Lecture No. 8 – Bayesian Learning

- Introduction to Bayesian Learning
- Bayes Theorem
- Naïve Bayes Algorithm
- Bayesian Belief Networks

Rev. Thomas Bayes
(1702-1761)

Introduction to Bayesian Learning

- Basic Assumption
  - The observed data is governed by *probability distributions*
- Features
  - Using prior knowledge on probability distributions
  - Incremental learning of probabilities
    - Each observed example can incrementally increase or decrease the estimated probability
  - Probabilistic predictions of target values
  - Prediction by multiple hypotheses
  - A standard of **optimal decision making**
- Practical Algorithms
  - Naïve Bayes Classifier
  - Bayesian Belief Networks

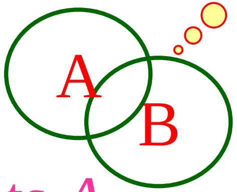
Basic Formulas for Probabilities

A – has other investments

B – credit = “Yes”

- Product Rule
  - Probability $P\ (A \wedge B)$ of a conjunction of two events A and B:
    - $P\ (A \wedge B) = P(A\,/\,B)\ P(B) = P(B\,/\,A)\ P(A)$

Independent events?



- Sum Rule
  - Probability of a disjunction of two events A and B:
    - $P\ (A \vee B) = P(A) + P(B) - P\ (A \wedge B)$

Disjoint events?

- Theorem of Total Probability
  - If events $A_1, \ldots, A_n$ are mutually exclusive with $\sum_i P(A_i) = 1$, then
    - $P(B) = \sum_i P(B/A_i)P(A_i)$

A_1 – has other investments

A_2 – no other investments

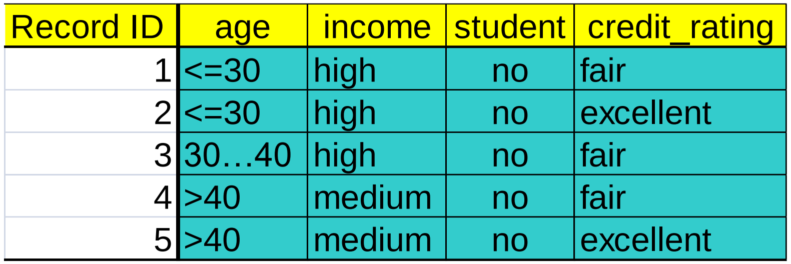
B – credit = “Yes”

Lecture No. 8 – Bayesian Learning

- Introduction to Bayesian Learning
- Bayes Theorem ⇦
- Naïve Bayes Algorithm
- Bayesian Belief Networks

Bayesian Theorem: Basics

- Let **X** be a record ("*evidence*"):



Record ID	age	income	student	credit_rating
1	<=30	high	no	fair
2	<=30	high	no	excellent
3	30…40	high	no	fair
4	>40	medium	no	fair
5	>40	medium	no	excellent

- Let H be a *hypothesis* that assigns *X* to class *C*
  - Optional classes: *Buys_Computer = Yes* and *Buys_Computer = No*
- Classification is to determine *P(H|**X**)*, the probability that the hypothesis holds given the observed data sample ***X***

Bayesian Theorem: Basics (cont.)

- *P(H)* (*prior probability*), the initial probability that the hypothesis *H* is correct
  - E.g., ***X*** will buy computer, regardless of age, income, …
- *P(**X**) (evidence):* probability to observe a given record
  - E.g., the prob. that *X* is 31..40, medium income, etc.
- *P(**X**|H)* (*likelihood*), the probability of observing the record ***X***, given that the hypothesis holds
  - E.g., Given *H* (***X*** will buy computer), the prob. that *X* is 31..40, medium income, etc.

Bayes Theorem

- Given training data *X*, posteriori probability of a hypothesis *H*, *P(H|X)* follows the Bayes theorem

$$P(H\,|\,X) = \frac{P(X\,|\,H)P(H)}{P(X)}$$

- Example
  - *P(Buys_Computer = Yes / Age = 31..40; income=medium; student = no)=*

$$\frac{P(Age\ \text{is}\ 31..40;\ \text{income}=\text{medium};\ \text{student}=\text{no}\,|\,Buys\_Computer=Yes)P(Buys\_Computer=Yes)}{P(X\ \text{is}\ 31..40)}$$

- Informally, this can be written as
  - posterior =likelihood x prior / evidence

MAP (Maximum Posteriori) Hypothesis

- MAP (maximum posteriori) hypothesis

$$h_{MAP} \equiv \arg\max_{h \in H} P(h|D) = \arg\max_{h \in H} P(D|h)P(h).$$

- *D* – training data set
- Optional hypotheses: ?
- Example
  - *X = 31..40*

$$h_{MAP} = \max\{P(31..40|Yes)P(Yes),\ \ P(31..40|No)P(No)\}.$$

- Practical difficulty: require initial knowledge of many probabilities, curse of dimensionality, significant computational cost

Bayesian Theorem Example

Does patient have cancer or not?

Source: Mitchell (1997)

- A patient takes a lab test and the result comes back positive.
- It is known that the test returns a correct positive result in only 98% of the cases and a correct negative result in only 97% of the cases.
- Furthermore, only 0.008 of the entire population has this disease.

1. What is the probability that this patient has cancer?
2. What is the probability that he does not have cancer?
3. What is the most probable diagnosis?

Does patient have cancer or not? (cont'd)

- Medical diagnosis problem (D = ?, h = {…}?)

  P (cancer) = .008, P (¬cancer) = .992

  P (⊕ / cancer) = .98, P (Θ / cancer) = .02

  P (⊕ / ¬cancer) = .03 P (Θ / ¬cancer) = .97

- Maximum A Posteriori Hypothesis

  P(⊕|cancer)P(cancer) = (.98).008 = .0078

  P(⊕ |¬cancer)P(¬cancer) = (.03).992 = .0298

  h_{MAP} = ¬cancer

- Probability of Cancer

  P(cancer|⊕) = 0.0078 / (0.0078 + 0.0298) = .21

Diagnosis?

- Practical implications?

Lecture No. 8 – Bayesian Learning

- Introduction to Bayesian Learning
- Bayes Theorem
- Naïve Bayes Algorithm ⇦
- Bayesian Belief Networks

Towards Naïve Bayesian Classifier

- Let *D* be a training set of tuples and their associated class labels, and each tuple is represented by an *n*-D attribute vector $\boldsymbol{X} = (x_1, x_2, \ldots, x_n)$
- Suppose there are *m* classes $C_1, C_2, \ldots, C_m$
- Classification is to derive the maximum posteriori, i.e., the maximal $P(C_i|\boldsymbol{X})$
- This can be derived from Bayes' theorem

$$P(C_i|\mathbf{X}) = \frac{P(\mathbf{X}|C_i)P(C_i)}{P(\mathbf{X})}$$

- Since *P(X)* is constant for all classes, only

$$P(C_i|\mathbf{X}) = P(\mathbf{X}|C_i)P(C_i)$$

needs to be maximized

Naïve Bayes Classifier (NBC)

- A simplified assumption: attributes are conditionally independent:
- The probability of occurrence of x_1 and x_2 given the current class is C: $P([x_1,x_2]/C) = P(x_1/C) * P(x_2/C)$
- No dependence relation between attributes
- Greatly reduces the computation cost, only count the class distribution.
- Once the probability $P(X|C_i)$ is known, assign X to the class with maximum $P(X|C_i)*P(C_i)$

$$P(X \mid C_i) \;=\; \prod_{k=1}^{n} P(x_k \mid C_i)$$

$$C_{NB} = \arg\max_{C_i} \; P(C_i) * \prod_{k=1}^{n} P(x_k \mid C_i)$$

Training dataset – Example 1

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

Data record
X =(age<=30,
Income=medium,
Student=yes
Credit_rating=
Fair)
Class = ?

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 1

- 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

- 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 0.667 =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")=0.044 * 9/14=0.028
- P(*X*|buys_computer="no") * P(buys_computer="no")=0.019 * 5/14 = 0.007
- *X* belongs to class "buys_computer=yes"
- **P(buys_computer ="yes" / *X*) = ?**

Naïve Bayes Classifier (NBC) Example 2: Text Classification

Documents are classified as being scientific or commercial by the occurrence of the following three words: “paper”, “research”, and “product”. The data obtained from 100 scientific documents and 100 commercial documents is summarized below:

Document	"Paper"	"Research"	"Product"
Scientific	80	90	20
Commercial	50	20	90

Explanation: 80 scientific documents included the word “paper”, 90 commercial documents included the word “product”, etc.

NBC Example 2 (cont.)

Classify the following text using the Naïve Bayes algorithm:

To combine the power of new and existing security investments made by our customers, the IBM Threat Protection System leverages information gathered from the Ready for IBM Security Intelligence ecosystem of more than 400 third-party **product**s from over 90 vendors. You can take advantage of these third-party solutions to increase visibility into security events, collapse information silos and gain insights on advanced attacks.

Source: http://www-03.ibm.com/security/threat-protection/?lnk=ushpls1

NBC Example 2 (cont.)

Training Data

Document	"Paper"	"Research"	"Product"
Scientific	80	90	20
Commercial	50	20	90
Document	"Paper"	"Research"	"Product"
Scientific	0.8	0.9	0.2
Commercial	0.5	0.2	0.9

Test Data

Relative probability of each class

	Apriori	"Paper"	"Research"	"Product"	Total	Rel.
P (Scientific)	0.5	0.2	0.1	0.2	**0.002**	**0.011**
P (Commercial)	0.5	0.5	0.8	0.9	**0.180**	**0.989**
					0.182	

Estimating Probabilities

$$C_{NB} = \arg\max_{C_i} P(C_i) * \prod_{k=1}^{n} P(x_k \mid C_i)$$

- Two difficulties of estimating probability
  1. $\frac{n_c}{n}$ produces a biased underestimate of the probability
  2. When this probability estimate is zero, this probability term will dominate the Bayes classifier
- Solution: using the m-estimate defined as follows

m-estimate of probability: $$\frac{n_c + mp}{n + m}$$

n_c : number of examples for which class $v = v_j$ and attribute $a = a_i$

n : number of training examples for which class $v = v_j$

m : equivalent sample size

p : prior (e.g., uniform)

If $m = 0$, the m-estimate is equivalent to $\frac{n_c}{n}$

m-Estimate Example

- Ex. Suppose a dataset with 1000 tuples, income=low (0), income= medium (990), and income = high (10)
- Let equivalent sample size *m* = 100
- Uniform prior: 1/3
- $mp = 33^1/_3$
- Use m- estimate

  Prob(income = low) = 33.33/1100 = 0.030

  Prob(income = medium) = (990+33.33)/1100 = 0.930

  Prob(income = high) = (10+33.33)/1100 = 0.039
- What if *m* = 0 ?

Laplacian Estimator

- Ex. Suppose a dataset with 1000 tuples, income=low (0), income= medium (990), and income = high (10)
- Use Laplacian correction (or Laplacian estimator)
  - Adding 1 to each case

    Prob(income = low) = 1/1003 = 0.001

    Prob(income = medium) = 991/1003 = 0.988

    Prob(income = high) = 11/1003 = 0.011
  - The “corrected” prob. estimates are close to their “uncorrected” counterparts

Naïve Bayesian Classifier: Comments

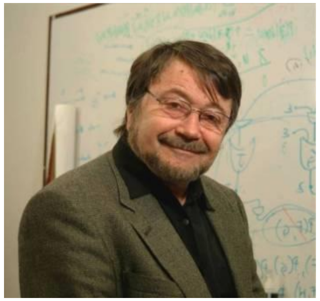
- Advantages :
  - Easy to implement
  - Good results obtained in most of the cases
- Disadvantages
  - Assumption: class conditional independence , therefore loss of accuracy
  - Practically, dependencies exist among variables
  - E.g., hospitals: patients: Profile: age, family history etc
  - Symptoms: fever, cough etc., Disease: lung cancer, diabetes etc
  - Dependencies among these cannot be modeled by Naïve Bayesian Classifier
- How to deal with these dependencies?
  - Bayesian Belief Networks

Lecture No. 8 – Bayesian Learning

- Introduction to Bayesian Learning
- Bayes Theorem
- Naïve Bayes Algorithm
- Bayesian Belief Networks ⇦

Bayesian Belief Networks

- Naïve Bayes is based on assumption of conditional independence
- Bayesian networks provide a tractable method for specifying dependencies among variables



Example (from Judea Pearl*)

* - 2011 winner of the ACM Turing Award

You have a new burglar alarm installed at home. It is fairly reliable at detecting a burglary, but also responds on occasion to minor earthquakes. You also have two neighbors, John and Mary, who have promised to call you at work when they hear the alarm. John always calls when he hears the alarm, but sometimes confuses the telephone ringing with the alarm and calls then, too. Mary, on the other hand, likes rather loud music and sometimes misses the alarm altogether. Given the evidence of who has or has not called, we would like to estimate the probability of a burglary.

Terminology (see the example)

- A Bayesian Belief Network describes the probability distribution over a set of random variables $Y_1, Y_2, \ldots Y_n$
- Each variable Y_i can take on the set of values *V(Yi)*
- The joint space of the set of variables *Y* is the cross product
- $V(Y_1) \times V(Y_2) \times \ldots \times V(Y_n)$
- Each item in the joint space corresponds to one possible assignment of values to the tuple of variables *<Y1, …Yn>*
- Joint probability distribution: specifies the probabilities of the items in the joint space
- A Bayesian Network provides a way to describe the joint probability distribution in a compact manner.

Conditional Independence (see the example)

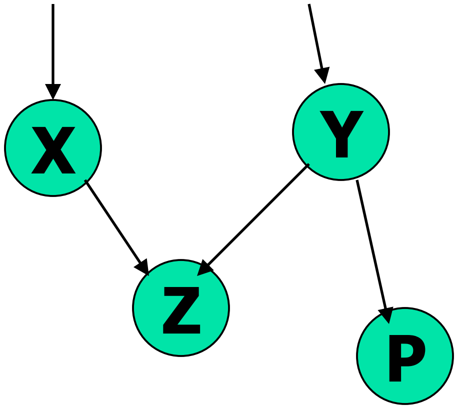
- Let *X*, *Y*, and *Z* be three discrete-valued random variables.
- We say that *X* is conditionally independent of *Y* given *Z* if the probability distribution governing *X* is independent of the value of *Y* given a value for *Z*

$$\forall x_i, y_j, z_k P(X = x_i \mid Y = y_j, Z = z_k) = P(X = x_i \mid Z = z_k)$$

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

Bayesian Belief Network

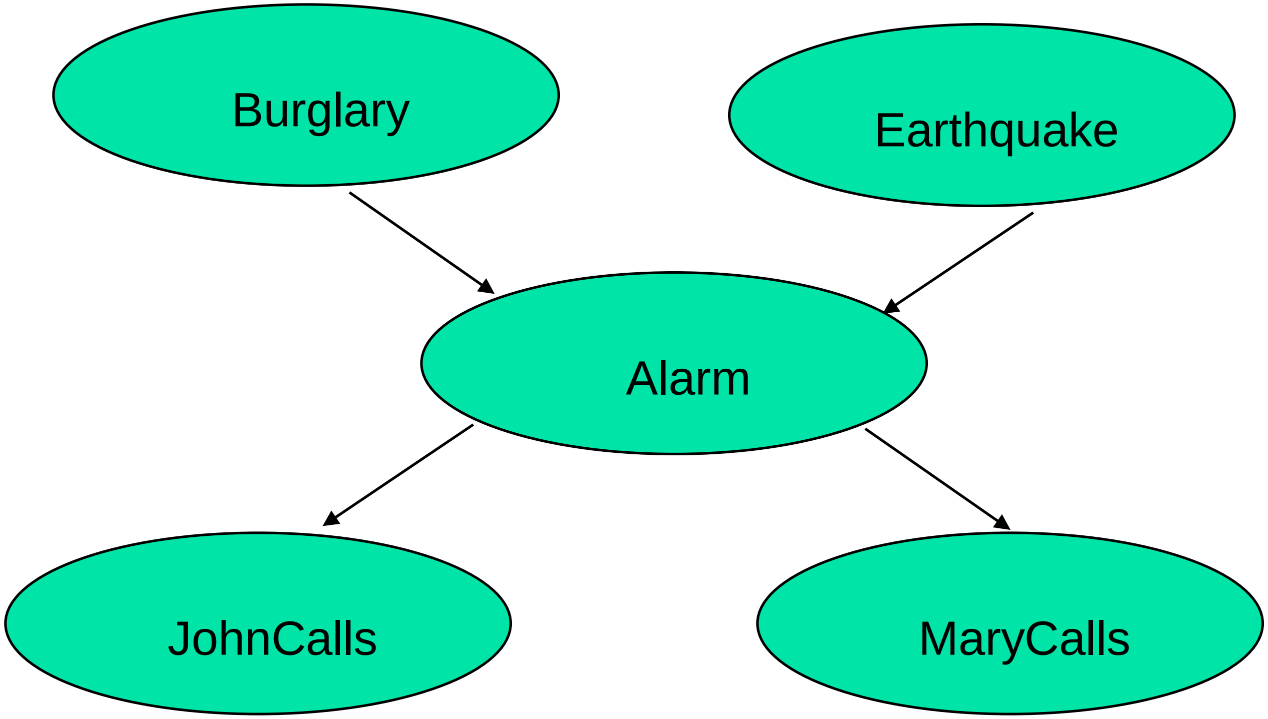
- A set of random variables makes up the *nodes* of the network
- A set of *directed links* or arrows connects pairs of nodes.
- Each node has a *conditional probability table* that quantifies the effects that the *parents* have on the node.
- The graph has no directed cycles (it is a DAG)



Step 1

- Determine what the propositional (random) variables should be
- Determine causal (or another type of influence) relationships and develop the topology of the network

Topology of Belief Network



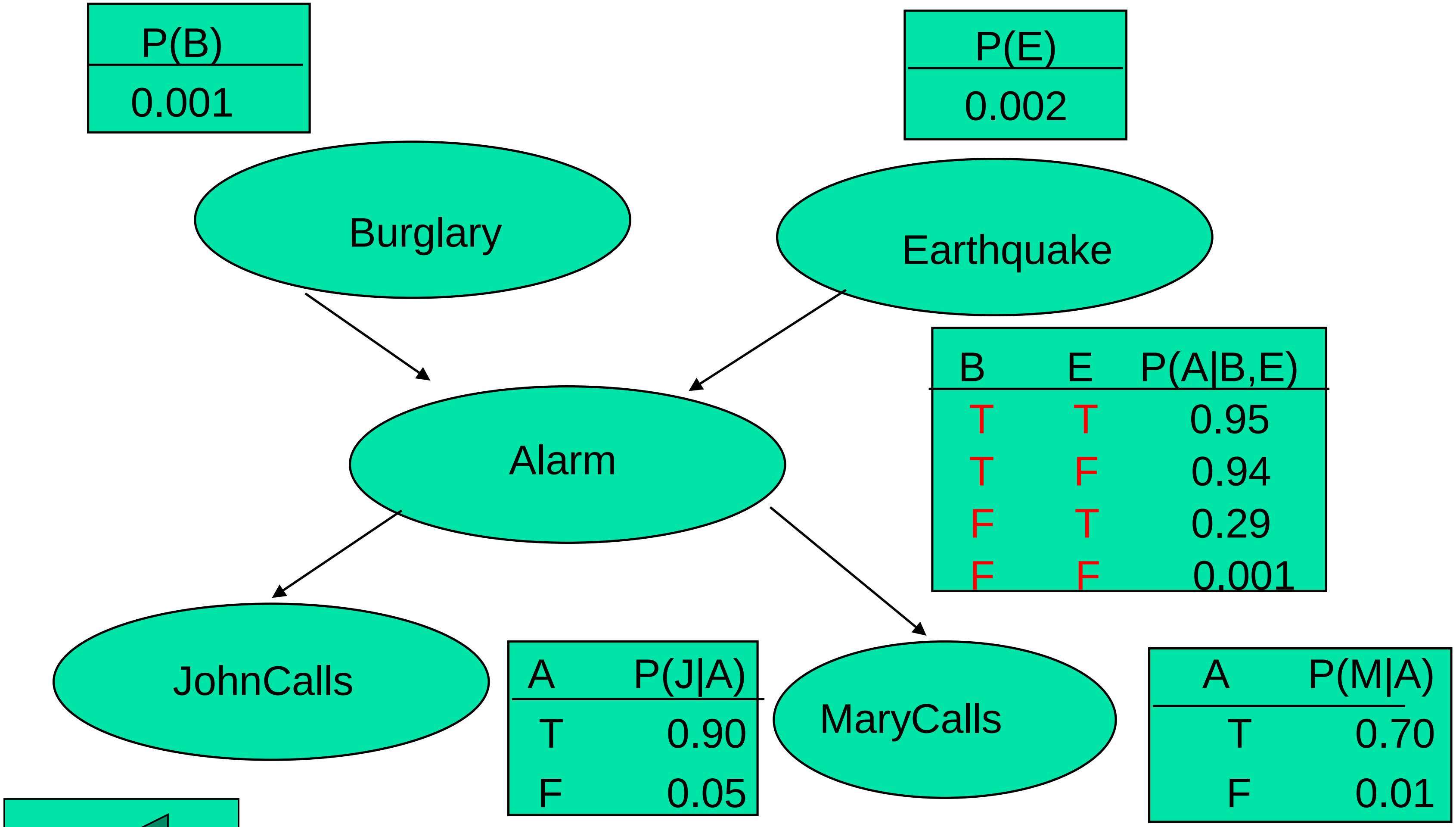
Step 2

- Specify a *conditional probability table* or CPT for each node.
- Each row in the table contains the conditional probability of each node value for a conditioning case (possible combinations of values for parent nodes).
- In the example, the possible values for each node are true/false.
- The sum of the probabilities for each value of a node given a particular conditioning case is 1.

Example:
CPT for Alarm Node

Burglary	Earthquake	P(Alarm|Burglary,Earthquake)	
		True	False
True	True	0.950	0.050
True	False	0.940	0.060
False	True	0.290	0.710
False	False	0.001	0.999

Complete Belief Network



P(B)
0.001

P(E)
0.002

B	E	P(A|B,E)
T	T	0.95
T	F	0.94
F	T	0.29
F	F	0.001

A	P(J|A)
T	0.90
F	0.05

A	P(M|A)
T	0.70
F	0.01

Network as representation of joint

- A generic entry in the joint probability distribution is the probability of a conjunction of particular assignments to each variable, such as:

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

- Each entry in the joint is represented by the product of appropriate elements of the CPTs in the belief network.

Example Calculation

Calculate the probability of the event that the alarm has sounded but neither a burglary nor an earthquake has occurred, and both John and Mary call.

$P(J \wedge M \wedge A \wedge {\sim}B \wedge {\sim}E)$

$= P(J|A)\ P(M|A)\ P(A|{\sim}B,{\sim}E)\ P({\sim}B)\ P({\sim}E)$

$= 0.90 * 0.70 * 0.001 * 0.999 * 0.998$

$= 0.00062$

Inference Methods for Bayesian Networks

- We may want to infer the value of some target variable (Burglary) given observed values for other variables.
- What we generally want is the probability distribution
- Inference straightforward if all other values in network known
- More general case, if we know a subset of the values of variables, we can infer a probability distribution over other variables.
- NP-Hard problem
- But approximations work well

How Are Bayesian Networks Constructed?

- **Subjective construction**: Identification of (direct) causal structure
  - People are quite good at identifying direct causes from a given set of variables & whether the set contains all relevant direct causes
  - Markovian assumption: Each variable becomes independent of its non-effects once its direct causes are known
  - E.g., S ‹— F —› A ‹— T, path S—›A is blocked once we know F—›A
  - HMM (Hidden Markov Model): often used to model dynamic systems whose states are not observable, yet their outputs are
- **Synthesis from other specifications**
  - E.g., from a formal system design: block diagrams & info flow
- **Learning from data**
  - E.g., from medical records or student admission record
  - Learn parameters give its structure or learn both structure and parms
  - Maximum likelihood principle: favors Bayesian networks that maximize the probability of observing the given data set

Training Bayesian Networks

Variables believed to influence but not observable

- Several cases
  - Given both the network structure and all variables observable: learn only the CPTs (similar to NBC)
  - Network structure known, some *hidden variables*: method of gradient descent, analogous to neural network learning
  - Network structure unknown, all variables observable: search through the model space to reconstruct graph topology
  - Unknown structure, all hidden variables: no good algorithms known for this purpose
- D. Heckerman. A Tutorial on Learning with Bayesian Networks. In *Learning in Graphical Models*, M. Jordan, ed. MIT Press, 1999.