Bayesian Learning

Instructor: Jesse Davis

Announcements

- Homework 1 is due today
- Homework 2 is out
- Slides for this lecture are online
- We'll review some of homework 1 next class
  - Techniques for efficient implementation of collaborative filtering
  - Common mistakes made on rest of HW

Outline

- Probability overview
- Naïve Bayes
- Bayesian learning
- Bayesian networks

Random Variables

- A random variable is a number (or value) determined by chance
  - More formally it is drawn from a probability distribution
- Types of random variables
  - Continuous
  - Binary
  - Discrete

Why Random Variables

- Our goal is to predict a target variable
- We are not given the true function
- We are given observations
  - Number of times a dice lands on 4
  - Can estimate the probability of this event
  - We don’t know where the dice will land
  - Can only guess what is likely to happen

Bernoulli Distribution

- Bernoulli RV takes two values: 0 and 1
- Prob(1) = p and P(0) = 1 – p

$$P(x) = \begin{cases} p^x(1\text{-}p)^x, \text{ if } x = 0 \text{ or } 1 \\ 0, \text{ otherwise} \end{cases}$$

- The performance of one trial with fixed probability of success (p) is a Bernoulli trial

Binomial Distribution

- Like Bernoulli distribution, two values: 0 or 1 and probability P(1)=p and P(0)=1-p
- What is the probability of k successes, P(k), in a series of n independent trials? (n>=k)
- P(k) is a binomial random variable:

$$P(x) = \begin{bmatrix} n \\ k \end{bmatrix} p^k (1-p)^{n-k}, \text{ where } \begin{bmatrix} n \\ k \end{bmatrix} = \frac{n!}{k!(n-k)!}$$

- Bernoulli distribution is a special case of the binomial distribution (i.e., n=1)

Multinomial Distribution

- Generalizes binomial distribution to multiple outputs (classes)
- N independent trials
  - r possible outcomes
  - Each outcome c_r has $P(c_r) = p_r$
  - $\Sigma P(c_r) = 1$
- Multinomial RV: Probability that in n trials, the frequency of the r classes is $(n_1,\ldots,n_r)$

$$P(x) = \begin{bmatrix} n \\ n_1 \ldots n_r \end{bmatrix} p_1^{n1} * \ldots * p_r^{nr} \text{ , where } \begin{bmatrix} n \\ n_1 \ldots n_r \end{bmatrix} = \frac{n!}{n_1! * \ldots * n_r!}$$

Axioms of Probability Theory

Just three are enough to build entire theory!

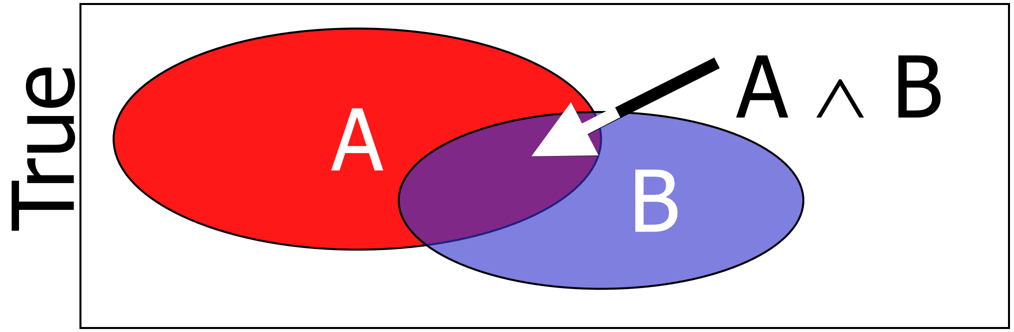
1. All probabilities between 0 and 1

   $0 \leq P(A) \leq 1$

2. P(true) = 1   and P(false) = 0

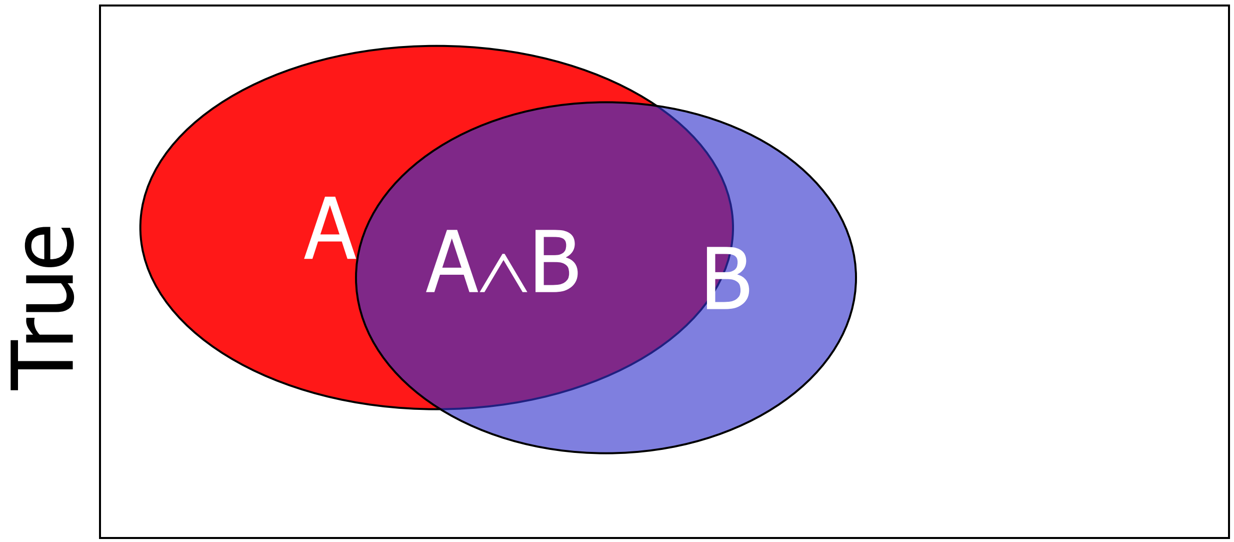
3. Probability of disjunction of events is:

   $P(A \vee B) = P(A) + P(B) - P(A \wedge B)$



Conditional Probability

- P(A | B) is the probability of *A* given *B*
- Assumes that *B* is the only info known.
- Defined as $$P(A \mid B) = \frac{P(A \wedge B)}{P(B)}$$



Independence

- A and B are independent iff:
  - $P(A \mid B) = P(A)$
  - $P(B \mid A) = P(B)$

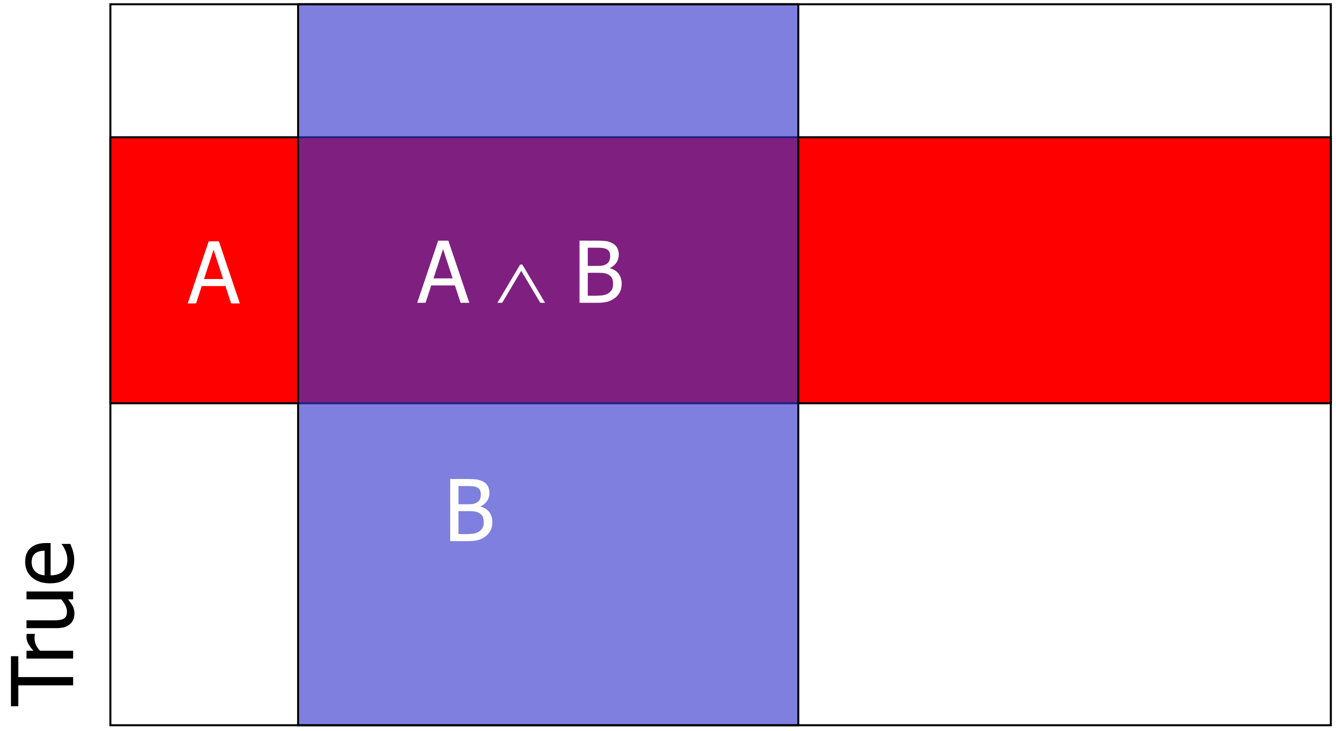
These two constraints are logically equivalent

- Therefore if A and B are independent

$$P(A \mid B) = \frac{P(A \wedge B)}{P(B)} = P(A)$$

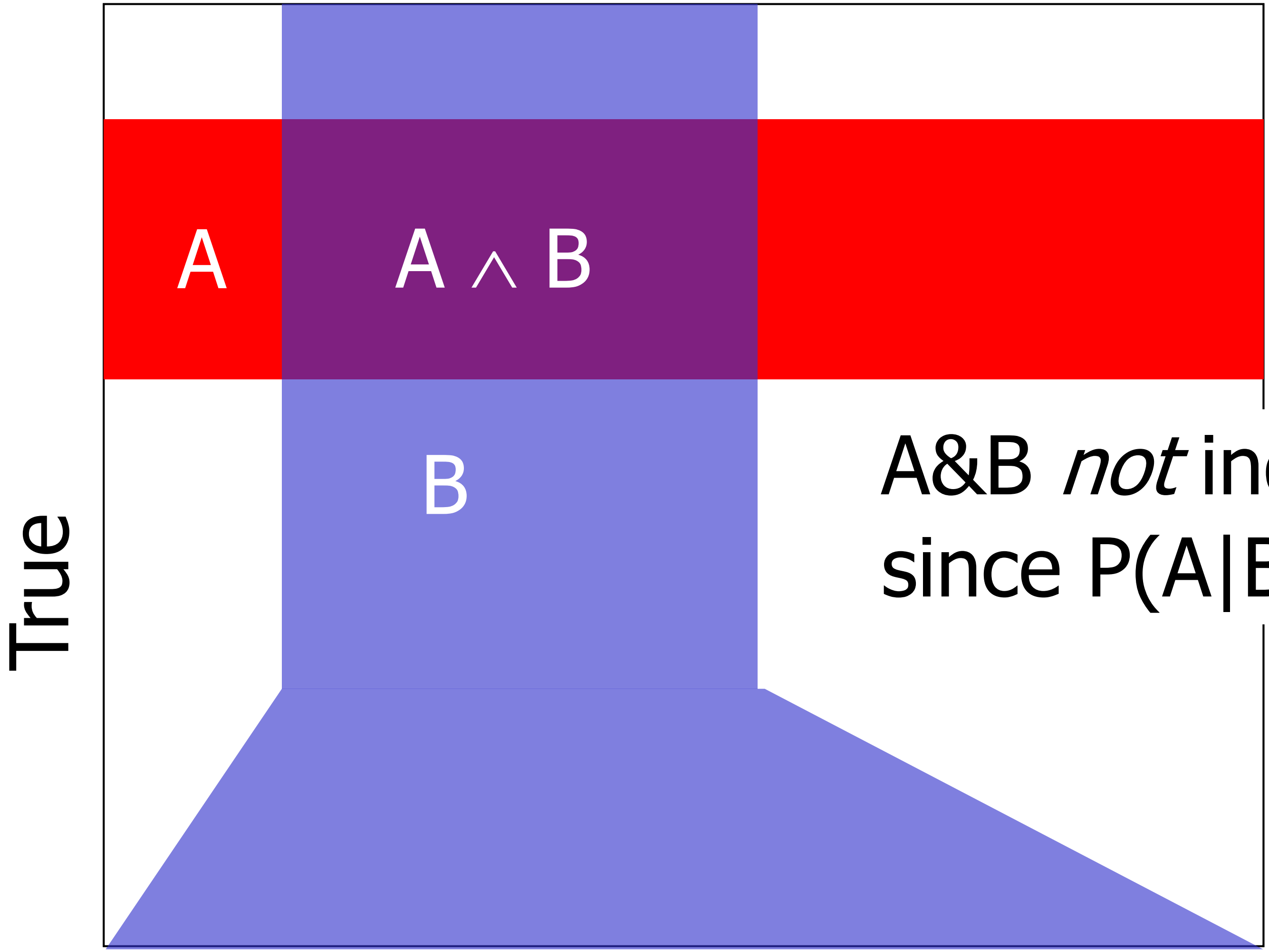
$$P(A \wedge B) = P(A)P(B)$$

Independence



Independence is powerful, but rarely holds

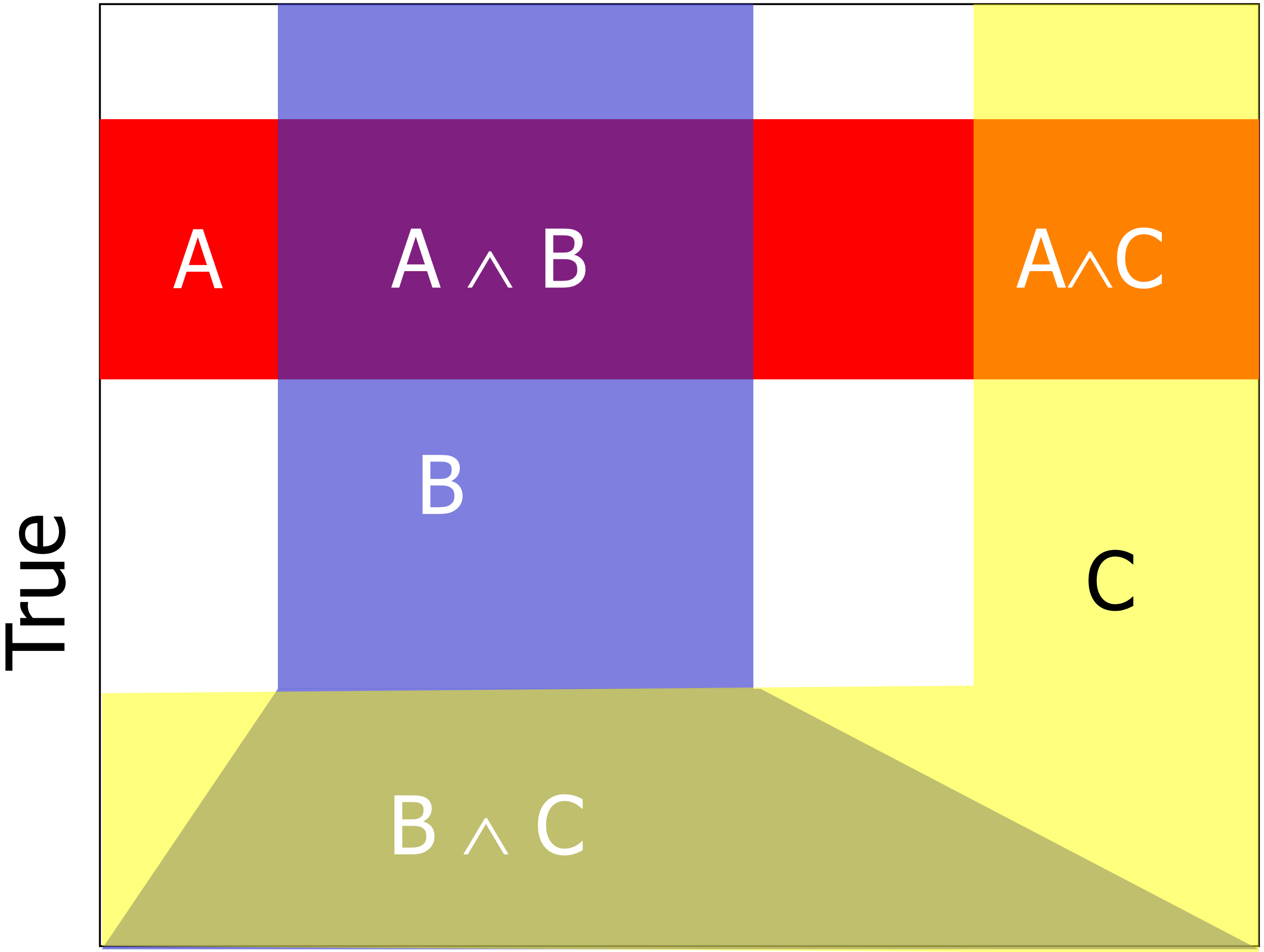
Conditional Independence



A&B *not* independent, since P(A|B) < P(A)

Conditional Independence

But: A&B are *made* independent by $\neg$C



$P(A|\neg C) = P(A|B,\neg C)$

Bayes Rule

- Bayes rule is: $P(A \mid B) = \dfrac{P(B \mid A)\, P(A)}{P(B)}$

- Proof:

$$\left.\begin{aligned} P(A \mid B) &= \frac{P(A \wedge B)}{P(B)} \\ P(B \mid A) &= \frac{P(A \wedge B)}{P(A)} \end{aligned}\right] \quad \text{Defn of cond. prob}$$

$$P(A \wedge B) = P(B \mid A)\, P(A) \qquad \text{Rearrange line 2}$$

$$P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)} \qquad \text{Sub in prev result}$$

Use to Compute Diagnostic Probability from Causal Probability

- For example, let M be meningitis, S be stiff neck
  - $P(M) = 0.0001$
  - $P(S) = 0.1$
  - $P(S|M) = 0.8$

- $P(M \mid S) = 0.8 \times 0.0001 / 0.1 = 0.0008$

- Probability is very low!

Outline

- Probability overview
- Naïve Bayes
- Bayesian learning
- Bayesian networks

Naïve Bayes: Motivation

- We will see many draws of $X_1,\ldots,X_n$ and the response (class) Y
- We want to estimate the most likely value of Y for the input, that is, $P(Y|\ X_1,\ldots,X_n)$
- What difficulty arises?
  - Exponentially many settings for $X_1,\ldots,X_n$
  - Next case probably has not been seen

One Approach: Assume Conditional Independence

- By Bayes Rule (with normalization):
  - $P(Y| X_1,\ldots,X_n) = \alpha P(X_1,\ldots,X_n|Y)P(Y)$
  - Normalization: Compute above for each value of Y, then normalize so sum to 1

- Recall Conditional independence: $P(X_1,\ldots,X_n|Y) = P(X_1|Y)\ldots P(X_n|Y)$

- $P(Y| X_1,\ldots,X_n)= \alpha P(X_1|Y)\ldots P(X_n|Y)P(Y)$

Naïve Bayes

- Assumes (naïvely) that all features are conditionally independent given the class
  - P(A ^ B | Class) = P(A | Class) * P(B | Class)
  - Avoids estimating P(A ^ B), etc.

- Surprisingly, though the assumption is often violated naïve Bayes works well in practice
  - Bag of words for text, spam filtering, etc.

Naïve Bayes in Practice

- Empirically, estimates relative probabilities more reliably than absolute ones:

$$\frac{P(\text{Pos} \mid \text{Features})}{P(\text{Neg} \mid \text{Features})} = \frac{P(\text{Features} \mid \text{Pos}) * P(\text{Pos})}{P(\text{Features} \mid \text{Neg}) * P(\text{Neg})}$$

- Better than

$$P(\text{Pos} \mid \text{Features}) = P(\text{Features} \mid \text{Pos}) * P(\text{Pos})$$

- Naïve Bayes tends to push probability estimates towards either 0 or 1

Technical Detail: Underflow

- Assume we have 100 features
  - We multiple 100 numbers in [0,1]
  - If values are small, we are likely to ‘underflow’ the min positive/float value
- Solution: $\prod \text{probs} = e^{\sum \log(\text{prob})}$
- Sum log’s of prob’s
- Subtract logs since $\log \frac{P(+)}{P(-)} = \log P(+) - \log P(-)$

Log Odds

$$\text{Odds} = \frac{P(F_1 \mid Pos) * \ldots * P(F_n \mid Pos) * P(Pos)}{P(F_1 \mid Neg) * \ldots * P(F_n \mid Neg) * P(Neg)}$$

$$\log(\text{Odds}) = [\Sigma \log\{ P(F_i \mid Pos) / P(F_i \mid Neg)\}] + \log(P(Pos) / P(Neg))$$

Notice if a feature value is more likely in a pos, the log is pos and if more likely in neg, the log is neg (0 if tie)

Naïve Bayes Example

Color	Shape	Size	Category
red	•	big	+
blue	Δ	small	+
red	□	small	+
red	Δ	big	–
blue	•	small	–
red	Δ	small	?

Naïve Bayes Example

- For the new example (red, Δ, small)

$$\frac{P(+|F's)}{P(-\ |F's)} = \frac{P(red|+)*P(\Delta|+)*P(small|+)*P(+)}{P(red|-)*P(\Delta|\ -)*P(small|\ -)*P(-)}$$

$$= \frac{2/3 * 1/3 * 2/3 * 3/5}{1/2 * 1/2 * 1/2 * 2/5} = 1.77$$

- So most likely a POS example

Dealing with Zeroes (and Small Samples)

- If we never see something (eg, in the train set), should we assume its probability is zero?

- If we only see 3 pos ex's and 2 are red, do we really think

$$P(\text{red}|\text{pos}) = 2/3 \ \ ?$$

M-estimates
(Eq 6.22 in Mitchell; Eq 7 in draft chapter)

- Imagine we had *m* hypothetical pos ex's
- Assume *p* is prob these examples are red
- Improved estimate:

$$P(\text{red} \mid \text{pos}) = \frac{2 + p * m}{3 + m}$$

(In general, red is some feature value and 2 and 3 are actual counts in the training set)

M-Estimate

$$\text{Prob} = \frac{n_c + m\,p}{n + m}$$

of $f_i = v_i$ examples

of actual examples

Equivalent sample size used in guess

Prior guess

Example: Of 10 examples, 8 have color = red

$$\text{Prob (color=red)} = \frac{8 + 100 \times 0.5}{10 + 100} = \frac{58}{110} = 0.53$$

M-Estimates More Generally

$$P(f_i=v_i) = \frac{\#\text{ times } f_i = v_i + \text{Equivalent sample size} \times \text{initial guess for } P(f_i = v_i)}{\#\text{ train ex's} + \text{Equivalent sample size}}$$

m

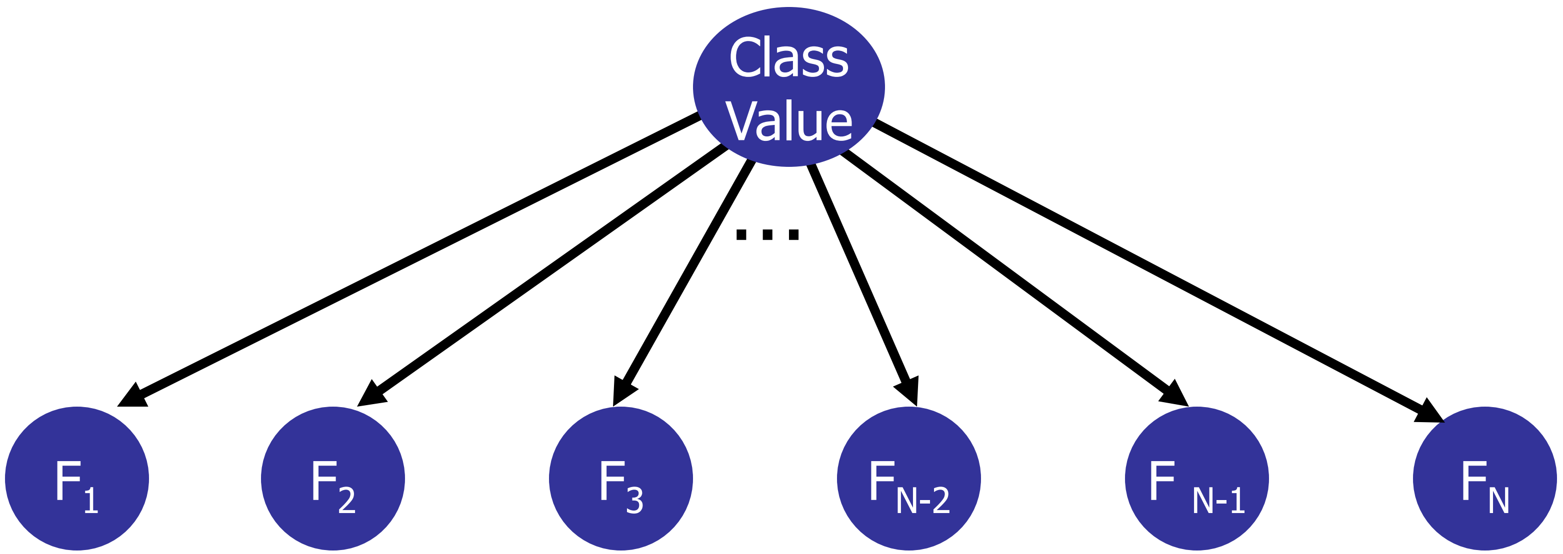
Estimate based on data

Estimate based on prior knowledge ("priors")

Laplace Smoothing

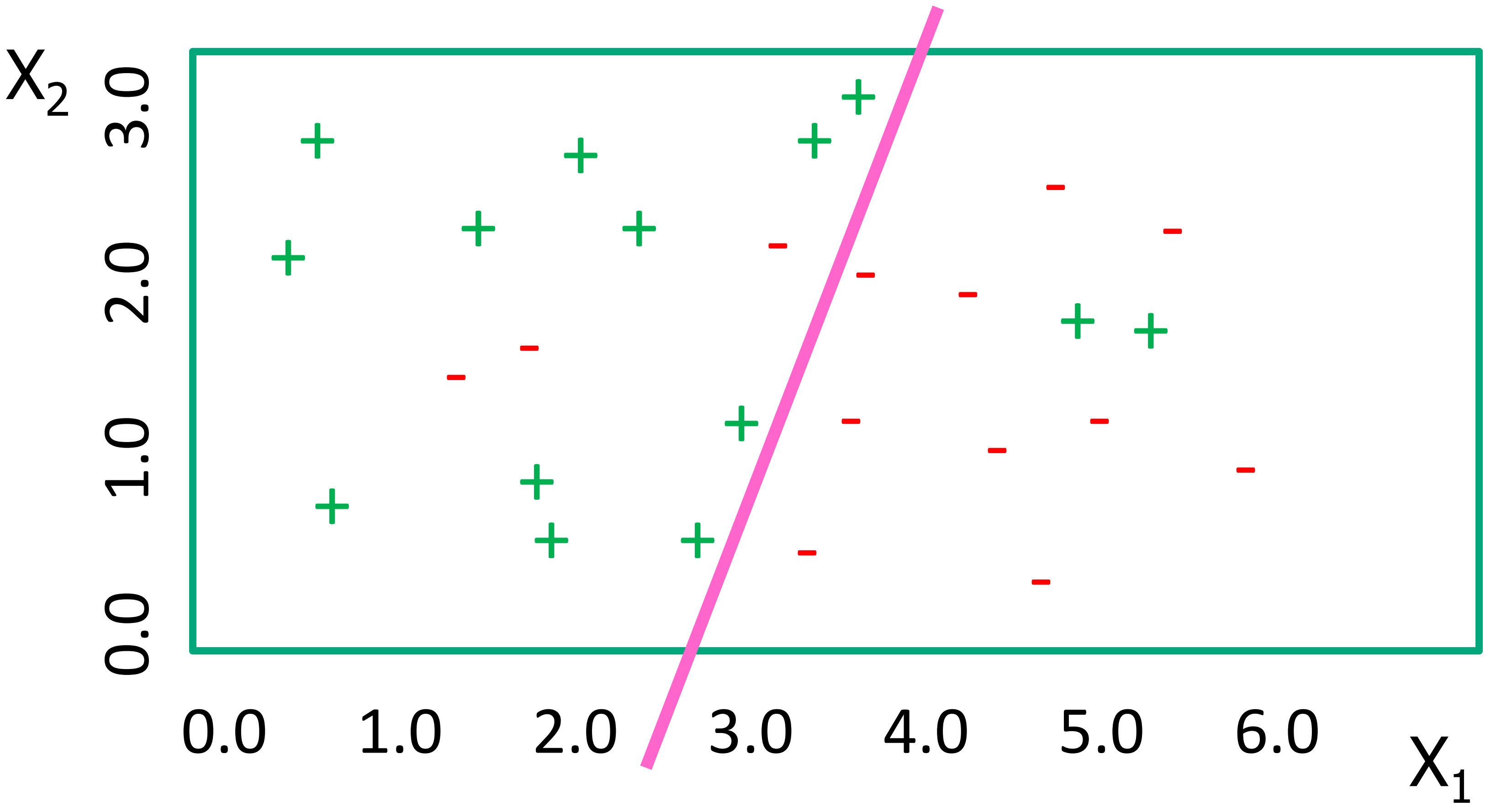
- Special case of *m* estimates
  - Let *m = #colors, p = 1/m*
  - Ie, assume one hypothetical pos ex of each color
- Implementation trick
  - Start all counters at 1 instead of 0
    - Eg, initialize count(pos, feature(i), value(i, j)) = 1
    - count(pos, color, red),
count(neg, color, red),
count(pos, color, blue),
…

Naïve Bayes as a Graphical Model



Node *i* stores $P(F_i \mid POS)$ and $P(F_i \mid NEG)$

How Does Naïve Bayes Partition Feature Space?



Homework: Spam Filtering

- Task:

From: Branded anti-ED Pills <otubu9068@telesp.net.br>
To: andrey.kolobov@gmail.com
Date: Fri, Apr 2, 2010 at 7:23 PM
Subject: Hot Sale, andrey.kolobov! 77% off on top goods Emen
Mailed-by: telesp.net.br

Why aren't you on our site, andrey.kolobov? We have 77% off today!!

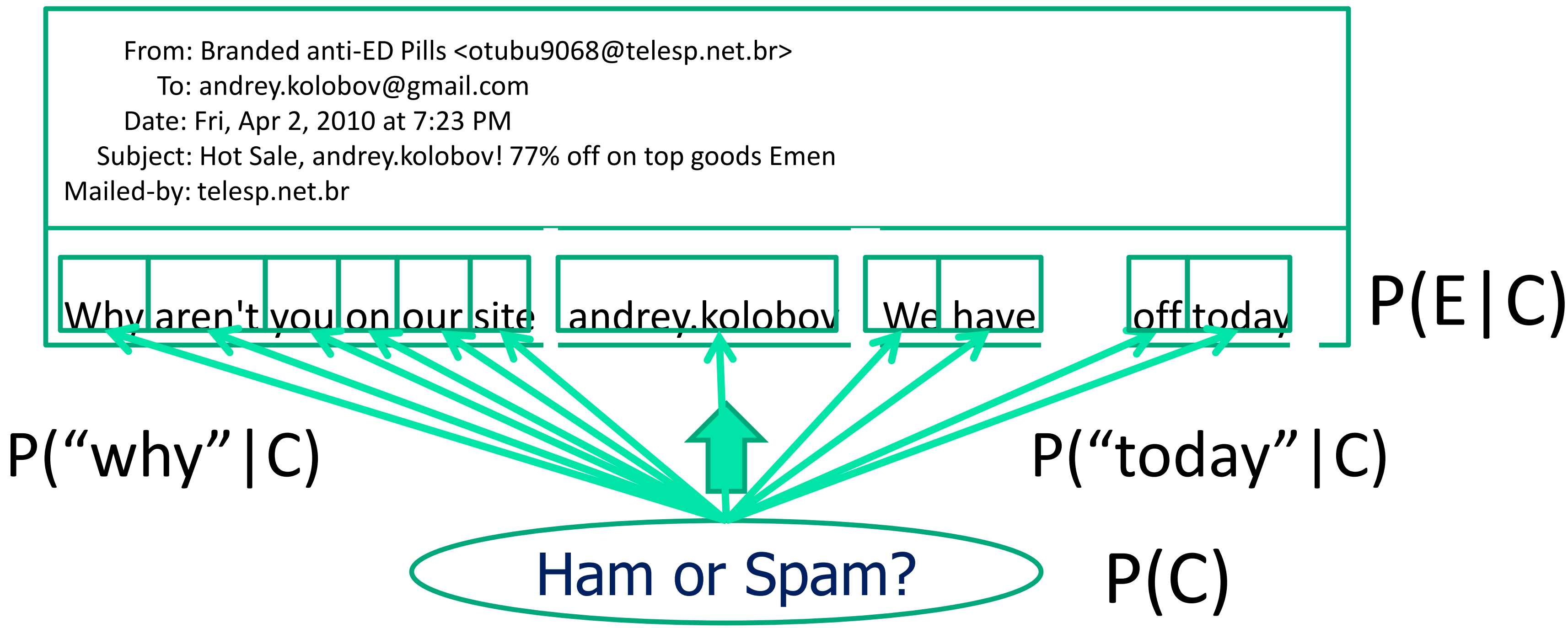
$P(E|C)$

Ham or Spam?

$P(C)$

- $P(C|E) = P(E|C)P(C) / P(E)$
- $C \leftarrow \underset{C \in \{h, m\}}{\operatorname{argmax}} \{ P(E|C)P(C) \}$

… with Naïve Bayes



$$C \leftarrow \operatorname*{argmax}_{C \in \{h, m\}} \{P(E|C)P(C)\} = \operatorname*{argmax}_{C \in \{h, m\}} \{P(C) \prod_{W \in E} P(W|C)\}$$

Estimating Parameters

- Given:
  - Set of training spam emails S
  - Set of training ham emails H
- Probabilities:
  - $P(w|c) = (1 + (\#_c w)) / (\sum_{w' \text{ in } V} (1 + \#_c w'))$
  - $P(c) = |c| / (|S| + |H|)$

To avoid getting P(w|c) = 0 due to data sparsity

Naïve Bayes Summary

- Fast, simple algorithm
- Effective in practice [good baseline comparison]
- Gives estimates of confidence in class label
- Makes simplifying assumptions
- Extensions to come…

Outline

- Probability overview
- Naïve Bayes
- Bayesian learning
- Bayesian networks

Coin Flip

C_1 C_2 C_3

$P(H|C_1) = 0.1$ $P(H|C_2) = 0.5$ $P(H|C_3) = 0.9$

Which coin will I use?

$P(C_1) = 1/3$ $P(C_2) = 1/3$ $P(C_3) = 1/3$

Prior: Probability of a hypothesis before we make any observations

Coin Flip

C_1

C_2

C_3

$P(H|C_1) = 0.1$ $P(H|C_2) = 0.5$ $P(H|C_3) = 0.9$

Which coin will I use?

$P(C_1) = 1/3$ $P(C_2) = 1/3$ $P(C_3) = 1/3$

Uniform Prior: All hypothesis are equally likely before we make any observations

Experiment 1: Heads

Which coin <u>did</u> I use?

$P(C_1|H) = ?$ $P(C_2|H) = ?$ $P(C_3|H) = ?$

$$P(C_1|H) = \frac{\boxed{P(H|C_1)}\boxed{P(C_1)}}{\boxed{P(H)}} \qquad \boxed{P(H) = \sum_{i=1}^{3} P(H|C_i)P(C_i)}$$

C_1

$\boxed{P(H|C_1)=0.1}$

$\boxed{P(C_1)=1/3}$

C_2

$P(H|C_2) = 0.5$

$P(C_2) = 1/3$

C_3

$P(H|C_3) = 0.9$

$P(C_3) = 1/3$

Experiment 1: Heads

Which coin <u>did</u> I use?

$P(C_1|H) = 0.066$ $P(C_2|H) = 0.333$ $P(C_3|H) = 0.6$

Posterior: Probability of a hypothesis given data

C_1

$P(H|C_1) = 0.1$

$P(C_1) = 1/3$

C_2

$P(H|C_2) = 0.5$

$P(C_2) = 1/3$

C_3

$P(H|C_3) = 0.9$

$P(C_3) = 1/3$

Terminology

- **Prior:** Probability of a hypothesis before we see any data
- **Uniform prior:** A prior that makes all hypothesis equally likely
- **Posterior:** Probability of hypothesis after we saw some data
- **Likelihood:** Probability of the data given the hypothesis

Experiment 2: Tails

Which coin <u>did</u> I use?

$P(C_1|HT) = ?$ $P(C_2|HT) = ?$ $P(C_3|HT) = ?$

$$P(C_1|HT) = \alpha P(HT|C_1)P(C_1) = \alpha P(H|C_1)P(T|C_1)P(C_1)$$

C_1

$P(H|C_1) = 0.1$

$P(C_1) = 1/3$

C_2

$P(H|C_2) = 0.5$

$P(C_2) = 1/3$

C_3

$P(H|C_3) = 0.9$

$P(C_3) = 1/3$

Experiment 2: Tails

Which coin <u>did</u> I use?

$P(C_1|HT) =$ 0.21 $P(C_2|HT) =$ 0.58 $P(C_3|HT) =$ 0.21

$$P(C_1|HT) = \alpha P(HT|C_1)P(C_1) = \alpha P(H|C_1)P(T|C_1)P(C_1)$$

C_1

$P(H|C_1) = 0.1$

$P(C_1) = 1/3$

C_2

$P(H|C_2) = 0.5$

$P(C_2) = 1/3$

C_3

$P(H|C_3) = 0.9$

$P(C_3) = 1/3$

Experiment 2: Tails

Which coin did I use?

$P(C_1|HT) = 0.21$ $P(C_2|HT) = 0.58$ $P(C_3|HT) = 0.21$

C_1

$P(H|C_1) = 0.1$

$P(C_1) = 1/3$

C_2

$P(H|C_2) = 0.5$

$P(C_2) = 1/3$

C_3

$P(H|C_3) = 0.9$

$P(C_3) = 1/3$

Your Estimate?

What is the probability of heads after two experiments?

Most likely coin:

C_2

Best estimate for P(H)

$P(H|C_2) = 0.5$

C_1

$P(H|C_1) = 0.1$

$P(C_1) = 1/3$

C_2

$P(H|C_2) = 0.5$

$P(C_2) = 1/3$

C_3

$P(H|C_3) = 0.9$

$P(C_3) = 1/3$

Your Estimate?

Maximum Likelihood Estimate: The best hypothesis that fits observed data assuming uniform prior

Most likely coin:

C_2 

Best estimate for P(H)

$P(H|C_2) = 0.5$

C_2

$P(H|C_2) = 0.5$

$P(C_2) = 1/3$

Using Prior Knowledge

- Should we always use a ***Uniform Prior*** ?
- Background knowledge:
  - Heads => we have take-home midterm
  - Jesse likes take-homes…
  - => Jesse is more likely to use a coin biased in his favor

C_1

C_2

C_3

$P(H|C_1) = 0.1$ $P(H|C_2) = 0.5$ $P(H|C_3) = 0.9$

Using Prior Knowledge

We can encode it in the prior:

$P(C_1) = 0.05$ | $P(C_2) = 0.25$ | $P(C_3) = 0.70$

C_1

C_2

C_3

$P(H|C_1) = 0.1$ | $P(H|C_2) = 0.5$ | $P(H|C_3) = 0.9$

Experiment 1: Heads

Which coin did I use?

$P(C_1|H) = ?$ $P(C_2|H) = ?$ $P(C_3|H) = ?$

$$P(C_1|H) = \alpha P(H|C_1)P(C_1)$$

C_1

C_2

C_3

$P(H|C_1) = 0.1$ $P(H|C_2) = 0.5$ $P(H|C_3) = 0.9$

$P(C_1) = 0.05$ $P(C_2) = 0.25$ $P(C_3) = 0.70$

Experiment 1: Heads

Which coin <u>did</u> I use?

$P(C_1|H) = $ 0.006 $P(C_2|H) = $ 0.165 $P(C_3|H) = $ 0.829

Compare with ML posterior after Exp 1:

$P(C_1|H) = 0.066$ $P(C_2|H) = 0.333$ $P(C_3|H) = 0.600$

C_1

C_2

C_3

$P(H|C_1) = 0.1$ $P(H|C_2) = 0.5$ $P(H|C_3) = 0.9$

$P(C_1) = 0.05$ $P(C_2) = 0.25$ $P(C_3) = 0.70$

Experiment 2: Tails

Which coin <u>did</u> I use?

$P(C_1|HT) = ?$ $P(C_2|HT) = ?$ $P(C_3|HT) = ?$

$$P(C_1|HT) = \alpha P(HT|C_1)P(C_1) = \alpha P(H|C_1)P(T|C_1)P(C_1)$$

C_1

$P(H|C_1) = 0.1$

$P(C_1) = 0.05$

C_2

$P(H|C_2) = 0.5$

$P(C_2) = 0.25$

C_3

$P(H|C_3) = 0.9$

$P(C_3) = 0.70$

Experiment 2: Tails

Which coin did I use?

$P(C_1|HT) = 0.035$ $P(C_2|HT) = 0.481$ $P(C_3|HT) = 0.485$

$$P(C_1|HT) = \alpha P(HT|C_1)P(C_1) = \alpha P(H|C_1)P(T|C_1)P(C_1)$$

C_1 C_2 C_3



$P(H|C_1) = 0.1$ $P(H|C_2) = 0.5$ $P(H|C_3) = 0.9$

$P(C_1) = 0.05$ $P(C_2) = 0.25$ $P(C_3) = 0.70$

Experiment 2: Tails

Which coin did I use?

$P(C_1|HT) = 0.035$ $P(C_2|HT)=0.481$ $P(C_3|HT) = 0.485$

C_1

$P(H|C_1) = 0.1$

$P(C_1) = 0.05$

C_2

$P(H|C_2) = 0.5$

$P(C_2) = 0.25$

C_3

$P(H|C_3) = 0.9$

$P(C_3) = 0.70$

Your Estimate?

What is the probability of heads after two experiments?

Most likely coin: Best estimate for P(H)

C_3  $P(H|C_3) = 0.9$

C_1

$P(H|C_1) = 0.1$

$P(C_1) = 0.05$

C_2

$P(H|C_2) = 0.5$

$P(C_2) = 0.25$

C_3

$P(H|C_3) = 0.9$

$P(C_3) = 0.70$

Your Estimate?

Maximum A Posteriori (MAP) Estimate:
The best hypothesis that fits observed data assuming a non-uniform prior

Most likely coin:

C_3 

Best estimate for P(H)

$P(H|C_3) = 0.9$

C_3

$P(H|C_3) = 0.9$

$P(C_3) = 0.70$

Did We Do The Right Thing?

$P(C_1|HT)=0.035$ $P(C_2|HT)=0.481$ $P(C_3|HT)=0.485$

C_1

$P(H|C_1) = 0.1$

C_2

$P(H|C_2) = 0.5$

C_3

$P(H|C_3) = 0.9$

Did We Do The Right Thing?

$P(C_1|HT) = 0.035$ $P(C_2|HT) = 0.481$ $P(C_3|HT) = 0.485$

C_2 and C_3 are almost equally likely

C_1

$P(H|C_1) = 0.1$

C_2

$P(H|C_2) = 0.5$

C_3

$P(H|C_3) = 0.9$

A Better Estimate

Recall: $P(H) = \sum_{i=1}^{3} P(H|C_i)P(C_i)$ = 0.680

$P(C_1|HT)=0.035$ $P(C_2|HT)=0.481$ $P(C_3|HT)=0.485$

C_1

$P(H|C_1) = 0.1$

C_2

$P(H|C_2) = 0.5$

C_3

$P(H|C_3) = 0.9$

Bayesian Estimate

Bayesian Estimate: Minimizes prediction error, given data and (generally) assuming a non-uniform prior

$$P(H) = \sum_{i=1}^{3} P(H|C_i)P(C_i) = 0.680$$

$P(C_1|HT)=0.035$ $P(C_2|HT)=0.481$ $P(C_3|HT)=0.485$

C_1 C_2 C_3

$P(H|C_1) = 0.1$ $P(H|C_2) = 0.5$ $P(H|C_3) = 0.9$

Comparison After more Experiments

- Seen: HTHHHHHHHH
- Maximum likelihood:
  - P(H) = 0.5
  - After 10 experiments: P(H) = 0.9
- Maximum a posteriori:
  - P(H) = 0.9
  - After 10 experiments: P(H) = 0.9
- Bayesian:
  - P(H) = 0.68
  - After 10 experiments: P(H) = 0.9

Comparison

- ML:
  - Easy to compute
- MAP:
  - Easy to compute
  - Incorporates prior knowledge
- Bayesian:
  - Minimizes error -> great with little data
  - Potentially very difficult to compute

Brute-Force MAP Hypothesis Learner

1. For each hypothesis h in H, calculate the posterior probability

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

2. Output the hypothesis h_{MAP} with the highest posterior probability

$$h_{MAP} = \operatorname*{argmax}_{h \in H} P(h|D)$$

Relation to Concept Learning

Let $D = \langle c(x_1), \ldots, c(x_m) \rangle$ (examples' classes)

Choose $P(D|h)$

- $P(D|h) = 1$ if h consistent with D
- $P(D|h) = 0$ otherwise

Choose $P(h)$ to be *uniform* distribution

- $P(h) = \frac{1}{|H|}$ for all h in H

Then

$$P(h|D) = \begin{cases} \frac{1}{|VS_{H,D}|} & \text{if } h \text{ is consistent with } D \\ 0 & \text{otherwise} \end{cases}$$

Most Probable Classification of New Instances

So far we've sought the most probable *hypothesis* given the data D (i.e., h_{MAP})

Given new instance x, what is its most probable *classification*? Not $h_{MAP}(x)$!

Consider:

- Three possible hypotheses:

  $$P(h_1|D) = .4,\ P(h_2|D) = .3,\ P(h_3|D) = .3$$

- Given new instance x,

  $$h_1(x) = +,\ h_2(x) = -,\ h_3(x) = -$$

- What's most probable classification of x?

Bayes Optimal Classifier

Bayes optimal classification:

$$\arg\max_{v_j \in V} \sum_{h_i \in H} P(v_j|h_i)P(h_i|D)$$

Example:

$$P(h_1|D) = .4, \quad P(-|h_1) = 0, \quad P(+|h_1) = 1$$
$$P(h_2|D) = .3, \quad P(-|h_2) = 1, \quad P(+|h_2) = 0$$
$$P(h_3|D) = .3, \quad P(-|h_3) = 1, \quad P(+|h_3) = 0$$

therefore

$$\sum_{h_i \in H} P(+|h_i)P(h_i|D) = .4$$
$$\sum_{h_i \in H} P(-|h_i)P(h_i|D) = .6$$

and

$$\arg\max_{v_j \in V} \sum_{h_i \in H} P(v_j|h_i)P(h_i|D) = -$$

Gibbs Classifier

Bayes optimal classifier is hopelessly inefficient

Gibbs algorithm:

1. Choose one hypothesis at random, according to $P(h|D)$
2. Use this to classify new instance

Surprising fact: Assume target concepts are drawn at random from H according to priors on H. Then

$$E[error_{Gibbs}] \leq 2 \times E[error_{BayesOptimal}]$$

Outline

- Probability overview
- Naïve Bayes
- Bayesian learning
- Bayesian networks
  - Representation
  - Inference
  - Parameter learning
  - Structure learning

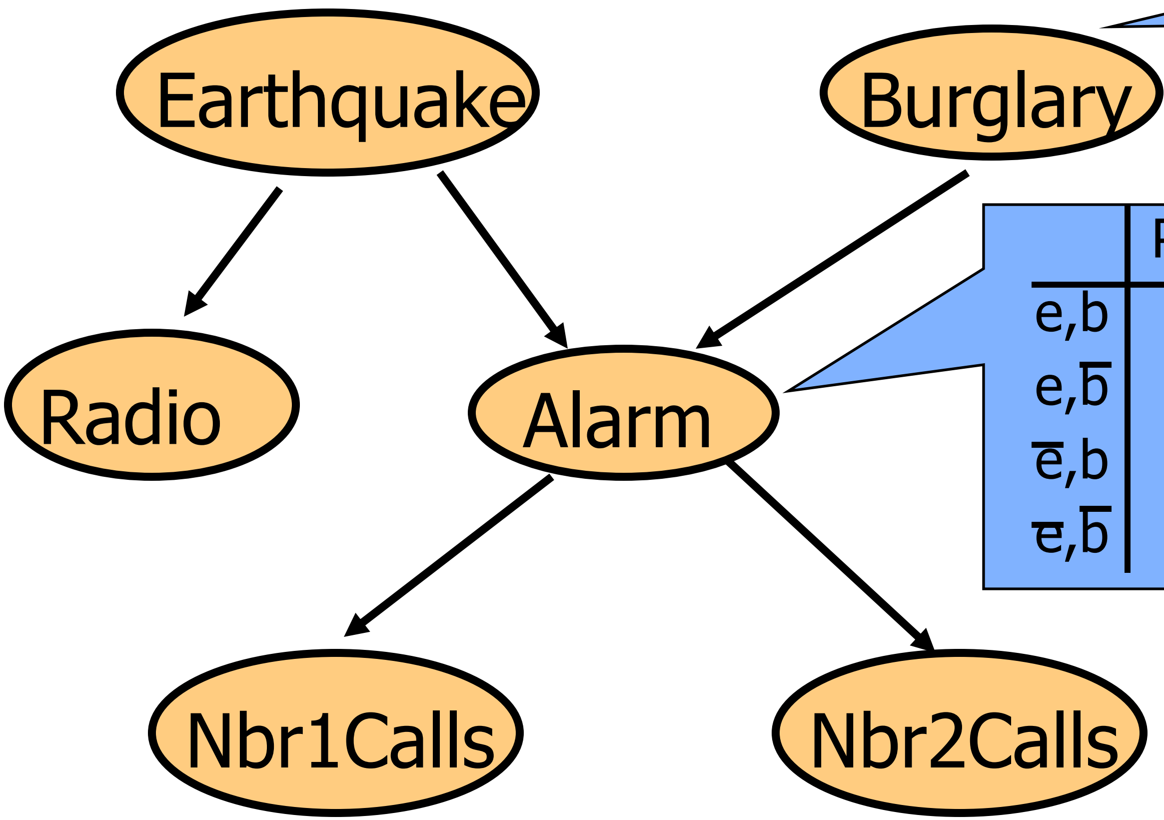
Bayesian Network

- In general, a joint distribution P over variables $(X_1,\ldots,X_n)$ requires exponential space
- A Bayesian network is a graphical representation of the conditional independence relations in P
  - Usually quite compact
  - Requires fewer parameters than the full joint distribution
  - Can yield more efficient inference and belief updates

Bayesian Network

- Formally, a Bayesian network is
  - A directed, acyclic graph
  - Each node is a random variable
  - Each node X has a conditional probability distribution P(X | Parents(X))
  - Intuitively, an arc from X to Y means that X and Y are related

An Example Bayes Net



Pr(B=t)	Pr(B=f)
0.05	0.95

	Pr(A|E,B)
e,b	0.9 (0.1)
e,$\bar{b}$	0.2 (0.8)
$\bar{e}$,b	0.85 (0.15)
$\bar{e}$,$\bar{b}$	0.01 (0.99)

Terminology

- If X and its parents are discrete, we represent $\mathbf{P}(X|Parents(X))$ by
  - A *conditional probability table (CPT)*
  - It specifies the probability of each value of X, given all possible settings for the variables in $Parents(X)$.
  - Number of parameters *locally* exponential in $|Parents(X)|$
- A *conditioning case* is a row in this CPT: A setting of values for the parent nodes

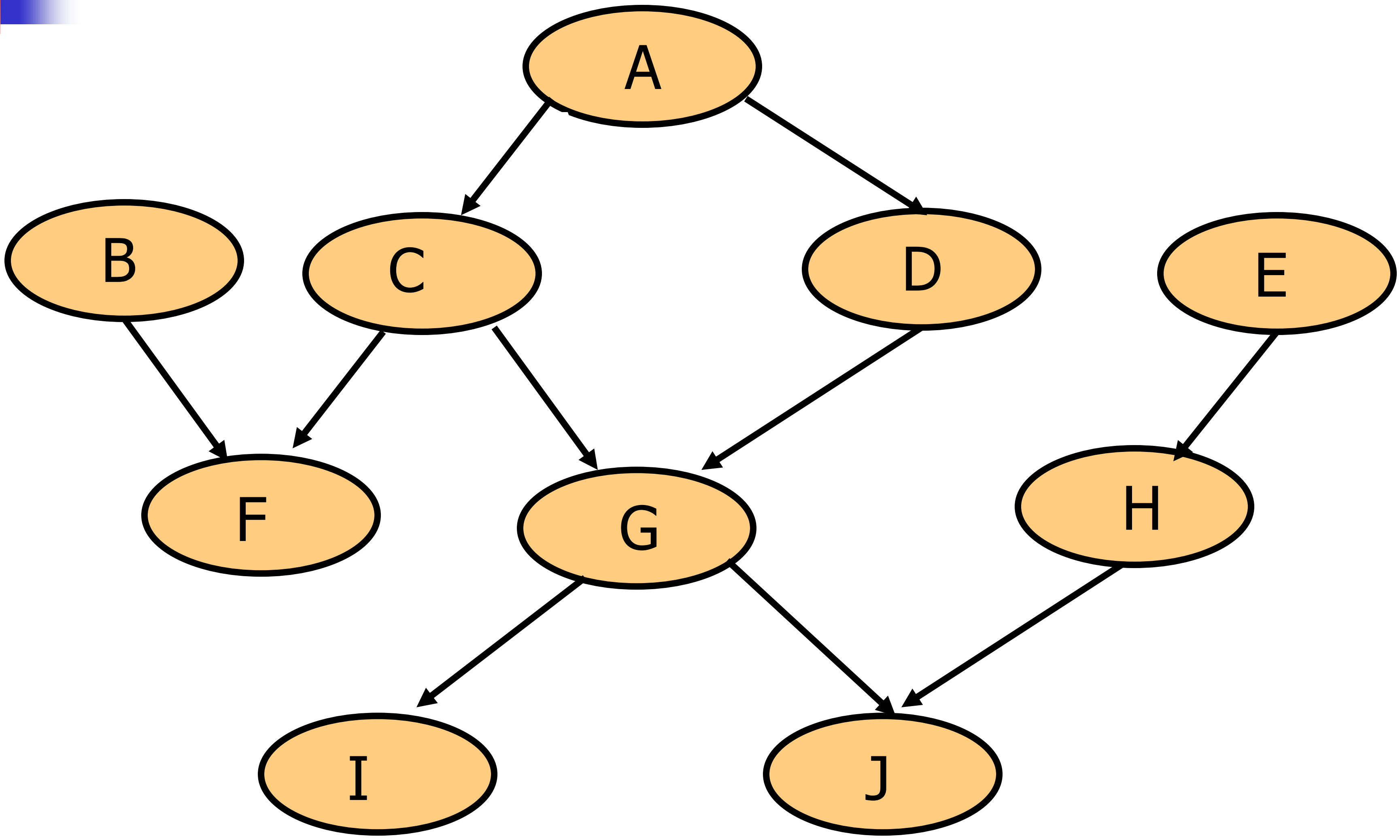
Bayesian Network Semantics

- A Bayesian network completely specifies a full joint distribution over variables $X_1,\ldots,X_n$
- $P(x_1,\ldots,x_n) = \prod_{i}^{n} P(x_i \mid Parents(x_i))$
- Here $P(x_1,\ldots,x_n)$ represents a specific setting for all variables (i.e., $P(X_1 = x_1,\ldots, X_n = x_n)$)

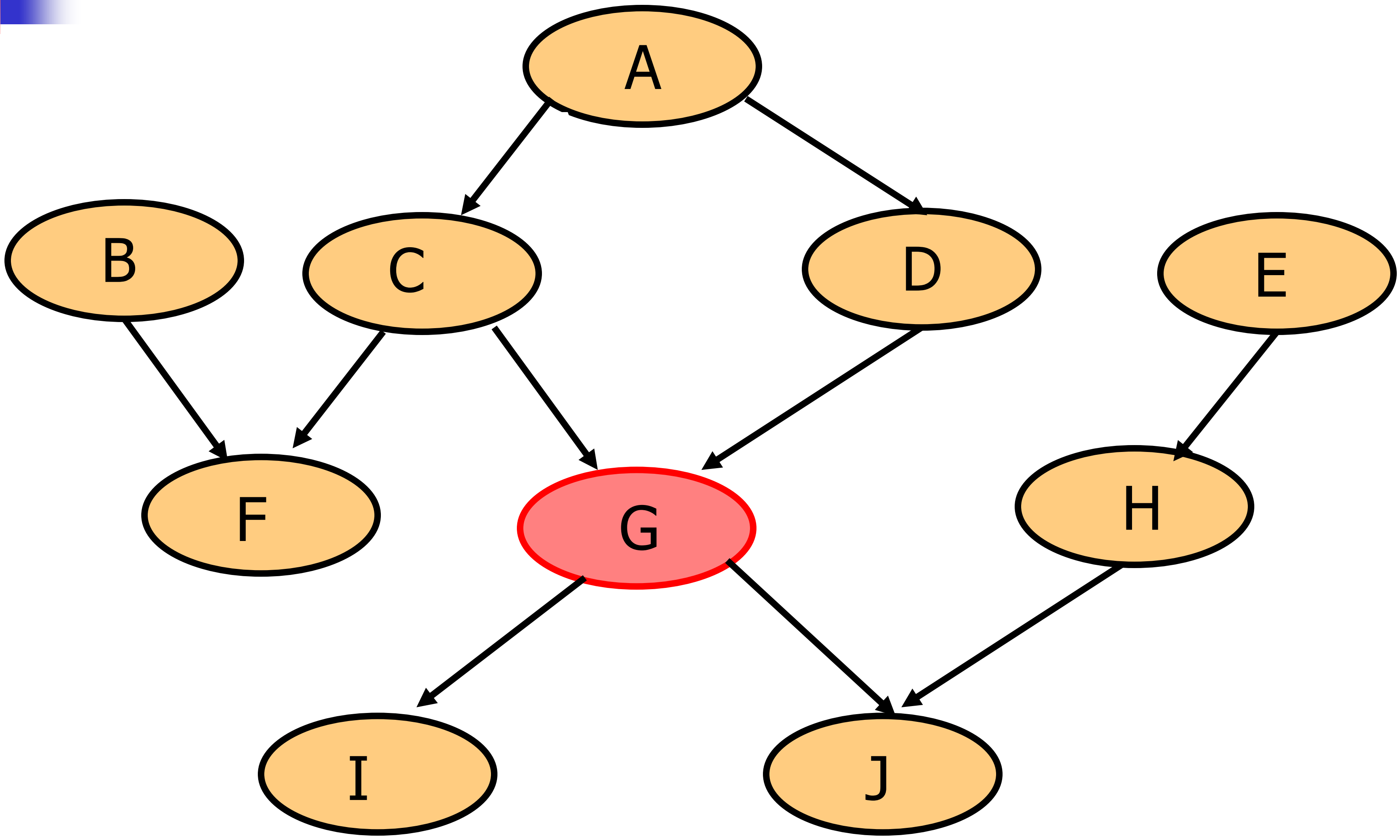
Conditional Indepencies

- A node X is conditionally independent of its predecessors given its parents
- Markov Blanket of X_i consists of:
  - Parents of X_i
  - Children of X_i
  - Other parents of X_i's children
- X is conditionally independent of all nodes in the network given its Markov Blanket

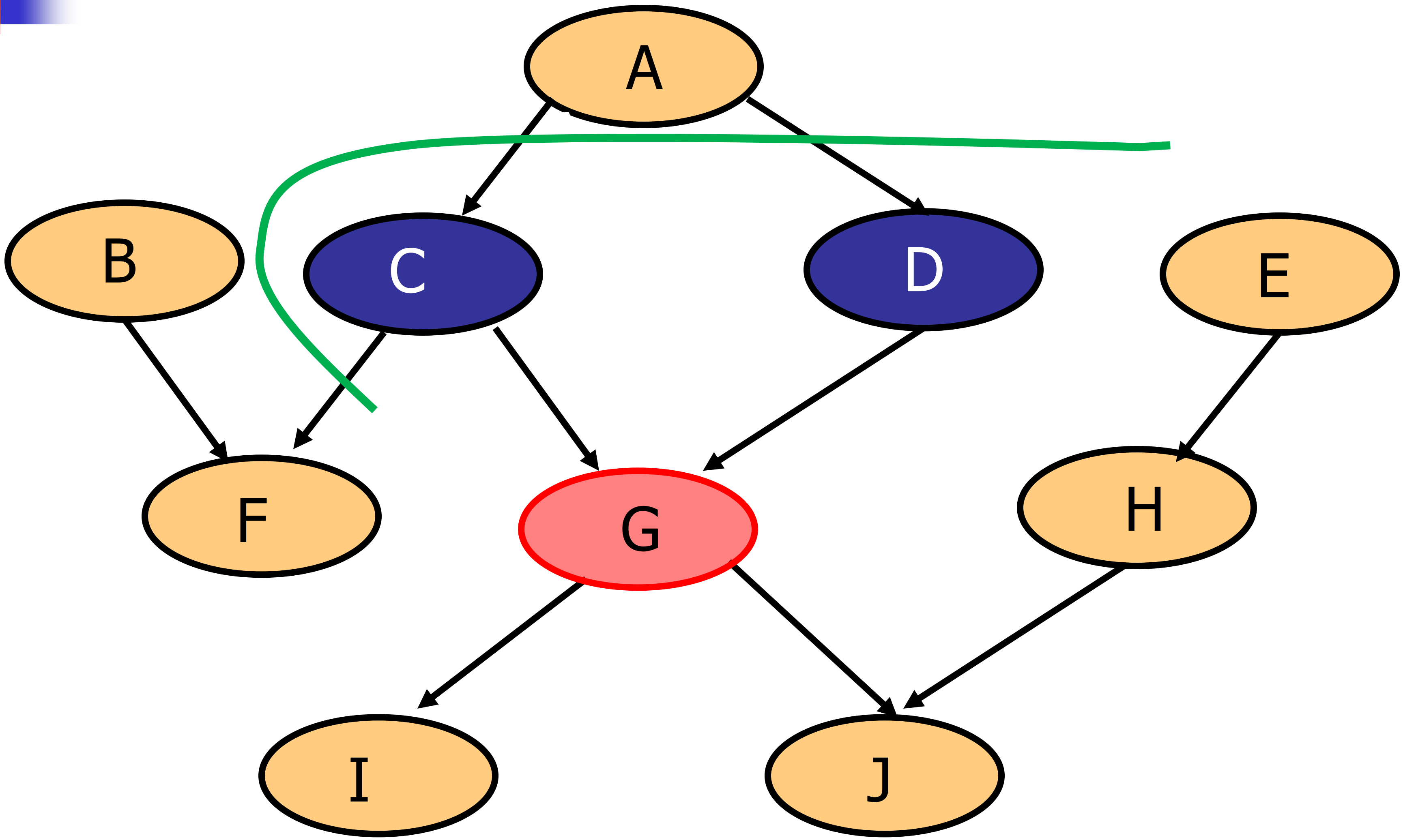
Example: Parents



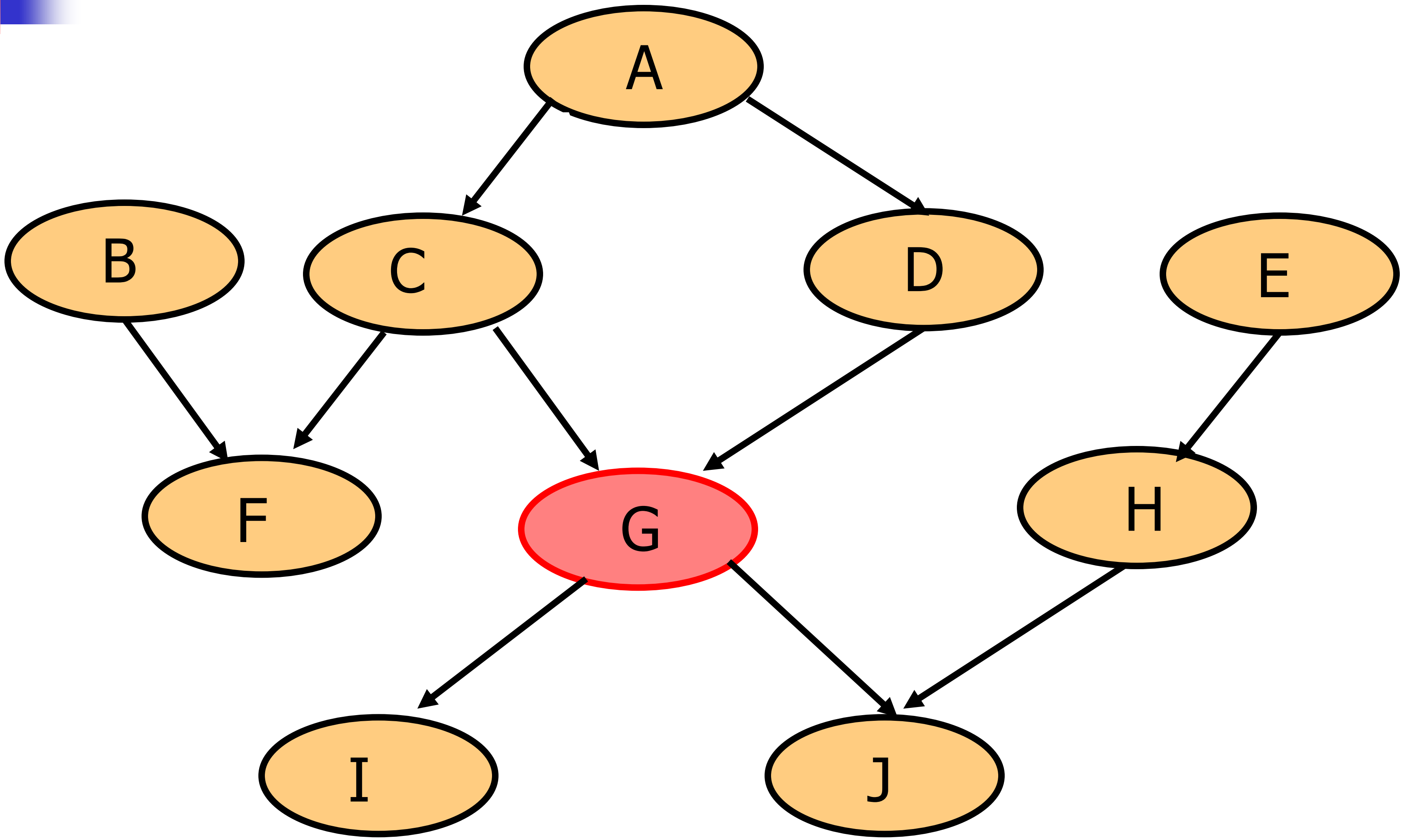
Example: Parents



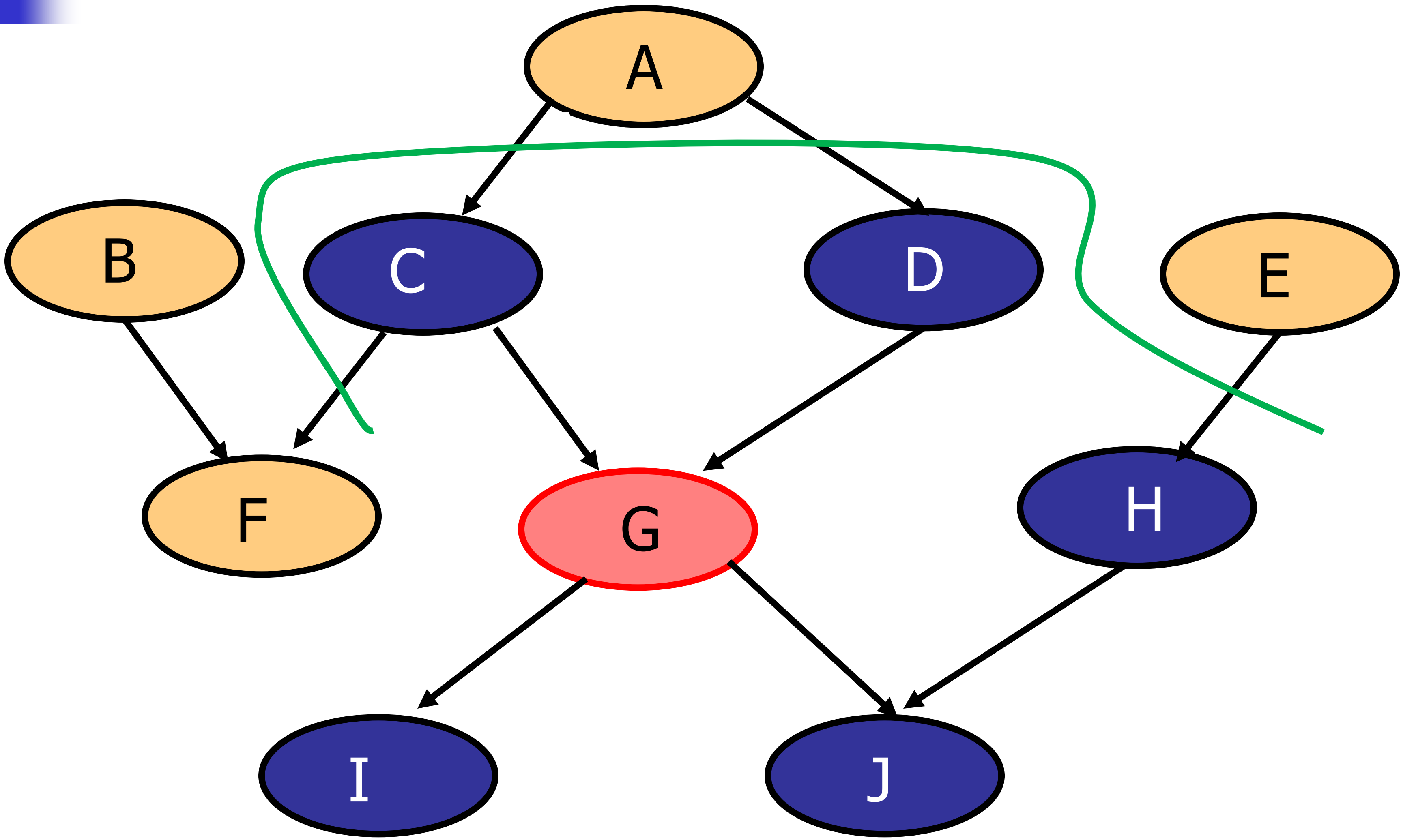
Example: Parents



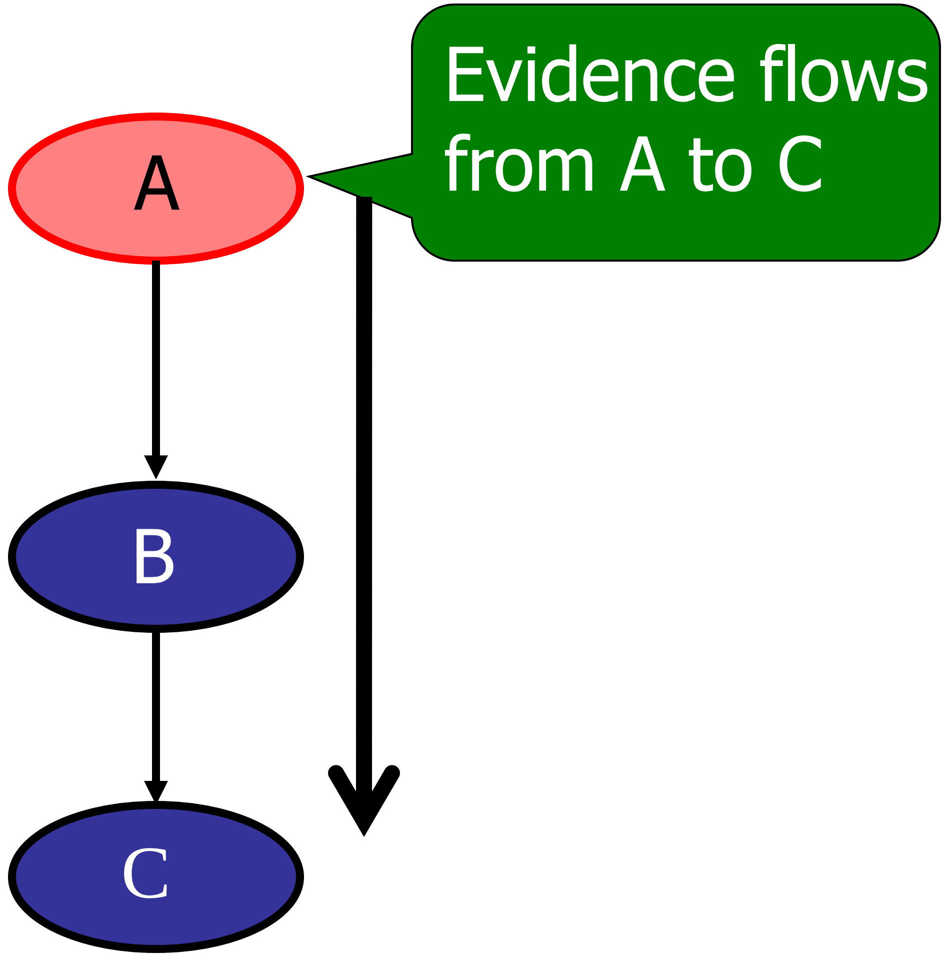
Example: Markov Blanket



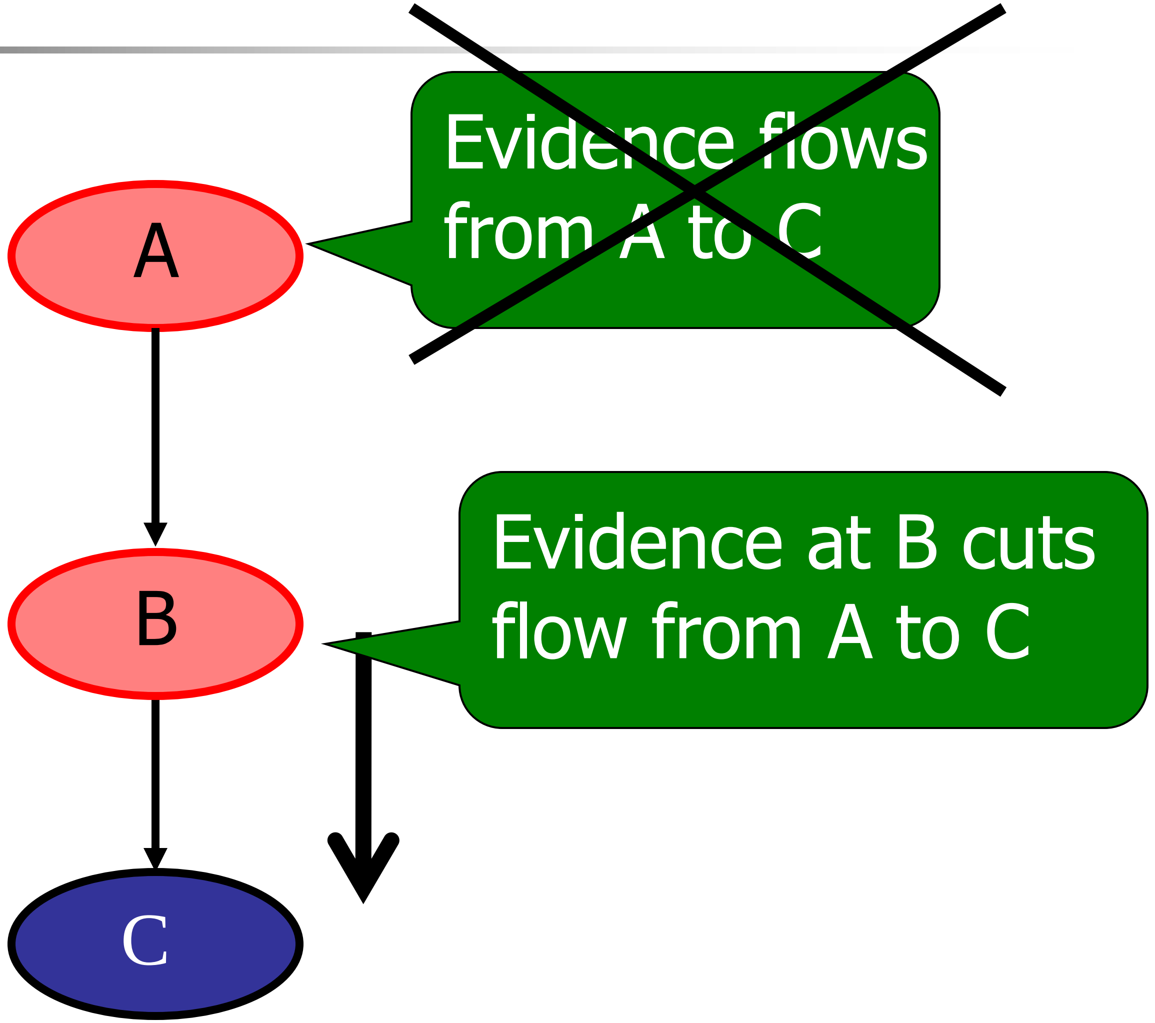
Example: Markov Blanket



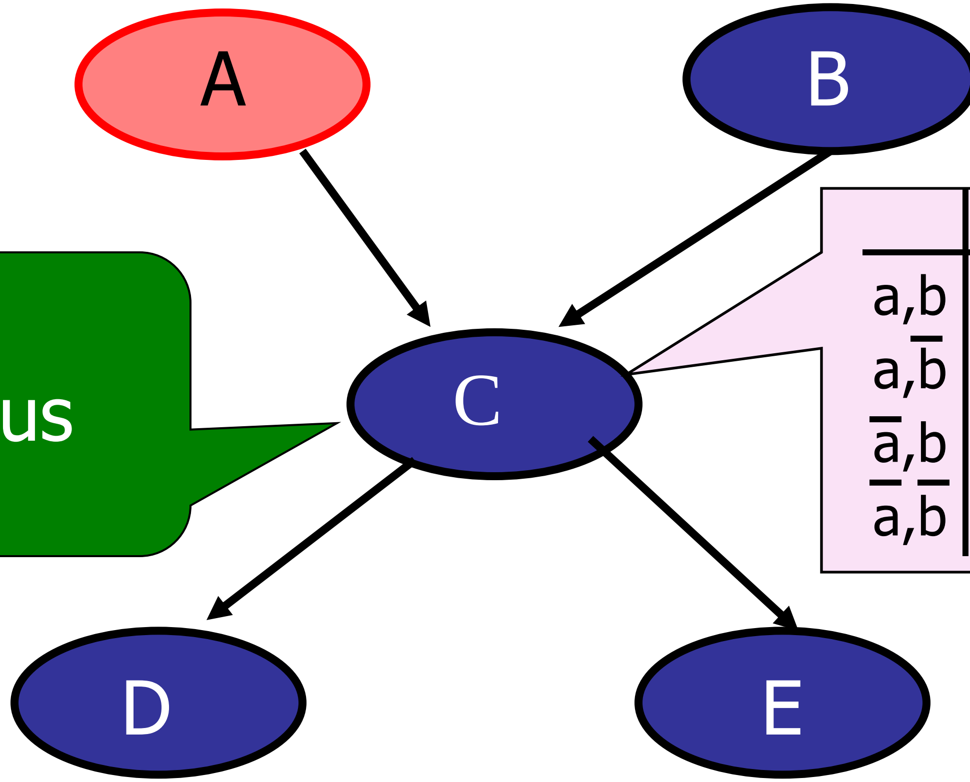
D-Separation



D-Separation



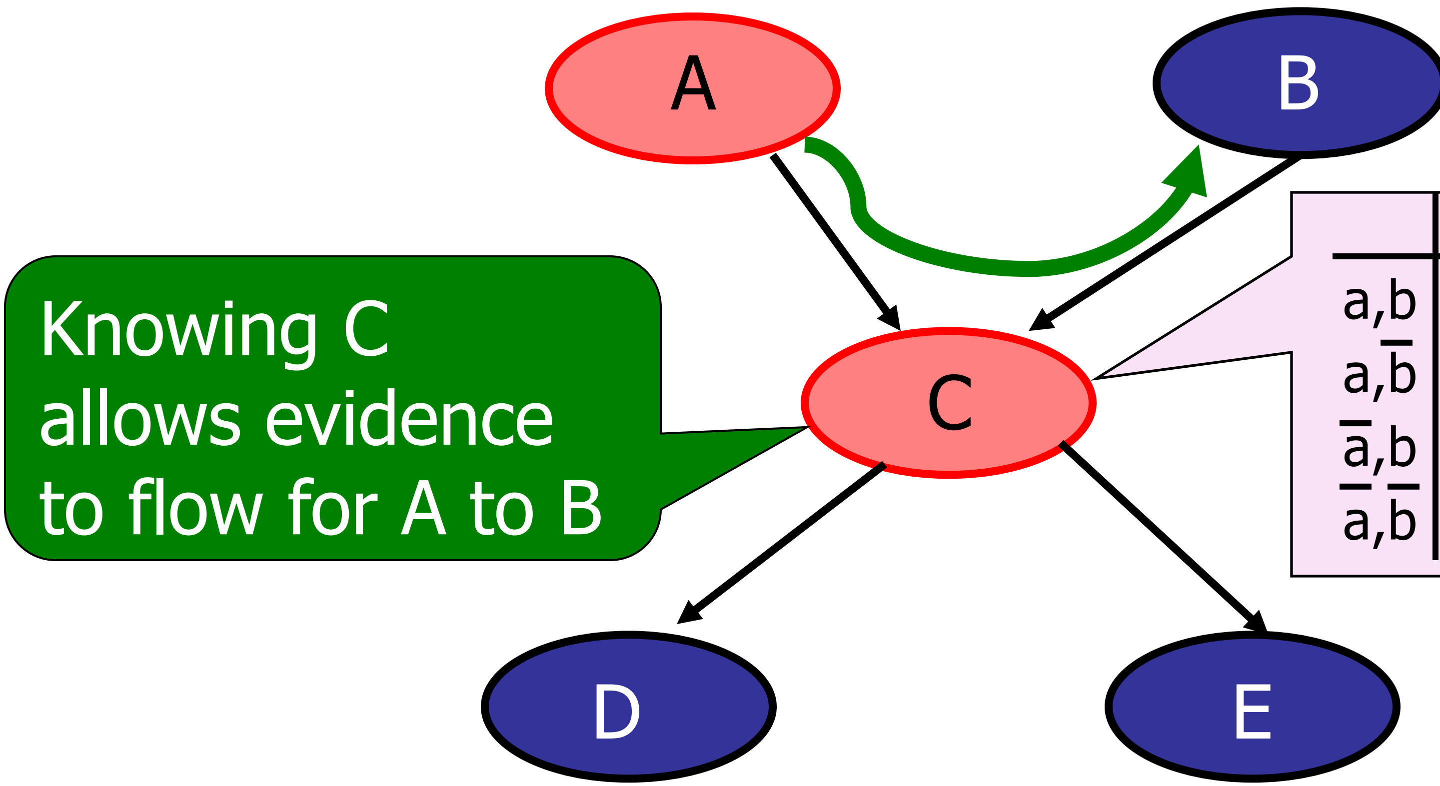
D-Separation



Knowing A does not tell us about B

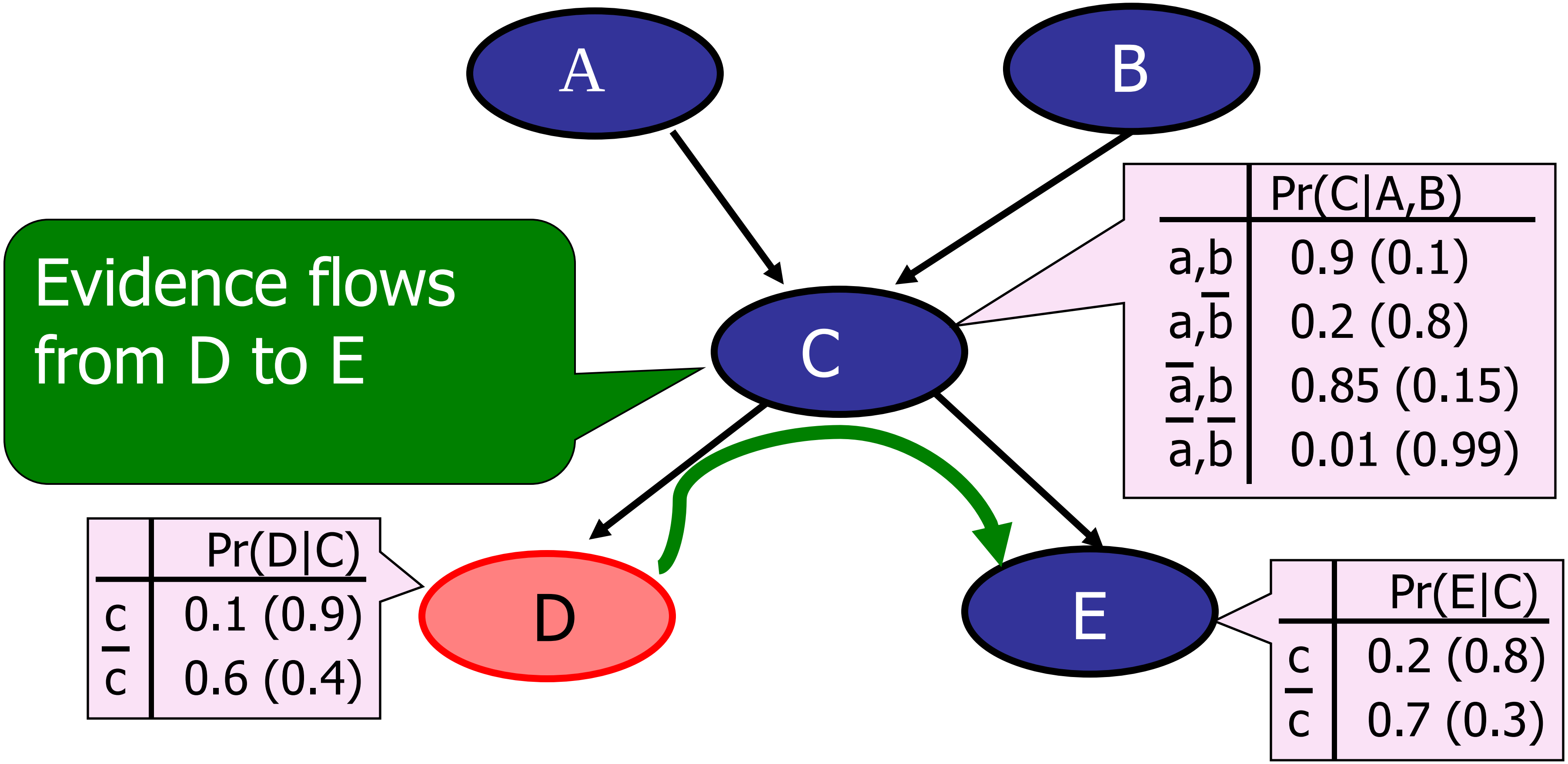
	Pr(C|A,B)
a,b	0.9 (0.1)
$a,\bar{b}$	0.2 (0.8)
$\bar{a},b$	0.85 (0.15)
$\bar{a},\bar{b}$	0.01 (0.99)

D-Separation

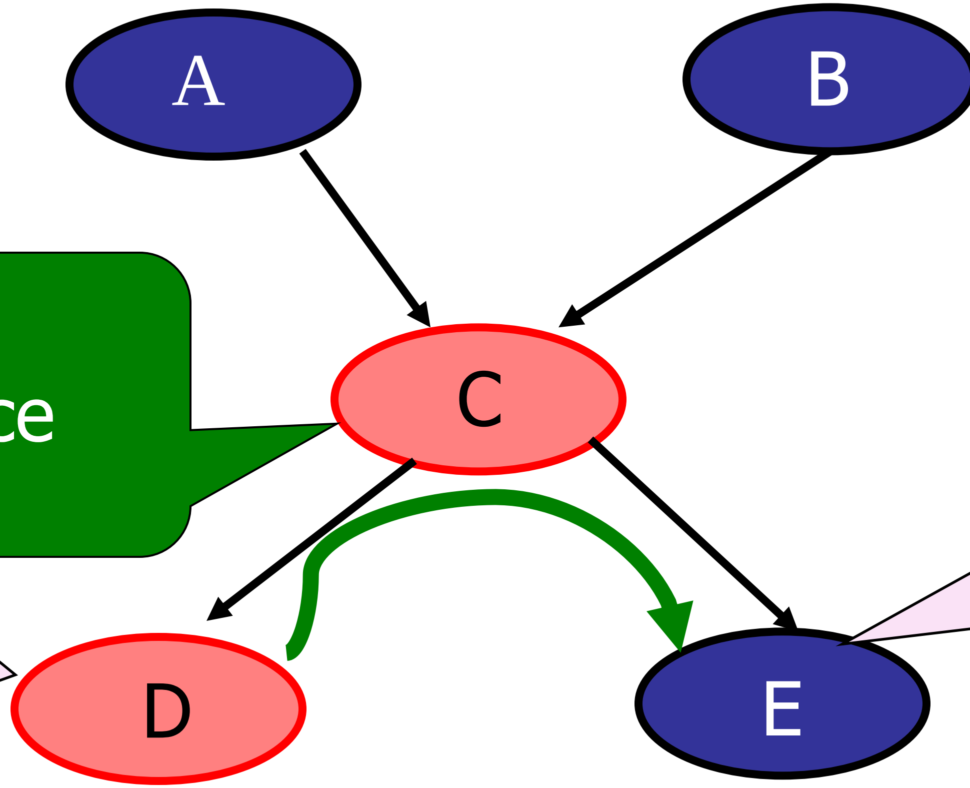


	Pr(C|A,B)
a,b	0.9 (0.1)
$a,\bar{b}$	0.2 (0.8)
$\bar{a},b$	0.85 (0.15)
$\bar{a},\bar{b}$	0.01 (0.99)

D-Separation



D-Separation



	Pr(D|C)
c	0.1 (0.9)
$\bar{c}$	0.6 (0.4)

	Pr(E|C)
c	0.2 (0.8)
$\bar{c}$	0.7 (0.3)

Outline

- Probability overview
- Naïve Bayes
- Bayesian learning
- Bayesian networks
  - Representation
  - Inference
  - Parameter learning
  - Structure learning

Inference in BNs

- The graphical independence representation yields efficient inference schemes
- Generally, we want to compute
  - P(X) or
  - P(X | E), where E is (conjunctive) evidence
- Computations organized by network topology
- Two well-known exact algorithms:
  - Variable elimination
  - Junction trees

Variable Elimination

- A factor is a function from set of variables to a specific value: CPTS are factors
  - E.g.: p(A | E,B) is a function of A,E,B

- VE works by eliminating all variables in turn until there is a factor with only query variable

Joint Distributions & CPDs Vs. Potentials

CPT for P(B | A)

	b	¬*b*
a	.1	.9
¬*a*	.6	.4

Represent probability distributions

1. For CPT, specific setting of parents, values of child must sum to 1
2. For joint, all entries sum to 1

Potential

	b	¬*b*
¬*a*	.2	.4
¬*a*	.3	.5

Potentials occur when we temporarily forget meaning associated with table

1. Must be non-negative
2. Doesn't have to sum to 1

Arise when incorporating evidence

Multiplying Potentials

	a	$\neg a$
b	.1	.5
$\neg b$	.2	.8

X

	c	$\neg c$
b	.2	.4
$\neg b$	.3	.5

=

	a		$\neg a$	
	c	$\neg c$	c	$\neg c$
b	.02			
$\neg b$				

Multiplying Potentials

	a	¬*a*
b	.1	.5
¬*b*	.2	.8

X

	c	¬*c*
b	.2	.4
¬*b*	.3	.5

=

	a		¬*a*	
	c	¬*c*	c	¬*c*
b	.02	.04		
¬*b*				

Multiplying Potentials

	a	$\neg a$
b	.1	.5
$\neg b$	.2	.8

X

	c	$\neg c$
b	.2	.4
$\neg b$	.3	.5

=

	a		$\neg a$	
	c	$\neg c$	c	$\neg c$
b	.02	.04		
$\neg b$	.06			

Multiplying Potentials

	a	¬*a*
b	.1	.5
¬*b*	.2	.8

X

	c	¬*c*
b	.2	.4
¬*b*	.3	.5

=

	a		¬*a*	
	c	¬*c*	c	¬*c*
b	.02	.04		
¬*b*	.06	.10		

Multiplying Potentials

	a	¬*a*
b	.1	.5
¬*b*	.2	.8

X

	c	¬*c*
b	.2	.4
¬*b*	.3	.5

=

	a		¬*a*	
	c	¬*c*	c	¬*c*
b	.02	.04	.10	.20
¬*b*	.06	.10	.24	.40

Marginalize/sum out a variable

	b	$\neg b$
a	.1	.5
$\neg a$	.2	.8

$$\xrightarrow{\quad\sum_a\quad}$$

b	$\neg b$
.3	1.3

Normalize a potential

	b	$\neg b$
a	.1	.5
$\neg a$	.2	.8

α =

	b	$\neg b$
a	.0625	.3125
$\neg a$	.125	.5

Key Observation

$$\Sigma_a(P_1 \times P_2) = (\Sigma_a P_1)\times P_2 \text{ if A is not in } P_2$$

	b	¬b
a	.1	.5
¬a	.2	.8

x

	c	¬c
b	.2	.4
¬b	.3	.5

=

	a		¬a	
	c	¬c	c	¬c
b	.02	.04	.04	.08
¬b	.06	.25	.24	.40

Key Observation

$$\Sigma_a (P_1 \text{ x } P_2) = (\Sigma_a P_1) \text{x } P_2 \text{ if A is not in } P_2$$

	b	¬b
a	.1	.5
¬a	.2	.8

x

	c	¬c
b	.2	.4
¬b	.3	.5

=

	a		¬a	
	c	¬c	c	¬c
b	.02	.04	.04	.08
¬b	.15	.25	.24	.40

Σ_a

	c	¬c
b	.06	.12
¬b	.39	.65

	b	¬b
a	.1	.5
¬a	.2	.8

$\xrightarrow{}$ Σ_a

b	.3
¬b	1.3

x

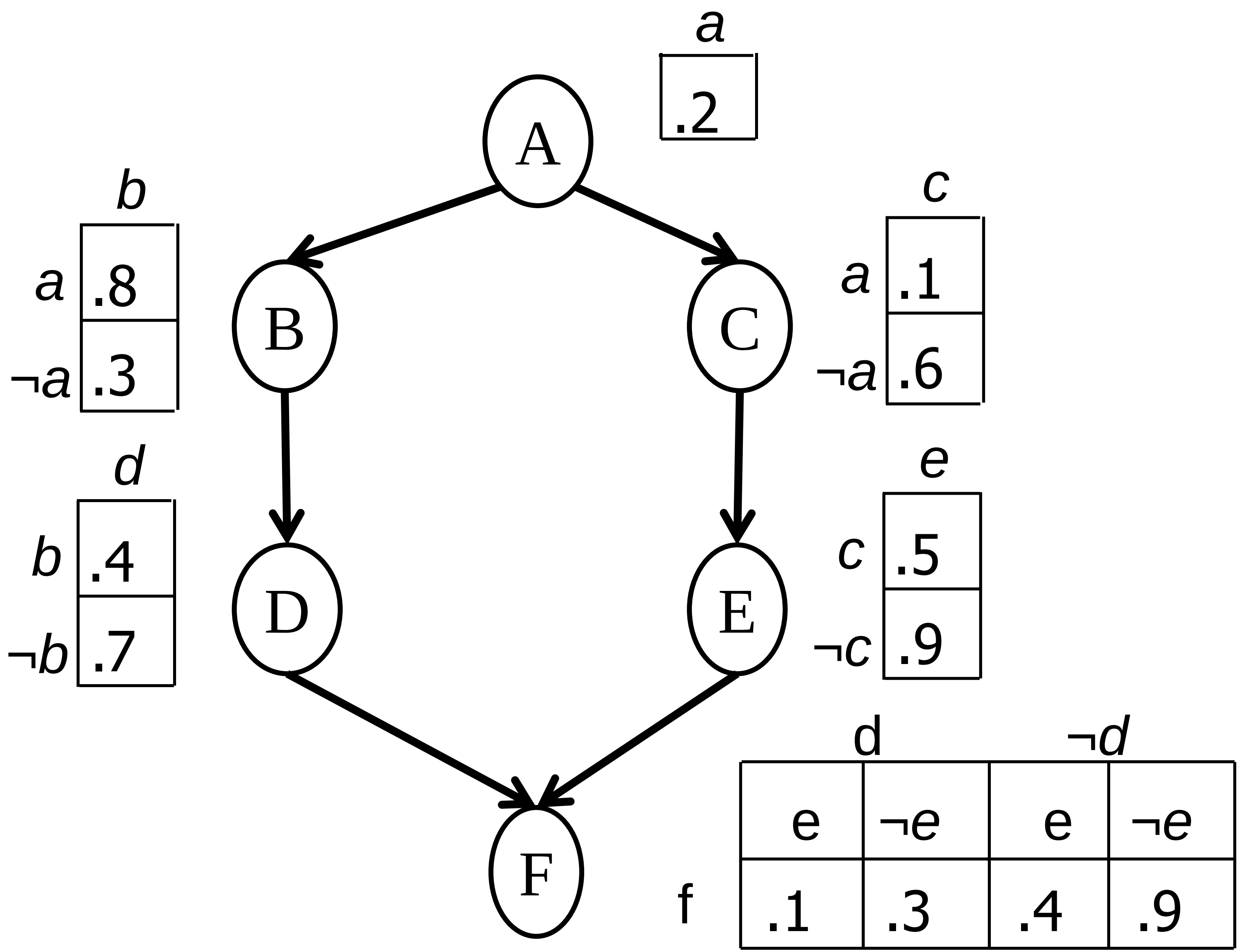
	c	¬c
b	.2	.4
¬b	.3	.5

=

	c	¬c
b	.06	.12
¬b	.39	.65

Variable Elimination Procedure

- The initial potentials are the CPTS in the BN
- Repeat until only query variable remains:
  - Choose a variable to eliminate
  - Multiply all potentials that contain the variable
  - If no evidence for the variable then sum the variable out and replace original potential by the new result
  - Else, remove variable based on evidence
- Normalize the remaining potential to get the final distribution over the query variable

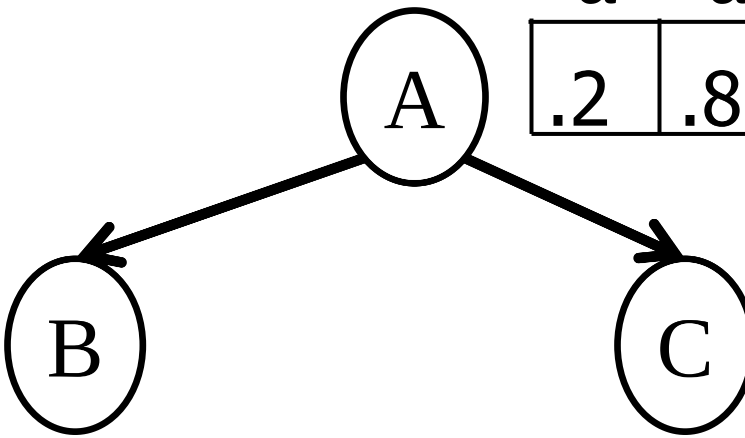


$$P(A,B,C,D,E,F) = P(A)\, P(B|A)\, P(C|A)\, P(D|B)\, P(E|C)\, P(F|D,E)$$

Query: P(F| C = true)

Elimination Ordering: A,B,C,D,E

	a	¬*a*
	.2	.8



	b	¬*b*
a	.8	.2
¬*a*	.3	.7

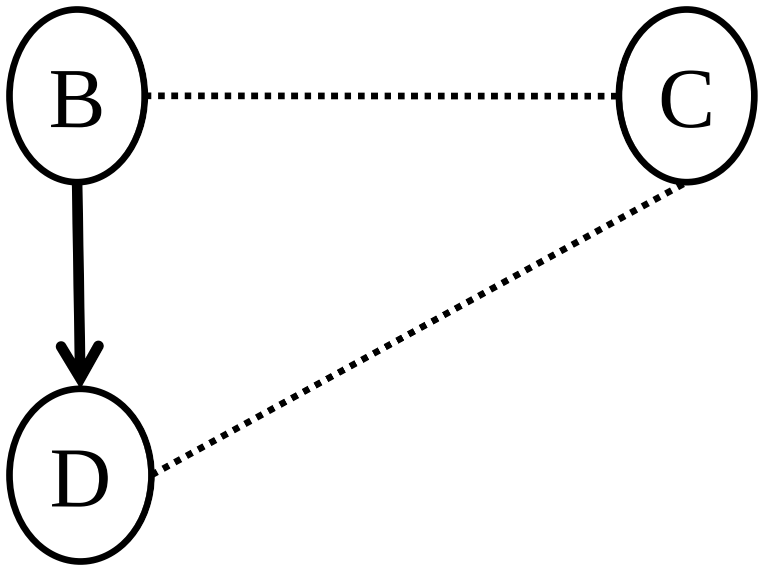
	c	¬*c*
a	.1	.9
¬*a*	.6	.4

Before eliminating A, multiple all potentials involving A

	b		¬*b*	
	c	¬*c*	c	¬*c*
a	.016	.144	.004	.036
¬*a*	.144	.096	.336	.224
	.16	.24	.34	.26

Sum out A

Now, eliminate B, multiple all potentials involving B



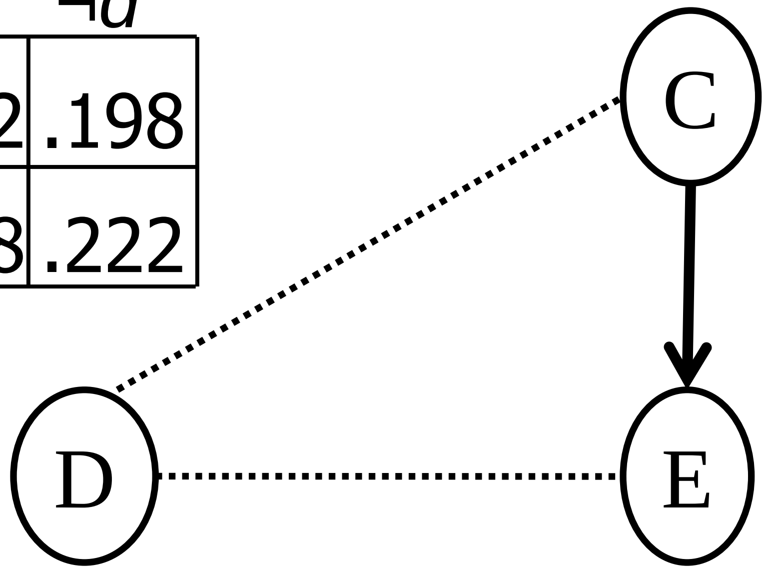
	c	¬c
b	.16	.24
¬b	.34	.26

	d	¬d
b	.4	.6
¬b	.7	.3

	c		¬c	
	d	¬d	d	¬d
b	.064	.096	.096	.144
¬b	.238	.102	.182	.078
Sum out B	.302	.198	.278	.222

Next, eliminate C, multiple all potentials involving C

	d	¬*d*
c	.302	.198
¬*c*	.278	.222

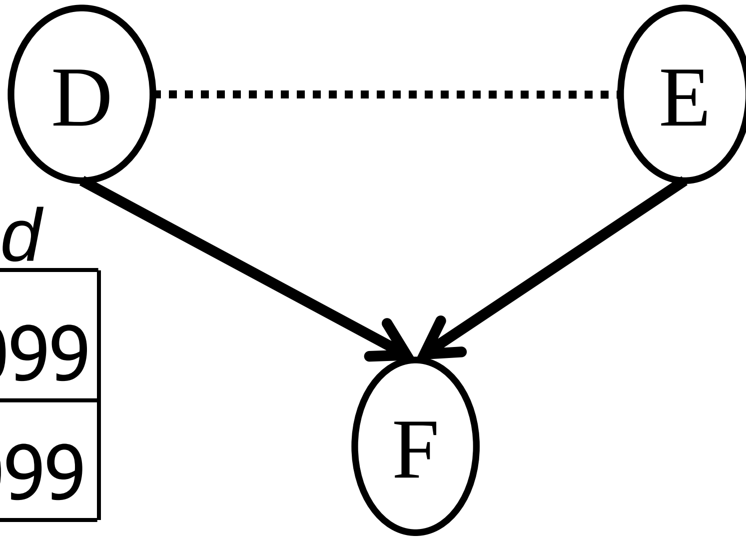


	e	¬*e*
c	.5	.5
¬*c*	.9	.1

	d		¬*d*	
	e	¬*e*	e	¬*e*
c	.151	.151	.099	.099
~~¬*c*~~	~~.250~~	~~.028~~	~~.200~~	~~.022~~

We have evidence for C, so eliminate ¬*c*

Next, eliminate D, multiple all potentials involving D



	d		¬d	
	e	¬e	e	¬e
f	.1	.3	.4	.9
¬f	.9	.7	.6	.1

	d	¬d
e	.151	.099
¬e	.151	.099

	e		¬e	
	f	¬f	f	¬f
d	.015	.136	.040	.106
¬d	.040	.059	.089	.010

Sum out d

.055 .195 .129 .116

	f	¬f
e	.055	.195
¬e	.129	.116

Next, eliminate E

	f	$\neg f$
e	.055	.195
$\neg e$	.129	.116

$\xrightarrow{\;\sum_e\;}$

f	$\neg f$
.184	.311

Normalize
α = 2.0202

f	$\neg f$
.372	.628

Notes on Variable Elimination

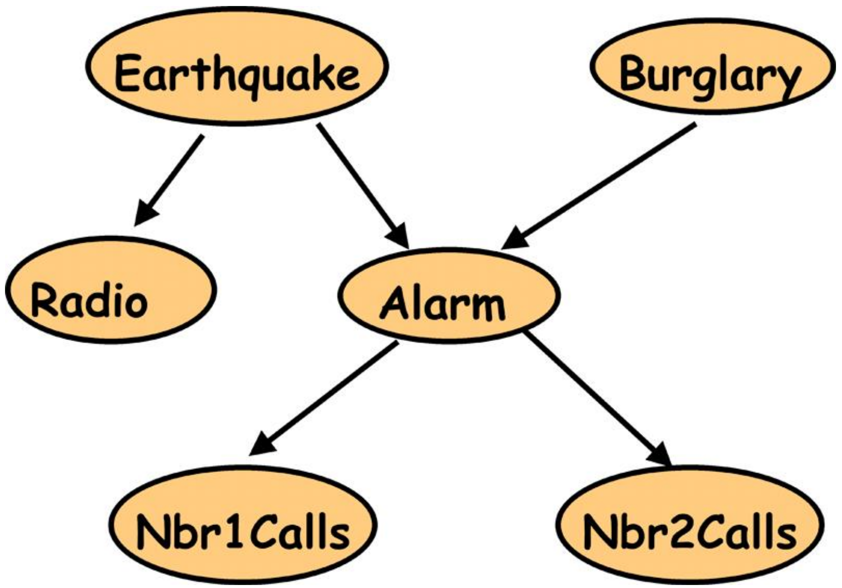
- Each operation is a simple multiplication of factors and summing out a variable

- Complexity determined by size of largest factor
  - E.g., in example 3 variables (not 6)
  - Linear in number of variables, exponential in largest factor
  - Elimination ordering greatly impacts factor size
  - Optimal elimination ordering: NP-hard

Outline

- Probability overview
- Naïve Bayes
- Bayesian learning
- Bayesian networks
  - Representation
  - Inference
  - Parameter learning
  - Structure learning

Parameter Estimate for Bayesian Networks

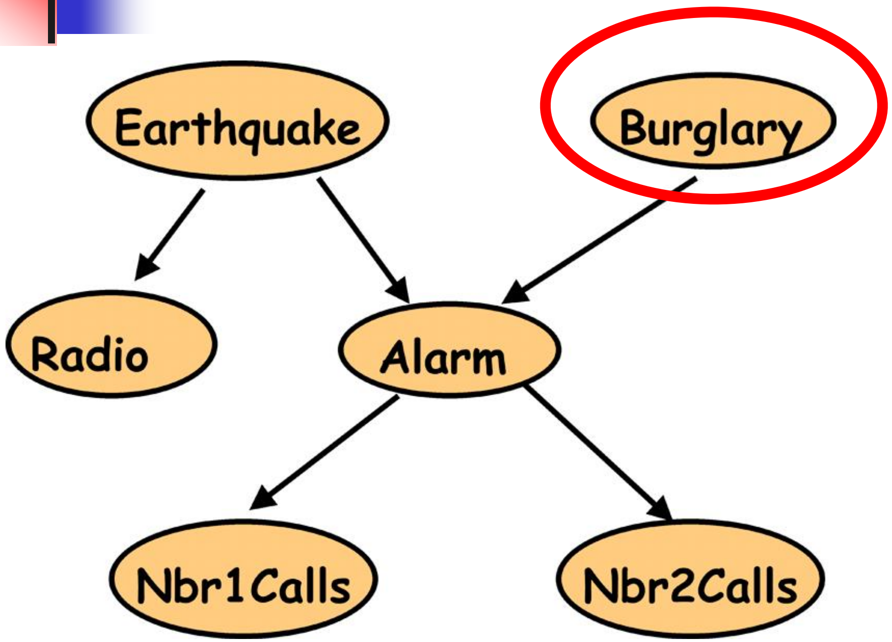


E	B	R	A	J	M
T	F	T	T	F	T
F	F	F	F	F	T
F	T	F	T	T	T
F	F	F	T	T	T
F	T	F	F	F	F
...					

We have:

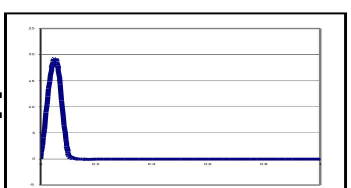
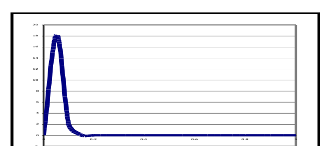
- Bayes Net structure and observations
- We need: Bayes Net parameters

Parameter Estimate for Bayesian Networks



B
F
F
T
F
T

Prior

P(B) =  + data = 

Now compute either MAP or Bayesian estimate

What Prior to Use

- The following are two common priors
- Binary variable Beta
  - Posterior distribution is binomial
  - Easy to compute posterior

- Discrete variable Dirichlet
  - Posterior distribution is multinomial
  - Easy to compute posterior

One Prior: Beta Distribution

$$\beta(x) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} x^{a-1}(1-x)^{b-1},$$

a,b

$$0 \le x \le 1 \text{ and } a, b > 0$$

$$\text{Here } \Gamma(y) = \int_0^\infty x^{y-1} e^{-x} dx$$

For any positive integer *y*, $\Gamma(y) = (y-1)!$

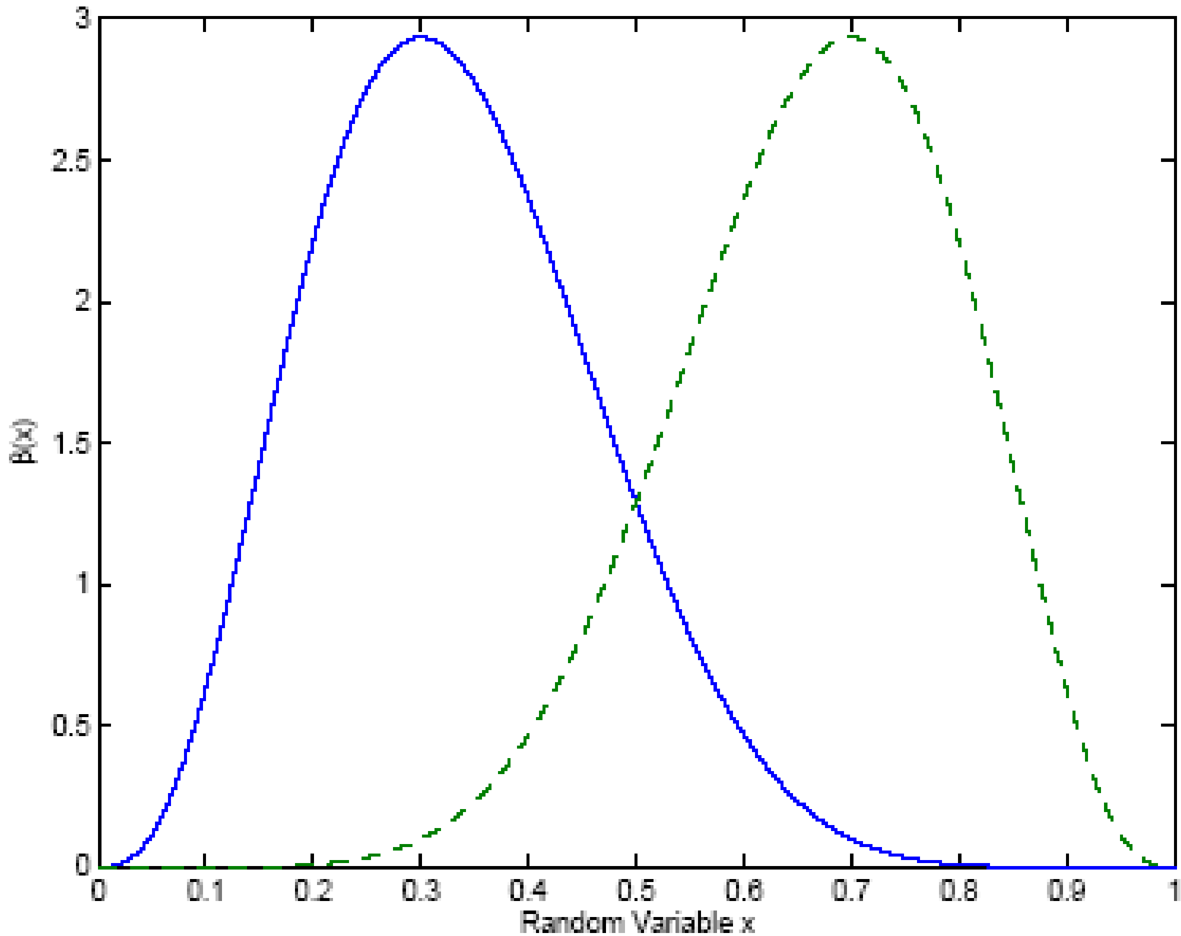
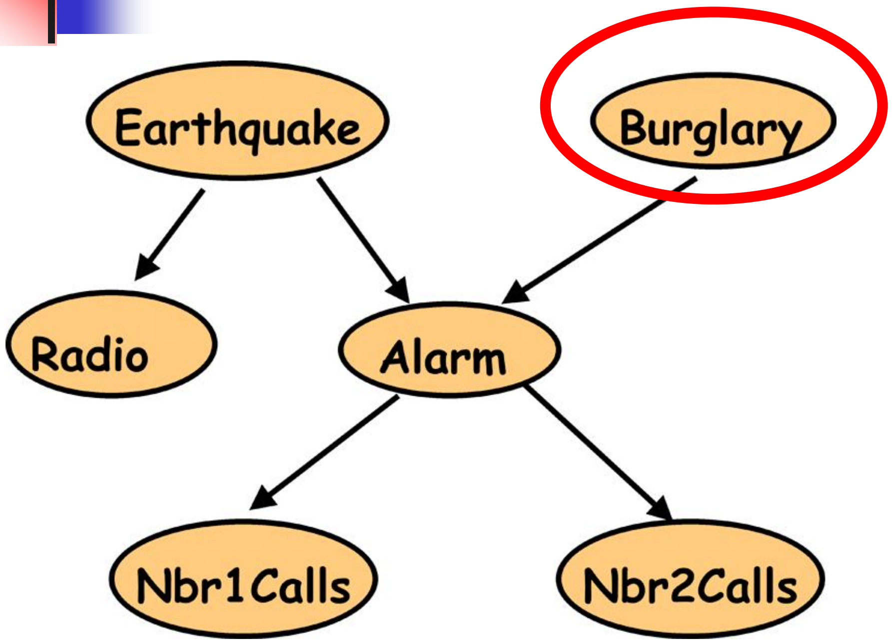


Figure 3. Beta distributions with $a = 4$ and $b = 8$ (solid line) and with $a = 8$ and $b = 4$ (dashed line). To get a higher peak (and stronger skew), use a and b that sum to a higher value.

Beta Distribution

- Example: Flip coin with Beta distribution as prior p [prob(heads)]
  - Parameterized by two positive numbers: a and b
  - Mode of distribution (E[p]) is a /(a+b)
  - Specify our prior belief for p = a / (a+b)
  - Specify confidence with initial values of a and b
- Updating our prior belief based on data by
  - Increment a for each head outcome
  - Increment b for each tail outcome
- Posterior is a binomial distribution!

Parameter Estimate for Bayesian Networks



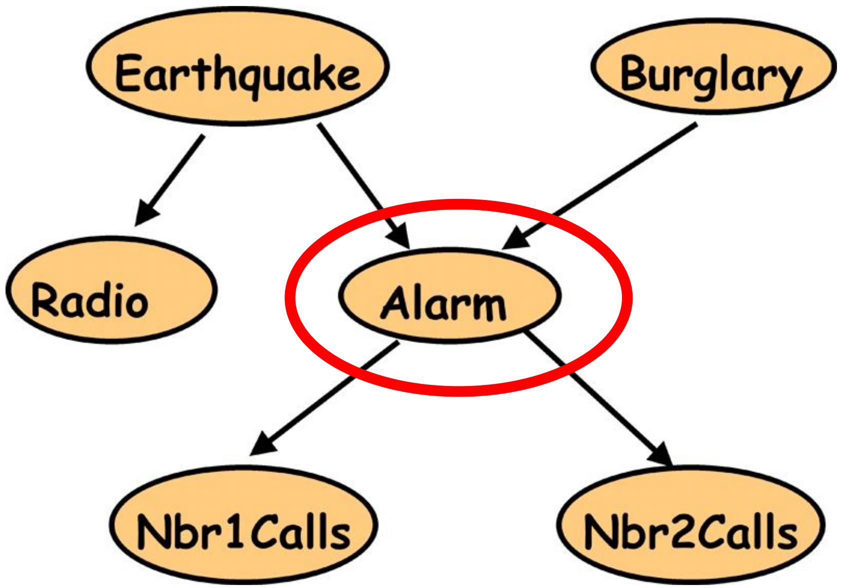
B
F
F
T
F
T

Prior

P(B) = Beta(1,4) + data = (3,7)

B	$\neg B$
.3	.7

Parameter Estimate for Bayesian Networks



E	B
T	F
F	F
F	T
F	F
F	T
…	

A
T
F
T
T
F

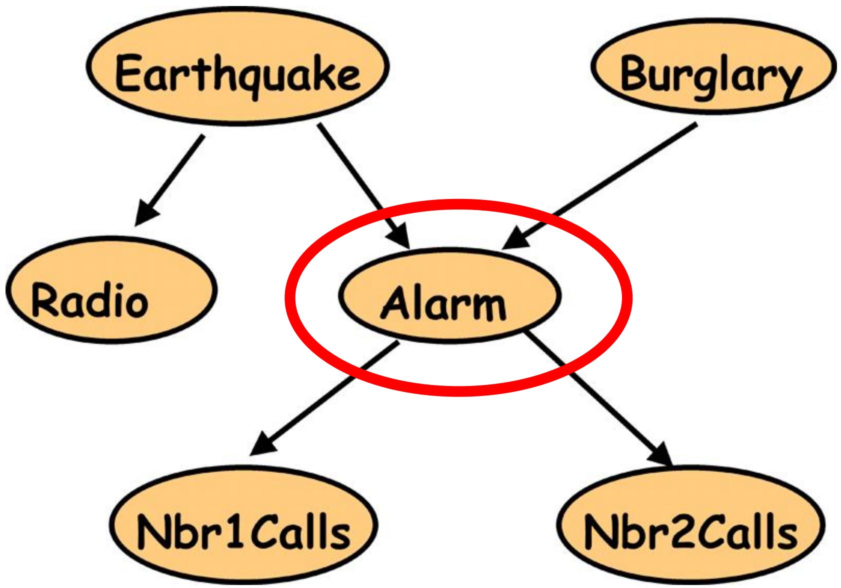
$P(A|E,B) = ?$

$P(A|E,\neg B) = ?$

$P(A|\neg E,B) = ?$

$P(A|\neg E,\neg B) = ?$

Parameter Estimate for Bayesian Networks



E	B		A
T	F		T
F	F		F
F	T		T
F	F		T
F	T		F
…			

P(A|E,B) = ? Prior

P(A|E,¬B) = ?

P(A|¬E,B) = Beta(2,3)+ data= (3,4)

P(A|¬E,¬B) = ?

General EM Framework: Handling Missing Values

- Given: Data with missing values, space of possible models, initial model

- Repeat until no change greater than threshold:
  - Expectation (E) Step: Compute expectation over missing values, given model.
  - Maximization (M) Step: Replace current model with model that maximizes probability of data.

"Soft" EM vs. "Hard" EM

- Soft EM: Expectation is a probability distribution
- Hard EM: Expectation is "all or nothing," assign most likely/probable value
- Advantage of hard EM is computational efficiency when expectation is over state consisting of values for multiple variables

EM for Parameter Learning: E Step

- For each data point with missing values
  - Compute the probability of each possible completion of that data point
  - Replace the original data point with all completions, weighted by probabilities
- Computing the probability of each completion (expectation) is just answering query over missing variables given others

EM For Parameter Learning: M Step

- Use the completed data set to update our Beta/Dirichlet distributions
  - Same as if complete data set
  - Note: Counts may be fractional now

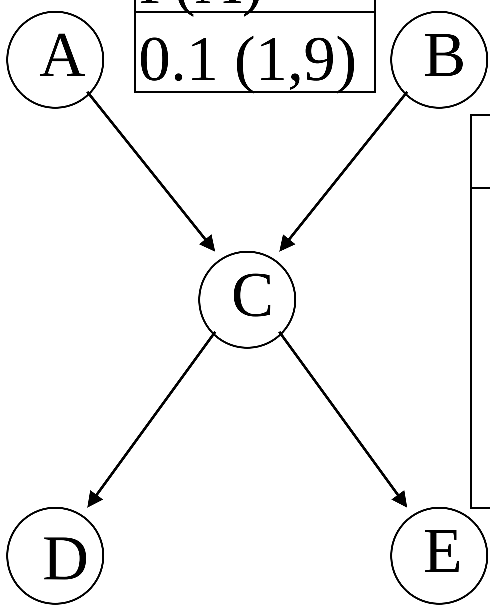
- Update CPTs based on new Beta/Dirichlet distribution
  - Same as if complete data set

Subtlety for Parameter Learning

- Overcounting based on number of iterations required to converge to settings for the missing values

- After each E step, reset all Beta/Dirichlet distributions before repeating M step.

EM for Parameter Learning



P(A)
0.1 (1,9)

P(B)
0.2 (1,4)

A B	P(C)
T T	0.9 (9,1)
T F	0.6 (3,2)
F T	0.3 (3,7)
F F	0.2 (1,4)

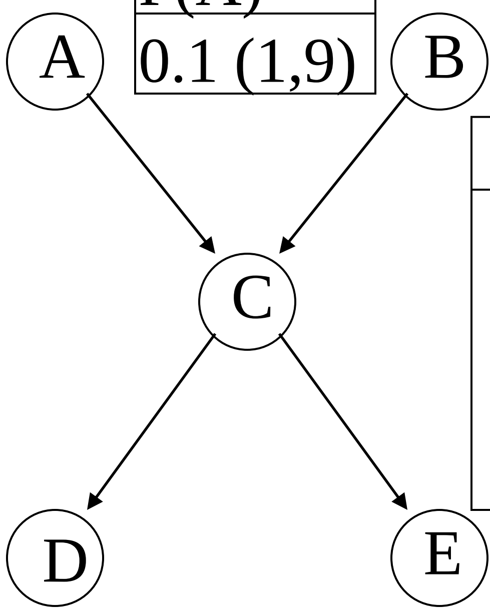
C	P(D)
T	0.9 (9,1)
F	0.2 (1,4)

C	P(E)
T	0.8 (4,1)
F	0.1 (1,9)

Data

A	B	C	D	E
0	0	?	0	0
0	0	?	1	0
1	0	?	1	1
0	0	?	0	1
0	1	?	1	0
0	0	?	0	1
1	1	?	1	1
0	0	?	0	0
0	0	?	1	0
0	0	?	0	1

EM for Parameter Learning: E Step



P(A)
0.1 (1,9)

P(B)
0.2 (1,4)

A B	P(C)
T T	0.9 (9,1)
T F	0.6 (3,2)
F T	0.3 (3,7)
F F	0.2 (1,4)

C	P(D)
T	0.9 (9,1)
F	0.2 (1,4)

C	P(E)
T	0.8 (4,1)
F	0.1 (1,9)

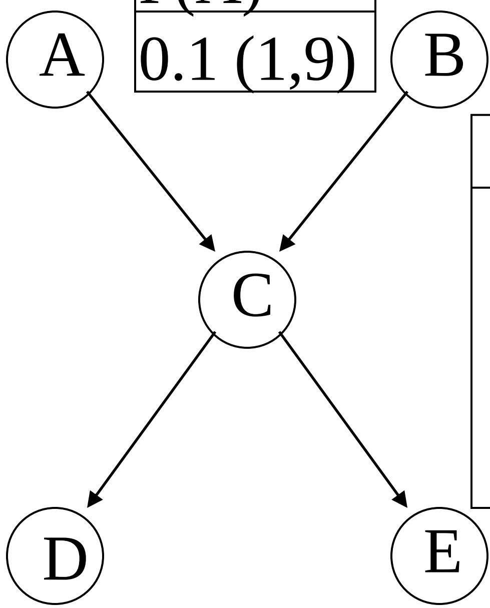
Data

A	B	C	D	E
0	0	?	0	0

P(A=0) * P(B=0) *
P(C=0 | A=0,B=0)
*P(D=0 | C=0)
*P(E=0 | C =0) = ?

P(A=0) * P(B=0) *
P(C=1 | A=0,B=0)
*P(D=0 | C=1)
*P(E=0 | C =1) = ?

EM for Parameter Learning: E Step



P(A)
0.1 (1,9)

P(B)
0.2 (1,4)

A B	P(C)
T T	0.9 (9,1)
T F	0.6 (3,2)
F T	0.3 (3,7)
F F	0.2 (1,4)

C	P(D)
T	0.9 (9,1)
F	0.2 (1,4)

C	P(E)
T	0.8 (4,1)
F	0.1 (1,9)

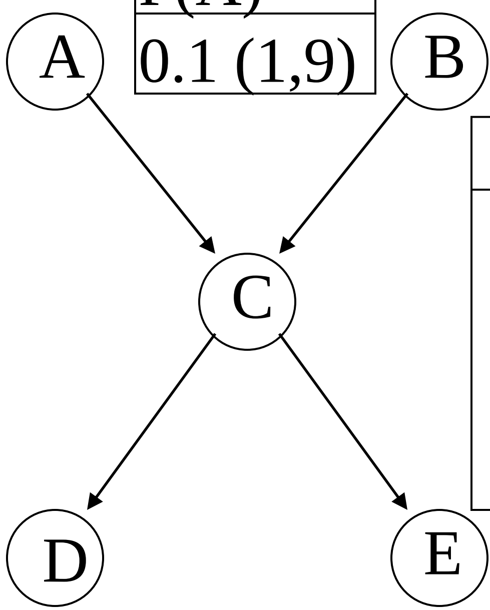
Data

A	B	C	D	E
0	0	?	0	0

$P(A=0) * P(B=0) * P(C=0 \mid A=0,B=0) * P(D=0 \mid C=0) * P(E=0 \mid C=0) = .41472$

$P(A=0) * P(B=0) * P(C=1 \mid A=0,B=0) * P(D=0 \mid C=1) * P(E=0 \mid C=1) = .00288$

EM for Parameter Learning: E Step



P(A)
0.1 (1,9)

P(B)
0.2 (1,4)

A B	P(C)
T T	0.9 (9,1)
T F	0.6 (3,2)
F T	0.3 (3,7)
F F	0.2 (1,4)

C	P(D)
T	0.9 (9,1)
F	0.2 (1,4)

C	P(E)
T	0.8 (4,1)
F	0.1 (1,9)

Data

A	B	C	D	E
0	0	?	0	0

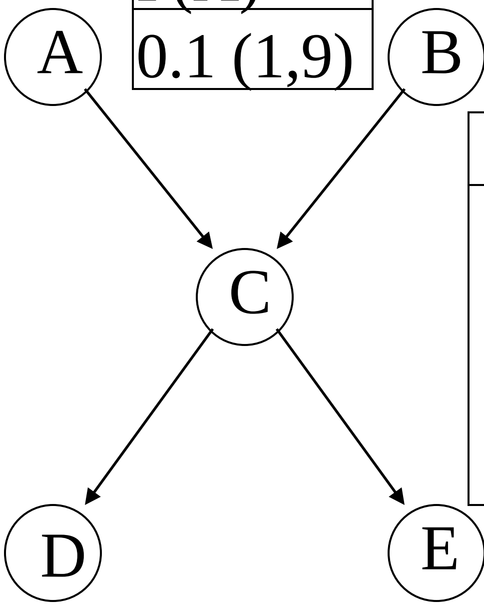
$$P(C=0) = \frac{.41472}{.4176}$$

$$P(C=0) = .99$$

$$P(C=1) = \frac{.00288}{.4176}$$

$$P(C=1) = .01$$

EM for Parameter Learning: E Step



P(A)
0.1 (1,9)

P(B)
0.2 (1,4)

A B	P(C)
T T	0.9 (9,1)
T F	0.6 (3,2)
F T	0.3 (3,7)
F F	0.2 (1,4)

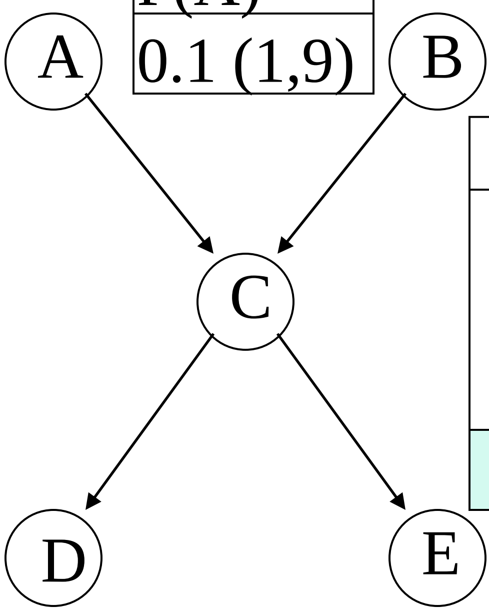
C	P(D)
T	0.9 (9,1)
F	0.2 (1,4)

C	P(E)
T	0.8 (4,1)
F	0.1 (1,9)

Data

A	B	C	D	E
0	0	0: 0.99 1: 0.01	0	0
0	0	0: 0.80 1: 0.20	1	0
1	0	0: 0.02 1: 0.98	1	1
0	0	0: 0.80 1: 0.20	0	1
0	1	0: 0.70 1: 0.30	1	0
0	0	0: 0.80 1: 0.20	0	1
1	1	0: 0.003 1: 0.997	1	1
0	0	0: 0.99 1: 0.01	0	0
0	0	0: 0.80 1: 0.20	1	0
0	0	0: 0.80 1: 0.20	0	1

EM for Parameter Learning: M Step



P(A)
0.1 (1,9)

P(B)
0.2 (1,4)

A B	P(C)
T T	0.9 (9,1)
T F	0.6 (3,2)
F T	0.3 (3,7)
F F	0.2 (1,4)

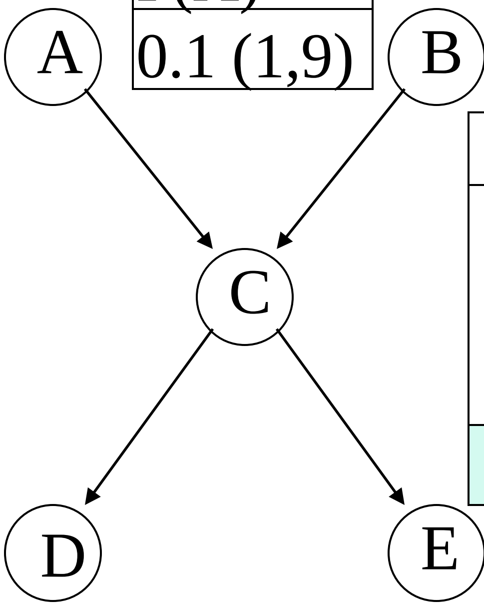
C	P(D)
T	0.9 (9,1)
F	0.2 (1,4)

C	P(E)
T	0.8 (4,1)
F	0.1 (1,9)

Data

A	B	C	D	E	C = 0
0	0	0: 0.99 1: 0.01	0	0	4+
0	0	0: 0.80 1: 0.20	1	0	.99 +
1	0	0: 0.02 1: 0.98	1	1	.8 +
0	0	0: 0.80 1: 0.20	0	1	.8 +
0	1	0: 0.70 1: 0.30	1	0	.8 +
0	0	0: 0.80 1: 0.20	0	1	.99+
1	1	0: 0.003 1: 0.997	1	1	.8+
0	0	0: 0.99 1: 0.01	0	0	.8 +
0	0	0: 0.80 1: 0.20	1	0	=
0	0	0: 0.80 1: 0.20	0	1	9.98

EM for Parameter Learning: M Step



P(A)
0.1 (1,9)

P(B)
0.2 (1,4)

A B	P(C)
T T	0.9 (9,1)
T F	0.6 (3,2)
F T	0.3 (3,7)
F F	0.2 (1,4)

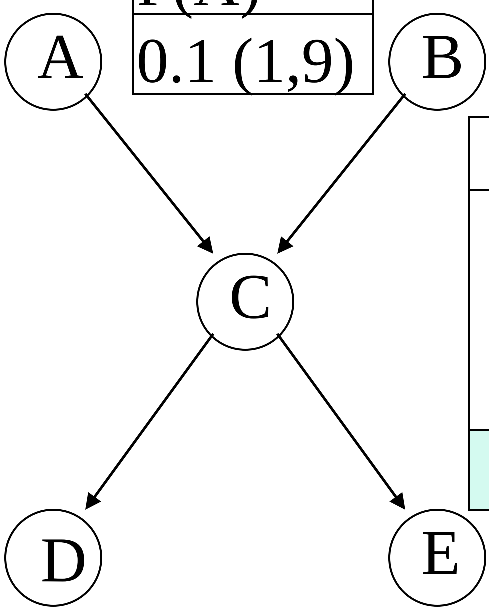
C	P(D)
T	0.9 (9,1)
F	0.2 (1,4)

C	P(E)
T	0.8 (4,1)
F	0.1 (1,9)

Data

A	B	C	D	E	C = 1
0	0	0: 0.99 1: 0.01	0	0	1+
0	0	0: 0.80 1: 0.20	1	0	.01 +
1	0	0: 0.02 1: 0.98	1	1	.2 +
0	0	0: 0.80 1: 0.20	0	1	.2 +
0	1	0: 0.70 1: 0.30	1	0	.2 +
0	0	0: 0.80 1: 0.20	0	1	.01+
1	1	0: 0.003 1: 0.997	1	1	.2+
0	0	0: 0.99 1: 0.01	0	0	.2 +
0	0	0: 0.80 1: 0.20	1	0	=
0	0	0: 0.80 1: 0.20	0	1	2.02

EM for Parameter Learning: M Step



P(A)
0.1 (1,9)

P(B)
0.2 (1,4)

A B	P(C)
T T	0.9 (9,1)
T F	0.6 (3,2)
F T	0.3 (3,7)
F F	0.2 (1,4)

$\longrightarrow$ 2.02/(2.02 + 9.98) = .16

C	P(D)
T	0.9 (9,1)
F	0.2 (1,4)

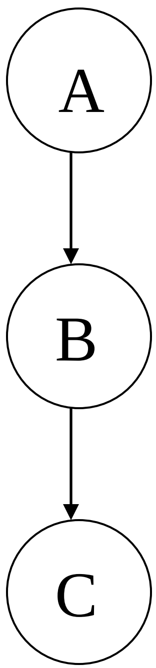
C	P(E)
T	0.8 (4,1)
F	0.1 (1,9)

Problems with EM

- Only local optimum
- Deterministic: Uniform priors can cause issues
  - See next slide
  - Use randomness to overcome this problem

What will EM do here?

P(A)
0.5 (1,1)



A	P(B)
T	0.5 (1,1)
F	0.5 (1,1)

B	P(C)
T	0.5 (1,1)
F	0.5 (1,1)

Data

A	B	C
0	?	0
1	?	1
0	?	0
1	?	1
0	?	0
1	?	1

Outline

- Probability overview
- Naïve Bayes
- Bayesian learning
- Bayesian networks
  - Representation
  - Inference
  - Parameter learning
  - Structure learning

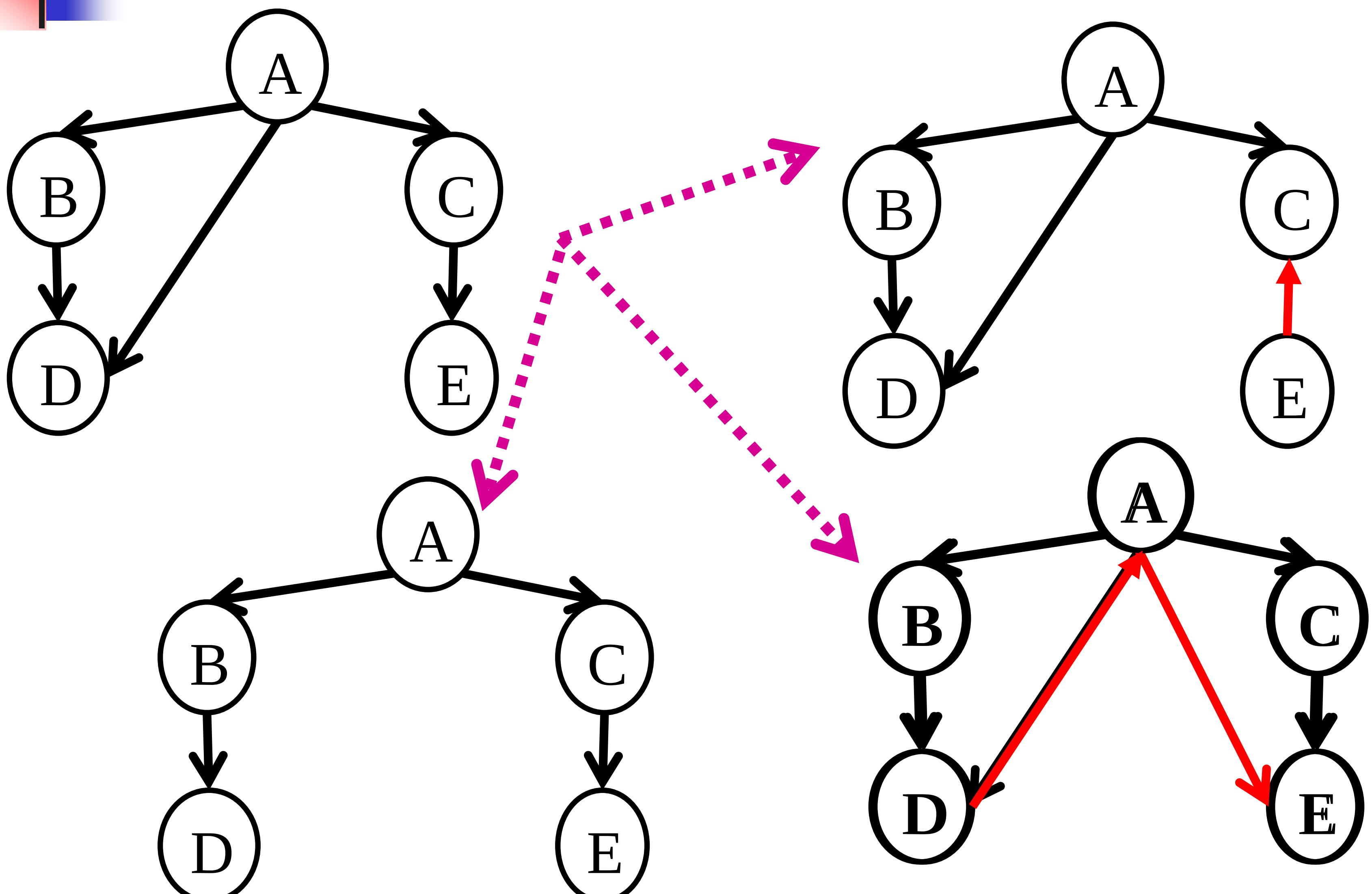
Learning the Structure of a Bayesian Network

- Search through the space of possible structures
- For each structure, learn parameters
- Pick the one that fits observed data the best
  - Problem: Will get a fully connected structure?
  - Solution: Add a penalty term
- Problem?
  - Exponential number of networks!
  - Exhaustive search infeasible
- What now?

Structure Learning as Search

- Local search
  - Start with some network structure
  - Try to make a change:
    Add, delete or reverse an edge
  - See if the new structure is better
- What should the initial state be
  - Uniform prior over random networks?
  - Based on prior knowledge
  - Empty network?
- How do we evaluate networks?

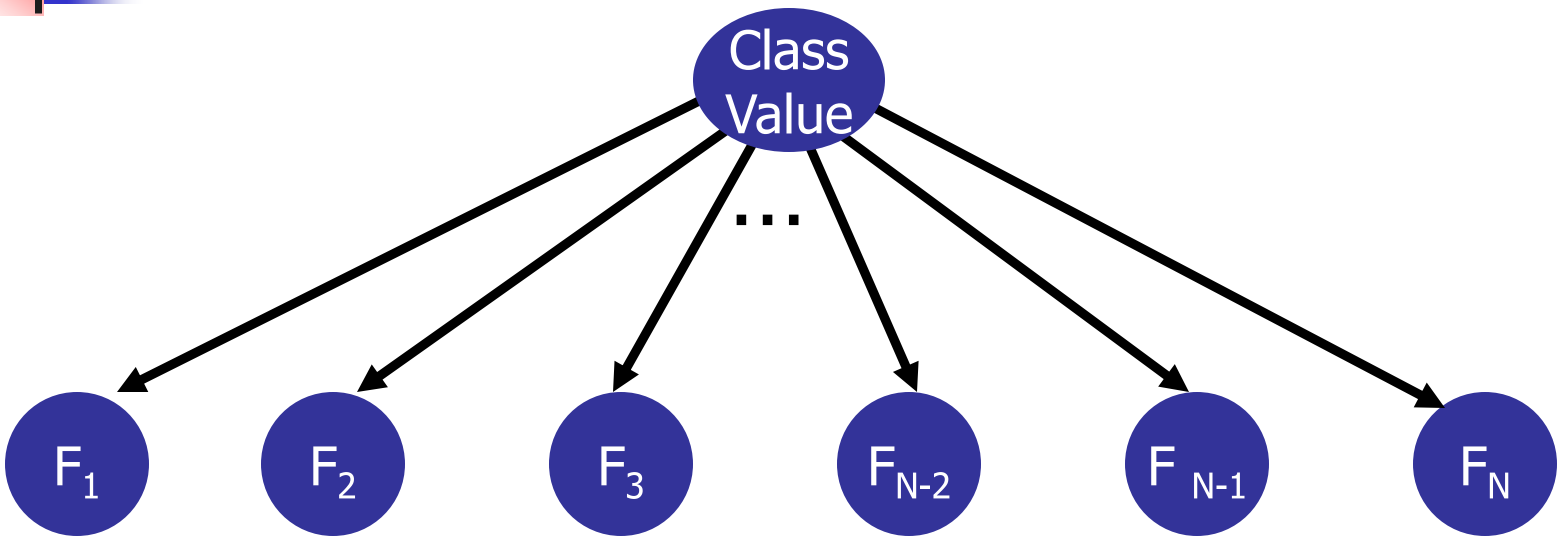
Structure Search Example



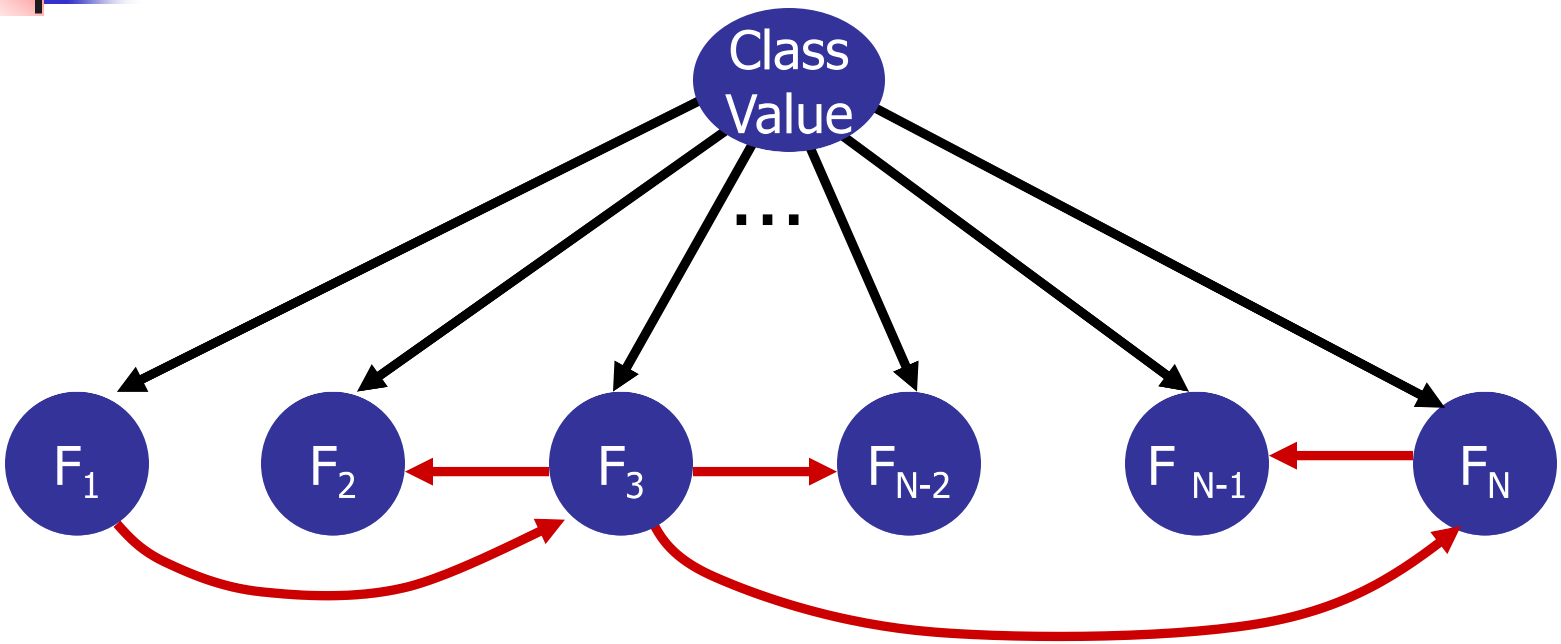
Score Functions

- Bayesian Information Criteion (BIC)
  - P(D | BN) – penalty
  - Penalty = ½ (# parameters) Log (# data points)
- MAP score
  - P(BN | D) = P(D | BN) P(BN)
  - P(BN) must decay exponential with # of parameters for this to work well
- Note: We use log P(D | BN)

Recall: Naïve Bayes



Tree Augmented Naïve Bayes (TAN) [Friedman,Geiger & Goldszmidt 1997]



Models limited set of dependencies
Guaranteed to find best structure
Runs in polynomial time

Tree-Augmented Naïve Bayes

- Each feature has at most one parent in addition to the class attribute
- For every pair of features, compute the conditional mutual information

  $I_{cm}(x;y|c) = \Sigma_{x,y,c} P(x,y,c) \log [p(x,y|c)/[p(x|c)*p(y|c)]]$
- Add arcs between all pairs of features, weighted by this value
- Compute the maximum weight spanning tree, and direct arcs from the root
- Compute parameters as already seen

Next Class

- Proposition rule induction
- First-order rule induction
- Read Mitchell Chapter 10

Summary

- Homework 2 is now available
- Naïve Bayes: Reasonable, simple baseline
- Different ways to incorporate prior beliefs
- Bayesian networks are an efficient way to represent joint distributions
  - Representation
  - Inference
  - Learning

Questions?