Steffen L. Lauritzen

Elements of Graphical Models DRAFT.

Lectures from the XXXVIth International Probability Summer School in Saint-Flour, France, 2006

December 2, 2009

Springer

Contents

Chapter 6
Estimation of Structure

6.1 Estimation of Structure and Bayes Factors

Previous chapters have considered the situation where the graph $\mathcal{G}$ defining the model has been known and the inference problems were concerned with an unknown P_θ with $\theta \in \Theta$. This chapter discusses inference concerning the graph $\mathcal{G}$, specifying only a family Γ of possible graphs.

It is important to ensure that any methods used must scale well with data size we typically need to consider many structures and also huge collections of high-dimensional data.

What we here choose to term *structure estimation* is also known under other names as *model selection* (mainstream statistics), *system identification* (engineering), or *structural learning* (AI or machine learning.) Different situations occur depending on the type of assumptions concerning Γ Common assumptions include that Γ is the set of *undirected graphs* over V; the set of *chordal graphs* over V; the set of *forests* over V; the set of *trees* over V; the set of *directed acyclic graphs* over V; or potentially other types of conditional independence structure.

Why estimation of structure?

It may be worthwhile to dwell somewhat on the rationale behind structure estimation. We think of it as a method to get a quick overview of relations between a huge set of variables in a complex stochastic system and see it in many ways as a parallel to e.g. histograms or density estimation which gives a rough overview of the features of univariate data. It will typically be used in areas such as, for example, general data mining, identification of gene regulatory networks, or for reconstructing family trees from DNA information. Established methods exist and are in daily routine use, but there is a clear need for better understanding of their statistical properties.

We begin by showing a few simple examples of structure estimation to motivate that the issue is not a priori impossible.

Example 6.1 (Markov mesh model). Figure 6.1 shows the graph of a so-called Markov mesh model with 36 variables. All variables are binary and the only variable

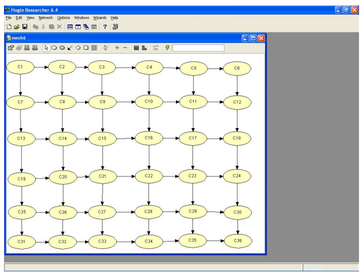


Fig. 6.1 Graph of a Markov mesh model with 36 binary variables.

without parents, in the upper left-hand corner is uniformly distributed. The remaining variables on the upper and left sides of the 6×6 square have a single parent and the conditional probability that it is in a given state is $3/4$ if the state is the same as its parent. The remaining nodes have two parents and if these are identical, the child with have that state with probability $3/4$ whereas it will otherwise follow the upper parent with probability 2/3.

Figure 6.2 shows two different attempts of estimating the structure based on the same 10,000 simulated cases. The two methods are to be described in more detail later, but it is apparent that the estimated structure in both cases have a strong similarity to the true one. In fact, one of the methods reconstructs the Markov mesh model perfectly. Both methods used search for a DAG structure which is compatible with the data.

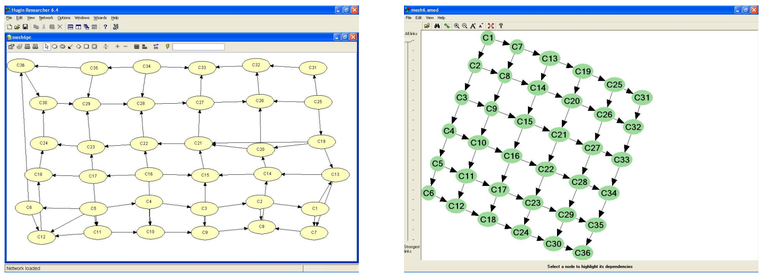


Fig. 6.2 Structure estimate of Markov mesh model from 10000 simulated cases. The left-hand side shows the estimate using the crudest algorithm (PC) implemented in HUGIN. The right-hand side the Bayesian estimate using greedy equivalence search (GES) as implemented in WINMINE.

Example 6.2 (Tree model). The graph of this example has a particular simple structure which is that of a rooted tree. Since a rooted tree with arrows pointing away from a root is a perfect DAG, the associated structure is equivalent to the corresponding undirected tree. The state at the root is uniformly distributed and any other node reproduces the state of the parent node with probability $3/4$.

Figure 6.3 shows the structure estimate of the tree based on 10,000 simulated cases and using the same methods as for the Markov mesh model. In both cases, the method has attempted to estimate the structure based on the assumption that the structure was a DAG. Note that in this case it is the first method which reconstruncts correctly whereas there are too many links in the second case.

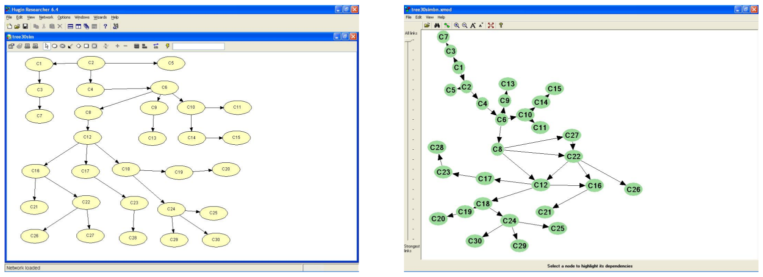


Fig. 6.3 Estimates of a tree model with 30 variables based on 10000 observations. The graph to the left represents the estimate using the PC algorithm and yields a 100% correct reconstruction. The graph to the right represents the Bayesian estimate using GES.

Example 6.3 (Chest clinic). The next example is taken from Lauritzen and Spiegelhalter (1988) and reflects the structure involving risk factors and symptoms for lung-disease. The (fictitious) description given by the authors of the associated medical knowledge is as follows

> "Shortness–of–breath (dyspnoea) may be due to tuberculosis, lung cancer or bronchitis, or none of them, or more than one of them. A recent visit to Asia increases the chances of tuberculosis, while smoking is known to be a risk factor for both lung cancer and bronchitis. The results of a single chest X–ray do not discriminate between lung cancer and tuberculosis, as neither does the presence or absence of dyspnoea."

The actual probabilities involved in this example are given in the original reference and we abstain from repeating them here.

Figure 6.4 displays the network structure reflecting the knowledge as given above and three different structure estimates. Note that this problem is obviously more difficult than the previous examples, in particular because some of the diseases are rare and larger data sets as well as more refined structure estimators are needed to even get close to the original structure.

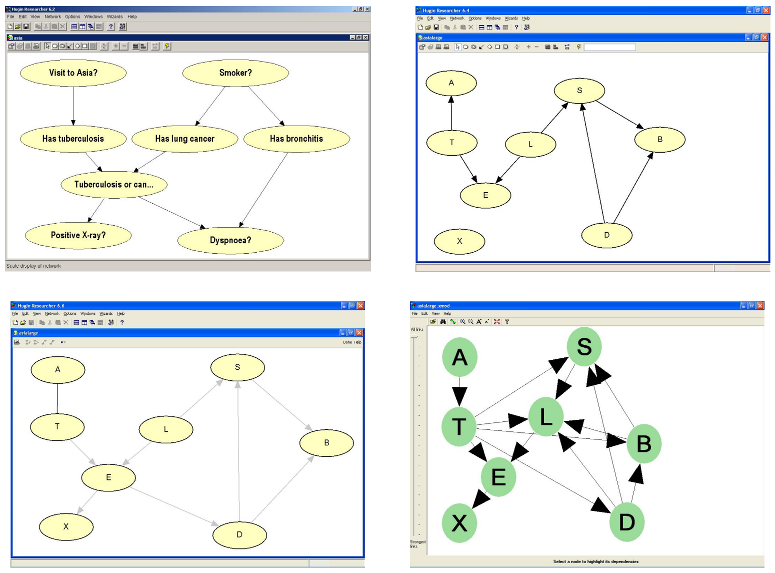


Fig. 6.4 A Bayesian network model for lung disease and estimates of the model based on simulated cases. The structure generating the data is in the upper left corner. Then, clockwise, estimates using the same data but different estimation algorithms: the PC algorithm, Bayesian GES, the NPC algorithm. In the latter case 100,000 cases were used.

Types of approach

Essentially all structure estimation methods combine a specification of potentially interesting structures with a way of judging the *adequacy of structure* and a *search strategy,* which evaluates a large number of space of possible structures.

As detailed further in the following sections, methods of judging adequacy include using

- tests of significance;
- penalised likelihood scores;

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

  with $\kappa = 1$ for AIC Akaike (1974), or $\kappa = \frac{1}{2}\log N$ for BIC Schwarz (1978);
- Bayesian posterior probabilities.

The search strategies are more or less based on heuristics, which all attempt to overcome the fundamental problem that a crude global search among all potential structures is not feasible as the number of structures is astronomical.

elaborate on each of these or rearrange

Bayes factors

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

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

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

where

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

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

Posterior distribution over graphs

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

$$\pi^*(\mathscr{G}) = \pi(\mathscr{G} \,|\, x^{(n)}) \propto f(x^{(n)} \,|\, \mathscr{G})\pi(\mathscr{G})$$

or equivalently

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

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

6.2 Estimating Trees and Forests

Estimating trees

Let us assume that the distribution P of $X = X_v, v \in V$ over a discrete state space $\mathscr{X}$ factorizes w.r.t. an unknown *tree* τ and that we have observations $X^1 = x_1, \ldots, X^n = x^n$ as independent and identically distributed according to P.

Chow and Liu (1968) showed that *the maximum likelihood estimate* $\hat{\tau}$ *of* τ *is a maximal weight spanning tree* (MWST), where the *weight* of a tree τ is

$$\lambda(\tau) = \sum_{e \in E(\tau)} \lambda_n(e) = \sum_{e \in E(\tau)} H_n(e)$$

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

$$H_n(e) = \sum_{x_u x_v} \frac{n(x_u,x_v)}{n} \log \frac{n(x_u,x_v)/n}{n(x_u)n(x_v)/n^2} = \sum_{x_u,x_v} n(x_u,x_v) \log \frac{n(x_u,x_v)}{n(x_u)n(x_v)}.$$

This result is easily *extended to Gaussian graphical models,* just with the weight $\lambda_n(e)$ of an edge in a tree determined as any strictly increasing function of the empirical cross-entropy along the edge

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

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

$$r_e^2 = \frac{(\sum_{i=1}^n x_u^i x_v^i)^2}{(\sum_{i=1}^n (x_u^i)^2)(\sum_{i=1}^n (x_v^i)^2)} = \frac{w_{uv}^2}{w_{uu}w_{vv}}.$$

To see this, use the expression (4.20) for the determinant of the MLE which in the case of a tree reduces to

$$\begin{aligned}\det(\hat{K}) &= \frac{\prod_{v\in V}(w_{vv})^{\deg(v)-1}}{\prod_{e\in E}\det(w_e)} n^d \\ &\propto \prod_{v\in V}(w_{vv})^{-1} \prod_{\{u,v\}\in E} \frac{w_{uu}w_{vv}}{w_{uu}w_{vv} - w_{uv}^2} \propto (1-r_e^2)^{-1}.\end{aligned}$$

From (4.16) we know that the maximized likelihood function for a fixed tree is proportional to a power of this determinant and hence is maximized when the logarithm of the determinant is maximized. But since we then have

$$\log\det\hat{K}(\tau) = 2\sum_{e\in E(\tau)} H_n(e) = 2\lambda(\tau),$$

maximizing $\hat{L}(\tau)$ over all possible trees is equivalent to maximizing $\lambda(\tau)$.

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

Fast algorithms Kruskal Jr. (1956) compute maximal weight spanning tree (or forest) from weights $W = (w_{uv}, u, v \in V)$.

Chow and Wagner (1978) show *a.s. consistency in total variation of $\hat{P}$:* If P factorises w.r.t. τ, then

$$\sup_x |p(x) - \hat{p}(x)| \to 0 \text{ for } n \to \infty,$$

so *if τ is unique for P, $\hat{\tau} = \tau$ for all $n > N$ for some N.*

If P does not factorize w.r.t. a tree, *$\hat{P}$ converges to closest tree-approximation $\tilde{P}$ to P* (Kullback-Leibler distance).

Strong hyper Markov prior laws

For strong hyper Markov prior laws, $X^{(n)}$ is itself marginally Markov so

$$f(x^{(n)} \mid \mathscr{G}) = \frac{\prod_{Q\in\mathscr{Q}} f(x_Q^{(n)} \mid \mathscr{G})}{\prod_{S\in\mathscr{S}} f(x_S^{(n)} \mid \mathscr{G})^{\nu_{\mathscr{G}}(S)}}, \tag{6.1}$$

where $\mathscr{Q}$ are the prime components and $\mathscr{S}$ the minimal complete separators of $\mathscr{G}$.

Hyper inverse Wishart laws

Denote the normalisation constant of the hyper inverse Wishart density as

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

i.e. the usual Wishart constant if $Q = C$ is a clique.

Combining with the Gaussian likelihood, it is easily seen that for Gaussian graphical models we have

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

Comparing with (6.1) leads to a similar factorization of the normalising constant

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

For *chordal graphs* all terms in this expression reduce to known Wishart constants, and we can thus calculate the normalization constant explicitly.

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

The marginal distribution of $W^{(n)}$ is (weak) *hyper Markov* w.r.t. $\mathscr{G}$. It was termed the *hyper matrix F law* by Dawid and Lauritzen (1993).

Bayes factors for forests

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

$$f(\phi \mid x^{(n)}) \propto \frac{\prod_{e\in E(\phi)} f(x_e^{(n)})}{\prod_{v\in V} f(x_v^{(n)})^{d_\phi(v)-1}},$$

since all minimal complete separators are singletons and $\nu_\phi(\{v\}) = d_\phi(v) - 1$.

Multiplying the right-hand side with $\prod_{v\in V} f(x_v^{(n)})$ yields

$$\frac{\prod_{e\in E(\phi)} f(x_e^{(n)})}{\prod_{v\in V} f(x_v^{(n)})^{d_\phi(v)-1}} = \prod_{v\in V} f(x_v^{(n)}) \prod_{e\in\phi} \mathrm{BF}(e),$$

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

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

Thus the *posterior distribution* of ϕ is

$$\pi^*(\phi) \propto \prod_{e\in E(\phi)} \mathrm{BF}(e).$$

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

Bayesian analysis

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

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

These work less well for weights occurring with typical Bayes factors, as most of these are essentially zero, so methods based on the *Matrix Tree Theorem* seem currently more useful.

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

Some challenges for undirected graphs

- Find *feasible algorithm for (perfect) simulation* from a distribution over chordal graphs as
$$p(\mathscr{G}) \propto \frac{\prod_{C\in\mathscr{C}} w(C)}{\prod_{S\in\mathscr{S}} w(S)^{\nu_\mathscr{G}(S)}},$$
where $w(A), A \subseteq V$ are a prescribed set of positive weights.
- Find *feasible algorithm for obtaining MAP* in decomposable case. This may not be universally possible as problem most likely is NP-complete.

6.3 Learning Bayesian networks

6.3.1 Model search methods

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

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

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

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

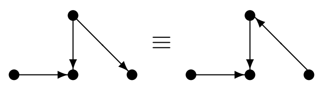
Spiegelhalter and Lauritzen (1990).

Markov equivalence

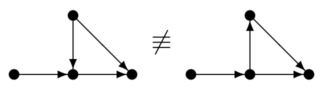
$\mathcal{D}$ and $\mathcal{D}'$ are equivalent if and only if:

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 $\mathscr{D}$ *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

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

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

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

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

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 $\mathscr{D}_0$, and λ is the *equivalent sample size* for the prior information.

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

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

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).

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:*
$$\mathrm{pa}_{\mathscr{D}}(v) = \mathrm{pa}_{\mathscr{D}'}(v) \Rightarrow \theta_{v\,|\,\mathrm{pa}_{\mathscr{D}}(v)} \sim \theta_{v\,|\,\mathrm{pa}_{\mathscr{D}'}(v)};$$
2. *score equivalent:*
$$\mathscr{D} \equiv \mathscr{D}' \Rightarrow f(x^{(n)}\,|\,\mathscr{D}) = f(x^{(n)}\,|\,\mathscr{D}').$$

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

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

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)

6.3.2 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 $\mathscr{D}$, PC-algorithm finds $\mathscr{D}'$ equivalent to $\mathscr{D}$.

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 $\mathscr{D}$.

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

Sampling properties are less well understood although consistency results exist.

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 $A \perp_{\mathscr{D}} B \mid S$, the removal of this edge is suppressed.
   This leads to *unresolved queries* which are then passed to the user.

6.4 Summary

Types of approach

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

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

  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.*

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

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

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

where

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

is known as the *marginal likelihood* of $\mathscr{G}$. Posterior distribution over graphs If $\pi(\mathscr{G})$ is a prior probability distribution over a given set of graphs Γ, the posterior distribution is determined as

$$\pi^*(\mathscr{G}) = \pi(\mathscr{G} \,|\, x^{(n)}) \propto f(x^{(n)} \,|\, \mathscr{G}) \pi(\mathscr{G})$$

or equivalently

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

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* $\mathscr{G}^*$ maximizing $\pi^*(\mathscr{G})$ over Γ, or attempts to *sample from the posterior* using e.g. Monte-Carlo methods. Estimating trees Assume P factorizes w.r.t. an unknown *tree* $\mathscr{T}$. *MLE $\hat{\tau}$ of $\mathscr{T}$ 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

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

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 $W = (w_{uv}, u, v \in V)$.

Hyper inverse Wishart laws Denote the normalisation constant of the hyper inverse Wishart density as

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

The marginal likelihood is then

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

where

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

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.