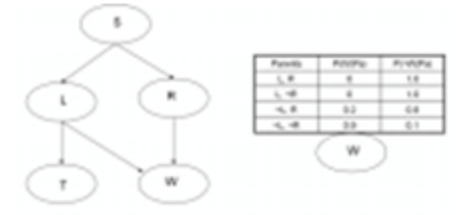
Bayes Net – Inferences and Learning

Bayesian Network Definition



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

A Bayes network is a derected acyclic graph and a set of CPD's

- Each node denotes a random variable
- Edges denote dependencies
- CPD for each node X_i define $P(X_i \mid Pa(X_i))$
- The joint distribution over all variables id defined as

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

***Pa*(*X*)=immediate parent of X in the graph**

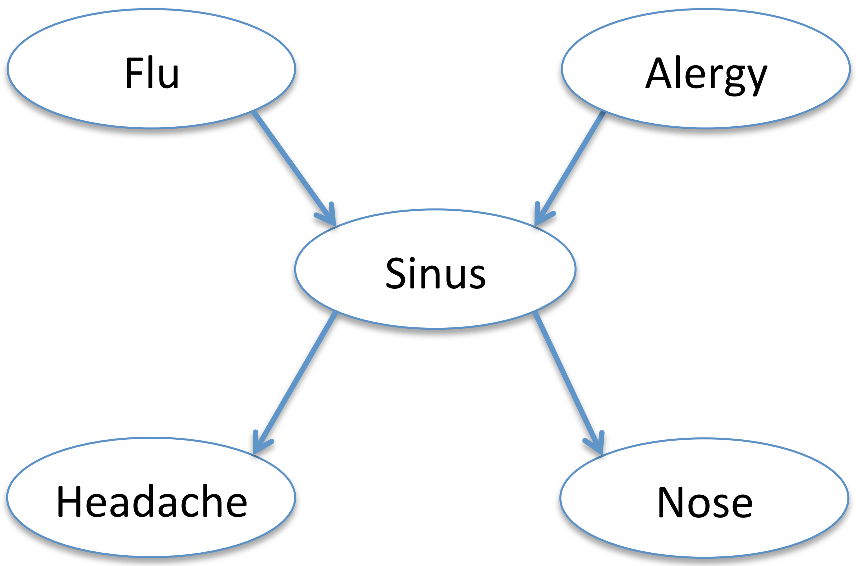
Inference in Bayes Nets

- **In general, intractable (NP-complete)**
- **For certain cases, tractable**
  - Assigning probability to fully observed set of variable
  - Or if just one variable unobserved
  - Or for singly connected graphs (ie., no undirected loops)
    - Belief propagation
- **For multiply connected graphs**
    - Junction tree
- **Sometimes use Monte Carlo methods**
  - Generate many samples according to the Bayes Net distribution, then count up the results
- **Variational methods for tractable approximate solutions**

Prob. of joint assignment : easy

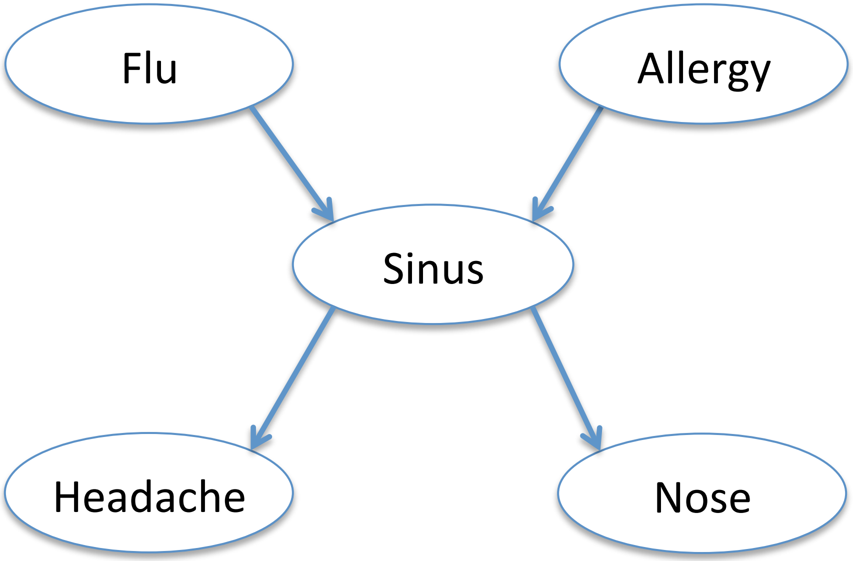
- Suppose we are interested in joint assigment <F=f, A=a, S=s, H=h, N=n>

- What is P(f,a,s,h,n) ?



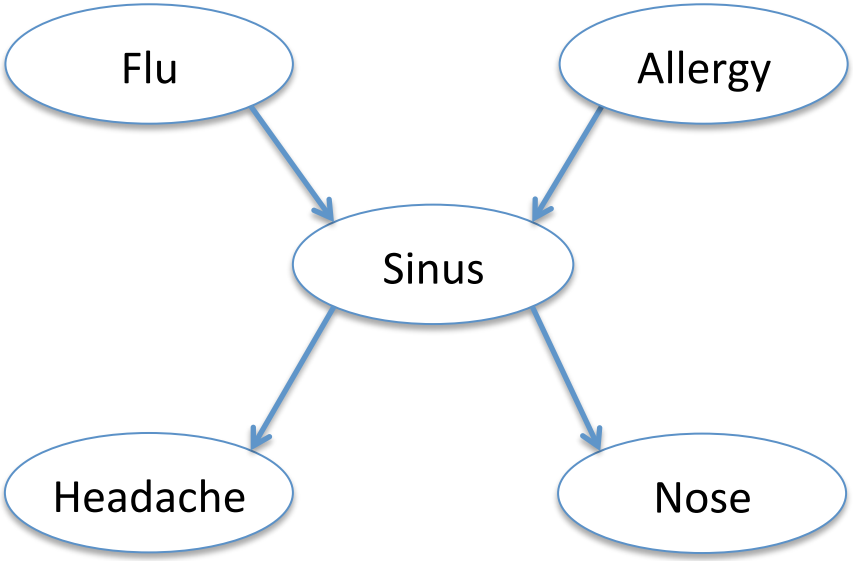
Prob. of marginals : not so easy

- How do we calculate P(N=n)?



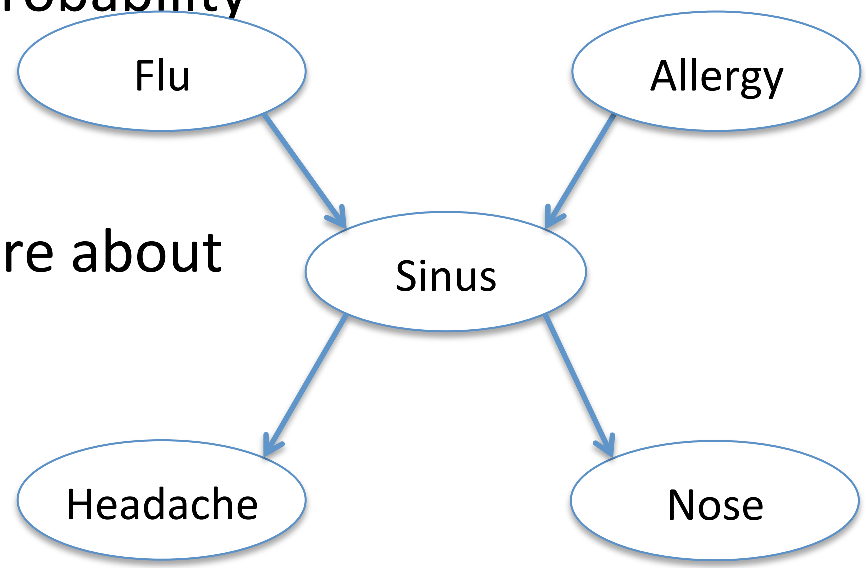
Generating a sample from joint distribution :easy

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



Generating a sample from joint distribution :easy

Note we can estimate marginals like P(N=n) by generating many samples from joint distribution, by summing the probability mass for which N=n



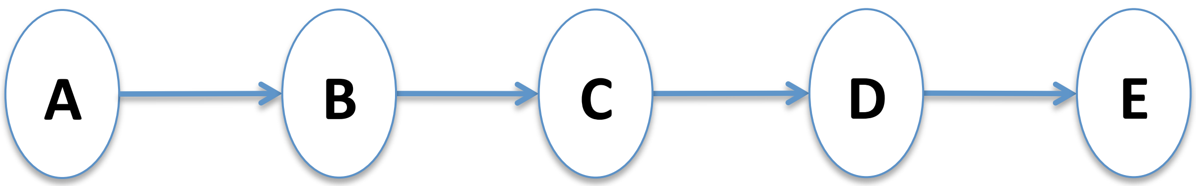
Similarity, for anything else we care about P(F=1|H=1, N=0)

→ Weak but general method for estimating any probability term …

Prob. of marginals : not so easy

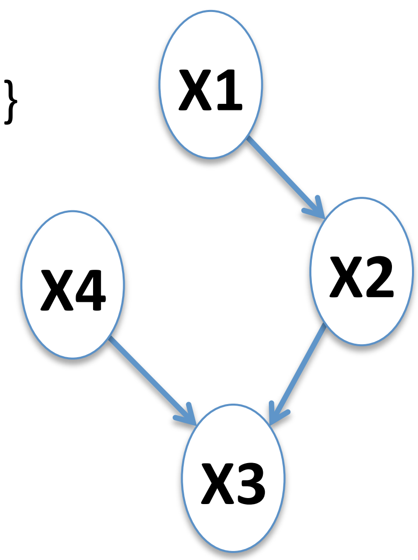
But sometimes the structure of the network allows us to be clever → avoid exponential work

e.g., chain



Conditional Independence, Revisited

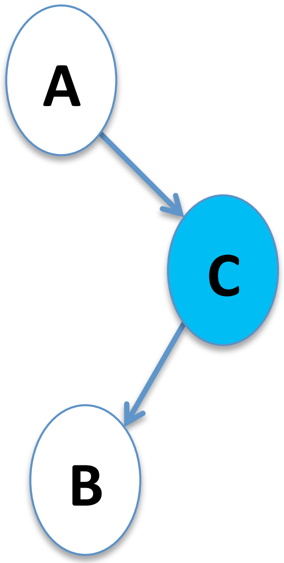
- We said :
  - Each node is conditionally independent of its non-descendents, given its immediate parents
- Does this rule given us all of the conditional independence relations implied by the Bayes network ?
  - No !
  - E.g. X1 and X4 are conditionally indep given {X2,X3}
  - But X1 and X4 not conditionally indep given X3
  - For this, we need to understand D-separation ...



Easy Network 1 : Head to Tail

Prove A cond indep of B given C ?

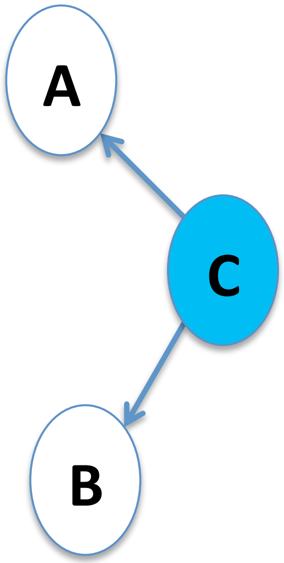
ie., $p(a,b|c)=p(a|c)p(b|c)$



Easy Network 2 : Head to Tail

Prove A cond indep of B given C ?

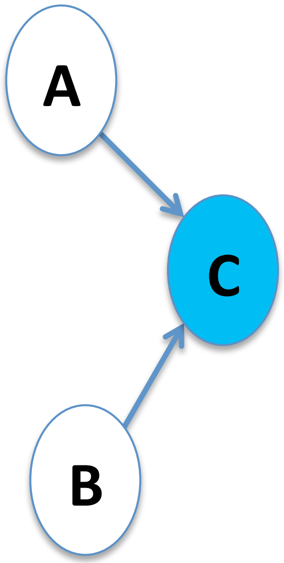
ie., $p(a,b|c)=p(a|c)p(b|c)$



Easy Network 3 : Head to Tail

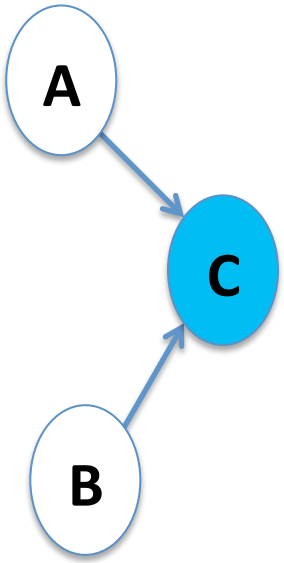
Prove A cond indep of B given C ?

ie., $p(a,b|c)=p(a|c)p(b|c)$



Easy Network 1 : Head to Head

Prove A cond indep of B given C ? No!



Summary :

- p(a,b) =p(a)p(b)
- P(a,b|c) NotEqual p(a|c)p(b|c)

Explaining away

e.g.,

- A=earthquake
- B=breakIn
- C=motionAlarm

X and Y are conditionally independent given Z, **<u>if and only if</u>** X and Y are D-separated by Z.

[Bishop, 8.2.2]

Suppose we have three sets of random variable : X, Y and Z

X and Y are **<u>D-separated</u>** by Z (and threrefore conditionally indep, given Z) iff every path from variable in X to any variable in Y is **<u>blocked</u>**.

A path from variable A to variable B is **blocked** if it includes a node such that either

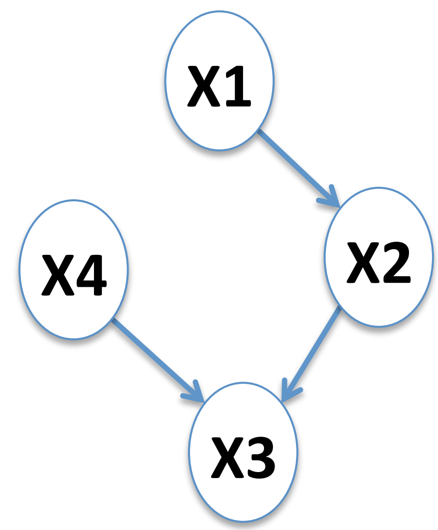
1. Arrows on the path meet either head-to-tail or tail-to-tail at the node and this node is in Z.
2. the arrows meet head-to-head at the node, and neither the node, nor any of its descendants, is in Z

X and Y are **D-separated** by Z (and threrefore conditionally indep, given Z) iff every path from variable in X to any variable in Y is **blocked**.

A path from variable A to variable B is **blocked** if it includes a node such that either

1. Arrows on the path meet either head-to-tail or tail-to-tail at the node and this node is in Z.
2. the arrows meet head-to-head at the node, and neither the node, nor any of its descendants, is in Z

X1 indeep of X3 given X2 ?
X3 indeep of X1 given X2 ?
X4 indeep of X1 given X2 ?



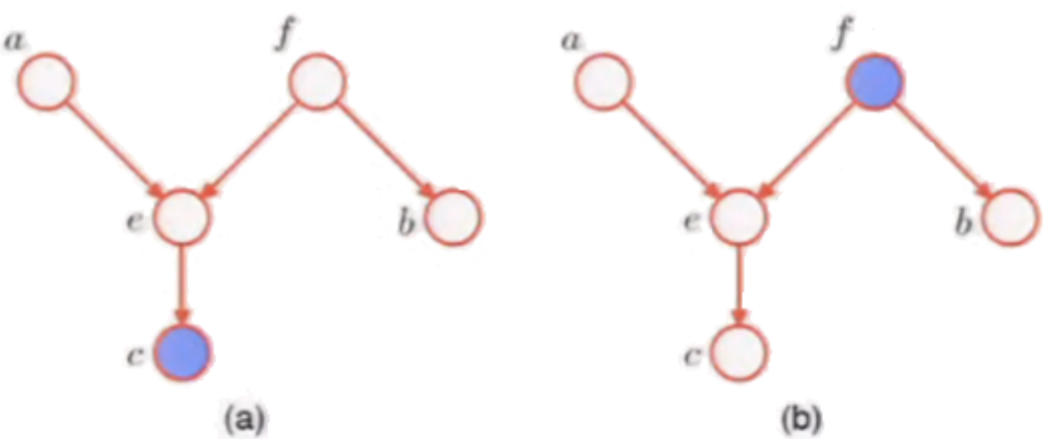
X and Y are **<u>D-separated</u>** by Z (and threrefore conditionally indep, given Z) iff every path from variable in X to any variable in Y is **<u>blocked</u>**.

A path from variable A to variable B is **blocked** if it includes a node such that either

1. Arrows on the path meet either head-to-tail or tail-to-tail at the node and this node is in Z.
2. the arrows meet head-to-head at the node, and neither the node, nor any of its descendants, is in Z

a indep of b given c ?

a indep of b given f ?

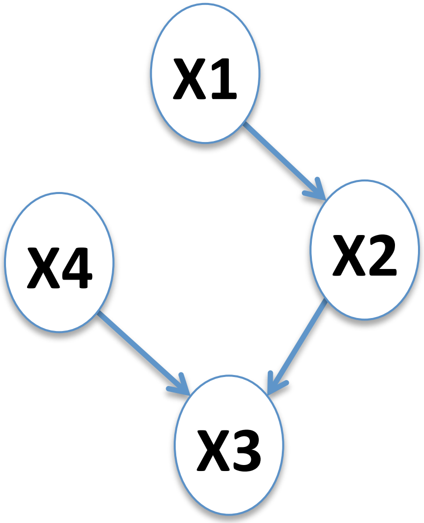


X and Y are **D-separated** by Z (and threrefore conditionally indep, given Z) iff every path from variable in X to any variable in Y is **blocked**.

A path from variable A to variable B is **blocked** if it includes a node such that either

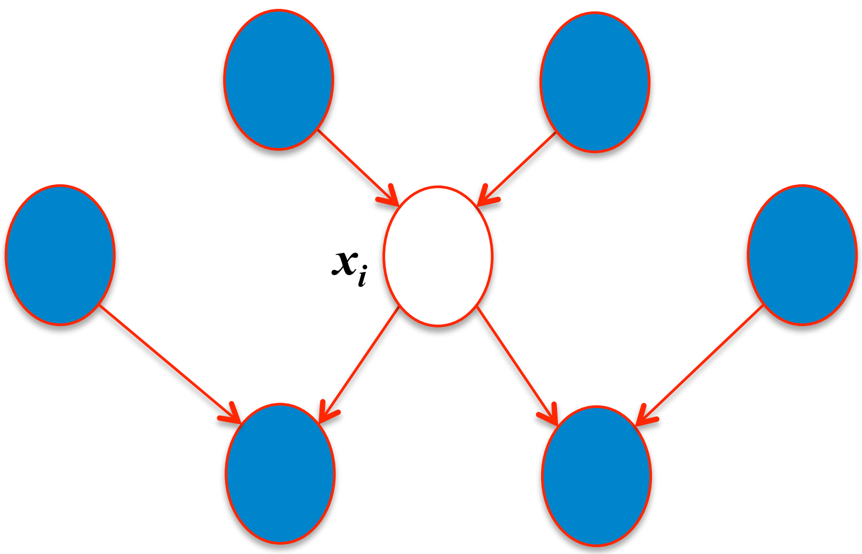
1. Arrows on the path meet either head-to-tail or tail-to-tail at the node and this node is in Z.
2. the arrows meet head-to-head at the node, and neither the node, nor any of its descendants, is in Z

X1 indeep of X3 given X2 ?
X3 indeep of X1 given X2 ?
X4 indeep of X1 given X2 ?



Markov Blanket

The Markov blanket of a node x_i comprises the set of parents, children and co-parents of the node. It has the property that the conditional distribution of 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]

What You Should Know

- **Bayes nets are convenient representation for encoding dependencies / conditional independence**
- **BN = Graph plus parameters of CPD's**
  - Definens joint distribution over variables
  - Can calculate everything else from that
  - Though inference may be intractable
- **Reading conditional independence relations from the graph**
  - Each node is cond indep of non-descendents, given its immediate parents
  - D-separation
  - 'Explaining away'

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 for graph structure is *known*, and data is *fully observed*

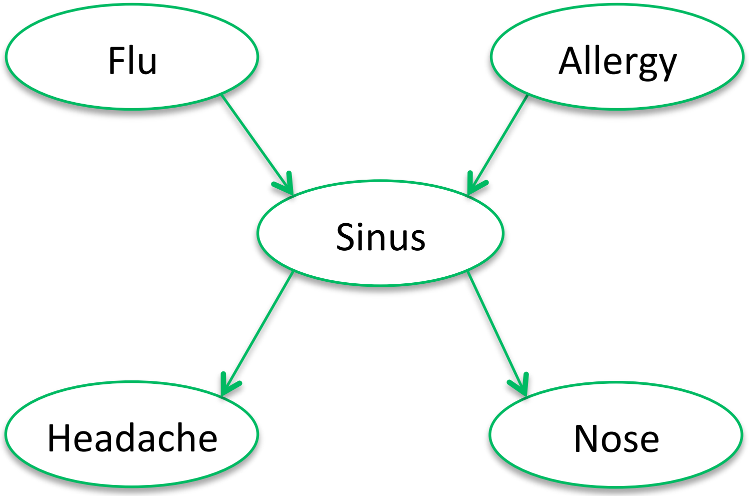
- Interesting case : graph *known*, data *partly known*

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

Learning CPTs from Fully Observed Data

- Example : Consider learning the parameter

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

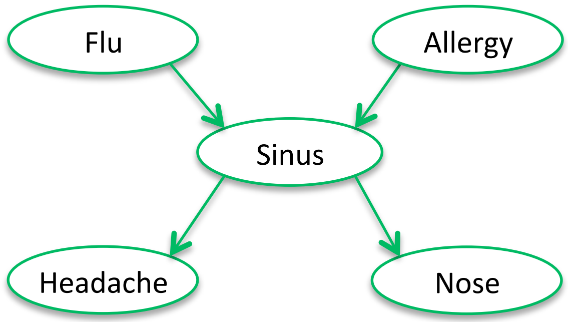


- MLE (Max Likelihood Estimate) is

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

k^th training example

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



- Maximum likelihood estimate

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

- Our case

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

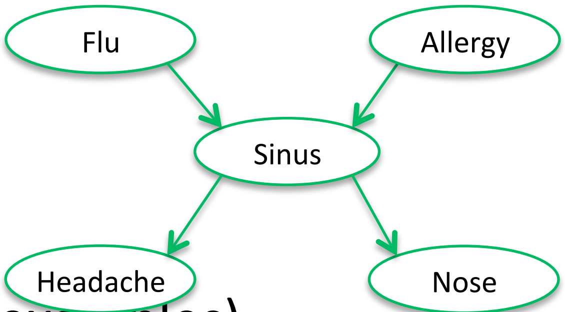
$$P(data \mid \theta) = \prod_{k=1}^{K} P(f_k) P(a_k) P(s_k \mid f_k a_k) P(h_k \mid s_k) P(n_k \mid s_k)$$

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

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

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

Estimate θ from partly oabserved data



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

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

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

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

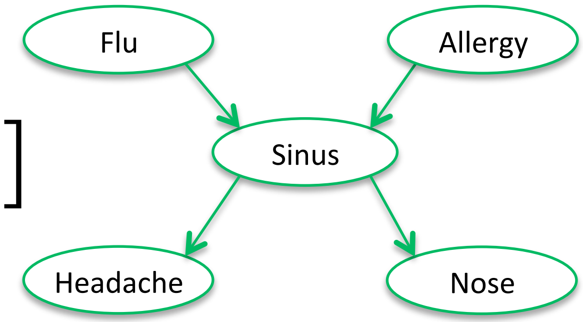
- EM seeks* to estimate :

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

*EM guaranteed to find local maximum

- EM seeks estimate :

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



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

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

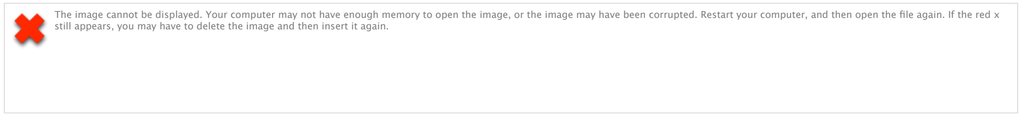
$$E_{P(Z|X,\theta)} \log P(X,Z \mid \theta) = \sum_{k=1}^{K} \sum_{i=0}^{1} P(s_k = i \mid f_k, a_k, h_k, n_k)$$

$$\left[\log P(f_k) + \log P(a_k) + \log P(s_k \mid f_k a_k) + \log P(h_k \mid s_k) + \log P(n_k \mid s_k)\right]$$

EM Algorithm

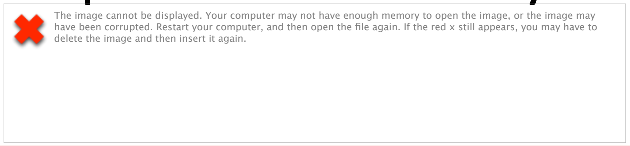
EM is a general procedure for learning from partly observed data

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

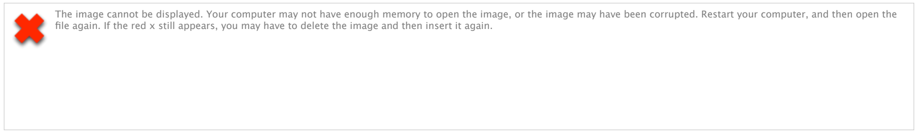
Define 

Iterate until convergence :

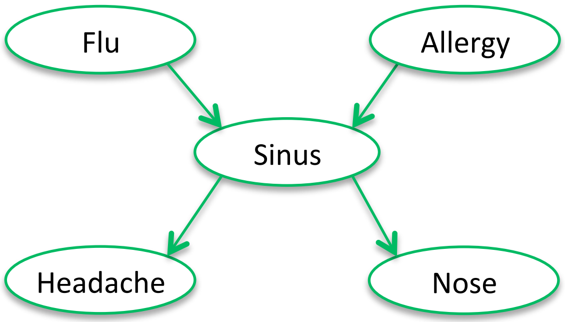
- E Step : Use X and current θ to calculate P(Z|X, θ)
- M Step : Replace current θ by



Guaranteed to find local maximum.

Each iteration increases 

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



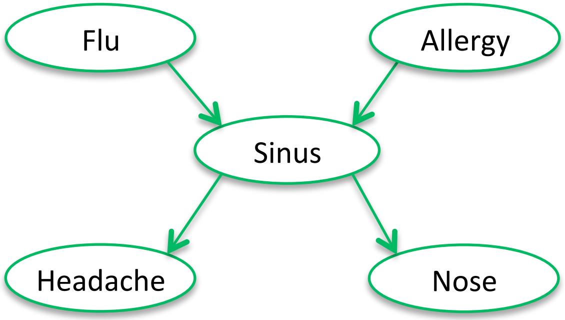
Observed X={F,A,H,N},
Unobserved Z={S}

- How ? Bayes net inference problem.

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

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

Observed X={F,A,H,N},
Unobserved Z={S}

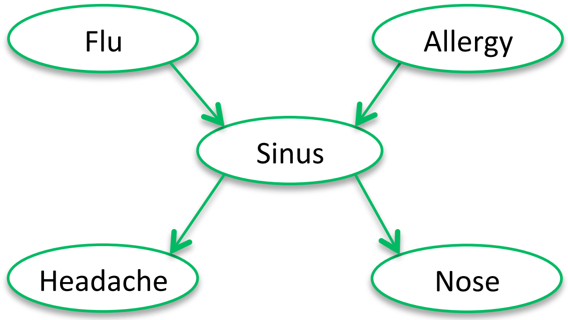


- How ? Bayes net inference problem.

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

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

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

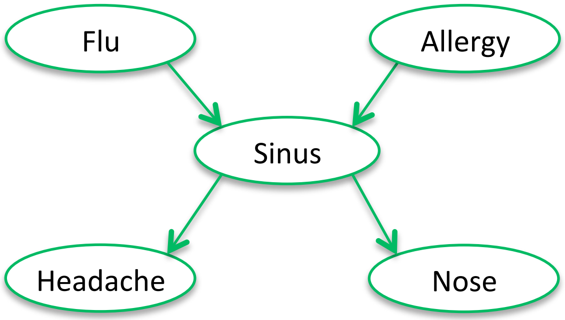
$$P(S_k=1|f_k a_k s_k h_k n_k,\theta) = E[S_k] = \frac{P(S_k=1, f_k a_k s_k h_k n_k|\theta)}{P(S_k=1, f_k a_k s_k h_k n_k|\theta) + P(S_k=0, f_k a_k s_k h_k n_k|\theta)}$$

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

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

M step :

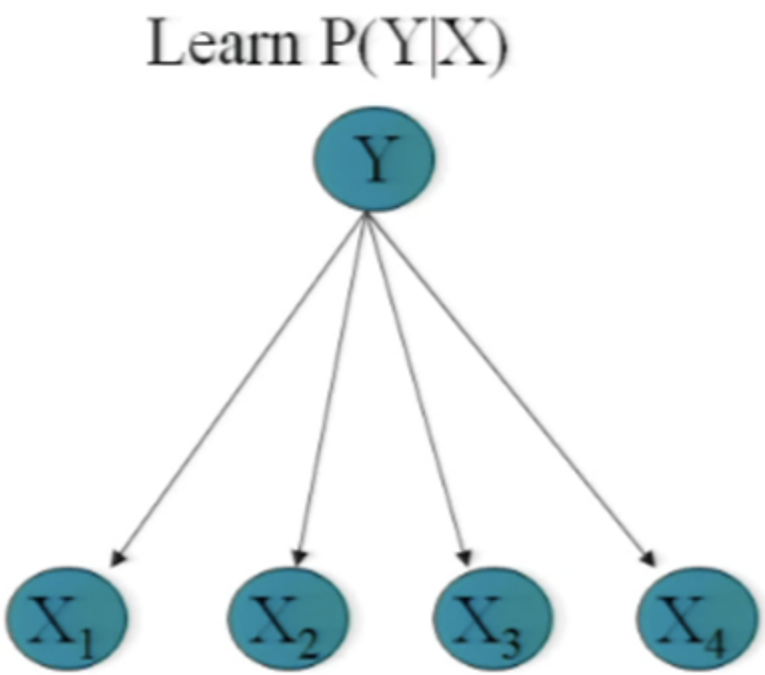
Calculate estimates similar to MLE, but

replacing each count by its expected count

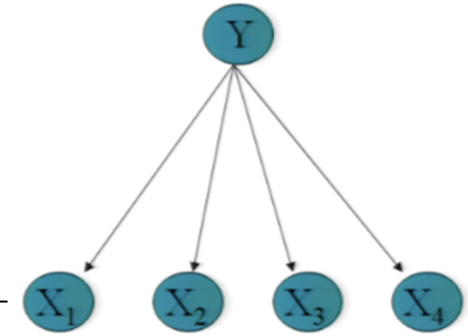
$$\delta(Y=1) \rightarrow E_{Z|X,\theta}[Y]$$

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

Using Unlabeled Data to Help Train Naïve Bayes Classifier



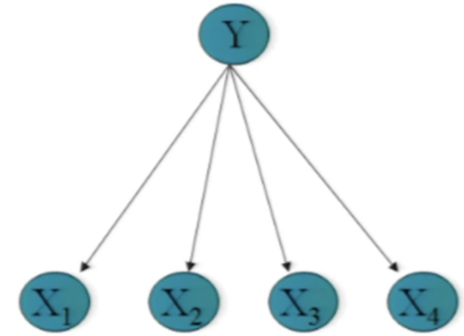
Y	X1	X2	X3	X4
1	0	0	1	1
0	0	1	0	0
0	0	0	1	0
?	0	1	1	0
?	0	1	0	1



E Step : Calculate for each training example, k

the expected value of each unobserved variable

EM and estimating θ



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

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

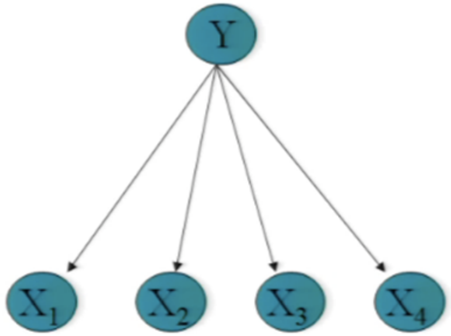
M step :

Calculate estimates similar to MLE, but

replacing each count by its <u>expected count</u>

let's use y(k) to indicate value of Y on k^{th} example

EM and estimating θ



Given observed set X, unobserved set Z of boolean values

E Step : Calculate for each training example, k

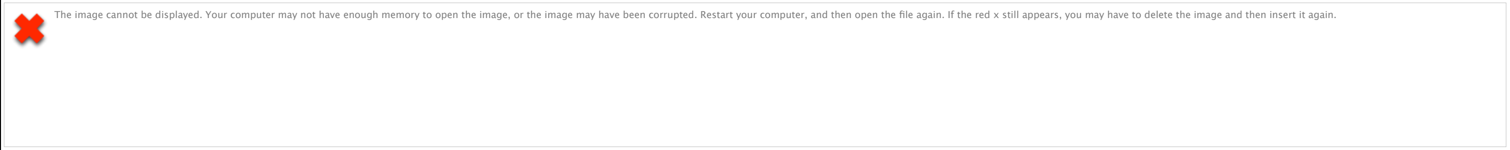
the expected value of each unobserved variable

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

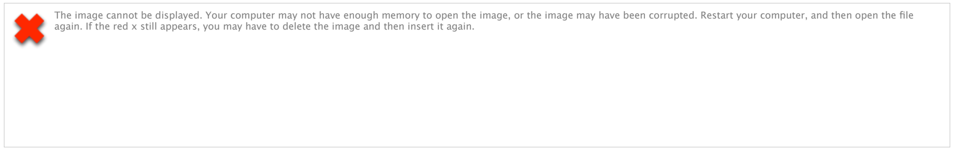
M step :

Calculate estimates similar to MLE, but

replacing each count by its expected count



MLE would be :



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

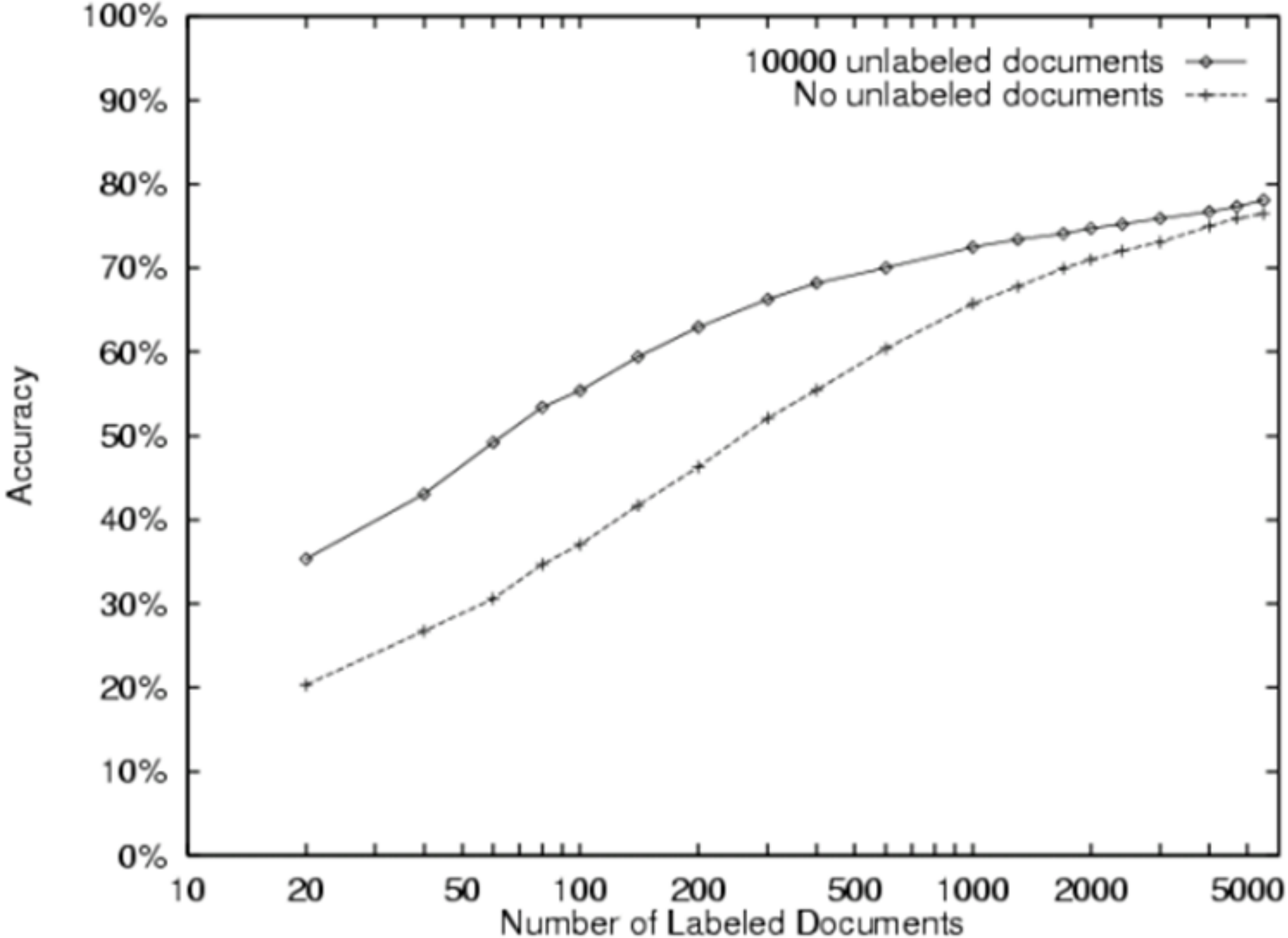
Table 3. Lists of the words most predictive of the course class in the WebKB data set, as they change over iterations of EM for a specific trial. By the second iteration of EM, many common course-related words appear. The symbol D indicates an arbitrary digit.

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

Using one labeled example per class

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

20 Newsgroups



20 Newsgroups