Naïve Bayesian Categorization

- If we assume features of an instance are independent **given the category** (***conditionally independent***).

$$P(X \mid Y) = P(X_1, X_2, \cdots X_n \mid Y) = \prod_{i=1}^{n} P(X_i \mid Y)$$

- Therefore, we then only need to know P(X_i | Y) for each possible pair of a feature-value and a category.
- If Y and all X_i and binary, this requires specifying only $2n$ parameters:
  - P(X_i=true | Y=true) and P(X_i=true | Y=false) for each X_i
  - P(X_i=false | Y) = 1 – P(X_i=true | Y)

- Compared to specifying 2^n parameters without any independence assumptions.

Naïve Bayes Example

Probability	positive	negative
P(Y)	0.5	0.5
P(small | Y)	0.4	0.4
P(medium | Y)	0.1	0.2
P(large | Y)	0.5	0.4
P(red | Y)	0.9	0.3
P(blue | Y)	0.05	0.3
P(green | Y)	0.05	0.4
P(square | Y)	0.05	0.4
P(triangle | Y)	0.05	0.3
P(circle | Y)	0.9	0.3

Test Instance:
<medium, red, circle>

P(+ | X) = ?
P(- | X) = ?

Naïve Bayes Example

Probability	positive	negative
P(Y)	0.5	0.5
P(medium | Y)	0.1	0.2
P(red | Y)	0.9	0.3
P(circle | Y)	0.9	0.3

Test Instance:
<medium, red, circle>

P(positive | X) = P(positive)*P(medium | positive)*P(red | positive)*P(circle | positive) / P(X)
0.5 * 0.1 * 0.9 * 0.9
= 0.0405 / P(X) = 0.0405 / 0.0495 = 0.8181

P(negative | X) = P(negative)*P(medium | negative)*P(red | negative)*P(circle | negative) / P(X)
0.5 * 0.2 * 0.3 * 0.3
= 0.009 / P(X) = 0.009 / 0.0495 = 0.1818

P(positive | X) + P(negative | X) = 0.0405 / P(X) + 0.009 / P(X) = 1

P(X) = (0.0405 + 0.009) = 0.0495

Probability Estimation Example

Ex	Size	Color	Shape	Category
1	small	red	circle	positive
2	large	red	circle	positive
3	small	red	triangle	negitive
4	large	blue	circle	negitive

Probability	positive	negative
P(Y)	0.5	0.5
P(small | Y)	0.5	0.5
P(medium | Y)	0.0	0.0
P(large | Y)	0.5	0.5
P(red | Y)	1.0	0.5
P(blue | Y)	0.0	0.5
P(green | Y)	0.0	0.0
P(square | Y)	0.0	0.0
P(triangle | Y)	0.0	0.5
P(circle | Y)	1.0	0.5

Test Instance X:
<medium, red, circle>

P(positive | X) = 0.5 * 0.0 * 1.0 * 1.0 / P(X) = 0

P(negative | X) = 0.5 * 0.0 * 0.5 * 0.5 / P(X) = 0

Estimating Probabilities

- Normally, probabilities are estimated based on observed frequencies in the training data.
- If D contains n_k examples in category y_k, and n_{ijk} of these n_k examples have the jth value for feature X_i, x_{ij}, then:

$$P(X_i = x_{ij} \mid Y = y_k) = \frac{n_{ijk}}{n_k}$$

- However, estimating such probabilities from small training sets is error-prone.
- If due only to chance, a rare feature, X_i, is always false in the training data, $\forall y_k$:$P(X_i=\text{true} \mid Y=y_k) = 0$.
- If X_i=true then occurs in a test example, X, the result is that $\forall y_k$: $P(X \mid Y=y_k) = 0$ and $\forall y_k$: $P(Y=y_k \mid X) = 0$

Smoothing

- To account for estimation from small samples, probability estimates are adjusted or *smoothed*.
- Laplace smoothing using an *m*-estimate assumes that each feature is given a prior probability, *p*, that is assumed to have been previously observed in a "virtual" sample of size *m*.

$$P(X_i = x_{ij} \mid Y = y_k) = \frac{n_{ijk} + mp}{n_k + m}$$

- For binary features, *p* is simply assumed to be 0.5.

Laplace Smothing Example

- Assume training set contains 10 positive examples:
  - 4: small
  - 0: medium
  - 6: large
- Estimate parameters as follows (if m=1, p=1/3)
  - P(small | positive) = (4 + 1/3) / (10 + 1) = 0.394
  - P(medium | positive) = (0 + 1/3) / (10 + 1) = 0.03
  - P(large | positive) = (6 + 1/3) / (10 + 1) = 0.576
  - P(small or medium or large | positive) = 1.0

Comments on Naïve Bayes

- Tends to work well despite strong assumption of conditional independence.
- Experiments show it to be quite competitive with other classification methods on standard UCI datasets.
- Although it does not produce accurate probability estimates when its independence assumptions are violated, it may still pick the correct maximum-probability class in many cases.
  - Able to learn conjunctive concepts in any case
- Does not perform any search of the hypothesis space. Directly constructs a hypothesis from parameter estimates that are easily calculated from the training data.
  - Strong bias
- Not guaranteed to be consistent with training data.
- Typically handles noise well since it does not even focus on completely fitting the training data.

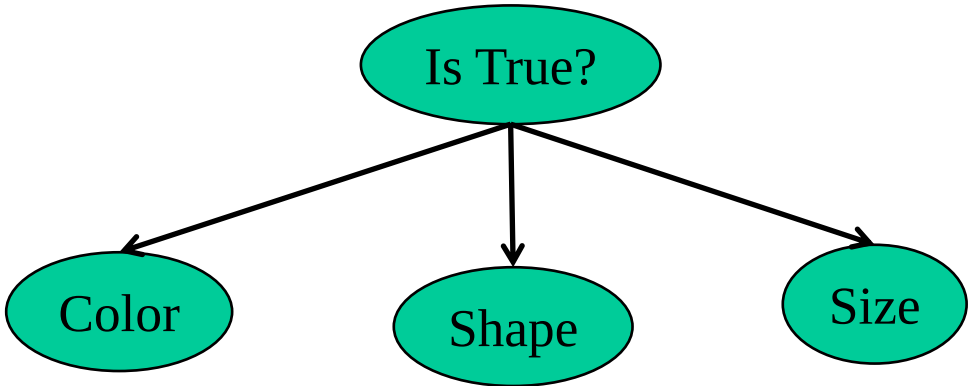
Graphical Models

- If no assumption of independence is made, then an exponential number of parameters must be estimated for sound probabilistic inference.
- No realistic amount of training data is sufficient to estimate so many parameters.
- If a blanket assumption of conditional independence is made, efficient training and inference is possible, but such a strong assumption is rarely warranted.
- **Graphical models** use directed or undirected graphs over a set of random variables to explicitly specify variable dependencies and allow for less restrictive independence assumptions while limiting the number of parameters that must be estimated.
  - **Bayesian Networks**: Directed acyclic graphs that indicate causal structure.
  - **Markov Networks**: Undirected graphs that capture general dependencies.

Bayesian Network

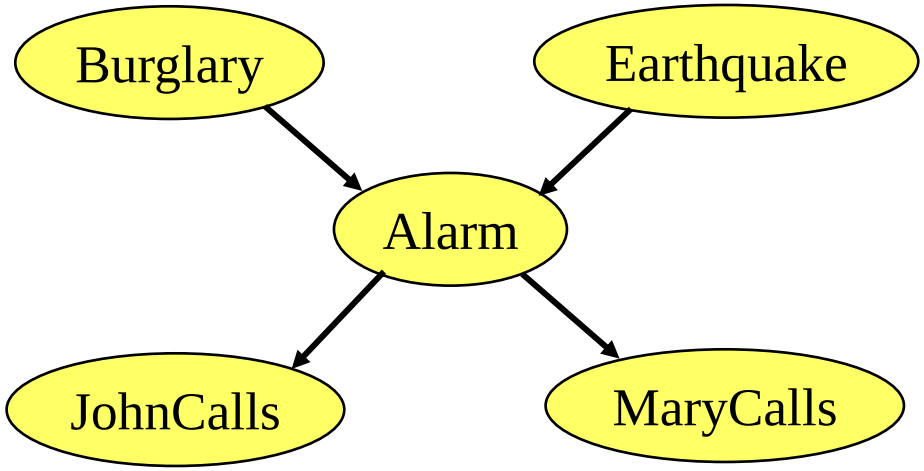
- Also: belief network, probabilistic network, causal network, and knowledge map

- Node = random variable
- DAG
- Each node has $P(X_i \mid Parents(X_i))$



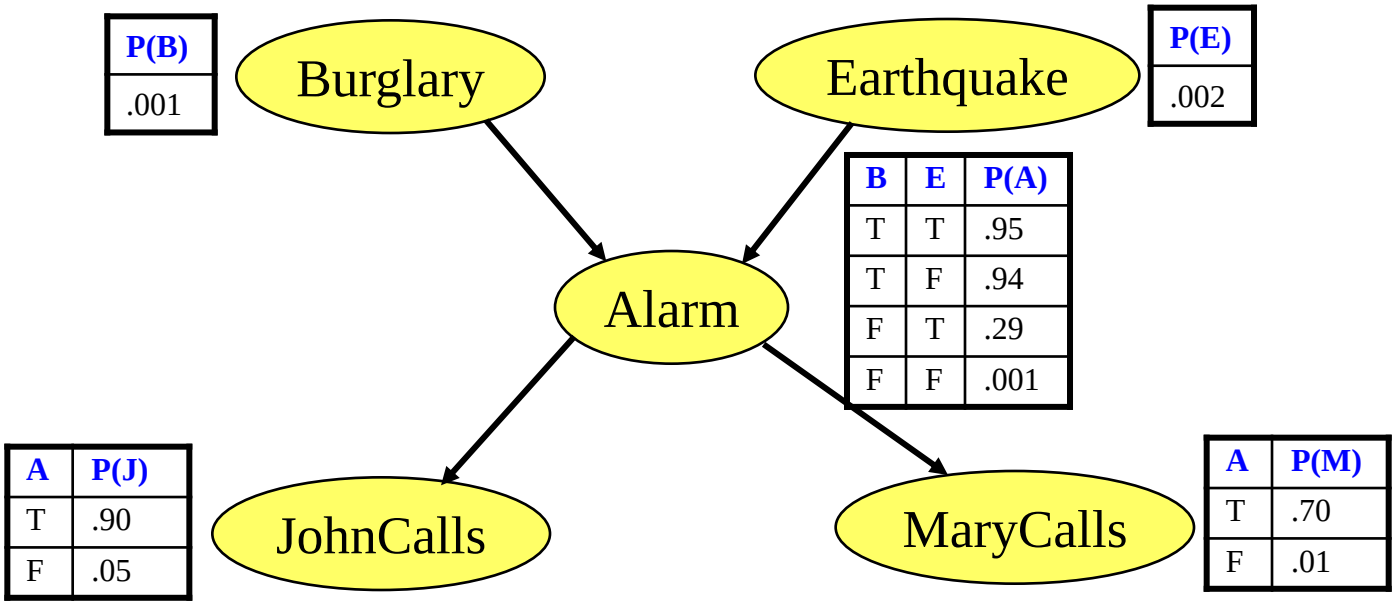
Earthquake! (Judea Perl)

- Alarm may go off, depending on burglary and earthquake
- If alarm goes off, John and/or Mary may call
- Ignores a lot of complexities: actually ∞ set of circumstances…



CPTs

- Each node has a **conditional probability table** (**CPT**) that gives the probability of each of its values given every possible combination of values for its parents (conditioning case).
  - Roots (sources) of the DAG that have no parents are given prior 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(J)
T	.90
F	.05

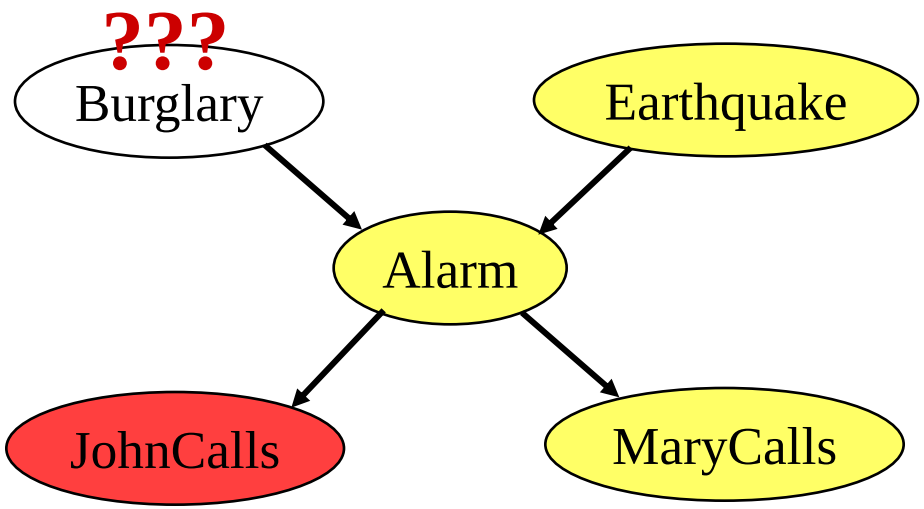
A	P(M)
T	.70
F	.01

CPT comments

- Probability of false not given since rows must add to 1.
- Example requires 10 parameters rather than $2^5-1=31$ for specifying the full joint distribution.
- Number of parameters in the CPT for a node is exponential in the number of parents.

Bayes Net Inference

- Given known values for some **evidence variables**, determine the posterior probability of some **query variables**.
- Example: Given that John calls, what is the probability that there is a Burglary?

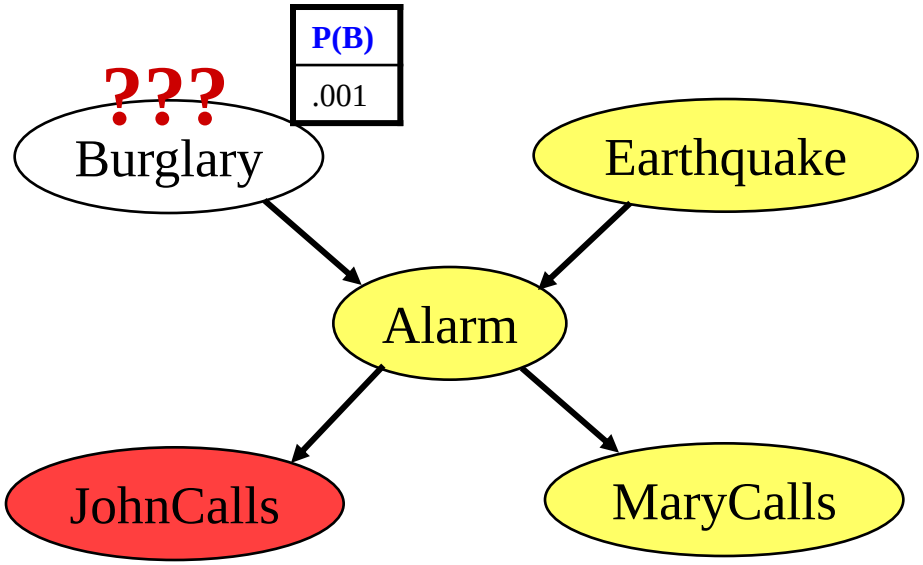


John calls 90% of the time there is an Alarm and the Alarm detects 94% of Burglaries so people generally think it should be fairly high.

However, this ignores the prior probability of John calling.

Bayes Net Inference

- Example: Given that John calls, what is the probability that there is a Burglary?



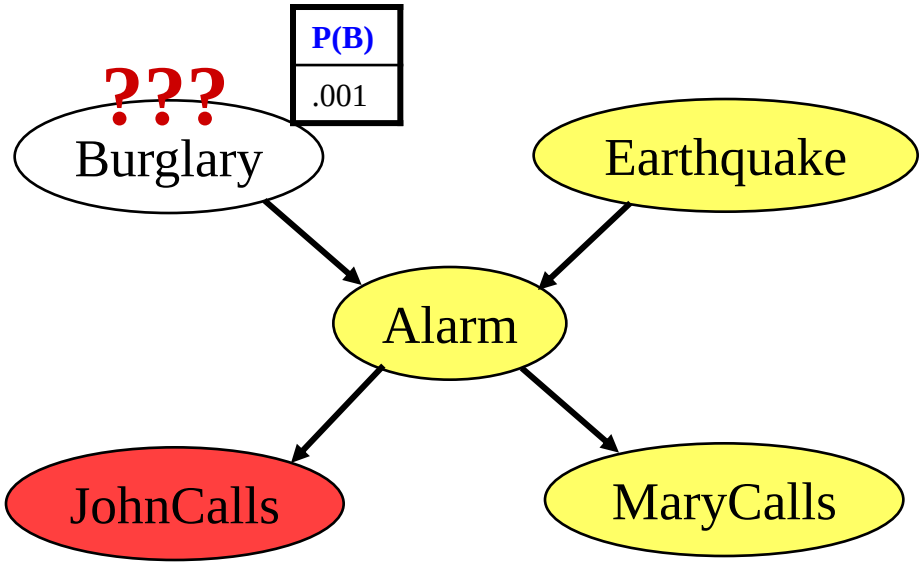
P(B)
.001

A	P(J)
T	.90
F	.05

John also calls 5% of the time when there is no Alarm. So over 1,000 days we expect 1 Burglary and John will probably call. However, he will also call with a false report 50 times on average. So the call is about 50 times more likely a false report: P(Burglary | JohnCalls) ≈ 0.02

Bayes Net Inference

- Example: Given that John calls, what is the probability that there is a Burglary?



P(B)
.001

A	P(J)
T	.90
F	.05

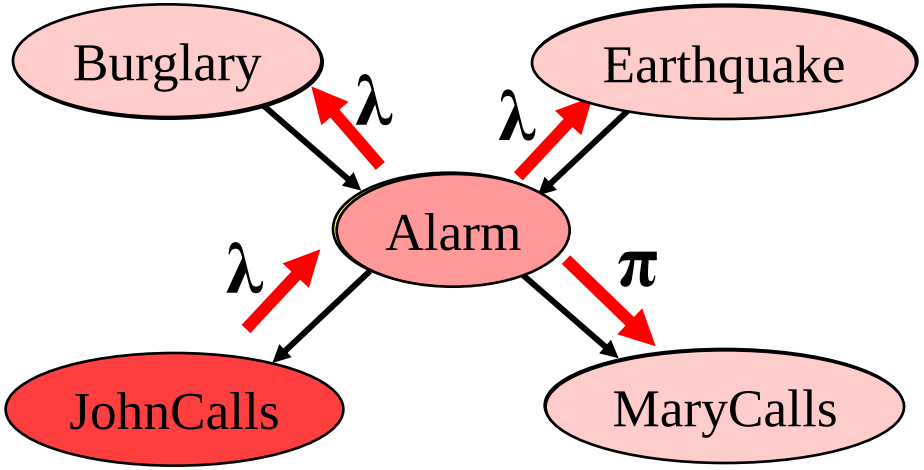
Actual probability of Burglary is 0.016 since the alarm is not perfect (an Earthquake could have set it off or it could have gone off on its own). On the other side, even if there was not an alarm and John called incorrectly, there could have been an undetected Burglary anyway, but this is unlikely.

Complexity of Bayes Net Inference

- In general, the problem of Bayes Net inference is NP-hard (exponential in the size of the graph).
- For **singly-connected networks** or **polytrees** in which there are no undirected loops, there are linear-time algorithms based on **belief propagation**.
  - Each node sends local evidence messages to their children and parents.
  - Each node updates belief in each of its possible values based on incoming messages from it neighbors and propagates evidence on to its neighbors.
- There are approximations to inference for general networks based on **loopy belief propagation** that iteratively refines probabilities that converge to accurate values in the limit.

Belief Propagation Example

- λ messages are sent from children to parents representing abductive evidence for a node.
- π messages are sent from parents to children representing causal evidence for a node.



Learning Graphical Models

- **Structure Learning**: Learn the graphical structure of the network.
- **Parameter Learning**: Learn the real-valued parameters of the network
  - CPTs for Bayes Nets

Structure Learning

- Use greedy top-down search through the space of networks, considering adding each possible edge one at a time and picking the one that maximizes a statistical evaluation metric that measures fit to the training data.
- Alternative is to test all pairs of nodes to find ones that are statistically correlated and adding edges accordingly.
- Bayes net learning requires determining the direction of causal influences.
- Special algorithms for limited graph topologies.
  - TAN (Tree Augmented Naïve-Bayes) for learning Bayes nets that are trees.

Parameter Learning

- If values for all variables are available during training, then parameter estimates can be directly estimated using frequency counts over the training data.
  - Must smooth estimates to compensate for limited training data.
- If there are hidden variables, some form of gradient descent or Expectation Maximization (EM) must be used to estimate distributions for hidden variables.
  - Like setting the weights feeding hidden units in backpropagation neural nets.