ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

D-BSSE
Department of Biosystems
Science and Engineering

Model selection

Niko Beerenwinkel

Learning Bayesian networks

$$(G,\ \theta)$$

Model selection
(structure learning)

Parameter estimation
(parameter learning)

Learning Bayesian networks

	Fully observed data	**Missing data / hidden variables**
Known graph structure (*parameter estimation*)	Sample statistics	EM algorithm Gradient ascent Sampling Variational inference
Unknown graph structure (*model selection*)	?	?

MCMC for DAGs (→ lecture 6)

Outline

- Bayesian tree models
- Evidence approximation
- Search and score
- Structural EM
- D-separation and the PC algorithm

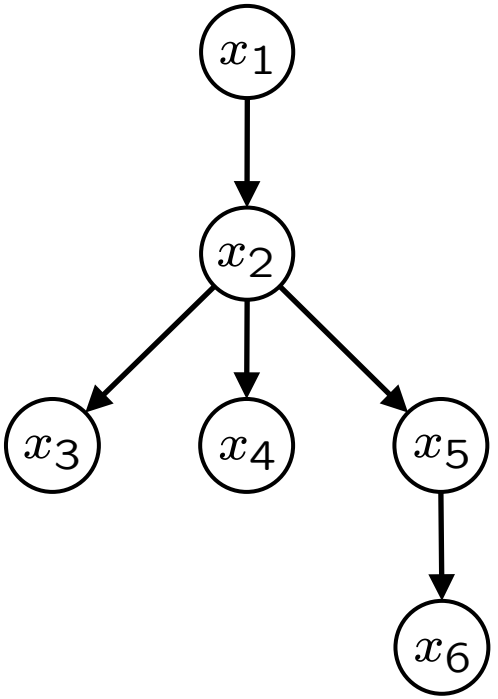
Bayesian tree models

Tree structured Bayesian networks

- Let T be a Bayesian tree model on the vertices {1, ..., p},

$$P_T(\mathbf{x}) = \prod_{i=1}^{p} P(x_i \mid x_{\mathrm{pa}(i)})$$

- If $\mathbf{x} = (x_1, ..., x_p)$ is discrete, then the model parameters are the probability tables $P(x_i \mid x_{pa(i)})$.
- Given observed data $\mathcal{D} = \{\mathbf{x}^1, ..., \mathbf{x}^N\}$, we want to find the "optimal" tree T.



KL divergence and entropy

- Recall that the KL divergence between two probability distributions P(X) and Q(X) is

$$\mathrm{D_{KL}}(P \parallel Q) = -\sum_X P(X) \log \frac{Q(X)}{P(X)} \geq 0$$

and $\mathrm{D_{KL}}(P \parallel Q) = 0 \Leftrightarrow P = Q$.

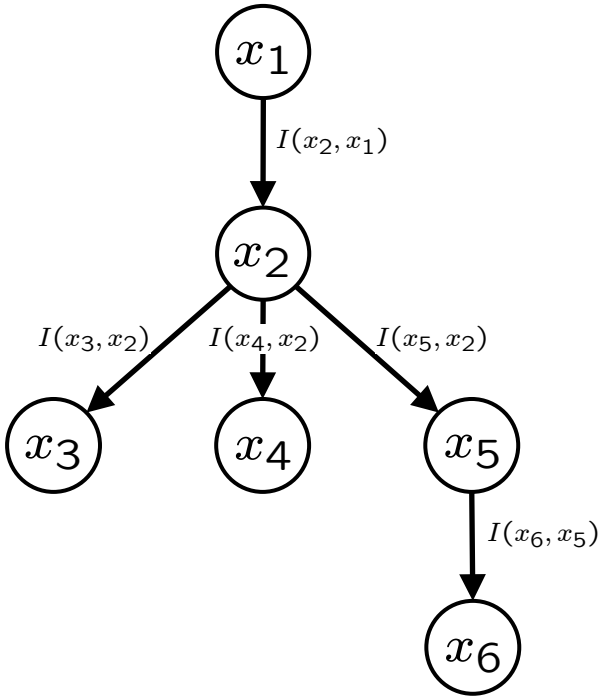
- Entropy: $H(X) = \mathrm{E}_X[-\log P(X)] = -\sum_X P(X) \log P(X)$

- Mutual information between two random variables:

$$\begin{aligned} I(X,Y) &= H(X) - H(X \mid Y) \\ &= \mathrm{D_{KL}}(P(X,Y) \parallel P(X)P(Y)) \end{aligned}$$

Optimal Bayesian tree models

- Define edge weights $I(x_i, x_{pa(i)})$ on T.
- T has total weight $\sum_{i=1}^{p} I(x_i, x_{\mathrm{pa}(i)})$
- **Theorem**: $D_{KL}(P \,||\, P_T)$ is minimal if and only if T has maximal weight.



Proof

$$
\begin{aligned}
\mathrm{D_{KL}}(P \parallel P_T) &= \sum_{\mathbf{x}} P(\mathbf{x}) \log \frac{P(\mathbf{x})}{\prod_i P(x_i \mid x_{\mathrm{pa}(i)})} \\
&= \sum_{\mathbf{x}} P(\mathbf{x}) \log P(\mathbf{x}) - \sum_{\mathbf{x}} P(\mathbf{x}) \sum_i \log P(x_i \mid x_{\mathrm{pa}(i)}) \\
&= -H(\mathbf{x}) - \sum_{\mathbf{x}} P(\mathbf{x}) \sum_i \log \left[\frac{P(x_i,\, x_{\mathrm{pa}(i)})}{P(x_i) P(x_{\mathrm{pa}(i)})} P(x_i) \right] \\
&= -H(\mathbf{x}) - \sum_i I(x_i,\, x_{\mathrm{pa}(i)}) + \sum_i H(x_i)
\end{aligned}
$$

independent of T

Kruskal's maximum weight spanning tree algorithm

- Input:
  - Complete graph G on {1, ..., p}
  - Weights $l(e) = l(i,j) = l(x_i, x_j)$ for all edges $e = (i, j) \in E(G)$
- Output:
  - A spanning tree of maximum weight

1. Sort edges such that $l(e_1) \geq l(e_2) \geq ... \geq l(e_m)$
2. Set $T := (V(G), \emptyset)$
3. For i = 1 to m:
   If $T + e_i$ contains no circuit, then set $T := T + e_i$

- $O(mp) = O(p^3)$, but can be implemented in $O(p^2 \log p)$

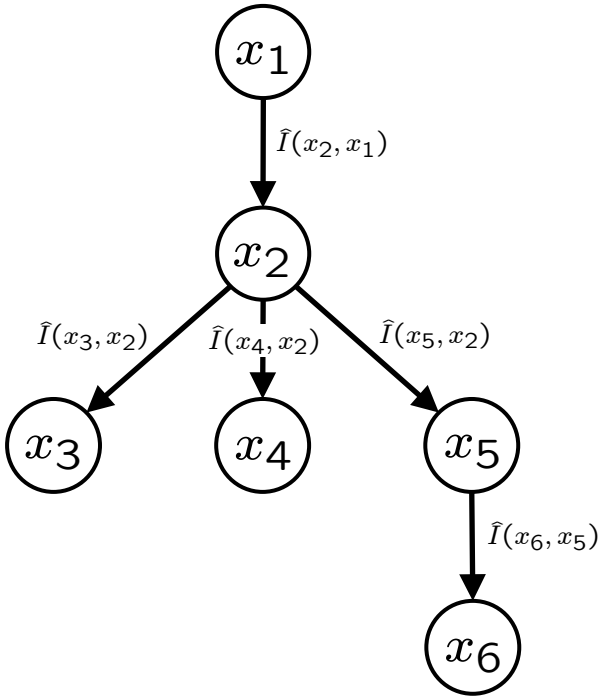
Maximum likelihood Bayesian tree models

- Let $f_{ij}(u,v)$ be the MLE of $P(x_i = u, x_j = v)$, and similarly $f_i(u)$.
- The mutual information can be estimated as

$$\hat{I}(x_i, x_j) = \sum_{u,v} f_{ij}(u,v) \log \frac{f_{ij}(u,v)}{f_i(u) f_j(v)}$$

- For a given tree, the ML parameter estimates are $f_{ij}(u,v) / f_j(v)$.

- **Theorem**: $D_{KL}(P \,||\, P_T)$ is minimal if and only if the likelihood $P(\mathcal{D} \mid T, P_{|T})$ is maximal.



Proof

$$L(\mathbf{x}^1,\ldots,\mathbf{x}^N) = \prod_{k=1}^{N} P_T(\mathbf{x}^k) = \prod_k \prod_i P(x_i^k \mid x_{\mathsf{pa}(i)}^k)$$

$$\Rightarrow \quad \max \ell(\mathbf{x}^1,\ldots,\mathbf{x}^N) = \max \left\{ \sum_i \sum_k \log P(x_i^k \mid x_{\mathsf{pa}(i)}^k) \right\}$$

$$= \max_T \left\{ \sum_i \max_{P|T} \left[\sum_k \log P(x_i^k \mid x_{\mathsf{pa}(i)}^k) \right] \right\}$$

$$= \max_T \left\{ \sum_i \left[\sum_k \log \frac{f_{i,\mathsf{pa}(i)}\left(x_i^k, x_{\mathsf{pa}(i)}^k\right)}{f_{\mathsf{pa}(i)}\left(x_{\mathsf{pa}(i)}^k\right)} \right] \right\}$$

$$= \max_T \left\{ \sum_i \hat{I}\left(x_i, x_{\mathsf{pa}(i)}\right) \right\} + \underbrace{\sum_i \sum_k \log P(x_i^k)}_{\text{independent of T}}$$

Evidence approximation

Bayesian learning of network structure

- MAP learning: $G^* = \underset{G}{\operatorname{argmax}} P(G \mid \mathcal{D})$
- Inference of the full posterior (e.g., by sampling, or variational inference):

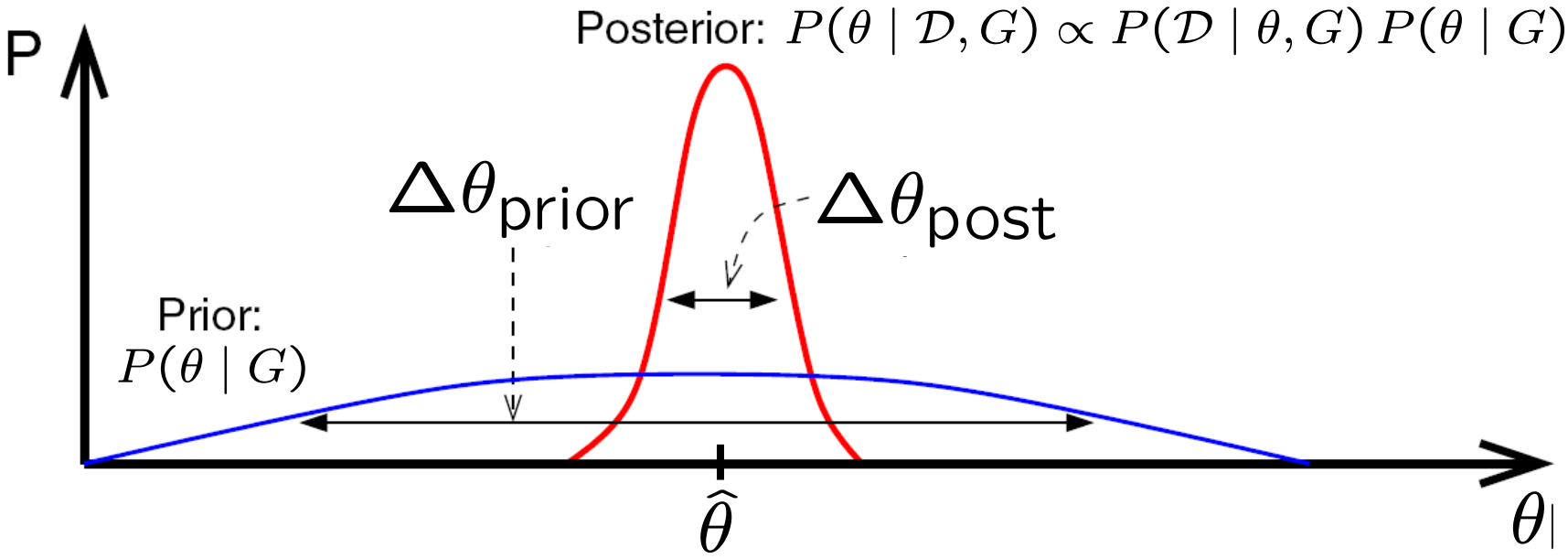
$$P(G \mid \mathcal{D}) = \frac{P(\mathcal{D} \mid G) P(G)}{P(\mathcal{D})}$$

where

$$P(\mathcal{D} \mid G) = \int P(\mathcal{D} \mid \theta, G) P(\theta \mid G) d\theta$$

is the marginal likelihood

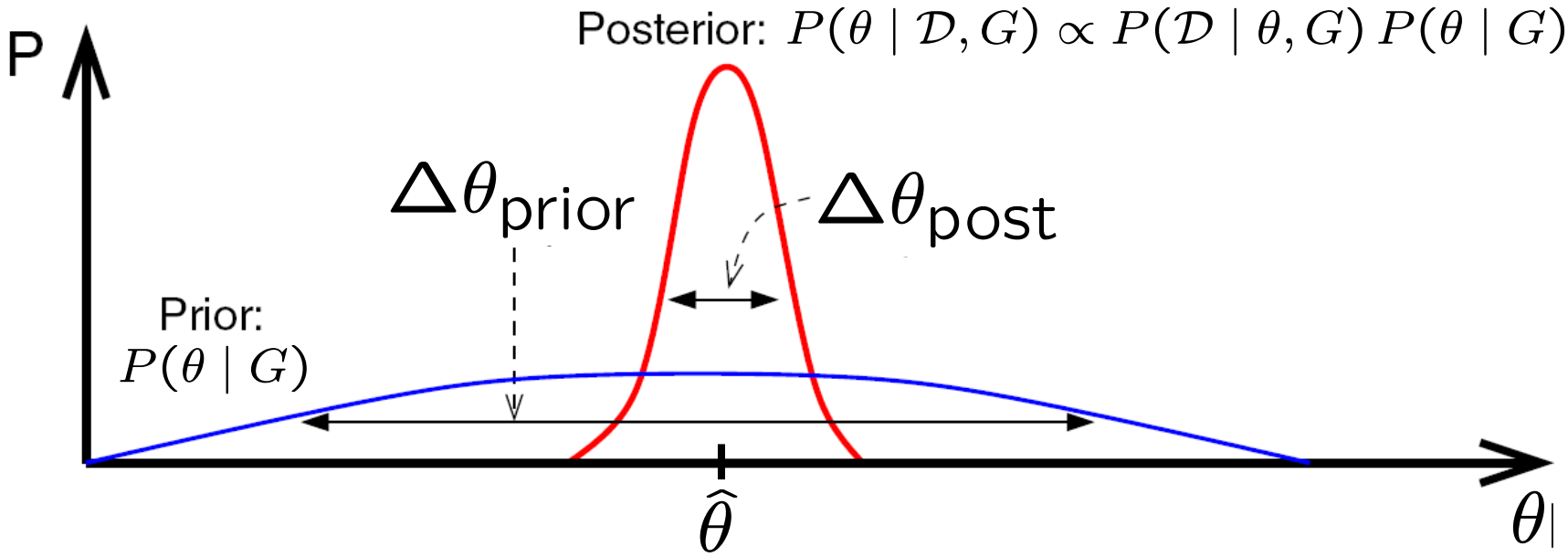
Marginal likelihood



$$P(\mathcal{D} \mid G) = \int P(\mathcal{D} \mid \theta, G) P(\theta \mid G) d\theta$$

$$\approx P(\mathcal{D} \mid \hat{\theta}, G) \frac{\Delta\theta_{\text{post}}}{\Delta\theta_{\text{prior}}}$$

The marginal likelihood penalizes complexity



$$P(\mathcal{D} \mid G) \approx P(\mathcal{D} \mid \hat{\theta}, G) \frac{\Delta\theta_{\text{post}}}{\Delta\theta_{\text{prior}}}$$

Likelihood at MLE, large if model fits well

Occam factor, small if model is overfitted

Notation

- We assume prior parameter independence and P($\theta \mid \mathcal{D}$, G) to be dominated by the likelihood, and set

$$P(\theta \mid G) = \prod_{i=1}^{\nu} P(\theta_i \mid G) = c^{\nu}$$

  where ν is the dimension of the parameter space.

- Define

$$\begin{aligned} E(\theta) &= -\log P(\mathcal{D} \mid \theta, G) \\ \hat{\theta} &= \operatorname{argmax}_{\theta} P(\mathcal{D} \mid \theta, G) \\ \mathbf{H} &= \left[\nabla_{\theta} \nabla_{\theta}^{t} E(\theta)\right]_{\theta=\hat{\theta}} \end{aligned}$$

- Taylor approximation: $E(\theta) \approx E(\hat{\theta}) + \frac{1}{2}(\theta - \hat{\theta})^{t} \mathbf{H} (\theta - \hat{\theta})$

Evidence approximation

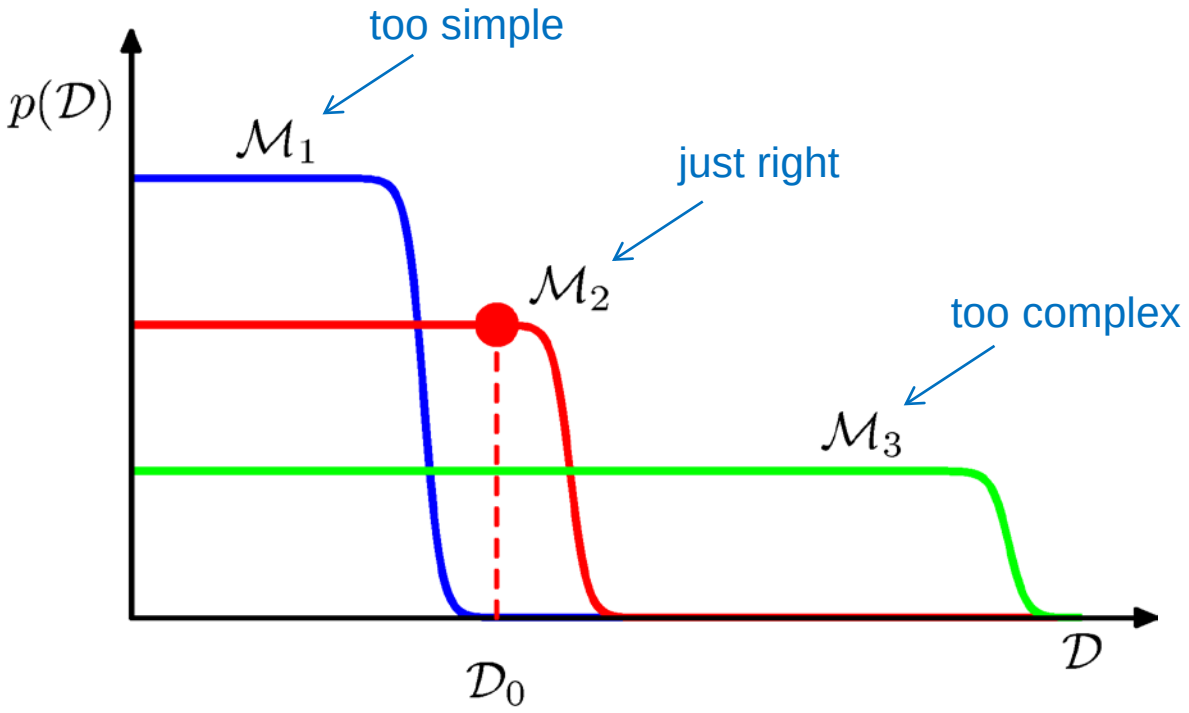
- Laplace approximation of the marginal likelihood gives

$$\log P(\mathcal{D} \mid G) \approx \underbrace{\log P(\mathcal{D} \mid \hat{\theta}, G)}_{\text{MLE score}} \underbrace{- \frac{1}{2} \log \det \mathbf{H} + \frac{\nu}{2} \log(2\pi c^2)}_{\text{penalty term / regularization term}}$$

- We can assume that the eigenvalues ϵ_i of H are proportional to sample size, $\epsilon_i \propto N$.
- If we further assume equal curvature along all eigendirections (isotropy), then $\epsilon_i \approx 2\pi c^2 N$, and we obtain the Bayesian Information Criterion (BIC):

$$\log P(\mathcal{D} \mid G) \approx \log P(\mathcal{D} \mid \hat{\theta}, G) - \frac{\nu}{2} \log N$$

Model complexity



Data sets

Search and score

Local likelihood decomposition

- Search-and-score is efficient for *decomposable* scores, because each vertex can be optimized separately.
- For fully observed models, the likelihood is decomposable:

$$\begin{aligned}
\ell(\theta) &= \log P(\mathcal{D} \mid G, \theta) \\
&= \sum_{l=1}^{N} \sum_{i=1}^{n} \log P\left(x_i^{(l)} \mid x_{\mathsf{pa}(i)}^{(l)}\right) \\
&= \sum_{i=1}^{n} \left[\sum_{l=1}^{N} \log P\left(x_i^{(l)} \mid x_{\mathsf{pa}(i)}^{(l)}\right) \right]
\end{aligned}$$

Fully observed ML estimates

- For a given graph structure G and fully observed discrete random variables, the MLEs of the parameters

$$\theta_{x_i, x_{\mathrm{pa}(i)}} = P\left(x_i \mid x_{\mathrm{pa}(i)}\right)$$

  are

$$\hat{\theta}_{x_i, x_{\mathrm{pa}(i)}} = N(x_i, x_{\mathrm{pa}(i)}) / N(x_{\mathrm{pa}(i)})$$

  where N(x) is the number of times that x appears in the data set $\mathcal{D}$ = {$\mathbf{x}^{(1)}$, ..., $\mathbf{x}^{(N)}$}.

- The log-likelihood at this maximum is

$$\ell(\hat{\theta}) = \sum_i \sum_{x_i, x_{\mathrm{pa}(i)}} N(x_i, x_{\mathrm{pa}(i)}) \log \theta_{x_i, x_{\mathrm{pa}(i)}}$$

Local BIC decomposition

- Recall the BIC approximation of the model posterior:

$$\log P(G \mid \mathcal{D}) \approx \log P(\mathcal{D} \mid G, \hat{\theta}) - \frac{\dim(G)}{2} \log N$$

- The BIC score of the BN (G, θ) decomposes as

$$\mathrm{BIC}(G, \theta \mid \mathcal{D}) = \ell_G(\theta) - \frac{\dim(G)}{2} \log N$$

$$= \sum_{i=1}^{n} \mathrm{BIC}_i \left((X_i, X_{\mathrm{pa}(i)}), \theta_{X_i, X_{\mathrm{pa}(i)}} \mid \mathcal{D} \right)$$

$$= \sum_{i=1}^{n} \left[\sum_{x_i, x_{\mathrm{pa}(i)}} N(x_i, x_{\mathrm{pa}(i)}) \log \theta_{x_i, x_{\mathrm{pa}(i)}} - \frac{\dim(X_i, X_{\mathrm{pa}(i)})}{2} \log N \right]$$

Search and score

- Any local search procedure can exploit local score decompositions:
  - If a single edge is modified, few local computations suffice to find the optimal parameters (because most remain unchanged) and to update the global score (because most local contributions are unaffected).

- If the training data is incomplete (due to missing data and/or hidden variables), then the BIC score does no longer decompose over the vertices of the BN.

Model Selection EM (MS-EM), or Structural EM (SEM)

- Suppose we have observed r. v. **X** and hidden r. v. **Z**.
- We want to maximize a score (e.g., BIC)

$$S_X(G, \theta \mid \mathcal{D}) = \log P(X \mid G, \theta) - \text{Penalty}(G, \theta, X)$$

  over model structures G and parameters θ.
- We assume that we can solve this optimization problem, if we have fully observed data.

- Idea: We maximize the *expected* score w.r.t. P(Z | $\mathcal{D}$).

Structural EM

- If (G', θ') is our current model estimate, then we maximize the expected score

$$Q(G, \theta \mid G', \theta') = \mathrm{E}\left[\log P(X, Z \mid G, \theta) - \mathrm{Penalty}(G, \theta, X)\right]$$

  where the expectation is w.r.t. P(Z | $\mathcal{D}$, G', θ').

- **Theorem**: If Q(G, θ | G', θ') > Q(G', θ' | G', θ'), then S_X(G, θ) > S_X(G', θ').

- Algorithm:
  - Choose G^0, θ^0 at random.
  - For n = 0, 1, 2, ... until convergence:
    - Compute the expected statistics w.r.t. P(Z | $\mathcal{D}$, G^n, θ^n)
    - Find a model (G^{n+1}, θ^{n+1}) that maximizes Q(G^{n+1}, θ^{n+1} | G^n, θ^n)

Expected BIC score

- The expected BIC score decomposes over vertices:

$$Q(G, \theta \mid G', \theta') = \sum_{i=1}^{n} Q_i(X_i, X_{\mathrm{pa}(i)} \mid G', \theta')$$

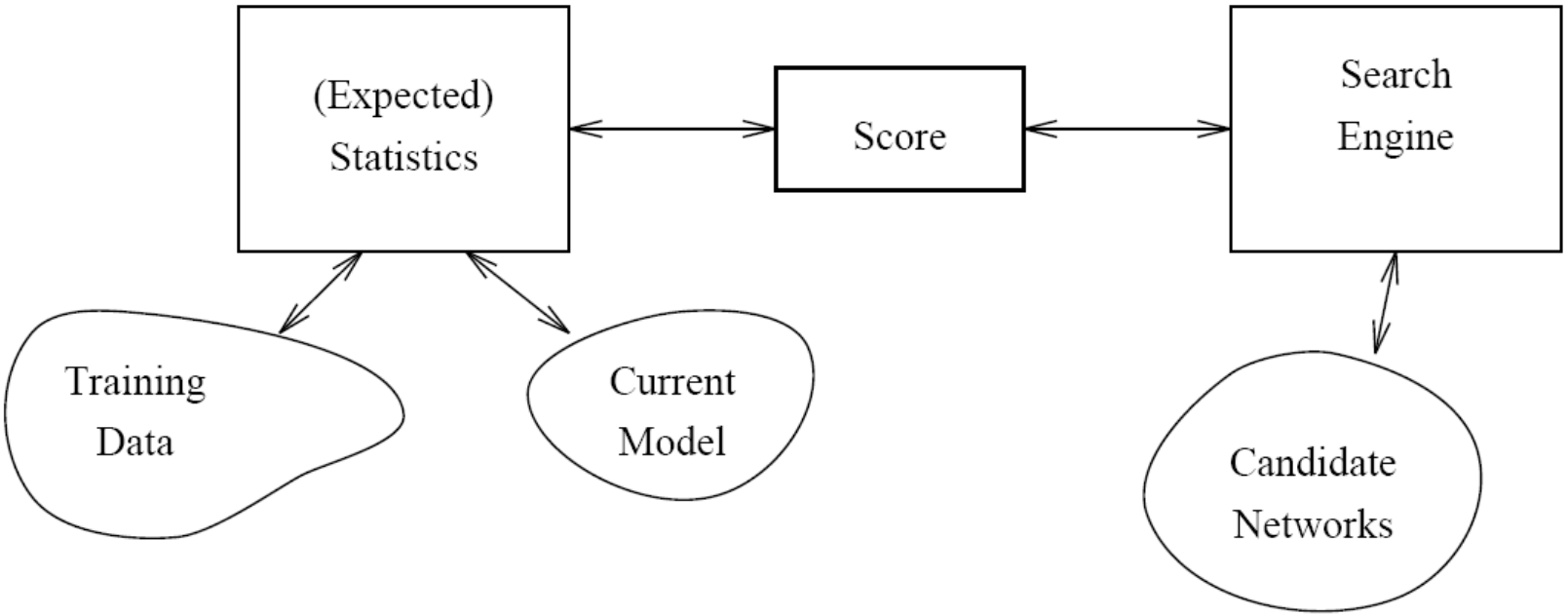
where

$$Q_i(X_i, X_{\mathrm{pa}(i)} \mid G', \theta') = \sum_{x_i, x_{\mathrm{pa}(i)}} \mathsf{E}_{P'} \left[N(x_i, x_{\mathrm{pa}(i)}) \right] \log \theta_{x_i, x_{\mathrm{pa}(i)}} - \frac{\dim(X_i, X_{\mathrm{pa}(i)})}{2} \log N$$

- Analogous to the fully observed case, the MLEs are

$$\hat{\theta}_{x_i, x_{\mathrm{pa}(i)}} = \mathsf{E}_{P'} \left[N(x_i, x_{\mathrm{pa}(i)}) \right] / \mathsf{E}_{P'} \left[N(x_{\mathrm{pa}(i)}) \right]$$

Implementation



D-separation and graph separation

Recall: Conditional independence

- Let A, B, and C be non-intersecting subsets of nodes in a directed graph.
- We say that A and B are conditionally independent given C if

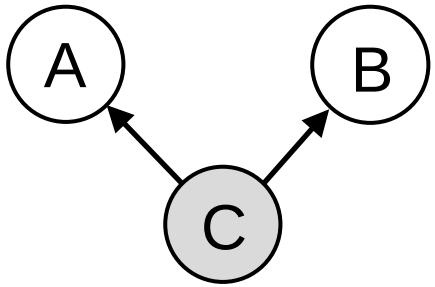
$$P(A, B \mid C) = P(A \mid C)\, P(B \mid C)$$

- Notation:

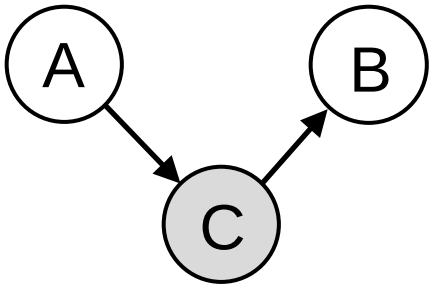
$$A \perp B \mid C$$

Recall: Three basic examples

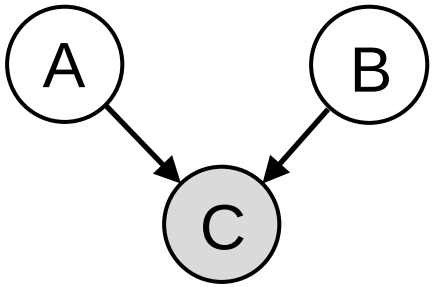
explaining away



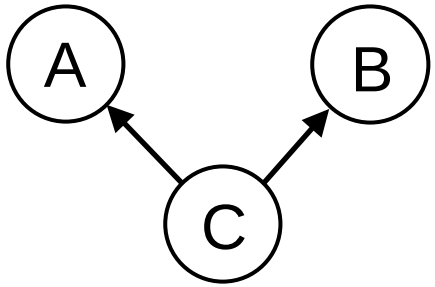
$A \perp B \mid C$



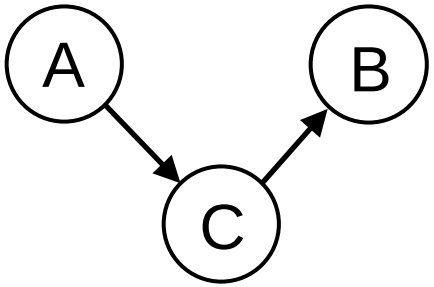
$A \perp B \mid C$



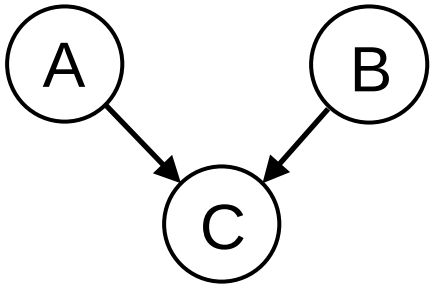
$A \not\perp B \mid C$



$A \not\perp B$



$A \not\perp B$



$A \perp B$

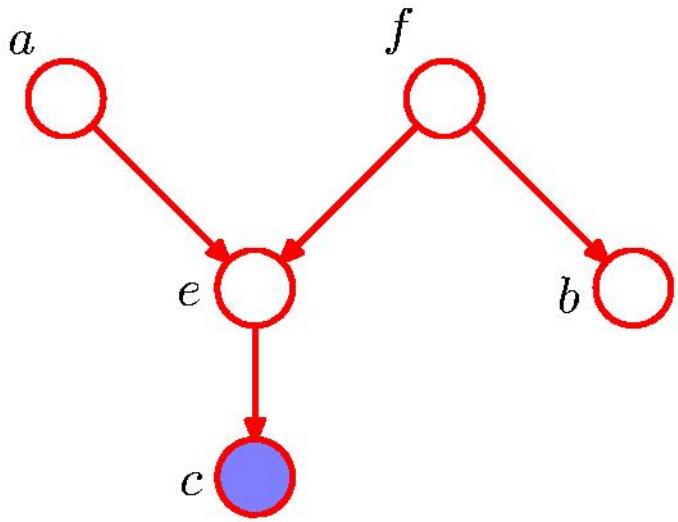
D-separation

- A path from A to B is blocked by C if it contains a node such that either
  a) the arrows on the path meet either head-to-tail or tail-to-tail at the node, and the node is in the set C, or
  b) the arrows meet head-to-head at the node, and neither the node, nor any of its descendants, are in C. ("no explaining away")
- If all paths from A to B are blocked by C, A is said to be d-separated from B by C.

- **Theorem** (Verma & Pearl, 1988): A is d-separated from B by C if, and only if, the joint distribution over all variables in the graph satisfies

$$A \perp B \mid C$$

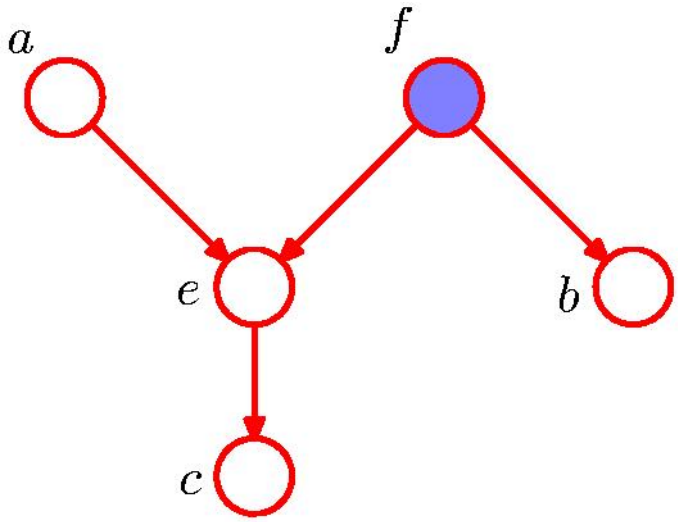
D-separation: Example



$$a \not\perp b \mid c$$

Is the *a*-*b* path blocked by *e*?

No, because *c* is a descendant of the head-to-head node *e* (case b).

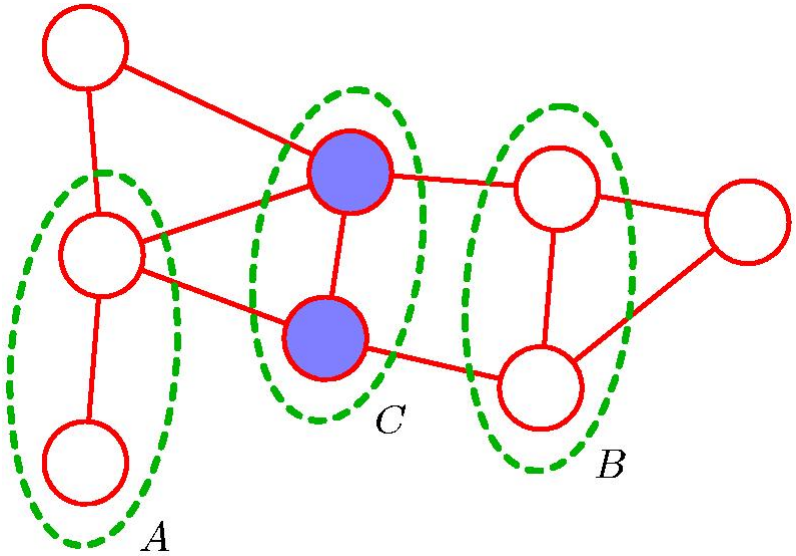


$$a \perp b \mid f$$

Is the *a*-*b* path blocked by *f*?

Yes, because the arrows meet tail-to-tail at *f* (case a).

Graph separation in Markov random fields



- **Theorem**: A is graph separated from B by C if, and only if, the joint distribution over all variables in the graph satisfies

$$A \perp B \mid C$$

Directed versus undirected graphical models

- The directed graphical models (Bayesian networks) are exactly the joint probability distributions for which d-separation holds (Verma & Pearl, 1988):

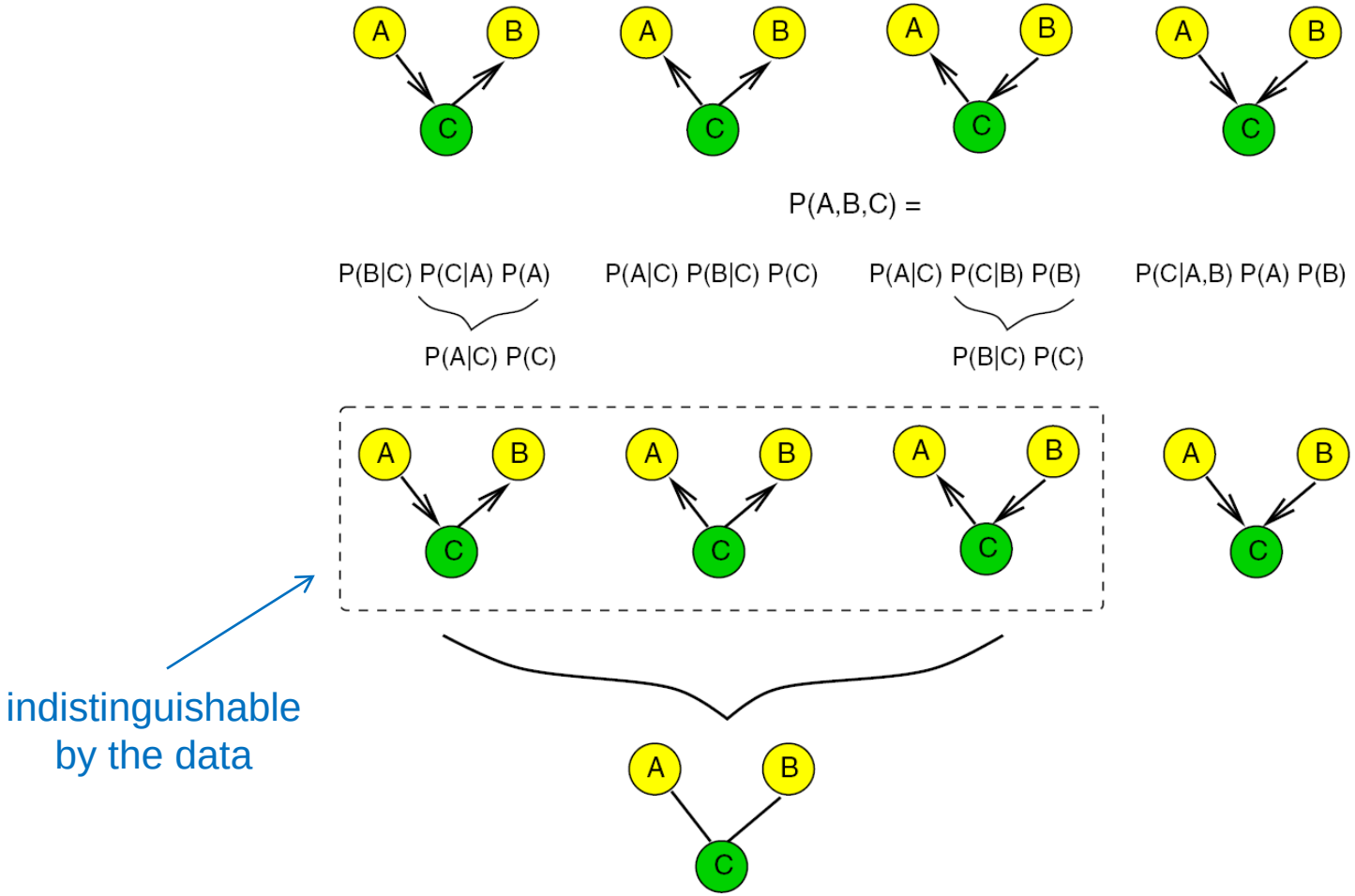
$$P(X_1, \ldots, X_L) = \prod_{n=1}^{L} P\left(X_n \mid X_{\mathrm{pa}(n)}\right) \iff \text{d-separation}$$

- The (positive) undirected graphical models (Markov random fields) are exactly the joint probability distributions for which graph separation holds (Hammersley-Clifford, 1971):

$$P(X_1, \ldots, X_L) = \frac{1}{Z} \prod_{\{\text{max. cliques } C\}} \psi_C(X_C) \iff \text{graph separation}$$

PC algorithm

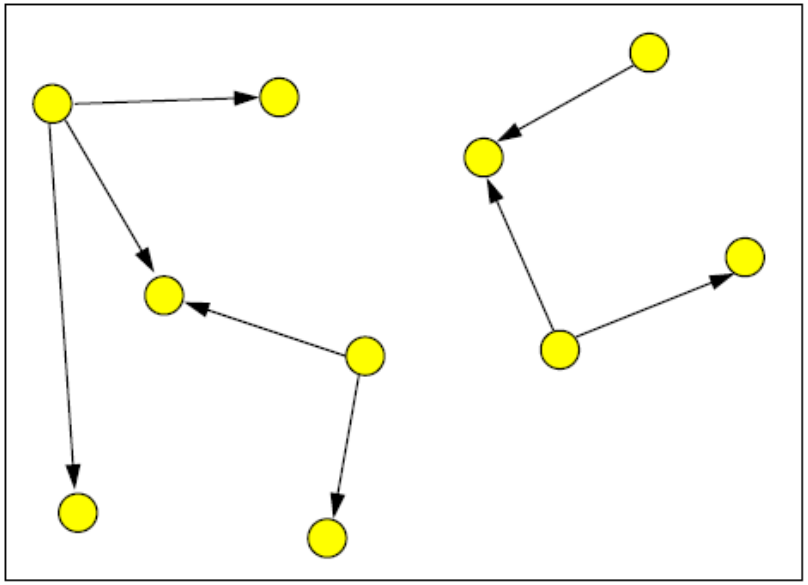
Equivalence classes

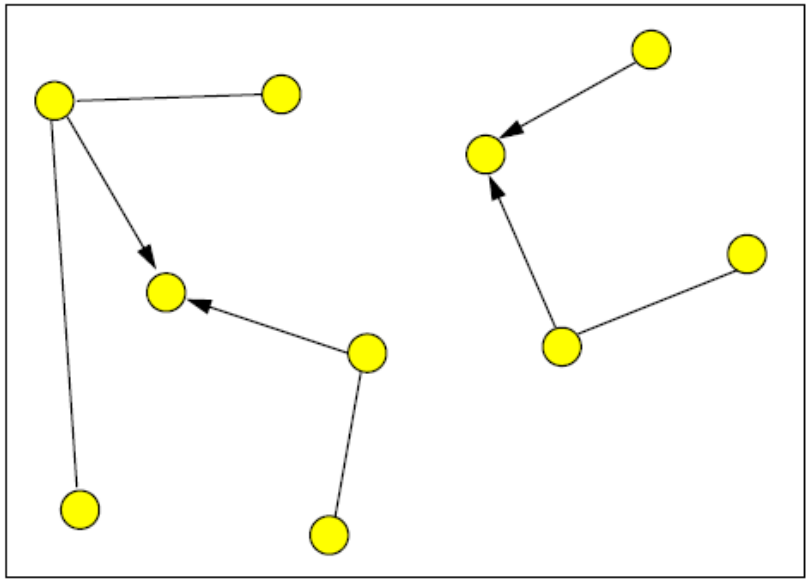


Equivalent Bayesian networks

- Equivalent Bayesian networks encode the same conditional independencies and have (under fairly general conditions) the same likelihood.
- Thus, we can only learn *equivalence classes* of Bayesian networks from data.
- Equivalence classes are represented by partially directed acyclic graphs (PDAGs).
- **Theorem**: Two graphs are equivalent if and only if they agree on their skeleton and on all colliders.
  - The skeleton is the induced undirected graph obtained by dropping edge directions.
  - A collider (or v-structure) is a converging edge pair, such that the parents are not adjacent.

DAG and PDAG

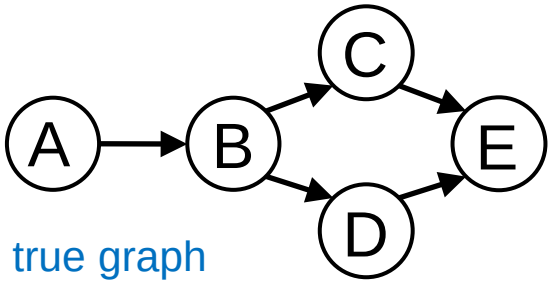




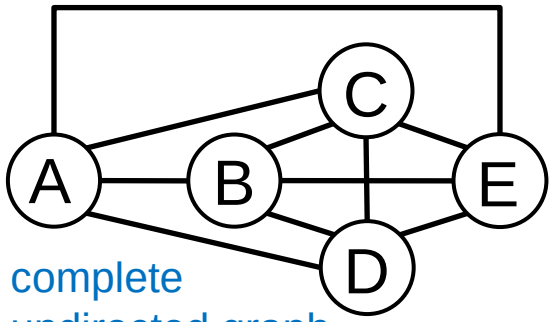
PC algorithm

- Input:
  - Vertex set V
  - Conditional independence information (estimated from data, for example using statistical tests of conditional independence)
- Step 1:
  - Start with complete undirected graph and successively “thin” it by testing conditional independencies of increasing order.
  - Output: skeleton, separation sets
- Step 2:
  - Direct all edges that can be directed
  - Output: PDAG

Example (step 1)

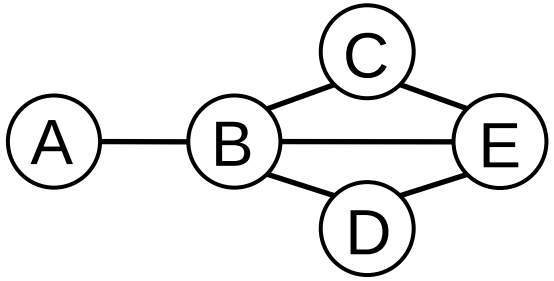


true graph

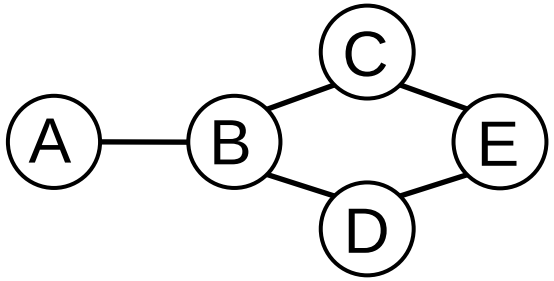


complete
undirected graph

$$n = 1: \quad A \perp C \mid B \quad A \perp D \mid B$$
$$A \perp E \mid B \quad C \perp D \mid B$$



$$n = 2: \quad B \perp E \mid \{C, D\}$$



Summary

- Model selection involves defining a criterion of optimality and a procedure for finding the optimal model.
- Model selection in Bayesian tree models is efficient.
- Computing the marginal likelihood is challenging; the evidence approximation (BIC) is a popular choice.
- Bayesian network learning using search-and-score is efficient for decomposable scores (e.g., BIC).
- D-separation and graph separation characterize, resp., directed and undirected probabilistic graphical models.
- The PC algorithm is based on testing conditional independence statements.

Learning Bayesian networks

	Fully observed data	**Missing data / hidden variables**
Known graph structure	Sample statistics	EM algorithm Gradient ascent Sampling Variational inference
Unknown graph structure	Search-and-score (BIC) PC algorithm Sampling	Structural EM Sampling

Learning Bayesian networks

	Fully observed data	Missing data / hidden variables
Known graph structure	Sample statistics easy	EM algorithm Gradient ascent Sampling Variational inference doable
Unknown graph structure	Search-and-score (BIC) PC algorithm Sampling doable	Structural EM Sampling hard

References

- Bishop CM. Pattern Recognition and Machine Learning. Springer, 2007. Sections 3.4, 3.5.
- Chow CK, Liu CN (1968). Approximating discrete probability distributions with dependence trees. *IEEE Trans Inform Theor* 14(3):462-467.
- Friedman N. Learning belief networks in the presence of missing values and hidden variables. ICML 1997.
- Spirtes P, Glymour C, Scheines R. Causation, Prediction, and Search. MIT Press, 2000. Chapter 5.
- Kalisch T, Bühlmann P (2007). Estimating high-dimensional directed acyclic graphs with the PC algorithm. *J Mach Learn Res* 8:613-636.
- Chickering DM, Heckerman D (1997). Efficient Approximations for the Marginal Likelihood of Bayesian Networks with Hidden Variables. *Machine Learning* 29:181-212.
- Husmeier D, Dybowski R, Roberts S (eds.). Probabilistic Modeling in Bioinformatics and Medical Informatics. Section 2.3.2.