More about Structure Estimation

Lecture 9
Saint Flour Summerschool, July 17, 2006

Steffen L. Lauritzen, University of Oxford

Overview of lectures

1. Conditional independence and Markov properties
2. More on Markov properties
3. Graph decompositions and junction trees
4. Probability propagation and related algorithms
5. Log-linear and Gaussian graphical models
6. Hyper Markov laws
7. More on Hyper Markov Laws
8. Structure estimation and Bayes factors
9. *More on structure estimation.*

Types of approach

- Methods for *judging adequacy of structure* such as
  - Tests of significance
  - Penalised likelihood scores

    $$I_\kappa(\mathcal{G}) = \log \hat{L} - \kappa \dim(\mathcal{G})$$

    with $\kappa = 1$ for AIC Akaike (1974), or $\kappa = \frac{1}{2} \log n$ for BIC (Schwarz 1978).
  - *Bayesian posterior probabilities.*
- *Search strategies* through space of possible structures, more or less based on *heuristics.*

Estimating trees

Assume P factorizes w.r.t. an unknown *tree* $\mathcal{T}$. *MLE* $\hat{\tau}$ *of* $\mathcal{T}$ *has maximal weight*, where the *weight* of τ is

$$w(\tau) = \sum_{e \in E(\tau)} w_n(e) = \sum_{e \in E(\tau)} H_n(e)$$

and $H_n(e)$ is the empirical *cross-entropy* or *mutual information* between endpoint variables of the edge $e = \{u, v\}$. For *Gaussian trees* this becomes

$$w_n(e) = -\frac{1}{2} \log(1 - r_e^2),$$

where r_e^2 is *correlation coeffient* along edge $e = \{u, v\}$.

Highest AIC or BIC scoring forest *also available as MWSF,* with modified weights

$$w_n^{\text{pen}}(e) = nw_n(e) - \kappa_n \text{df}_e,$$

with $\kappa_n = 1$ for AIC, $\kappa_n = \frac{1}{2}\log n$ for BIC and df_e the *degrees of freedom for independence* along e.

Use *maximal weight spanning tree* (or forest) algorithm from weights $W = (w_{uv}, u, v \in V)$.

Bayes factors

For $\mathcal{G} \in \Gamma$, $\Theta_{\mathcal{G}}$ is associated parameter space so that P factorizes w.r.t. $\mathcal{G}$ if $P = P_\theta$ for some $\theta \in \Theta_{\mathcal{G}}$. $\mathcal{L}_{\mathcal{G}}$ is prior law on $\Theta_{\mathcal{G}}$.

The *Bayes factor* for discriminating between $\mathcal{G}_1$ and $\mathcal{G}_2$ based on $X^{(n)} = x^{(n)}$ is

$$\mathrm{BF}(\mathcal{G}_1 : \mathcal{G}_2) = \frac{f(x^{(n)} \,|\, \mathcal{G}_1)}{f(x^{(n)} \,|\, \mathcal{G}_2)},$$

where

$$f(x^{(n)} \,|\, \mathcal{G}) = \int_{\Theta_{\mathcal{G}}} f(x^{(n)} \,|\, \mathcal{G}, \theta)\, \mathcal{L}_{\mathcal{G}}(d\theta)$$

is known as the *marginal likelihood* of $\mathcal{G}$.

Posterior distribution over graphs

If $\pi(\mathcal{G})$ is a prior probability distribution over a given set of graphs Γ, the posterior distribution is determined as

$$\pi^*(\mathcal{G}) = \pi(\mathcal{G} \mid x^{(n)}) \propto f(x^{(n)} \mid \mathcal{G})\pi(\mathcal{G})$$

or equivalently

$$\frac{\pi^*(\mathcal{G}_1)}{\pi^*(\mathcal{G}_2)} = \mathrm{BF}(\mathcal{G}_1 : \mathcal{G}_2)\frac{\pi(\mathcal{G}_1)}{\pi(\mathcal{G}_2)}.$$

The *BIC is an $O(1)$-approximation to* $\log \mathrm{BF}$ using Laplace's method of integrals on the marginal likelihood.

Bayesian analysis looks for the *MAP estimate* $\mathcal{G}^*$ maximizing $\pi^*(\mathcal{G})$ over Γ, or attempts to *sample from the posterior* using e.g. Monte-Carlo methods.

Hyper inverse Wishart laws

Denote the normalisation constant of the hyper inverse Wishart density as

$$h(\delta, \Phi; \mathcal{G}) = \int_{\mathcal{S}^+(\mathcal{G})} (\det K)^{\delta/2} e^{-\operatorname{tr}(K\Phi)}\, dK,$$

The marginal likelihood is then

$$f(x^{(n)} \,|\, \mathcal{G}) = \frac{h(\delta + n, \Phi + W^n; \mathcal{G})}{h(\delta, \Phi; \mathcal{G})}.$$

where

$$h(\delta, \Phi; \mathcal{G}) = \frac{\prod_{Q \in \mathcal{Q}} h(\delta, \Phi_Q; \mathcal{G}_Q)}{\prod_{S \in \mathcal{S}} h(\delta, \Phi_S; S)^{\nu_{\mathcal{G}}(S)}}.$$

For *chordal graphs* all terms reduce to known Wishart constants.

In general, Monte-Carlo simulation or similar methods must be used (Atay-Kayis and Massam 2005).

Bayes factors for forests

Trees and forests are decomposable graphs, so for a forest ϕ we get

$$\begin{aligned}\pi^*(\phi) &\propto \frac{\prod_{e\in E(\phi)} f(x_e^{(n)})}{\prod_{v\in V} f(x_v^{(n)})^{d_\phi(v)-1}} \\ &\propto \prod_{e\in E(\phi)} \mathrm{BF}(e),\end{aligned}$$

where $\mathrm{BF}(e)$ is the *Bayes factor* for independence along the edge e:

$$\mathrm{BF}(e) = \frac{f(x_u^{(n)}, x_v^{(n)})}{f(x_u^{(n)})f(x_v^{(n)})}.$$

MAP estimates of forests can thus be computed using an MWSF algorithm, using $w(e) = \log BF(e)$ as weights.

When ϕ is restricted to contain a *single tree*, the normalization constant can be explicitly obtained via the *Matrix Tree Theorem,* see e.g. Bollobás (1998).

Algorithms exist for generating random spanning trees (Aldous 1990), so *full posterior analysis is in principle possible for trees*.

Only heuristics available for MAP estimators or maximizing penalized likelihoods such as AIC or BIC, for other than trees.

Directed hyper Markov property

$\mathcal{L} = \mathcal{L}(\theta)$ is *directed hyper Markov* w.r.t. a DAG $\mathcal{D}$ if θ is directed Markov on $\mathcal{D}$ for all $\theta \in \Theta$ and

$$\theta_{v \mid \mathrm{pa}(v)} \perp\!\!\!\perp_{\mathcal{L}} \theta_{\mathrm{nd}(v)} \,|\, \theta_{\mathrm{pa}(v)}.$$

A law $\mathcal{L}$ is directed hyper Markov on $\mathcal{D}$ if and only if $\mathcal{L}_A$ is hyper Markov on $(\mathcal{D}_A)^m$ for any ancestral set $A \subseteq V$.

$\mathcal{L}$ is *strongly directed hyper Markov* if in addition $\theta_{v \mid \mathrm{pa}(v)} \perp\!\!\!\perp_{\mathcal{L}} \theta_{\mathrm{pa}(v)}$ for all v or, equivalently *if the conditional distributions $\theta_{v \mid \mathrm{pa}(v)}, v \in V$ are mutually independent.*

Graphically, this is most easily displayed by introducing one *additional parent* $\theta_{v \mid \mathrm{pa}(v)}$ for every vertex V in $\mathcal{D}$, so then

$$f(x \mid \theta) = \prod_{v \in V} f(x_v \mid x_{\mathrm{pa}(v)}, \theta_{v \mid \mathrm{pa}(v)}).$$

Exploiting independence and taking expectations over θ yields that *also marginally*,

$$f(x \mid \mathcal{D}) = \int_{\Theta_{\mathcal{D}}} f(x \mid \theta) \, \mathcal{L}_{\mathcal{D}}(\theta) = \prod_{v \in V} f(x_v \mid x_{\mathrm{pa}(v)}).$$

If $\mathcal{L}$ is strongly directed hyper Markov and $\mathcal{L}^*$ it holds that *also the posterior law* $\mathcal{L}^*$ *is is strongly directed hyper Markov* and

$$\mathcal{L}^*(\theta_{v \mid \mathrm{pa}(v)}) \propto f(x_v \mid x_{\mathrm{pa}(v)}, \theta_{v \mid \mathrm{pa}(v)}) \mathcal{L}(\theta_{v \mid \mathrm{pa}(v)})$$

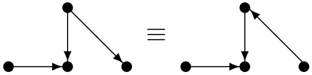
(Spiegelhalter and Lauritzen 1990).

Markov equivalence

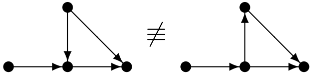
$\mathcal{D}$ and $\mathcal{D}'$ are equivalent if and only if:

1. $\mathcal{D}$ and $\mathcal{D}'$ have same *skeleton* (ignoring directions)
2. $\mathcal{D}$ and $\mathcal{D}'$ have same unmarried parents

so



but



Searching equivalence classes

In general, there is no hope of distinguishing Markov equivalent DAGs, so $\mathcal{D}$ *can at best be identified up to Markov equivalence.*

The number D_n of unlabelled DAGs with n vertices is given by the recursion (Robinson 1977)

$$D_n = \sum_{i=1}^{n} (-1)^{i+1} \binom{n}{i} 2^{i(n-i)} D_{n-i}$$

which grows superexponentially. For $n = 10$, $D_n \approx 4.2 \times 10^{18}$. The number of equivalence classes is smaller, but is conjectured still to grow superexponentially.

Conjugate priors for DAGs

In the discrete case, the obvious conjugate prior is for fixed v to let

$$\{\theta_{v \mid \mathrm{pa}_{\mathcal{D}}(v)}(x_v \mid x^*_{\mathrm{pa}_{\mathcal{D}}(v)}), x_v \in \mathcal{X}_v\}$$

be *Dirichlet distributed* and independent for $v \in V$ and $x^*_{\mathrm{pa}_{\mathcal{D}}(v)} \in \mathcal{X}_{\mathrm{pa}_{\mathcal{D}}(v)}$ (Spiegelhalter and Lauritzen 1990).

We can derive these Dirichlet distributions from a fixed *master Dirichlet* distribution $\mathcal{D}(\alpha)$, where $\alpha = \alpha(x), x \in \mathcal{X}$, by letting

$$\{\theta_{v \mid \mathrm{pa}(v)}(x_v \mid x^*_{\mathrm{pa}_{\mathcal{D}}(v)})\} \sim \mathcal{D}(\alpha(x_v, x^*_{\mathrm{pa}_{\mathcal{D}}(v)}),$$

where as usual $\alpha(x_a) = \sum_{y: y_a = x_a} \alpha(y)$.

Typically, α is specified by letting $\alpha = \lambda p_0(x)$ where p_0 is an initial guess on the joint distribution, for example specified through a DAG $\mathcal{D}_0$, and λ is the *equivalent sample size* for the prior information.

The values $\alpha(x_v, x^*_{\mathrm{pa}_{\mathcal{D}}(v)}) = \lambda p_0(x_v, x^*_{\mathrm{pa}_{\mathcal{D}}(v)})$ can then be calculated by *probability propagation*.

Common default values is $\lambda = 1$ and $\alpha(x) = |\mathcal{X}|^{-1}$.

A similar construction is possible in the Gaussian case using the Wishart distribution (Geiger and Heckerman 1994) and for mixed discrete Gaussian networks (Bøttcher 2001), the latter implemented in the R-package DEAL (Bøttcher and Dethlefsen 2003).

Characterization of strong hyper priors

In all cases, it was shown (Geiger and Heckerman 1997, 2002) that *prior distributions constructed in this way are the* only *distributions which are*

1. *modular:*

$$\mathrm{pa}_{\mathcal{D}}(v) = \mathrm{pa}_{\mathcal{D}'}(v) \implies \theta_{v \mid \mathrm{pa}_{\mathcal{D}}(v)} \sim \theta_{v \mid \mathrm{pa}_{\mathcal{D}'}(v)};$$

2. *score equivalent:*

$$\mathcal{D} \equiv \mathcal{D}' \implies f(x^{(n)} \mid \mathcal{D}) = f(x^{(n)} \mid \mathcal{D}').$$

Marginal likelihood

Bayes factors derived from these *strongly directed hyper Dirichlet priors* have a simple form

$$f(x^{(n)} \,|\, \mathcal{D}) \;=\; \prod_{v} \prod_{x_{\mathrm{pa}(v)}} \frac{\Gamma(\alpha(x_{\mathrm{pa}_{\mathcal{D}}(v)}))}{\Gamma(\alpha(x_{\mathrm{pa}_{\mathcal{D}}(v)}) + n(x_{\mathrm{pa}_{\mathcal{D}}(v)}))} \times \prod_{x_v} \frac{\Gamma(\alpha(x_{v\cup \mathrm{pa}_{\mathcal{D}}(v)}) + n(x_{v\cup \mathrm{pa}_{\mathcal{D}}(v)}))}{\Gamma(\alpha(x_{v\cup \mathrm{pa}_{\mathcal{D}}(v)}))}.$$

(Cooper and Herskovits 1992;
Heckerman $et\ al$. 1995)

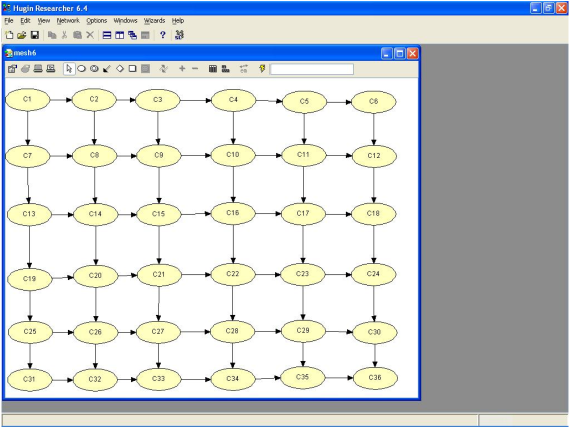
Challenge: Find *good algorithm for sampling* from the full posterior over DAGs or equivalence classes of DAGs. *Issue:* prior uniform over equivalence classes or over DAGs?

Greedy equivalence class search

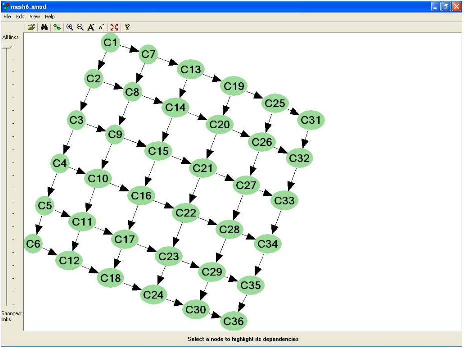
1. Initialize with empty DAG
2. Repeatedly search among equivalence classes with a *single additional edge* and go to class with highest score - until no improvement.
3. Repeatedly search among equivalence classes with a *single edge less* and move to one with highest score - until no improvement.

For BIC or Bayesian posterior score with directed hyper Dirichlet priors, this algorithm yields consistent estimate of equivalence class for P. (Chickering 2002)

Markov mesh model

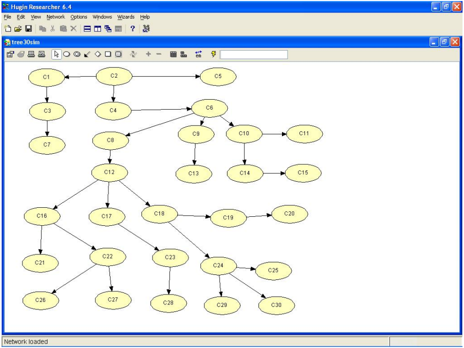


Bayesian GES

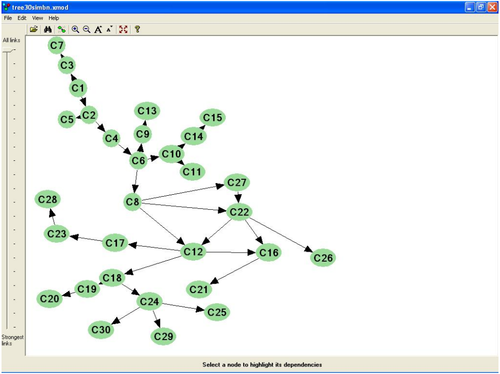


Crudest algorithm (WinMine), 10000 simulated cases

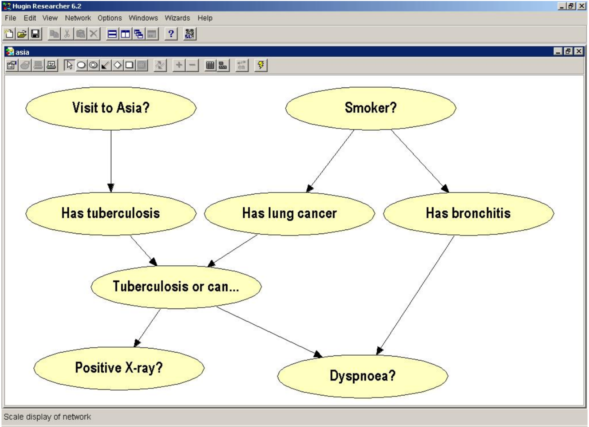
Tree model


Hugin Researcher 6.4
File Edit View Network Options Windows Wizards Help
tree30sim
C1
C2
C5
C3
C4
C6
C7
C8
C9
C10
C11
C12
C13
C14
C15
C16
C17
C18
C19
C20
C21
C22
C23
C24
C25
C26
C27
C28
C29
C30
Network loaded

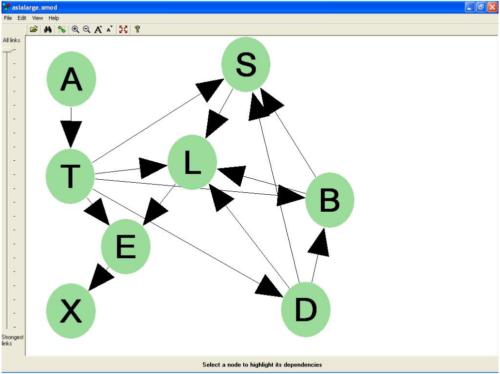
Bayesian GES on tree



Chest clinic



Bayesian GES



Constraint-based search

Another alternative search algorithm is known as *constraint based search*.

Essentially, the search methods generate queries of the type "$A \perp\!\!\!\perp B \mid S$?", and the answer to such a query divides Γ into those graphs conforming with the query and those that do not.

These type of methods were originally designed by computer scientists in the context where P was fully available, so queries could be answered without error.

The advantage of this type of method is that relatively few queries are needed to identify a DAG $\mathcal{D}$ (or rather its equivalence class).

The disadvantage is that there seems to be no coherent and principled method to answer the query in the presence of statistical uncertainty, which is computable.

SGS and PC algorithms

SGS-algorithm (Spirtes $et\ al.$ 1993):

Step 1: Identify *skeleton* using that, for P faithful,

$$u \not\sim v \iff \exists S \subseteq V \setminus \{u, v\} : X_u \perp\!\!\!\perp X_v \mid X_S.$$

Begin with complete graph, check for $S = \emptyset$ and remove edges when independence holds. Then continue for increasing $|S|$.

PC-algorithm (same reference) exploits that only S with $S \subseteq \mathrm{bd}(u) \setminus v$ or $S \subseteq \mathrm{bd}(v) \setminus u$ needs checking where bd refers to current skeleton.

Step 2: Identify directions to be consistent with independence relations found in Step 1.

Exact properties of PC-algorithm

If P is faithful to DAG $\mathcal{D}$, PC-algorithm finds $\mathcal{D}'$ equivalent to $\mathcal{D}$.

It uses N independence checks where N is at most

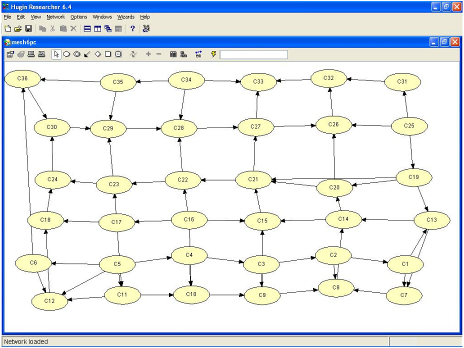
$$N \leq 2\binom{|V|}{2}\sum_{i=0}^{d}\binom{|V|-1}{i} \leq \frac{|V|^{d+1}}{(d-1)!},$$

where d is the maximal degree of any vertex in $\mathcal{D}$.

So worst case complexity is exponential, but *algorithm fast for sparse graphs.*

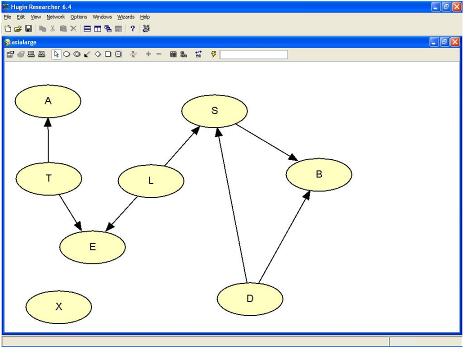
Sampling properties are less well understood although consistency results exist.

PC algorithm



Crudest algorithm (HUGIN), 10000 simulated cases

PC algorithm



10000 simulated cases

NPC algorithm

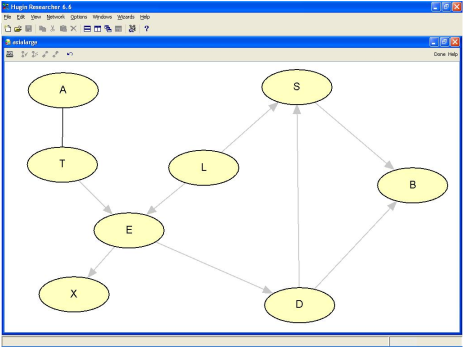
The NPC algorithm (Steck and Tresp 1996) stabilises the PC algorithm by adding a *necessary path condition*.

The general idea has these elements:

1. When a query is decided negatively, $\neg(A \perp\!\!\!\perp B \mid S)$, it is *taken at face value*; When a query is decided positively, $A \perp\!\!\!\perp B \mid S$, it is *recorded with care*;
2. If at some later stage, the PC algorithm would remove an edge so that a negative query $\neg(A \perp\!\!\!\perp B \mid S)$ would conflict with $A \perp_{\mathcal{D}} B \mid S$, the removal of this edge is suppressed.

   This leads to *unresolved queries* which are then passed to the user.

NPC algorithm



10000 simulated cases

References

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, **19**, 716–23.

Aldous, D. (1990). A random walk construction of uniform spanning trees and uniform labelled trees. *SIAM Journal on Discrete Mathematics*, **3**, (4), 450–65.

Atay-Kayis, A. and Massam, H. (2005). A Monte Carlo method for computing the marginal likelihood in non-decomposable graphical Gaussian models. *Biometrika*, **92**, 317–35.

Bollobás, B. (1998). *Modern graph theory*. Springer-Verlag, New York.

Bøttcher, S. G. (2001). Learning Bayesian networks with mixed variables. In *Proceedings of the eighth international workshop in artificial intelligence and statistics*, pp. 149–56.

Bøttcher, S. G. and Dethlefsen, C. (2003). `deal`: A package for learning Bayesian networks. *Journal of Statistical Software*, **8**, 1–40.

Chickering, D. M. (2002). Optimal structure identification with greedy search. *Journal of Machine Learning Research*, **3**, 507–54.

Cooper, G. F. and Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, **9**, 309–47.

Geiger, D. and Heckerman, D. (1994). Learning Gaussian networks. In *Proceedings of the 10th conference on uncertainty in artificial intelligence*, (ed. R. L. de Mantaras and D. Poole), pp. 235–43. Morgan Kaufmann Publishers, San Francisco, CA.

Geiger, D. and Heckerman, D. (1997). A characterization of the Dirichlet distribution through global and local independence. *Annals of Statistics*, **25**, 1344–69.

Geiger, D. and Heckerman, D. (2002). Parameter priors for directed acyclic graphical models and the characterization of several probability distributions. *Annals of Statistics*, **30**, 1412–40.

Heckerman, D., Geiger, D., and Chickering, D. M. (1995). Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, **20**, 197–243.

Robinson, R. W. (1977). Counting unlabelled acyclic digraphs. In *Lecture notes in mathematics: Combina-*

torial mathematics V, (ed. C. H. C. Little). Springer-Verlag, New York.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, **6**, 461–4.

Spiegelhalter, D. J. and Lauritzen, S. L. (1990). Sequential updating of conditional probabilities on directed graphical structures. *Networks*, **20**, 579–605.

Spirtes, P., Glymour, C., and Scheines, R. (1993). *Causation, prediction and search*. Springer-Verlag, New York. Reprinted by MIT Press.

Steck, H. and Tresp, V. (1996). Bayesian belief networks for data mining. In *Proceedings of 2nd workshop on Data Mining und Data Warehousing als Grundlage moderner entscheidungsunterstützender Systeme*, pp. 145–54. Magdeburg, Germany. University of Magdeburg.