CS 1675 Introduction to Machine Learning
Lecture 15

Bayesian belief networks (learning and inference)

Milos Hauskrecht
milos@pitt.edu
5329 Sennott Square

Midterm exam

Midterm Thursday, March 2, 2017

- **in-class (75 minutes)**
- **closed book**

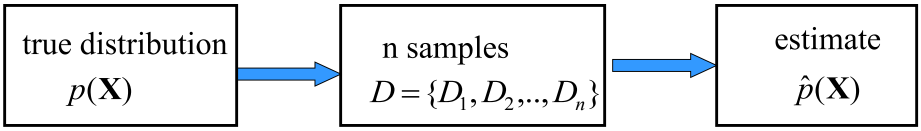
- No programing questions
- Know/understand well what is in lecture notes and assignments
- Know key ideas/principle behind different models, learning algorithms
- Know principles for ML and Error function optimization
- Know how to differentiate simple functions
- True/false questions will require justification

Density estimation

Data: $D = \{D_1, D_2, .., D_n\}$

$D_i = \mathbf{x}_i$ a vector of attribute values

Objective: try to estimate the underlying true probability distribution over variables $\mathbf{X}$, $p(\mathbf{X})$, using examples in D



true distribution $p(\mathbf{X})$

n samples $D = \{D_1, D_2, .., D_n\}$

estimate $\hat{p}(\mathbf{X})$

Standard (iid) assumptions: Samples

- **are independent of each other**
- **come from the same (identical) distribution (fixed $p(\mathbf{X})$)**

Modeling complex distributions

Question: How to model and learn complex multivariate distributions $\hat{p}(\mathbf{X})$ with a large number of variables?

Example: modeling of disease – symptoms relations

- **Disease:** pneumonia
- **Patient symptoms (findings, lab tests)**:
  - Fever, Cough, Paleness, WBC (white blood cells) count, Chest pain, etc.
- **Model of the full joint distribution**:
  P(Pneumonia, Fever, Cough, Paleness, WBC, Chest pain)

One probability per assignment of values to variables:
P(Pneumonia =T, Fever =T, Cought=T, WBC=High, Chest pain=T)

Bayesian belief networks (BBNs)

Bayesian belief networks (late 80s, beginning of 90s)

- Give solutions to the space, acquisition bottlenecks
- Partial solutions for time complexities

Key features:

- 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
- **X and Y are independent** $P(X,Y) = P(X)P(Y)$
- **X and Y are conditionally independent given Z**

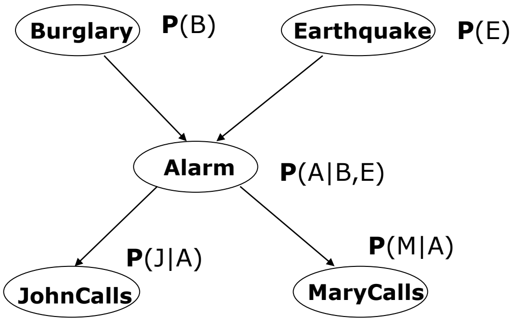
$$P(X,Y \mid Z) = P(X \mid Z)P(Y \mid Z)$$

$$P(X \mid Y,Z) = P(X \mid Z)$$

Bayesian belief network

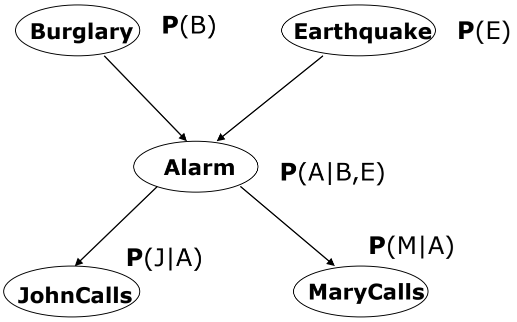
1. **Directed acyclic graph**
   - **Nodes** = random variables
     Burglary, Earthquake, Alarm, Mary calls and John calls
   - **Links** = direct (causal) dependencies between variables.

     The chance of Alarm being is influenced by Earthquake,
     The chance of John calling is affected by the Alarm

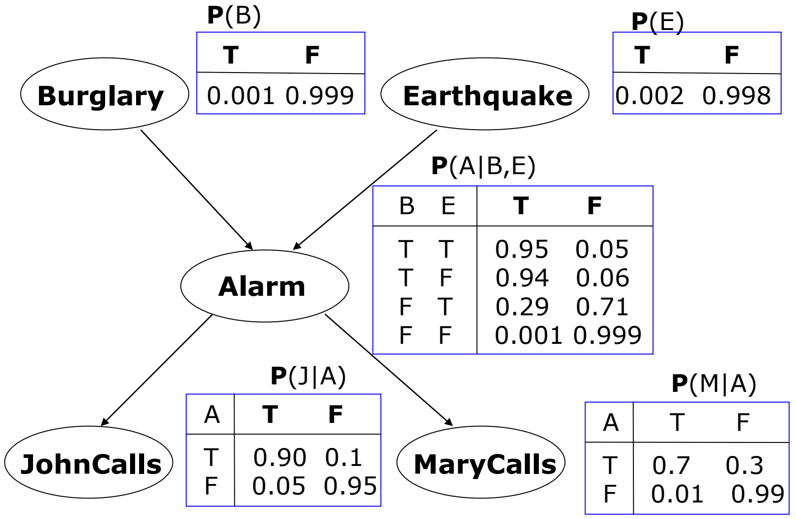


Bayesian belief network

2. **Local conditional distributions**
   - relating 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

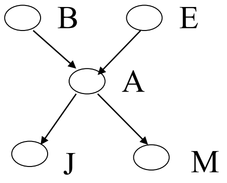
P(M|A)

A	T	F
T	0.7	0.3
F	0.01	0.99

Full joint distribution in BBNs

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

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



Example:

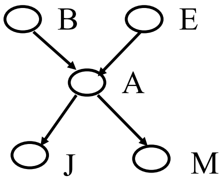
Assume the following assignment of values to random variables

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

Then its probability is:

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

Full joint distribution in BBNs



Rewrite the full joint probability using the product rule:

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

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

Parameter complexity problem

- In the BBN the **full joint distribution** is defined as:

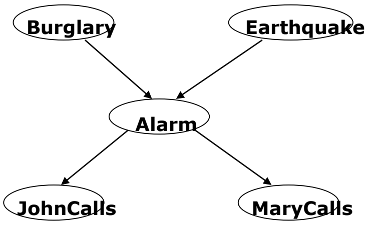
$$\mathbf{P}(X_1, X_2, .., X_n) = \prod_{i=1,..n} \mathbf{P}(X_i \mid pa(X_i))$$

- **What did we save?**

Alarm example: binary (True, False) variables

# of parameters of the full joint:

?



Parameter complexity problem

- In the BBN the **full joint distribution** is defined as:

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

- **What did we save?**

Alarm example: binary (True, False) variables

# of parameters of the full joint:

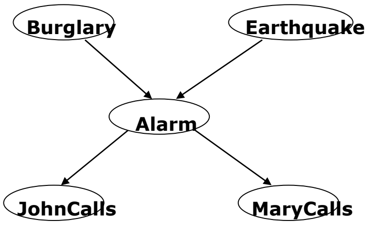
$$2^5 = 32$$

One parameter is for free:

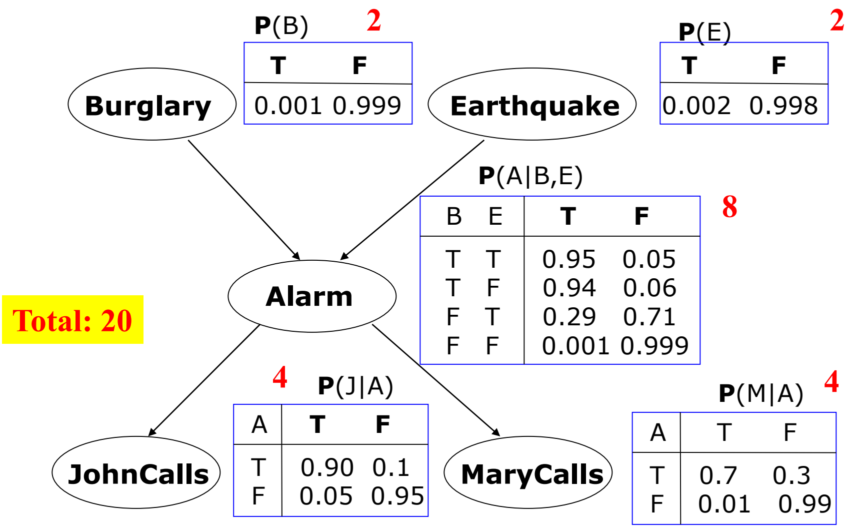
$$2^5 - 1 = 31$$

# of parameters of the BBN:

?



Bayesian belief network: parameters count



P(B) 2

T	**F**
0.001	0.999

P(E) 2

T	**F**
0.002	0.998

P(A|B,E) 8

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

4 **P**(J|A)

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

P(M|A) 4

A	T	F
T	0.7	0.3
F	0.01	0.99

Total: 20

Parameter complexity problem

- In the BBN the **full joint distribution** is defined as:

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

- **What did we save?**

Alarm example: 5 binary (True, False) variables

# of parameters of the full joint:

$$2^5 = 32$$

One parameter is for free:

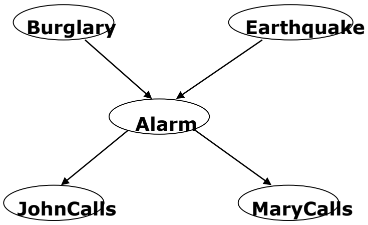
$$2^5 - 1 = 31$$

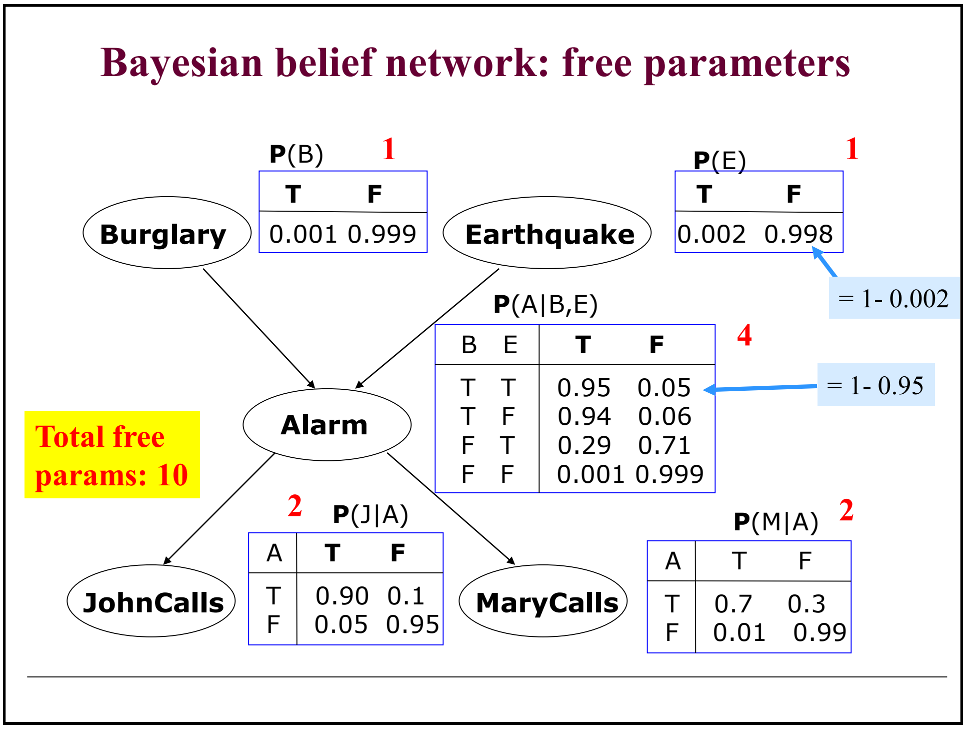
# of parameters of the BBN:

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

One parameter in every conditional is for free:

?





Bayesian belief network: free parameters

Burglary

Earthquake

Alarm

JohnCalls

MaryCalls

P(B) 1

T	**F**
0.001	0.999

P(E) 1

T	**F**
0.002	0.998

= 1- 0.002

P(A|B,E) 4

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

= 1- 0.95

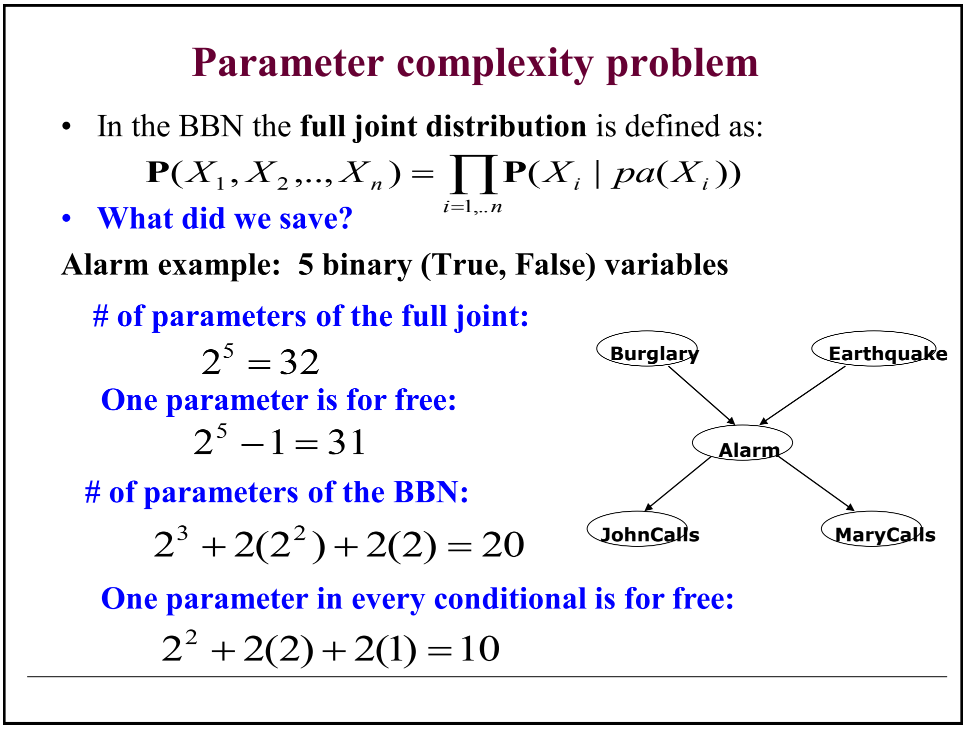
Total free params: 10

2 **P**(J|A)

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

P(M|A) 2

A	T	F
T	0.7	0.3
F	0.01	0.99



Parameter complexity problem

- In the BBN the **full joint distribution** is defined as:

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

- **What did we save?**

Alarm example: 5 binary (True, False) variables

# of parameters of the full joint:

$$2^5 = 32$$

One parameter is for free:

$$2^5 - 1 = 31$$

# of parameters of the BBN:

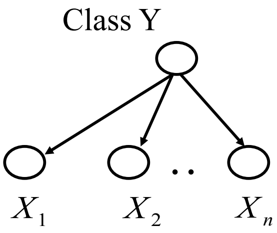
$$2^3 + 2(2^2) + 2(2) = 20$$

One parameter in every conditional is for free:

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

Burglary
Earthquake
Alarm
JohnCalls
MaryCalls

Naïve Bayes model



A **special (simple) Bayesian belief network**

- **used as a generative classifier model**
  - Class variable Y
  - Attributes are independent given Y

$$p(\mathbf{x} \mid Y = i, \mathbf{\Theta}) = \prod_{j=1}^{n} p(x_j \mid Y = i, \Theta_{ij})$$

Learning: ML, Bayesian estimates of parameters

Classification: given *x* we need to determine the class

- Choose the class with the maximum posterior

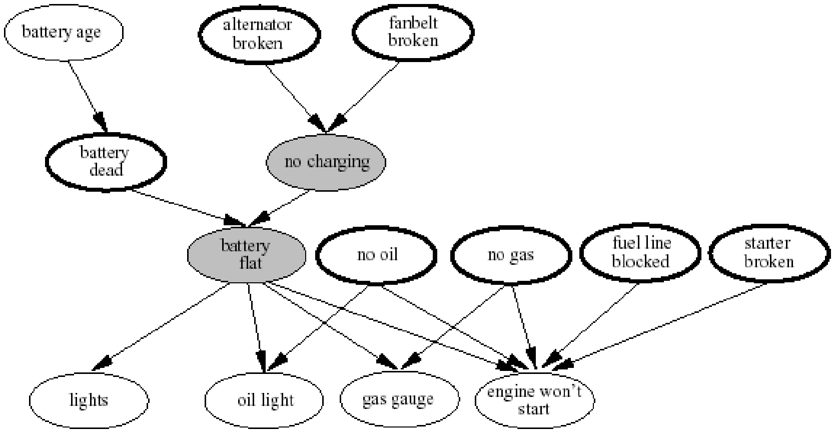
$$p(Y = i \mid \mathbf{x}, \mathbf{\Theta}) = \frac{p(Y = i \mid \mathbf{\Theta}) p(\mathbf{x} \mid Y = i, \mathbf{\Theta})}{\sum_{j=1}^{k} p(Y = j \mid \mathbf{\Theta}) p(\mathbf{x} \mid Y = j, \mathbf{\Theta})}$$

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

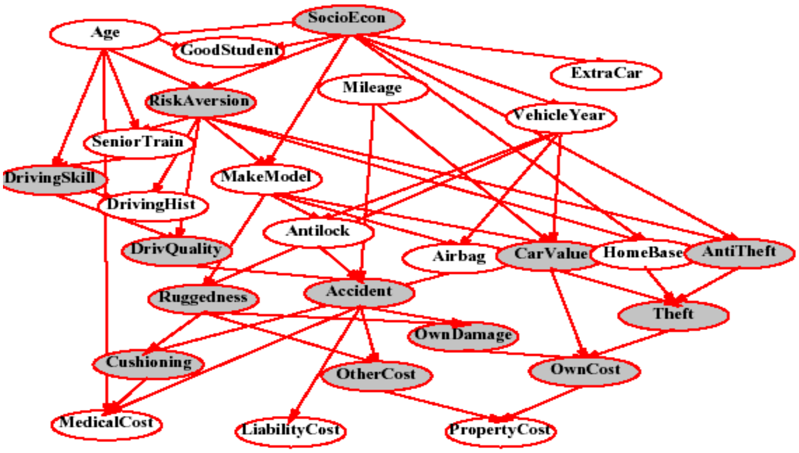
Diagnosis of car engine

- Diagnose the engine start problem

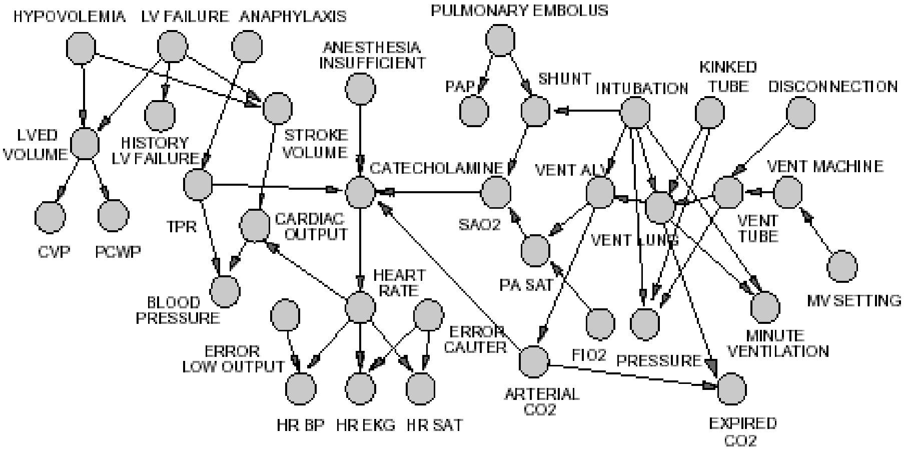


Car insurance example

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

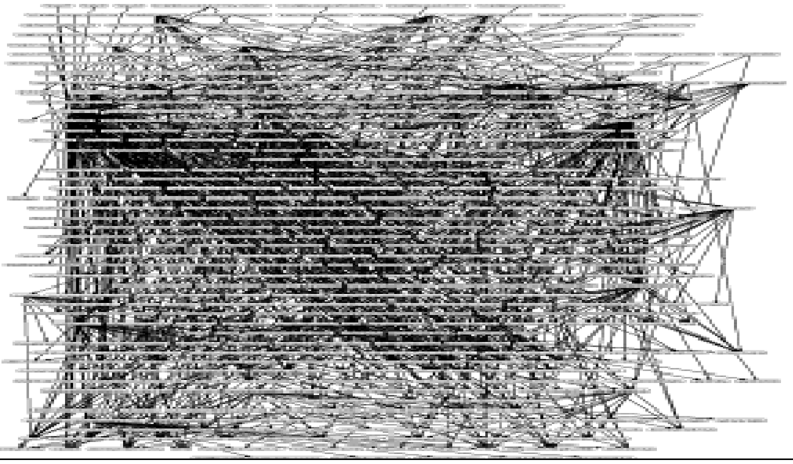


(ICU) Alarm network



CPCS

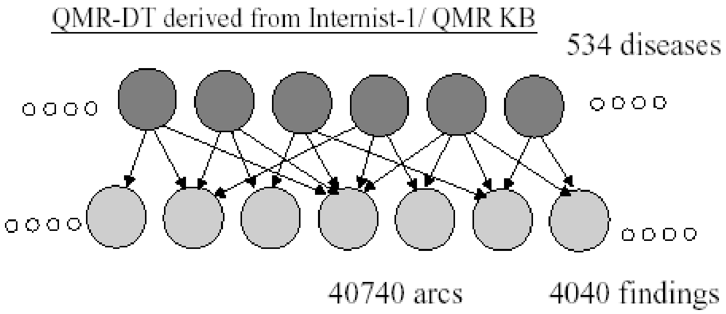
- **C**omputer-based **P**atient **C**ase **S**imulation system (CPCS-PM) developed by Parker and Miller (at University of Pittsburgh)
- 422 nodes and 867 arcs



QMR-DT

- **Medical diagnosis in internal medicine**

Bipartite network of disease/findings relations



Learning of BBN

Learning.

- **Learning of parameters of conditional probabilities**
- **Learning of the network structure**

Variables:

- **Observable** – values present in every data sample
- **Hidden** – they values are never observed in data
- **Missing values** – values sometimes present, sometimes not

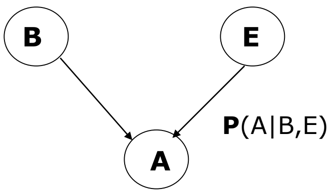
Next:

- Learning of the parameters of BBN
- Values for all variables are observable

Estimation of parameters of BBN

- **Idea:** decompose the estimation problem for the full joint over a large number of variables to a set of smaller estimation problems corresponding to local parent-variable conditionals.
- **Example:** Assume A,E,B are binary with *True, False* values

Learning of P(A|B,E) = 4 estimation problems



P(A|B=T,E=T)
P(A|B=T,E=F)
P(A|B=F,E=T)
P(A|B=F,E=F)

- **Assumption that enables the decomposition:** parameters of conditional distributions are independent

Estimates of parameters of BBN

- Two assumptions that permit the decomposition:
  - **Sample independence**

$$P(D \mid \boldsymbol{\Theta},\xi)=\prod_{u=1}^{N}P(D_u \mid \boldsymbol{\Theta},\xi)$$

  - **Parameter independence**

# of nodes

# of parents' values

$$p(\boldsymbol{\Theta} \mid D,\xi)=\prod_{i=1}^{n}\prod_{j=1}^{q_i}p(\theta_{ij} \mid D,\xi)$$

Parameters of **each conditional** (one for every assignment of values to parent variables) can be learned independently

Learning of BBN parameters. Example.

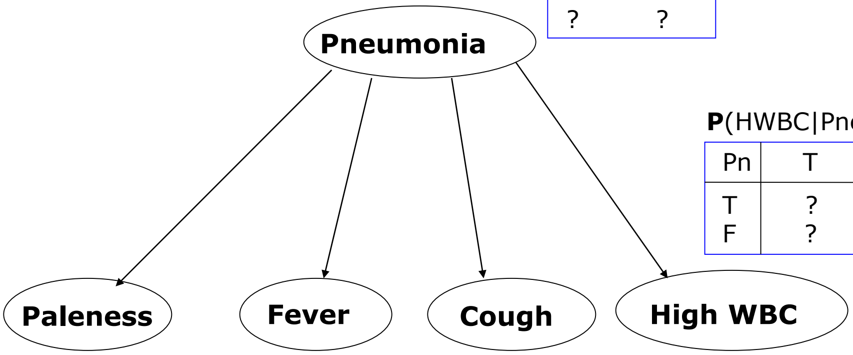
Example:

P(Pneumonia)

T	F
?	?

P(HWBC|Pneum)

Pn	T	F
T	?	?
F	?	?



P(Palen|Pneum) ?

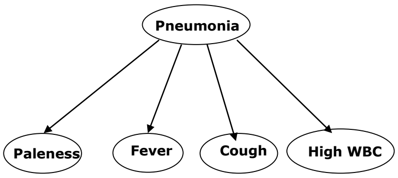
P(Fever|Pneum) ?

P(Cough|Pneum) ?

Learning of BBN parameters. Example.

Data D (different patient cases):

Pal	Fev	Cou	HWB	Pneu
T	T	T	T	F
T	F	F	F	F
F	F	T	T	T
F	F	T	F	T
F	T	T	T	T
T	F	T	F	F
F	F	F	F	F
T	T	F	F	F
T	T	T	T	T
F	T	F	T	T
T	F	F	T	F
F	T	F	F	F



Estimates of parameters of BBN

- Much like multiple **coin toss or roll of a dice** problems.
- A “smaller” learning problem corresponds to the learning of exactly one conditional distribution
- **Example:**

$$\mathbf{P}(Fever \mid Pneumonia = T)$$

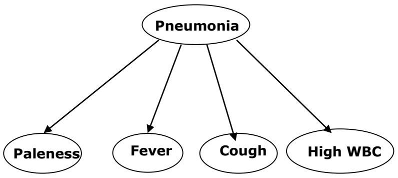
- **Problem:** How to pick the data to learn?

Learning of BBN parameters. Example.

Learn: $\mathbf{P}(Fever \mid Pneumonia = T)$

Step 1: Select data points with Pneumonia=T

Pal	Fev	Cou	HWB	Pneu
T	T	T	T	F
T	F	F	F	F
F	F	T	T	T
F	F	T	F	T
F	T	T	T	T
T	F	T	F	F
F	F	F	F	F
T	T	F	F	F
T	T	T	T	T
F	T	F	T	T
T	F	F	T	F
F	T	F	F	F

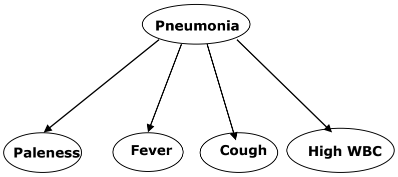


Learning of BBN parameters. Example.

Learn: $\mathbf{P}(Fever \mid Pneumonia = T)$

Step 1: Ignore the rest

Pal	Fev	Cou	HWB	Pneu
F	F	T	T	T
F	F	T	F	T
F	T	T	T	T
T	T	T	T	T
F	T	F	T	T

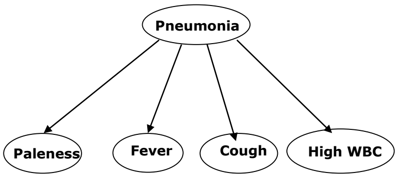


Learning of BBN parameters. Example.

Learn: $\mathbf{P}(Fever \mid Pneumonia = T)$

Step 2: Select values of the random variable defining the distribution of Fever

Pal	Fev	Cou	HWB	Pneu
F	F	T	T	T
F	F	T	F	T
F	T	T	T	T
T	T	T	T	T
F	T	F	T	T

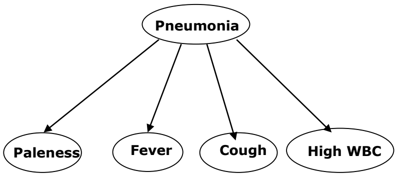


Learning of BBN parameters. Example.

Learn: $\mathbf{P}(Fever \mid Pneumonia = T)$

Step 2: Ignore the rest

Fev
F
F
T
T
T

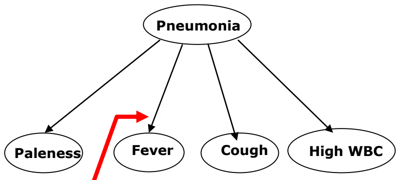


Learning of BBN parameters. Example.

Learn: $\mathbf{P}(Fever \mid Pneumonia = T)$

Step 3a: **Learning the ML estimate**

Fev
F
F
T
T
T



$\mathbf{P}(Fever \mid Pneumonia = T)$

T	F
0.6	0.4

Learning of BBN parameters. Bayesian learning.

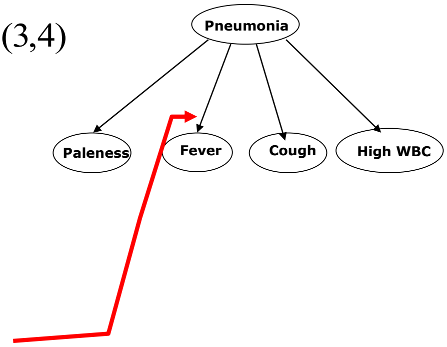
Learn: $\mathbf{P}(Fever \mid Pneumonia = T)$

Step 3b: Learning the Bayesian estimate

Assume the prior

$$\theta_{Fever|Pneumonia=T} \sim Beta(3,4)$$

Fev
F
F
T
T
T



Posterior:

$$\theta_{Fever|Pneumonia=T} \sim Beta(6,6)$$

Estimates of parameters of BBN

Much like multiple **coin toss or roll of a dice** problems.

- A "smaller" learning problem corresponds to the learning of exactly one conditional distribution

Example:

$$\mathbf{P}(Fever \mid Pneumonia = T)$$

Problem: How to pick the data to learn?

Answer:

1. Select data points with Pneumonia=T (ignore the rest)
2. Focus on (select) only values of the random variable defining the distribution (Fever)
3. Learn the parameters of the conditional the same way as we learned the parameters of the biased coin or dice

Probabilistic inferences

- BBN models compactly the full joint distribution by taking advantage of existing independences between variables
- Simplifies the representation and learning of a model

- Can be used for the different **inference tasks** ….

Bayes theorem

Conditional/joint probability relations.

$$P(A\mid B)=\frac{\boxed{P(A,B)}}{P(B)} \longrightarrow P(A,B)=P(B\mid A)P(A)$$

Bayes theorem (switches conditioning events) :

$$P(A\mid B)=\frac{P(B\mid A)P(A)}{P(B)}$$

When is it useful?

- When we are interested in computing the diagnostic query from the causal probability

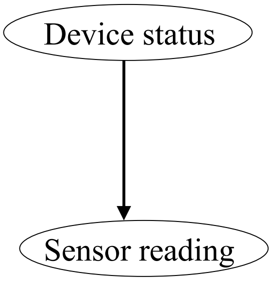
$$P(cause\mid effect)=\frac{P(effect\mid cause)P(cause)}{P(effect)}$$

- **Reason:** It is often easier to assess causal probability
  - E.g. Probability of pneumonia causing fever
    vs. probability of pneumonia given fever

Example: a simple diagnostic inference

- **Device** (equipment) operating *normally* or *malfunctioning*.
  - Operation of the device sensed indirectly via a sensor
- **Sensor reading** is either *High* or *Low*

BBN



P(Device status)

normal	**malfunctioning**
0.9	0.1

P(Sensor reading| Device status)

Status\Sensor	**High**	**Low**
normal	0.1	0.9
malfunc	0.6	0.4

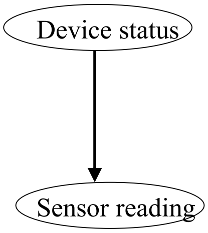
Example: a simple diagnostic inference

- **Diagnostic inference:** compute the probability of device operating normally or malfunctioning **given a sensor reading**

$$\mathbf{P}(\text{Device status} \mid \text{Sensor reading} = high) =$$

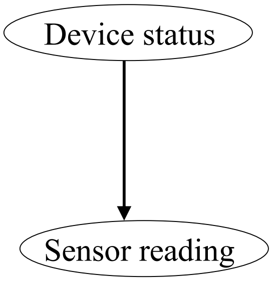
$$= \begin{pmatrix} P(\text{Device status} = normal \mid \text{Sensor reading} = high) \\ P(\text{Device status} = malfunct \mid \text{Sensor reading} = high) \end{pmatrix}$$

- Note we have the opposite conditional probabilities: they are much easier to estimate
- **Solution:** apply **Bayes theorem** to reverse the conditioning variables



Example: a simple diagnostic inference

- **Device** (equipment) operating *normally* or *malfunctioning*.
  - Operation of the device sensed indirectly via a sensor
- **Sensor reading** is either *High* or *Low*



Device status

Sensor reading

P(Device status)

normal	**malfunctioning**
0.9	0.1

P(Sensor reading| Device status)

Status\Sensor	**High**	**Low**
normal	0.1	0.9
malfunc	0.6	0.4

$$\mathbf{P}(\text{Device status} \mid \text{Sensor reading} = high) = ?$$

Bayes theorem

Assume a variable A with multiple values $a_1, a_2, \ldots a_k$

Bayes theorem can be rewritten as:

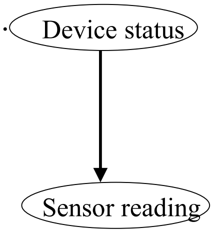
$$P(A = a_j \mid B = b) = \frac{P(B = b \mid A = a_j) P(A = a_j)}{P(B = b)}$$

$$= \frac{P(B = b \mid A = a_j) P(A = a_j)}{\sum_{j=1}^{k} P(B = b \mid A = a_j) P(A = a_j)}$$

Used in practice when we want to compute:

$\mathbf{P}(A \mid B = b)$ for all values of $a_1, a_2, \ldots a_k$

Example: a simple diagnostic inference



P(Device status)

normal	malfunctioning
0.9	0.1

P(Sensor reading| Device status)

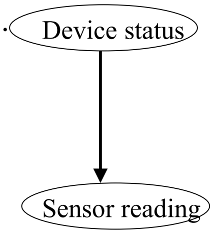
Status\Sensor	High	Low
normal	0.1	0.9
malfunc	0.6	0.4

$$\mathbf{P}(\text{Device status} \mid \text{Sensor reading} = high) = ?$$

$$P(\text{Device status} = norm \mid \text{Sensor reading} = high) =$$

$$= \frac{P(\text{Sensor reading} = high, \text{Device status} = norm)}{P(\text{Sensor reading} = high)}$$

Example: a simple diagnostic inference



P(Device status)

normal	malfunctioning
0.9	0.1

P(Sensor reading| Device status)

Status\Sensor	High	Low
normal	0.1	0.9
malfunc	0.6	0.4

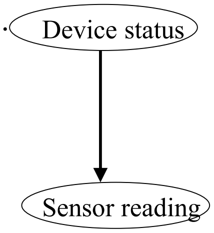
$$\mathbf{P}(\text{Device status} \mid \text{Sensor reading} = high) = ?$$

$$P(\text{Device status} = norm \mid \text{Sensor reading} = high) =$$

$$= \frac{P(\text{Sensor reading} = high, \text{Device status} = norm)}{P(\text{Sensor reading} = high)}$$

$$= \frac{P(\text{Sensor reading} = high \mid \text{Device status} = norm) P(\text{Device status} = norm)}{P(\text{Sensor reading} = high)}$$

Example: a simple diagnostic inference



P(Device status)

normal	malfunctioning
0.9	0.1

P(Sensor reading| Device status)

Status\Sensor	High	Low
normal	0.1	0.9
malfunc	0.6	0.4

$$\mathbf{P}(\text{Device status} \mid \text{Sensor reading} = high) = ?$$

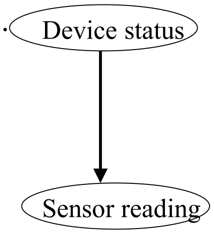
$$P(\text{Device status} = norm \mid \text{Sensor reading} = high) =$$

$$= \frac{P(\text{Sensor reading} = high \mid \text{Device status} = norm)P(\text{Device status} = norm)}{\boxed{P(\text{Sensor reading} = high)}}$$

$$\downarrow$$

$$\boxed{\begin{aligned} &P(\text{Sensor reading} = high, \text{Device status} = norm) \\ &+ P(\text{Sensor reading} = high, \text{Device status} = malf) \end{aligned}}$$

Example: a simple diagnostic inference



P(Device status)

normal	malfunctioning
0.9	0.1

P(Sensor reading| Device status)

Status\Sensor	High	Low
normal	0.1	0.9
malfunc	0.6	0.4

$$\mathbf{P}(\text{Device status} \mid \text{Sensor reading} = high) = ?$$

$$P(\text{Device status} = norm \mid \text{Sensor reading} = high) =$$

$$= \frac{P(\text{Sensor reading} = high \mid \text{Device status} = norm)P(\text{Device status} = norm)}{\boxed{P(\text{Sensor reading} = high)}}$$

$$\downarrow$$

$$\boxed{\begin{aligned} &P(\text{Sensor reading} = high \mid \text{Device status} = norm)P(\text{Device status} = norm) \\ &+ P(\text{Sensor reading} = high \mid \text{Device status} = malf)\ P(\text{Device status} = malf) \end{aligned}}$$

Inference in Bayesian networks

- BBN models compactly the full joint distribution by taking advantage of existing independences between variables
- Simplifies the representation and learning of a model
- 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)**

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

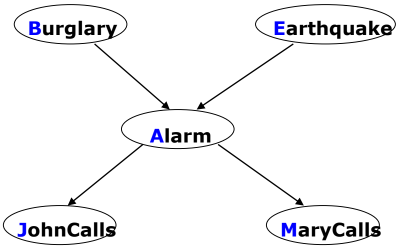
  - **Other probabilistic queries** **(**queries on joint distributions).

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

- **Main question:** Can we take advantage of independences to construct special algorithms and speeding up 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: ?

Number of products: ?

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: ?

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+1+2*1]=**?**

Number of products: 2*[2+2*(1+2*1)]=?

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+1+2*1]=**9**

Number of products: 2*[2+2*(1+2*1)]=?

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+1+2*1]=**9**

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