Learning Bayesian Networks

Harald Steck
MIT AI Lab
harald@ai.mit.edu

December 3, 2002

Outline

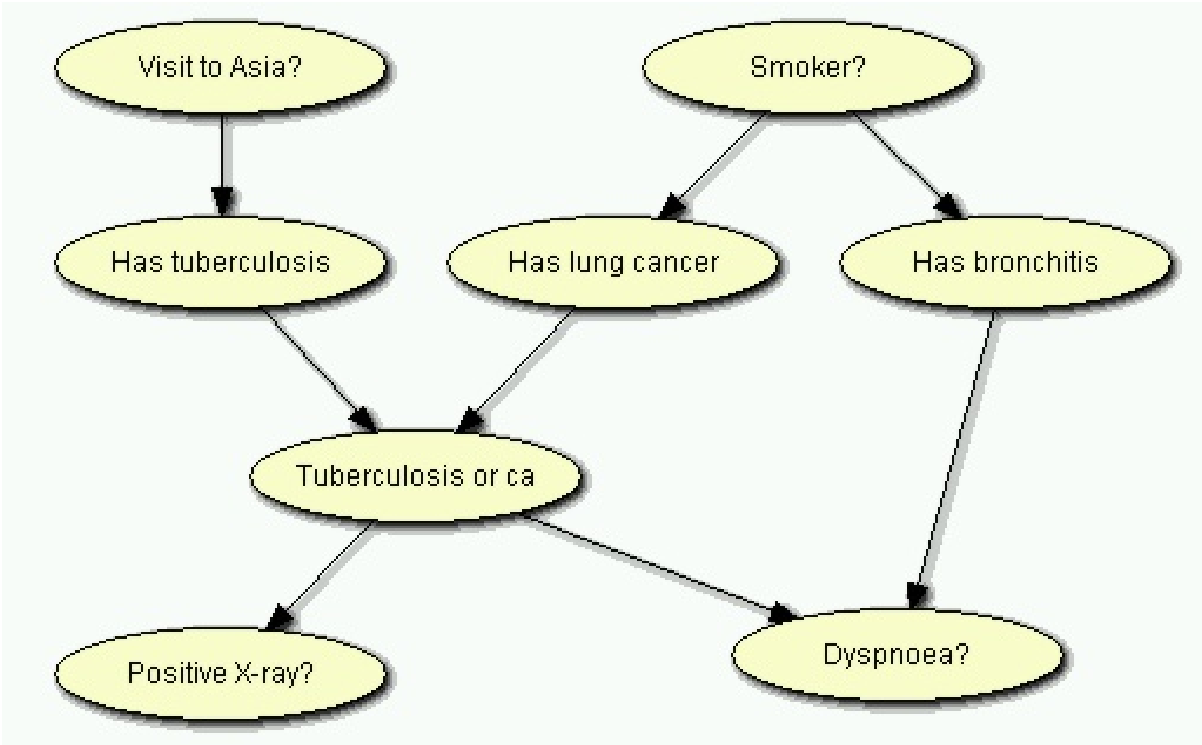
- Bayesian networks (BN)
- Learning discrete Bayesian Networks
  - Scoring Functions
  - Search Strategies
- Applications

How to obtain Bayesian Networks ?

- construct them manually: experts / knowledge needed
- learn them from data
- combine prior knowledge and data

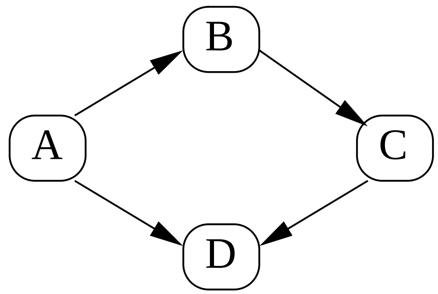
Properties of Bayesian Networks

- qualitative:
  - graph structure visualizes relevant (in)dependencies among random variables in a domain
  - interpretation in causal manner: requires add'l assumptions
- quantitative: make predictions (inference)
- Example (Visit to Asia):



Bayesian Networks (more formal)

- network structure: directed acyclic graph (DAG)



  - directed edge: asymmetric relations (but not necessarily causal)
  - missing edges represent conditional independences (d-separation criterion)

- parameters: conditional probabilities
  $p(A, B, C, D) = p(D|C, \cancel{B}, A) \cdot p(C|B, \cancel{A}) \cdot p(B|A) \cdot p(A)$

- BN describes probability distribution over n variables in a modular way:

$$p(X) = \prod_{i=1}^{n} p(X_i|\Pi_i)$$

How to model conditional probability distributions ?

- discrete variables (tables)
- continuous variables:
  - multivariate Gaussian (linear regression)

$$\boxed{A} \longrightarrow \boxed{B}$$

$$\begin{aligned} p(A)&: \quad A \sim N(\mu_A, \sigma_A^2) \\ p(B|A)&: \; B \sim N(\mu_B + \theta_{A,B} \cdot A,\ \sigma_B^2) \end{aligned}$$

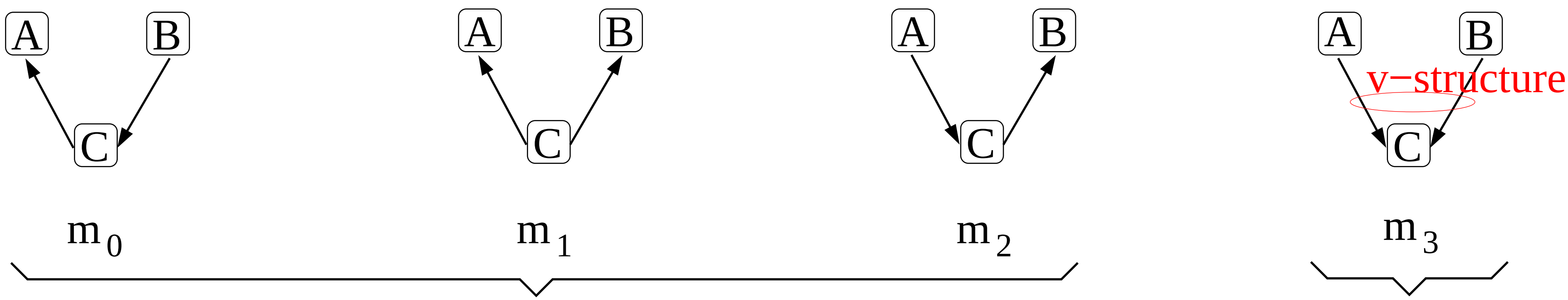
  - nonlinear relations:
    - nonlinear regression

$$p(B|A):\ B \sim N(\mu_B + \theta_{A,B,1} \cdot A + \theta_{A,B,2} \cdot A^2,\ \sigma_B^2)$$

    - other models: neural networks + noise, ...
- both discrete and continuous variables

Markov-Equivalence

- applies to discrete BNs and continuous Gaussian BNs
- Example:



A and B dependent (marginally)

A and B conditionally independent given C

A and B independent

A and B dependent given C

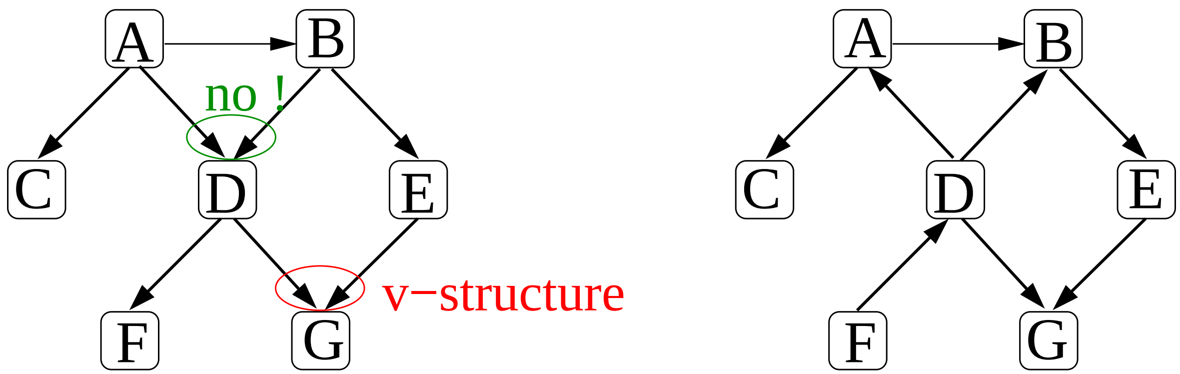
$$p(A,B,C) = \underbrace{p(A|C)\,p(C|B)\,p(B)}_{m_0} = \underbrace{p(A|C)\,p(B|C)\,p(C)}_{m_1} = \underbrace{p(C|A)\,p(B|C)\,p(A)}_{m_2}$$

- DAGs can be partitioned into equivalence classes that represent the same conditional independences

Markov-Equivalence (cont'd)

- Two DAGs are Markov-equivalent iff they have
  - the same edges when ignoring their orientations
  - and the same v-structures ($\searrow\swarrow$)

Example: 2 Markov-equivalent DAGs



Outline

- Bayesian networks
- Learning discrete Bayesian Networks
  - Scoring Functions
  - Search Strategies
- Applications

- in the following assumed:
  - no hidden variables
  - no missing data
  - discrete BNs (blue=discrete)

Scoring Functions

- Maximum Likelihood

$$\widehat{\theta}_{x_i|\pi_i} = \frac{N_{x_i,\pi_i}}{N_{\pi_i}}$$

$$l(\widehat{\theta}_m) = \log L(\theta_m) = \sum_i \sum_{x_i,\pi_i} N_{x_i,\pi_i} \log \frac{N_{x_i,\pi_i}}{N_{\pi_i}}$$

not useful for model selection: over-fitting

Scoring Functions (cont'd)

- BIC (Bayesian Information Criterion, aka (Jeffreys-)Schwarz Criterion)

$$f_{\text{BIC}}(m) = l(\widehat{\theta}_m) - \frac{1}{2}\,|\widehat{\theta}_m|\ \log N$$

  - trade-off between goodness of fit and model complexity
  - BIC coincides with MDL (Minimum Description Length)

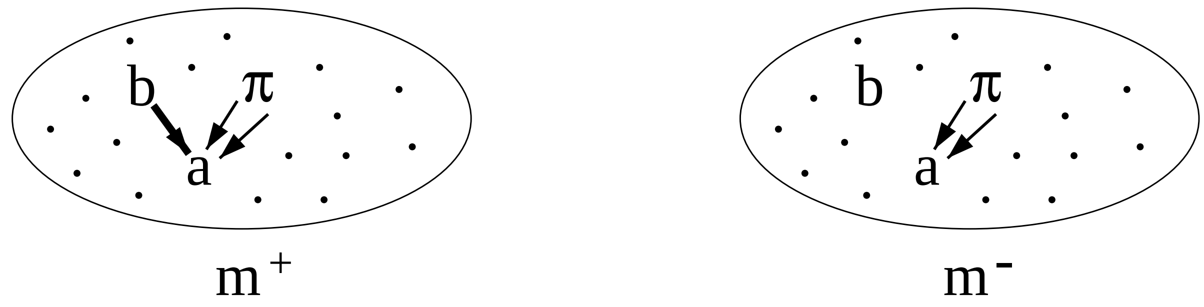
- AIC (Akaike Information Criterion)

$$f_{\text{AIC}}(m) = l(\widehat{\theta}_m) - |\widehat{\theta}_m|$$

where the number of independent parameters is

$$|\widehat{\theta}_m| = \sum_i \quad (|X_i| - 1) \cdot \underbrace{|\Pi_i|}_{=\prod_{X \in \Pi_i} |X|}$$

Score Difference



- compare two graphs m^+ and m^- that differ in one edge only
- Example: BIC for discrete variables

$$\begin{aligned}
\Delta f(m^+, m^-) &= f(m^+) - f(m^-) \\
&= \sum_{a,b,\pi} N_{a,b,\pi} \log \frac{N_{a,b,\pi} N_\pi}{N_{a,\pi} N_{b,\pi}} - \frac{1}{2} d_{\mathsf{DF}} \log N \\
&= g(A, B | \Pi)
\end{aligned}$$

... independent of remaining variables

where d_{DF} are the degrees of freedom:

$$d_{\mathsf{DF}} = |\theta_{m^+}| - |\theta_{m^-}| = (|A| - 1) \cdot (|B| - 1) \cdot \underbrace{|\Pi|}_{=\prod_{X \in \Pi} |X|}$$

Score Difference (cont'd)

- **Conditional Independences** (which are represented by BNs):

$g(A,B\|\Pi) < 0$	... absence of egde $A \leftarrow B$ favored given Π
	... A independent of B given Π
$g(A,B\|\Pi) > 0$	... presence of egde $A \leftarrow B$ favored given Π
	... A dependent on B given Π

- **Markov equivalence**:
  - data cannot help distinguish among Markov equivalent DAGs
  - a "local" property of equivalent DAGs: an edge $A \leftarrow B$ can be reversed if $\Pi_A \setminus \{B\} = \Pi_B \setminus \{A\}$
  - for BIC: $g(A,B|\Pi) = g(B,A|\Pi)$
  - hence BIC assigns the same score to equivalent DAGs

Outline

- Bayesian networks
- Learning discrete Bayesian Networks
  - Scoring Functions
  - Search Strategies
- Applications

Search Strategies

- in discrete or continuous Gaussian BNs:
  - data can help distinguish only among equivalence classes
  - search in space of equivalence classes is thus most appropriate, but very involved

- search in space of DAGs
- number of DAGs with n variables: $2^{\binom{n}{2}} < \#\text{DAGs} \leq 3^{\binom{n}{2}}$
- finding optimal DAG w.r.t. a scoring function f is NP-hard

- resort to approximate search strategies

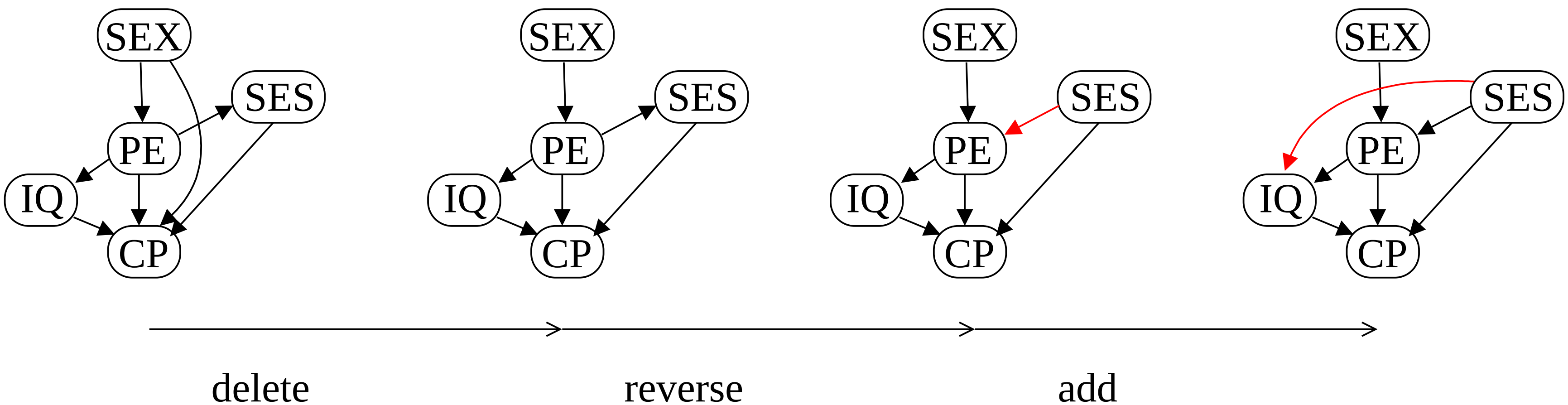
Local Search

- general-purpose search strategy
- choose a starting graph
- proceed through search space along a sequence of neighboring DAGs, guided by scoring function
- DAGs differing in a single edge may be defined as neighbors
- hence, 3 possible transitions in local search:
  - add an edge (if permissible)
  - delete an edge
  - reverse the orientation of an edge (if permissible)
- score difference due to transition: $\Delta = f(m_{\text{new}}) - f(m_{\text{old}})$

Local Search and Greedy Hill Climbing

- choose transition that maximizes Δ
- repeat until $\Delta < 0$ for all permissible steps
- result: graph that is a local optimum

- Example:



Local Search and Simulated Annealing

- general purpose optimization procedure to avoid local optima
- inspired by cooling down an ensemble of particles (statistical physics)
- temperature of system: T
- procedure
  - start with high temperature and lower it slowly over time
  - randomly choose a transition
  - make transition with probability $p(\Delta) = \min\{1, \exp(\Delta/T)\}$

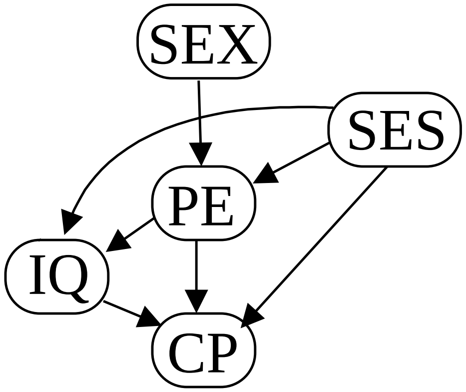
- theory: finds global minimum of $-f$ with probability 1 if starting temperature is sufficiently high and is lowered sufficiently slowly
- practice: limited computation time, may only find a local optimum

Outline

- Bayesian networks
- Learning discrete Bayesian Networks
  - Scoring Functions
  - Search Strategies
- Applications

Analysis of Questionnaires

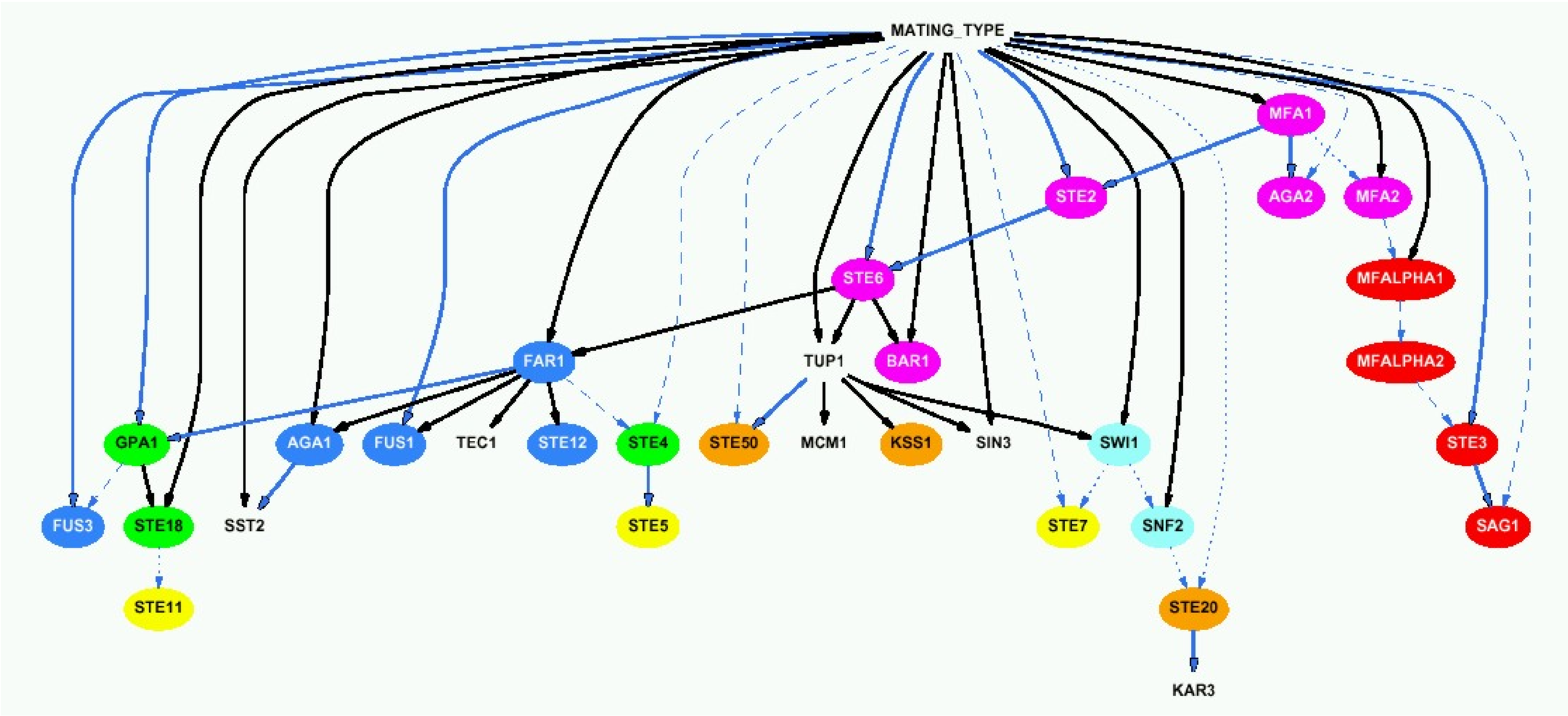
- find relevant conditional dependencies
- e.g., Wisconsin High-School Students (Sewell and Shah, 1968):
  - survey among 10,318 students
  - learn BN from that data:



SES: socioeconomic status
SEX: gender of student
PE: parental encouragement
CP: college plans
IQ: intelligence quotient

Analysis of Noisy Measurements

- e.g., gene expression data from bio-tech labs
  - graph: recovery of regulatory networks



(Hartemink et al., 2002)

  - prediction: what is the most informative next experiment to be conducted (active learning)?