Lecture 29 of 42

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

Wednesday, 01 November 2006

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-2006/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

Bayes's Theorem: Review

- **Theorem**

$$P(h \mid D) = \frac{P(D \mid h)P(h)}{P(D)} = \frac{P(h \wedge D)}{P(D)}$$

- ***P*(*h*) ≡ Prior Probability of Assertion (Hypothesis) *h***
  - **Measures <u>initial beliefs</u> (BK) <u>before</u> any information is obtained (hence <u>prior</u>)**
- ***P*(*D*) ≡ Prior Probability of Data (Observations) *D***
  - **Measures probability of obtaining sample *D* (i.e., expresses D)**
- ***P*(*h* | *D*) ≡ Probability of *h* Given *D***
  - **| denotes <u>conditioning</u> - hence P(h | D) is a <u>conditional</u> (*aka* <u>posterior</u>) probability**
- ***P*(*D* | *h*) ≡ Probability of *D* Given *h***
  - **Measures probability of observing *D* given that *h* is correct ("<u>generative</u>" model)**
- ***P*(*h* ∧ *D*) ≡ <u>Joint</u> Probability of *h* and *D***
  - **Measures probability of observing *D* and of *h* being correct**

Choosing Hypotheses

- Bayes's Theorem

$$P(h \mid D) = \frac{P(D \mid h)P(h)}{P(D)} = \frac{P(h \wedge D)}{P(D)}$$

- MAP Hypothesis
  - Generally want most probable hypothesis given the training data
  - Define: $\arg\max_{x \in \Omega}[f(x)]$ ≡ the value of *x* in the sample space Ω with the highest *f*(*x*)
  - <u>M</u>aximum <u>*a*</u> <u>*p*</u>*osteriori* hypothesis, h_{MAP}

$$\begin{aligned} h_{MAP} &= \arg\max_{h \in H} P(h \mid D) \\ &= \arg\max_{h \in H} \frac{P(D \mid h)P(h)}{P(D)} \\ &= \arg\max_{h \in H} P(D \mid h)P(h) \end{aligned}$$

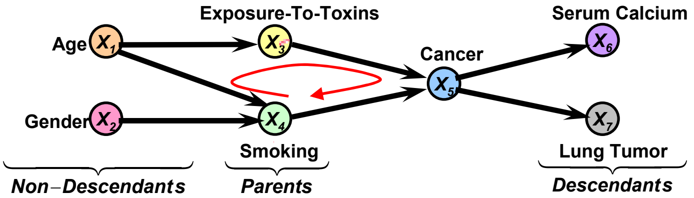
- ML Hypothesis
  - Assume that $p(h_i) = p(h_j)$ for all pairs *i*, *j* (<u>uniform priors</u>, i.e., P_H ~ Uniform)
  - Can further simplify and choose the <u>m</u>aximum <u>l</u>ikelihood hypothesis, h_{ML}

$$h_{ML} = \arg\max_{h_i \in H} P(D \mid h_i)$$

Graphical Models of Probability

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

$$P(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} P(X_i \mid parents(X_i))$$



P(*20s*, *Female*, *Low*, *Non-Smoker*, *No-Cancer*, *Negative*, *Negative*)
$= P(T) \cdot P(F) \cdot P(L \mid T) \cdot P(N \mid T, F) \cdot P(N \mid L, N) \cdot P(N \mid N) \cdot P(N \mid N)$

Semantics of Bayesian Networks

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

$$\mathbf{P}(X_1, \ldots, X_n) = \Pi_{i=1}^{n} \mathbf{P}(X_i \mid Parents(X_i))$$

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

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

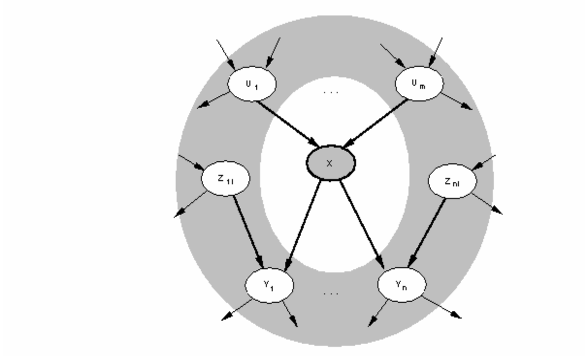
"Local" semantics: each node is conditionally independent of its nondescendants given its parents

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 $X_1, \ldots, X_n$
2. For $i = 1$ to n
   add X_i to the network
   select parents from $X_1, \ldots, X_{i-1}$ such that
   $\mathbf{P}(X_i|Parents(X_i)) = \mathbf{P}(X_i|X_1, \ldots, X_{i-1})$

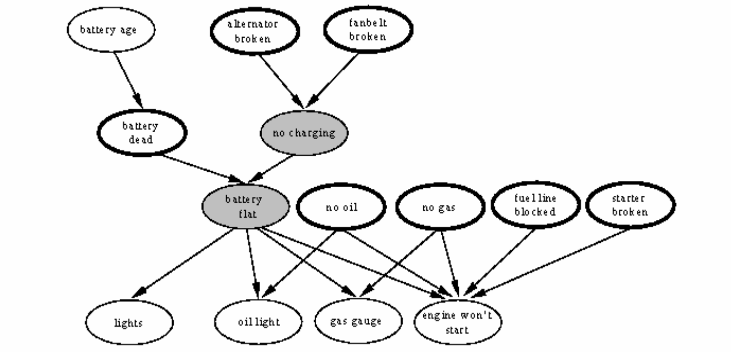
This choice of parents guarantees the global semantics:

$$\mathbf{P}(X_1, \ldots, X_n) = \Pi_{i=1}^{n} \mathbf{P}(X_i|X_1, \ldots, X_{i-1}) \text{ (chain rule)}$$
$$= \Pi_{i=1}^{n} \mathbf{P}(X_i|Parents(X_i)) \text{ by construction}$$

Adapted from slides by S. Russell, UC Berkeley

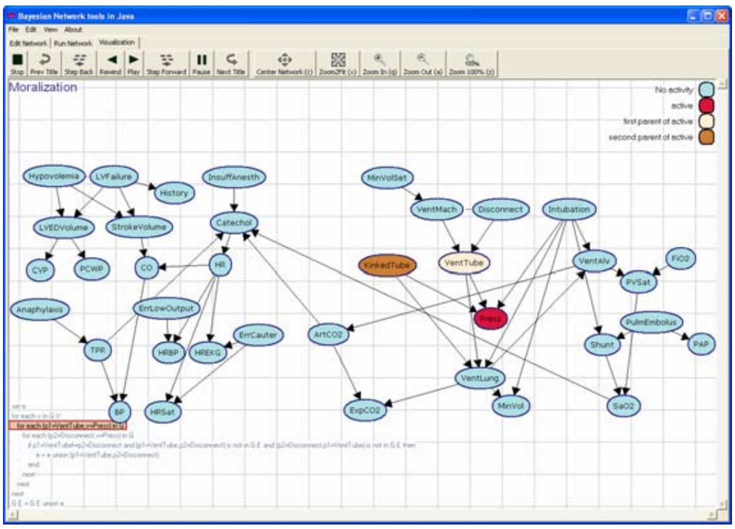
Example:
Evidential Reasoning for Car Diagnosis

Initial evidence: engine won't start
Testable variables (thin ovals), diagnosis variables (thick ovals)
Hidden variables (shaded) ensure sparse structure, reduce parameters



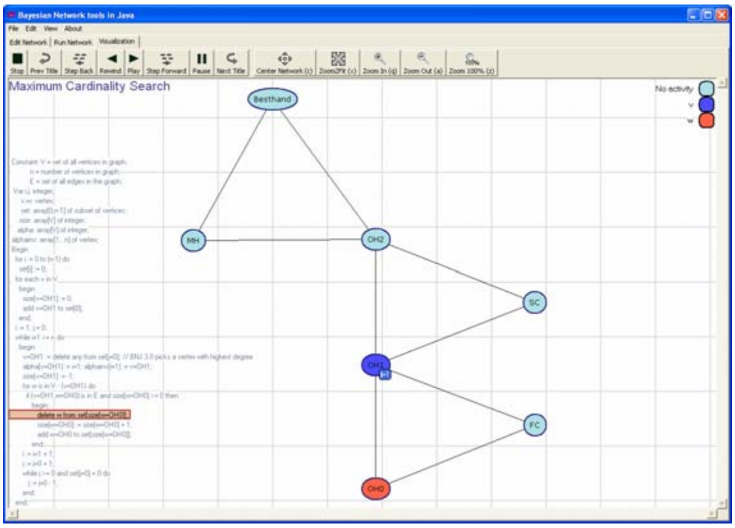
Adapted from slides by S. Russell, UC Berkeley

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



ALARM Network

BNJ Visualization [3]
Network



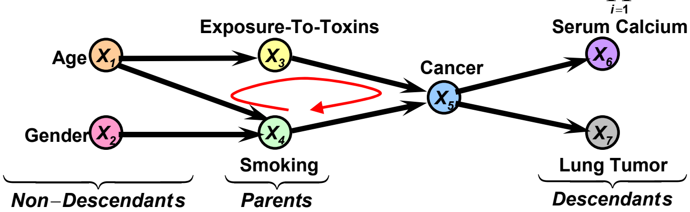
Poker Network

© 2004 KSU BNJ Development Team

Graphical Models Overview [1]:
Bayesian Networks

- Conditional Independence
  - X is conditionally independent (CI) from Y given Z (sometimes written $X \perp Y \mid Z$) iff $P(X \mid Y, Z) = P(X \mid Z)$ for all values of *X, Y*, and *Z*
  - Example: $P(Thunder \mid Rain, Lightning) = P(Thunder \mid Lightning) \Leftrightarrow T \perp R \mid L$
- 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

$$P(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} P(X_i \mid parents(X_i))$$



$$P(20s, Female, Low, Non\text{-}Smoker, No\text{-}Cancer, Negative, Negative)$$
$$= P(T) \cdot P(F) \cdot P(L \mid T) \cdot P(N \mid T, F) \cdot P(N \mid L, N) \cdot P(N \mid N) \cdot P(N \mid N)$$

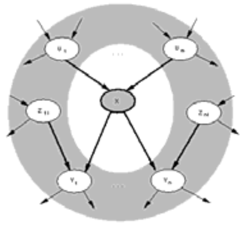
Graphical Models Overview [2]: Markov Blankets and *d*-Separation Property

__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.

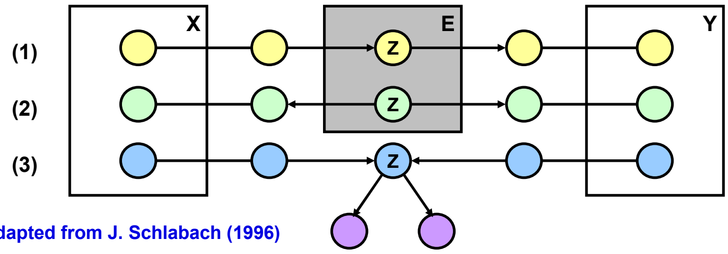
__Definition__: A set of evidence nodes *E d*-separates two sets of nodes *X* and *Y* if every undirected path from a node in *X* to a node in *Y* is *blocked* given *E*.

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

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



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 $\mathbf{Y}$
given specific values $\mathbf{e}$ 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:

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

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 $O(d^n)$ 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???

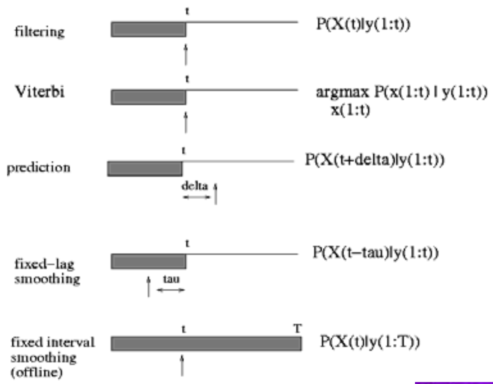
Multiply-connected case: exact, approximate inference are #P-complete

Adapted from slides by S. Russell, UC Berkeley http://aima.cs.berkeley.edu/

Other Topics in Graphical Models [1]: Temporal Probabilistic Reasoning

- Goal: Estimate $P(X_t^i \mid y_{1 \ldots r})$
- 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 enhancement
- CF Tasks
  - Plan recognition by smoothing
  - Prediction cf. *WebCANVAS* – Cadez *et al.* (2000)

Adapted from Murphy (2001), Guo (2002)



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

$$P\left(\vec{x}^{(m+1)} \mid D\right) = P\left(x_1, x_2, \ldots, x_n \mid \vec{x}^{(1)}, \vec{x}^{(2)}, \ldots, \vec{x}^{(m)}\right)$$

$$= \sum_{h \in H} P\left(\vec{x}^{(m+1)} \mid D, h\right) \cdot P(h \mid D)$$

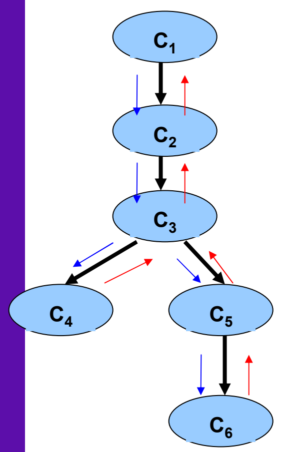
Posterior Score | Marginal Likelihood | Prior over Parameters

$$P(h \mid D) \propto P(D \mid h) \cdot P(h)$$

$$= P(h) \cdot \int P(D \mid h, \Theta) \cdot P(\Theta \mid h)\, d\Theta$$

Prior over Structures | Likelihood

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



Upward (child-to-parent) λ messages

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

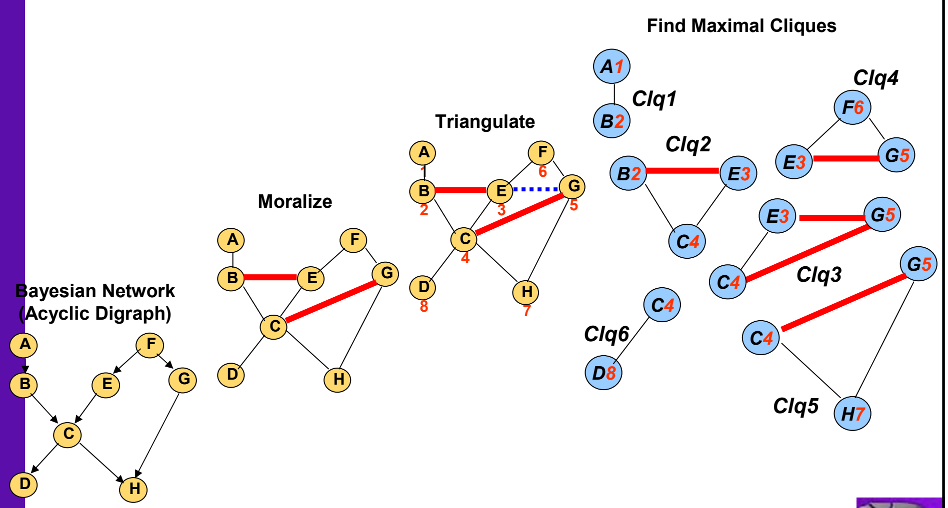
Downward π messages

$P'(C_i')$ 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]: 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 $\Psi(Clq_i)$** for all cliques

Algorithm:

1. **S_i** = $Clq_i \cap (Clq_1 \cup Clq_2 \cup \ldots \cup Clq_{i-1})$
2. **R_i** = 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 $v \cup c(v) \subseteq Clq_i$
5. Compute $\Psi(Clq_i) = \Pi_{f(v)\ Clqi} = P(v \mid c(v))$ {1 if no v is assigned to Clq_i}
6. Store Clq_i , **R_i** , **S_i**, and **$\Psi(Clq_i)$** at each vertex in the tree of cliques

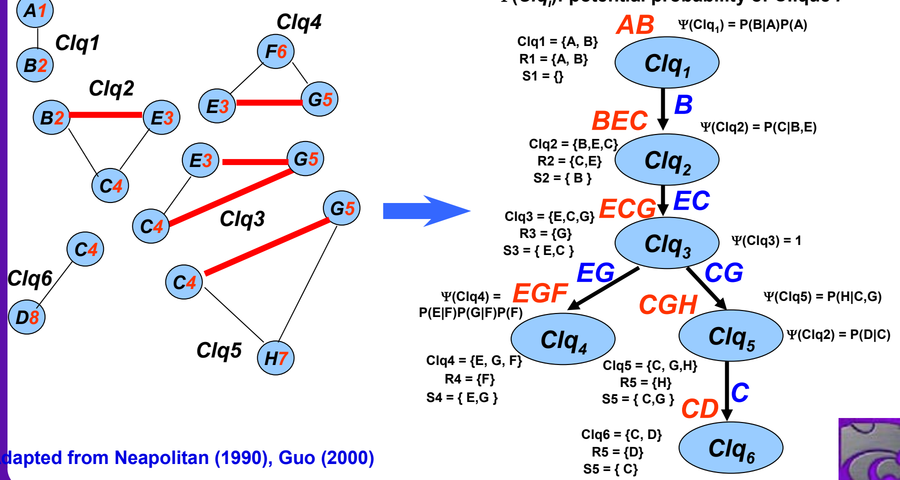
Adapted from Neapolitan (1990), Guo (2000)

Inference by Clustering [3]: Clique-Tree Operations

***R_i*: residual nodes**

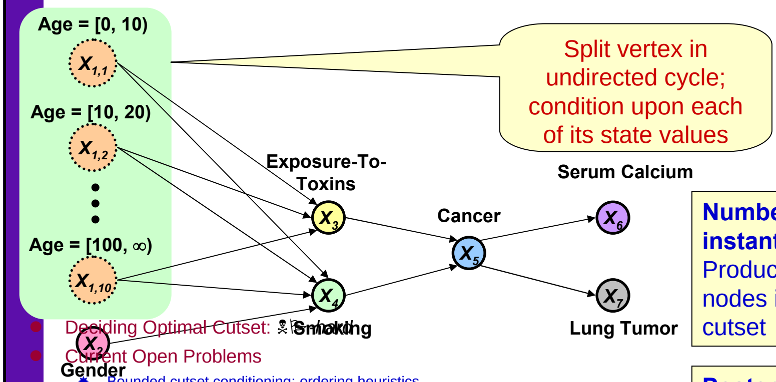
***S_i*: separator nodes**

$\Psi(Clq_i)$: potential probability of Clique *i*



Adapted from Neapolitan (1990), Guo (2000)

Inference by Loop Cutset Conditioning



- Deciding Optimal Cutset: NP-hard
- Current Open Problems
  - Bounded cutset conditioning: ordering heuristics
  - Finding randomized algorithms for loop cutset optimization

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

Posterior: marginal conditioned upon cutset variable values

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

$$
\begin{aligned}
\mathbf{P}(B|J=true, M=true)
&= \alpha \underbrace{\mathbf{P}(B)}_{B} \Sigma_e \underbrace{P(e)}_{E} \Sigma_a \underbrace{\mathbf{P}(a|B,e)}_{A} \underbrace{P(J=true|a)}_{J} \underbrace{P(M=true|a)}_{M} \\
&= \alpha \mathbf{P}(B) \Sigma_e P(e) \Sigma_a \mathbf{P}(a|B,e) P(J=true|a) f_M(a) \\
&= \alpha \mathbf{P}(B) \Sigma_e P(e) \Sigma_a \mathbf{P}(a|B,e) f_J(a) f_M(a) \\
&= \alpha \mathbf{P}(B) \Sigma_e P(e) \Sigma_a f_A(a,b,e) f_J(a) f_M(a) \\
&= \alpha \mathbf{P}(B) \Sigma_e P(e) f_{\bar{A}JM}(b,e) \text{ (sum out } A) \\
&= \alpha \mathbf{P}(B) f_{\bar{E}\bar{A}JM}(b) \text{ (sum out } E) \\
&= \alpha f_B(b) \times f_{\bar{E}\bar{A}JM}(b)
\end{aligned}
$$

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:

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

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

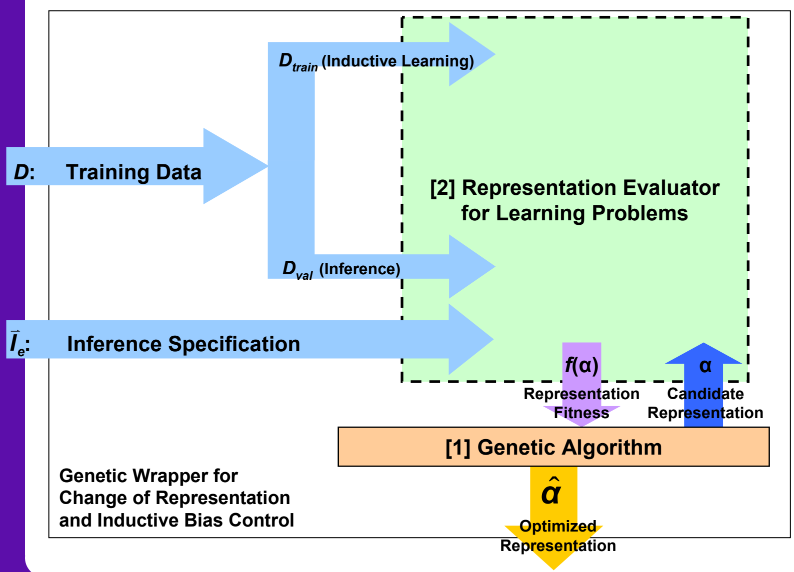
Summing out a variable from a product of factors: move any constant factors outside the summation:

$$\Sigma_x f_1 \times \cdots \times f_k = f_1 \times \cdots \times f_i \, \Sigma_x \, f_{i+1} \times \cdots \times f_k = f_1 \times \cdots \times f_i \times f_{\bar{X}}$$

assuming $f_1,\ldots,f_i$ 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