Bayesian Networks

Probabilistic Classification

- Goal:
  - Gather Labeled Training Data
  - Build/Learn a Probability Model
  - Use the model to infer class labels for unlabeled data points
- Example: Spam Filtering...

(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:

$$P(spam) = ??$$

$$P(\neg spam) = ??$$

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

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

(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:

$P(spam) = 8000/8800 \approx .91$

$P(\neg spam) \approx .09$

$P(\neg viagra, discount, cs444 | spam) = 8/8000 = .001$

$P(\neg viagra, discount, cs444 | \neg spam) = 6/800 = .0075$

Aside: Maximum Likelihood Learning

- This approach makes sense... What is the justification?
- This is the maximum likelihood estimate:

$$\hat{\phi} = \underset{\phi \in \Phi}{argmax}\, P(observed\ data | \phi)$$

- Where ϕ represents the parameters we are trying to learn.

Bayes' Optimal Classifier I
(No Independence Assumptions)

- 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 (?):

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

Bayes' Optimal Classifier II
(No Independence Assumptions)

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

- Apply Bayes' rule

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

- Conditioning (applying law of total probability):

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

An Aside: MAP vs. ML

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

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

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

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

Bayes' Optimal Classifier III
(No Independence Assumptions)

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

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

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

- If you have the true distributions, this is the <u>best</u> choice.

Bayes' Optimal Classifier Example

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

- Email contains "discount", "CS444", but not "viagra". Is it spam?
- Recall: $P(spam) \approx .91$

  $P(\neg spam) \approx .9$

  $P(\neg viagra, discount, cs444 | spam) = .001$

  $P(\neg viagra, discount, cs444 | \neg spam) = .0075$

Bayes' Optimal Classifier Example

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

- Email contains "discount", "CS444", but not "viagra". Is it spam?
- Recall: $P(spam) \approx .91$

$$P(\neg spam) \approx .09$$

$$P(\neg viagra, discount, cs444 | spam) = .001$$

$$P(\neg viagra, discount, cs444 | \neg spam) = .0075$$

$$P(\neg viagra, discount, cs444 | spam) P(spam) = .001 * .91 = .00091$$

$$P(\neg viagra, discount, cs444 | \neg spam) P(\neg spam) = .0075 * .09 = .000675$$

The Problem...

- Assume we want to use *hundreds* of words.
- What is the problem here?

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

The Problem...

- Assume we want to use *hundreds* of words.
- What is the problem here?

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

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

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

(Changing Gears) Review

- Joint probability distributions are the key to probabilistic inference.
- If all N variables are completely independent, we can represent the full joint distribution using N numbers.
- If every variable depends on every other, we need to store $2^N - 1$ values
- There must be something in-between...

Conditional Independence

- Saying that

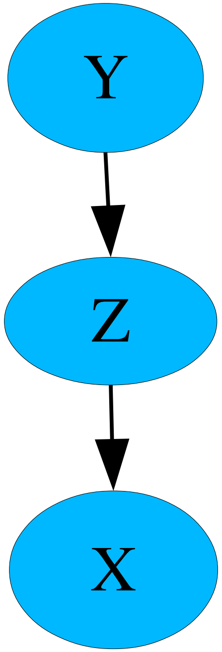
  "X is conditionally independent of Y given Z"

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

  equivalently:

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

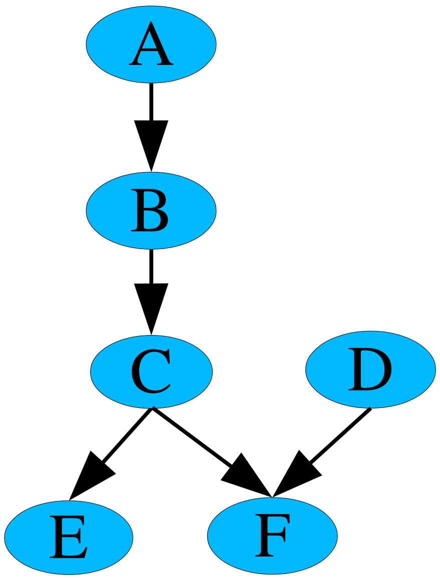
- This graph encodes the same thing:



Bayesian Networks

- Directed acyclic graphs that have the following interpretation:

  Each variable is conditionally independent of all it's non-descendants, given the value of it's parents.



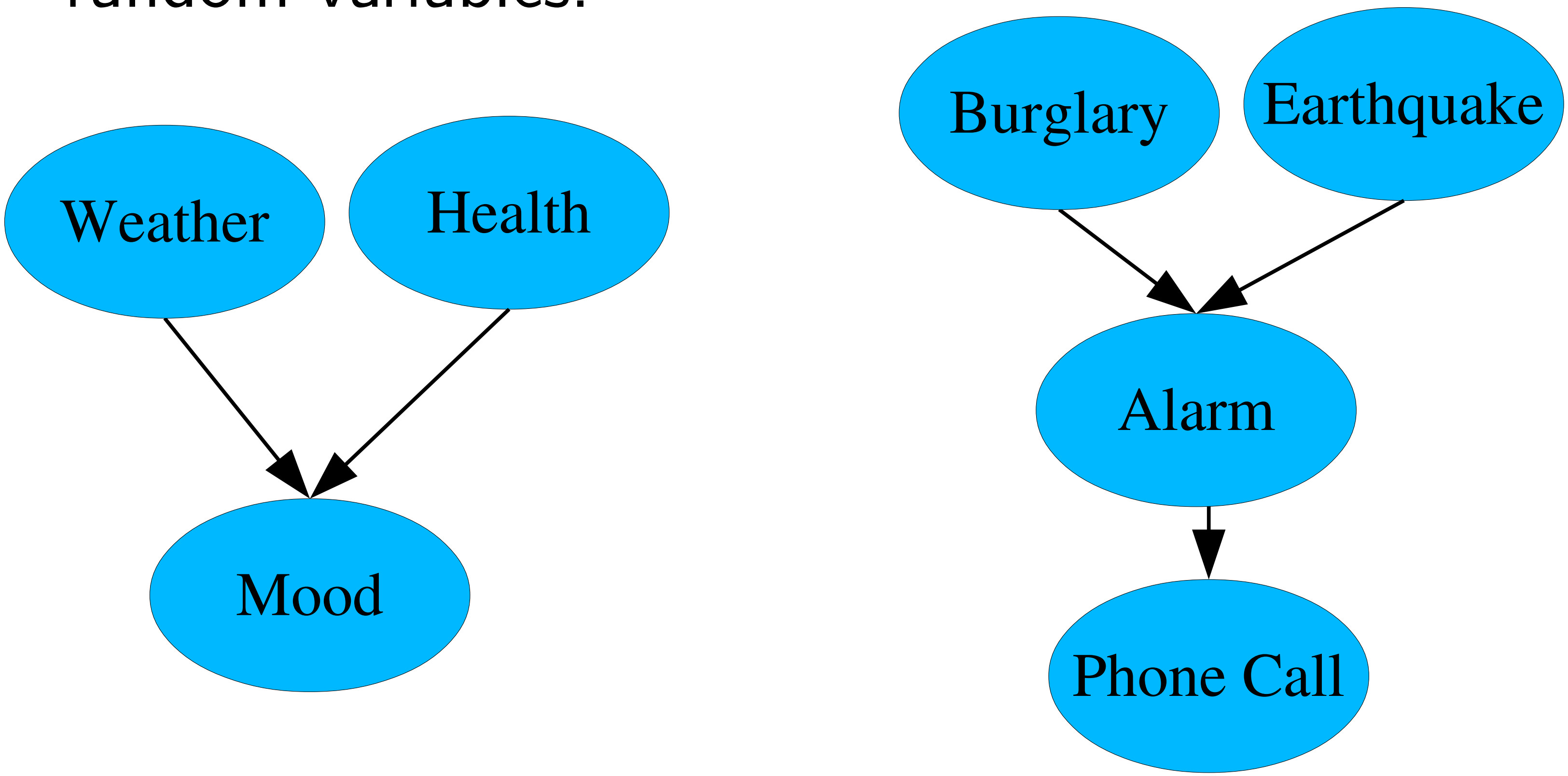
$$P(C \mid A, B, D) = P(C \mid B)$$

$$P(E \mid A, B, C, D, F) = P(E \mid C)$$

$$P(F \mid A, B, C, D, E) = P(F \mid C, D)$$

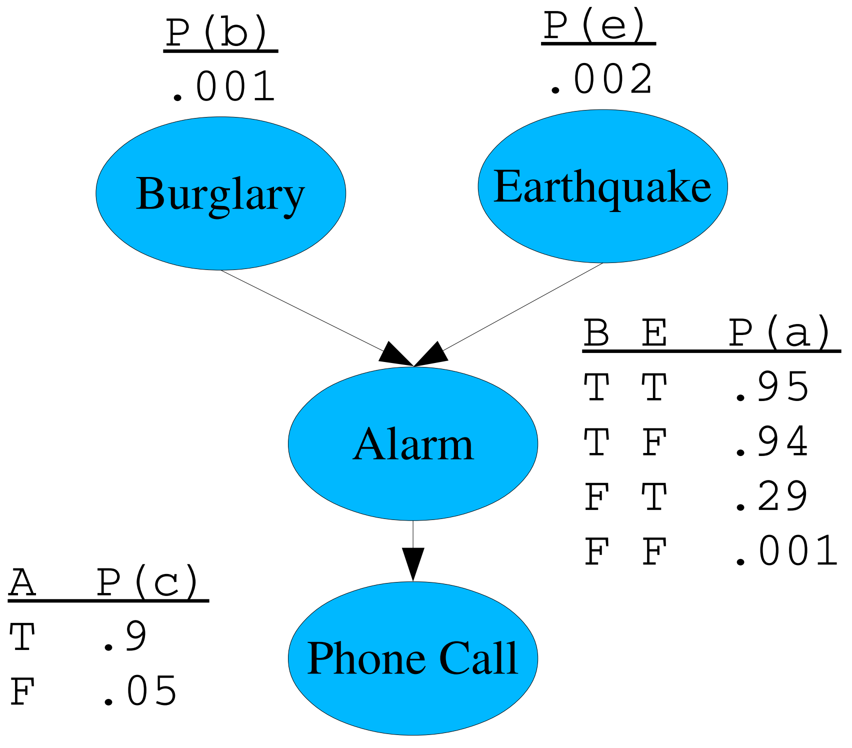
Bayesian Networks

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



Specifying a Bayes' Net

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



Bayesian Networks

- We can then reconstruct any entry from the joint distribution using:

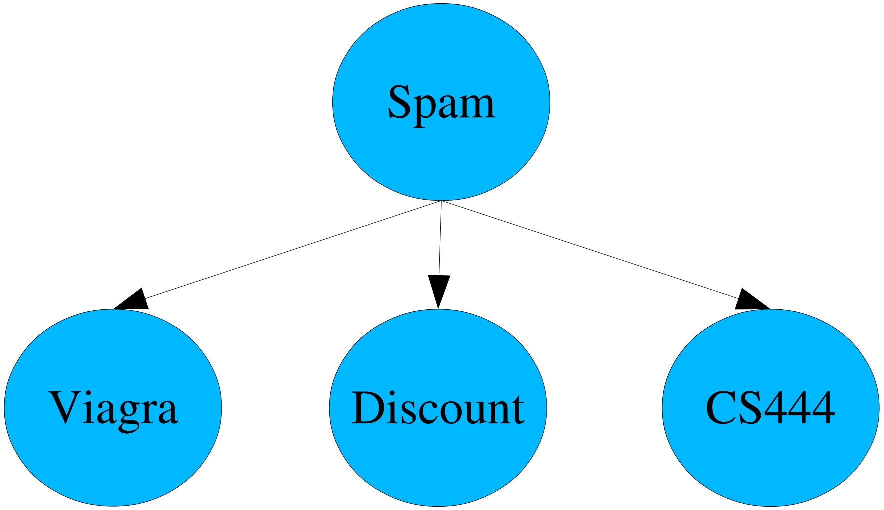
$$P(X_1, X_2, \ldots, X_N) = \prod_{i=1}^{N} P(X_i | parents(X_i))$$

- 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. $N * 2^M$

Simple Application: Naive Bayes' classifier

- Back to spam filtering...
- Assumption: spamminess impacts the probability of different words. Words are conditionally independent of each other given spam/non-spam status.
- Consider four boolean random variables:
  - Spam (message is spam)
  - Viagra (message contains word "viagra")
  - Discount (message contains word "discount")
  - CS444 (message contains word "CS444")
- What will the graphical model look like?

Belief Network



- Now we need to specify the probabilities
- How does this help with inference...

Naïve Bayes' Classifier

- Remember: If M is large, it is impossible to learn

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

- The solution (?): assuming that the X_j are independent given C:

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

- The naïve Bayes' classifier:

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

Naïve Probability Estimation

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:

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

Naïve Probability Estimation

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:

$$P(\neg viagra, discount, cs444 | spam) = \frac{6620}{8000} \times \frac{788}{8000} \times \frac{20}{8000} \approx .0002$$

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.
- Do you see any advantage of this relative to the classifiers we saw earlier?

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 $\log(ab) = \log(a) + \log(b)$