Machine Learning 10-601

Tom M. Mitchell
Machine Learning Department
Carnegie Mellon University

October 4, 2012

Today:

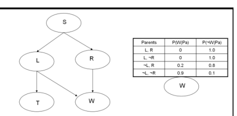
- Graphical models
- Bayes Nets:
  - Inference
  - Learning

Readings:

Required:

- Bishop chapter 8

Bayesian Networks Definition



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

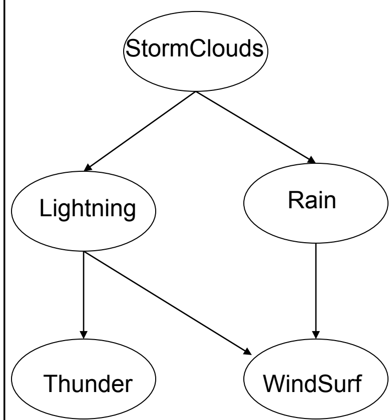
A Bayes network is a directed acyclic graph and a set of conditional probability distributions (CPD's)

- Each node denotes a random variable
- Edges denote dependencies
- For each node X_i its CPD defines $P(X_i \mid Pa(X_i))$
- 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

Bayesian Network

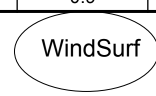


What can we say about conditional independencies in a Bayes Net?

One thing is this:

Each node is conditionally independent of its non-descendents, given only its immediate parents.

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



What You Should Know

- Bayes nets are convenient representation for encoding dependencies / conditional independence
- BN = Graph plus parameters of CPD's
  - Defines 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 only its parents

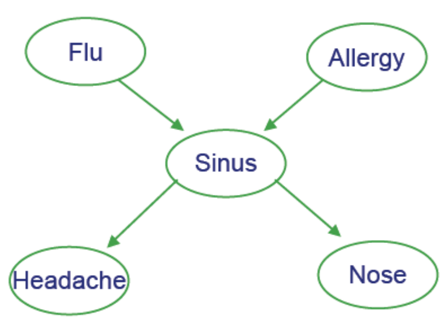
See Bayes Net applet: http://www.cs.cmu.edu/~javabayes/Home/applet.html

Inference in Bayes Nets

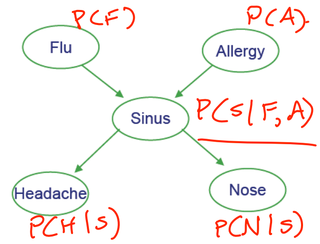
- In general, intractable (NP-complete)
- For certain cases, tractable
  - Assigning probability to fully observed set of variables
  - Or if just one variable unobserved
  - Or for singly connected graphs (ie., no undirected loops)
    - Belief propagation
- For multiply connected graphs
    - Junction 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

Example

- Bird flu and Allegies both cause Sinus problems
- Sinus problems cause Headaches and runny Nose



Prob. of joint assignment: easy

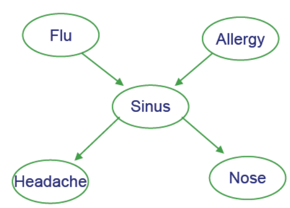


- Suppose we are interested in joint assignment <F=f,A=a,S=s,H=h,N=n>

What is P(f,a,s,h,n)? $= P(f)\,P(a)\,P(s|f,a)\,P(h|s)\,P(n|s)$

let's use p(a,b) as shorthand for p(A=a, B=b)

Prob. of marginals: not so easy



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

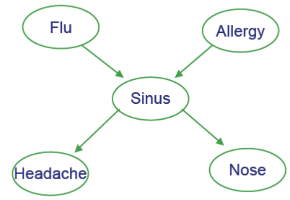
$$P(N=n) = \underbrace{\sum_{f,a,h,s}}_{F} P(F=f, A=a, H=h, S=s, N=n)$$

n vars $\rightarrow$ $2^{(n-1)}$

let's use p(a,b) as shorthand for p(A=a, B=b)

Generating a sample from joint distribution: easy

$P(N=1)$

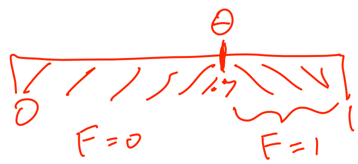


How can we generate random samples drawn according to P(F,A,S,H,N)?

F	Pr	
0	.7	← θ
1	.3	← $1-\theta$

CPD

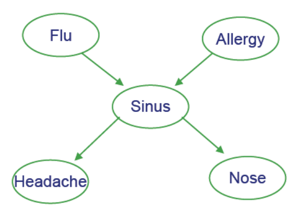
$= P(F)\, P(A)\, P(S|F,A)\, P(H|S)\, P(N|S)$



F A	$P(S=0\mid F,A)$
0 0	.9
0 1	.3
1 0	.5
1 1	.4

let's use p(a,b) as shorthand for p(A=a, B=b)

Generating a sample from joint distribution: easy



Note we can estimate marginals like 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) $\equiv \dfrac{P(F=1, H=1, N=0)}{P(H=1, N=0)}$

→ 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

eg., chain $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$

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
    - 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

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

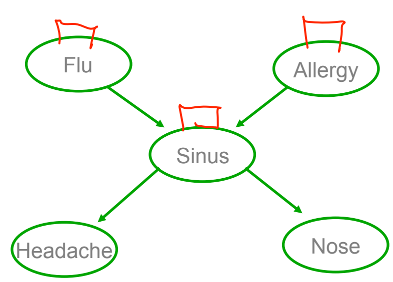
$$\theta_{s|ij} \equiv P(S=1|F=i, A=j)$$

- MLE (Max Likelihood Estimate) is

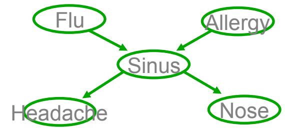
$$\theta_{s|ij} = \frac{\sum_{k=1}^{K} \delta(f_k = i, a_k = j, s_k = 1)}{\sum_{k=1}^{K} \delta(f_k = i, a_k = j)}$$

k^{th} training example

- Remember why?



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



- Maximum likelihood estimate
  $\theta \leftarrow \arg\max_{\theta} \log P(data|\theta)$
- Our case:

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

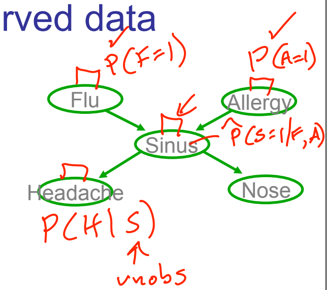
$$P(data|\theta) = \prod_{k=1}^{K} P(f_k)P(a_k)P(s_k|f_k a_k)P(h_k|s_k)P(n_k|s_k)$$

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

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

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

Estimate θ from partly observed data



obs unobs

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

Vector of params.

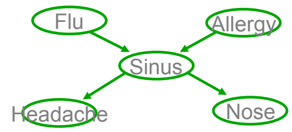
$$\theta_{MLE} \leftarrow \arg\max_{\theta} \log \prod_{k} P(f_k, a_k, s_k, h_k, n_k|\theta)$$

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

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

- 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:

$E[f(x)] = \sum_x P(x=x) f(x)$

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

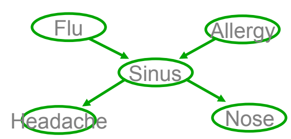
- EM seeks* to estimate:

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

$P(Z|X,\theta)$

* EM guaranteed to find local maximum

- 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}

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

unobs obs

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

$$[log P(f_k) + \log P(a_k) + \log P(s_k|f_k a_k) + \log P(h_k|s_k) + \log P(n_k|s_k)]$$

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

current

M step new

Iterate until convergence:

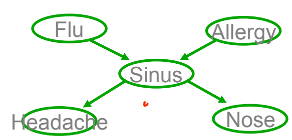
- E Step: Use X and current θ to calculate $P(Z|X,\theta)$
- M Step: Replace current θ by

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

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 $P(Z|X,\theta)$

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



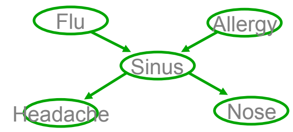
- How? Bayes net inference problem.

$$P(S_k = 1|f_k a_k h_k n_k, \theta) = \frac{P(S=1, f_k\ a\ h\ n\ \theta)}{P(f\ a\ h n; \theta)}$$

∴ work this out as midterm prep.

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

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



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

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

$$P(S_k=1|f_k a_k h_k n_k,\theta) = \underbrace{\boxed{E[s_k]}_{P(Z|X;\theta)}}_{P(S=s_k|X)} = \frac{P(S_k=1, f_k a_k h_k n_k|\theta)}{P(S_k=1, f_k a_k h_k n_k|\theta) + P(S_k=0, f_k a_k h_k n_k|\theta)}$$

M step: update all relevant parameters. For example:

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

1 if args is true

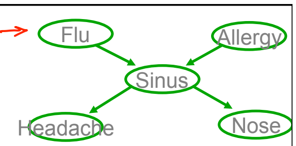
$E[s_k] = P(s_k=1)$

θ h|s

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

P(F=1) →



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

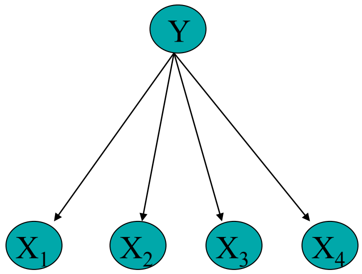
M step:

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

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

Using Unlabeled Data to Help Train Naïve Bayes Classifier

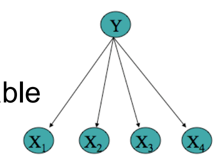
Learn P(Y|X)



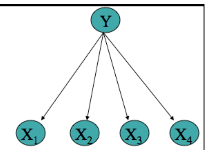
Y	X1	X2	X3	X4
1	0	0	1	1
0	0	1	0	0
0	0	0	1	0
?	0	1	1	0
?	0	1	0	1

E step: Calculate for each training example, k

the expected value of each unobserved variable



EM and estimating θ



Given observed set X, unobserved set Y of boolean values

E step: Calculate for each training example, k

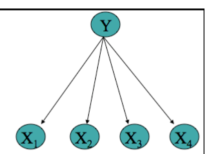
the expected value of each unobserved variable Y

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

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

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

- **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

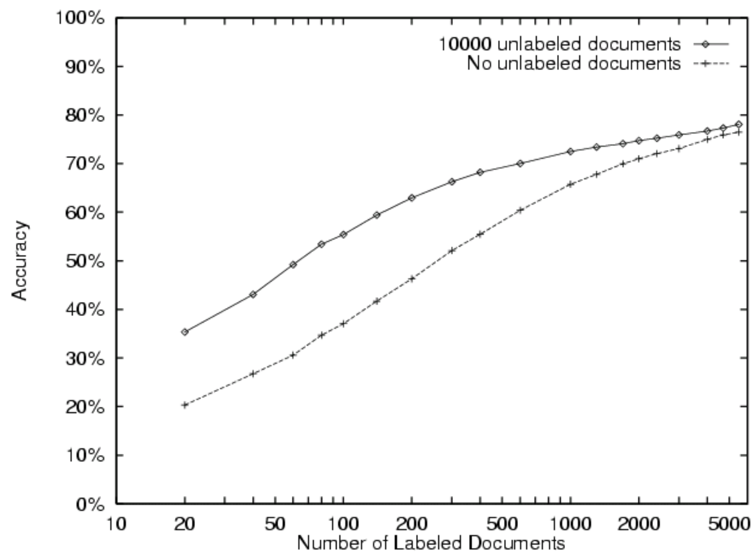


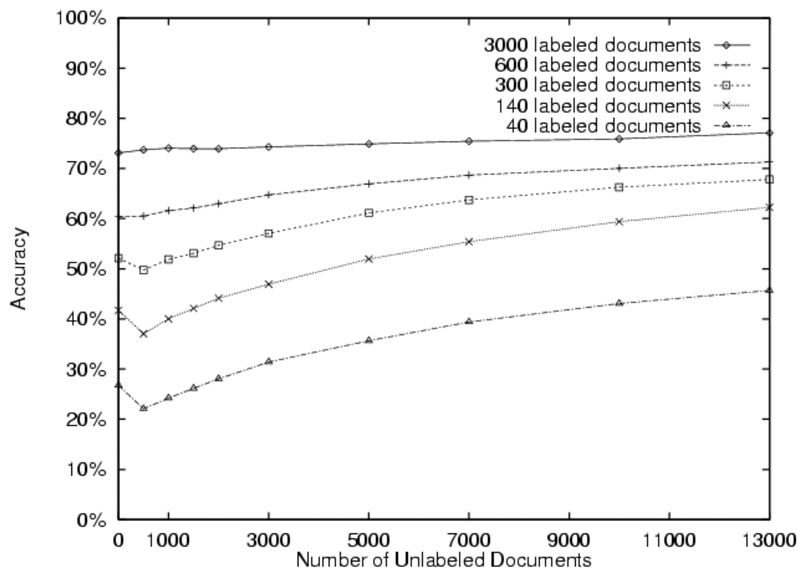
Table 3. Lists of the words most predictive of the course class in the WebKB data set, as they change over iterations of EM for a specific trial. By the second iteration of EM, many common course-related words appear. The symbol D indicates an arbitrary digit.

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^*	$DD{:}DD$
dist	$DD{:}DD$	due
identical	handout	D^*
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



Usupervised clustering

Just extreme case for EM with zero labeled examples…

Clustering

- Given set of data points, group them
- Unsupervised learning
- Which patients are similar? (or which earthquakes, customers, faces, web pages, …)

Mixture Distributions

Model joint $P(X_1 \ldots X_n)$ as mixture of multiple distributions.

Use discrete-valued random var Z to indicate which distribution is being use for each random draw

So $$P(X_1 \ldots X_n) = \sum_i P(Z=i) \;\; P(X_1 \ldots X_n|Z)$$

Mixture of *Gaussians*:

- Assume each data point X=<X1, … Xn> is generated by one of several Gaussians, as follows:

1. randomly choose Gaussian i, according to P(Z=i)
2. randomly generate a data point <x1,x2 .. xn> according to $N(\mu_i, \Sigma_i)$

EM for Mixture of Gaussian Clustering

Let's simplify to make this easier:

1. assume $X=<X_1 \ldots X_n>$, and the X_i are conditionally independent given Z.

$$P(X|Z=j) = \prod_i N(X_i|\mu_{ji}, \sigma_{ji})$$

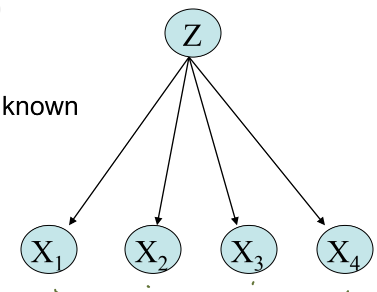
2. assume only 2 clusters (values of Z), and $\forall i, j, \sigma_{ji} = \sigma$

$$P(\mathbf{X}) = \sum_{j=1}^{2} P(Z=j|\pi) \prod_i N(x_i|\mu_{ji}, \sigma)$$

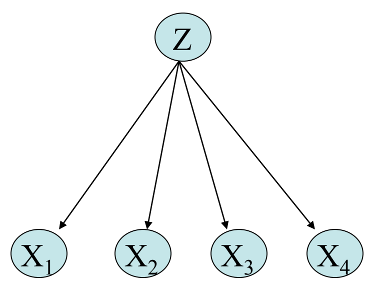
3. Assume σ known, $\pi_1 \ldots \pi_K, \mu_{1i} \ldots \mu_{Ki}$ unknown

Observed: $X=<X_1 \ldots X_n>$

Unobserved: Z



EM



Given observed variables X, unobserved Z

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

where $\theta = \langle \pi, \mu_{ji} \rangle$

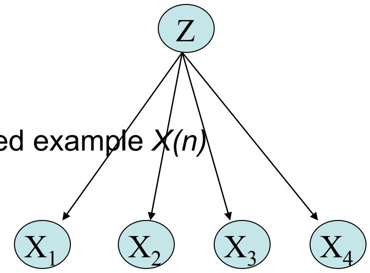
Iterate until convergence:

• E Step: Calculate $P(Z(n)|X(n),\theta)$ for each example $X(n)$. Use this to construct $Q(\theta'|\theta)$

• M Step: Replace current θ by

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

EM – E Step



Calculate *P(Z(n)|X(n),θ)* for each observed example *X(n)*

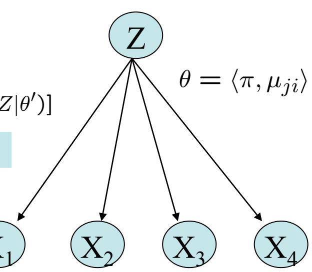
X(n)=<x₁(n), x₂(n), … x_T(n)>.

$$P(z(n)=k|x(n),\theta)=\frac{P(x(n)|z(n)=k,\theta)\ \ P(z(n)=k|\theta)}{\sum_{j=0}^{1} p(x(n)|z(n)=j,\theta)\ P(z(n)=j|\theta)}$$

$$P(z(n)=k|x(n),\theta)=\frac{[\prod_i P(x_i(n)|z(n)=k,\theta)]\ \ P(z(n)=k|\theta)}{\sum_{j=0}^{1}\prod_i P(x_i(n)|z(n)=j,\theta)\ P(z(n)=j|\theta)}$$

$$\boxed{P(z(n)=k|x(n),\theta)=\frac{[\prod_i N(x_i(n)|\mu_{k,i},\sigma)]\ \ (\pi^k(1-\pi)^{(1-k)})}{\sum_{j=0}^{1}[\prod_i N(x_i(n)|\mu_{j,i},\sigma)]\ \ (\pi^j(1-\pi)^{(1-j)}))}}$$

EM – M Step



$$\theta=\langle\pi,\mu_{ji}\rangle$$

First consider update for π

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

π' has no influence

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

Count z(n)=1

$$E_{Z|X,\theta}\left[\log P(Z|\pi')\right]=E_{Z|X,\theta}\left[\log\left(\pi'^{\sum_n z(n)}(1-\pi')^{\sum_n(1-z(n))}\right)\right]$$

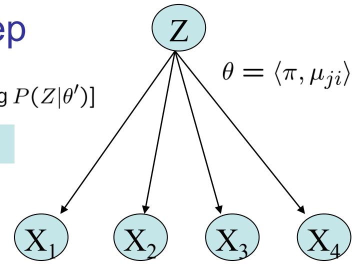
$$=E_{Z|X,\theta}\left[\left(\sum_n z(n)\right)\log\pi'+\left(\sum_n(1-z(n))\right)\log(1-\pi')\right]$$

$$=\left(\sum_n E_{Z|X,\theta}[z(n)]\right)\log\pi'+\left(\sum_n E_{Z|X,\theta}[(1-z(n))]\right)\log(1-\pi')$$

$$\frac{\partial E_{Z|X,\theta}[\log P(Z|\pi')]}{\partial\pi'}=\left(\sum_n E_{Z|X,\theta}[z(n)]\right)\frac{1}{\pi'}+\left(\sum_n E_{Z|X,\theta}[(1-z(n))]\right)\frac{(-1)}{1-\pi'}$$

$$\boxed{\pi\leftarrow\frac{\sum_{n=1}^{N}E[z(n)]}{\left(\sum_{n=1}^{N}E[z(n)]\right)+\left(\sum_{n=1}^{N}(1-E[z(n)])\right)}=\frac{1}{N}\sum_{n=1}^{N}E[z(n)]}$$

EM – M Step



Now consider update for μ_{ji}

$\theta = \langle \pi, \mu_{ji} \rangle$

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

μ_{ji}' has no influence

$$\mu_{ji} \leftarrow \arg\max_{\mu'_{ji}} E_{Z|X,\theta}[\log P(X|Z, \theta')]$$

…

…

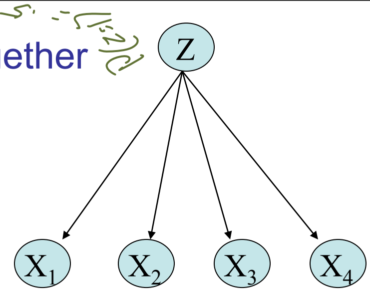
…

$$\boxed{\mu_{ji} \leftarrow \frac{\sum_{n=1}^{N} P(z(n) = j|x(n), \theta) \quad x_i(n)}{\sum_{n=1}^{N} P(z(n) = j|x(n), \theta)}}$$

Compare above to MLE if Z were observable:

$$\mu_{ji} \leftarrow \frac{\sum_{n=1}^{N} \delta(z(n) = j) \quad x_i(n)}{\sum_{n=1}^{N} \delta(z(n) = j)}$$

EM – putting it together



Given observed variables X, unobserved Z

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

where $\theta = \langle \pi, \mu_{ji} \rangle$

Iterate until convergence:

- E Step: For each observed example X(n), calculate $P(Z(n)|X(n), \theta)$

$$P(z(n) = k \mid x(n), \theta) = \frac{[\prod_i N(x_i(n)|\mu_{k,i}, \sigma)] \quad (\pi^k (1-\pi)^{(1-k)})}{\sum_{j=0}^{1} [\prod_i N(x_i(n)|\mu_{j,i}, \sigma)] \quad (\pi^j (1-\pi)^{(1-j)})}$$

- M Step: Update $\theta \leftarrow \arg\max_{\theta'} Q(\theta'|\theta)$

P(z=1)

$$\pi \leftarrow \frac{1}{N} \sum_{n=1}^{N} E[z(n)] \qquad \mu_{ji} \leftarrow \frac{\sum_{n=1}^{N} P(z(n) = j|x(n), \theta) \quad x_i(n)}{\sum_{n=1}^{N} P(z(n) = j|x(n), \theta)}$$

Mixture of Gaussians applet

Go to: http://www.socr.ucla.edu/htmls/SOCR_Charts.html
then go to Go to "Line Charts" → SOCR EM Mixture Chart

- try it with 2 Gaussian mixture components ("kernels")
- try it with 4

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 $P(Z^k | X^k, \theta)$
  - 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 methods to constrain search

One key result:

- Chow-Liu algorithm: finds "best" tree-structured network
- What's best?
  - suppose P(**X**) is true distribution, T(**X**) is our tree-structured network, where **X** = <X_1, … X_n>
  - Chow-Liu minimizes Kullback-Leibler divergence:

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

Chow-Liu Algorithm

<u>Key result</u>: To minimize KL(P || T), 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 vars A,B, use data to estimate P(A,B), P(A), P(B)

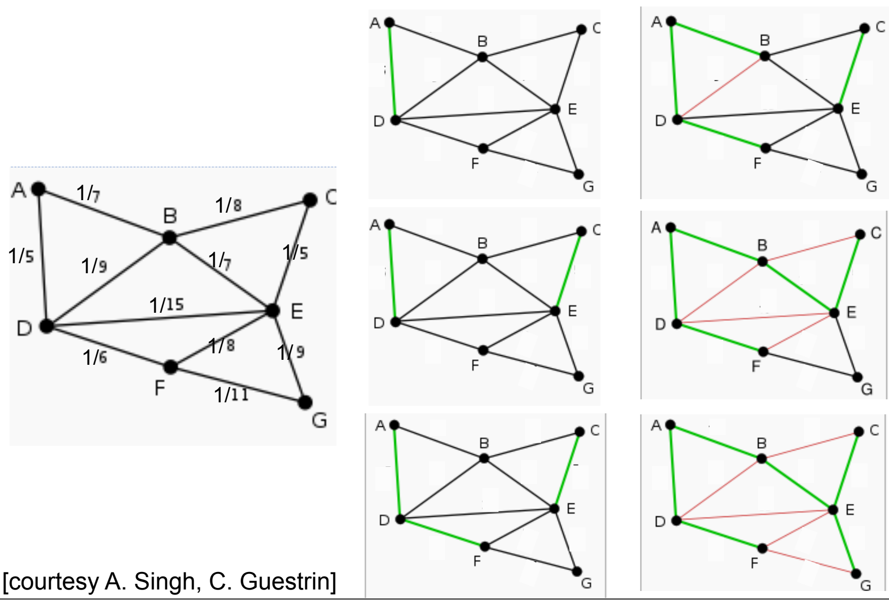
2. for each pair of vars A,B calculate mutual information
$$I(A,B) = \sum_a \sum_b P(a,b) \log \frac{P(a,b)}{P(a)P(b)}$$

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]

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