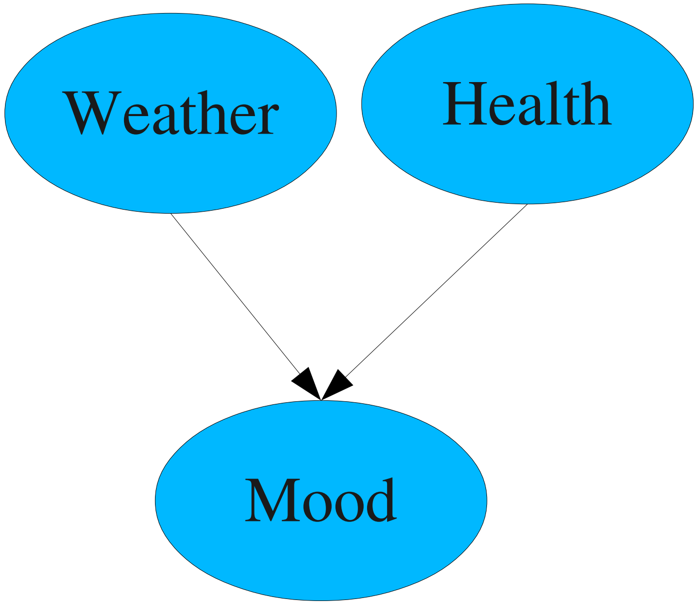
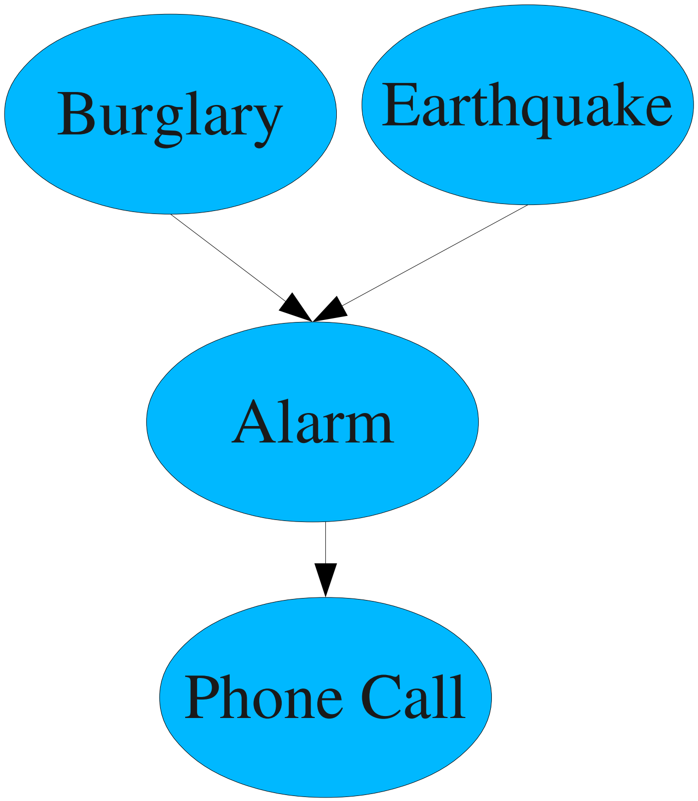
Bayesian Networks

Bayesian Networks

- A Bayesian network is a directed acyclic graph that represents causal (?) relationships between random variables.





Bayesian Networks

- Bayes' nets have the following property:
  - Each variable is conditionally independent of all it's non-descendants in the graph, given the value of it's parents.

$$P(X_1, X_2, \ldots, X_N) = \prod_{i=1}^{N} P(X_i | parents(X_i))$$

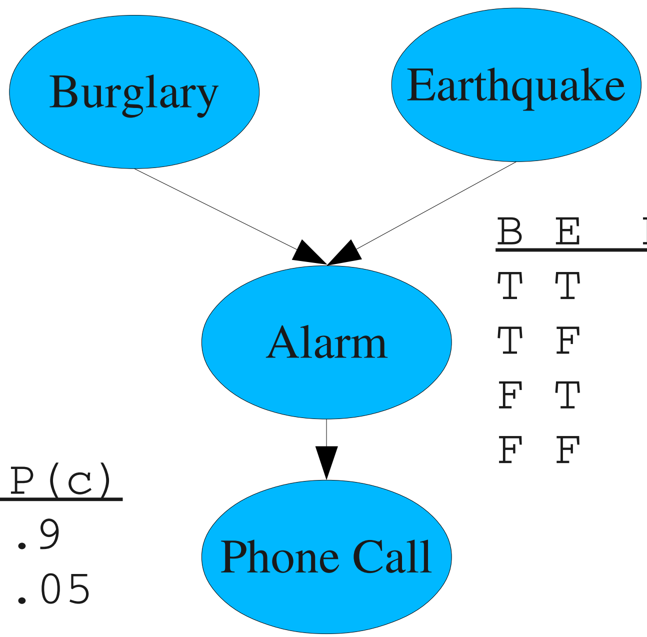
- In other words, the complete joint probability distribution can be reconstructed from the *N* conditional distributions.
- For *N* binary valued variables with *M* parents each
  - 2^N vs. $N * 2^M$

Specifying a Bayes' Net

- We need to specify:
  - The topology of the network.
  - The conditional probabilities.

P(b)
.001

P(e)
.002



B	E	P(a)
T	T	.95
T	F	.94
F	T	.29
F	F	.001

A	P(c)
T	.9
F	.05

(Simplistic) Spam Filtering

SPAM

viagra	discount	cs444	count
T	T	T	0
T	T	F	180
T	F	T	0
T	F	F	1200
F	T	T	8
F	T	F	600
F	F	T	12
F	F	F	6000
Total:			8000

NON-SPAM

viagra	discount	cs444	count
T	T	T	0
T	T	F	0
T	F	T	1
T	F	F	3
F	T	T	6
F	T	F	20
F	F	T	70
F	F	F	700
Total:			800

Quiz! Estimate:

$P(\neg spam) = ??$ $\qquad P(\neg viagra, discount, cs444 | spam) = ??$

Bayes' Classifier I

- Assume a multivalued random variable C that can take on the values c_i for i = 1 to i=K.
- Assume M input attributes X_j for j = 1 to M.
- Learn $P(X_1, X_2, \ldots, X_M \mid c\_i)$ for each i.
  - Treat this as K different joint PDs.
- Given a set of input values ($X_1=u_1, X_2=u_2, \ldots, X_M=u_M$), classification is easy (?):

$$C^{predict} = \underset{c_i}{argmax}\, P\left(C=c_i \mid X_1=u_1, \ldots, X_M=u_m\right)$$

An Aside: MAP vs. ML

- This is a maximum a posteriori (MAP) classifier:

$$C^{predict} = \underset{c_i}{\operatorname{argmax}} P\left(C = c_i \middle| X_1 = u_1, \ldots, X_M = u_m\right)$$

- We could also consider a maximum likelihood (ML) classifier:

$$C^{predict} = \underset{c_i}{\operatorname{argmax}} P\left(X_1 = u_1, \ldots, X_M = u_m \middle| C = c_i\right)$$

Bayes' Classifier II

$$C^{predict} = \underset{c_i}{argmax}\, P\left(C = c_i \middle| X_1 = u_1, \ldots, X_M = u_m\right)$$

- Apply Bayes' rule

$$C^{predict} = \underset{c_i}{argmax} \frac{P\left(X_1 = u_1, \ldots, X_M = u_m \middle| C = c_i\right) P\left(C = c_i\right)}{P\left(X_1 = u_1, \ldots, X_M = u_m\right)}$$

- Conditioning:

$$C^{predict} = \underset{c_i}{argmax} \frac{P\left(X_1 = u_1, \ldots, X_M = u_m \middle| C = c_i\right) P\left(C = c_i\right)}{\sum_{i=1}^{K} P\left(X_1 = u_1, \ldots, X_M = u_m \middle| C = c_i\right) P\left(C = c_i\right)}$$

Bayes' Classifier III

$$C^{predict} = \underset{c_i}{argmax} \frac{P(X_1 = u_{1,}\ldots, X_M = u_m | C = c_i) P(C = c_i)}{\sum_{i=1}^{K} P(X_1 = u_1, \ldots, X_M = u_m | C = c_i) P(C = c_i)}$$

- Notice that the denominator is the same for all classes.
- We can simplify this to:

$$C^{predict} = \underset{c_i}{argmax}\, P(X_1 = u_1, X_2 = u_2, \ldots, X_M = u_m | C = c_i) P(C = c_i)$$

- If you have the true distributions, this is the <u>best</u> choice.
- What's the problem?

Naïve Bayes' Classifier

- If *M* is largish it is impossible to learn

  $P(X_1, X_2, \ldots, X_M \mid c\_i)$.

- The solution (?): assume that the X_j are independent given C (that the symptoms are independent, given the disease.)

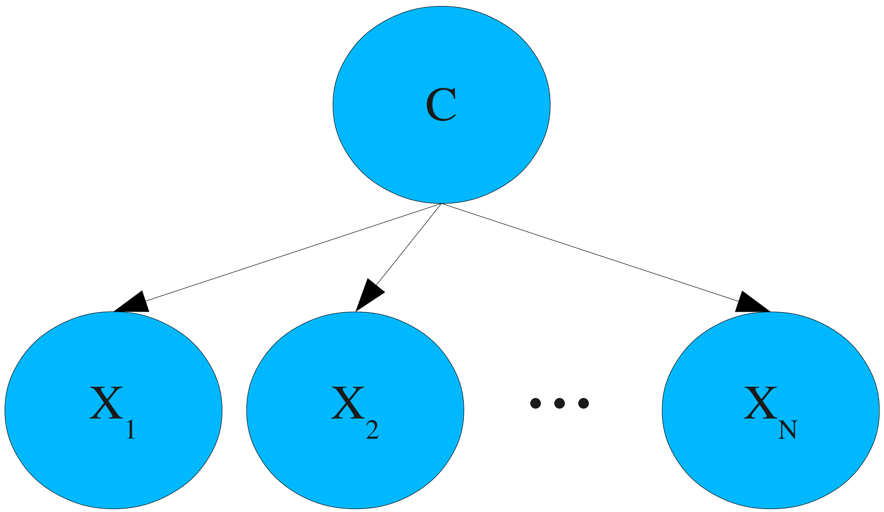
$$P(X_1,\ldots,X_M|c_i)=\prod_{j=1}^{M} P(X_j|c_i)$$

- Factorization!
- The naïve Bayes' classifier:

$$C^{predict}=\underset{c_i}{argmax}\, P(C=c_i)\prod_{j=1}^{M} P(X_j|c_i)$$

Belief Network

- Our naïve Bayes' classifier can be represented as a Belief network.



Why is that Naïve?

- The symptoms probably *aren't* independent given the disease.
- Assuming they are allows us to classify based on thousands of attributes.
- This seems to work pretty well in practice.

An Note on Implementation

- if M is largish this product can get really small. Too small.

$$C^{predict} = \underset{c_i}{argmax}\, P(C = c_i) \prod_{j=1}^{M} P(X_j | c_i)$$

- Solution:

$$C^{predict} = \underset{c_i}{argmax} \left(\log P(C = c_i) + \sum_{j=1}^{M} \log P(X_j | c_i) \right)$$

- Remember that log(ab) = log(a) + log(b)