

Bayesian AI Tutorial

Ann E. Nicholson and Kevin B. Korb

Faculty of Information Technology
Monash University
Clayton, VIC 3168
AUSTRALIA

{annn,korb}@csse.monash.edu.au

[HTTP://WWW.CSSE.MONASH.EDU.AU/BAI](http://www.csse.monash.edu.au/bai)

Text: *Bayesian Artificial Intelligence*, Kevin B. Korb
and Ann E. Nicholson, Chapman & Hall/CRC, 2004.

Overview

1. Bayesian AI
2. Introduction to Bayesian networks
3. Extensions to Bayesian networks
 - (a) Decision networks
 - (b) Dynamic Bayesian networks
4. Learning Bayesian networks
5. Knowledge Engineering with Bayesian networks
6. Monash BAI Group: BN Applications

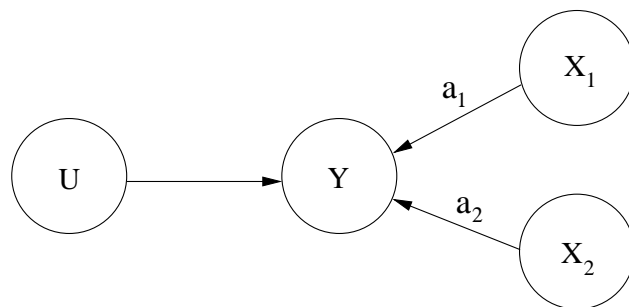
Learning Bayesian Networks

- Linear and Discrete Models
- Learning Network Parameters
 - Linear Coefficients
 - Learning Probability Tables
- Learning Causal Structure
- Conditional Independence Learning
 - Statistical Equivalence
 - TETRAD II
- Bayesian Learning of Bayesian Networks
 - Cooper & Herskovits: K2
 - Learning Variable Order
 - Statistical Equivalence Learners
- Full Causal Learners
- Minimum Encoding Methods
 - Lam & Bacchus's MDL learner
 - MML metrics
 - MML search algorithms
 - MML Sampling
- Empirical Results

Linear and Discrete Models

Linear Models: Used in biology & social sciences since Sewall Wright (1921)

Linear models represent causal relationships as sets of linear functions of “independent” variables.



Equivalently:

$$X_3 = a_{13}X_1 + a_{23}X_2 + \epsilon_1$$

Structural equation models (SEMs) are close relatives

Discrete models: “Bayesian nets” replace vectors of linear coefficients with CPTs.

Learning Linear Parameters

Maximum likelihood methods have been available since Wright's path model analysis (1921).

Equivalent methods:

- Simon-Blalock method (Simon, 1954; Blalock, 1964)
- Ordinary least squares multiple regression (OLS)

Learning Conditional Probability Tables

Spiegelhalter & Lauritzen (1990):

- assume parameter independence
- each CPT cell i = a parameter in a Dirichlet distribution

$$D[\alpha_1, \dots, \alpha_i, \dots, \alpha_K]$$

for K parents

- prob of outcome i is $\alpha_i / \sum_{k=1}^K \alpha_k$
- observing outcome i update D to

$$D[\alpha_1, \dots, \alpha_i + 1, \dots, \alpha_K]$$

Others are looking at learning without parameter independence. E.g.,

- Decision trees to learn structure within CPTs (Boutillier et al. 1996).
- Dual log-linear and full CPT models (Neil, Wallace, Korb 1999).

Learning Causal Structure

This is the *real* problem; parameterizing models is relatively straightforward estimation problem.

Size of the dag space is superexponential:

- Number of possible orderings: $n!$
- Times number of ways of pairing up (for arcs): $2^{C_2^n}$
- Minus number of possible cyclic graphs

Without the subtraction (which is a small proportion):

n	$n!2^{C_2^n}$
0	0
1	1
2	4
3	48
4	1536
5	12280
10	127677049435953561600
100	[too many digits to show]

Learning Causal Structure

There are two basic methods:

- Learning from conditional independencies (CI learning)
- Learning using a scoring metric (Metric learning)

CI learning (Verma and Pearl, 1991)

Suppose you have an Oracle who can answer yes or no to any question of the type:

$$X \perp\!\!\!\perp Y | S?$$

(i.e., is X conditional independence Y given S)

Then you can learn the correct causal model, up to statistical equivalence (patterns).

Verma-Pearl Algorithm

two rules allow discovery of the set of causal models consistent with all such answers (“patterns”):

1. **Principle I** Put an undirected link between any two variables X and Y iff for every \mathbf{S} s.t. $X, Y \notin \mathbf{S}$

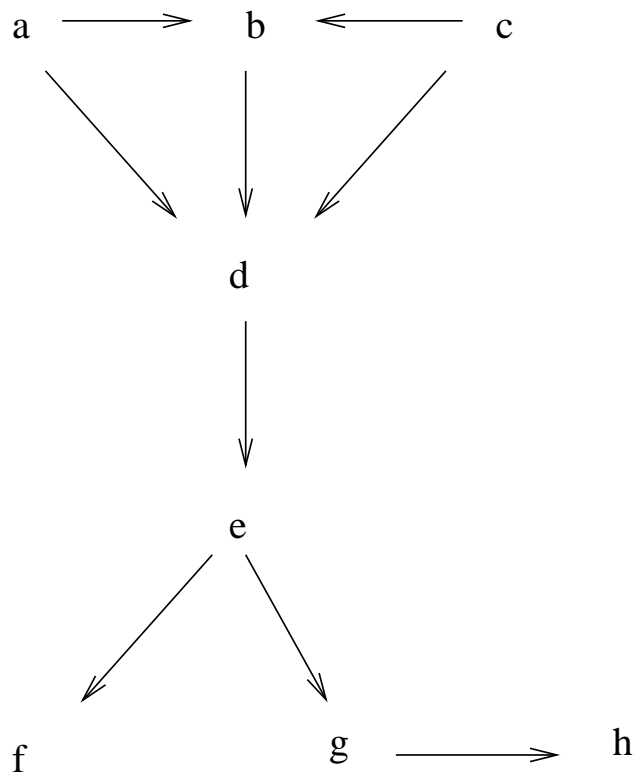
$$\neg(X \perp\!\!\!\perp Y) | \mathbf{S}$$

2. **Principle II** For every undirected v-structure $X - Z - Y$ orient the arcs $X \rightarrow Z \leftarrow Y$ iff

$$\neg(X \perp\!\!\!\perp Y) | \mathbf{S}$$

for **every** \mathbf{S} s.t. $X, Y \notin \mathbf{S}$ and $Z \in \mathbf{S}$.

CI learning example



CI learning example

$$1) a - b - c \quad a \rightarrow b \leftarrow c$$

b [induces a dependency]

$$2) a - d - c \quad a \rightarrow d \leftarrow c$$

$$3) c - d - e \quad \neg(c \rightarrow d \leftarrow e)$$

therefore $c \rightarrow d \rightarrow e$

$$4) a - d - e \quad \text{no news}$$

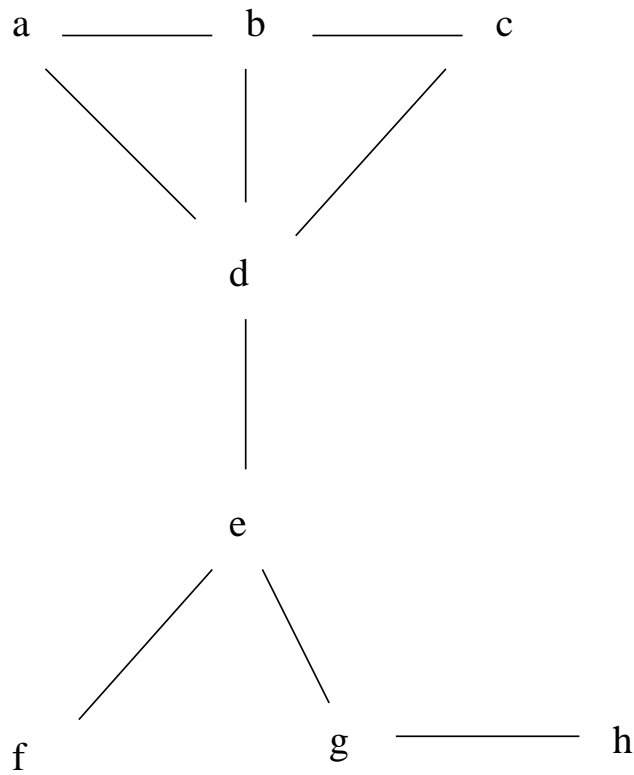
$$5) b - d - e \quad \text{no news}$$

$$6) d - e - f \quad \neg(d \rightarrow e \leftarrow f)$$

$$7) d - e - g \quad \neg(d \rightarrow e \leftarrow g)$$

$$6) e - g - h \quad \neg(e \rightarrow g \leftarrow h)$$

CI learning example



Statistical Equivalence

Verma and Pearl's rules identify the set of causal models which are statistically equivalent —

Two causal models H_1 and H_2 are **statistically equivalent** iff they contain the same variables and joint samples over them provide no statistical grounds for preferring one over the other.

Examples

- All fully connected models are equivalent.
- $A \rightarrow B \rightarrow C$ and $A \leftarrow B \leftarrow C$.
- $A \rightarrow B \rightarrow D \leftarrow C$ and $A \leftarrow B \rightarrow D \leftarrow C$.

Statistical Equivalence

- (Verma and Pearl, 1991): Any two causal models over the same variables which have the same skeleton (undirected arcs) and the same directed v-structures are statistically equivalent.
- Chickering (1995): If H_1 and H_2 are statistically equivalent, then they have the same maximum likelihoods relative to any joint samples:

$$\max P(e|H_1, \theta_1) = \max P(e|H_2, \theta_2)$$

where θ_i is a parameterization of H_i

TETRAD II

— Spirtes, Glymour and Scheines (1993)

Replace the Oracle with statistical tests:

- for linear models a significance test on partial correlation

$$X \perp\!\!\!\perp Y | \mathbf{S} \text{ iff } \rho_{XY \cdot \mathbf{S}} = 0$$

- for discrete models a χ^2 test on the difference between CPT counts expected with independence (E_i) and observed (O_i)

$$X \perp\!\!\!\perp Y | \mathbf{S} \text{ iff } \sum_i O_i \ln \left(\frac{O_i}{E_i} \right)^2 \approx 0$$

Implemented in their **PC Algorithm**

TETRAD II: Weak Links and Small Samples

Main weakness of TETRAD II: orthodox sig tests.

- As the order of partials goes up, the number of correlations required to be estimated goes up.
- Since sig tests are not robust, TETRAD II may work ok on small models with large samples, but unlikely to work on large models with modest samples
- This point was demonstrated empirically in Dai, Korb, Wallace & Wu (1997).

Bayesian LBN: Cooper & Herskovits' K2

— Cooper & Herskovits (1991, 1992)

Compute $P(h_i|e)$ by brute force, under the assumptions:

1. All variables are discrete.
2. Samples are i.i.d.
3. No missing values.
4. All values of child variables are uniformly distributed.
5. Priors over hypotheses are uniform.

With these assumptions, Cooper & Herskovits reduce the computation of $P_{CH}(h, e)$ to a polynomial time counting problem.

Cooper & Herskovits

But the hypothesis space is exponential; they go for dramatic simplification:

6. Assume we know the temporal ordering of the variables.

In that case, for any pair of variables the only problem is

- deciding whether they are connected by an arc
 - arc direction is trival
 - cycles are impossible.

New hypothesis space has size only $2^{(n^2-n)/2}$ (still exponential).

Algorithm “K2” does a greedy search through this reduced space.

Learning Variable Order

Reliance upon a given variable order is a major drawback to K2

And many other algorithms (Buntine 1991, Bouckert 1994, Suzuki 1996, Madigan & Raftery 1994)

What's wrong with that?

- We want autonomous AI (data mining). If experts can order the variables they can likely supply models.
- Determining variable ordering is half the problem. If we know A comes before B , the only remaining issue is whether there is a link between the two.
- The number of orderings consistent with dags is exponential (Brightwell & Winkler 1990; number complete). So iterating over all possible orderings will not scale up.

Statistical Equivalence Learners

Heckerman & Geiger (1995) advocate learning only up to statistical equivalence classes (a la TETRAD II).

Since observational data cannot distinguish btw equivalent models, there's no point trying to go further.

⇒ Madigan, Andersson, Perlman & Volinsky (1996) follow this advice, use uniform prior over equivalence classes.

⇒ Geiger and Heckerman (1994) define Bayesian metrics for linear and discrete equivalence classes of models (BGe and BDe)

GES

Greedy Equivalence Search (GES)

- Product of the CMU-Microsoft group (Meek, 1996; Chickering, 2002)
- Two-stage greedy search: Begin with unconnected pattern
 1. Greedily add single arcs until reaching a local maximum
 2. Prune back edges which don't contribute to the score
- Uses a Bayesian score over patterns only
- Implemented in TETRAD and Murphy's BNT

Statistical Equivalence Learners

Wallace & Korb (1999): This is not right!

- These are **causal** models; they *are* distinguishable on *experimental* data.
 - Failure to collect some data is no reason to change prior probabilities.
E.g., If your thermometer topped out at 35° , you wouldn't treat $\geq 35^\circ$ and 34° as equally likely.
- Not all equivalence classes are created equal:
 $\{ A \leftarrow B \rightarrow C, A \rightarrow B \rightarrow C, A \leftarrow B \leftarrow C \}$
 $\{ A \rightarrow B \leftarrow C \}$
- *Within* classes some dags should have greater priors than others... E.g.,
LightsOn \rightarrow InOffice \rightarrow LoggedOn v.
LightsOn \leftarrow InOffice \rightarrow LoggedOn

Full Causal Learners

So... a full causal learner is an algorithm that:

1. Learns causal connectedness.
 2. Learns v-structures.
Hence, learns equivalence classes.
 3. Learns full variable order.
Hence, learns full causal structure (order + connectedness).
- TETRAD II: 1, 2.
 - Madigan et al.; Heckerman & Geiger (BGe, BDe): 1, 2.
 - GES: 1, 2.
 - Cooper & Herskovits' K2: 1.
 - Lam and Bacchus MDL: 1, 2 (partial), 3 (partial).
 - Wallace, Neil, Korb MML: 1, 2, 3.

CaMML

Minimum Message Length (Wallace & Boulton 1968)
uses Shannon's measure of information:

$$I(m) = -\log P(m)$$

Applied in reverse, we can compute $P(h, e)$ from $I(h, e)$.

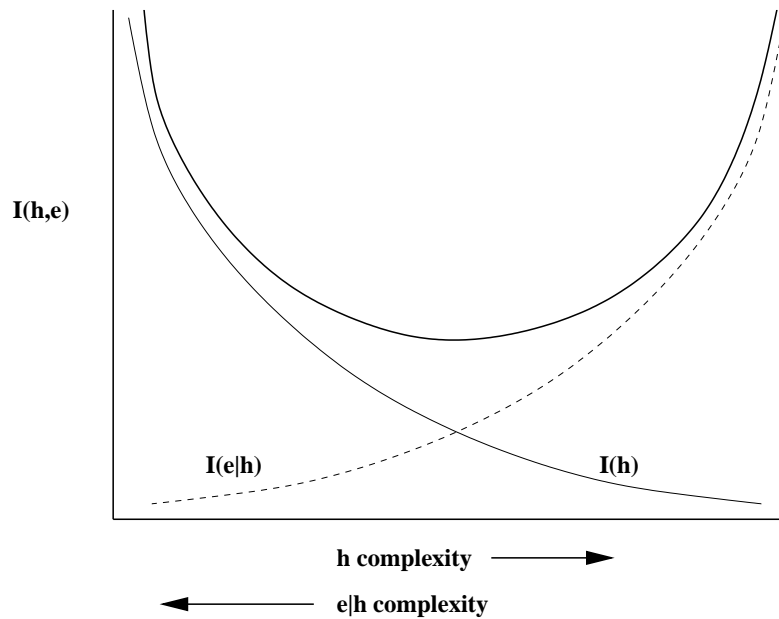
Given an *efficient* joint encoding method for the hypothesis & evidence space (i.e., satisfying Shannon's law), MML:

Searches $\{h_i\}$ for that hypothesis h that
minimizes $I(h) + I(e|h)$.

Applies a trade-off between

- Model simplicity
- Data fit

MML Metric



Equivalent to that h that maximizes $P(h)P(e|h)$ — i.e., $P(h|e)$.

$$\begin{aligned} I(h, e) &= I(h) + I(e|h) \\ -\log P(h, e) &= -\log P(h) - \log P(e|h) \\ -\log P(h, e) &= -\log P(h)P(e|h) \\ P(h, e) &= P(h)P(e|h) \end{aligned}$$

Hence, $\min I(h, e) \equiv \max P(h, e)$.

MML Metric for Linear Models

- Network:

$$\log n! + \frac{n(n-1)}{2} - \log E$$

- $\log n!$ for variable order
- $\frac{n(n-1)}{2}$ for connectivity
- $-\log E$ restore efficiency by subtracting cost of selecting a linear extension

- Parameters given dag h :

$$\sum_{X_j} -\log \frac{f(\theta_j|h)}{\sqrt{F(\theta_j)}}$$

where θ_j are the parameters for X_j and $F(\theta_j)$ is the Fisher information. $f(\theta_j|h)$ is assumed to be $N(0, \sigma_j)$.

(Cf. with MDL's fixed length for parms)

MML Metric for Linear Models

- Sample for X_j given h and θ_j :

$$-\log P(e|h, \theta_j) = \prod_{k=1}^K \frac{1}{\sqrt{2\pi\sigma_j}} e^{-\epsilon_{jk}^2/2\sigma_j^2}$$

where K is the number of sample values and ϵ_{jk} is the difference between the observed value of X_j and its linear prediction.

MML Metric for discrete models

We can use $P_{CH}(h_i, e)$ (from Cooper & Herskovits) to define an MML metric for discrete models.

Difference between MML and Bayesian metrics:

MML partitions the parameter space and selects optimal parameters.

Equivalent to a penalty of $\frac{1}{2} \log \frac{\pi e}{6}$ per parameter (Wallace & Freeman 1987); hence:

$$I(e, h_i) = \frac{s_j}{2} \log \frac{\pi e}{6} - \log P_{CH}(h_i, e) \quad (1)$$

Applied in MML Sampling algorithm.

MML search algorithms

MML metrics need to be combined with search. This has been done three ways:

1. Wallace, Korb, Dai (1996): greedy search (linear).
 - Brute force computation of linear extensions (small models only).
2. Neil and Korb (1999): genetic algorithms (linear).
 - Asymptotic estimator of linear extensions
 - GA chromosomes = causal models
 - Genetic operators manipulate them
 - Selection pressure is based on MML
3. Wallace and Korb (1999): MML sampling (linear, discrete).
 - Stochastic sampling through space of totally ordered causal models
 - No counting of linear extensions required

MML Sampling

Search space of totally ordered models (TOMs).

Sampled via a Metropolis algorithm (Metropolis et al. 1953).

From current model M , find the next model M' by:

- Randomly select a variable; attempt to swap order with its predecessor.
- Or, randomly select a pair; attempt to add/delete an arc.

Attempts succeed whenever $P(M')/P(M) > U$ (per MML metric), where U is uniformly random from $[0 : 1]$.

MML Sampling

Metropolis: this procedure samples TOMs with a frequency proportional to their posterior probability.

To find posterior of dag h : keep count of visits to all TOMs consistent with h

Estimated by counting visits to all TOMs with identical max likelihoods to h

Output: Probabilities of

- Top dags
- Top statistical equivalence classes
- Top MML equivalence classes

Empirical Results

A weakness in this area — and AI generally.

- Paper publications based upon very small models, loose comparisons.
- ALARM net often used — everything gets it to within 1 or 2 arcs.

Neil and Korb (1999) compared CaMML and BGe (Heckerman & Geiger's Bayesian metric over equivalence classes), using identical GA search over linear models:

- On KL distance and topological distance from the true model, CaMML and BGe performed nearly the same.
- On test prediction accuracy on *strict effect nodes* (those with no children), CaMML clearly outperformed BGe.