



## Lecture 30 of 42

### Inference and Software Tools 1 Discussion: Projects, BNJ

Friday, 02 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 14, Russell & Norvig 2<sup>nd</sup> edition



## Lecture Outline

- Today'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





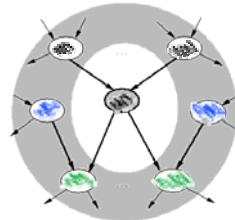


## Graphical Models Overview [2]: Markov Blankets and *d*-Separation Property

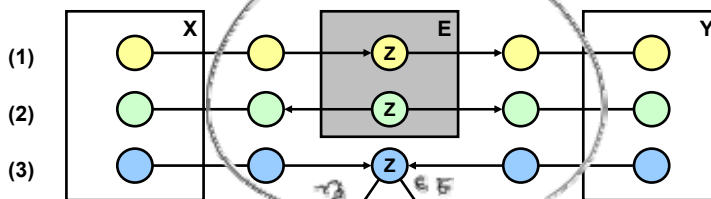
**Motivation:** The conditional independence status of nodes within a BBN might change as the availability of evidence  $E$  changes. *Direction-dependent separation (d-separation)* is a technique used to determine conditional independence of nodes as evidence changes.

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

**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:



From S. Russell & P. Norvig (1995)

Adapted from J. Schlabach (1996)



## Graphical Models Overview [3]: Inference Problem

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

Let the hidden variables be  $H = X - Y - E$

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

$$P(Y|E=e) = \frac{1}{Z} P(Y, E=e) = \frac{1}{Z} \sum_{\mathbf{H}} P(Y, E=e, \mathbf{H}=\mathbf{h})$$

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

Obvious problems:

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

Multinomially connected case: exact 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**  $P(X_t^i | y_{1:r})$

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

- \* Intuition: infer current state from observations

- \* Applications: signal identification

- \* Variation: Viterbi algorithm

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

- \* Intuition: infer future state

- \* Applications: prognostics

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

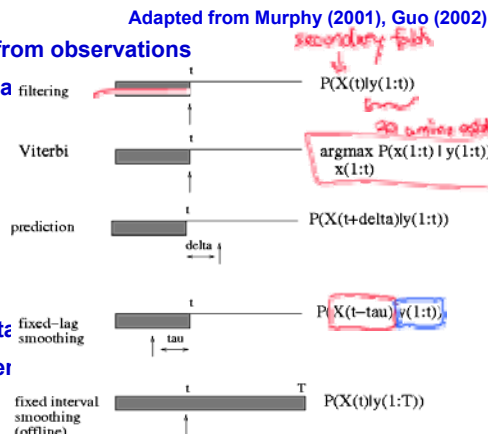
- \* Intuition: infer past hidden state

- \* Applications: signal enhancer

- **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

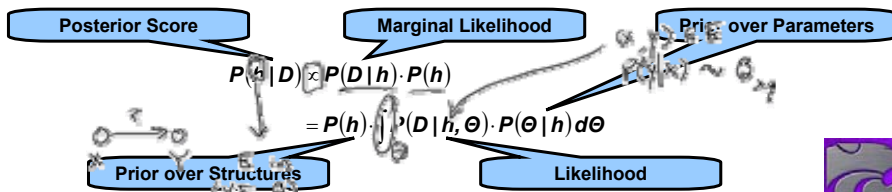
- Compute conditional probability of hypothesis  $h$  given observed data  $D$

- i.e., compute expectation over unknown  $h$  for unseen cases

- Let  $h \equiv$  structure, parameters  $\Theta \equiv$  CPTs

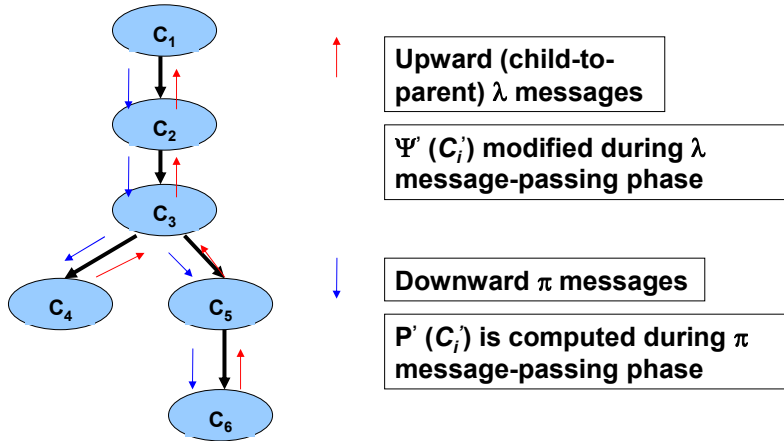
$$P(\bar{x}^{(m+1)} | D) = P(x_1, x_2, \dots, x_n | \bar{x}^{(1)}, \bar{x}^{(2)}, \dots, \bar{x}^{(m)})$$

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





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

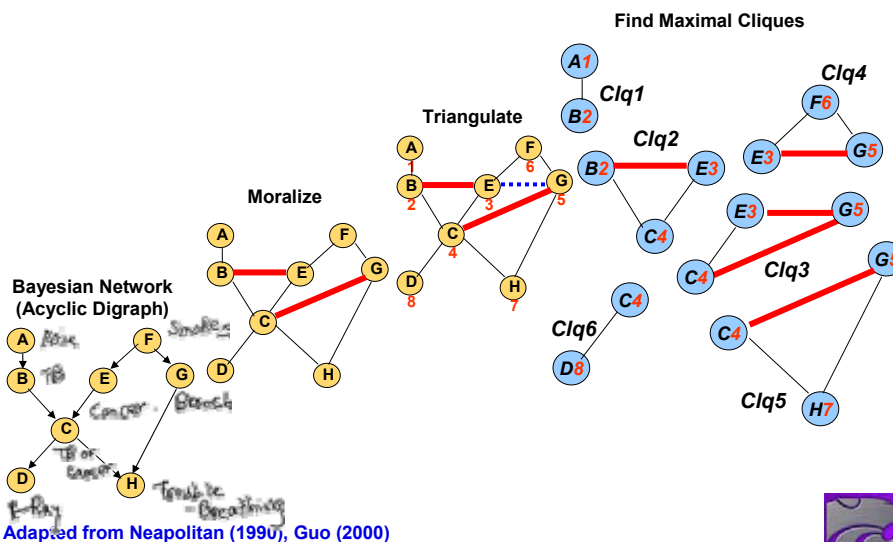


Multiply-connected case: exact, approximate inference are  $\#P$ -complete (counting problem is  $\#P$ -complete iff decision problem is  $NP$ -complete)

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

Separator nodes  $S_i$ ,

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

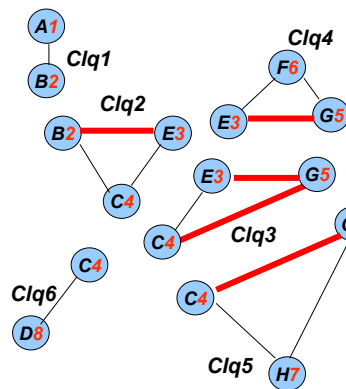
**Algorithm:**

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  $\text{Clq}_j$  is a parent of  $\text{Clq}_i$
4. Assign each node  $v$  to a unique clique  $\text{Clq}_i$  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  $\text{Clq}_i$ }
6. Store  $\text{Clq}_i$ ,  $R_i$ ,  $S_i$ , and  $\Psi(\text{Clq}_i)$  at each vertex in the tree of cliques

Adapted from Neapolitan (1990), Guo (2000)



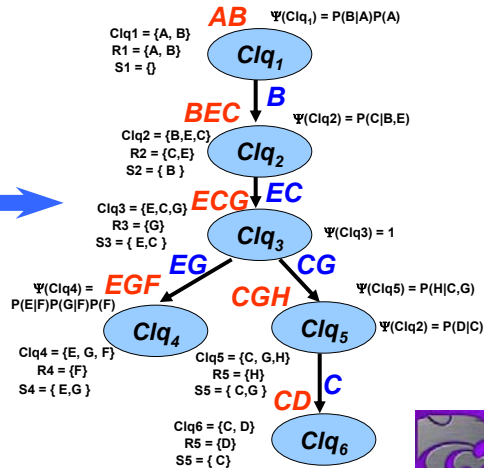
## Inference by Clustering [3]: Clique-Tree Operations



$R_i$ : 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

Age = [0, 10)  $X_{1,1}$

Age = [10, 20)  $X_{1,2}$

...

Age = [100,  $\infty$ )  $X_{1,10}$

Gender  $X_2$

Exposure-To-Toxins  $X_3$

Smoking  $X_4$

Cancer  $X_5$

Serum Calcium  $X_6$

Lung Tumor  $X_7$

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

CIS 530 / 730: Artificial Intelligence Friday, 02 Nov 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

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

Adapted from slides by S. Russell, UC Berkeley

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



## 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:

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

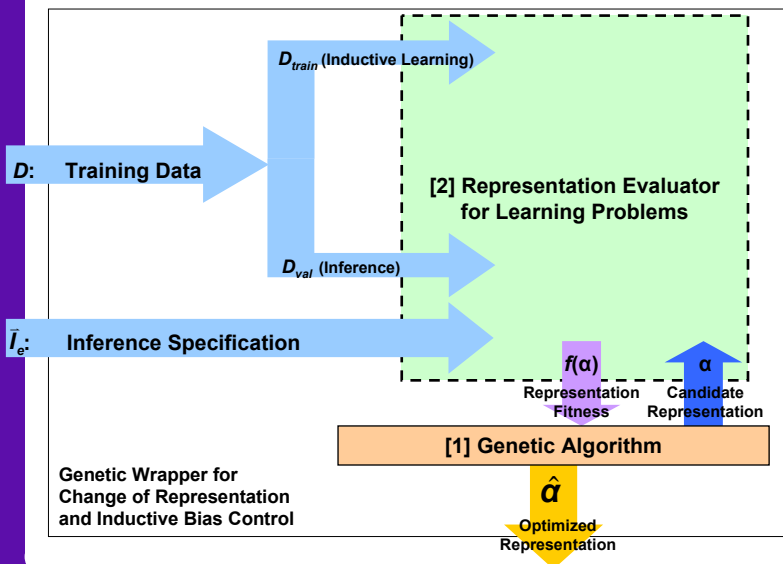
assuming  $f_1, \dots, f_i$  do not depend on  $X$

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

Bayesian  
Network tools in  
Java



## 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>





## Using Graphical Models

- **A Graphical View of Simple (Naïve) Bayes**

- \*  $x_i \in \{0, 1\}$  for each  $i \in \{1, 2, \dots, n\}$ ;  $y \in \{0, 1\}$

- \* Given:  $P(x_i | y)$  for each  $i \in \{1, 2, \dots, n\}$ ;  $P(y)$

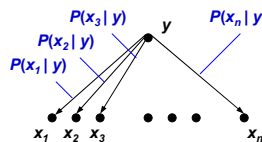
- \* Assume conditional independence

- $\forall i \in \{1, 2, \dots, n\} \Rightarrow P(x_i | x_{\setminus i}, y) \equiv P(x_i | x_1, x_2, \dots, x_{i-1}, x_{i+1}, x_{i+2}, \dots, x_n, y) = P(x_i | y)$

- NB: this assumption entails the Naïve Bayes assumption

- Why?  $P(x_1, x_2, \dots, x_n | y) = \prod_i P(x_i | x_{\setminus i}, y) = \prod_i P(x_i | y)$

- \* Can compute  $P(y | x)$  given this info



- \* Can also compute the joint pdf over all  $n + 1$  variables

$$P(\vec{x}, y) = P(y)P(\vec{x} | y) = P(y) \prod_{i=1}^n P(x_i | x_{\setminus i}, y) = P(y) \prod_{i=1}^n P(x_i | y)$$

- **Inference Problem for a (Simple) Bayesian Network**

- \* Use the above model to compute the probability of any *conditional event*

- \* Exercise:  $P(x_1, x_2, y | x_3, x_4)$



## In-Class Exercise: Probabilistic Inference

- **Inference Problem for a (Simple) Bayesian Network**

- \* Model: Naïve Bayes

- \* Objective: compute the probability of any *conditional event*

- **Exercise**

- \* Given

- $P(x_i | y)$ ,  $i \in \{1, 2, 3, 4\}$

- $P(y)$

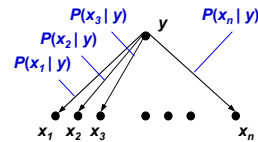
- \* Want:  $P(x_1, x_2, y | x_3, x_4)$

$$\begin{aligned} P(x_1, x_2, y | x_3, x_4) &= \frac{P(x_3, x_4 | x_1, x_2, y)P(x_1, x_2, y)}{P(x_3, x_4)} \\ &= \frac{P(x_1, x_2, x_3, x_4, y)}{P(x_3, x_4)} \\ &= \frac{P(y) \prod_{i=1}^4 P(x_i | y)}{\sum_y P(x_3 | y)P(x_4 | y)} \end{aligned}$$



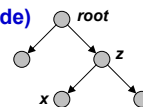
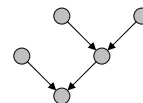
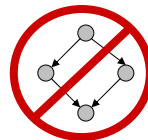
## Unsupervised Learning and Conditional Independence

- **Given:  $(n + 1)$ -Tuples**  $(x_1, x_2, \dots, x_n, x_{n+1})$ 
  - \* No notion of instance variable or label
  - \* After seeing some examples, want to know something about the domain
    - Correlations among variables
    - Probability of certain events
    - Other properties
- **Want to Learn: Most Likely Model that Generates Observed Data**
  - \* In general, a very hard problem
  - \* *Under certain assumptions*, have shown that we can do it
- **Assumption: Causal Markovity**
  - \* Conditional independence among “effects”, given “cause”
  - \* *When is the assumption appropriate?*
  - \* *Can it be relaxed?*
- **Structure Learning**
  - \* Can we learn more general probability distributions?
  - \* Examples: automatic speech recognition (ASR), natural language, etc.



## Tree Dependent Distributions

- **Polytrees**
  - \* singly-connected Bayesian networks
  - \* **Definition:** a Bayesian network with no undirected loops
  - \* **Idea:** restrict distributions (CPTs) to single nodes
  - \* **Theorem:** inference in singly-connected BBN requires linear time
    - Linear in network size, including CPT sizes
    - Much better than for unrestricted (multiply-connected) BBNs
- **Tree Dependent Distributions**
  - \* Further restriction of polytrees: every node has at one parent
  - \* Now only need to keep 1 prior,  $P(\text{root})$ , and  $n - 1$  CPTs (1 per node)
  - \* All CPTs are 2-dimensional:  $P(\text{child} | \text{parent})$
- **Independence Assumptions**
  - \* As for general BBN:  $x$  is independent of non-descendants given (single) parent  $z$
  - \* *Very strong assumption* (applies in some domains but not most)





## Terminology

- **Introduction to Reasoning under Uncertainty**
  - \* Probability foundations
  - \* Definitions: subjectivist, frequentist, logistic
  - \* (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, logistic
    - ⇒ 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

