Machine Learning 10-601

Tom M. Mitchell
Machine Learning Department
Carnegie Mellon University

October 16, 2017

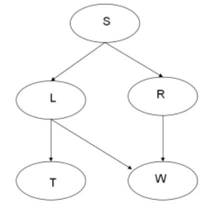
Today:

- Graphical models
- Inference
  - Markov Blanket
  - Gibbs sampling
- Learning
  - fully observed: MLE, MAP
  - partly observed data: EM

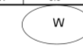
Readings:

- Bishop chapter 8, 9-9.2

Bayesian Networks Definition



Parents	P(W|Pa)	P(¬W|Pa)
L, R	0	1.0
L, ¬R	0	1.0
¬L, R	0.2	0.8
¬L, ¬R	0.9	0.1



A Bayes network represents the joint probability distribution over a collection of random variables

A Bayes network is a directed acyclic graph and a set of conditional probability distributions (CPD's)

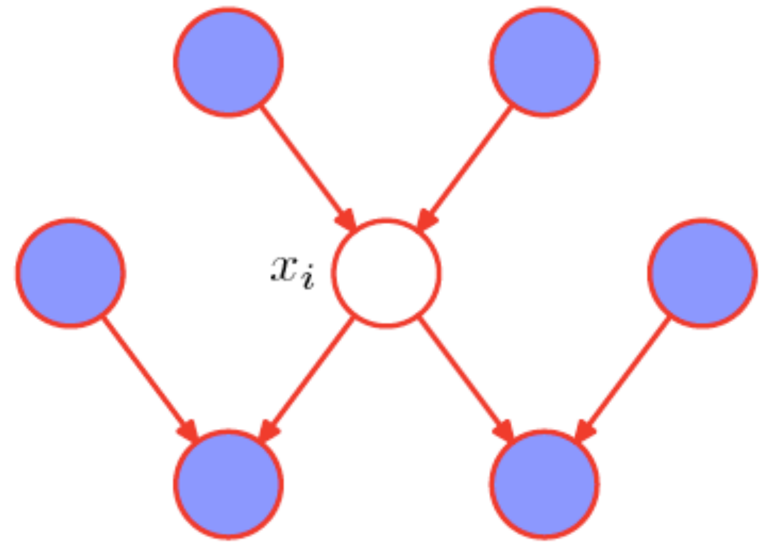
- Each node denotes a random variable
- Edges denote dependencies
- For each node X_i its CPD defines $P(X_i \,/\, Pa(X_i))$
- The joint distribution over all variables is defined to be

$$P(X_1 \ldots X_n) = \prod_i P(X_i | Pa(X_i))$$

Pa(X) = immediate parents of X in the graph

Markov Blanket

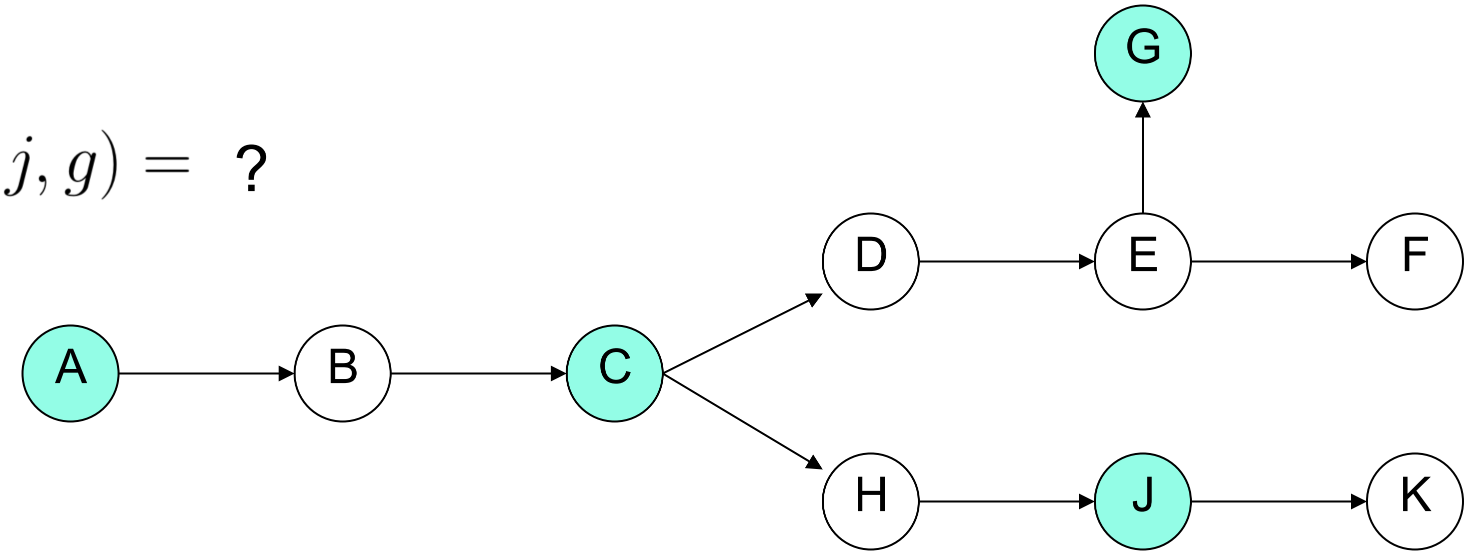
The Markov blanket of a node $\mathbf{x}_i$ comprises the set of parents, children and co-parents of the node. It has the property that the conditional distribution of $\mathbf{x}_i$, conditioned on all the remaining variables in the graph, is dependent only on the variables in the Markov blanket.



from [Bishop, 8.2]

Why Markov Blanket is Useful for Inference

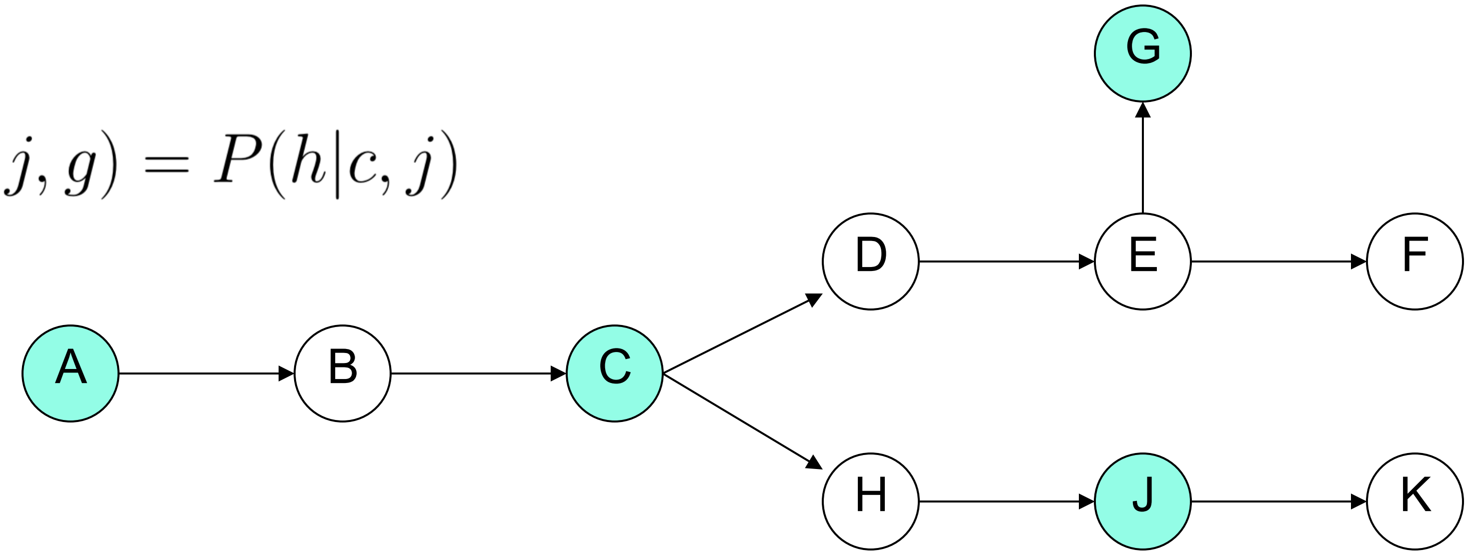
$$P(h|a, c, j, g) = \text{ ?}$$



let's use shorthand P(a) to represent P(A=a)

Why Markov Blanket is Useful for Inference

$$P(h|a, c, j, g) = P(h|c, j)$$

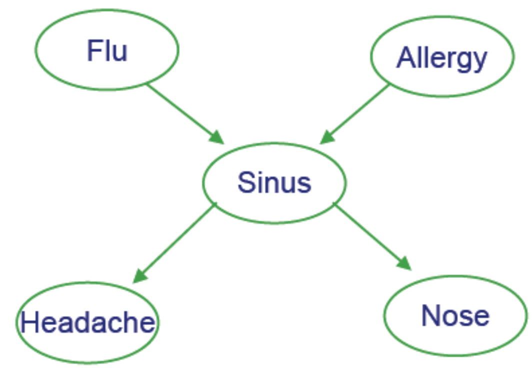


$$P(h|c, j) = \frac{P(h, c, j)}{P(c, j)} = \frac{P(c)P(h|c)P(j|h)}{P(c)P(h|c)P(j|h) + P(c)P(\neg h|c)P(j|\neg h)}$$

$$= \frac{P(h|c)P(j|h)}{P(h|c)P(j|h) + P(\neg h|c)P(j|\neg h)}$$

let's use shorthand P(a) to represent P(A=a)

Generating a sample from joint distribution: easy



How can we generate random samples drawn according to P(F,A,S,H,N)?

To generate a random sample for roots of network (F or A):

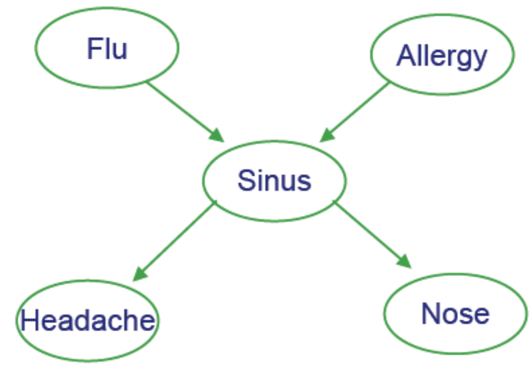
1. let θ = P(F=1) # look up from CPD
2. r = random number drawn uniformly between 0 and 1
3. if $r<\theta$ then output F=1, else F=0

To generate a random sample for S, given F,A:

1. let θ = P(S=1|F=f,A=a) # use f, a from above step
2. r = random number drawn uniformly between 0 and 1
3. if $r<\theta$ then output S=1, else S=0

Continue, generating H|S=s and N|S=s

Generating a sample from joint distribution: easy



Note we can estimate anything!

e.g., $P(N=n)$ by generating many samples

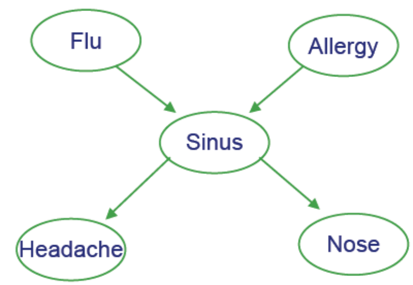
from joint distribution, then count the fraction of samples for which $N=n$

Similarly, for anything else we care about

$P(F=1|H=1, N=0)$

→ weak but general method for estimating any probability term…

Gibbs Sampling:



Goal: Directly sample conditional distributions

$$P(X_1,\ldots,X_n \mid X_{n+1}, \ldots, X_m)$$

Approach:

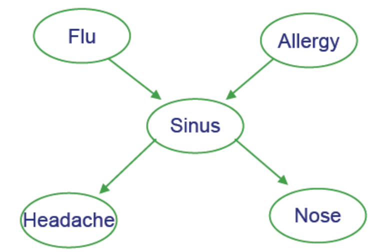
- start with arbitrary initial values for unobserved $X_1^{(0)},\ldots,X_n^{(0)}$ (and the observed $X_{n+1}, \ldots, X_m$)
- iterate for s=0 to a big number:

$$X_1^{s+1} \sim P(X_1 | X_2^s, X_3^s \ldots X_n^s, X_{n+1}, \ldots X_m)$$
$$X_2^{s+1} \sim P(X_2 | X_1^{s+1}, X_3^s \ldots X_n^s, X_{n+1}, \ldots X_m)$$
$$\ldots$$
$$X_n^{s+1} \sim P(X_n | X_1^{s+1}, X_2^{s+1}, \ldots X_{n-1}^{s+1}, X_{n+1}, \ldots X_m)$$

Eventually (after burn-in), the collection of samples will constitute a sample of the true $P(X_1,\ldots,X_n \mid X_{n+1}, \ldots, X_m)$

* but often use every 100th sample, since iters not independent

Gibbs Sampling:

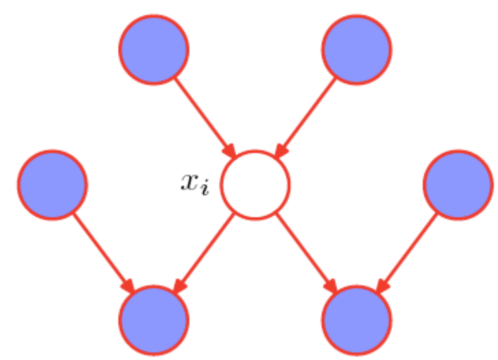


Approach:

- start with arbitrary initial values for $X_1^{(0)},\ldots,X_n^{(0)}$ (and observed $X_{n+1}, \ldots, X_m$)
- iterate for s=0 to a big number:

$$X_1^{s+1} \sim P(X_1 | X_2^s, X_3^s \ldots X_n^s, X_{n+1}, \ldots X_m)$$
$$X_2^{s+1} \sim P(X_2 | X_1^{s+1}, X_3^s \ldots X_n^s, X_{n+1}, \ldots X_m)$$
$$\ldots$$
$$X_n^{s+1} \sim P(X_n | X_1^{s+1}, X_2^{s+1}, \ldots X_{n-1}^{s+1}, X_{n+1}, \ldots X_m)$$

Only need Markov Blanket at each step!



Gibbs is special case of Markov Chain Monte Carlo method

Inference in Bayes Nets

- In general, intractable (NP-complete)
- For certain cases, tractable
  - Assigning probability to fully observed set of variables
  - Or if just one variable unobserved
  - Or for singly connected graphs (ie., no undirected loops)
    - Variable elimination
- Can often use Monte Carlo methods
  - Generate many samples, then count up the results
  - Gibbs sampling (example of Markov Chain Monte Carlo)

- Many other approaches
  - Variational methods for tractable approximate solutions
  - Junction tree, Belief propagation, …

see Graphical Models course 10-708

What you should know: Inference in Bayes Nets

- In general, intractable (NP-complete)
- Probability for a given joint assignment
- Probability for one unobserved variable given all the others
- Conditional independence / D-separation
- Markov blanket
- How to use Markov blanket to simplify inference
- How to generate samples from joint distribution
  - and how to use samples to estimate anything
- Gibbs sampling

Learning Bayes Nets from Data

Learning of Bayes Nets

- Four categories of learning problems
  - Graph structure may be known/unknown
  - Variable values may be fully observed / partly unobserved

- Easy case: learn parameters when graph structure is *known*, and training data is *fully observed*

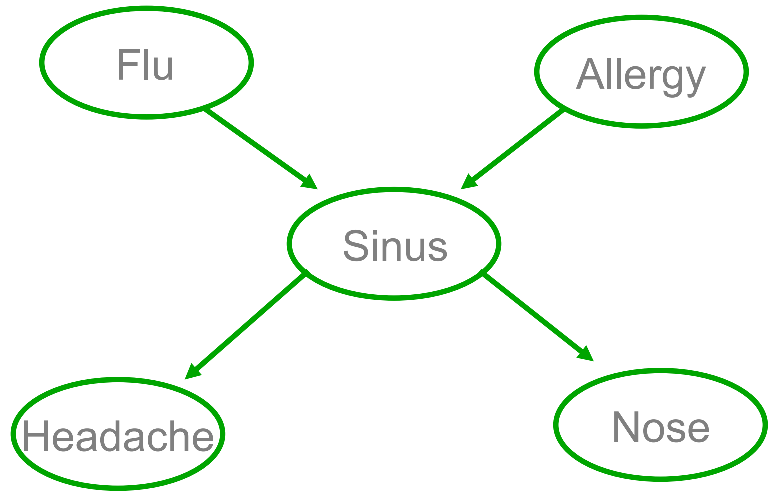
- Interesting case: graph *known*, data *partly observed*

- Gruesome case: graph structure *unknown*, data *partly unobserved*

Learning CPTs from Fully Observed Data

- Example: Consider learning the parameter

$$\theta_{s|ij} \equiv P(S=1|F=i, A=j)$$



- MLE (Max Likelihood Estimate) is

$$\theta_{s|ij} = \frac{\sum_{k=1}^{K} \delta(f_k = i, a_k = j, s_k = 1)}{\sum_{k=1}^{K} \delta(f_k = i, a_k = j)}$$

k^{th} training example

δ(X) = 1 if X is true

0 otherwise

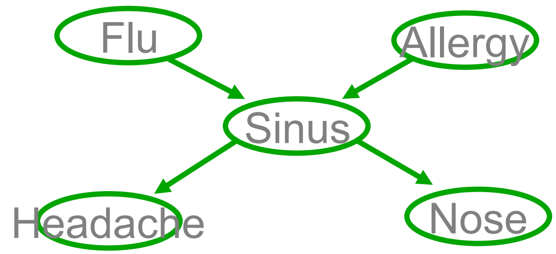
- Remember why?

let's use a_k to represent value of A on the kth example

MLE estimate of $\theta_{s|ij}$ from fully observed data

- Maximum likelihood estimate

$$\theta \leftarrow \arg\max_{\theta} \log P(data|\theta)$$



- Our case:

$$P(data|\theta) = \prod_{k=1}^{K} P(f_k, a_k, s_k, h_k, n_k)$$

$$P(data|\theta) = \prod_{k=1}^{K} P(f_k)P(a_k)P(s_k|f_k a_k)P(h_k|s_k)P(n_k|s_k)$$

$$\log P(data|\theta) = \sum_{k=1}^{K} \log P(f_k) + \log P(a_k) + \log P(s_k|f_k a_k) + \log P(h_k|s_k) + \log P(n_k|s_k)$$

$$\frac{\partial \log P(data|\theta)}{\partial \theta_{s|ij}} = \sum_{k=1}^{K} \frac{\partial \log P(s_k|f_k a_k)}{\partial \theta_{s|ij}}$$

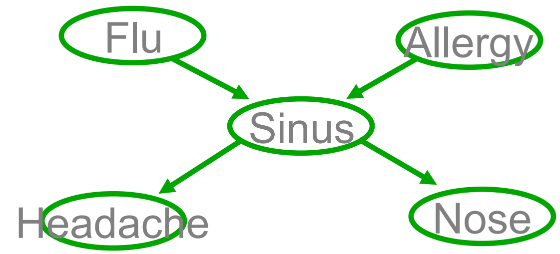
$$\theta_{s|ij} = \frac{\sum_{k=1}^{K} \delta(f_k = i, a_k = j, s_k = 1)}{\sum_{k=1}^{K} \delta(f_k = i, a_k = j)}$$

MLE for $\theta_{s|ij} = P(S=1|F=i, A=j)$ from fully observed data

- Maximum likelihood estimate

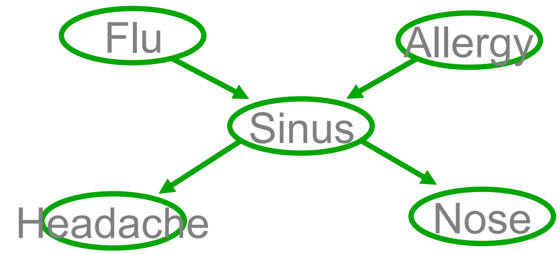
$$\theta \leftarrow \arg\max_{\theta} \log P(data|\theta)$$

$$\theta_{s|ij} = \frac{\sum_{k=1}^{K} \delta(f_k = i, a_k = j, s_k = 1)}{\sum_{k=1}^{K} \delta(f_k = i, a_k = j)}$$



like flipping coin $\sum_{k=1}^{K} \delta(f_k = i, a_k = j)$ times to see how often $s_k = 1$

MAP for $\theta_{s|ij} = P(S=1|F=i, A=j)$ from fully observed data



- Maximum likelihood estimate

$$\theta \leftarrow \arg\max_{\theta} \log P(data|\theta)$$

$$\theta_{s|ij} = \frac{\sum_{k=1}^{K} \delta(f_k=i, a_k=j, s_k=1)}{\sum_{k=1}^{K} \delta(f_k=i, a_k=j)}$$

- MAP estimate

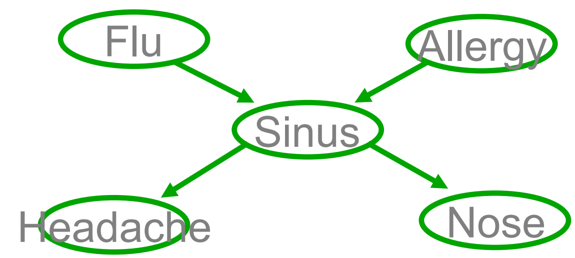
$$\theta \leftarrow \arg\max_{\theta} \log P(\theta|data) = \arg\max_{\theta} \log \left[P(data|\theta)P(\theta)\right]$$

If assume prior $P(\theta_{s|ij}) = Beta(\beta_1, \beta_0) = \frac{1}{B(\beta_1,\beta_0)} \theta_{s|ij}^{\beta_1 - 1}(1-\theta_{s|ij})^{\beta_0 - 1}$

$$\theta_{s|ij} = \frac{(\beta_1 - 1) + \sum_{k=1}^{K} \delta(f_k=i, a_k=j, s_k=1)}{(\beta_1 - 1) + (\beta_0 - 1) + \sum_{k=1}^{K} \delta(f_k=i, a_k=j)}$$

like coin flipping, including hallucinated examples

Estimate θ from partly observed data



- What if FAHN observed, but not S?
- Can't calculate MLE

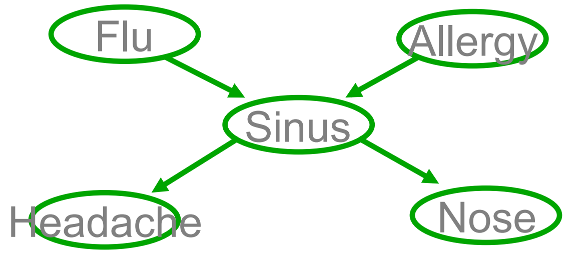
$$\theta \leftarrow \arg\max_{\theta} \log \prod_k P(f_k, a_k, s_k, h_k, n_k | \theta)$$

- Let X be all *observed* variable values (over all examples)
- Let Z be all *unobserved* variable values
- Can't calculate MLE:

$$\theta \leftarrow \arg\max_{\theta} \log P(X, Z | \theta)$$

- WHAT TO DO?

Estimate θ from partly observed data



- What if FAHN observed, but not S?
- Can't calculate MLE

$$\theta \leftarrow \arg\max_{\theta} \log \prod_{k} P(f_k, a_k, s_k, h_k, n_k|\theta)$$

- Let X be all *observed* variable values (over all examples)
- Let Z be all *unobserved* variable values
- Can't calculate MLE:

$$\theta \leftarrow \arg\max_{\theta} \log P(X, Z|\theta)$$

- EM seeks* the estimate:

$$\theta \leftarrow \arg\max_{\theta} E_{Z|X,\theta}[\log P(X, Z|\theta)]$$

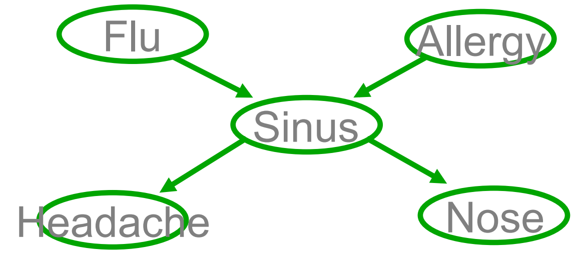
* EM guaranteed to find local maximum

Expected value

$$E_{P(X)}[f(X)] = \sum_{x} P(X = x) f(x)$$

- EM seeks estimate:

$$\theta \leftarrow \arg\max_{\theta} E_{Z|X,\theta}[\log P(X, Z|\theta)]$$



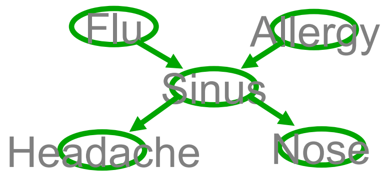
- here, observed X={F,A,H,N}, unobserved Z={S}

$$\log P(X, Z|\theta) = \sum_{k=1}^{K} \log P(f_k) + \log P(a_k) + \log P(s_k|f_k a_k) + \log P(h_k|s_k) + \log P(n_k|s_k)$$

$$E_{P(Z|X,\theta)} \log P(X, Z|\theta) = \sum_{k=1}^{K} \sum_{i=0}^{1} P(s_k = i|f_k, a_k, h_k, n_k)$$
$$[logP(f_k) + \log P(a_k) + \log P(s_k|f_k a_k) + \log P(h_k|s_k) + \log P(n_k|s_k)]$$

let's use a_k to represent value of A on the kth example

EM Algorithm - Informally



EM is a general procedure for learning from partly observed data

Given observed variables X, unobserved Z (X={F,A,H,N}, Z={S})

Begin with arbitrary choice for parameters θ

Iterate until convergence:

- E Step: use X, θ to estimate the unobserved Z values
- M Step: use X values and estimated Z values to derive a better θ

Guaranteed to find local maximum.
Each iteration increases $E_{P(Z|X,\theta)}[\log P(X, Z|\theta')]$

EM Algorithm - Precisely

EM is a general procedure for learning from partly observed data

Given observed variables X, unobserved Z (X={F,A,H,N}, Z={S})

Define $Q(\theta'|\theta) = E_{P(Z|X,\theta)}[\log P(X, Z|\theta')]$

current

M step new

Iterate until convergence:

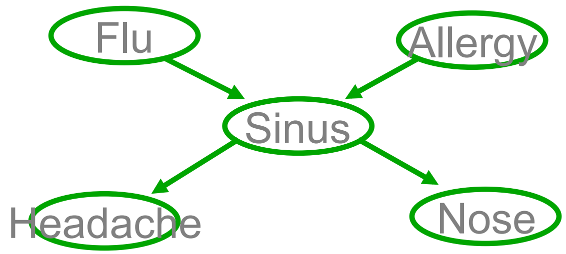
- E Step: Use X and current θ to calculate P(Z|X,θ)
- M Step: Replace current θ by

$$\theta \leftarrow \arg\max_{\theta'} Q(\theta'|\theta)$$

Guaranteed to find local maximum.
Each iteration increases $E_{P(Z|X,\theta)}[\log P(X, Z|\theta')]$

E Step: Use X, θ, to Calculate P(Z|X,θ)

observed X={F,A,H,N},
unobserved Z={S}



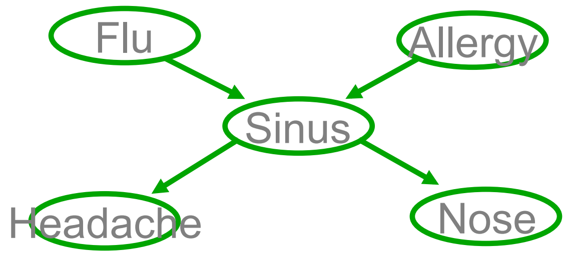
How? Bayes net inference problem.

$$P(S_k = 1 | f_k a_k h_k n_k, \theta) =$$

let's use a_k to represent value of A on the kth example

E Step: Use X, θ, to Calculate P(Z|X,θ)

observed X={F,A,H,N},
unobserved Z={S}



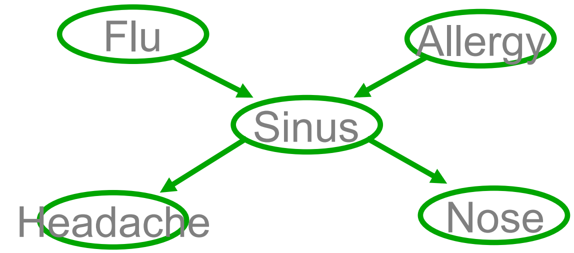
How? Bayes net inference problem.

$$P(S_k = 1 | f_k a_k h_k n_k, \theta) =$$

$$P(S_k = 1 | f_k a_k h_k n_k, \theta) = \frac{P(S_k = 1, f_k a_k h_k n_k | \theta)}{P(S_k = 1, f_k a_k h_k n_k | \theta) + P(S_k = 0, f_k a_k h_k n_k | \theta)}$$

let's use a_k to represent value of A on the kth example

EM and estimating $\theta_{s|ij}$



observed X = {F,A,H,N}, unobserved Z={S}

E step: Calculate $P(Z_k|X_k; \theta)$ for each training example, k

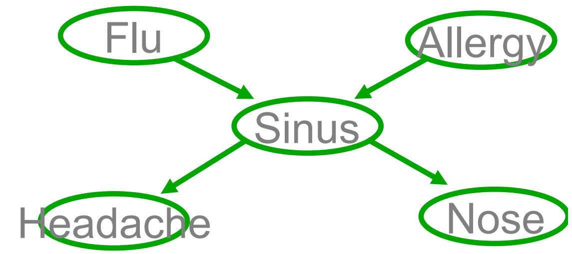
$$P(S_k = 1|f_k a_k h_k n_k, \theta) = E[s_k] = \frac{P(S_k = 1, f_k a_k h_k n_k|\theta)}{P(S_k = 1, f_k a_k h_k n_k|\theta) + P(S_k = 0, f_k a_k h_k n_k|\theta)}$$

M step: update all relevant parameters. For example:

$$\theta_{s|ij} \leftarrow \frac{\sum_{k=1}^{K} \delta(f_k = i, a_k = j)\ E[s_k]}{\sum_{k=1}^{K} \delta(f_k = i, a_k = j)}$$

Recall MLE was: $\theta_{s|ij} = \frac{\sum_{k=1}^{K} \delta(f_k = i, a_k = j, s_k = 1)}{\sum_{k=1}^{K} \delta(f_k = i, a_k = j)}$

EM and estimating θ



More generally,
Given observed set X, unobserved set Z of boolean values

E step: Calculate for each training example, k

the expected value of each unobserved variable in each training example

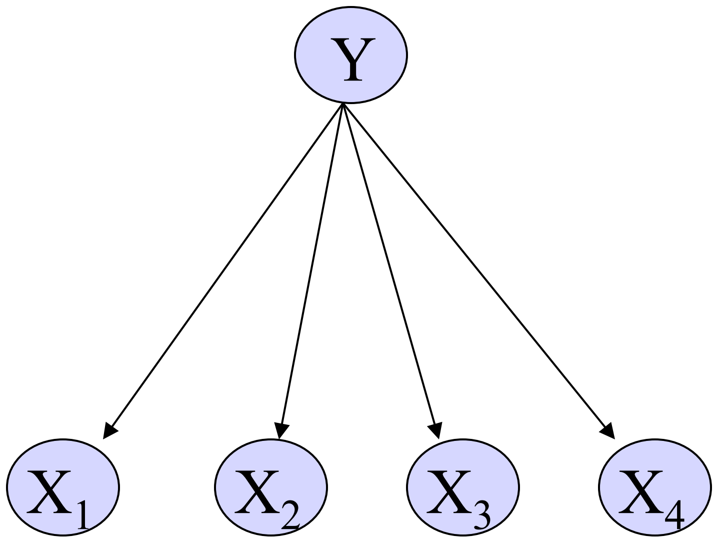
M step:

Calculate θ similar to MLE estimates, but replacing each count by its <u>expected count</u>

$$\delta(Z=1) \rightarrow E_{Z|X,\theta}[Z] \qquad \delta(Z=0) \rightarrow (1 - E_{Z|X,\theta}[Z])$$

Using Unlabeled Data to Help Train Naïve Bayes Classifier

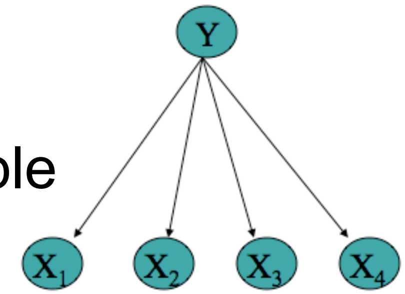
Learn P(Y|X)



Y	X1	X2	X3	X4
1	0	0	1	1
0	0	1	0	0
0	0	0	1	0
?	0	1	1	0
?	0	1	0	1

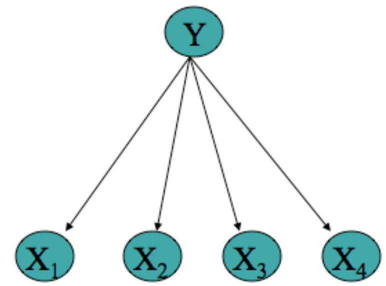
E step: Calculate for each training example, k

the expected value of each unobserved variable



Y	X1	X2	X3	X4
1	0	0	1	1
0	0	1	0	0
0	0	0	1	0
?	0	1	1	0
?	0	1	0	1

EM and estimating θ



Given observed set X, unobserved set Y of boolean values

E step: Calculate for each training example, k

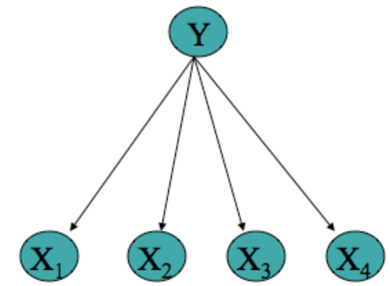
the expected value of each unobserved variable Y

$$E_{P(Y|X_1 \ldots X_N)}[y(k)] = P(y(k) = 1|x_1(k), \ldots x_N(k); \theta) = \frac{P(y(k) = 1) \prod_i P(x_i(k)|y(k) = 1)}{\sum_{j=0}^{1} P(y(k) = j) \prod_i P(x_i(k)|y(k) = j)}$$

M step: Calculate estimates similar to MLE, but replacing each count by its expected count

let's use y(k) to indicate value of Y on kth example

EM and estimating θ



Given observed set X, unobserved set Y of boolean values

E step: Calculate for each training example, k

the expected value of each unobserved variable Y

$$E_{P(Y|X_1 \ldots X_N)}[y(k)] = P(y(k)=1|x_1(k), \ldots x_N(k); \theta) = \frac{P(y(k)=1) \prod_i P(x_i(k)|y(k)=1)}{\sum_{j=0}^{1} P(y(k)=j) \prod_i P(x_i(k)|y(k)=j)}$$

M step: Calculate estimates similar to MLE, but replacing each count by its <u>expected count</u>

$$\theta_{ij|m} = \hat{P}(X_i = j|Y=m) = \frac{\sum_k P(y(k)=m|x_1(k) \ldots x_N(k)) \; \delta(x_i(k)=j)}{\sum_k P(y(k)=m|x_1(k) \ldots x_N(k))}$$

MLE would be: $\hat{P}(X_i = j|Y=m) = \dfrac{\sum_k \delta((y(k)=m) \wedge (x_i(k)=j))}{\sum_k \delta(y(k)=m)}$

- **Inputs:** Collections $\mathcal{D}^l$ of labeled documents and $\mathcal{D}^u$ of unlabeled documents.
- Build an initial naive Bayes classifier, $\hat{\theta}$, from the labeled documents, $\mathcal{D}^l$, only. Use maximum a posteriori parameter estimation to find $\hat{\theta} = \arg\max_\theta \mathrm{P}(\mathcal{D}|\theta)\mathrm{P}(\theta)$ (see Equations 5 and 6).
- Loop while classifier parameters improve, as measured by the change in $l_c(\theta|\mathcal{D};\mathbf{z})$ (the complete log probability of the labeled and unlabeled data
    - **(E-step)** Use the current classifier, $\hat{\theta}$, to estimate component membership of each unlabeled document, *i.e.*, the probability that each mixture component (and class) generated each document, $\mathrm{P}(c_j|d_i;\hat{\theta})$ (see Equation 7).
    - **(M-step)** Re-estimate the classifier, $\hat{\theta}$, given the estimated component membership of each document. Use maximum a posteriori parameter estimation to find $\hat{\theta} = \arg\max_\theta \mathrm{P}(\mathcal{D}|\theta)\mathrm{P}(\theta)$ (see Equations 5 and 6).
- **Output:** A classifier, $\hat{\theta}$, that takes an unlabeled document and predicts a class label.

From [Nigam et al., 2000]

Experimental Evaluation

- Newsgroup postings
  - 20 newsgroups, 1000/group
- Web page classification
  - student, faculty, course, project
  - 4199 web pages
- Reuters newswire articles
  - 12,902 articles
  - 90 topics categories

20 Newsgroups

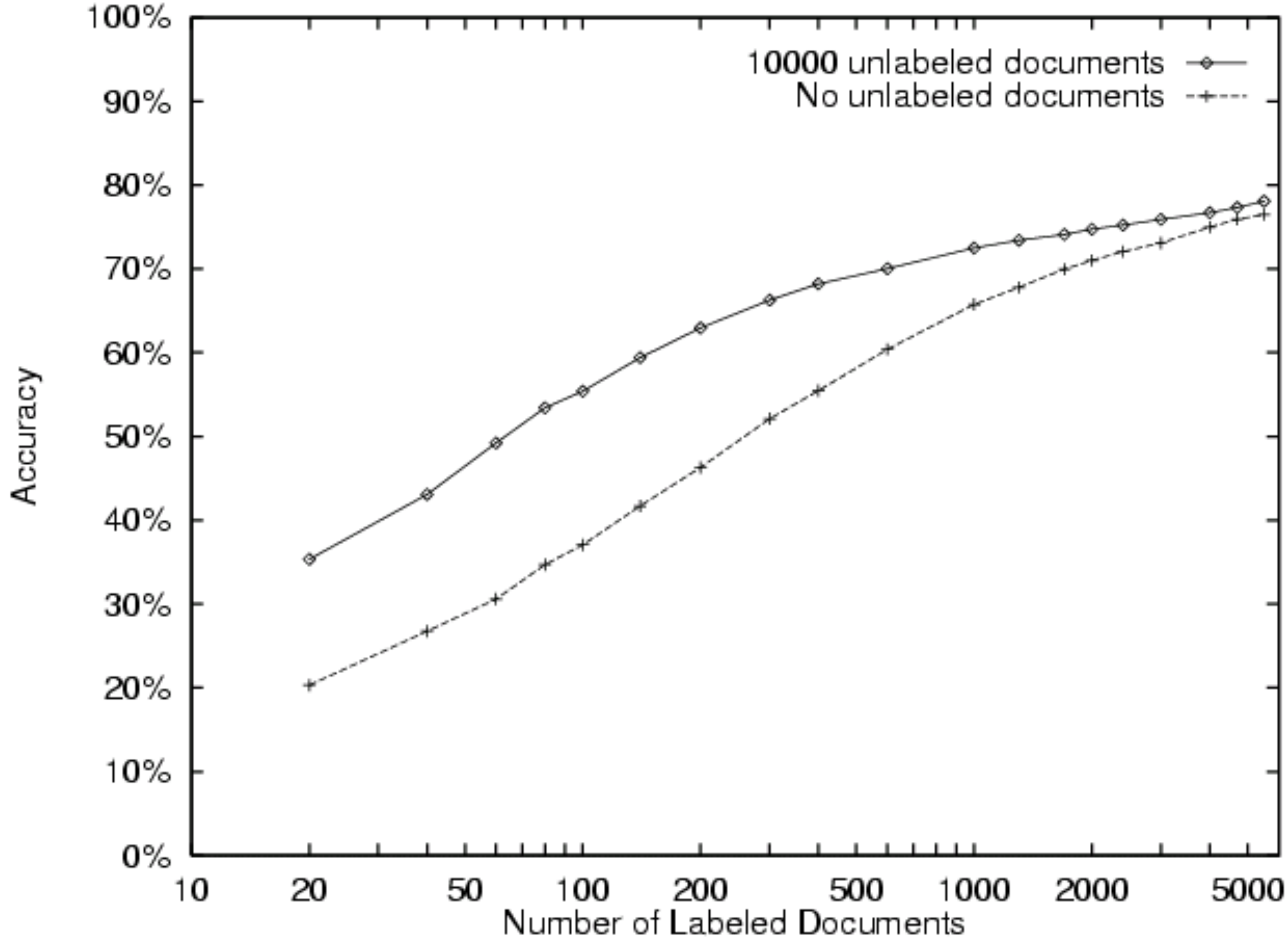


Table 3. Lists of the words most predictive of the course class in the WebKB data set, as they change over iterations of EM for a specific trial. By the second iteration of EM, many common course-related words appear. The symbol D indicates an arbitrary digit.

Iteration 0	Iteration 1	Iteration 2
intelligence	DD	D
DD	D	DD
artificial	lecture	lecture
understanding	cc	cc
DDw	$D^\star$	$DD{:}DD$
dist	$DD{:}DD$	due
identical	handout	$D^\star$
rus	due	homework
arrange	problem	assignment
games	set	handout
dartmouth	tay	set
natural	DDam	hw
cognitive	yurttas	exam
logic	homework	problem
proving	kfoury	DDam
prolog	sec	postscript
knowledge	postscript	solution
human	exam	quiz
representation	solution	chapter
field	assaf	ascii

word w ranked by
P(w|Y=course) /
P(w|Y ≠ course)

Using one labeled
example per class

What you should know about EM

- For learning from partly unobserved data
- MLE of θ = $\arg\max_{\theta} \log P(data|\theta)$
- EM estimate: θ = $\arg\max_{\theta} E_{Z|X,\theta}[\log P(X,Z|\theta)]$

  Where X is observed part of data, Z is unobserved

- EM for training Bayes networks
- Can also develop MAP version of EM
- Can also derive your own EM algorithm for your own problem
  - write out expression for $E_{Z|X,\theta}[\log P(X,Z|\theta)]$
  - E step: for each training example X^k, calculate $P(Z^k \mid X^k, \theta)$
  - M step: chose new θ to maximize $E_{Z|X,\theta}[\log P(X,Z|\theta)]$

Learning Bayes Net Structure

How can we learn Bayes Net graph structure?

In general case, open problem

- can require lots of data (else high risk of overfitting)
- can use Bayesian priors, or other kinds of prior assumptions about graph structure to constrain search

One key result:

- Chow-Liu algorithm: finds "best" tree-structured network
- What's '
  - suppo
    netwo

$$KL(P(\mathbf{X}) \mid\mid T(\mathbf{X})) \equiv \sum_k P(\mathbf{X}=k) \log \frac{P(\mathbf{X}=k)}{T(\mathbf{X}=k)}$$

  - Chow-Liu minimizes Kullback-Leibler divergence:

Chow-Liu Algorithm

Key result: To minimize KL(P || T) over possible tree networks T representing true P, it suffices to find the tree network T that maximizes the sum of mutual informations over its edges

Mutual information for an edge between variable A and B:

$$I(A,B) = \sum_a \sum_b P(a,b) \log \frac{P(a,b)}{P(a)P(b)}$$

This works because for tree networks with nodes $\mathbf{X} \equiv \langle X_1 \ldots X_n \rangle$

$$\begin{aligned} KL(P(\mathbf{X}) \mid\mid T(\mathbf{X})) &\equiv \sum_k P(\mathbf{X}=k) \log \frac{P(\mathbf{X}=k)}{T(\mathbf{X}=k)} \\ &= -\sum_i I(X_i, Pa(X_i)) + \sum_i H(X_i) - H(X_1 \ldots X_n) \end{aligned}$$

Chow-Liu Algorithm

1. for each pair of variables A,B, use data to estimate P(A,B), P(A), and P(B)

2. for each pair A, B calculate mutual information

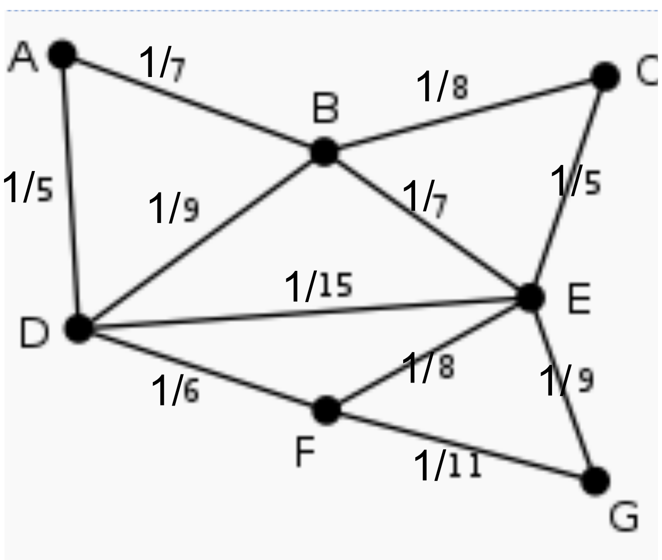
$$I(A,B) = \sum_a \sum_b P(a,b) \log \frac{P(a,b)}{P(a)P(b)}$$

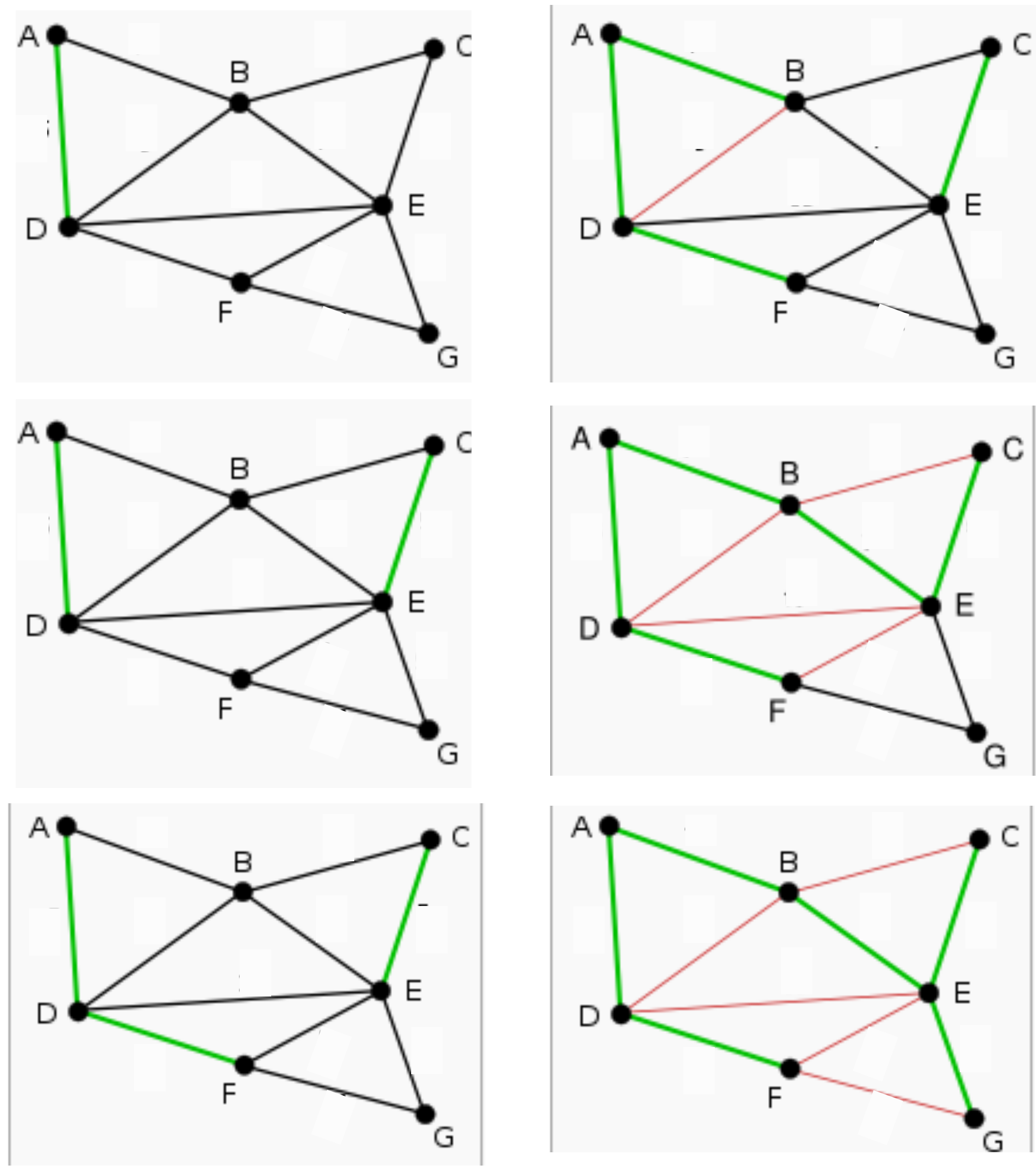
3. calculate the maximum spanning tree over the set of variables, using edge weights *I(A,B)*
   (given N vars, this costs only $O(N^2)$ time)

4. add arrows to edges to form a directed-acyclic graph

5. learn the CPD's for this graph

Chow-Liu algorithm example
Greedy Algorithm to find Max-Spanning Tree

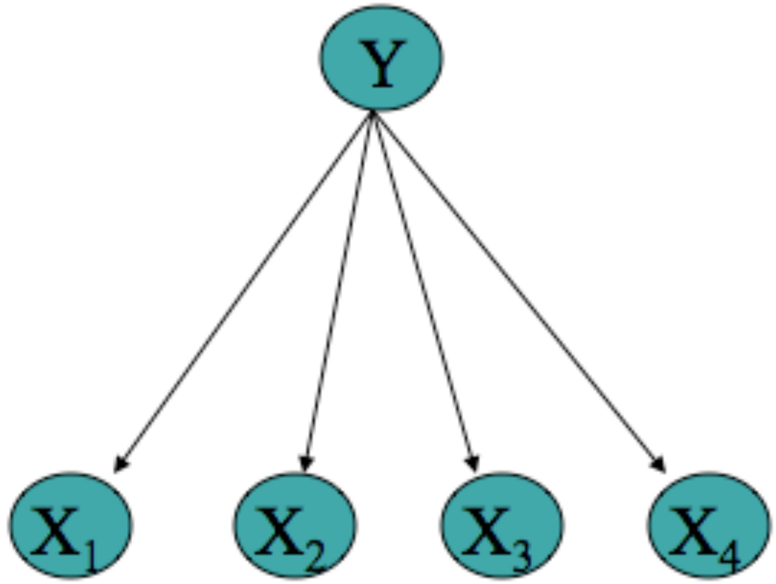




[courtesy A. Singh, C. Guestrin]

Tree Augmented Naïve Bayes

[Nir Friedman et al., 1997]



Bayes Nets – What You Should Know

- Representation
  - Bayes nets represent joint distribution as a DAG + Conditional Distributions
  - D-separation lets us decode conditional independence assumptions
- Inference
  - NP-hard in general
  - For some graphs, closed form inference is feasible
  - Approximate methods too, e.g., Monte Carlo methods, …
- Learning
  - Easy for known graph, fully observed data (MLE's, MAP est.)
  - EM for partly observed data, known graph
  - Learning graph structure: Chow-Liu for tree-structured networks
  - Hardest when graph unknown, data incompletely observed