CS 2750 Machine Learning
Lecture 13

Bayesian belief networks. Inference.

Milos Hauskrecht
milos@cs.pitt.edu
5329 Sennott Square

Midterm exam

Monday, March 17, 2003

- **In class**
- **Closed book**
- **Material covered by Wednesday, March 12**
- **Last year midterm will be posted on the web**

Project proposals

Due: **Monday, March 24, 2002**

- **1-2 pages long**

Proposal

- **Written proposal:**
  1. Outline of a learning problem, type of data you have available. Why is the problem important?
  2. Learning methods you plan to try and implement for the problem. References to previous work.
  3. How do you plan to test, compare learning approaches
  4. Schedule of work (approximate timeline of work)
- **Short 5 minute PPT presentation summarizing points 1-4**

Modeling uncertainty with probabilities

- **Full joint distribution:** joint distribution over all random variables defining the domain
  - it is sufficient to represent the complete domain and to do any type of probabilistic inferences

Problems:

- **Space complexity.** To store full joint distribution requires to remember $O(d^n)$ numbers.

  n – number of random variables, d – number of values
- **Inference complexity.** To compute some queries requires $O(d^n)$ steps.
- **Acquisition problem.** Who is going to define all of the probability entries?

Pneumonia example. Complexities.

- **Space complexity**.
  - Pneumonia (2 values: T,F), Fever (2: T,F), Cough (2: T,F), WBCcount (3: high, normal, low), paleness (2: T,F)
  - Number of assignments: 2*2*2*3*2=48
  - We need to define at least 47 probabilities.
- **Time complexity**.
  - Assume we need to compute the probability of Pneumonia=T from the full joint

$$P(Pneumonia = T) =$$
$$= \sum_{i \in T,F} \sum_{j \in T,F} \sum_{k=h,n,l} \sum_{u \in T,F} P(Fever = i, Cough = j, WBCcount = k, Pale = u)$$

  - Sum over 2*2*3*2=24 combinations

Bayesian belief networks (BBNs)

Bayesian belief networks.

- Represent the full joint distribution over the variables more compactly with a **smaller number of parameters**.
- Take advantage of **conditional and marginal independences** among random variables

- **A and B are independent**

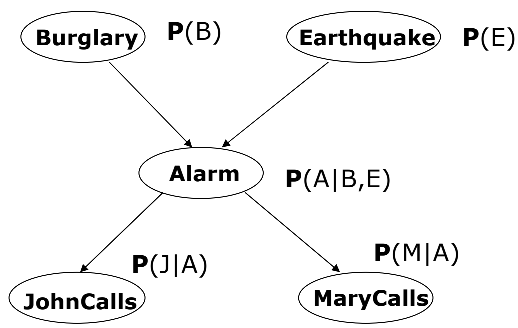
$$P(A, B) = P(A)P(B)$$

- **A and B are conditionally independent given C**

$$P(A, B \mid C) = P(A \mid C)P(B \mid C)$$
$$P(A \mid C, B) = P(A \mid C)$$

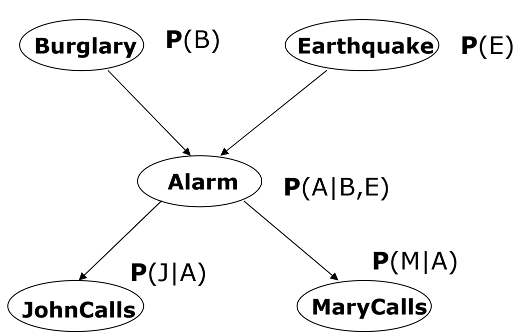
Bayesian belief network.

1. **Directed acyclic graph**
   - **Nodes** = random variables
   - **Links** = missing links encode independences.

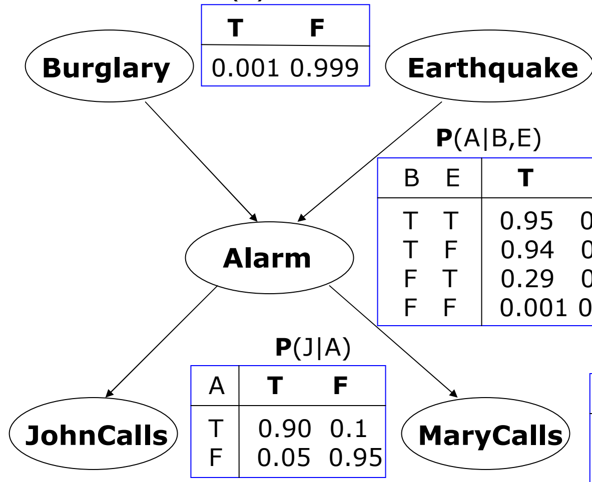


Bayesian belief network.

2. **Local conditional distributions**
   - relate variables and their parents



Bayesian belief network.



P(B)

T	**F**
0.001	0.999

P(E)

T	**F**
0.002	0.998

P(A|B,E)

B	E	**T**	**F**
T	T	0.95	0.05
T	F	0.94	0.06
F	T	0.29	0.71
F	F	0.001	0.999

P(J|A)

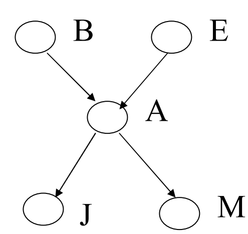
A	**T**	**F**
T	0.90	0.1
F	0.05	0.95

P(M|A)

A	T	F
T	0.7	0.3
F	0.01	0.99

Bayesian belief networks (general)

Two components: $B = (S, \Theta_S)$



- **Directed acyclic graph**
  - Nodes correspond to random variables
  - (Missing) links encode independences
- **Parameters**
  - Local conditional probability distributions for every variable-parent configuration

$$\mathbf{P}(X_i \mid pa(X_i))$$

Where:

$pa(X_i)$ - stand for parents of X_i

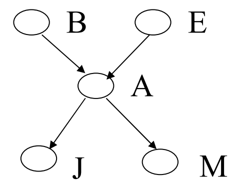
P(A|B,E)

B	E	**T**	**F**
T	T	0.95	0.05
T	F	0.94	0.06
F	T	0.29	0.71
F	F	0.001	0.999

Full joint distribution in BBNs

Full joint distribution is defined in terms of local conditional distributions (obtained via the chain rule):

$$\mathbf{P}(X_1, X_2, .., X_n) = \prod_{i=1,..n} \mathbf{P}(X_i \mid pa(X_i))$$



Example:

Assume the following assignment of values to random variables

$B=T, E=T, A=T, J=T, M=F$

Then its probability is:

$$P(B=T, E=T, A=T, J=T, M=F) =$$

$$P(B=T)P(E=T)P(A=T \mid B=T, E=T)P(J=T \mid A=T)P(M=F \mid A=T)$$

Bayesian belief networks (BBNs)

Bayesian belief networks

- Represent the full joint distribution over the variables more compactly using the product of local conditionals.
- **But how did we get to local parameterizations?**

Answer:

- **Graphical structure** encodes **conditional and marginal independences** among random variables
- **A and B are independent** $P(A, B) = P(A)P(B)$
- **A and B are conditionally independent given C**

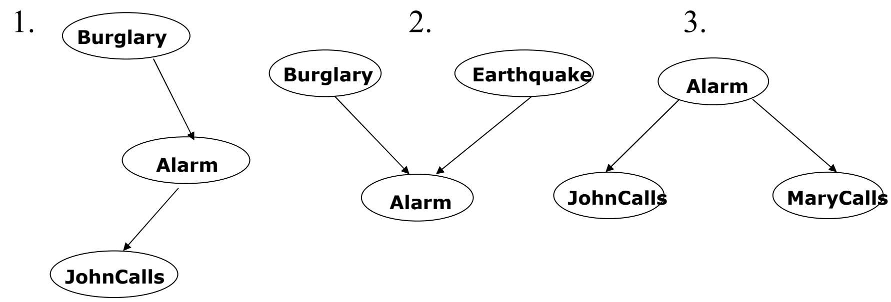
$$P(A \mid C, B) = P(A \mid C)$$

$$P(A, B \mid C) = P(A \mid C)P(B \mid C)$$

- **The graph structure implies the decomposition !!!**

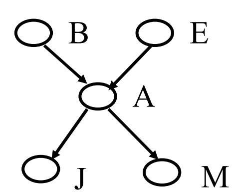
Independences in BBNs

3 basic independence structures:



Full joint distribution in BBNs

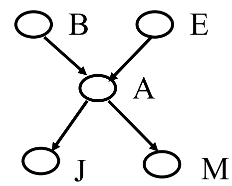
Rewrite the full joint probability using the product rule:



$$P(B=T,E=T,A=T,J=T,M=F)=$$

Full joint distribution in BBNs

Rewrite the full joint probability using the product rule:



$$P(B=T,E=T,A=T,J=T,M=F)=$$

$$=P(J=T\,|\,B=T,E=T,A=T,M=F)P(B=T,E=T,A=T,M=F)$$

$$=\underline{P(J=T\,|\,A=T)}P(B=T,E=T,A=T,M=F)$$

$$P(M=F\,|\,B=T,E=T,A=T)P(B=T,E=T,A=T)$$

$$\underline{P(M=F\,|\,A=T)}P(B=T,E=T,A=T)$$

$$\underline{P(A=T\,|\,B=T,E=T)}P(B=T,E=T)$$

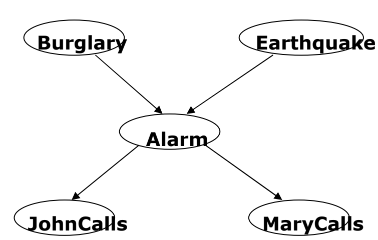
$$P(B=T)P(E=T)$$

$$=P(J=T\,|\,A=T)P(M=F\,|\,A=T)P(A=T\,|\,B=T,E=T)P(B=T)P(E=T)$$

Parameter complexity problem

- In the BBN the full joint distribution is expressed as a product of conditionals (of smaller) complexity

$$\mathbf{P}(X_1, X_2, .., X_n) = \prod_{i=1,..n} \mathbf{P}(X_i \mid pa(X_i))$$



Parameters:

full joint: $2^5 = 32$

BBN: $2^3 + 2(2^2) + 2(2) = 20$

Parameters to be defined:

full joint: $2^5 - 1 = 31$

BBN: $2^2 + 2(2) + 2(1) = 10$

Model acquisition problem

The structure of the BBN typically reflects causal relations

- BBNs are also sometime referred to as **causal networks**
- Causal structure is very intuitive in many applications domain and it is relatively easy to obtain from the domain expert

Probability parameters of BBN correspond to conditional distributions relating a random variable and its parents only

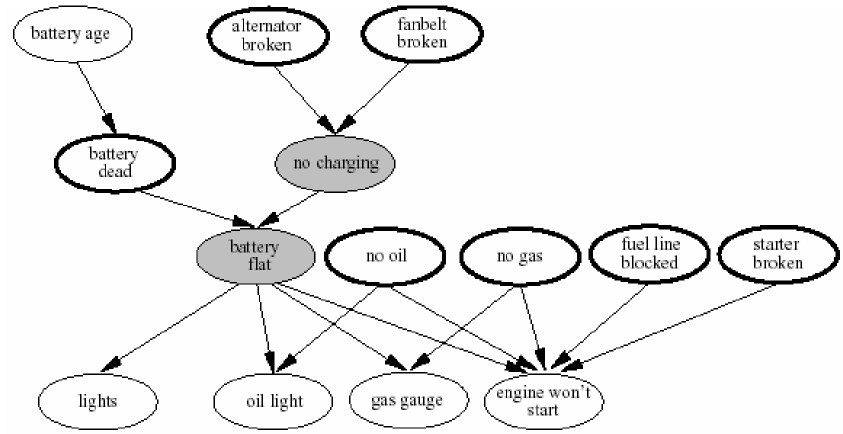
- Their complexity much smaller than the full joint
- Easier to come up (estimate) the probabilities from expert or automatically by learning from data

BBNs built in practice

- **In various areas:**
  - Intelligent user interfaces (Microsoft)
  - Troubleshooting, diagnosis of a technical device
  - Medical diagnosis:
    - Pathfinder (Intellipath)
    - CPSC
    - Munin
    - QMR-DT
  - Collaborative filtering
  - Military applications
  - Insurance, credit applications

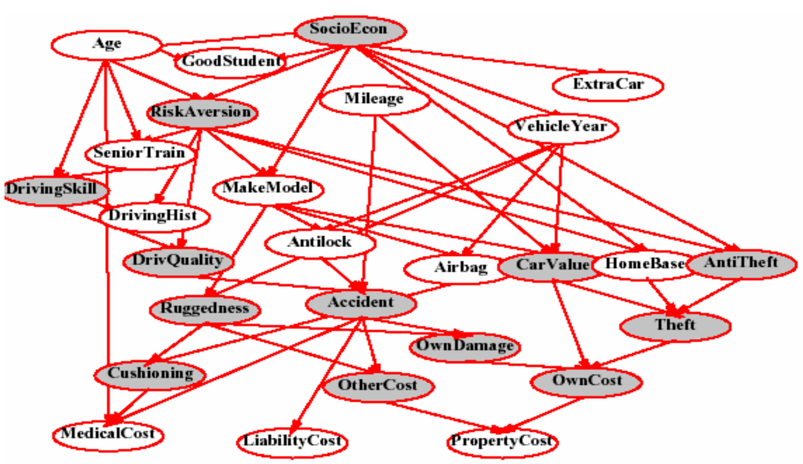
Diagnosis of car engine

- Diagnose the engine start problem

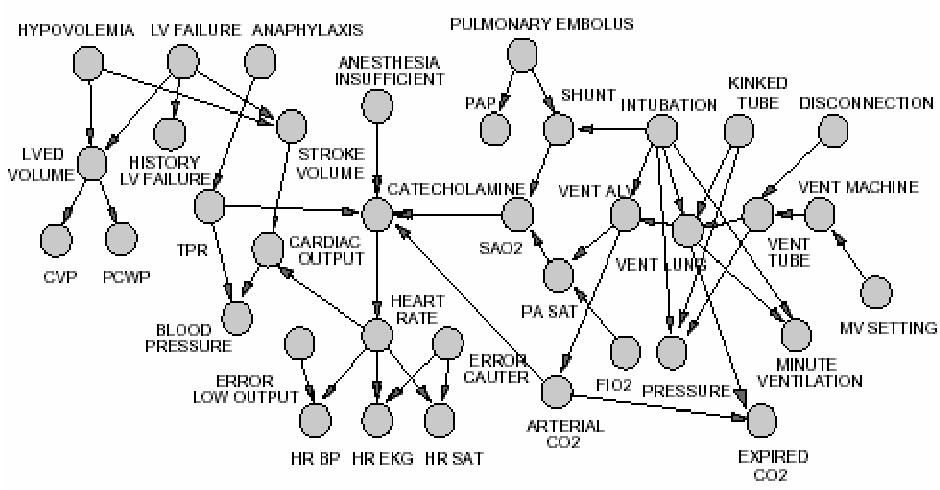


Car insurance example

- Predict claim costs (medical, liability) based on application data

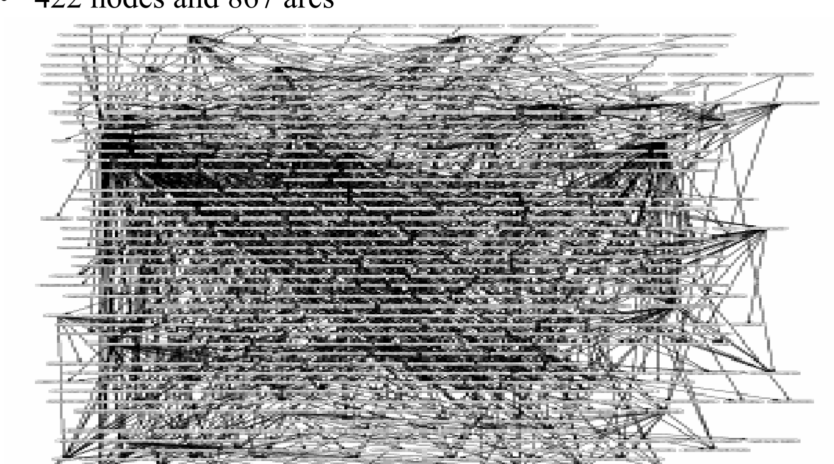


(ICU) Alarm network



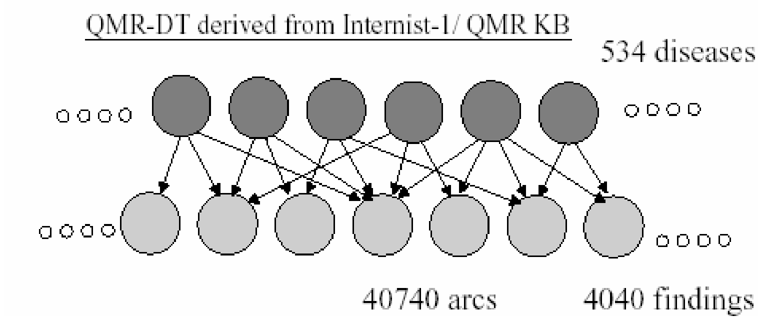
CPCS

- **C**omputer-based **P**atient **C**ase **S**imulation system (CPCS-PM) developed by Parker and Miller (at University of Pittsburgh)
- 422 nodes and 867 arcs



QMR-DT

- **Medical diagnosis in internal medicine**
  - Bipartite network of disease/findings relations
  - Derived from the Internist-1/QMR knowledge base



Inference in Bayesian networks

- BBN models compactly the full joint distribution by taking advantage of existing independences between variables
  - Smaller number of parameters
- But we are interested in solving various **inference tasks**:
  - **Diagnostic task. (from effect to cause)**

    $$\mathbf{P}(Burglary \mid JohnCalls = T)$$

  - **Prediction task. (from cause to effect)**

    $$\mathbf{P}(JohnCalls \mid Burglary = T)$$

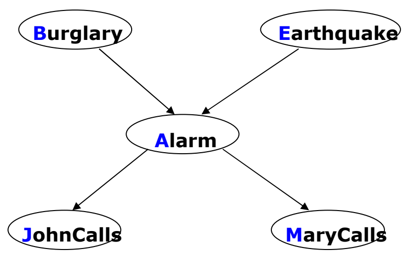
  - **Other probabilistic queries** (queries on joint distributions).

    $$\mathbf{P}(Alarm)$$

- **Question:** Can we take advantage of independences to construct special algorithms and speedup the inference?

Inference in Bayesian network

- **Bad news:**
  - Exact inference problem in BBNs is NP-hard (Cooper)
  - Approximate inference is NP-hard (Dagum, Luby)
- **But** very often we can achieve significant improvements
- Assume our Alarm network



- Assume we want to compute: $P(J = T)$

Inference in Bayesian networks

Computing: $P(J = T)$

Approach 1. Blind approach.

- Sum out all un-instantiated variables from the full joint,
- express the joint distribution as a product of conditionals

$$P(J = T) =$$

$$= \sum_{b \in T,F} \sum_{e \in T,F} \sum_{a \in T,F} \sum_{m \in T,F} P(B = b, E = e, A = a, J = T, M = m)$$

$$= \sum_{b \in T,F} \sum_{e \in T,F} \sum_{a \in T,F} \sum_{m \in T,F} P(J = T \mid A = a) P(M = m \mid A = a) P(A = a \mid B = b, E = e) P(B = b) P(E = e)$$

Computational cost:

Number of additions: **15**

Number of products: 16*4=**64**

Inference in Bayesian networks

Approach 2. Interleave sums and products

- Combines sums and product in a smart way (multiplications by constants can be taken out of the sum)

$$P(J=T)=$$

$$=\sum_{b\in T,F}\sum_{e\in T,F}\sum_{a\in T,F}\sum_{m\in T,F}P(J=T\mid A=a)P(M=m\mid A=a)P(A=a\mid B=b,E=e)P(B=b)P(E=e)$$

$$=\sum_{b\in T,F}\sum_{a\in T,F}\sum_{m\in T,F}P(J=T\mid A=a)P(M=m\mid A=a)P(B=b)[\sum_{e\in T,F}P(A=a\mid B=b,E=e)P(E=e)]$$

$$=\sum_{a\in T,F}P(J=T\mid A=a)[\sum_{m\in T,F}P(M=m\mid A=a)][\sum_{b\in T,F}P(B=b)[\sum_{e\in T,F}P(A=a\mid B=b,E=e)P(E=e)]]$$

Computational cost:

Number of additions: 1+ 2*(1)+2*(1+2*(1))=**9**

Number of products: 2*(2+2*(1)+2*(2*(1)))=**16**

Inference in Bayesian networks

- The smart interleaving of sums and products can help us to speed up the computation of joint probability queries
- What if we want to compute: $P(B=T,J=T)$

$$P(B=T,J=T)=$$

$$=\sum_{a\in T,F}P(J=T\mid A=a)[\sum_{m\in T,F}P(M=m\mid A=a)][P(B=T)[\sum_{e\in T,F}P(A=a\mid B=T,E=e)P(E=e)]$$

$$P(J=T)=$$

$$=\sum_{a\in T,F}P(J=T\mid A=a)[\sum_{m\in T,F}P(M=m\mid A=a)][\sum_{b\in T,F}P(B=b)[\sum_{e\in T,F}P(A=a\mid B=b,E=e)P(E=e)]]$$

- A lot of shared computation
  - Smart cashing of results can save the time for more queries

Inference in Bayesian networks

- The smart interleaving of sums and products can help us to speed up the computation of joint probability queries
- What if we want to compute: $P(B=T, J=T)$

$$P(B=T, J=T) =$$
$$= \sum_{a\in T,F} P(J=T \mid A=a)\boxed{[\sum_{m\in T,F} P(M=m \mid A=a)]}\boxed{[P(B=T)[\sum_{e\in T,F} P(A=a \mid B=T, E=e)P(E=e)]]}$$

$$P(J=T) =$$
$$= \sum_{a\in T,F} P(J=T \mid A=a)\boxed{[\sum_{m\in T,F} P(M=m \mid A=a)]}[\sum_{b\in T,F} \boxed{P(B=b)[\sum_{e\in T,F} P(A=a \mid B=b, E=e)P(E=e)]]}$$

- A lot of shared computation
  - Smart cashing of results can save the time if more queries

Inference in Bayesian networks

- When cashing of results becomes handy?
- What if we want to compute a diagnostic query:

$$P(B=T \mid J=T) = \frac{P(B=T, J=T)}{P(J=T)}$$

- Exactly probabilities we have just compared !!
- There are other queries when cashing and ordering of sums and products can be shared and saves computation

$$\mathbf{P}(B \mid J=T) = \frac{\mathbf{P}(B, J=T)}{P(J=T)} = \alpha \mathbf{P}(B, J=T)$$

- **General technique: Variable elimination**

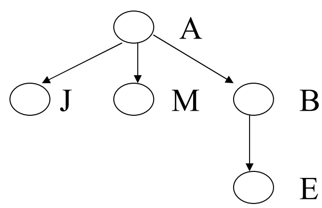
Inference in Bayesian networks

- General idea of variable elimination

$$P(True) = 1 =$$

$$= \sum_{a\in T,F} \underbrace{[\sum_{j\in T,F} P(J=j \mid A=a)]}_{f_J(a)} \underbrace{[\sum_{m\in T,F} P(M=m \mid A=a)]}_{f_M(a)} \underbrace{[\sum_{b\in T,F} P(B=b) \underbrace{[\sum_{e\in T,F} P(A=a \mid B=b,E=e)P(E=e)]]}_{f_E(a,b)}}_{f_B(a)}$$

Variable order:



Results cashed in the tree structure

Inferences in Bayesian network

- **Exact inference algorithms:**
  - Symbolic inference (D'Ambrosio)
  - Message passing algorithm (Pearl)
  - Clustering and joint tree approach (Lauritzen, Spiegelhalter)
  - Arc reversal (Olmsted, Schachter)

- **Approximate inference algorithms:**
  - Monte Carlo methods:
    - Forward sampling, Likelihood sampling
  - Variational methods