



Lecture 29 of 42

Graphical Models of Probability 2 Discussion: Distributions, KA & Learning

Wednesday, 31 October 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:

Sections 14.3 – 14.5, Russell & Norvig 2nd edition



Lecture Outline

- Today and Friday's Reading: Sections 14.3 – 14.5, R&N 2e
- Next Week's Reading: Sections 14.6 – 14.8, Chapter 15
- Today: Graphical models
 - * Bayesian networks and causality
 - * Inference and learning
 - * BNJ interface (<http://bnj.sourceforge.net>)
 - * Causality





Graphical Models of Probability

- **Conditional Independence**

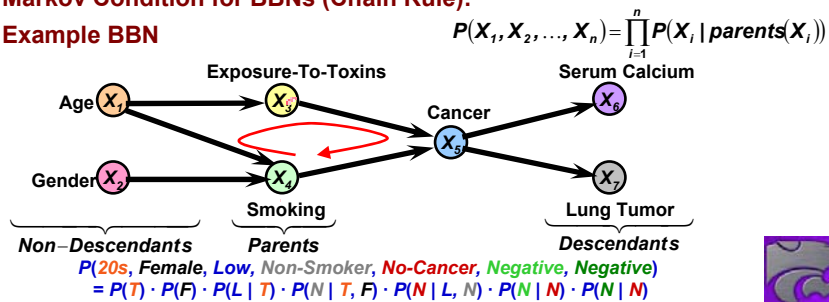
- * X is **conditionally independent (CI)** from Y given Z iff $P(X | Y, Z) = P(X | Z)$ for all values of $X, Y,$ and Z
- * Example: $P(\text{Thunder} | \text{Rain}, \text{Lightning}) = P(\text{Thunder} | \text{Lightning}) \Leftrightarrow T \perp R | L$

- **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**



Semantics of Bayesian Networks

“Global” semantics defines the full joint distribution as the product of the local conditional distributions:

$$P(X_1, \dots, X_n) = \prod_{i=1}^n P(X_i | \text{Parents}(X_i))$$

e.g., $P(J \wedge M \wedge A \wedge \neg B \wedge \neg E)$ is given by??

$$= P(\neg B)P(\neg E)P(A|\neg B \wedge \neg E)P(J|A)P(M|A)$$

“Local” semantics: each node is conditionally independent of its nondescendants given its parents

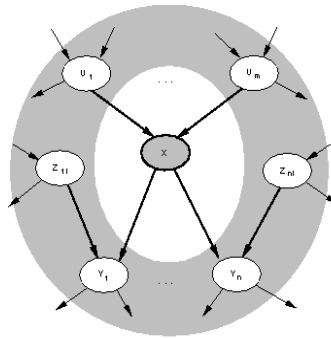
Theorem: Local semantics \Leftrightarrow global semantics

Adapted from slides by S. Russell, UC Berkeley



Markov Blanket

Each node is conditionally independent of all others given its Markov blanket: parents + children + children's parents



Adapted from slides by S. Russell, UC Berkeley



Constructing Bayesian Networks: The Chain Rule of Inference

Need a method such that a series of locally testable assertions of conditional independence guarantees the required global semantics

1. Choose an ordering of variables X_1, \dots, X_n
2. For $i = 1$ to n
add X_i to the network

select parents from X_1, \dots, X_{i-1} such that

$$\mathbf{P}(X_i | \text{Parents}(X_i)) = \mathbf{P}(X_i | X_1, \dots, X_{i-1})$$

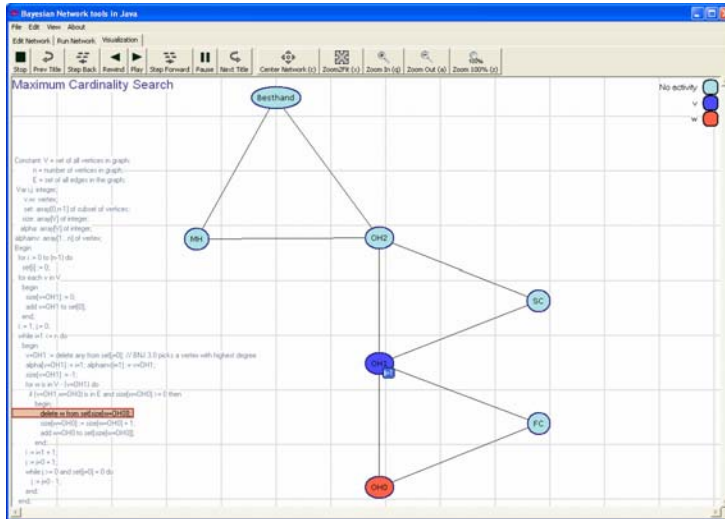
This choice of parents guarantees the global semantics:

$$\begin{aligned} \mathbf{P}(X_1, \dots, X_n) &= \prod_{i=1}^n \mathbf{P}(X_i | X_1, \dots, X_{i-1}) \text{ (chain rule)} \\ &= \prod_{i=1}^n \mathbf{P}(X_i | \text{Parents}(X_i)) \text{ by construction} \end{aligned}$$

Adapted from slides by S. Russell, UC Berkeley



BNJ Visualization [3] Network



Poker
Network

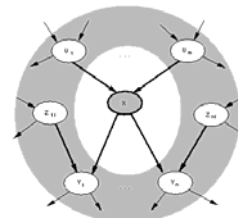
© 2004 KSU BNJ Development Team



Graphical Models Overview [2]: Markov Blankets and d -Separation Property

Motivation: The conditional independence status of nodes within a BBN might change as the availability of evidence E changes. *Direction-dependent separation (d -separation)* is a technique used to determine conditional independence of nodes as evidence changes.

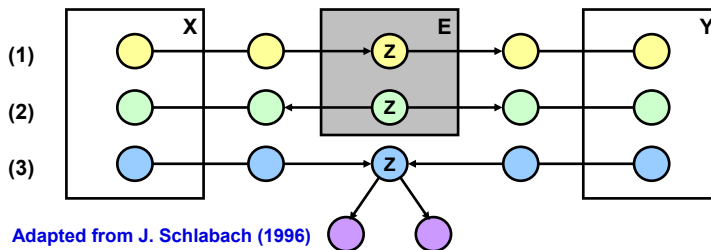
Each node is conditionally independent of all others given its Markov blanket: parents + children + children's parents



From S. Russell & P. Norvig (1995)

Definition: A set of evidence nodes E d -separates two sets of nodes X and Y if every undirected path from a node in X to a node in Y is *blocked* given E .

A path is *blocked* if one of three conditions holds:



Adapted from J. Schlabach (1996)



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 [1]: Temporal Probabilistic Reasoning

- **Goal: Estimate** $P(X_t | y_{1..r})$

- **Filtering: $r = t$**

- * Intuition: infer current state from observations
- * Applications: signal identification
- * Variation: Viterbi algorithm

Adapted from Murphy (2001), Guo (2002)

- **Prediction: $r < t$**

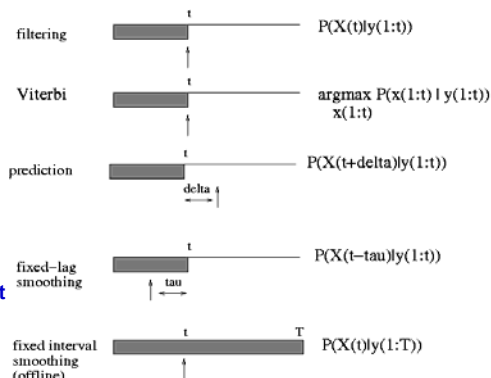
- * Intuition: infer future state
- * Applications: prognostics

- **Smoothing: $r > t$**

- * Intuition: infer past hidden state
- * Applications: signal enhancement

- **CF Tasks**

- * Plan recognition by smoothing
- * Prediction cf. *WebCANVAS* – Cadez et al. (2000)





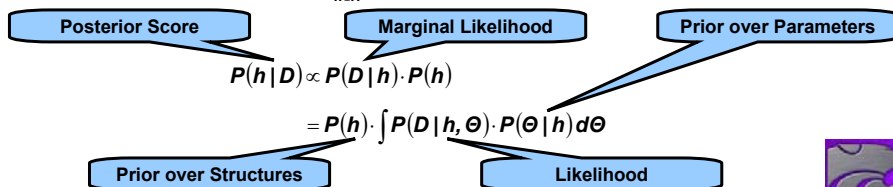
Other Topics in Graphical Models [2]: Learning Structure from Data

- General-Case BBN Structure Learning: *Use Inference to Compute Scores*
- Optimal Strategy: Bayesian Model Averaging

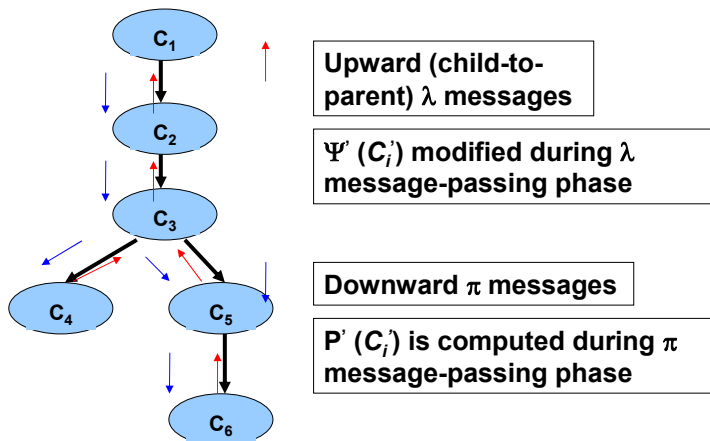
- * **Assumption:** models $h \in H$ are **mutually exclusive and exhaustive**
- * **Combine predictions of models in proportion to marginal likelihood**
 - Compute conditional probability of hypothesis h given observed data D
 - i.e., compute expectation over unknown h for unseen cases
 - Let $h \equiv$ structure, parameters $\Theta \equiv$ 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)

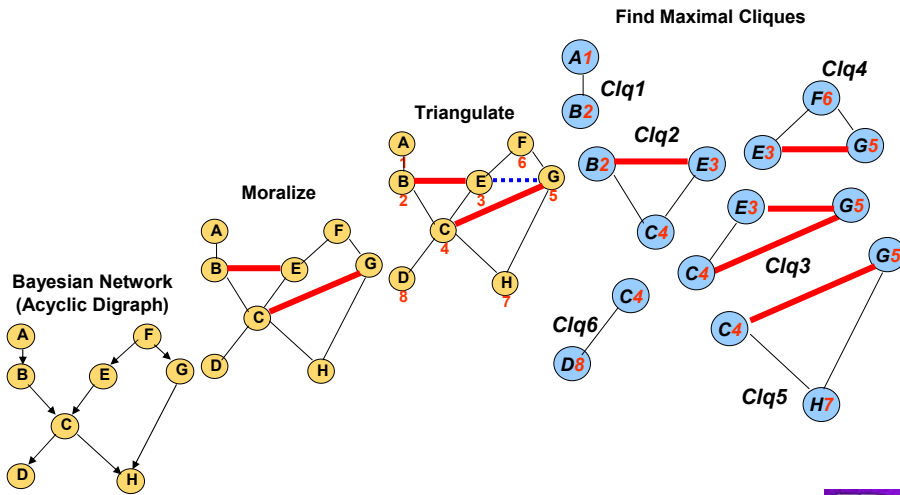


Multiply-connected case: exact, approximate inference are #P-complete
(counting problem is #P-complete iff decision problem is NP-complete)

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]: Junction Tree – Lauritzen & Spiegelhalter (1988)

Input: list of cliques of triangulated, moralized graph G_u

Output:

Tree of cliques

Separators nodes S_i ,

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

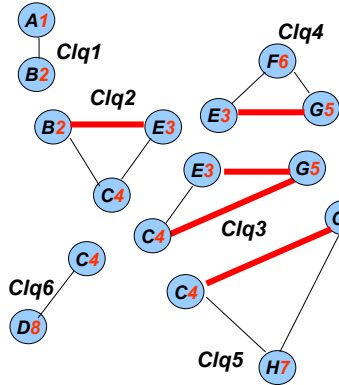
Algorithm:

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 $\Psi(\text{Clq}_i) = \prod_{f(v) \in \text{Clq}_i} P(v | c(v))$ {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

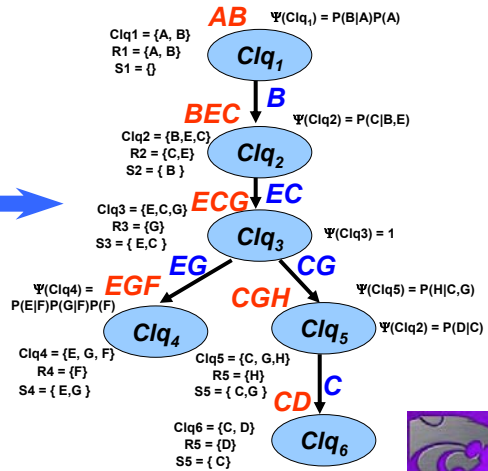


Adapted from Neapolitan (1990), Guo (2000)

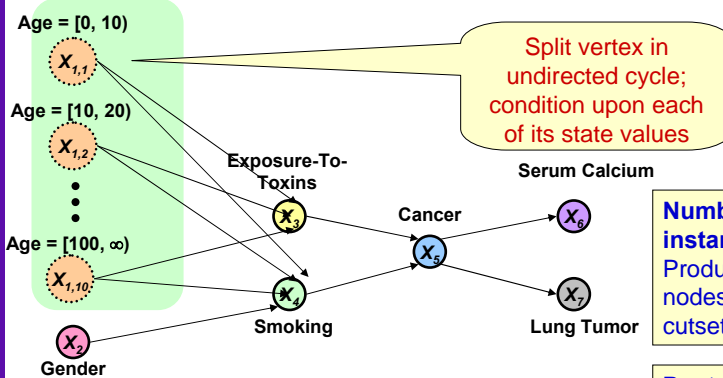
R_i : residual nodes

S_i : separator nodes

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



Inference by Loop Cutset Conditioning



Split vertex in undirected cycle; condition upon each of its state values

Number of network instantiations:
Product of arity of nodes in minimal loop cutset

Posterior: marginal conditioned upon cutset variable values

● Deciding Optimal Cutset: *NP-hard*

● Current Open Problems

* Bounded cutset conditioning: ordering heuristics

* Finding randomized algorithms for loop cutset optimization



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 [1]: 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>



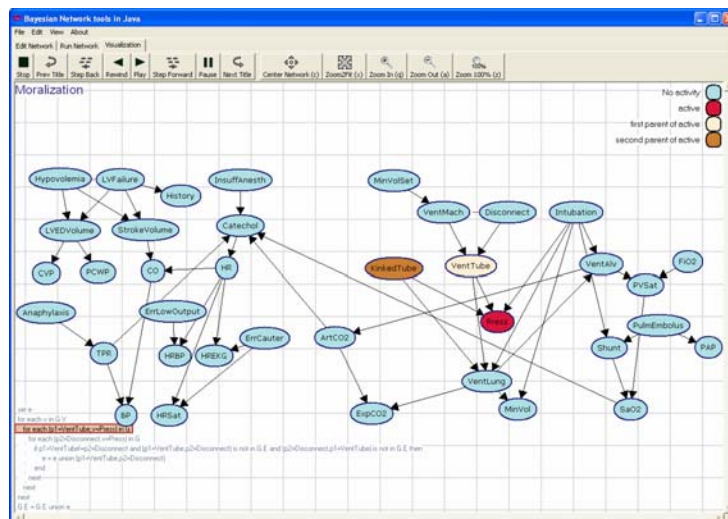


References [2]: Machine Learning, KDD, and Bioinformatics

- **Machine Learning, Data Mining, and Knowledge Discovery**
 - * **K-State KDD Lab: literature survey and resource catalog (1999-present)**
<http://www.kddresearch.org/Resources>
 - * **Bayesian Network tools in Java (BNJ): Hsu, Barber, King, Meyer, Thornton (2002-present)**
<http://bnj.sourceforge.net>
 - * **Machine Learning in Java (BNJ): Hsu, Louis, Plummer (2002)**
<http://mldev.sourceforge.net>
- **Bioinformatics**
 - * **European Bioinformatics Institute Tutorial: Brazma et al. (2001)**
http://www.ebi.ac.uk/microarray/biology_intro.htm
 - * **Hebrew University: Friedman, Pe'er, et al. (1999, 2000, 2002)**
<http://www.cs.huji.ac.il/labs/compbio/>
 - * **K-State BMI Group: literature survey and resource catalog (2002-2005)**
<http://www.kddresearch.org/Groups/Bioinformatics>



BNJ Visualization [2] Pseudo-Code Annotation (Code Page)



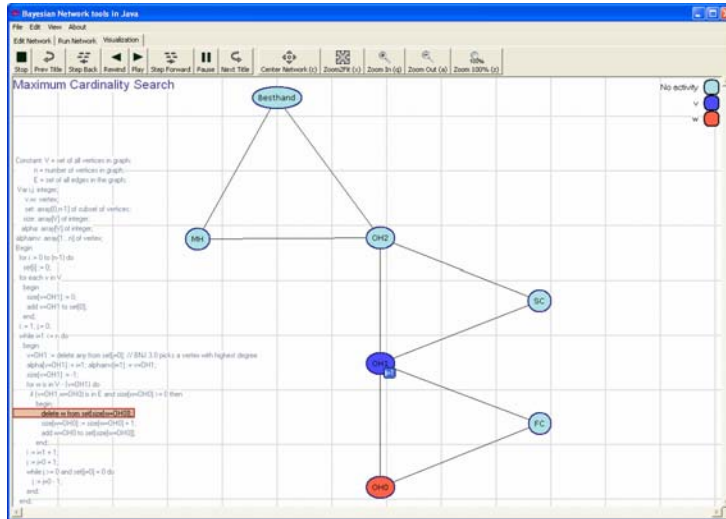
ALARM
Network

© 2004 KSU BNJ Development Team





BNJ Visualization [3] Network



Poker
Network

© 2004 KSU BNJ Development Team

CIS 530 / 730: Artificial Intelligence

Wednesday, 31 Oct 2007

Computing & Information Sciences
Kansas State University



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

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

Adapted from slides by S. Russell, UC Berkeley

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

CIS 530 / 730: Artificial Intelligence

Wednesday, 31 Oct 2007

Computing & Information Sciences
Kansas State University



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/>

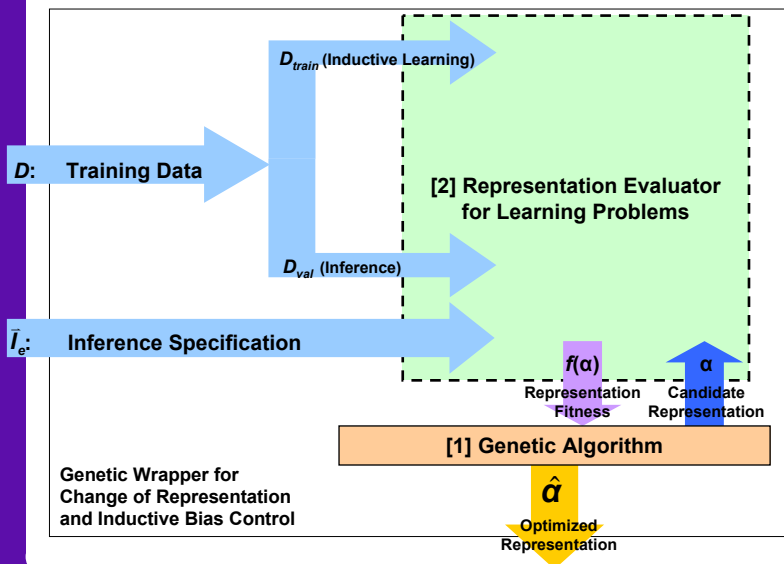
CIS 530 / 730: Artificial Intelligence

Wednesday, 31 Oct 2007

Computing & Information Sciences
Kansas State University



Genetic Algorithms for Parameter Tuning in Bayesian Network Structure Learning



CIS 530 / 730: Artificial Intelligence

Wednesday, 31 Oct 2007

Computing & Information Sciences
Kansas State University



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, logician
 - * (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