Lecture 21: Bayesian Networks

Andreas Wichert
Department of Computer Science and Engineering
Técnico Lisboa

Joint distribution

The joint distribution for n possible variables is described by 2^n possible combinations. The probability distribution $d_1 \times d_2 \times \cdots \times d_n$ corresponds to a vector of length 2^n. For a joint distribution of n possible variables, the exponential growth of combinations being true or false becomes an intractable problem for large n. For

$$P(h_i|d_1, d_2, d_3, .., d_n) = \frac{P(d_1, d_2, d_3, ..., d_n|h_i) \cdot P(h_i)}{P(d_1, d_2, d_3, ..., d_n)}$$

all $2^n - 1$ possible combinations must be known. There are two possible solutions to this problem.

The first solution is the decomposition of large probabilistic domains into weakly connected subsets via conditional independence,

$$P(d_1, d_2, d_3, ..., d_n|h_i) = \prod_{j=1}^{n} P(d_j|h_i).$$

This approach is known as the Naïve Bayes assumption and is one of the most important developments in the recent history of Artificial Intelligence. It assumes that a single cause directly influences a number of events, all of which are conditionally independent,

$$h_{map} = arg\max_{h_i} \prod_{j=1}^{n} P(d_j|h_i) \cdot P(h_i).$$

However, this conditional independence is very restrictive. Often, it is not present in real life events. Dependence between some events is always present.

Bayesian networks represent the second and more realistic solution. Bayesian networks can describe a probability distribution of a set of variables by combining conditional independence assumptions with conditional probabilities. Unlike the Naïve Bayes assumption, which states that all of the variables are conditionally independent given the value of the target variable, Bayesian networks enable these conditional independence assumptions to be applied to subsets of variables, providing a model with fewer constraints than the Bayes assumption.

Naive Bayes Classifier

- Along with decision trees, neural networks, nearest neighbor, one of the most practical learning methods

- When to use:
  - Moderate or large training set available
  - Attributes that describe instances are conditionally independent given classification
- Successful applications:
  - Diagnosis
  - Classifying text documents

Naive Bayes Classifier

- Assume target function *f: X* ➔ *V*, where each instance *x* described by attributes $a_1, a_2 .. a_n$
- Most probable value of *f(x)* is:

$$v_{MAP} = \arg\max_{v_j \in V} P(v_j | a_1, a_2 \ldots a_n)$$

$$v_{MAP} = \arg\max_{v_j \in V} \frac{P(a_1, a_2 \ldots a_n | v_j) P(v_j)}{P(a_1, a_2 \ldots a_n)}$$

$$= \arg\max_{v_j \in V} P(a_1, a_2 \ldots a_n | v_j) P(v_j)$$

V_{NB}

- Naive Bayes assumption:

$$P(a_1, a_2 \ldots a_n | v_j) = \prod_i P(a_i | v_j)$$

- which gives

$$\text{Naive Bayes classifier: } v_{NB} = \arg\max_{v_j \in V} P(v_j) \prod_i P(a_i | v_j)$$

Naive Bayes Algorithm

- For each target value v_j
- $\hat{P}(v_j)$ ← estimate $P(v_j)$
- For each attribute value a_i of each attribute a
- $\hat{P}(a_i|v_j)$ ← estimate $P(a_i|v_j)$

$$v_{NB} = \arg\max_{v_j \in V} \hat{P}(v_j) \prod_{a_i \in x} \hat{P}(a_i|v_j)$$

Training dataset

Class:
C1:buys_computer='yes'
C2:buys_computer='no'

Data sample:

X =
(age<=30,
Income=medium,
Student=yes
Credit_rating=Fair)

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
30…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

Naïve Bayesian Classifier: Example

- Compute P(X|C_i) for each class

P(age="<30" | buys_computer="yes") = 2/9=0.222
P(age="<30" | buys_computer="no") = 3/5 =0.6
P(income="medium" | buys_computer="yes")= 4/9 =0.444
P(income="medium" | buys_computer="no") = 2/5 = 0.4
P(student="yes" | buys_computer="yes)= 6/9 =0.667
P(student="yes" | buys_computer="no")= 1/5=0.2
P(credit_rating="fair" | buys_computer="yes")=6/9=0.667
P(credit_rating="fair" | buys_computer="no")=2/5=0.4

P(buys_computer=„yes")=9/14
P(buys_computer=„no")=5/14

- X=(age<=30 ,income =medium, student=yes,credit_rating=fair)

P(X|C_i) : P(X|buys_computer="yes")= 0.222 x 0.444 x 0.667 x 0.0.667 =0.044
P(X|buys_computer="no")= 0.6 x 0.4 x 0.2 x 0.4 =0.019

P(X|C_i)*P(C_i) : P(X|buys_computer="yes") * P(buys_computer="yes")=0.028
P(X|buys_computer="no") * P(buys_computer="no")=0.007

- X belongs to class "buys_computer=yes"

Estimating Probabilities

- We have estimated probabilities by the fraction of times the event is observed to n_c occur over the total number of opportunities n
- It provides poor estimates when n_c is very small

- If none of the training instances with target value v_j have attribute value a_i?
  - n_c is 0

- When n_c is very small:

$$\hat{P}(a_i|v_j) = \frac{n_c + mp}{n + m}$$

- n is number of training examples for which $v=v_j$
- n_c number of examples for which $v=v_j$ and $a=a_i$
- p is **prior** estimate
- m is weight given to prior (i.e. number of ``virtual'' examples)

$$v_{NB} =_{v_j \in V} P(v_j) \prod_i \hat{P}(a_i|v_j)$$

- Naive Bayes assumption of conditional independence too restrictive
- But it's intractable without some such assumptions...

- Bayesian Belief networks describe conditional independence among **subsets** of variables
- allows combining prior knowledge about (in)dependencies among variables with observed training data

Law of Total Probability

For uncertain events we can list all the logical possibilities. These are called the elementary events or states. For binary events there are two states true and false, for any event a there is an event $\neg a$, the event that a does not occur. Binary events are described by binary variables.

For binary events

$$p(x) + p(\neg x) = 1, \quad p(y) + p(\neg y) = 1$$

the law of total probability is represented by

$$p(y) = p(y, x) + p(y, \neg x) = p(y|x) \cdot p(x) + p(y|\neg x) \cdot p(\neg x)$$

and

$$p(\neg y) = p(\neg y, x) + p(\neg y, \neg x) = p(\neg y|x) \cdot p(x) + p(\neg y|\neg x) \cdot p(\neg x).$$

Law of Total Probability

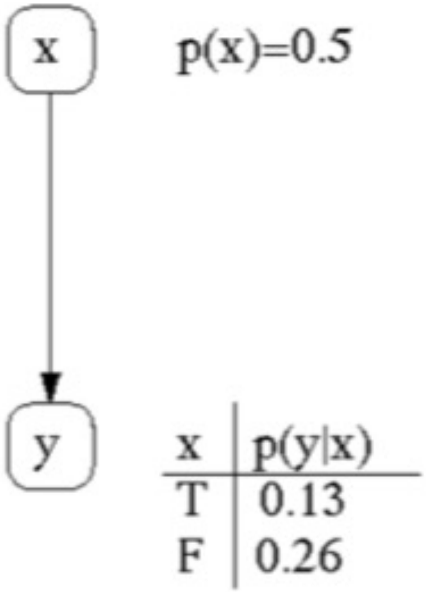


Figure 1.1: The causal relation between events x and y represented by a direct graph of to nodes.

If two events x and y are independent, then the probability that events x and y both occur is

$$p(x, y) = p(x \wedge y) = p(x) \cdot p(y).$$

In this case the conditional probability is

$$p(x|y) = p(x).$$

If all N possible variables are independent, then

$$p(x_1, x_2, \cdots, x_N) = p(x_1) \cdot p(x_2) \cdot \cdots \cdot p(x_N) = \prod_{i=1}^{N} p(x_i)$$

In the case not all variables are independent we can decompose the probabilistic domain into subsets via conditional independence, for M subsets

$$p(x_1, x_2, \cdots, x_N) = \prod_{i=1}^{M} p(x_k, x_{k+1}, \cdots)_i$$

For a subset of dependent variables

$$p(x_1, x_2) = p(x_1|x_2) \cdot p(x_2) = p(x_2|x_1) \cdot p(x_1).$$

This follows from the Bayes's rule

$$p(x_1|x_2) = \frac{p(x_1, x_2)}{p(x_2)} = \frac{p(x_2|x_1) \cdot p(x_1)}{p(x_2)}$$

Two variables x_1 and x_2 are conditionally independent given x_3

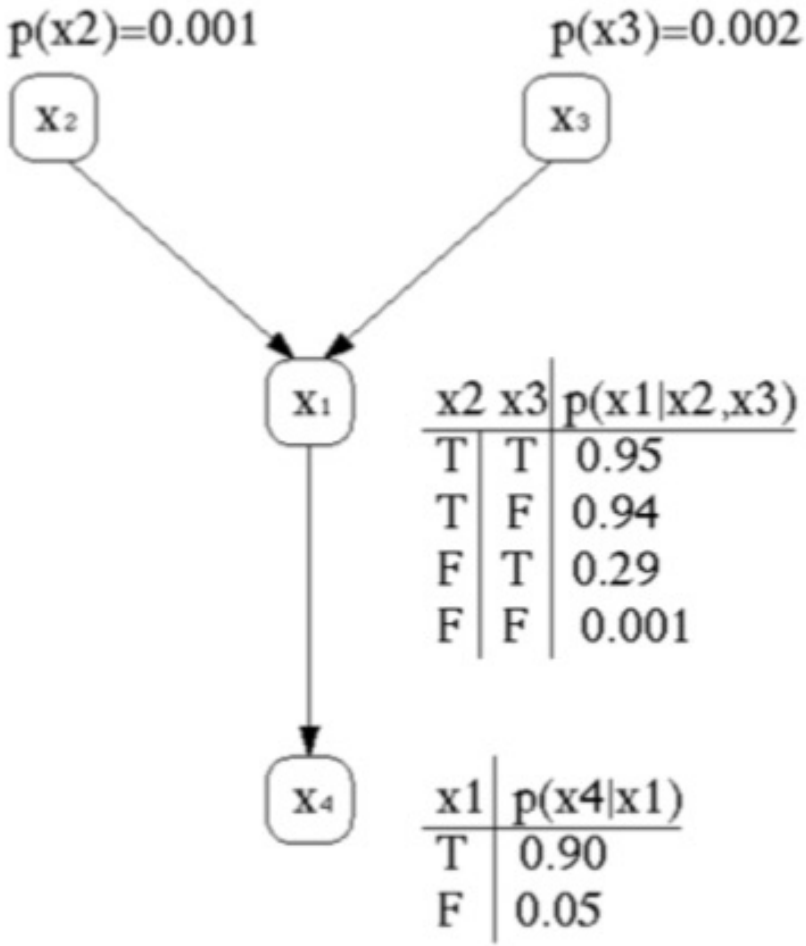
$$p(x_1|x_2, x_3) = p(x_1|x_3).$$

Assuming x_2 and x_3 are independent, but x_1 is conditionally dependent given x_2 and x_3 then

$$p(x_1, x_2, x_3) = p(x_1|x_2, x_3) \cdot p(x_2) \cdot p(x_3).$$

Assuming x_4 is conditionally dependent given x_1 but independent of x_2 and x_3 then

$$p(x_1, x_2, x_3, x_4) = p(x_1|x_2, x_3) \cdot p(x_2) \cdot p(x_3) \cdot p(x_4|x_1).$$



x2	x3	p(x1|x2,x3)
T	T	0.95
T	F	0.94
F	T	0.29
F	F	0.001

x1	p(x4|x1)
T	0.90
F	0.05

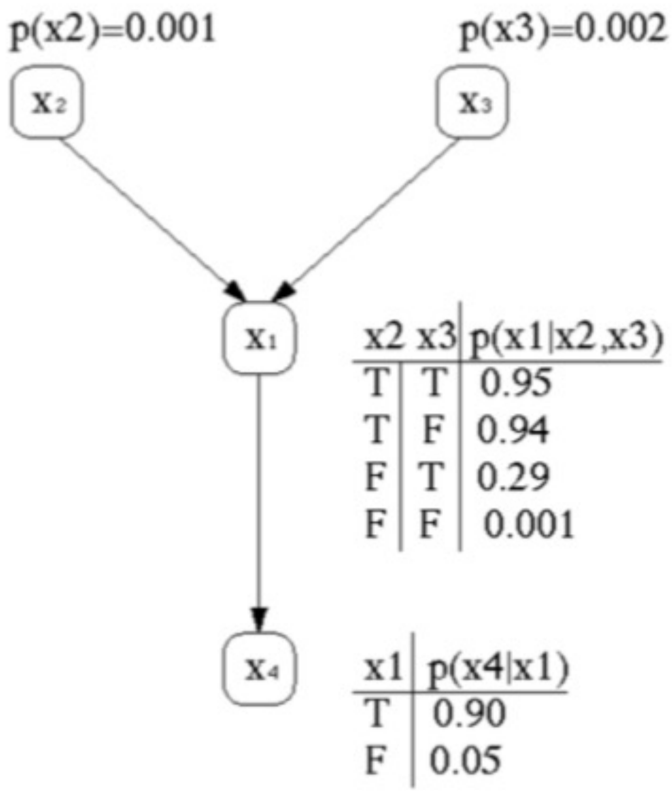
$$p(x_1, x_2, x_3, x_4) = p(x_1|x_2, x_3) \cdot p(x_2) \cdot p(x_3) \cdot p(x_4|x_1).$$

Causality

- This relationship between occurrence of events called causality is represented by conditional dependency inducing *time*.
- In our example x_2 and x_3 cause x_1 and only then x_1 *causes* x_4.
- This kind of decomposition via conditional independence is modelled by Bayesian networks.
- Bayesian networks provide a natural representation for (causally induced) conditional independence.
- They represent a set of conditional independence assumptions, by the topology of an acyclic directed graph and sets of conditional probabilities.

Example

- In the example, there are four variables, namely, Burglary(= x_2), Earthquake(= x_3), Alarm(= x_1) and JohnCalls(= x_4).
- The corresponding network topology reflects the following “causal” knowledge:
    - A burglar can set the alarm off.
    - An earthquake can set the alarm off.
    - The alarm can cause John to call.



- Given the *x* query variable which value has to be determined and *e* evidence variable which is known and the remaining unobservable variables we preform a summation over all possible *y*.

- In the following for simplification the variables are binary and describe binary events. All possible values (true/false) of the unobservable variables y are determined according to the law of total probability

$$p(x|e) = \alpha \sum_y p(x,e,y) = \alpha \cdot (p(x,e,y) + p(x,e,\neg y)).$$

or

$$p(x|e) = \alpha \sum_y p(x,e,y) = \alpha \cdot (p(x,e|y) \cdot p(y) + p(x,e|\neg y) \cdot p(\neg y)).$$

with

$$\alpha = \frac{1}{p(e)} = \frac{1}{\sum_y p(x,e,y) + \sum_y p(\neg x,e,y)}$$

and

$$1 = \alpha \cdot \left(\sum_y p(x,e,y) + \sum_y p(\neg x,e,y) \right).$$

Causality

In a Bayesian network the time line corresponds to the causal relationship between events represented by conditional probabilities. For the preceding example

$$p(x_4|x_1, x_2, x_3) = \alpha \cdot p(x_1|x_2, x_3) \cdot p(x_2) \cdot p(x_3) \cdot p(x_4|x_1).$$

$$p(x_1, x_2, X_3, x_4) = p(x_1, x_2, x_3, x_4) + p(x_1, x_2, \neg x_3, x_4)$$

$$p(x_1, x_2, X_3, x_4) = p(x_2) \cdot p(x_4|x_1) \cdot \left(\sum_{x_3} p(x_1|x_2, x_3) \cdot p(x_3) \right).$$

$$p(x_4|x_1, x_2) = \alpha \cdot p(x_2) \cdot p(x_4|x_1) \cdot \left(\sum_{x_3} p(x_1|x_2, x_3) \cdot p(x_3) \right).$$

$$p(x_4|x_1, x_2) = \alpha \cdot p(x_2) \cdot p(x_4|x_1) \cdot (p(x_1|x_2, x_3) \cdot p(x_3) + p(x_1|x_2, \neg x_3) \cdot p(\neg x_3))$$

with

$$\alpha = \frac{1}{p(x_1, x_2)} = \frac{1}{p(x_1, x_2, X_3, x_4) + p(x_1, x_2, X_3, \neg x_4)}$$

and

$$p(x_1, x_2) = p(x_1, x_2, X_3, x_4) + p(x_1, x_2, X_3, \neg x_4).$$

$$p(x_4|x_1, x_2) = \alpha \cdot p(x_2) \cdot p(x_4|x_1) \cdot (p(x_1|x_2, x_3) \cdot p(x_3) + p(x_1|x_2, \neg x_3) \cdot p(\neg x_3))$$

with

$$\alpha = \frac{1}{p(x_1, x_2)} = \frac{1}{p(x_1, x_2, X_3, x_4) + p(x_1, x_2, X_3, \neg x_4)}$$

and

$$p(x_1, x_2) = p(x_1, x_2, X_3, x_4) + p(x_1, x_2, X_3, \neg x_4).$$

After calculating we arrive at

$$p(x_4|x_1, x_2) = \frac{p(x_4|x_1)}{p(x_4|x_1) + p(\neg x_4|x_1)}.$$

For unknown variables x_3, x_1 indicated by X_1 andX_3 we apply the law of total probability.

$$p(X_1, x_2, X_3, x_4) = p(x_1, x_2, x_3, x_4) + p(\neg x_1, x_2, x_3, x_4) +$$

$$+ p(x_1, x_2, \neg x_3, x_4) + p(\neg x_1, x_2, \neg x_3, x_4)$$

$$p(X_1, x_2, X_3, x_4) = p(x_2) \cdot \sum_{x_4} \left(p(x_4|x_1) \cdot \left(\sum_{x_3} p(x_1|x_2, x_3) \cdot p(x_3) \right) \right).$$

$$p(x_4|x_2) = \alpha \cdot p(x_2) \cdot (p(x_4|x_1) \cdot p(x_1|x_2, x_3) \cdot p(x_3) + p(x_4|x_1) \cdot p(x_1|x_2, \neg x_3) \cdot p(\neg x_3) +$$

$$+ p(x_4|\neg x_1) \cdot p(\neg x_1|x_2, x_3) \cdot p(x_3) + p(x_4|\neg x_1) \cdot p(\neg x_1|x_2, \neg x_3) \cdot p(\neg x_3))$$

with

$$\alpha = \frac{1}{p(x_2)} = \frac{1}{p(X_1, x_2, X_3, x_4) + p(X_1, x_2, X_3, \neg x_4)}.$$

For no present evidence

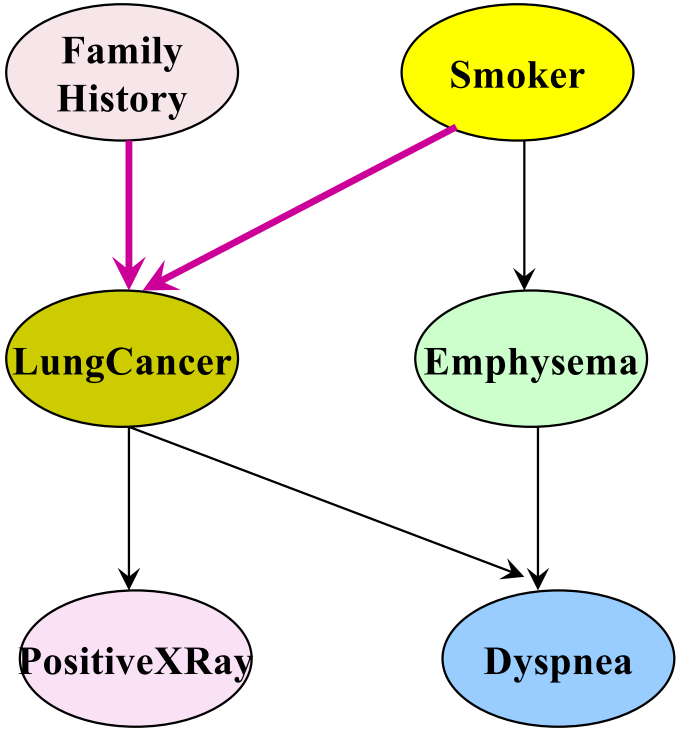
$$p(x_4) = p(X_1, X_2, X_3, x_4) = p(x_1, x_2, x_3, x_4) + p(\neg x_1, x_2, x_3, x_4) +$$

$$+ p(x_1, x_2, \neg x_3, x_4) + p(\neg x_1, x_2, \neg x_3, x_4) +$$

$$+ p(x_1, \neg x_2, x_3, x_4) + p(\neg x_1, \neg x_2, x_3, x_4) +$$

$$+ p(x_1, \neg x_2, \neg x_3, x_4) + p(\neg x_1, \neg x_2, \neg x_3, x_4).$$

Bayesian Belief Network: An Example

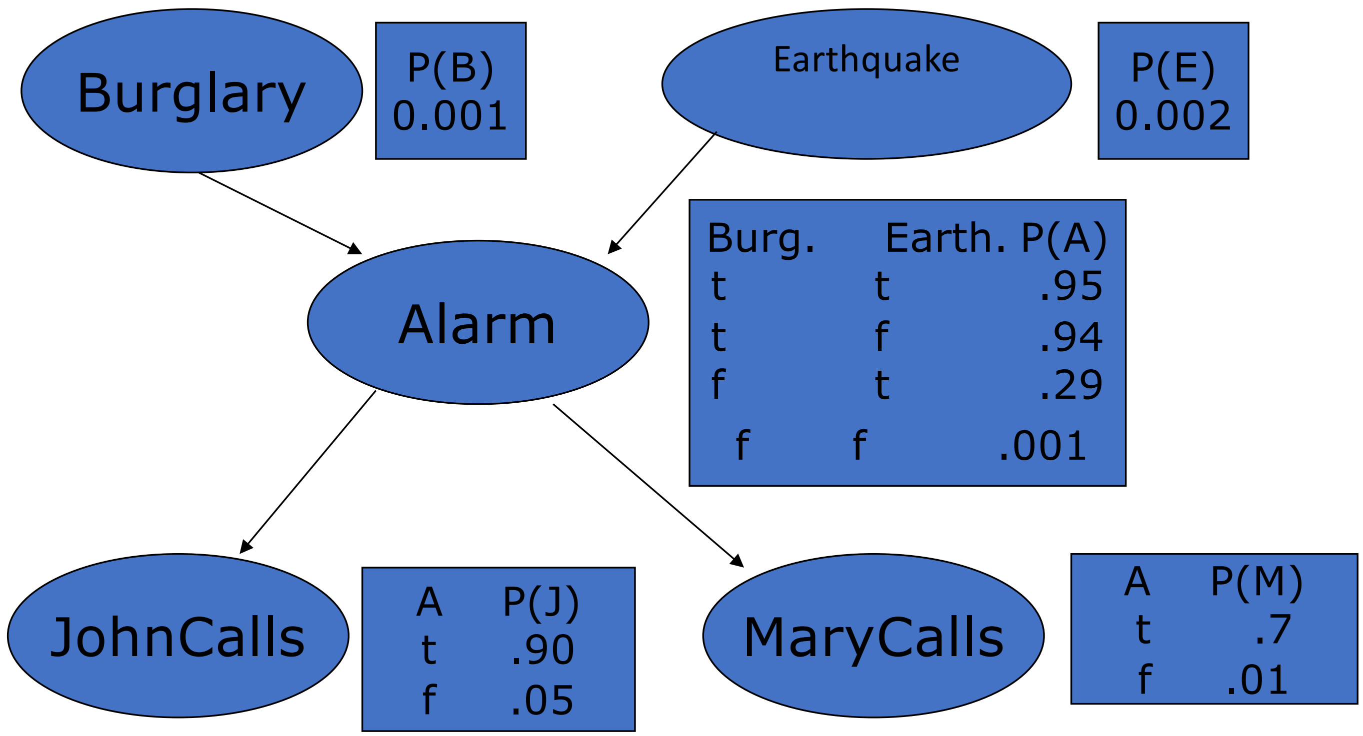


Bayesian Belief Networks

	(FH, S)	(FH, ~S)	(~FH, S)	(~FH, ~S)
LC	0.8	0.5	0.7	0.1
~LC	0.2	0.5	0.3	0.9

The **conditional probability table** for the variable LungCancer: Shows the conditional probability for each possible combination of its parents

Belief Networks



Full Joint Distribution

$$P(x_1,\ldots,x_n) = \prod_{i=1}^{n} P(x_i \mid parents(X_i))$$

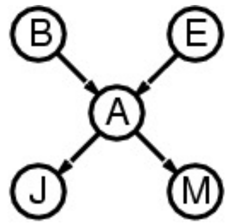
$$P(j \wedge m \wedge a \wedge \neg b \wedge \neg e)$$

$$= P(j \mid a)P(m \mid a)P(a \mid \neg b \wedge \neg e)P(\neg b)P(\neg e)$$

$$= 0.9 \times 0.7 \times 0.001 \times 0.999 \times 0.998 = 0.00062$$

Compactness

- A CPT for Boolean X_i with k Boolean parents has 2^k rows for the combinations of parent values



- Each row requires one number p for $X_i = true$
  (the number for $X_i = false$ is just $1-p$)

- If each variable has no more than k parents, the complete network requires $O(n \cdot 2^k)$ numbers

- I.e., grows linearly with n, vs. $O(2^n)$ for the full joint distribution

- For burglary net, $1 + 1 + 4 + 2 + 2 = 10$ numbers (vs. $2^5-1 = 31$)

Inference in Bayesian Networks

- How can one infer the (probabilities of) values of one or more network variables, given observed values of others?
- Bayes net contains all information needed for this inference
- If only one variable with unknown value, easy to infer it
- In general case, problem is NP hard

Example

- In the burglary network, we migth observe the event in which *JohnCalls=true* and *MarryCalls=true*
- We could ask for the probability that the burglary has occurred
  - *P(Burglary|JohnCalls=ture,MarryCalls=true)*

Normalization

$$1 = P(y \mid x) + P(\neg y \mid x)$$

$$P(Y \mid X) = \alpha \times P(X \mid Y) P(Y)$$

$$\alpha \langle P(y \mid x), P(\neg y \mid x) \rangle$$

$$\alpha \langle 0.12, 0.08 \rangle = \langle 0.6, 0.4 \rangle$$

Normalization

$$P(Cavity \mid toothache) = \alpha P(Cavity, toothache)$$

$$= \alpha[P(Cavity, toothache, catch) + P(Cavity, toothache, \neg catch)]$$

$$= \alpha[< 0.108, 0.016 > + < 0.012, 0.064 >] = \alpha < 0.12, 0.08 > = < 0.6, 0.4 >$$

- X is the query variable
- E evidence variable
- *Y remaining unobservable variable*

$$P(X \mid e) = \alpha P(X, e) = \alpha \sum_{y} P(X, e, y)$$

- Summation over **all possible y** (all possible values of the unobservable variables Y)

- *P(Burglary|JohnCalls=ture,MarryCalls=true)*
  - The hidden variables of the query are *Earthquake* and *Alarm*

$$P(B \mid j,m) = \alpha P(B,j,m) = \alpha \sum_{e} \sum_{a} P(B,e,a,j,m)$$

  - For *Burglary=true* in the Bayesain network

$$P(b \mid j,m) = \alpha \sum_{e} \sum_{a} P(b)P(e)P(a \mid b,e)P(j \mid a)P(m \mid a)$$

- To compute we had to add four terms, each computed by multiplying five numbers
- In the worst case, where we have to sum out almost all variables, the complexity of the network with *n* Boolean variables is $O(n2^n)$

Variable Elimination

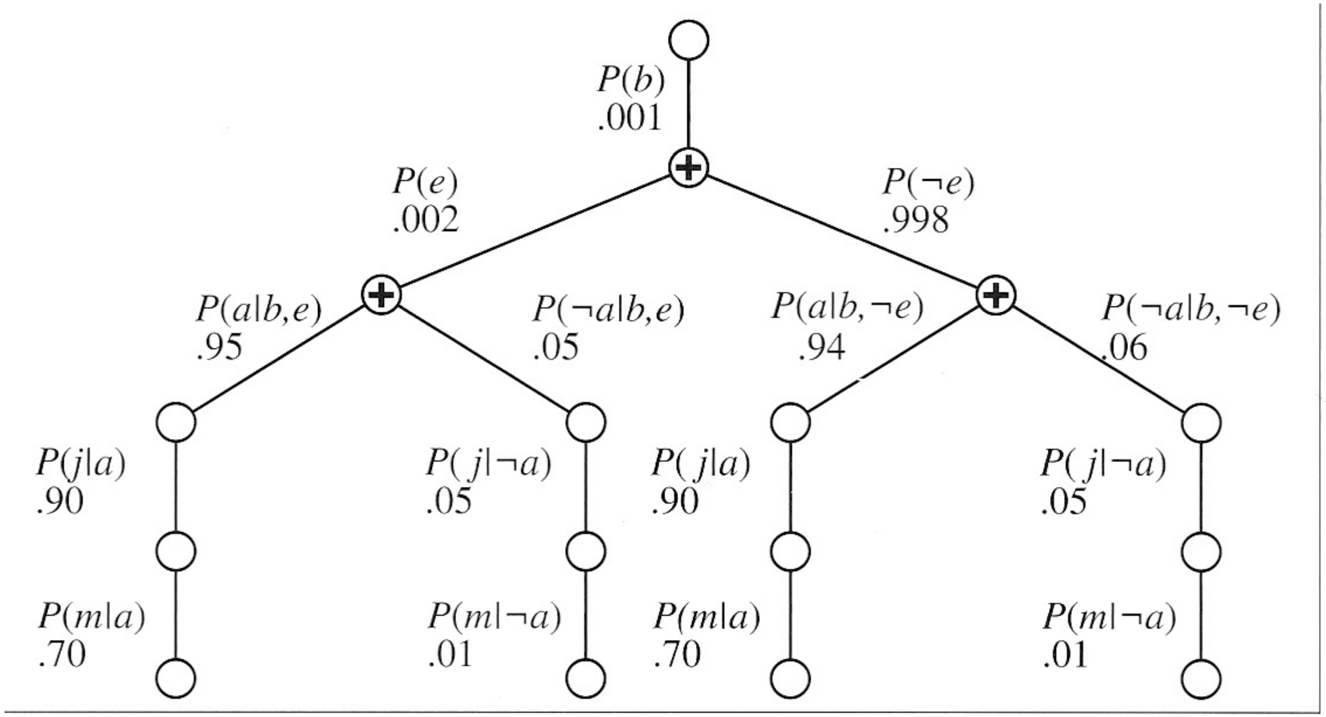
- *P(b)* is constant and can be moved out, *P(e)* term can be moved outside summation *a*

$$P(b \mid j,m) = \alpha P(b) \sum_e P(e) \sum_a P(a \mid b,e) P(j \mid a) P(m \mid a)$$

  - *JohnCalls=true* and *MarryCalls=true,* the probability that the burglary has occurred is about 28%

$$P(B \mid j,m) = \alpha < 0.00059224, 0.0014919 > \approx < 0.284, 0.716 >$$

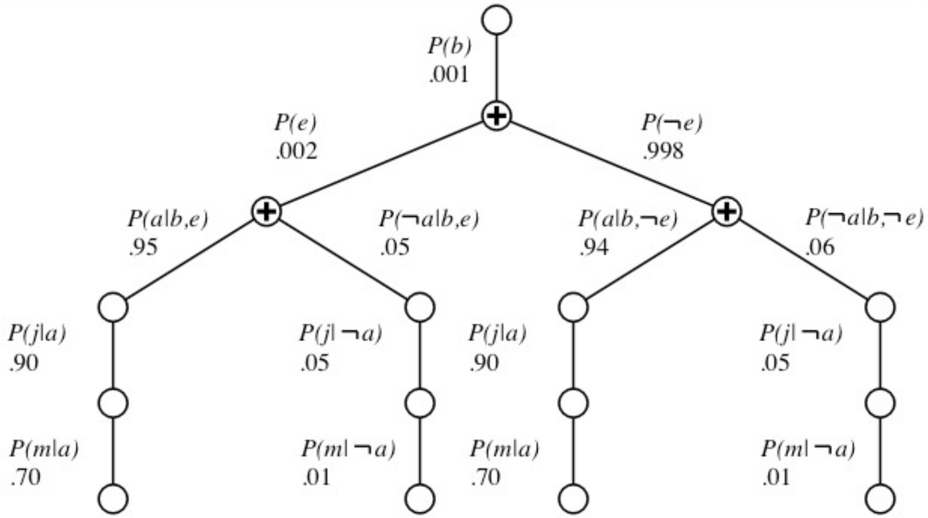
Computation for *Burglary=true*



Variable elimination algorithm

- Eliminate repeated calculation
  - Dynamic programming

Evaluation tree

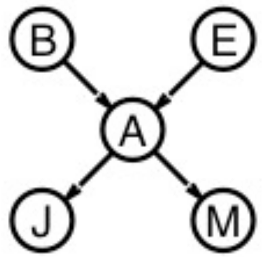


Enumeration is inefficient: repeated computation
e.g., computes $P(j|a)P(m|a)$ for each value of e

Irrelevant variables

- (X query variable, E evidence variables)

Consider the query $P(JohnCalls|Burglary = true)$



$$P(J|b) = \alpha P(b) \sum_e P(e) \sum_a P(a|b,e) P(J|a) \sum_m P(m|a)$$

Sum over m is identically 1; M is **irrelevant** to the query

Thm 1: Y is irrelevant unless $Y \in Ancestors(\{X\} \cup \mathbf{E})$

Here, $X = JohnCalls$, $\mathbf{E} = \{Burglary\}$, and
$Ancestors(\{X\} \cup \mathbf{E}) = \{Alarm, Earthquake\}$
so $MaryCalls$ is irrelevant

Irrelevant variables

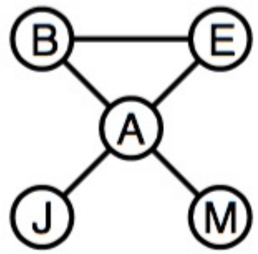
- (X query variable, E evidence variables)

Defn: moral graph of Bayes net: marry all parents and drop arrows

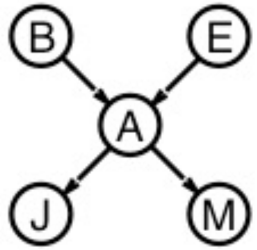
Defn: $\mathbf{A}$ is m-separated from $\mathbf{B}$ by $\mathbf{C}$ iff separated by $\mathbf{C}$ in the moral graph

Thm 2: Y is irrelevant if m-separated from X by $\mathbf{E}$

For $P(JohnCalls|Alarm = true)$, both $Burglary$ and $Earthquake$ are irrelevant

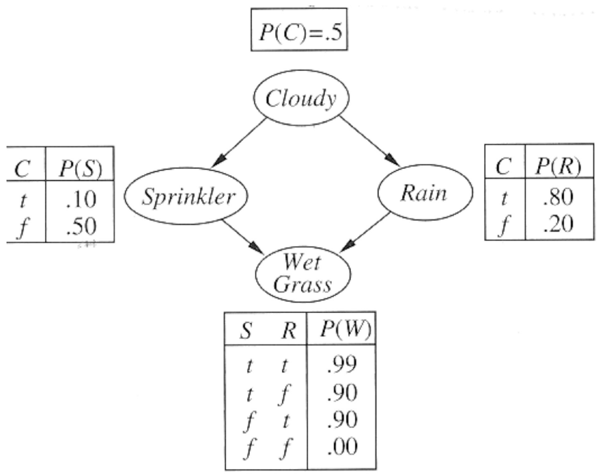


Complexity of exact inference



- The burglary network belongs to a family of networks in which there is *at most one undirected path* between tow nodes in the network
  - These are called singly connected networks or polytrees
- The time and space complexity of exact inference in polytrees is linear in the size of network
  - Size is defined by the number of CPT entries
  - If the number of parents of each node is bounded by a constant, then the complexity will be also linear in the number of nodes

- For multiply connected networks variable elimination can have exponential time and space complexity



Conditional Independence relations in Bayesian networks

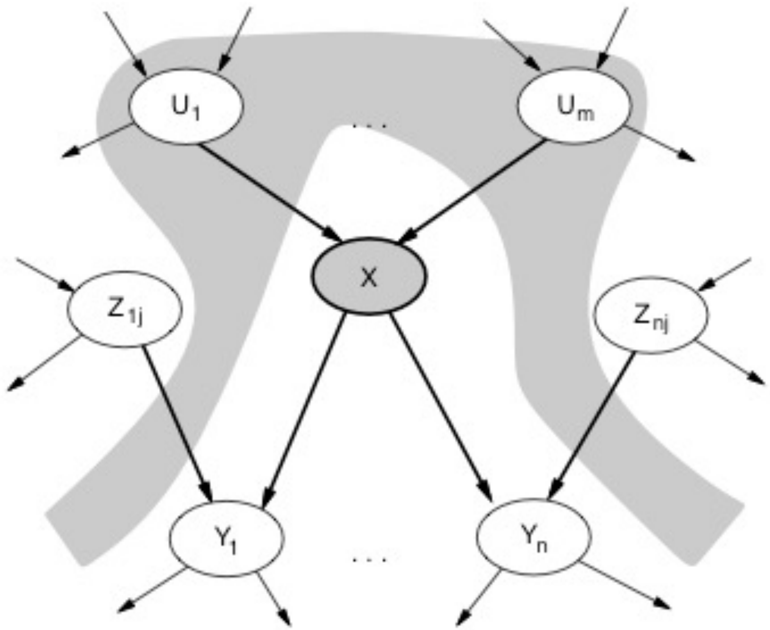
- A Bayesian network is a correct representation of the domain only if each node is conditionally independent of its predecessors in the ordering, given its parents

  $P(MarryCalls|JohnCalls,Alarm,Eathquake,Bulgary)=P(MaryCalls|Alarm)$

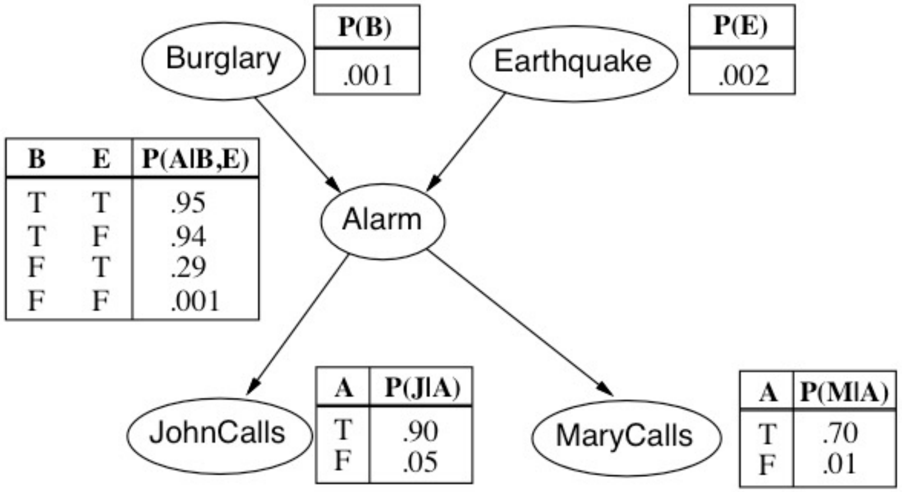
- The topological semantics is given either of the specifications of DESCENDANTS or MARKOV BLANKET

Local semantics

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

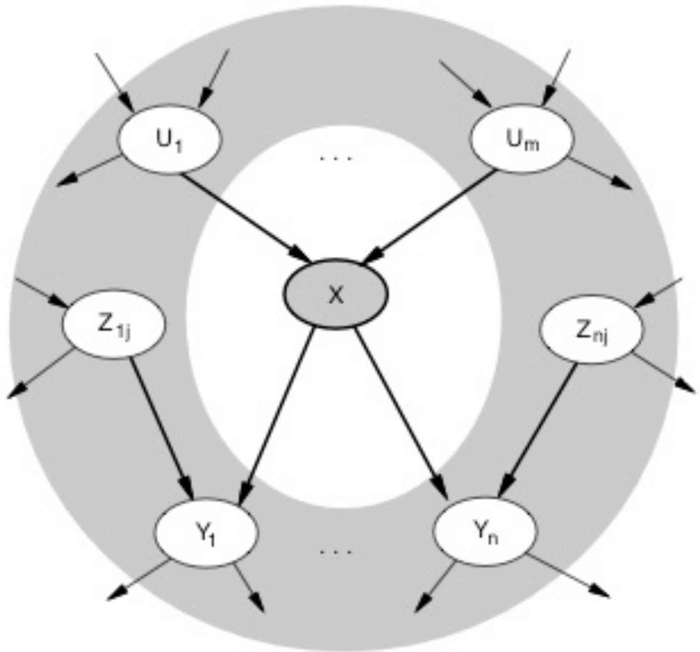


Example

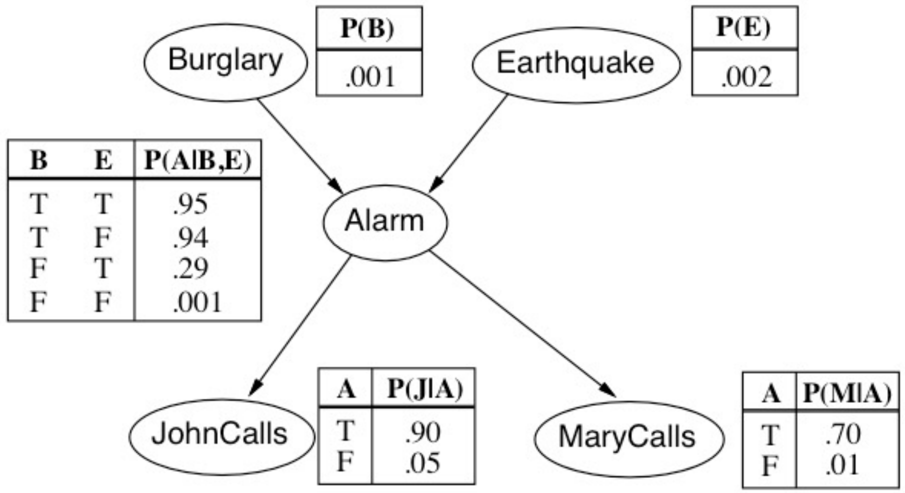


- *JohnCalls* is independent of *Burglary* and *Earthquake* given the value of *Alarm*

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



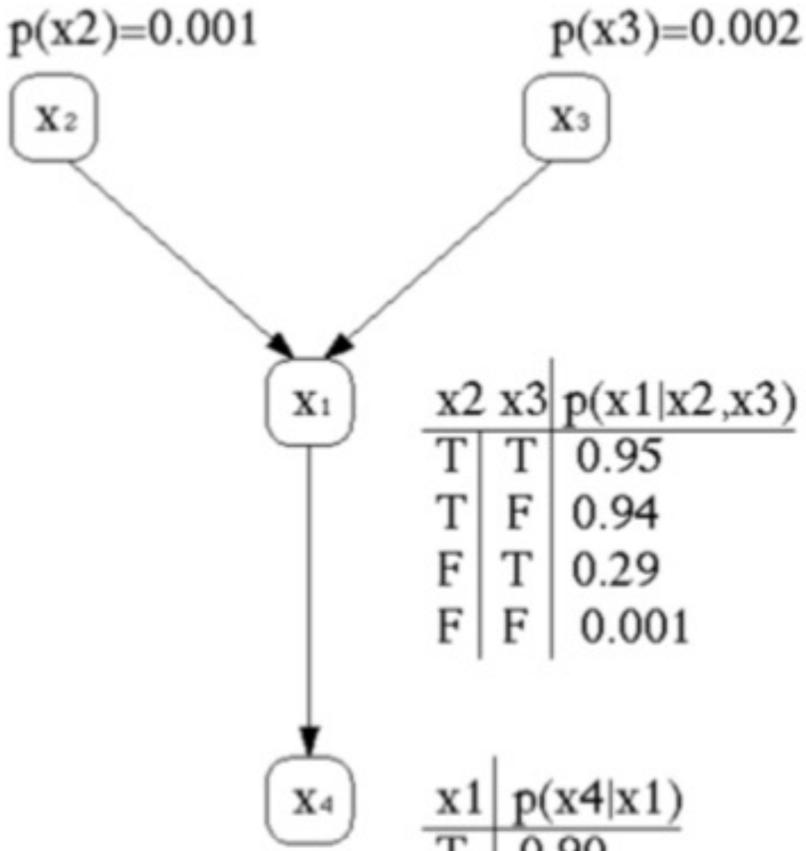
Example



- *Burglary* is independent of *JohnCalls* and *MaryCalls* given *Alarm* and *Earthquake*

Learning of Bayes Nets

- Four categories of learning problems
  - Graph structure may be known/unknown
  - Variable values may be fully observed / partly unobserved
- Easy case: learn parameters for graph structure is *known*, and data is *fully observed*
- Interesting case: graph *known*, data *partly known*
- Gruesome case: graph structure *unknown*, data *partly unobserved*



x2	x3	p(x1|x2,x3)
T	T	0.95
T	F	0.94
F	T	0.29
F	F	0.001

x1	p(x4|x1)
T	0.90
F	0.05

Learning CPTs from Fully Observed Data

Example: Consider learning the parameter $p(x_1|x_2, x_3)$

$$p(x_1|x_2,x_3) = \frac{p(x_1,x_2,x_3)}{p(x_2,x_3)} = \frac{card(x_1 \wedge x_2 \wedge x_3)}{card(x_2 \wedge x_3)} = \frac{\sum_{k=1}^{K} \delta(x_1 = 1, x_2 = 1, x_3 = 1)}{\sum_{k=1}^{K} \delta(x_2 = 1, x_3 = 1)}$$

one writes as well

$$\theta_{x_1|ij} = p(x_1 = 1|x_2 = i, x_3 = j) = \frac{\sum_{k=1}^{K} \delta(x_1 = 1, x_2 = i, x_3 = j)}{\sum_{k=1}^{K} \delta(x_2 = i, x_3 = j)}$$

Maximum likelihood estimate (MLE)

$$p(data|\theta) = \prod_{k=1}^{K} p(x_{(1,k)}x_{(2,k)}, x_{(3,k)}, x_{(4,k)})$$

$$p(data|\theta) = \prod_{k=1}^{K} p(x_{(4,k)}|x_{(1,k)}) \cdot p(x_{(1,k)}|x_{(2,k)}, x_{(3,k)}) \cdot p(x_{(2,k)}) \cdot p(x_{(3,k)})$$

$$\log p(data|\theta) = \sum_{k=1}^{K} \log p(x_{(4,k)}|x_{(1,k)}) + \log p(x_{(1,k)}|x_{(2,k)}, x_{(3,k)}) + \log p(x_{(2,k)}) + \log p(x_{(3,k)})$$

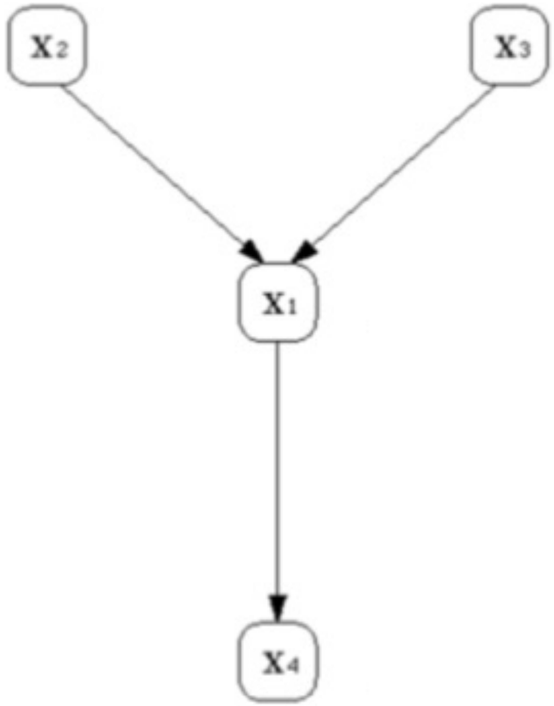
$$\frac{\partial \log p(data|\theta)}{\partial \theta_{x1|x2,x3}} = \sum_{k=1}^{K} \frac{\partial \log p(x_{(1,k)}|x_{(2,k)}, p(x_{(3,k)}))}{\partial \theta_{x1|x2,x3}}$$

$$\theta_{x_1|ij} = p(x_1 = 1|x_2 = i, x_3 = j) = \frac{\sum_{k=1}^{K} \delta(x_1 = 1, x_2 = i, x_3 = j)}{\sum_{k=1}^{K} \delta(x_2 = i, x_3 = j)}$$

Expectation Maximization

If $X = \{x_2, x_3, x_4\}$ observe

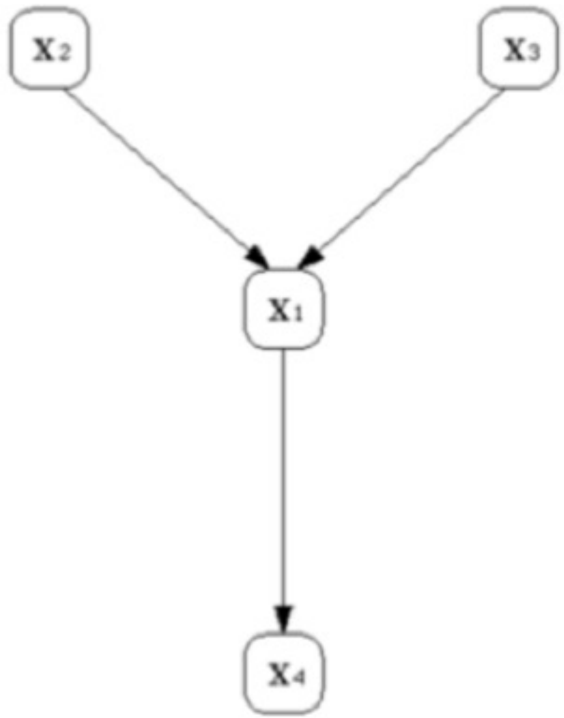
If $Z = \{x_1\}$ is unobserved



Initialization

Choosing an initial value $\boldsymbol{\theta}^{old}$

In our case random probability values for
$p(x_1|x_2, x_3)$, $p(x_1|x_2, \neg x_3)$
$p(x_1|\neg x_2, x_3)$, $p(x_1|\neg x_2, \neg x_3)$



E-Step

$$\log p(X, Z|\theta) = \sum_{k=1}^{K} \log p(x_{(4,k)}|x_{(1,k)}) + \log p(x_{(1,k)}|x_{(2,k)}, x_{(3,k)}) + \log p(x_{(2,k)}) + \log p(x_{(3,k)})$$

In E step we calculate for each training example k, $p(X, Z|\theta)$. In the first step we use the random probability values

$$p(x_{(1,k)}|x_{(2,k)}, x_{(3,k)}, x_{(4,k)}) = \frac{p(x_{(1,k)}, x_{(2,k)}, x_{(3,k)}, x_{(4,k)})}{p(x_{(1,k)}, x_{(2,k)}, x_{(3,k)}, x_{(4,k)}) + p(\neg x_{(1,k)}, x_{(2,k)}, x_{(3,k)}, x_{(4,k)})}$$

$$E[x_{(1,k)}] = p(x_{(1,k)} = 1|x_{(2,k)}, x_{(3,k)}, x_{(4,k)}) =$$

$$= \frac{p(x_{(1,k)} = 1, x_{(2,k)}, x_{(3,k)}, x_{(4,k)})}{p(x_{(1,k)} = 1, x_{(2,k)}, x_{(3,k)}, x_{(4,k)}) + p(x_{(1,k)} = 0, x_{(2,k)}, x_{(3,k)}, x_{(4,k)})}$$

M-Step

Update all relevant parameters

$$\theta_{x_1|ij} = p(x_1 = 1|x_2 = i, x_3 = j) = \frac{\sum_{k=1}^{K} \delta(x_2 = i, x_3 = j) \cdot E[x_{(1,k)}]}{\sum_{k=1}^{K} \delta(x_2 = i, x_3 = j)}$$

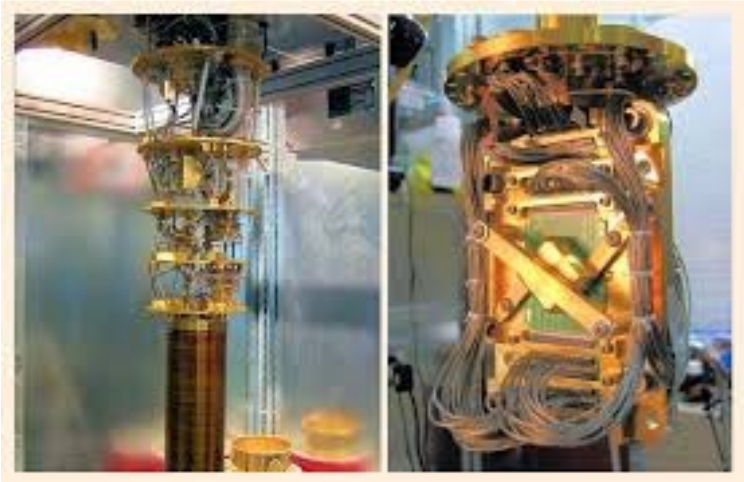
$$p(x_{(1,k)}|x_{(2,k)}, x_{(3,k)}, \theta) = \frac{\sum_{k=1}^{K} \delta(x_2, x_3) \cdot E[x_{(1,k)}]}{\sum_{k=1}^{K} \delta(x_2, x_3)}$$

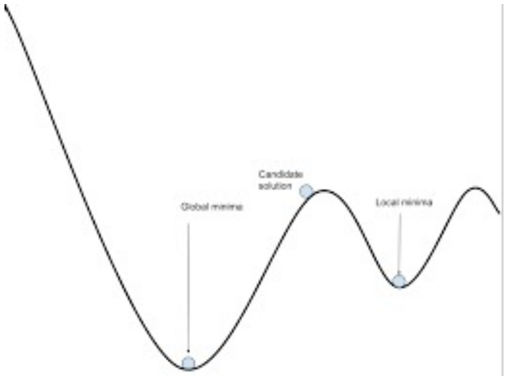
remember, before it was:

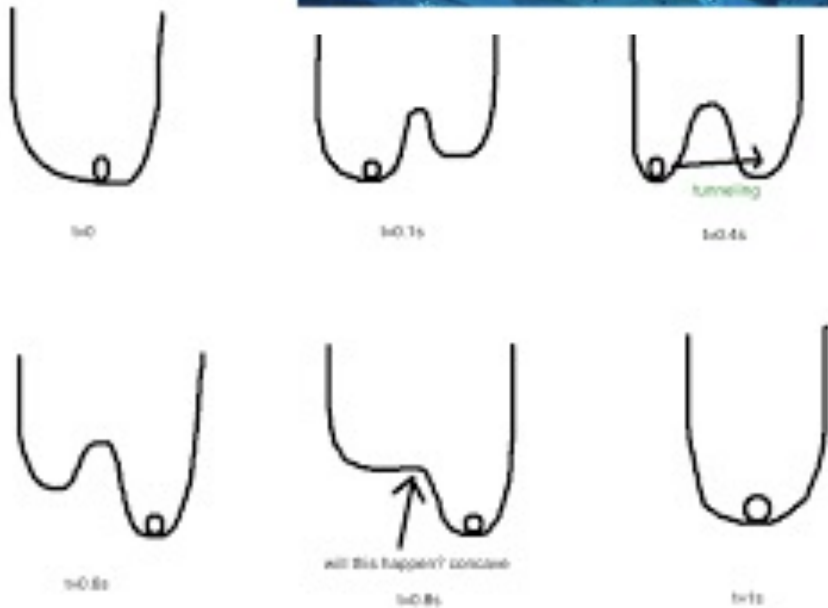
$$\theta_{x_1|ij} = p(x_1 = 1|x_2 = i, x_3 = j) = \frac{\sum_{k=1}^{K} \delta(x_1 = 1, x_2 = i, x_3 = j)}{\sum_{k=1}^{K} \delta(x_2 = i, x_3 = j)}$$

repeat until the value converges.

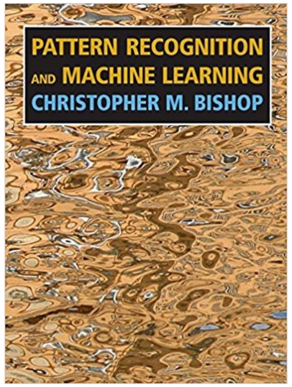
Next: Stochastic Methods



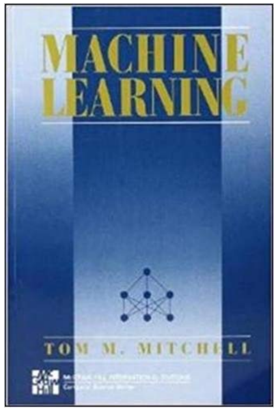




Literature

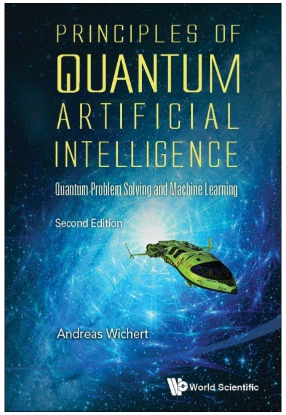


- Christopher M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), Springer 2006
  - Chapter 8



- Tom M. Mitchell, Machine Learning, McGraw-Hill; 1st edition (October 1, 1997)
  - Section 6.11, 6.12

Literature



- Principles of Quantum Artificial Intelligence: Quantum Problem Solving and Machine Learning, 2nd Edition, World Scientific, 2020
  - Chapter 6