

BAYESIAN LEARNING

[Read Ch. 6]

[Suggested exercises: 6.1, 6.2, 6.6]

- Bayes Theorem
- MAP, ML hypotheses, MAP learners
- Minimum description length principle
- Bayes optimal classifier, Naive Bayes learner
- Example: Learning over text data
- Bayesian belief networks
- Expectation Maximization algorithm

Two Roles for Bayesian Methods

Provides practical learning algorithms:

- Naive Bayes learning
- Bayesian belief network learning
- Combine prior knowledge (prior probabilities) with observed data
- Requires prior probabilities

Provides useful conceptual framework

- Provides “gold standard” for evaluating other learning algorithms
- Additional insight into Occam’s razor

Bayes Theorem

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

- $P(h)$ = *prior probability* of hypothesis h

Initial probability that hypothesis h holds, before we have observed training data D

- $P(D)$ = prior probability of training data D

Probability of D given no knowledge about which hypothesis holds

- $P(D|h)$ = probability of D given h

Posterior probability of D . Probability of observing D given some world in which h holds. *Likelihood* of D given h

- $P(h|D)$ = probability of h given D

Posterior probability of h . Probability that h holds given the observed D . Confidence that h holds after we have seen D . Reflects influence of D (unlike $P(h)$)

Choosing Among Hypotheses

- Generally we want the most probable or likely hypothesis given the training data

Maximum a posteriori (MAP) hypothesis h_{MAP} :

$$\begin{aligned}h_{MAP} &= \arg \max_{h \in H} P(h|D) \\ &= \arg \max_{h \in H} \frac{P(D|h)P(h)}{P(D)} \\ &= \arg \max_{h \in H} P(D|h)P(h)\end{aligned}$$

- If we assume $P(h_i) = P(h_j)$ then we can further simplify, and choose the

Maximum likelihood (ML) hypothesis

$$h_{ML} = \arg \max_{h_i \in H} P(D|h_i)$$

\Rightarrow simple learning rule

- Example: Want to decide if patient has cancer. Observe the priors, observe a new patient with positive test, and diagnose.

Bayes Theorem

- Does patient have cancer or not?

A patient takes a lab test and the result comes back positive. The test returns a correct positive result in only 98% of the cases in which the disease is actually present, and a correct negative result in only 97% of the cases in which the disease is not present. Furthermore, .008 of the entire population have this cancer.

- Summary of probabilities

$$\begin{aligned}P(\text{cancer}) &= .008 & P(\neg\text{cancer}) &= .992 \\P(\oplus|\text{cancer}) &= .98 & P(\ominus|\text{cancer}) &= .02 \\P(\oplus|\neg\text{cancer}) &= .03 & P(\ominus|\neg\text{cancer}) &= .97\end{aligned}$$

- Maximum a posteriori hypothesis?

$$\begin{aligned}P(\oplus|\text{cancer})P(\text{cancer}) &= (.98).008 = .0078 \\P(\oplus|\neg\text{cancer})P(\neg\text{cancer}) &= (.03).992 = .0298\end{aligned}$$

$$\Rightarrow h_{MAP} = \neg\text{cancer}$$

- Since $P(\text{cancer}|\oplus) + P(\neg\text{cancer}|\oplus) = 1$

$$\text{and } P(\text{cancer}|\oplus) = \frac{.0078}{.0078+.0298} = .21$$

then we can compute $P(\oplus)$

Basic Formulas for Probabilities

- *Product Rule*: probability $P(A \wedge B)$ of a conjunction of two events A and B:

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

- *Sum Rule*: probability of a disjunction of two events A and B:

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

- *Theorem of total probability*: if events A_1, \dots, A_n are mutually exclusive with $\sum_{i=1}^n P(A_i) = 1$, then

$$P(B) = \sum_{i=1}^n P(B|A_i)P(A_i)$$

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

$$h_{MAP} = \arg \max_{h \in H} P(h|D)$$

Relation to Concept Learning

- Consider our usual concept learning task
 1. Instances X , space H , examples D
 2. Consider the FindS learning algorithm

What would Bayes rule produce as the MAP hypothesis? Does *FindS* output a MAP hypothesis??

- Assume fixed set of instances $\langle x_1, \dots, x_m \rangle$
Assume D is the set of classifications $D = \langle c(x_1), \dots, c(x_m) \rangle$

1. Choose $P(D|h)$

$$P(D|h) = 1 \text{ if } h \text{ consistent with } D$$

$$P(D|h) = 0 \text{ otherwise}$$

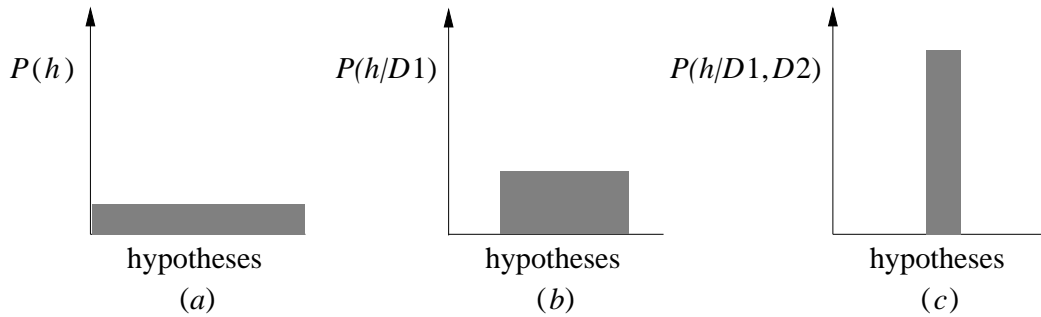
2. Choose $P(h)$ to be *uniform* distribution

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

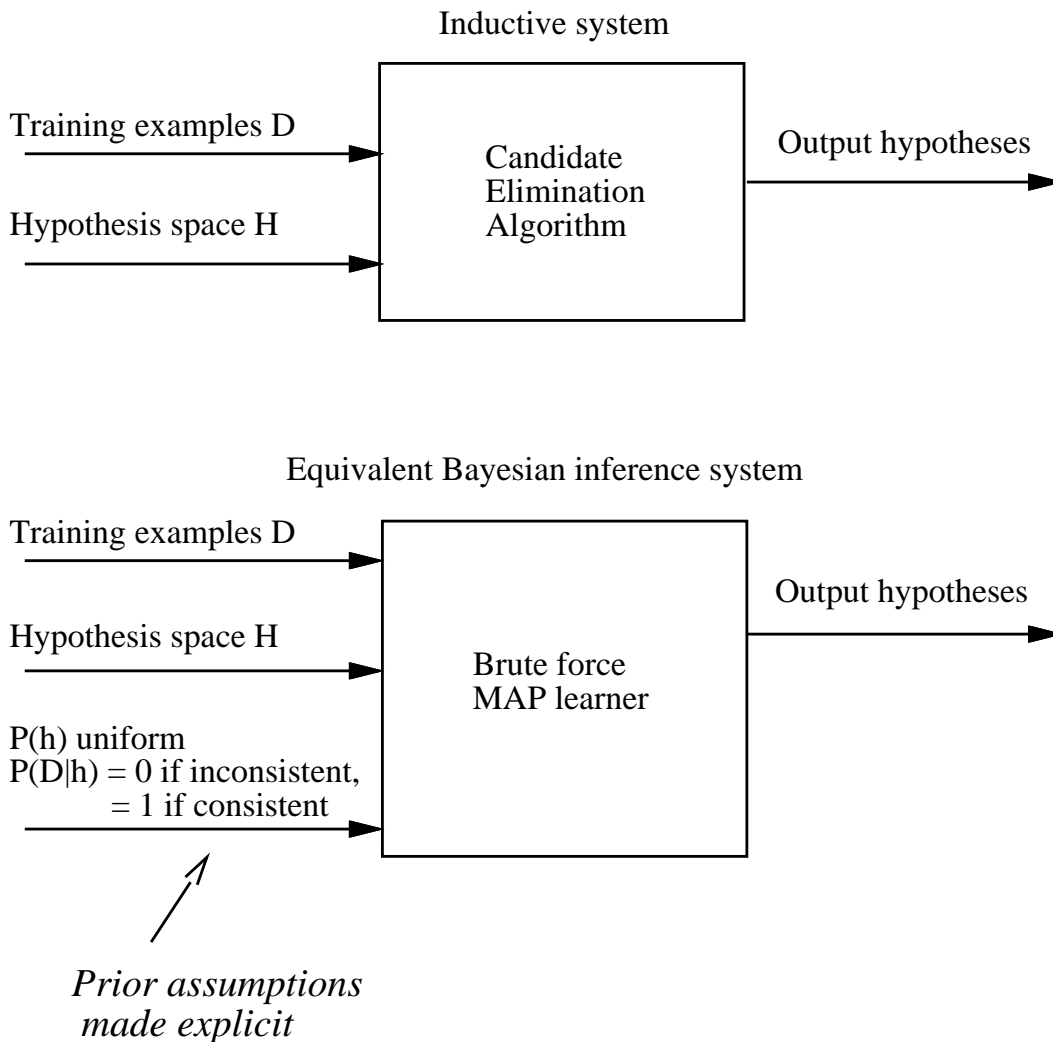
Then,

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

Evolution of Posterior Probabilities



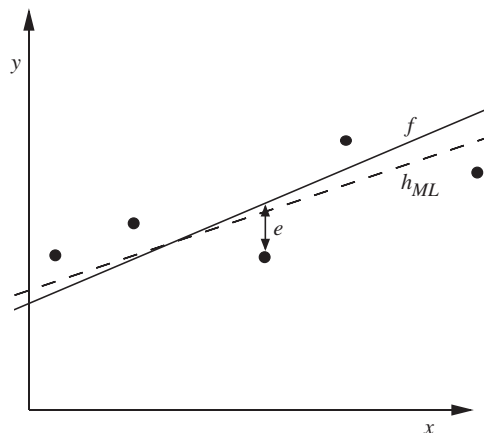
Characterizing Learning Algorithms by Equivalent MAP Learners



Learning A Real Valued Function

- Consider any real-valued target function f
- Consider noisy training examples $\langle x_i, d_i \rangle$, where
 1. $d_i = f(x_i) + e_i$
 2. e_i is random variable (noise) drawn independently for each x_i according to some Gaussian distribution with mean=0
- Then the maximum likelihood hypothesis h_{ML} is the one that minimizes the sum of squared errors:

$$h_{ML} = \arg \min_{h \in H} \sum_{i=1}^m (d_i - h(x_i))^2$$



Learning A Real Valued Function

Assuming Gaussian distribution of noise, then

$$\begin{aligned}h_{ML} &= \arg \max_{h \in H} p(D|h) \\ &= \arg \max_{h \in H} \prod_{i=1}^m p(d_i|h) \\ &= \arg \max_{h \in H} \prod_{i=1}^m \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{d_i - h(x_i)}{\sigma}\right)^2}\end{aligned}$$

Maximize natural log of this instead...

$$\begin{aligned}h_{ML} &= \arg \max_{h \in H} \sum_{i=1}^m \ln \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2} \left(\frac{d_i - h(x_i)}{\sigma} \right)^2 \\ &= \arg \max_{h \in H} \sum_{i=1}^m -\frac{1}{2} \left(\frac{d_i - h(x_i)}{\sigma} \right)^2 \\ &= \arg \max_{h \in H} \sum_{i=1}^m -(d_i - h(x_i))^2 \\ &= \arg \min_{h \in H} \sum_{i=1}^m (d_i - h(x_i))^2\end{aligned}$$

Learning to Predict Probabilities

- Consider predicting survival probability from given patient data
- With training examples $\langle x_i, d_i \rangle$, where d_i is 1 or 0
- We want to train neural network to output a *probability* given x_i (not a 0 or 1)
- In this case can show that

$$h_{ML} = \arg \max_{h \in H} \sum_{i=1}^m d_i \ln h(x_i) + (1 - d_i) \ln(1 - h(x_i))$$

- Weight update rule for a sigmoid unit is then:

$$w_{jk} \leftarrow w_{jk} + \Delta w_{jk}$$

where

$$\Delta w_{jk} = \eta \sum_{i=1}^m (d_i - h(x_i)) x_{ijk}$$

Minimum Description Length Principle

$$\begin{aligned}h_{MAP} &= \arg \max_{h \in H} P(D|h)P(h) \\ &= \arg \max_{h \in H} \log_2 P(D|h) + \log_2 P(h) \\ &= \arg \min_{h \in H} -\log_2 P(D|h) - \log_2 P(h) \quad (1)\end{aligned}$$

- Interesting fact from information theory:

The optimal (shortest expected coding length) code for an event with probability p is

$$-\log_2 p \text{ bits}$$

- So interpret (1):

1. $-\log_2 P(h)$ is length of h under optimal code
2. $-\log_2 P(D|h)$ is length of D given h under optimal code

⇒ Prefer the hypothesis that minimizes

$$\text{length}(h) + \text{length}(\text{misclassifications})$$

Minimum Description Length Principle

- Occam's razor: Prefer the shortest hypothesis
- MDL: Prefer the hypothesis h that minimizes

$$h_{MDL} = \arg \min_{h \in H} L_{C_1}(h) + L_{C_2}(D|h)$$

Where $L_C(x)$ is the description length of x under optimal encoding/representation C

- Example: If

H = decision trees, and

D = training data labels

1. $L_{C_1}(h)$ is # bits to describe tree h
2. $L_{C_2}(D|h)$ is # bits to describe D given h

Note $L_{C_2}(D|h) = 0$ if examples classified perfectly by h . Need only describe exceptions

⇒ Hence h_{MDL} trades off tree size for training errors

⇒ Bayesian learning implies Occam's Razor

Most Probable Classification of New Instances

- h_{MAP} = Most probable *hypothesis* given the data D
- Given new instance x , what is its most probable *classification*?

$h_{MAP}(x)$ is not the most probable classification!

- For instance, consider:

1. Three possible hypotheses h_1, h_2, h_3 such that:

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

$$\Rightarrow h_{MAP} = h_1)$$

2. Given a new instance x ,

$$h_1(x) = \oplus, \quad h_2(x) = \ominus, \quad h_3(x) = \ominus$$

3. What is the most probable classification of x ?

$$P(x = \oplus|D) = .4 \text{ by } h_1$$

$$P(x = \ominus|D) = .6 \text{ by } h_2 \text{ and } h_3$$

\Rightarrow Most probable classification of x is not $h_1(x)$

Bayes Optimal Classifier

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

Example:

$$P(h_1 | D) = .4, \quad P(\ominus | h_1) = 0, \quad P(\oplus | h_1) = 1$$

$$P(h_2 | D) = .3, \quad P(\ominus | h_2) = 1, \quad P(\oplus | h_2) = 0$$

$$P(h_3 | D) = .3, \quad P(\ominus | h_3) = 1, \quad P(\oplus | h_3) = 0$$

Therefore

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

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

And

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

- BOC maximizes the probability that a new instance is classified correctly, given D , H and prior probabilities over H

Gibbs Classifier

- BOC gives best result but can be quite costly

It computes the posteriors for every $h \in H$ and combine their predictions

- **Gibbs Algorithm:**

1. Randomly choose an $h \in H$, according to $P(h|D)$
2. Use h to classify a new instance

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

$$E[\text{error}_{Gibbs}] \leq 2E[\text{error}_{BayesOptimalClassifier}]$$

- Suppose correct, uniform prior distribution over H , then

Pick any hypothesis from VS , with uniform probability

Its expected error no worse than twice Bayes optimal

Naive Bayes Classifier

- When to use NBC
 1. Moderate or large training set available
 2. Attributes that describe instances are conditionally independent, given classification
- Assume target function $f : X \rightarrow V$, where each instance x is described by attributes $\langle a_1, a_2 \dots a_n \rangle$. Most probable value of $f(x)$ is:

$$v_{MAP} = \arg \max_{v_j \in V} P(v_j | a_1, a_2 \dots a_n)$$

$$\begin{aligned} v_{MAP} &= \arg \max_{v_j \in V} \frac{P(a_1, a_2 \dots a_n | v_j) P(v_j)}{P(a_1, a_2 \dots a_n)} \\ &= \arg \max_{v_j \in V} P(a_1, a_2 \dots a_n | v_j) P(v_j) \end{aligned}$$

- Naive Bayes assumption (of independence of attribute values):

$$P(a_1, a_2 \dots a_n | v_j) = \prod_i P(a_i | v_j)$$

Which gives the **Naive Bayes Classifier**

$$v_{NB} = \arg \max_{v_j \in V} P(v_j) \prod_i P(a_i | v_j)$$

Naive Bayes Learning Algorithm

Naive_Bayes_Learn(*Examples*)

- For each target value v_j
 1. $\hat{P}(v_j) \leftarrow$ estimate $P(v_j)$
 2. For each attribute value a_i of each attribute a
 $\hat{P}(a_i|v_j) \leftarrow$ estimate $P(a_i|v_j)$

Classify_New_Instance(x)

$$v_{NB} = \arg \max_{v_j \in V} \hat{P}(v_j) \prod_{a_i \in x} \hat{P}(a_i|v_j)$$

- In NBC, the conditional probabilities are *estimated* as normalized frequencies: how many times a given attribute value is associated with a given class

No search is needed (or done) at all.

Example: NBC for *PlayTennis*

Consider *PlayTennis* again, and a new instance

$\langle \text{Outlk} = \text{sun}, \text{Temp} = \text{cool}, \text{Humid} = \text{high}, \text{Wind} = \text{strong} \rangle$

We want to compute:

$$v_{NB} = \arg \max_{v_j \in V} P(v_j) \prod_i P(a_i | v_j)$$

We have:

- $P(\text{Yes}) = \frac{9}{14}$ and $P(\text{No}) = \frac{5}{14}$
- $P(\text{Wind} = \text{Strong} | \text{Yes}) = \frac{3}{9}$ and $P(\text{Wind} = \text{Strong} | \text{No}) = \frac{3}{5}$

Therefore, using NBC:

$$P(\text{Yes}) P(\text{sun} | \text{Yes}) P(\text{cool} | \text{Yes}) P(\text{high} | \text{Yes}) P(\text{strong} | \text{Yes}) = .0053$$

$$P(\text{No}) P(\text{sun} | \text{No}) P(\text{cool} | \text{No}) P(\text{high} | \text{No}) P(\text{strong} | \text{No}) = .0206$$

$$\rightarrow v_{NB} = \text{No}$$

We not only have a decision, but also the probability of that decision $\frac{.0206}{.0206 + .0053}$

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Naive Bayes: Subtleties

1. Conditional independence assumption is often violated . . .

$$P(a_1, a_2 \dots a_n | v_j) = \prod_i P(a_i | v_j)$$

- . . . But it works surprisingly well anyway. Note: don't need estimated posteriors $\hat{P}(v_j | x)$ to be correct; need only that

$$\arg \max_{v_j \in V} \hat{P}(v_j) \prod_i \hat{P}(a_i | v_j) = \arg \max_{v_j \in V} P(v_j) P(a_1 \dots, a_n | v_j)$$

- See [Domingos & Pazzani, 1996] for analysis
- Naive Bayes posteriors often unrealistically close to 1 or 0

2. What if none of the training instances with target value v_j have attribute value a_i ? Then

$$\hat{P}(a_i | v_j) = 0, \text{ and } \dots$$

$$\hat{P}(v_j) \prod_i \hat{P}(a_i | v_j) = 0$$

Typical solution is Bayesian estimate for $\hat{P}(a_i | v_j)$:

$$\hat{P}(a_i | v_j) \leftarrow \frac{n_c + mp}{n + m}$$

Where

- n is number of training examples for which $v = v_j$,
- n_c number of examples for which $v = v_j$ and $a = a_i$
- p is prior estimate for $\hat{P}(a_i | v_j)$
- m , called the *equivalent sample size* is weight given to prior (i.e. number of "virtual" examples)

Learning to Classify Text

- Why?
 1. Learn which news articles are of interest
 2. Learn to classify web pages by topic
- Naive Bayes is among most effective algorithms
- What attributes shall we use to represent text documents??
- Target concept *Interesting?* : $Document \rightarrow \{\oplus, \ominus\}$
 1. Representation: *Bag of words*. Take the union of all words occurring in all documents. A specific document is represented by a binary vector with 1's in the positions corresponding to words which occur in this document
 2. Learning: Use training examples to estimate

$$P(\oplus)$$

$$P(\ominus)$$

$$P(doc|\oplus)$$

$$P(doc|\ominus)$$

- Naive Bayes conditional independence assumption

$$P(doc|v_j) = \prod_{i=1}^{length(doc)} P(a_i = w_k|v_j)$$

Where $P(a_i = w_k|v_j)$ is probability that a word in position i is w_k , given v_j

One more assumption: $P(a_i = w_k|v_j) = P(a_m = w_k|v_j), \forall i, m$

Learning to Classify Text

Learn_naive_Bayes_text(*Examples*, *V*)

{Collect all words and other tokens that occur in *Examples*}

- *Vocabulary* \leftarrow All distinct words and other tokens in *Examples*

{Calculate the required $P(v_j)$ and $P(w_k|v_j)$ probability terms}

- For each target value v_j in *V* do
 1. $docs_j \leftarrow$ Subset of *Examples* for which the target value is v_j
 2. $P(v_j) \leftarrow \frac{|docs_j|}{|Examples|}$
 3. $Text_j \leftarrow$ A single document created by concatenating all members of $docs_j$
 4. $n \leftarrow$ Total number of words in $Text_j$ (counting duplicate words multiple times)
 5. For each word w_k in *Vocabulary*
 - $n_k \leftarrow$ Number of times word w_k occurs in $Text_j$
 - $P(w_k|v_j) \leftarrow \frac{n_k+1}{n+|Vocabulary|}$

Classify_naive_Bayes_text(*Doc*)

- *positions* \leftarrow All word positions in *Doc* that contain tokens found in *Vocabulary*
- Return v_{NB} , where

$$v_{NB} = \arg \max_{v_j \in V} P(v_j) \prod_{i \in positions} P(a_i|v_j)$$

Incorrectness of NBC's *independence assumption* for text classification: If "Ngom" occurs, the previous word is more likely to be "Alioune" than any other word

Example: Twenty NewsGroups Classification

- Given 1000 training documents from each of 20 newsgroups, learn to classify new documents according to which newsgroup it came from

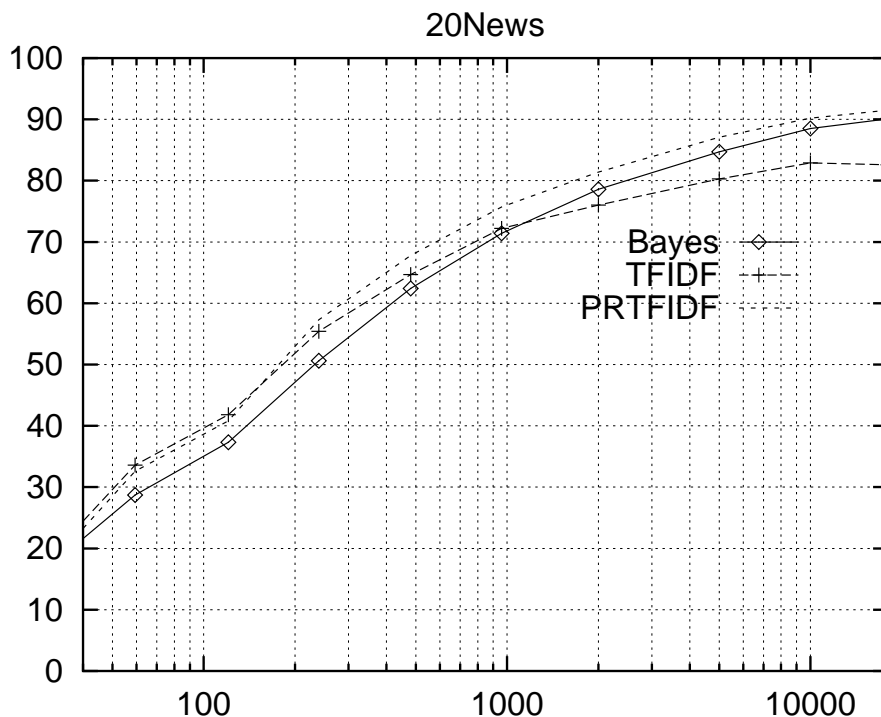
comp.graphics
comp.os.ms-windows.misc
comp.sys.ibm.pc.hardware
comp.sys.mac.hardware
comp.windows.x

misc.forsale
rec.autos
rec.motorcycles
rec.sport.baseball
rec.sport.hockey

alt.atheism
soc.religion.christian
talk.religion.misc
talk.politics.mideast
talk.politics.misc
talk.politics.guns

sci.space
sci.crypt
sci.electronics
sci.med

- Naive Bayes: 89% classification accuracy
- Learning Curve for 20 Newsgroups



Accuracy vs. Training set size (1/3 withheld for test)

Bayesian Belief Networks

- Interesting because:

1. Naive Bayes assumption of conditional independence is too restrictive
2. But it's intractable without some such assumptions . . .
3. Bayesian Belief networks capture conditional independence among *subsets* of variables

→ This allows combining prior knowledge about (in)dependencies among variables with observed training data

- Representation:

1. A Bayes Net is a DAG in which the nodes represent random variable
2. Each node is annotated with a probability distribution $P(X_i|Parents)$ representing the dependency of that node on its parents in the DAG
3. Each node is asserted to be conditionally independent of its non-descendants, given its immediate predecessors
4. Arcs represent direct dependencies

Conditional Independence

Definition: X is *conditionally independent* of Y given Z if the probability distribution governing X is independent of the value of Y given the value of Z ; that is, if

$$(\forall x_i, y_j, z_k) P(X = x_i | Y = y_j, Z = z_k) = P(X = x_i | Z = z_k)$$

More compactly, we write

$$P(X|Y, Z) = P(X|Z)$$

- Example: *Thunder* is conditionally independent of *Rain*, given *Lightning*

$$P(\text{Thunder} | \text{Rain}, \text{Lightning}) = P(\text{Thunder} | \text{Lightning})$$

- Naive Bayes uses conditional independence to justify

$$\begin{aligned} P(X, Y | Z) &= P(X | Y, Z) P(Y | Z) \\ &= P(X | Z) P(Y | Z) \end{aligned}$$

Representation specifying a set of conditional independence assertions:

1. Each node is asserted to be conditionally independent of its nondescendants, given its immediate predecessors.
2. Directed acyclic graph

Bayesian Belief Network

- BBN represents the joint probability distribution over all variables

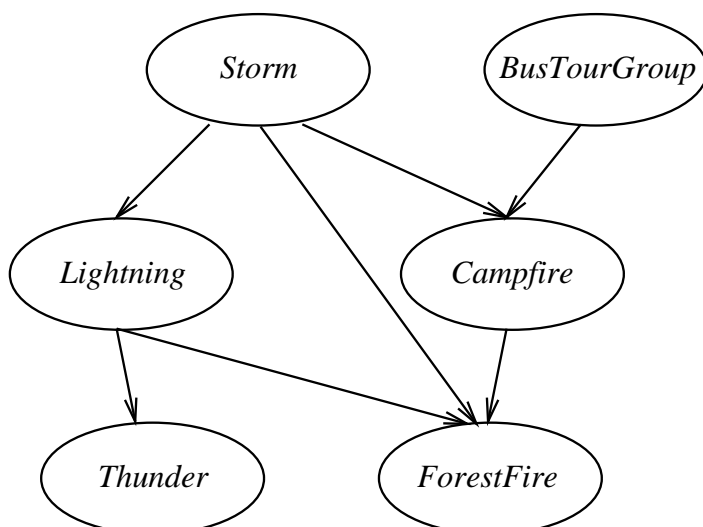
e.g., $P(\text{Storm}, \text{BusTourGroup}, \dots, \text{ForestFire})$

- In general,

$$P(y_1, \dots, y_n) = \prod_{i=1}^n P(y_i | \text{Parents}(Y_i))$$

Where $\text{Parents}(Y_i)$ denotes the immediate predecessors of Y_i in the graph

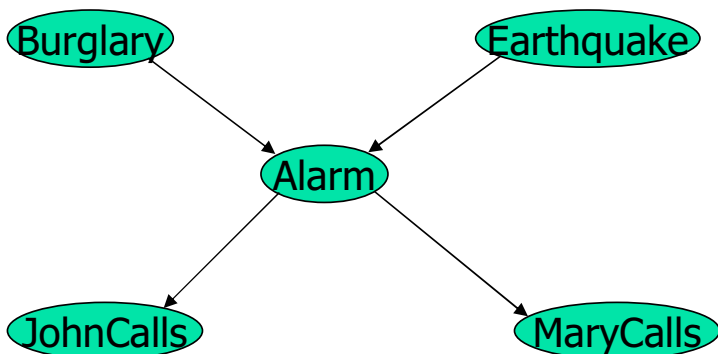
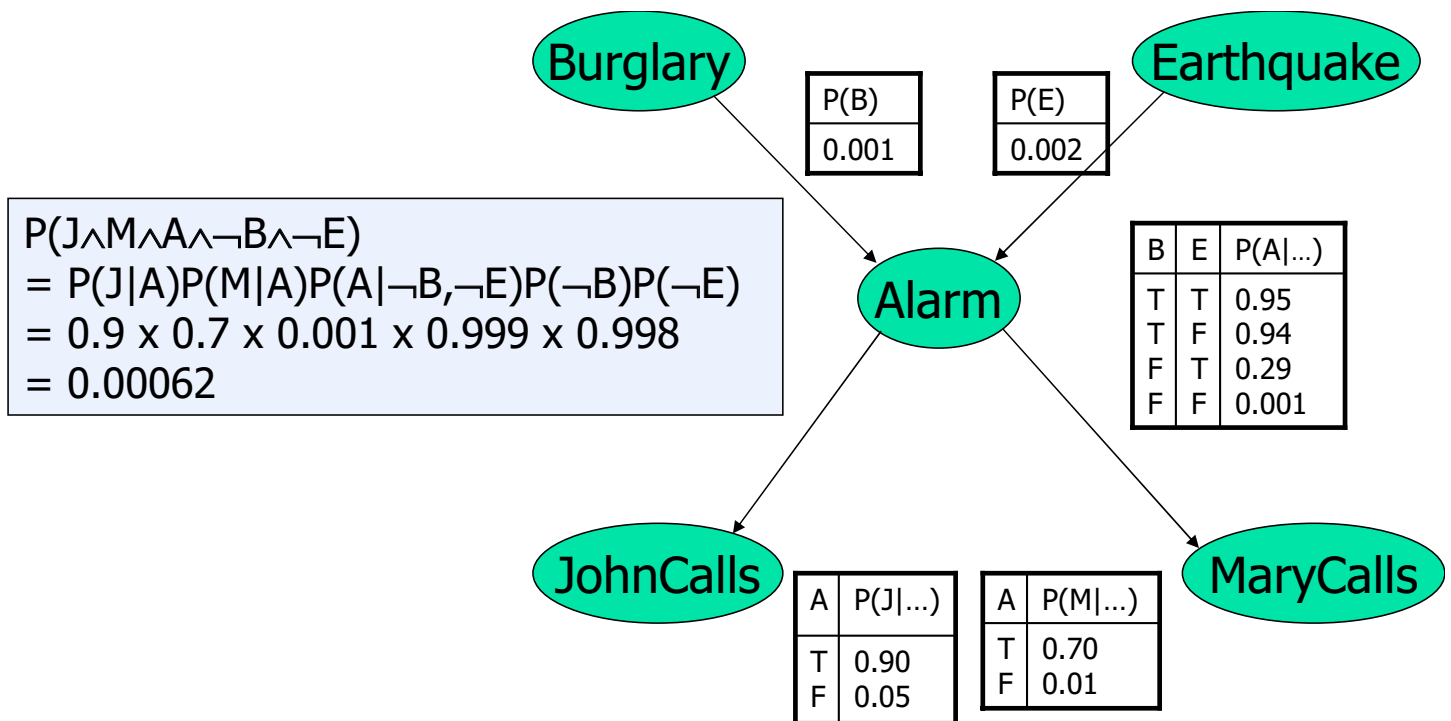
- So, the joint distribution is fully defined by the graph, plus the $P(y_i | \text{Parents}(Y_i))$



	S, B	$S, \neg B$	$\neg S, B$	$\neg S, \neg B$
C	0.4	0.1	0.8	0.2
$\neg C$	0.6	0.9	0.2	0.8



Example



For example, John does not observe any burglaries directly

Each of the beliefs JohnCalls and MaryCalls is independent of Burglary and Earthquake given Alarm or \neg Alarm

The beliefs JohnCalls and MaryCalls are independent given Alarm or \neg Alarm

Usually (but not always) the parents of X are its *causes* and X_i is the *effect* of these causes

Construction of a BBN

Choose the relevant sentences (random variables) that describe the domain

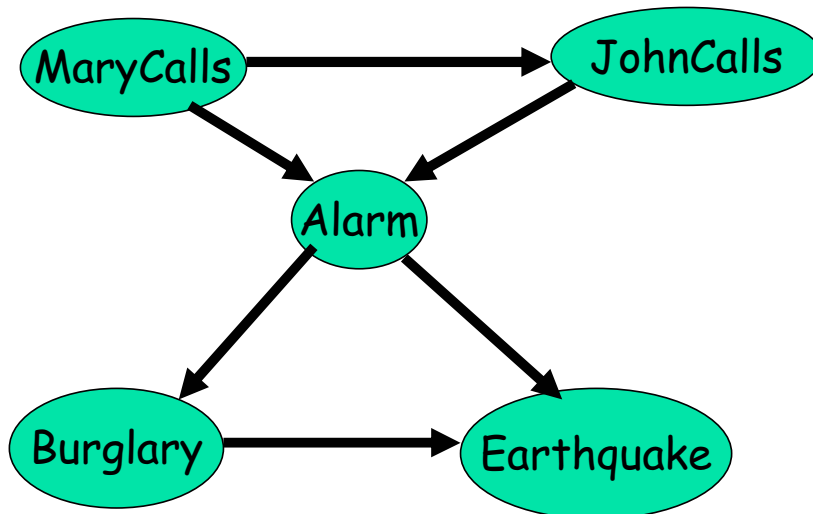
Select an ordering X_1, \dots, X_n , so that all the beliefs that directly influence X_i are before X_i

For $j=1, \dots, n$ do:

- Add a node in the network labeled by X_j
- Connect the node of its parents to X_j
- Define the CPT of X_j

The ordering guarantees that the BN will have no cycles
The CPT guarantees that exactly the correct number of probabilities will be defined

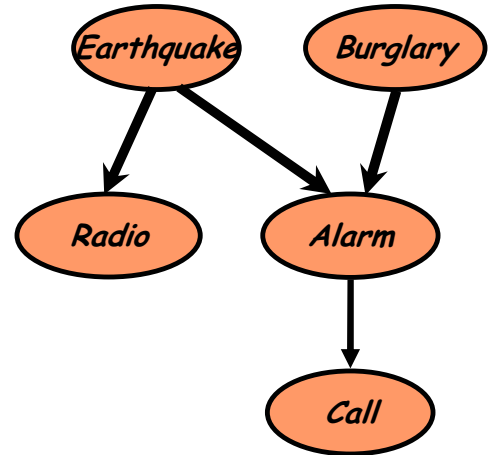
CPT = Conditional Probability Table



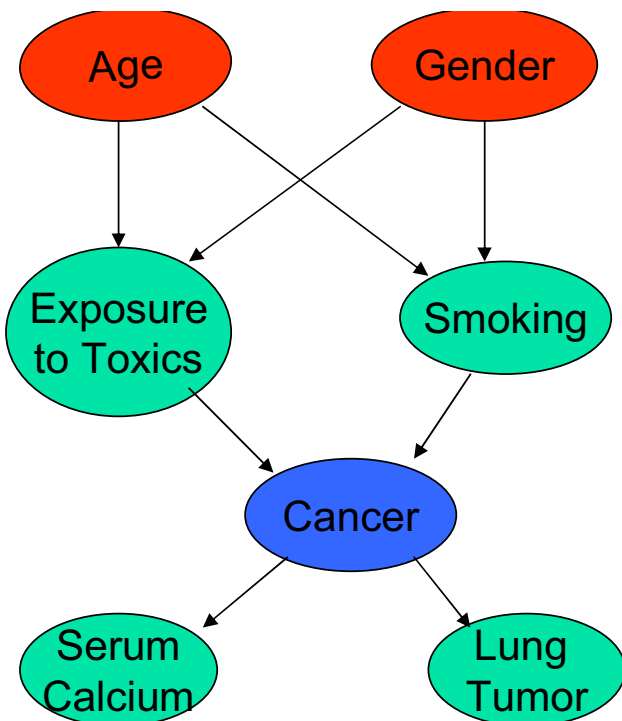
It is easy to create a “bad” Bayesian network. Although mathematically correct, it forces one make unnatural probability judgments. One should start with “root causes” and add variables they influence, etc...

What Can we Do with BBNs?

- Probabilistic inference: belief update
 - $P(E=Y | R=Y, C=Y)$
- Probabilistic inference: belief revision
 - $\text{Argmax}_{\{E,B\}} P(e, b | C=Y)$
- Qualitative inference
 - $I(R,C | A)$
- Complex inference
 - rational decision making (influence diagrams)
 - value of information
 - sensitivity analysis
- Causal inference



Predictive Inference: Cause to Effect

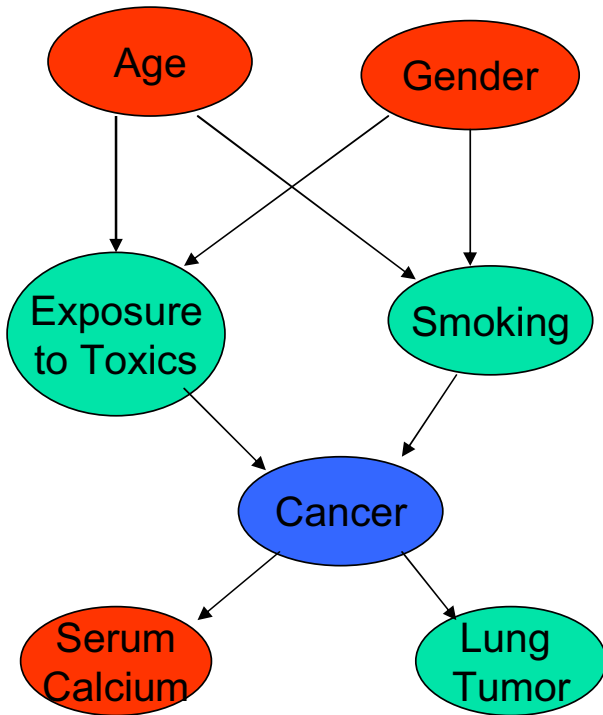


How likely are **elderly males** to get **malignant cancer**?

$$P(C=\text{malignant} | \text{Age}>60, \text{Gender}=\text{male})$$

What Can we Do with BBNs?

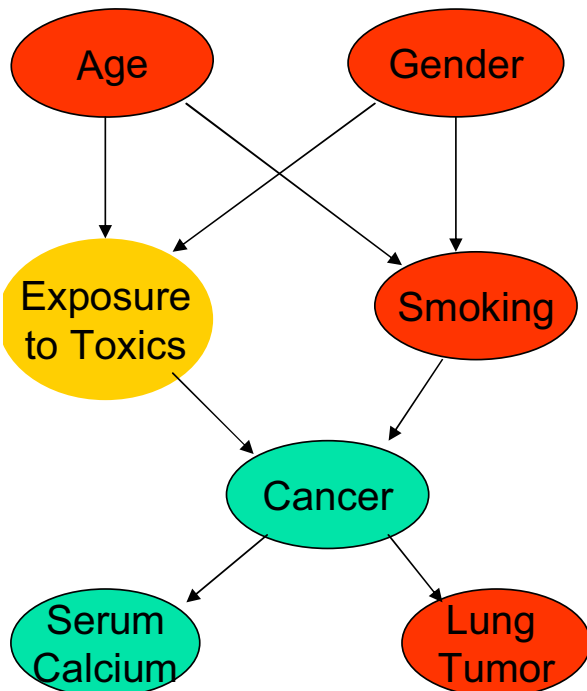
Mixed Inference



How likely is an **elderly male** patient with high serum calcium to have **malignant cancer**?

$$P(C=\text{malignant} \mid \text{Age}>60, \text{Gender}=\text{male}, \text{Serum Calcium} = \text{high})$$

Explaining Away



- If we observe a **lung tumor**, the probability of **heavy smoking** and of **exposure to toxics** both go up.
- If we then observe **heavy smoking** the probability of **exposure to toxics** goes back down.

Inference in Bayesian Networks

- How can one infer the (probabilities of) values of one or more network variables, given observed values of others?
 1. Bayes net contains all information needed for this inference
 2. If we only have one variable with unknown value, easy to infer it
 3. In general case, problem is NP hard
- In practice, we can succeed in many cases
 1. Exact inference methods work well for some network structures
 2. Monte Carlo methods “simulate” the network randomly to calculate approximate solutions

Learning Bayesian Networks

- Several variants of this learning task
 1. Network structure might be *known* or *unknown*
 2. Training examples might provide values of *all* network variables, or just *some*
- If structure is known and all variables are observable

Then it's as easy as training a Naive Bayes Classifier
- If structure known but variables partially observable

e.g., observe *ForestFire*, *Storm*, *BusTourGroup*, *Thunder*, but not *Lightning*, *Campfire* . . .

 1. Similar to training neural network with hidden units
 2. In fact, we can learn the network's conditional probability tables using gradient ascent!
 3. Will converge to a network h that (locally) maximizes $P(D|h)$, i.e. the *maximum likelihood* hypothesis

Gradient Ascent for Bayes Nets

Let w_{ijk} denote one entry in the conditional probability table for variable Y_i in the network

$$w_{ijk} = P(Y_i = y_{ij} | Parents(Y_i) = \text{the list } u_{ik} \text{ of values})$$

e.g., if $Y_i = \text{Campfire}$

then u_{ik} might be $\langle \text{Storm} = T, \text{BusTourGroup} = F \rangle$

Perform gradient ascent by repeatedly

1. Update all w_{ijk} using training data D

$$w_{ijk} \leftarrow w_{ijk} + \eta \sum_{d \in D} \frac{P_h(y_{ij}, u_{ik} | d)}{w_{ijk}}$$

2. Then, renormalize the w_{ijk} to assure

- $\sum_j w_{ijk} = 1$
- $0 \leq w_{ijk} \leq 1$

More on Learning Bayes Nets

- EM algorithm can also be used

Repeatedly:

1. Calculate probabilities of unobserved variables, assuming h
 2. Calculate new w_{ijk} to maximize $E[\ln P(D|h)]$ where D now includes both observed and (calculated probabilities of) unobserved variables
- When structure is unknown . . .
 - Algorithms use greedy search to add/subtract edges and nodes
 - Constraint-based approaches
 - Genetic methods
 - . . . Active research topic

Summary: Bayesian Belief Networks

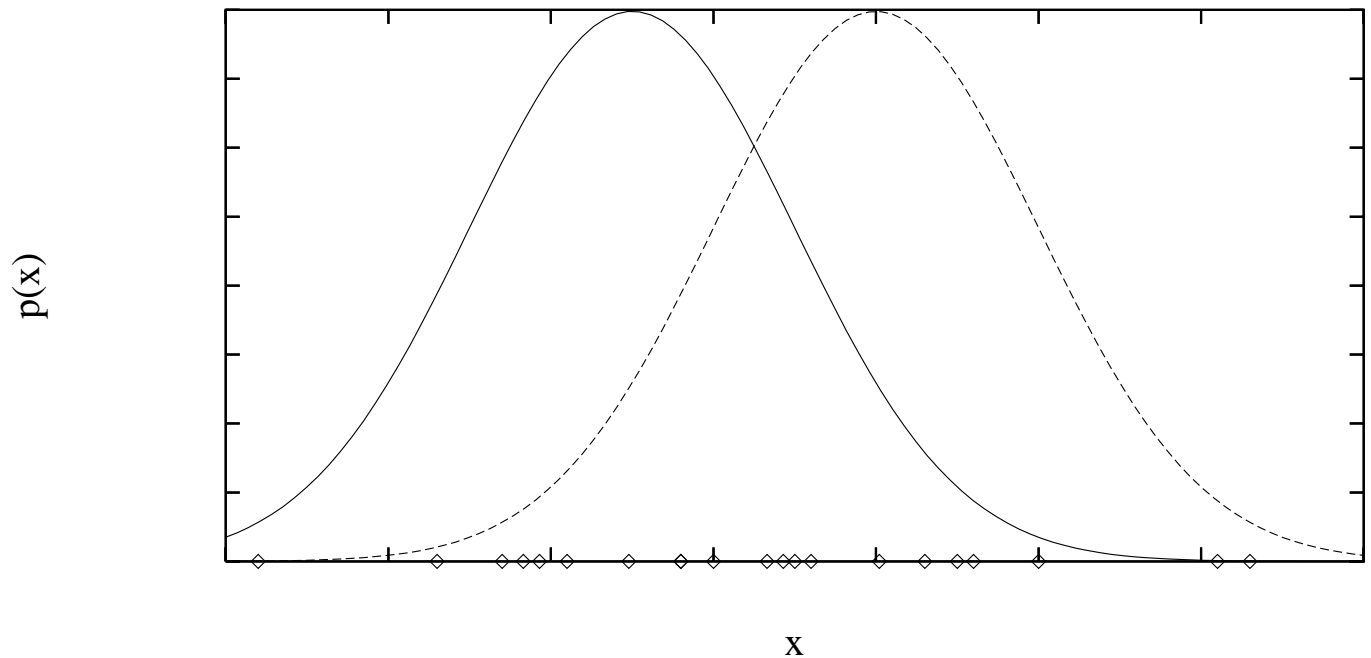
- Combine prior knowledge with observed data
- Impact of prior knowledge (when correct!) is to lower the sample complexity
- Active research area
 1. Extend from boolean to real-valued variables
 2. Parameterized distributions instead of tables
 3. Extend to first-order instead of propositional systems
 4. More effective inference methods
 5. . . .

Expectation Maximization (EM)

- When to use:
 1. Data is only partially observable
 2. Unsupervised clustering (target value unobservable)
 3. Supervised learning (some instance attributes unobservable)
- Some uses:
 1. Train Bayesian Belief Networks
 2. Unsupervised clustering (AUTOCLASS)
 3. Learning Hidden Markov Models

Generating Data from Mixture of k Gaussians

- Each instance x generated by
 1. Choosing one of the k Gaussians with uniform probability
 2. Generating an instance at random according to that Gaussian



EM for Estimating k Means

- Given:
 1. Instances from X generated by mixture of k Gaussian distributions
 2. Unknown means $\langle \mu_1, \dots, \mu_k \rangle$ of the k Gaussians
 3. Don't know which instance x_i was generated by which Gaussian

- Determine:

Maximum likelihood estimates of $\langle \mu_1, \dots, \mu_k \rangle$

- Think of full description of each instance as $y_i = \langle x_i, z_{i1}, z_{i2} \rangle$, where
 1. z_{ij} is 1 if x_i generated by j -th Gaussian
 2. x_i observable
 3. z_{ij} unobservable

EM for Estimating k Means

- EM Algorithm:

Pick random initial $h = \langle \mu_1, \mu_2 \rangle$, then iterate

1. **E step:** Calculate the expected value $E[z_{ij}]$ of each hidden variable z_{ij} , assuming the current hypothesis $h = \langle \mu_1, \mu_2 \rangle$ holds.

$$\begin{aligned} E[z_{ij}] &= \frac{p(x = x_i | \mu = \mu_j)}{\sum_{n=1}^2 p(x = x_i | \mu = \mu_n)} \\ &= \frac{e^{-\frac{1}{2\sigma^2}(x_i - \mu_j)^2}}{\sum_{n=1}^2 e^{-\frac{1}{2\sigma^2}(x_i - \mu_n)^2}} \end{aligned}$$

2. **M step:** Calculate a new maximum likelihood hypothesis $h' = \langle \mu'_1, \mu'_2 \rangle$, assuming the value taken on by each hidden variable z_{ij} is its expected value $E[z_{ij}]$ calculated above. Replace $h = \langle \mu_1, \mu_2 \rangle$ by $h' = \langle \mu'_1, \mu'_2 \rangle$.

$$\mu_j \leftarrow \frac{\sum_{i=1}^m E[z_{ij}] x_i}{\sum_{i=1}^m E[z_{ij}]}$$

EM Algorithm

- Converges to local maximum likelihood h
- ... and provides estimates of hidden variables z_{ij}
- In fact, local maximum in $E[\ln P(Y|h)]$
 1. Y is complete (observable plus unobservable variables) data
 2. Expected value is taken over possible values of unobserved variables in Y

General EM Problem

- Given:

1. Observed data $X = \{x_1, \dots, x_m\}$

2. Unobserved data $Z = \{z_1, \dots, z_m\}$

3. Parameterized probability distribution $P(Y|h)$, where

$Y = \{y_1, \dots, y_m\}$ is the full data $y_i = x_i \cup z_i$

h are the parameters

- Determine:

h that (locally) maximizes $E[\ln P(Y|h)]$

- Many uses:

1. Train Bayesian belief networks

2. Unsupervised clustering (e.g., k means)

3. Hidden Markov Models

General EM Method

- Define likelihood function $Q(h'|h)$ which calculates $Y = X \cup Z$ using observed X and current parameters h to estimate Z

$$Q(h'|h) \leftarrow E[\ln P(Y|h')|h, X]$$

- EM Algorithm:

1. **Estimation (E) step:** Calculate $Q(h'|h)$ using the current hypothesis h and the observed data X to estimate the probability distribution over Y .

$$Q(h'|h) \leftarrow E[\ln P(Y|h')|h, X]$$

2. **Maximization (M) step:** Replace hypothesis h by the hypothesis h' that maximizes this Q function.

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