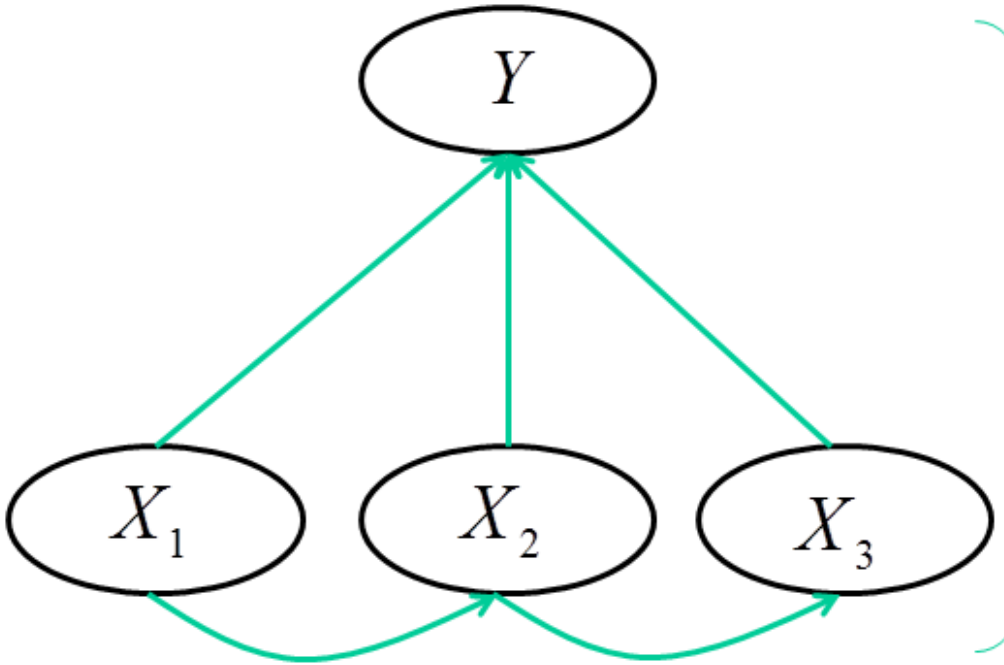
Bayesian Networks: Construction, Inference, Learning and Causal Interpretation

Volker Tresp
Summer 2019

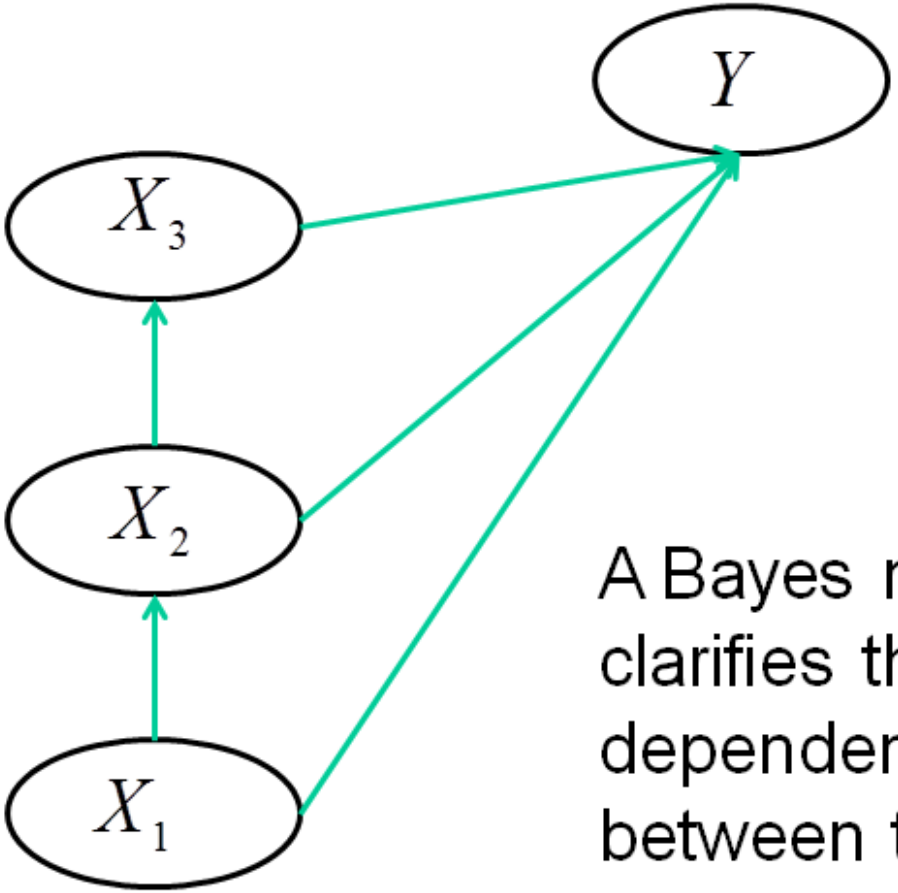
Introduction

- So far we were mostly concerned with supervised learning: we predicted one or several target variables based on information on one or several input variables
- If applicable, supervised learning is often the best solution and powerful models such as Neural Networks and kernel approaches are available
- But there are cases where supervised learning is not applicable: (1) there is not only one target variable of interest but many (2) for each data point different variables might be missing
- Typical example: medical domain with many kinds of diseases, symptoms, and context information: for a given patient little is known and one is interested in the prediction of many possible diseases
- Bayesian networks can deal with these challenges, which is the reason for their popularity in probabilistic reasoning and machine learning



Focus of supervised learning

- Dependencies between input variables are not modeled explicitly
- (Implicitly they appear in X^TX)

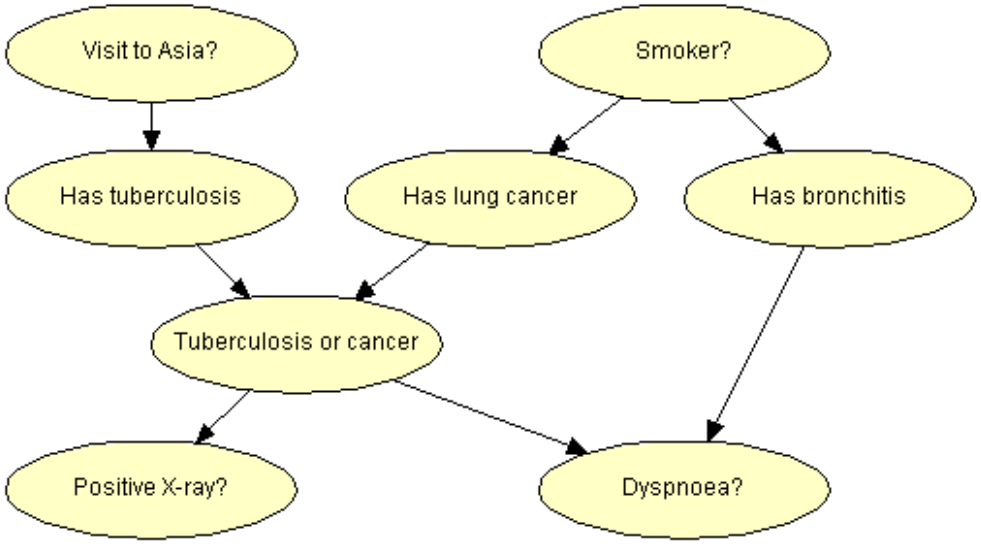


A Bayes net also clarifies the dependencies between the inputs

Bayes Nets

- Deterministic rule-based systems were the dominating approach during the first phase of AI. A major problem was that deterministic rules cannot deal with uncertainty and ambiguity
- Probability was rejected since a naive representation would require 2^M parameters; how should all these numbers be specified?
- In Bayes nets one constructs a high-dimensional probability distribution based on a set of local probabilistic rules
- Bayes nets are closely related to a causal world model
- Bayes Nets started as a small community and then developed to one of the main approaches in AI
- They also have a great influence on machine learning

Chest Clinic



Definition of a Bayes Net

- The random variables in a domain are displayed as nodes (vertices)
- Directed links (arcs, edges) represent direct (causal) dependencies between parent node and child node
- Quantification of the dependency:
  - For nodes without parents one specifies a priori probabilities

$$P(A = i) \quad \forall i$$

  - For nodes with parents, one specifies conditional probabilities. E.g., for two parents

$$P(A = i | B = j, C = k) \quad \forall i, j, k$$

Joint Probability Distribution

- A Bayes net specifies a probability distribution in the form

$$P(X_1, \ldots X_M) = \prod_{i=1}^{M} P(X_i | \text{par}(X_i))$$

where $\text{par}(X_i)$ is the set of parent nodes. This set can be empty.

Factorization of Probability Distributions

- Let's start with the factorization of a probability distribution (see review on Probability Theory)

$$P(X_1, \ldots X_M) = \prod_{i=1}^{M} P(X_i | X_1, \ldots, X_{i-1})$$

- This decomposition can be done with an arbitrary ordering of variables; each variable is conditioned on all predecessor variables

- The dependencies can be simplified if a variable does not depend on all of its predecessors

$$P(X_i | X_1, \ldots, X_{i-1}) = P(X_i | \text{par}(X_i))$$

with

$$\text{par}(X_i) \subseteq X_1, \ldots, X_{i-1}$$

Causal Ordering

- When the ordering of the variables corresponds to a causal ordering, we obtain a causal probabilistic network
- A decomposition obeying the causal ordering typically yields a representation with the smallest number of parent variables
- Deterministic and probabilistic causality: it might be that the underlying true model is deterministic causal. The probabilistic model leaves out many influential factors. The assumption is that the un-modeled factors should only significantly influence individual nodes (and thus appear as noise), but NOT pairs or larger sets of variables (which would induce dependencies)!

Design of a Bayes Net

- The expert needs to be clear about the important variables in the domain
- The expert must indicate direct causal dependencies by specifying the directed link in the net
- The expert needs to quantify the causal dependencies: define the conditional probability tables

Inference

- The most important important operation is inference: given that the state a set of random variables is known, what is the probability distribution of one or several of the remaining variables
- Let $\mathcal{X}$ be the set of random variables. Let $\mathcal{X}^m \subseteq \mathcal{X}$ be the set of known (measured) variables and let $X^q \in \mathcal{X} \setminus \mathcal{X}^m$ be the variable of interest and let $\mathcal{X}^r = \mathcal{X} \setminus (\mathcal{X}^m \cup X^q)$ be the set of remaining variables

Inference: Marginalization and Conditioning

- In the simplest inference approach one proceeds as follows :
  - We calculate the probability distribution of the known variables and the query variable via marginalization

$$P(X^q, \mathcal{X}^m) = \sum_{\mathcal{X}^r} P(X_1, \ldots X_M)$$

  - The normalization is calculated as

$$P(\mathcal{X}^m) = \sum_{\mathcal{X}^q} P(X^q, \mathcal{X}^m)$$

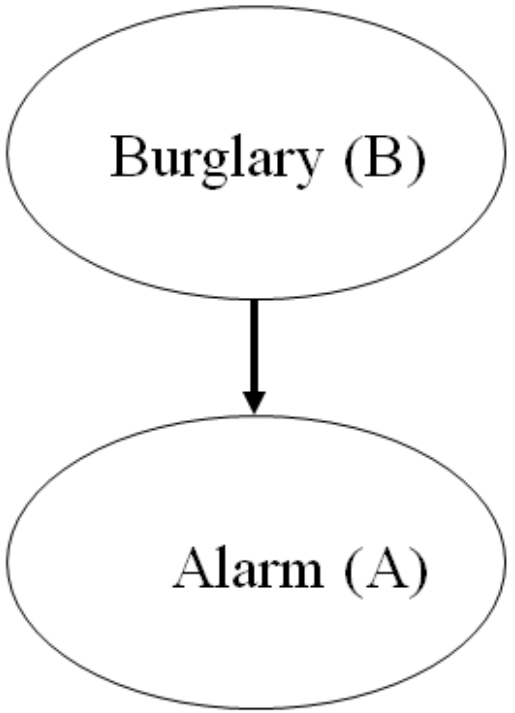
  - Calculation of the conditional probability distributions

$$P(X^q|\mathcal{X}^m) = \frac{P(X^q, \mathcal{X}^m)}{P(\mathcal{X}^m)}$$

Inference in a simple Bayes Net

$$P(B=1)=0.01$$
$$P(B=0)=0.99$$

$$P(A\mid B)=0.98$$
$$P(A\mid \neg B)=0.01$$



Marginalization (here not necessary):
$$P(A,B)=P(A\mid B)P(B)=0.98\times 0.01=0.0098$$

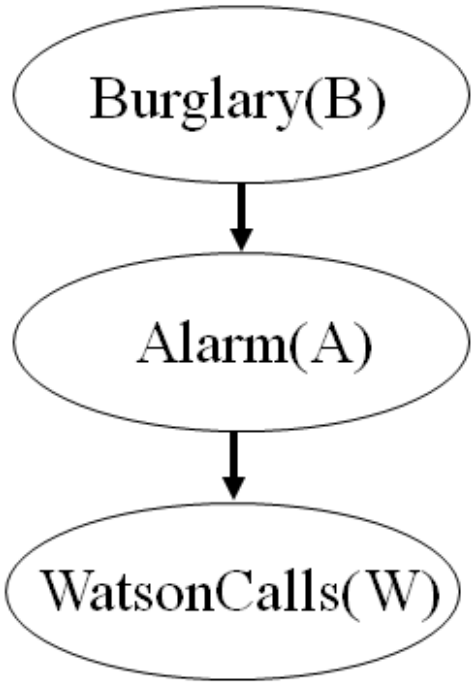
Normalization:
$$P(A)=P(A\mid B)P(B)+P(A\mid \neg B)P(\neg B)$$
$$=0.98\times 0.01+0.01\times 0.99=0.0197$$

Conditioning:
$$P(B\mid A)=\frac{P(A,B)}{P(A)}=\frac{0.0098}{0.0197}=0.49$$

Watson informs on alarm, but he likes to joke

$$P(W \mid A) = 0.7$$

$$P(W \mid \neg A) = 0.1$$



Joint Distribution:

$$P(W \mid A)P(A \mid B)P(B)$$

Marginalization:

$$P(W, B) = P(W \mid A)P(A \mid B)P(B) + P(W \mid \neg A)P(\neg A \mid B)P(B)$$

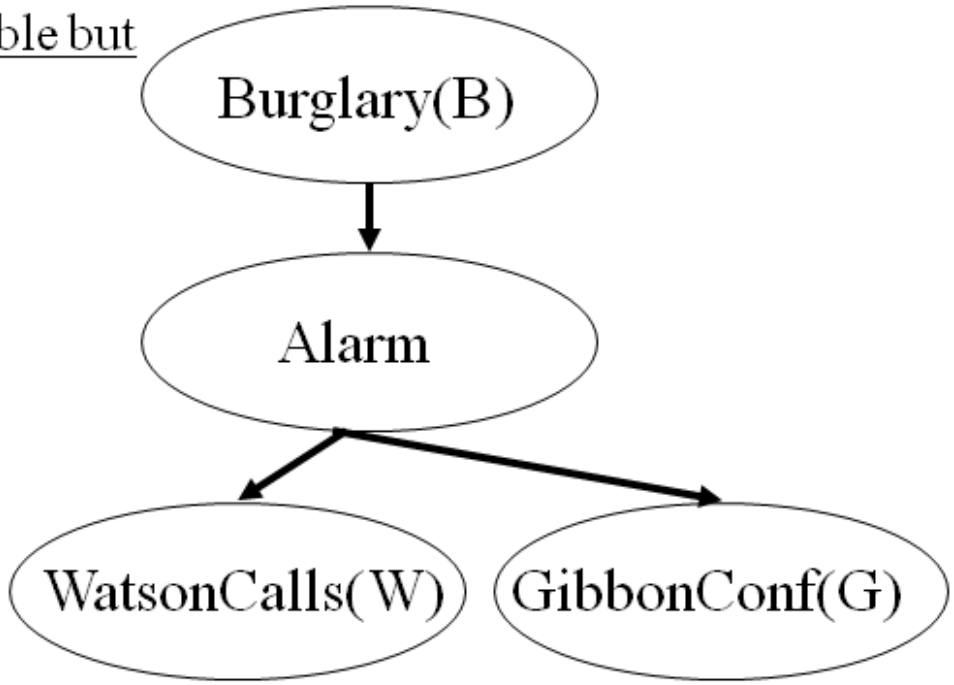
Normalization: $P(W) = P(B, W) + P(\neg B, W)$

Conditioning: $P(B \mid W) = \frac{P(B, W)}{P(W)} = 0.0615$

Holmes calls Mrs Gibbon, who is reliable but has an alcohol problem

$$P(G \mid A) = 0.8$$

$$P(G \mid \neg A) = 0.2$$



Joint Distribution:

$$P(G \mid A)P(W \mid A)P(A \mid B)P(B)$$

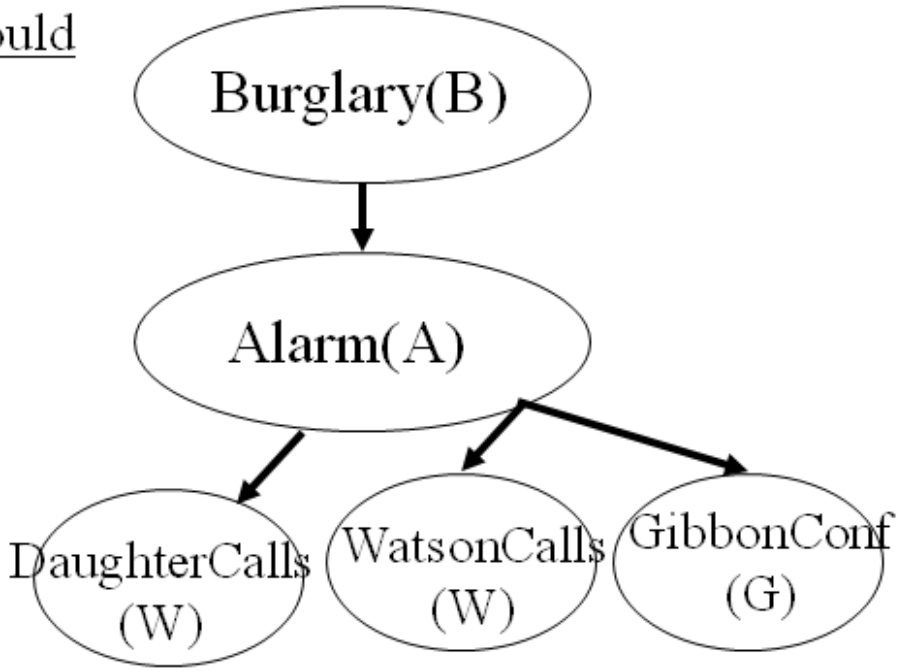
Following the same recipe:

Conditioning:

$$P(B \mid G) = \frac{P(B,G)}{P(G)} = 0.0372$$

$$P(B \mid G,W) = \frac{P(B,G,W)}{P(G,W)} = 0.179$$

Holmes has a reliable daughter who would call, if she had heard the alarm



$$P(D \mid A) = 0.7$$
$$P(D \mid \neg A) = 0.0$$

Joint Distribution:

$$P(D \mid A)P(G \mid A)P(W \mid A)$$
$$\times P(A \mid B)P(B)$$

Following the same recipe:

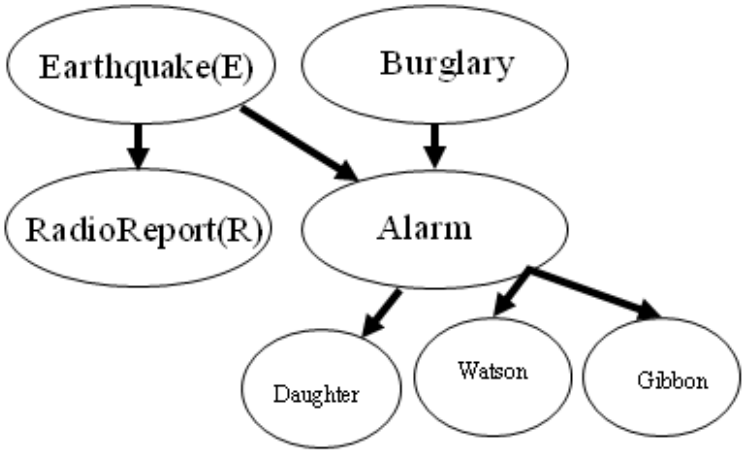
Conditioning: $$P(B \mid D) = \frac{P(B,D)}{P(D)} = 0.49 = P(B \mid A)$$

Since: $$P(A \mid D) = \frac{P(A,D)}{P(D)} = 1$$

Holmes hears in the radio, that there has been an earthquake in his neighborhood

$$P(E) = 0.0001 \qquad P(R \mid E) = 0.4$$

$$P(A \mid E, \neg B) = 0.20$$
$$P(A \mid \neg E, B) = 0.98$$
$$P(A \mid E, B) = 0.9840$$
$$P(A \mid \neg E, \neg B) = 0.01$$



Joint Distribution:

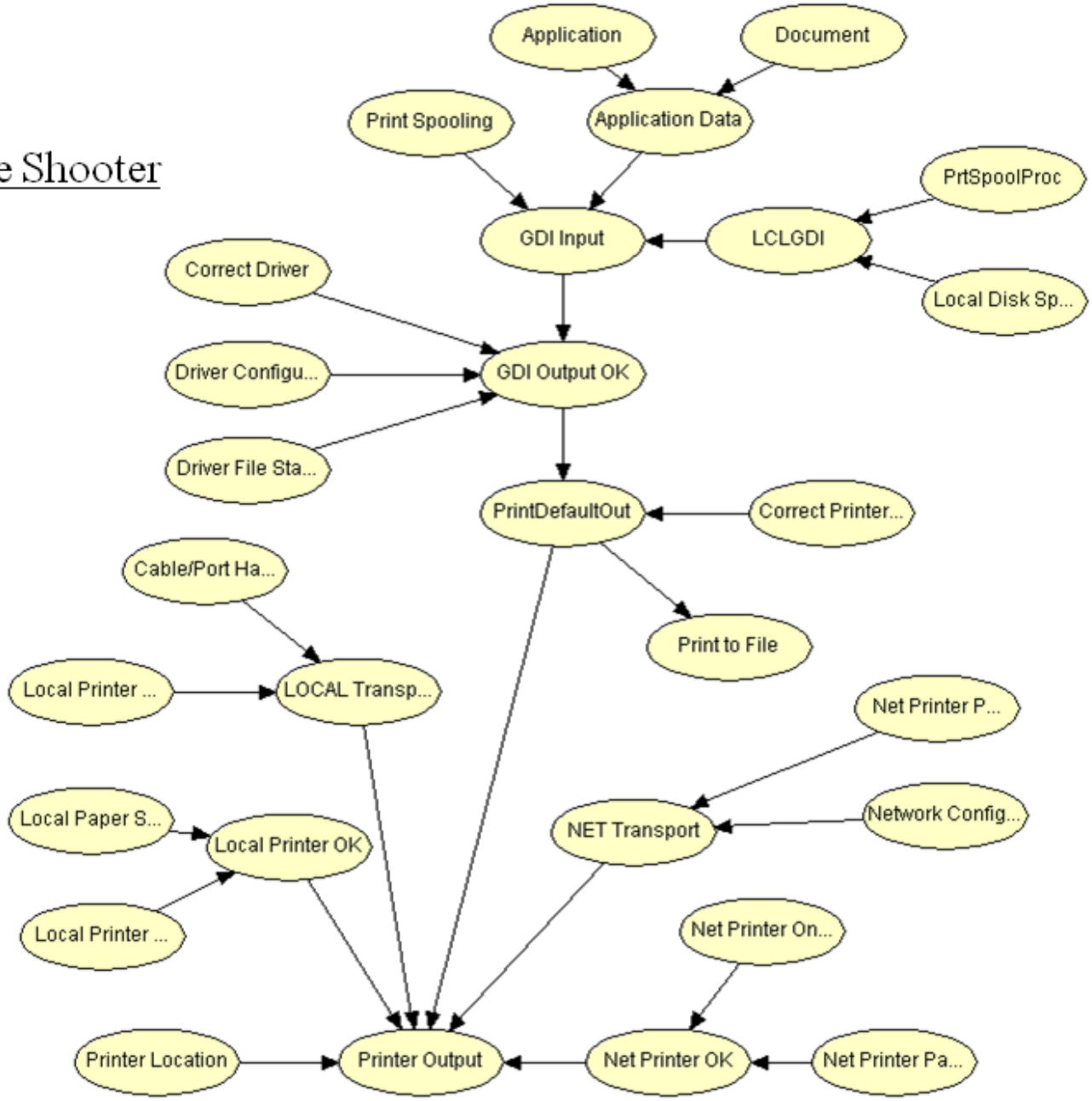
$$P(D \mid A)P(G \mid A)P(W \mid A)P(A \mid B, E)P(B)P(E)P(R \mid E)$$

$$P(B \mid D) = 0.49 \qquad P(B \mid D, R) = 0.0473$$

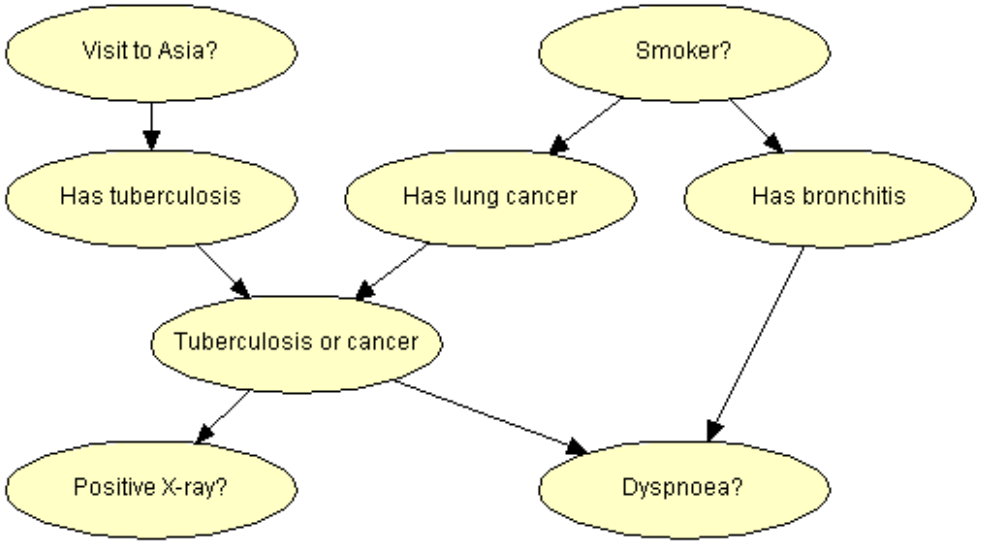
Explaining away: since the unlikely event of an earthquake explains the alarm, it makes the burglary less likely

Printer Trouble Shooter

(Microsoft)



Chest Clinic



A real Bayes net: Alarm

Domain: Monitoring Intensive-Care Patients

- 37 variables
- 509 parameters

…instead of 2^{37}

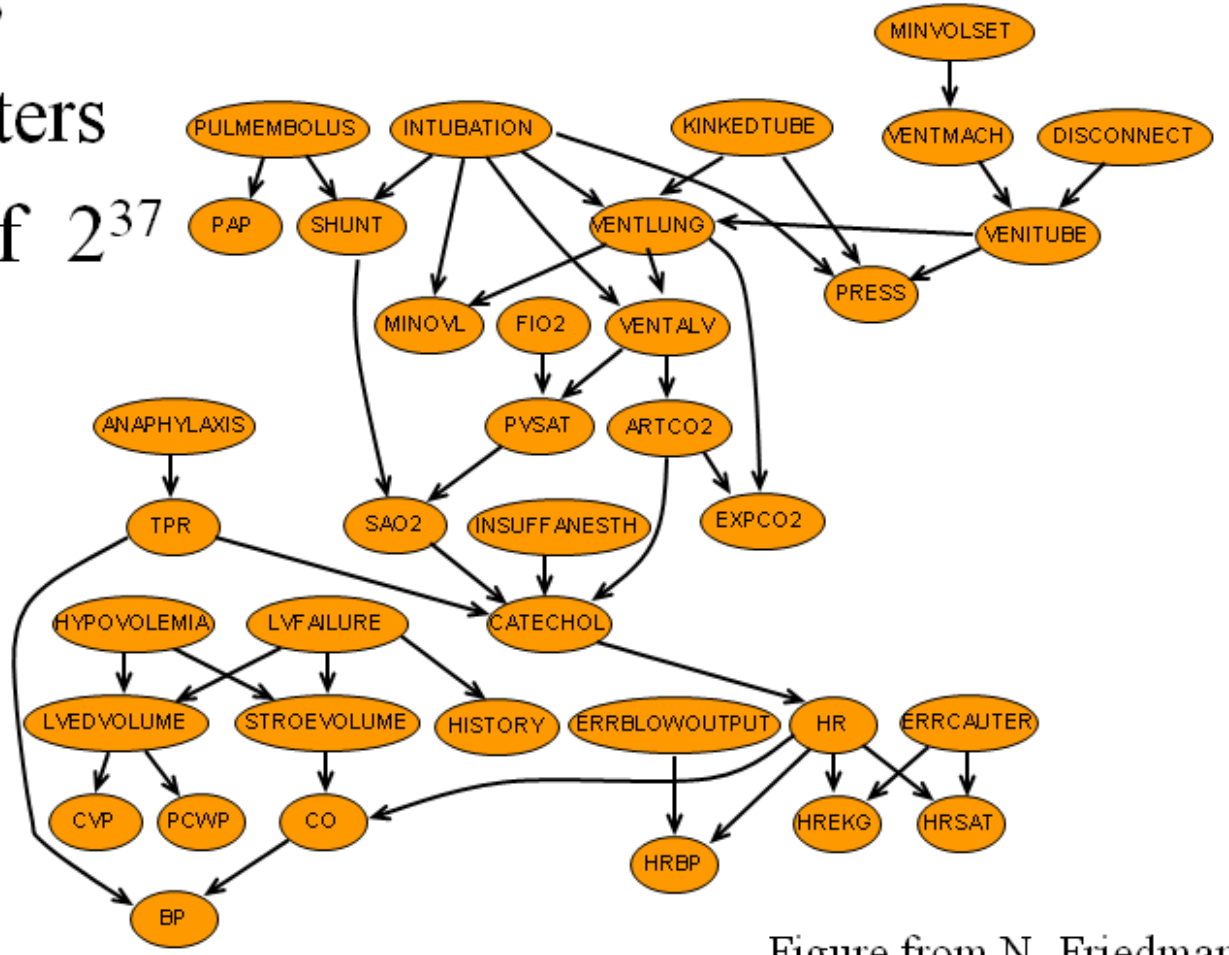


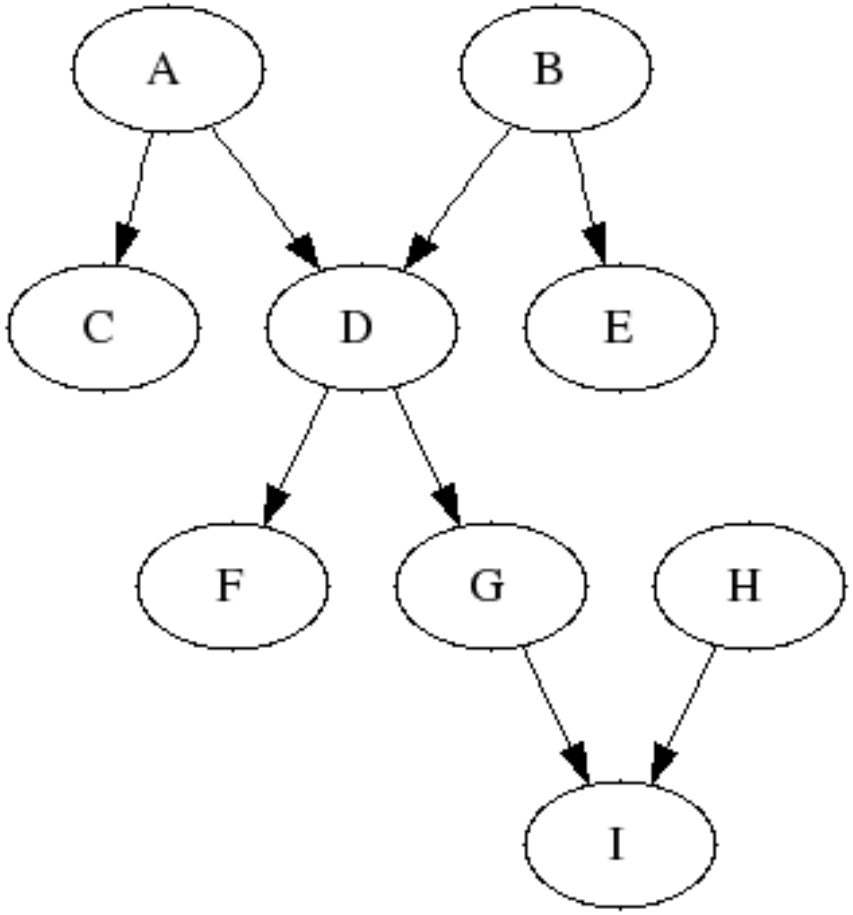
Figure from N. Friedman

Propositional Level and Template Level

- Note that the Holmes Network contains propositional Boolean variables such as Alarm or Burglary. Also, the Alarm Network might only be applicable to a particular clinic. Thus the Holmes Network and the Alarm Network are examples of propositional Bayesian networks. This can be related to propositional logic
- In contrast the Chest Clinic example and the Printer Trouble Shooter example are applicable to any patient, resp. to any printer. Thus here the Bayes net defines a template, applicable to all patients/printers. The formulation of templates which can be applied to many entities is a property of predicate logic

Inference in Bayes nets without Cycles in Undirected Net

- By construction there are no cycles in the directed net; the structure of a Bayesian net is a *directed acyclic graph* (DAG)
- But there might be cycles when one ignores the directions
- Let's first consider the simpler case without cycles in the undirected graph; the structure of the the Bayes net is a poly-tree: there is at most one directed path between two nodes


A
B
C
D
E
F
G
H
I

Marginalization

- The marginalization can be computationally expensive: the sum contains exponentially many terms; for binary variables $2^{|\mathcal{X}^r|}$
- The goal is now to exploit the structure of the net to calculate the marginalization efficient

Example: Markov Chain (prior evidence)

- Consider the Markov chain of length M with binary variables where the first variable $X_1 = x_1$ and the last variable $X_M = x_M$ are known
- We can apply the iterative formula for propagating information on $X_1 = x_1$ **down the chain** (prior evidence), $i = 2, 3, \ldots$

$$\pi_i(x_i) \equiv P(X_i = x_i | X_1 = x_1) = \sum_{x_{i-1}} P(X_i = x_i, X_{i-1} = x_{i-1} | X_1 = x_1)$$

$$= \sum_{x_{i-1}} P(X_i = x_i | X_{i-1} = x_{i-1}) P(X_{i-1} = x_{i-1} | X_1 = x_1)$$

$$= \sum_{x_{i-1}} P(X_i = x_i | X_{i-1} = x_{i-1}) \pi_{i-1}(x_{i-1})$$

- Thus prior evidence moves down the chain ($\pi_{i-1}(x_{i-1}) \rightarrow \pi_i(x_i)$). Note that $P(X_i = x_i | X_{i-1} = x_{i-1})$ is defined in the Markov chain

Example: Markov Chain (likelihood evidence)

- We can apply the iterative formula for propagating information on $X_M = x_M$ **up the chain** (likelihood evidence), $i = M-1, M-2, \ldots$

$$\lambda_i(x_i) \equiv P(X_M = x_M | X_i = x_i) = \sum_{X_{i+1}} P(X_M = x_M, X_{i+1} = x_{i+1} | X_i = x_i)$$

$$= \sum_{x_{i+1}} P(X_{i+1} = x_{i+1} | X_i = x_i) P(X_M = x_M | X_{i+1})$$

$$= \sum_{x_{i+1}} P(X_{i+1} = x_{i+1} | X_i = x_i) \lambda_{i+1}(x_{i+1})$$

- Thus likelihood evidence moves up the chain ($\lambda_{i+1}(x_{i+1}) \rightarrow \lambda_i(x_i)$). Again, $P(X_i = x_i | X_{i-1} = x_{i-1})$ is defined in the Markov chain

Posterior Probability

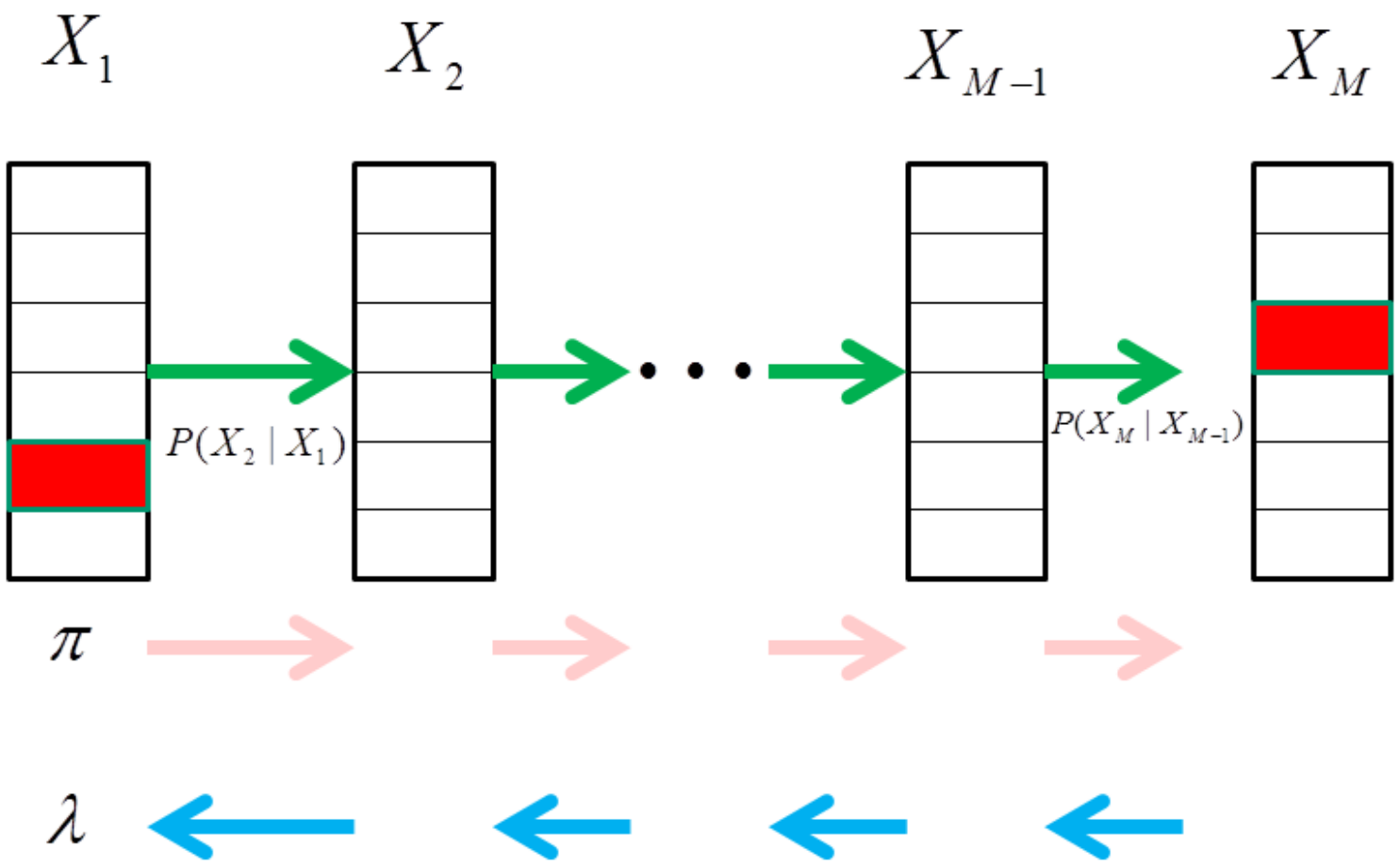
- Then we can calculate for any X_i

$$P(X_i = x_i | X_1 = x_1, X_M = x_M) \propto P(X_i = x_i, X_1 = x_1, X_M = x_M) =$$

$$P(X_1 = x_1) P(X_i = x_i | x_1) P(x_M | X_i = x_i) \propto \pi_i(x_i) \lambda_i(x_i)$$

With normalization

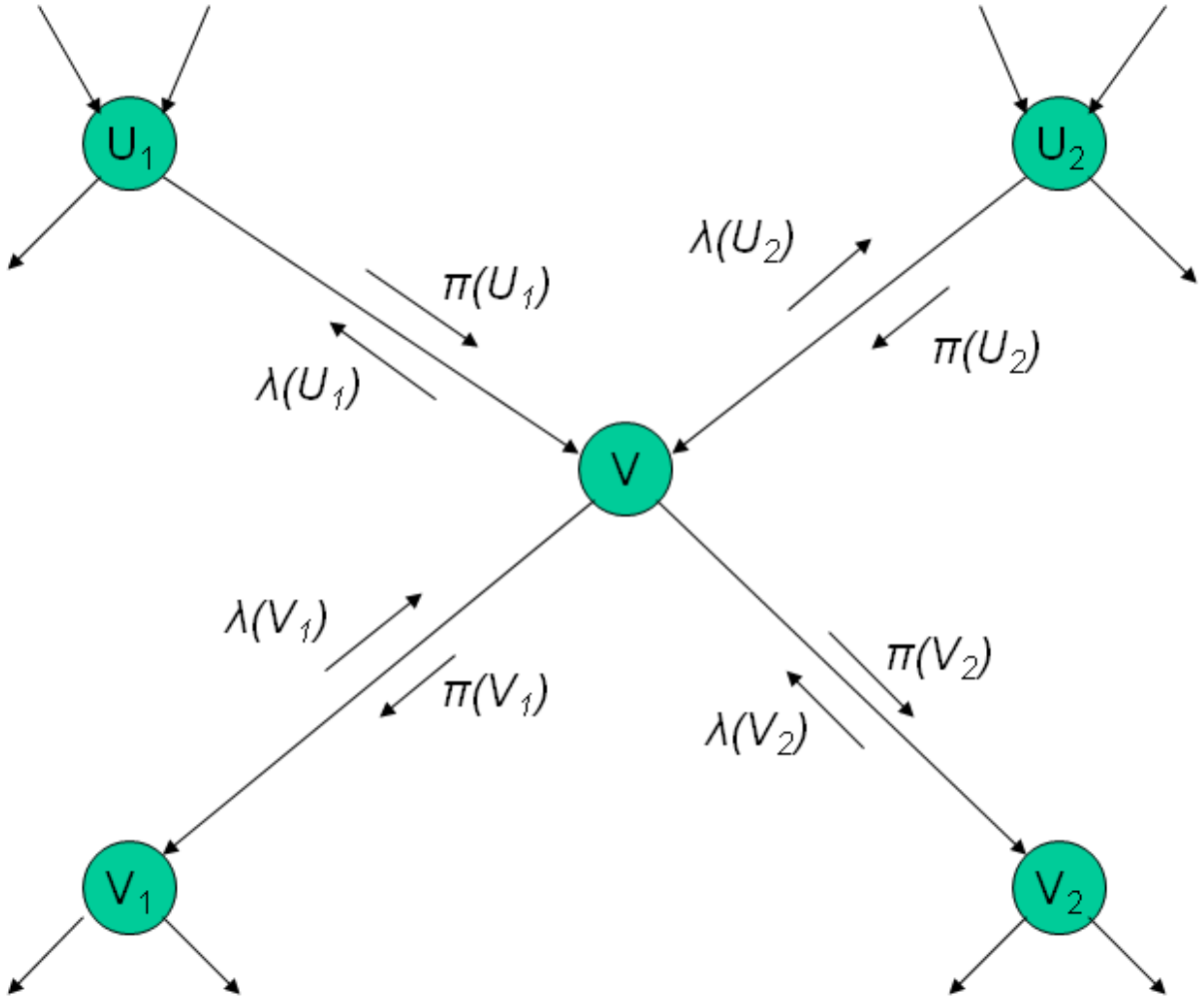
$$P(X_i = x_i | X_1 = x_1, X_M = x_M) = \frac{\pi_i(x_i)\lambda_i(x_i)}{\sum_{i'} \pi_i(x_{i'})\lambda_i(x_{i'})}$$


X_1
X_2
X_{M-1}
X_M
$P(X_2 \mid X_1)$
$P(X_M \mid X_{M-1})$
π
λ

Propagation in Polytrees

- Inference in polytrees can be performed in a similar (but more complex) way
- π and λ propagation is generalized to polytrees

Pearl's Belief Propagation



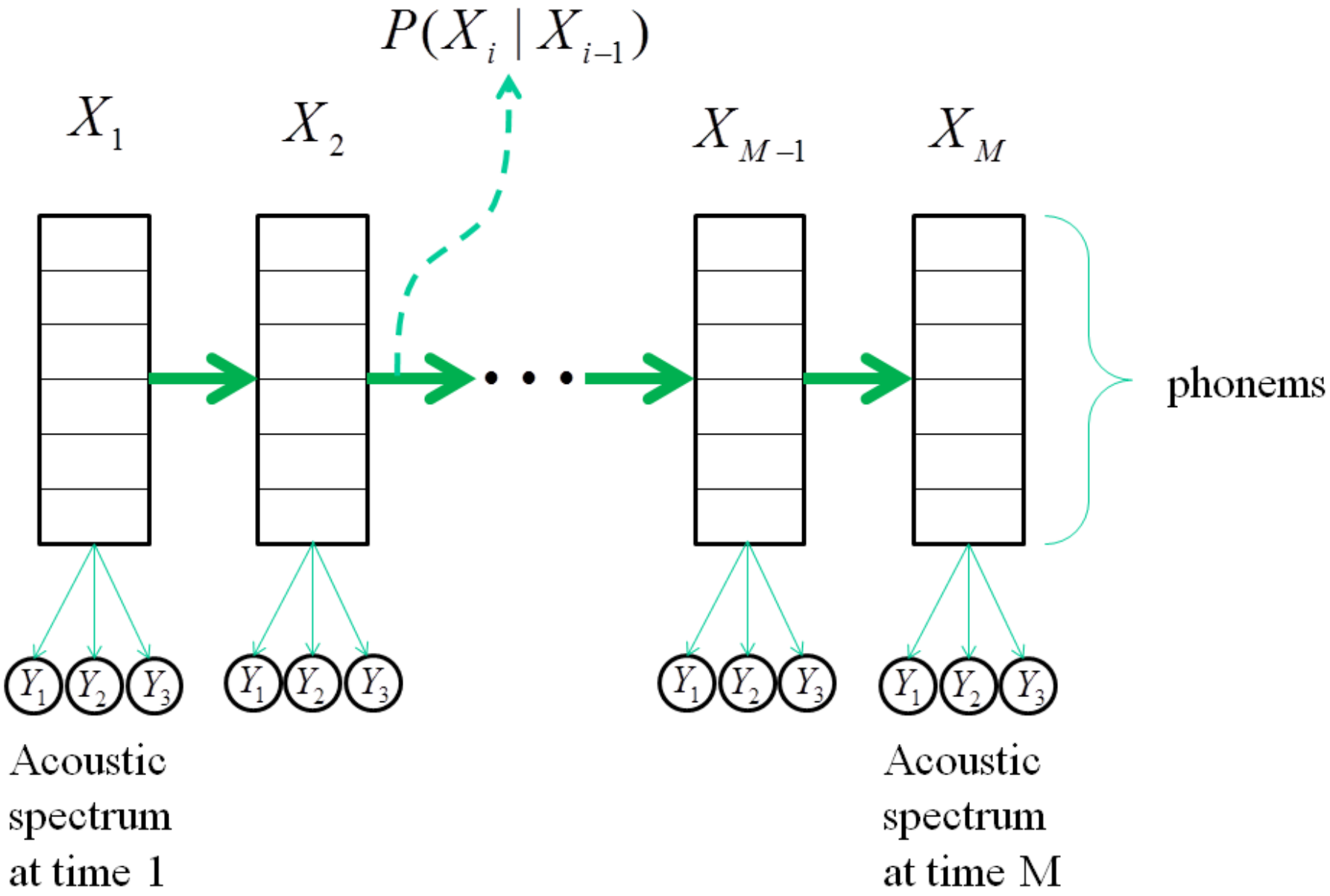
Max-Propagation

- With similar efficiency, one can calculate the maximum likely configuration (Max-product Rule)

$$\mathbf{x}^r_{max} = \arg\max_{\mathbf{x}^r} P(\mathcal{X}^r = \mathbf{x}^r, \mathcal{X}^m = \mathbf{x}^m)$$

Hidden Markov Models

- Hidden Markov Models (HMMs) are the basis of modern speech recognition systems. An HMM is a Bayesian network with latent variables
- States corresponds to phonemes; measurements correspond to the acoustic spectrum
- The HMM contains the transition probability between states $P(X_i|X_{i-1})$ and emission probabilities $P(Y|X)$.
- To find the most likely sequence of states, the Viterbi Algorithm is employed, which is identical to the Max-Prop Algorithm


$P(X_i \mid X_{i-1})$
X_1
X_2
X_{M-1}
X_M
phonems
Y_1 Y_2 Y_3
Acoustic spectrum at time 1
Acoustic spectrum at time M

Loopy Belief Propagation: Belief Propagation in Bayes Nets with Loops

- When there a loops in the undirected Bayes net, belief propagation is not applicable: There cannot be a local message passing rule since information arriving at a node from different paths can be correlated
- *Loopy Belief Propagation* the application of belief propagation to Bayes nets with cycles (although strictly not correct)
- The local update rules are applied until convergence is achieved (which is not always the case)
- Loopy Belief Propagation is applied in Probabilistic Relational Models which are typically large Bayes nets describing domains where relationships are important, e.g., friend of

Loopy Belief Propagation in Decoding

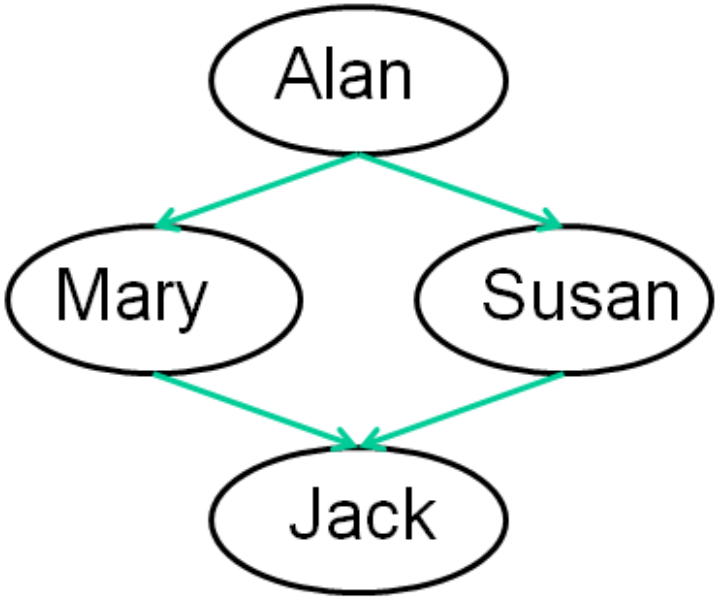
- Most interesting and surprising: *Turbo codes* (Berrou) and *Low-Density Parity-Check (LDPC) Codes* (Gallager) use Loopy Belief Propagation for decoding. Both become very close to the Shannon limit and require long code words and thus produce delays. Thus they are not useful for interactive communication but for broadcasting of information (mobile communication) and in space applications (NASA)
- Supposedly: 3G cell phones use turbo codes, 4G phones use Gallager's LDPC codes

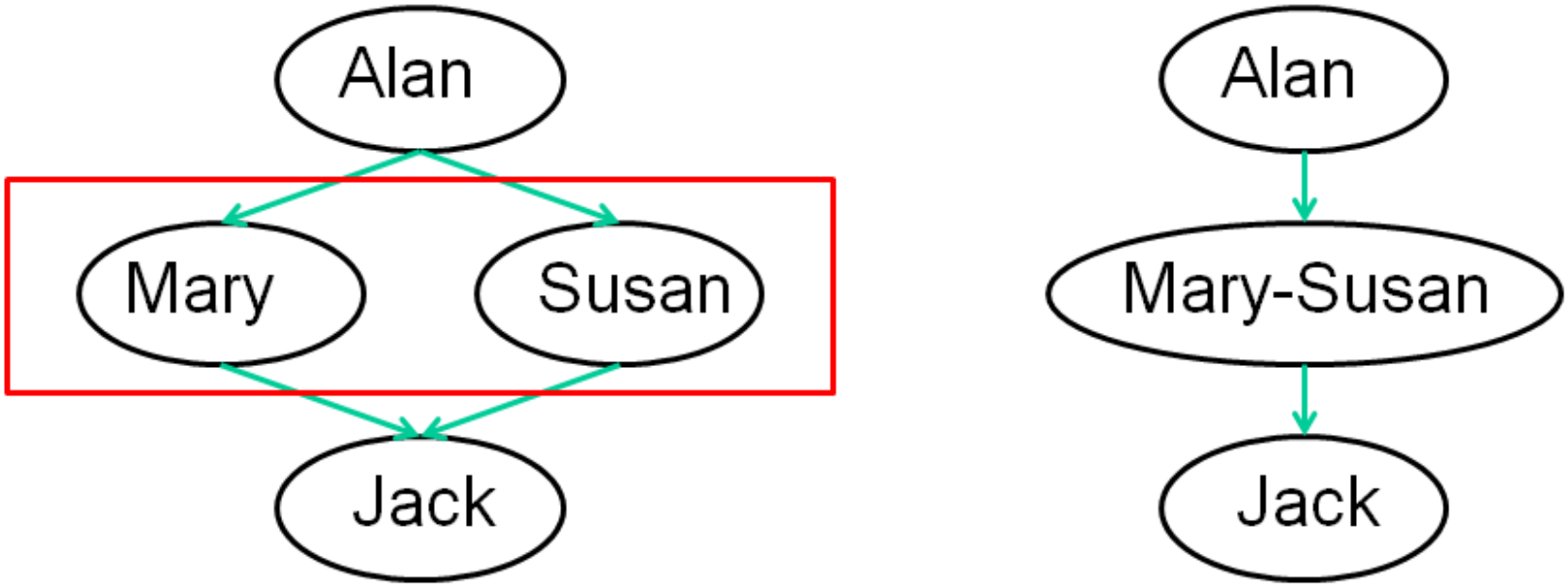
Junction tree algorithm: Correct Inference

- Most (non-)commercial Bayes nets contain the junction tree algorithm which realizes correct probabilistic inference
- The junction tree algorithm combined variables such that the resulting net has no loops
- The junction tree can be inefficient when the combines variables have many states

Junction tree algorithm: Basic Idea

- By combining variables to form a new super variable, one can remove the loops
- In the example, we see a loop in the network. The nodes *Mary* and *Susan* have each two states. The super variable *Mary-Susan* has 4 states.


Alan
Mary
Susan
Jack


Alan
Mary
Susan
Jack
Alan
Mary-Susan
Jack

Design of a Bayes Net

- The expert needs to be clear about the important variables in the domain
- The expert must indicate direct causal dependencies by specifying the directed link in the net
- The expert needs to quantify the causal dependencies: define the conditional probability tables
- This can be challenging if a node has many parents: if a binary node has n binary parents, then the expert needs to specify 2^n numbers!
- To simplify this task one often makes simplifying assumptions; the best-known one is the Noisy-Or Assumption

Noisy-Or

- Let $X \in \{0, 1\}$ be a binary node with binary parents $U_1, \ldots, U_n$
- Let q_i be the probability, that $X = 0$ (no symptom), when $U_i = 1$ (disease) and all other (diseases) $U_j = 0$; this means, that $c_i = 1 - q_i$ is the influence of a single parent (disease) on X (the symptom)
- Then one assumes, $P(X = 0|U_1, \ldots, U_n) = \prod_{i=1}^{n} q_i^{U_i}$, or equivalently,

$$P(X = 1|U_1, \ldots, U_n) = 1 - \prod_{i=1}^{n} q_i^{U_i}$$

- This means that if several diseases are present that can cause the symptom, then the probability of the symptom increases (if compared to the probabilities for the single disease

Maximum Likelihood Learning with Complete Data

- We assume that all nodes in the Bayesian net have been observed for N instances (e.g., N patients)
- ML-parameter estimation simply means counting
- Let $\theta_{i,j,k}$ be defines as

$$\theta_{i,j,k} = P(X_i = k | \text{par}(X_i) = j)$$

- This means that $\theta_{i,j,k}$ is the probability that X_i is in state k, when its parents are in the state j (we assume that the states of the parents can be enumerated in a systematic way)
- Then the ML estimate is

$$\theta^{ML}_{i,j,k} = \frac{N_{i,j,k}}{\sum_k N_{i,j,k}}$$

Here, $N_{i,j,k}$ is the number of samples in which $X_i = k$ and $\text{par}(X_i) = j$

MAP-estimate for Integrating Prior Knowledge

- Often counts are very small and a ML-estimate has high variance
- One simply specifies efficient counts (counts from virtual data) which then can be treated as real counts Let $\alpha_{i,j,k} \geq 0$ be virtual counts for $N_{i,j,k}$
- One obtains the *maximum a posteriori* (MAP) estimate as

$$\theta_{i,j,k}^{MAP} = \frac{\alpha_{i,j,k} + N_{i,j,k}}{\sum_k (\alpha_{i,j,k} + N_{i,j,k})}$$

Missing Data

- The problem of missing data is an important issue in statistical modeling
- In the simplest case, one can assume that data are missing at random
- Data is not missing at random, if for example, I analyse the wealth distribution in a city and rich people tend to refuse to report their income
- For some models the EM (Expectation Maximization)-algorithm can be applied and leads to ML or MAP estimates

EM

- Consider a particular data point l. In the E-step we calculate the probability for marginal probabilities of interest given the known information in that data point d_l and given the current estimates of the parameters $\widehat{\theta}$, using belief proagation or the junction tree algorithm. Then we get

$$E(N_{i,j,k}) = \sum_{l=1}^{N} P(X_i = k, \text{par}(X_i) = j | d_l, \widehat{\theta})$$

  Note that if the parent and child nodes are known then $P(X_i = k, \text{par}(X_i) = j | d_l, \widehat{\theta})$ is either zero or one, otherwise a number between zero and one

- Based on the E-step, we get in the M-step

$$\widehat{\theta}_{i,j,k}^{ML} = \frac{E(N_{i,j,k})}{\sum_k E(N_{i,j,k})}$$

- E-Step and M-Step are iterated until convergence. One can show that EM does not decrease the likelihood in each step; EM might converge to local optima, even if there might only be a global optimum for the model with complete data

EM-Learning with the HMM and the Kalman Filter

- HMMs are trained using the so-called Baum-Welch-Algorithm. This is exactly an EM algorithm
- Offline Versions of the Kalman filter can also be performed via EM

Alternatives for the E-step

- The E-step is really an inference step
- Here, also approximate inference can be used (loopy-belief propagation, MCMC, Gibbs, mean-field)

Beyond Table Representations

- For learning the conditional probabilities, many approaches have been employed
- Decision Trees, Neural networks, log-linear models, ...

Structural Learning in Bayes Nets

- One can also consider learning the structure of a Bayes net and maybe even discover causality
- In structural learning, several points need to be considered
- There are models that are structural equivalent. For example in a net with only two variables A and B one might show that there is statistical correlation between the two variables, but it is impossible to decide if $A \rightarrow B$ or $A \leftarrow B$. Colliders (nodes where arrow-head meet) can make directions identifiable
- In general, the structure of the Bayes model and its parameters model the joint distribution of all the variables under consideration
- Recall that the way the data was collected can also have influence: I will get a different distribution if I consider data from the general population or data collected from patients visiting a specialists in a rich neighborhood

Structure Learning via Greedy Search

- In the most common approaches one defines a cost function and looks for the structure that is optimal under the cost function. One has to deal with many local optima
- Greedy Search: One starts with an initial network (fully connected, empty)and makes local changes (removal of directed link, adding a link, reversing direction of a link, ...) and accepts the change, when the cost function improves
- Greedy search can be started from different initial conditions
- Alternatives: Simulated Annealing, Genetic Algorithms

Cost Functions

- As a cost function one might use a cross-validation set
- Often BIC (Bayesian Information Criterion) is used:

$$\frac{1}{N}\log L - \frac{M}{2N}\log N$$

(M is the number of parameters; N is the number of data points)

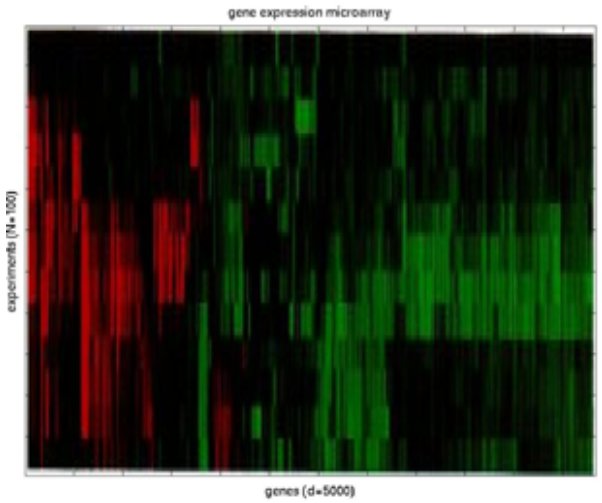
- Recall, that Bayesian model selection is based on $P(D|\mathcal{M})$. With complete data we get,

$$P(D|\mathcal{M}) = \prod_{i=1}^{N}\prod_{j}\frac{\Gamma(\alpha_{i,j})}{\Gamma(\alpha_{i,j}+N_{i,j})}\prod_{k}\frac{\Gamma(\alpha_{i,j,k}+N_{i,j,k})}{\Gamma(\alpha_{i,j,k})}$$

where $\alpha_{i,j} = \sum_k \alpha_{i,j,k}$ and $N_{i,j} = \sum_k N_{i,j,k}$ and where $\Gamma(\cdot)$ is the gamma function

Structure learning (data mining)

Gene expression data



Genetic pathway

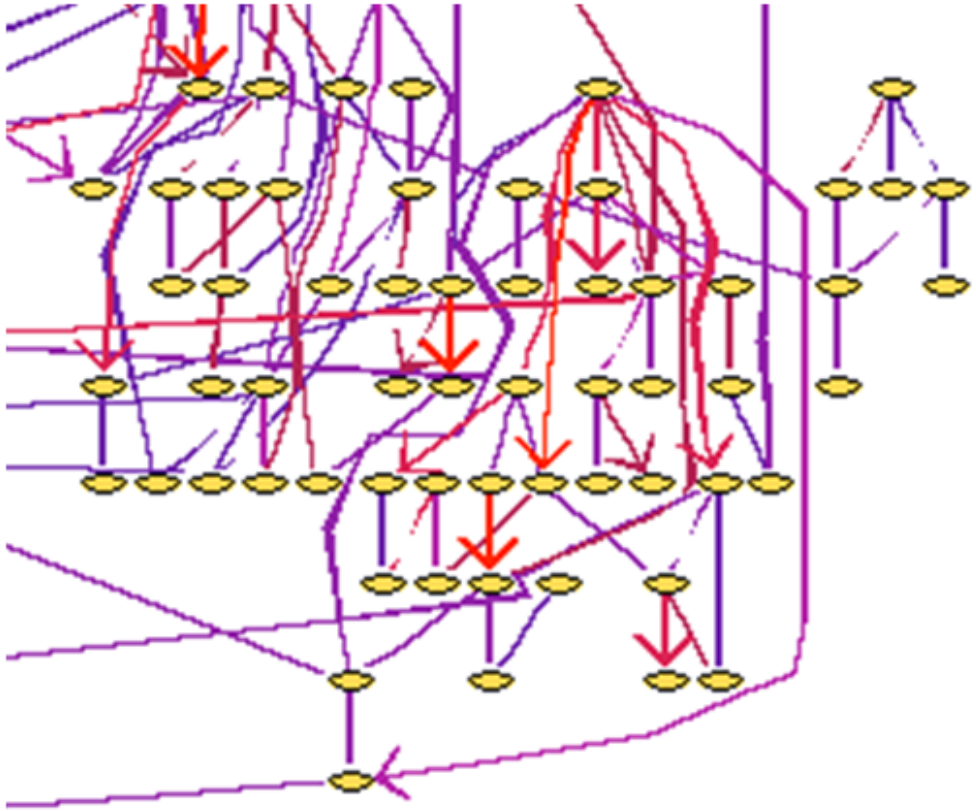
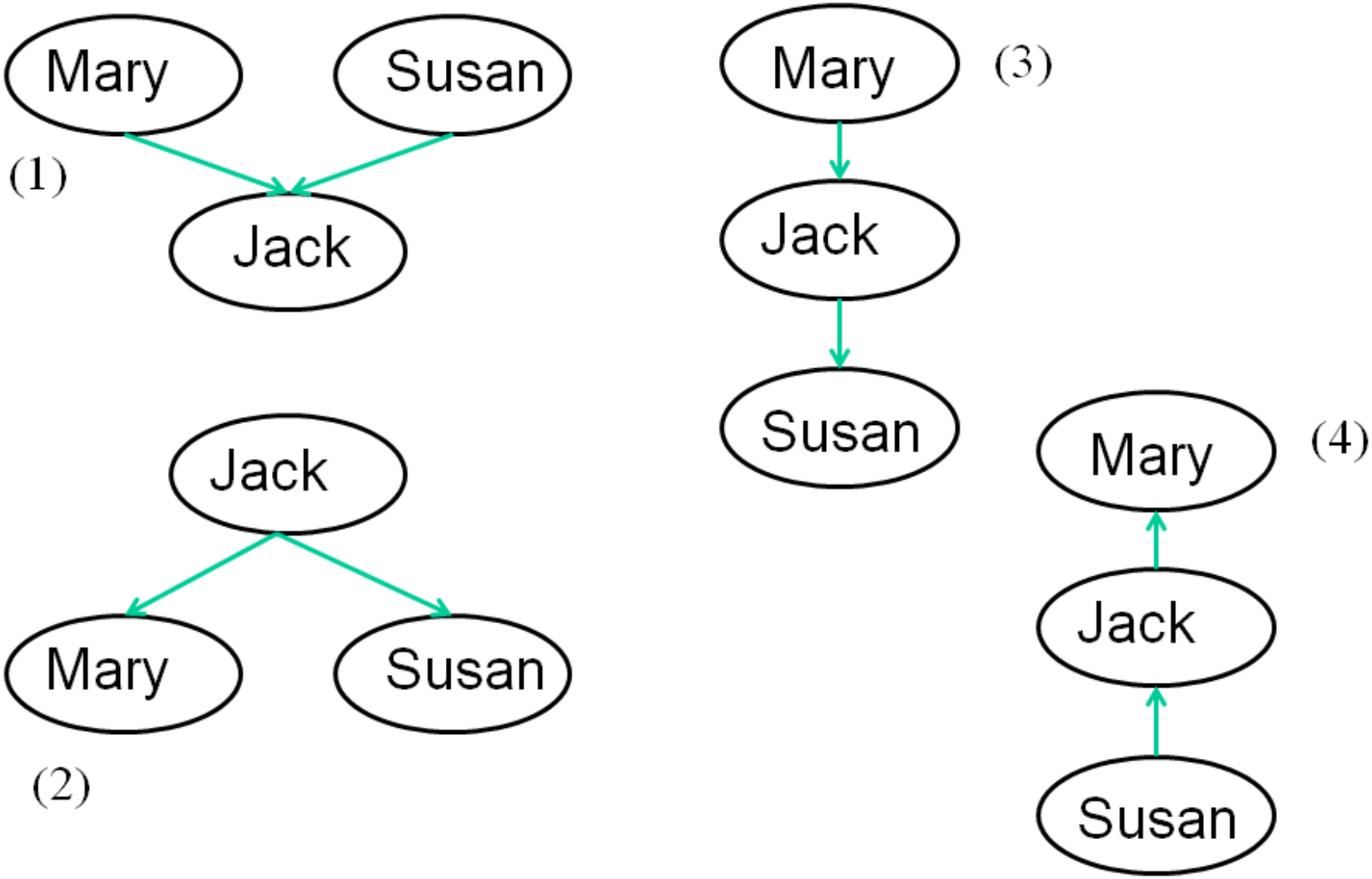


Figure from N. Friedman

Constrained-Based Methods for Structural Learning

- One performs statistical independence tests and uses those tests to decide on network structure
- For the example in the V-structure in the image (1), M and S are marginally mutually independent but they might become dependent given J. J is dependent on both M and S
- In the other structure (2), (3), (4), S and M are dependent when J is unknown. But now M and S become independent given that J is known!
- The structure with the collider (1) can be identified. The structures (2), (3), (4) are all structurally equivalent


Mary
Susan
(1)
Jack
Jack
Mary
Susan
(2)
Mary
(3)
Jack
Susan
Mary
(4)
Jack
Susan

Causal Interpretation of Structure and Parameters

- One has to be very cautious with a causal interpretation of a learned model
- One assumption is that the world under consideration is causal
- Another assumption is that all relevant information in part of the model
- Sometimes temporal information is available which constraints the direction of links (Granger causality)

Concluding Remarks

- Bayesian networks are used as expert systems in medical domains
- The underlying theory can be used to derive the likelihood of complex models (e.g., HMMs)
- Markov Nets are related to Bayesian networks and use undirected edges; they typical do not have a causal interpretation
- Bayes nets and Markov nets are the most important representatives of the class of *graphical models*
- In the lecture we focussed on nets with discrete variables. Also commonly studied are nets with continous variables and Gaussian distributions

A comparison of GM software

Name	Authors	Src.	Cts	GUI	θ	G	Free
Analytica	Lumina	N	Y	W	N	N	N
Bayda	U. Helsinki	Java	Y	Y	Y	N	F
BayesBuilder	Nijman (Nijmegen)	N	N	Y	N	N	N
B. Knl. Disc.	KMI/Open U.	N	D	Y	Y	Y	F
B-course	U. Helsinki	N	D	Y	Y	Y	F
BN pow. cstr.	Cheng (U.Alberta)	N	N	Y	Y	Y	F
BN Toolbox	Murphy (UCB)	Matlab	Y	N	Y	Y	F
BucketElim	Rish (UCI)	C++	N	N	N	N	F
BUGS	MRC/Imperial	N	Y	W	Y	N	F

www.ai.mit.edu/~murphyk/Software/Bayes/bnsoft.html

CIspace	Poole (UBC)	Java	N	Y	N	N	F
Ergo	Noetic Systems	N	N	Y	N	N	N
Genie/Smile	U. Pittsburgh	N	N	W	N	N	F
Hugin Light	Hugin	N	Y	W	N	N	N
Ideal	Rockwell	Lisp	N	Y	N	N	F
Java Bayes	Cozman (CMU)	Java	N	Y	N	N	F
MIM	HyperGraph	N	Y	Y	Y	Y	N
MSBN	Microsoft	N	N	W	N	N	F
Netica	Norsys	N	Y	W	Y	N	N
Pronel	Hugin	N	N	W	Y	Y	F
RISO	Dodier (Colorado)	Java	Y	Y	N	N	F
Tetrad	CMU	N	Y	N	Y	Y	F