



Lecture 31 of 42

Inference and Software Tools 2 Discussion: MP6, BNJ

Monday, 05 November 2007

William H. Hsu
Department of Computing and Information Sciences, KSU

KSOL course page: <http://snipurl.com/v9v3>
Course web site: <http://www.kddresearch.org/Courses/Fall-2007/CIS730>
Instructor home page: <http://www.cis.ksu.edu/~bhsu>

Reading for Next Class:
Chapter 15, Russell & Norvig 2nd edition



Lecture Outline

- Wednesday's Reading: Sections 14.6 – 14.8, R&N 2e, Chapter 15
- Friday: Sections 18.1 – 18.2, R&N 2e
- Today: Inference in Graphical Models
 - * Bayesian networks and causality
 - * Inference and learning
 - * BNJ interface (<http://bnj.sourceforge.net>)
 - * Causality

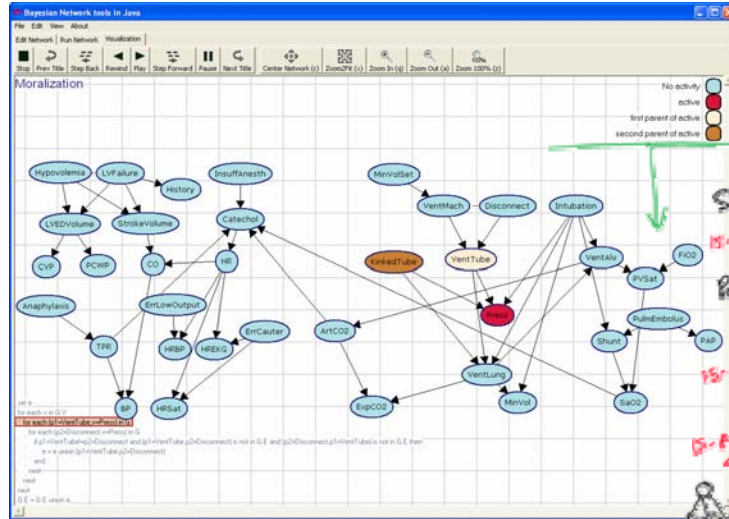
Genetic
& Evolutionary
Computation

Artificial
Neural
Networks





BNJ Visualization: Pseudo-Code Annotation (Code Page)

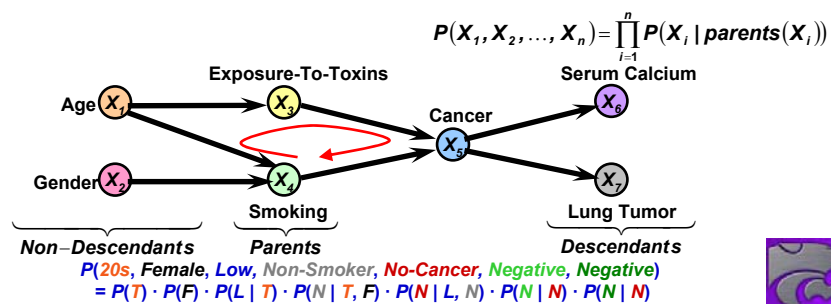


© 2004 KSU BNJ Development Team



Graphical Models Overview [1]: Bayesian Networks

- **Conditional Independence**
 - * X is conditionally independent (CI) from Y given Z (sometimes written $X \perp Y | Z$) if $P(X | Y, Z) = P(X | Z)$ for all values of $X, Y,$ and Z
 - * Example: $P(\text{Thunder} | \text{Rain}, \text{Lightning}) = P(\text{Thunder} | \text{Lightning}) \Leftrightarrow T \perp R | L$
- **Bayesian (Belief) Network**
 - * **Acyclic directed graph** model $B = (V, E, \Theta)$ representing CI assertions over Θ
 - * **Vertices** (nodes) V : denote events (each a random variable)
 - * **Edges** (arcs, links) E : denote conditional dependencies
- Markov Condition for BBNs (Chain Rule):
- Example BBN





Graphical Models Overview [3]: Inference Problem

Typically, we are interested in
the posterior joint distribution of the query variables \mathbf{Y}
given specific values e for the evidence variables \mathbf{E}

Let the hidden variables be $\mathbf{H} = \mathbf{X} - \mathbf{Y} - \mathbf{E}$

Then the required summation of joint entries is done by summing out
the hidden variables:

$$P(\mathbf{Y}|\mathbf{E}=e) = \alpha P(\mathbf{Y}, \mathbf{E}=e) = \alpha \sum_{\mathbf{h}} P(\mathbf{Y}, \mathbf{E}=e, \mathbf{H}=\mathbf{h})$$

The terms in the summation are joint entries because \mathbf{Y} , \mathbf{E} , and \mathbf{H}
together exhaust the set of random variables

Obvious problems:

- 1) Worst-case time complexity $O(d^m)$ where d is the largest arity
- 2) Space complexity $O(d^m)$ to store the joint distribution
- 3) How to find the numbers for $O(d^m)$ entries???

Multiply-connected case: exact, approximate inference are #P-complete

Adapted from slides by S. Russell, UC Berkeley

<http://aima.cs.berkeley.edu/>

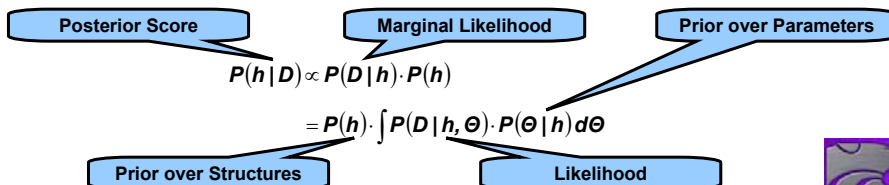


Other Topics in Graphical Models [2]: Learning Structure from Data

- General-Case BBN Structure Learning: Use Inference to Compute Scores
- Optimal Strategy: Bayesian Model Averaging
 - * Assumption: models $h \in H$ are mutually exclusive and exhaustive
 - * Combine predictions of models in proportion to marginal likelihood
 - Compute conditional probability of hypothesis h given observed data D
 - i.e., compute expectation over unknown h for unseen cases
 - Let $h =$ structure, parameters $\Theta =$ CPTs

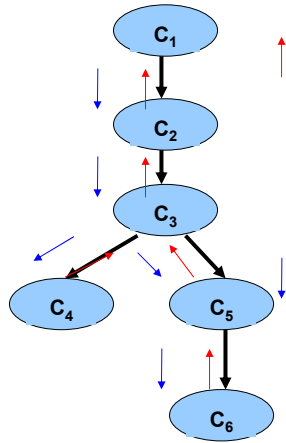
$$P(\bar{x}^{(m+1)} | D) = P(x_1, x_2, \dots, x_n | \bar{x}^{(1)}, \bar{x}^{(2)}, \dots, \bar{x}^{(m)})$$

$$= \sum_{h \in H} P(\bar{x}^{(m+1)} | D, h) \cdot P(h | D)$$





Propagation Algorithm in Singly-Connected Bayesian Networks – Pearl (1983)



Upward (child-to-parent) λ messages

$\Psi'(C_i)$ modified during λ message-passing phase

Downward π messages

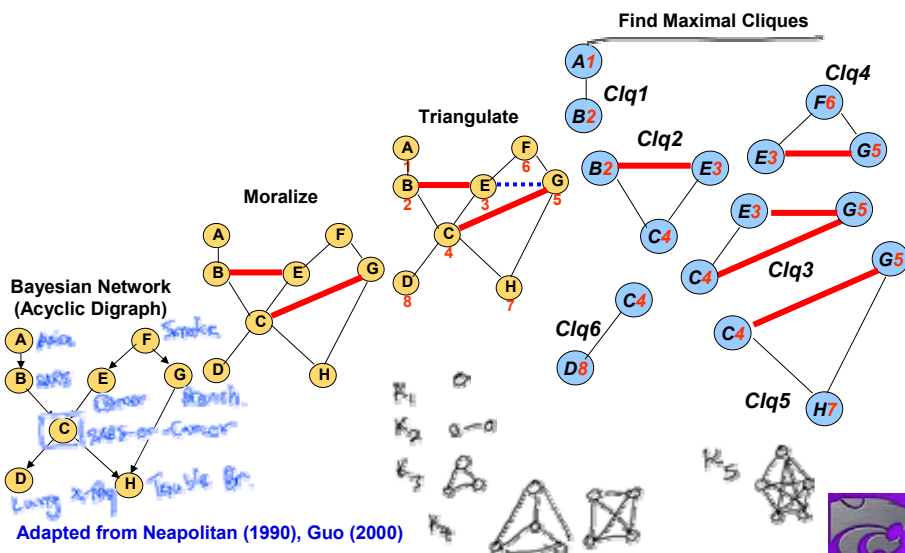
$P'(C_i)$ is computed during π message-passing phase

Multiply-connected case: exact, approximate inference are $\#P$ -complete (counting problem is $\#P$ -complete iff decision problem is NP -complete)

Adapted from Neapolitan (1990), Guo (2000)



Inference by Clustering [1]: Graph Operations (Moralization, Triangulation, Maximal Cliques)



Adapted from Neapolitan (1990), Guo (2000)



Inference by Clustering [2]: Function Tree – Lauritzen & Spiegelhalter (1988)

Input: list of cliques of triangulated, moralized graph G_u

Output:

Tree of cliques

Separator nodes S_i ,

Residual nodes R_i and potential probability $\Psi(\text{Clq}_i)$ for all cliques

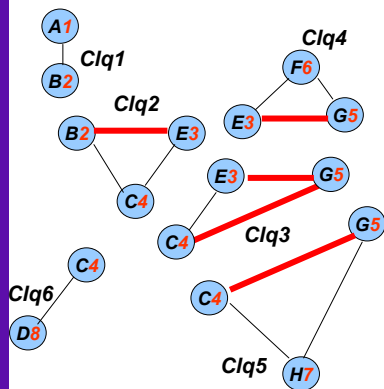
Algorithm:

1. $S_i = \text{Clq}_i \cap (\text{Clq}_1 \cup \text{Clq}_2 \cup \dots \cup \text{Clq}_{i-1})$
2. $R_i = \text{Clq}_i - S_i$
3. If $i > 1$ then identify a $j < i$ such that Clq_j is a parent of Clq_i
4. Assign each node v to a unique clique Clq_i that $v \cup c(v) \subseteq \text{Clq}_i$
5. Compute $\Psi(\text{Clq}_i) = \prod_{v \in \text{Clq}_i} P(v | c(v))$ {1 if no v is assigned to Clq_i }
6. Store Clq_i , R_i , S_i , and $\Psi(\text{Clq}_i)$ at each vertex in the tree of cliques

Adapted from Neapolitan (1990), Guo (2000)



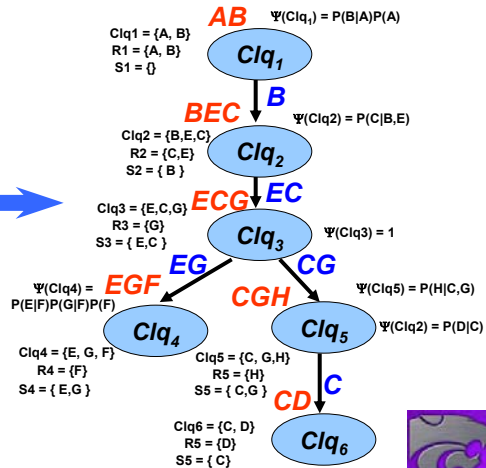
Inference by Clustering [3]: Clique-Tree Operations



R_i : residual nodes

S_i : separator nodes

$\Psi(\text{Clq}_i)$: potential probability of Clique i

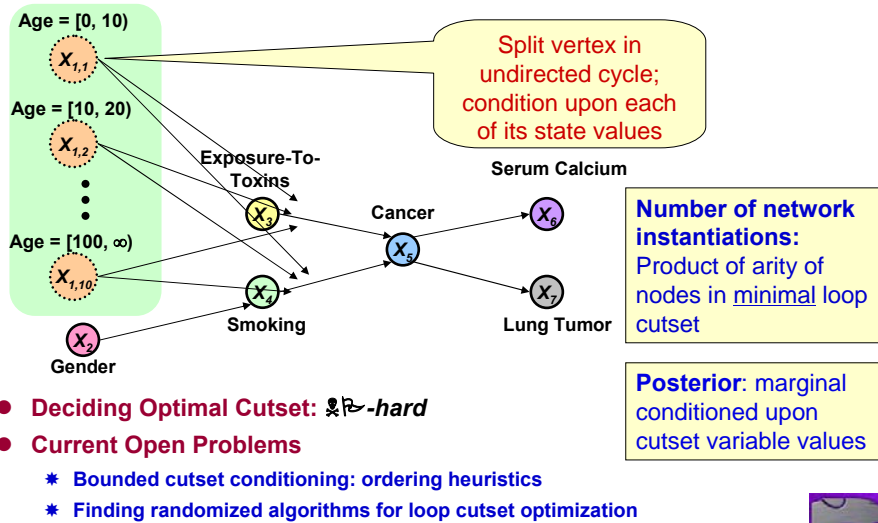


Adapted from Neapolitan (1990), Guo (2000)





Inference by Loop Cutset Conditioning



Inference by Variable Elimination [1]: Intuition

Enumeration is inefficient: repeated computation

e.g., computes $P(J = true|a)P(M = true|a)$ for each value of e

Variable elimination: carry out summations right-to-left, storing intermediate results (factors) to avoid recomputation

$$\begin{aligned}
 P(B|J = true, M = true) &= \alpha \underbrace{P(B)}_B \underbrace{\sum_e P(e)}_E \underbrace{\sum_a P(a|B, e)}_A \underbrace{P(J = true|a)}_J \underbrace{P(M = true|a)}_M \\
 &= \alpha P(B) \sum_e P(e) \sum_a P(a|B, e) f_J(a) f_M(a) \\
 &= \alpha P(B) \sum_e P(e) \sum_a P(a|B, e) f_J(a) f_M(a) \\
 &= \alpha P(B) \sum_e P(e) \sum_a f_{AJM}(a, b, e) f_J(a) f_M(a) \\
 &= \alpha P(B) \sum_e P(e) f_{AJM}(b, e) \text{ (sum out } A) \\
 &= \alpha P(B) f_{EAJM}(b) \text{ (sum out } E) \\
 &= \alpha f_B(b) \times f_{EAJM}(b)
 \end{aligned}$$



Inference by Variable Elimination [2]: Factoring Operations

Pointwise product of factors f_1 and f_2 :

$$f_1(x_1, \dots, x_j, y_1, \dots, y_k) \times f_2(y_1, \dots, y_k, z_1, \dots, z_l) \\ = f(x_1, \dots, x_j, y_1, \dots, y_k, z_1, \dots, z_l)$$

E.g., $f_1(a, b) \times f_2(b, c) = f(a, b, c)$

Summing out a variable from a product of factors: move any constant factors outside the summation:

$$\sum_x f_1 \times \dots \times f_k = f_1 \times \dots \times f_i \sum_x f_{i+1} \times \dots \times f_k = f_1 \times \dots \times f_i \times f_{\bar{X}}$$

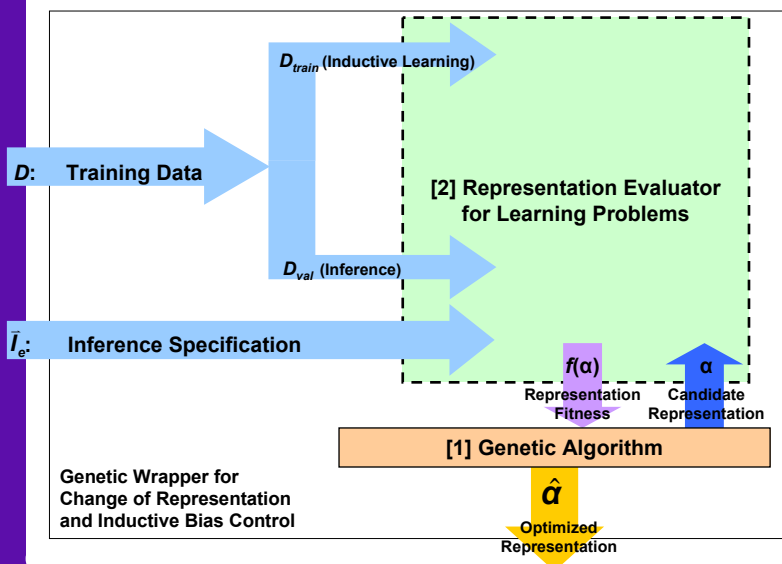
assuming f_1, \dots, f_i do not depend on X

Adapted from slides by S. Russell, UC Berkeley

<http://aima.cs.berkeley.edu/>



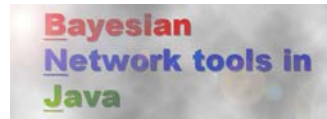
Genetic Algorithms for Parameter Tuning in Bayesian Network Structure Learning





Tools for Building Graphical Models

- **Commercial Tools: *Ergo, Netica, TETRAD, Hugin***
- **Bayes Net Toolbox (BNT) – Murphy (1997-present)**
 - * **Distribution page**
<http://http.cs.berkeley.edu/~murphyk/Bayes/bnt.html>
 - * **Development group**
<http://groups.yahoo.com/group/BayesNetToolbox>
- **Bayesian Network tools in Java (BNJ) – Hsu et al. (1999-present)**
 - * **Distribution page**
<http://bnj.sourceforge.net>
 - * **Development group**
<http://groups.yahoo.com/group/bndev>
 - * **Current (re)implementation projects for KSU KDD Lab**
 - **Continuous state:** Minka (2002) – Hsu, Guo, Li
 - **Formats:** XML BNIF (MSBN), Netica – Barber, Guo
 - **Space-efficient DBN inference** – Meyer
 - **Bounded cutset conditioning** – Chandak



References: Graphical Models and Inference Algorithms

- **Graphical Models**
 - * **Bayesian (Belief) Networks tutorial** – Murphy (2001)
<http://www.cs.berkeley.edu/~murphyk/Bayes/bayes.html>
 - * **Learning Bayesian Networks** – Heckerman (1996, 1999)
<http://research.microsoft.com/~heckerman>
- **Inference Algorithms**
 - * **Junction Tree (Join Tree, L-S, *Hugin*):** Lauritzen & Spiegelhalter (1988)
<http://citeseer.nj.nec.com/huang94inference.html>
 - * **(Bounded) Loop Cutset Conditioning:** Horvitz & Cooper (1989)
<http://citeseer.nj.nec.com/shachter94global.html>
 - * **Variable Elimination (Bucket Elimination, *ElimBel*):** Dechter (1986)
<http://citeseer.nj.nec.com/dechter96bucket.html>
 - * **Recommended Books**
 - Neapolitan (1990) – *out of print*; see [Pearl \(1988\)](#), Jensen (2001)
 - Castillo, Gutierrez, Hadi (1997)
 - Cowell, Dawid, Lauritzen, Spiegelhalter (1999)
 - * **Stochastic Approximation**
<http://citeseer.nj.nec.com/cheng00aisbn.html>



Terminology

- **Introduction to Reasoning under Uncertainty**
 - * Probability foundations
 - * Definitions: subjectivist, frequentist, logicist
 - * (3) Kolmogorov axioms
- **Bayes's Theorem**
 - * Prior probability of an event
 - * Joint probability of an event
 - * Conditional (posterior) probability of an event
- **Maximum A Posteriori (MAP) and Maximum Likelihood (ML) Hypotheses**
 - * MAP hypothesis: highest conditional probability given observations (data)
 - * ML: highest likelihood of generating the observed data
 - * ML estimation (MLE): estimating parameters to find ML hypothesis
- **Bayesian Inference: Computing Conditional Probabilities (CPs) in A Model**
- **Bayesian Learning: Searching Model (Hypothesis) Space using CPs**



Summary Points

- **Introduction to Probabilistic Reasoning**
 - * Framework: using probabilistic criteria to search H
 - * Probability foundations
 - ⇒ Definitions: subjectivist, objectivist; Bayesian, frequentist, logicist
 - ⇒ Kolmogorov axioms
- **Bayes's Theorem**
 - * Definition of conditional (posterior) probability
 - * Product rule
- **Maximum A Posteriori (MAP) and Maximum Likelihood (ML) Hypotheses**
 - * Bayes's Rule and MAP
 - * Uniform priors: allow use of MLE to generate MAP hypotheses
 - * Relation to version spaces, candidate elimination
- **Next Week: Chapter 14, Russell and Norvig**
 - * Later: Bayesian learning: MDL, BOC, Gibbs, Simple (Naïve) Bayes
 - * Categorizing text and documents, other applications

