)
> names(dimnames(a)) <-

> a
, , survival = no
```

```
                care
clinic          less more
  clinic 1       3    4
  clinic 2      17    2

, , survival = yes
```

	care	
clinic	less	more
clinic 1	176	293
clinic 2	197	23

We entered the data as

```
c("clinic","care","survival")
```

an array. The array data is given as a single vector, with the leftmost subscript moving fastest. Since the function `loglin` expects a table rather than an array, we convert it to a table. Finally, we add the variable names and print the data. We start by fitting the model where care and survival are independent given clinic:


```
> model.1 <- loglin(a,margin=list(c("clinic","care"),c("clinic","survival")),
                    fit=TRUE)
2 iterations: deviation 0
> model.1
$lrt
[1] 0.08228918

$pearson
```

```
[1] 0.08361853

$df
[1] 2

$margin
$margin[[1]]
[1] "clinic" "care"

$margin[[2]]
[1] "clinic"   "survival"


$fit
, , survival = no

          care
clinic            less       more
  clinic 1    2.632353   4.367647
  clinic 2   17.012552   1.987448

, , survival = yes

          care
```

```
clinic           less       more
  clinic 1 176.367647 292.632353
  clinic 2 196.987448  23.012552
```


The first argument we pass to loglin is the table with observed counts. The second argument specifies the model that has to be fitted by giving the list of highest order interaction terms. The call to loglin returns a list with a number of components. The component `lrt` gives the likelihood ratio test statistic (model deviance), and the component `df` gives the appropriate degrees of freedom (number of u-terms set to zero). Since in the call we specified `fit = TRUE`, the table with the fitted counts is also returned.

As a second example, we fit the independce model:

```
> model.2 <- loglin(a,margin=list(c("clinic"),c("care"),c("survival")),
                    fit=TRUE)
2 iterations: deviation 3.552714e-15
> model.2
$lrt
[1] 211.4820
```

```
$pearson
[1] 199.6457

$df
[1] 4

$margin
$margin[[1]]
[1] "clinic"

$margin[[2]]
[1] "care"

$margin[[3]]
[1] "survival"


$fit
, , survival = no

          care
```

```
clinic           less       more
  clinic 1   9.513948   7.795143
  clinic 2   4.776961   3.913948

, , survival = yes

          care
clinic           less       more
  clinic 1 252.119619 206.571291
  clinic 2 126.589472 103.719619
```

We observed from the output that the deviance of the independence model is 211.482. To perform the appropriate test in R, we can find the critical value for $\alpha = 0.05$ as follows:

```
> qchisq(0.05,df=4,lower.tail=F)
```

```
[1] 9.487729
> qchisq(0.95,df=4)
[1] 9.487729
```

The function `qchisc` gives the value of the test statistic for which $P(X^2 < c) = \alpha$ where X^2 is a random variable with chi-square distribution with `df` degrees of freedom. Since we actually want the value for which $P(X^2 > c) = \alpha$, we can either specify this explicitely, or pass $1 - \alpha$ instead of α to the function.

10 Model Selection

In the previous section we have shown how to fit a single hierarchical loglinear model in R. To get a data mining algorithm, all you have to do is superimpose some search strategy to search the model space. You also need a way to measure model quality.

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 of the model. This quality measure consists of two components: the lack-of-fit of the model as measured by the deviance, and the complexity of the model as measured by the number of parameters (i.e. the number of u-terms not constrained to be equal to zero). Notice the analogy with the total cost of a tree in cost-complexity pruning. By including the penalty for complexity we try to avoid overfitting. If we did not include this penalty term the saturated model would always win. Now it is possible that we prefer a simpler model that has a worse fit, over a more complex model. We give an example of stepwise search with AIC. To begin with, we fit a loglinear model that will be used as the initial model from which the search starts. We use a frontend to `loglin` available in the library `MASS`:

```
> library(MASS)
> model.init <- loglm( ~ clinic + care + survival,data=a)
> model.init
Call:
loglm(formula = ~clinic + care + survival, data = a)
```

```
Statistics:
                     X^2 df P(> X^2)
Likelihood Ratio 211.4820  4        0
Pearson          199.6457  4        0
```

The `loglm` function actually calls the function `loglin` that we used before, but allows (or requires, depending on your preference) you to specify the model differently. The first argument is a formula where on the right hand side of the tilde, you specify the highest order interaction terms. For example, to fit the homogeneous association model, the call should be:

```
> model.6 <- loglm( ~ clinic*care+clinic*survival+care*survival,data=a)
> model.6
Call:
loglm(formula = ~clinic*care + clinic*survival + care*survival,
    data = a)
```



```
Statistics:
```

```
                        X^2 df  P(> X^2)
Likelihood Ratio 0.04334249  1 0.8350817
Pearson          0.04410757  1 0.8336536
```

The reason we use `loglm` rather than `loglin` is that the stepwise search performed by `stepAIC` requires the format returned by `loglm`. Here we use `stepAIC` to search the model space:

```
> model.step <- stepAIC(model.init,scope=
Start:  AIC=219.48
~clinic + care + survival

                    Df    AIC
+ clinic:care        1  27.83
```

```
+ clinic:survival   1 203.74
+ care:survival     1 215.87
<none>                219.48
- care              1 224.54
- clinic            1 297.55
- survival          1 985.30

Step:  AIC=27.83
~clinic + care + survival + clinic:care

                    Df    AIC
+ clinic:survival   1   12.08
+ care:survival     1   24.22
```

```
<none>                    27.83
- clinic:care       1 219.48
- survival          1 793.65

Step:  AIC=12.08
~clinic + care + survival + clinic:care +

                   Df     AIC
<none>                 12.082
```

```
~ clinic*care*survival)
```