Statistical learning Methods

Chapter 20, Sections 1–3

Outline

♢ Bayesian learning

♢ Maximum likelihood and linear regression

Full Bayesian learning

View learning as Bayesian updating of probability distribution over the hypothesis space

Prior $\mathbf{P}(H)$, data $\mathbf{e} = e_1, \ldots, e_N$

Given the data so far, each hypothesis has a posterior probability:

$$P(h_i|\mathbf{e}) = \alpha P(\mathbf{e}|h_i)P(h_i)$$

Where:

$$\alpha = \frac{1}{P(e)} = \frac{1}{\sum_i P(\mathbf{e}|h_i)P(h_i)}$$

For diagnosis, just pick the hypothesis with the maximum a-posteriori probability (MAP).
This is called h_{MAP}.

MAP Learning

$$
\begin{aligned}
h_{Map} &= arg \max_{h_i \in H} P(h_i|\mathbf{e}) \\
&= arg \max_{h_i \in H} \frac{P(\mathbf{e}|h_i)P(h_i)}{\sum_i P(\mathbf{e}|h_i)P(h_i)} \\
&= arg \max_{h_i \in H} P(\mathbf{e}|h_i)P(h_i)
\end{aligned}
$$

This follows from a) Bayes Rule, and b) Common denominator for all terms we are maximizing over.

In addition, if each hypothesis (h_i) has the same a-priori probability, then:

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

Where h_{ML} is the Maximum-Likelihood hypothesis; i.e. the one that makes the evidence most likely.

Prediction Using Likelihood Weighting

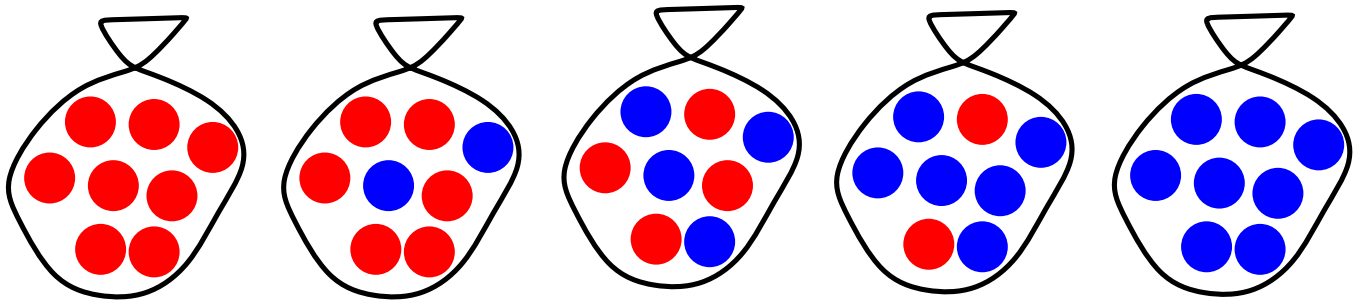
To predict the most likely next observation of X, use a likelihood-weighted average over the hypotheses:

$$
\begin{aligned}
\mathbf{P}(X|\mathbf{e}) &= \frac{\sum_i \mathbf{P}(X|\mathbf{e}, h_i) P(\mathbf{e}, h_i)}{P(\mathbf{e})} \\
&= \frac{\sum_i \mathbf{P}(X|\mathbf{e}, h_i) P(h_i|\mathbf{e}) P(\mathbf{e})}{P(\mathbf{e})} \\
&= \sum_i \mathbf{P}(X|\mathbf{e}, h_i) P(h_i|\mathbf{e}) \\
&= \sum_i \mathbf{P}(X|h_i) P(h_i|\mathbf{e}) \quad \text{X is cond. indep. of e, given } h_i
\end{aligned}
$$

Marble Example

Suppose there are five kinds of bags of marbles:

10% are h_1: 100% blue marbles
20% are h_2: 75% blue marbles + 25% red marbles
40% are h_3: 50% blue marbles + 50% red marbles
20% are h_4: 25% blue marbles + 75% red marbles
10% are h_5: 100% red marbles



Then we observe marbles drawn from JUST ONE of the bags:

Abduction or Diagnosis: What kind of bag is it?
Prediction: What color will the next marble be?

Abduction: Which Bag?

After seeing the first 3 red marbles, find posterior probs for each hypothesis?

$$\begin{aligned} P(h_2 \mid \text{3 red marbles}) &= \alpha P(\text{3 red marbles} \mid h_2) P(h_2) \\ &= \alpha(.25)^3(.2) = \alpha(.003125) \end{aligned}$$

Similarly:

$$\begin{aligned} P(h_1 \mid \text{3 red marbles}) &= \alpha(0)^3(.1) = \alpha(0) \\ P(h_3 \mid \text{3 red marbles}) &= \alpha(.5)^3(.4) = \alpha(.05) \\ P(h_4 \mid \text{3 red marbles}) &= \alpha(.75)^3(.2) = \alpha(.084) \\ P(h_5 \mid \text{3 red marbles}) &= \alpha(1)^3(.1) = \alpha(.1) \end{aligned}$$

Abduction: Which Bag? (2)

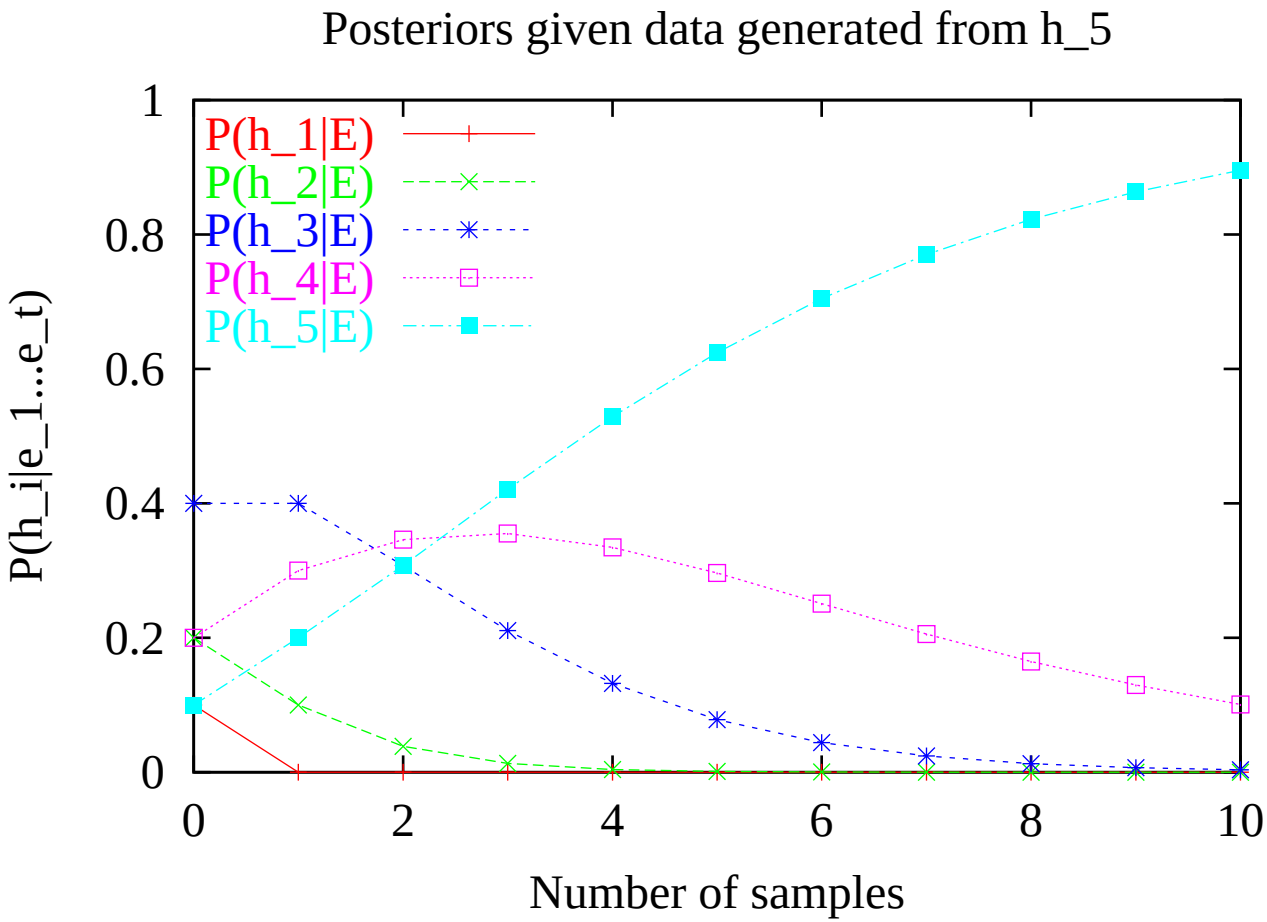
The normalizing constant is just the inverse of the sum of the numerators:
$\alpha = \frac{1}{.003125+0+.05+.084+.1} = \frac{1}{.237125} = 4.2172$

So:

$$
\begin{aligned}
P(h_1 \mid \text{3 red marbles}) &= 0 \\
P(h_2 \mid \text{3 red marbles}) &= .013 \\
P(h_3 \mid \text{3 red marbles}) &= .211 \\
P(h_4 \mid \text{3 red marbles}) &= .354 \\
P(h_5 \mid \text{3 red marbles}) &= .422
\end{aligned}
$$

Hence, $h_{MAP} = h_5$
Note: Since we are maximizing, we didn't need to compute α.

Posterior probability of hypotheses



Prediction: Color of Next Marble?

- After seeing 3 red marbles (**3R**), compute p(4th marble = red) (**P(R4)**) and p(4th marble = blue) (**P(B4)**).
- Here, the hypothesized bags serve as intermediaries that link the past evidence (3 red marbles) to the future (4th marble).

From our earlier derivation of the likelihood of new evidence:

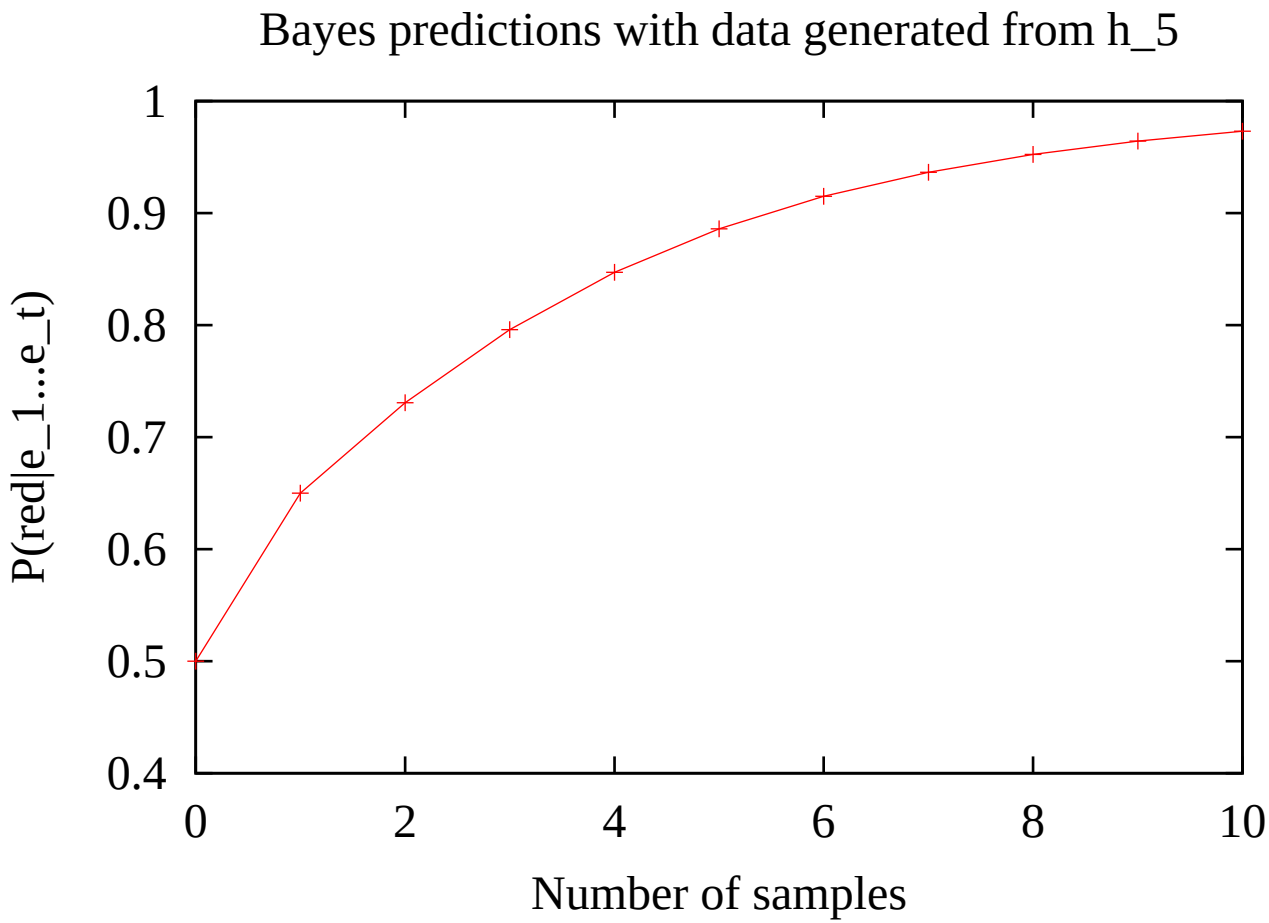
$$\mathbf{P}(X|\mathbf{e}) \;=\; \textstyle\sum_i \mathbf{P}(X|h_i)P(h_i|\mathbf{e})$$

Applying to this example and using the posterior probs for the 5 hypotheses (given 3R):

$$\begin{aligned} P(R4 \mid 3R) &= \textstyle\sum_i \mathbf{P}(R4|h_i)P(h_i|3R) \\ &= (0)(0) + (.25)(.013) + (.5)(.211) + (.75)(.354) + (1)(.422) = .79625 \\ P(B4 \mid 3R) &= \textstyle\sum_i \mathbf{P}(B4|h_i)P(h_i|3R) \\ &= (1)(0) + (.75)(.013) + (.5)(.211) + (.25)(.354) + (0)(.422) = .20375 \end{aligned}$$

Note: $P(B4 \mid 3R) = 1 - P(R4 \mid 3R)$.

Prediction probability



As we see more red marbles, $P(h_5 \mid \mathbf{e}) \uparrow$, and this increases our belief that the next marble will be red.

MAP and MDL

Maximum a posteriori (MAP) learning: choose h_{MAP} maximizing $P(h_i|\mathbf{e})$

$$\begin{aligned} arg\max_{h_i \in H} P(\mathbf{e}|h_i)P(h_i) &= arg\max_{h_i \in H} \log_2 P(\mathbf{e}|h_i) + \log_2 P(h_i) \\ &= arg\min_{h_i \in H} -\log_2 P(\mathbf{e}|h_i) - \log_2 P(h_i) \\ &= h_{MDL} \text{ Minimal Description Length Hypothesis} \end{aligned}$$

- $-\log_2 P(h_i)$: Bits to encode hypothesis
- $-\log_2 P(\mathbf{e}|h_i)$: Bits to encode extra data, given the hypothesis: the exceptions.

Information Theory:

- An optimal encoding of information is one where the length of the code for a particular item is $f(-\log_2 P(i))$.
- Hence, use short codes for highly probable items, and long codes for less-probable items.

So, by maximizing the a-posterior probability, we are minimizing the descriptive length under an optimal encoding scheme: $h_{MAP} = h_{MDL}$

Information-Exchange Scenario

- The sender (S) and receiver (R) both already have information about a particular problem set. I.e., they have a set of attribute instances that need to be classified.
- S needs to transfer a hypothesized answer (h) to R.
- R can then use h to compute the classes for some (or all) of the instances.
- The answer = h + any **exceptions**, i.e. instances whose class is not computed from h but is just listed (..(instance-index class-index) ...).
- S & R may also both have an enumerated space of possible hypotheses, H.
- Then S merely transmits the index, k, of the best hypothesis, $h_k \in H$, along with any exceptions.
- If S & R have worked with these types of problems before, then they may have an a-priori probability distribution over all $h_i \in H$.
- Then they can devise an optimal coding where higher probability h_i have shorter encodings than lower probability h_i, where length $\approx -\log_2 p(h_i)$.

Simplifying MAP Search

- Summing over the hypothesis space is often intractable
- E.g. $2^{2^6} = 18,446,744,073,709,551,616$ Boolean functions of 6 vars

For deterministic hypotheses, $P(\mathbf{e}|h_i)$ is 1 if consistent, 0 otherwise

$\Rightarrow$ MAP = simplest consistent hypothesis (cf. science)

- So if we have deterministic hypotheses, then we should begin our search for h_{MAP} with shorter hypotheses.
- As soon as we find one that is consistent with all the data, then we have h_{MAP}.
- So, only in the very worst case do we need to consider the whole hypothesis space.

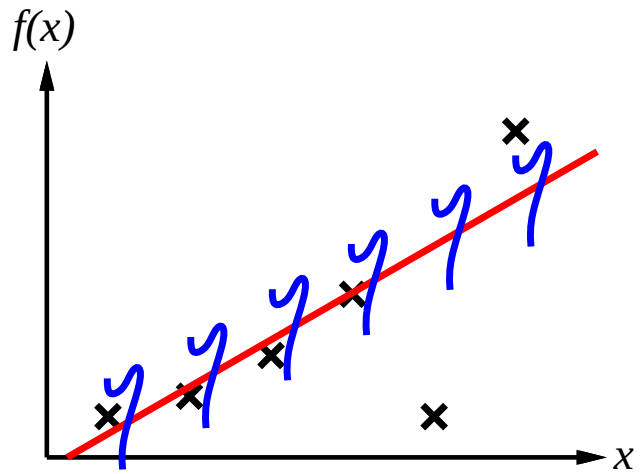
ML approximation for MAP

- For large data sets, prior becomes irrelevant
- Maximum likelihood (ML) learning: choose h_{ML} maximizing $P(\mathbf{e}|h_i)$
- In short, get the best fit to the data (ignoring prior probabilities of hypotheses).
- Same as MAP with assump that $p(h_i) = p(h_j) \forall i, j$
- This is reasonable if all hypotheses are of the same complexity.
- ML is the "standard" (non-Bayesian) statistical learning method

Example: linear regression

Data: pairs $(x_1, y_1), \ldots, (x_N, y_N)$

Hypotheses: straight lines $y = ax + b$ with Gaussian noise
Want to choose parameters $\theta = (a, b)$ to maximize likelihood of data



Linear regression contd.

Data assumed i.i.d. (independently and identically distributed)

$\Rightarrow$ likelihood $P(\mathbf{e}|h_i) = \prod_j P(e_j|h_i)$

Maximizing likelihood $P(\mathbf{e}|h_i) \Leftrightarrow$ maximizing log likelihood

$$L = \log P(\mathbf{e}|h_i) = \log \prod_j P(e_j|h_i) = \sum_j \log P(e_j|h_i)$$

For a continuous hypothesis space, set $\partial L/\partial\theta = 0$ and solve for θ

For Gaussian noise, $P(e_j|h_i) = \alpha \exp\left(-(y_j - (ax_j + b))^2/2\sigma^2\right)$, so

$$L = \sum_j \log P(e_j|h_i) = -\alpha' \sum_j (y_j - (ax_j + b))^2$$

so maximizing L = minimizing sum of squared errors

Linear regression contd.

To find the maximum, set derivatives to zero:

$$\frac{\partial L}{\partial a} = -\alpha' \Sigma_j \, 2(y_j - (ax_j + b)) \cdot (-x_j) = 0$$
$$\frac{\partial L}{\partial b} = -\alpha' \Sigma_j \, 2(y_j - (ax_j + b)) \cdot (-1) = 0$$

Solutions are

$$a = \frac{\Sigma_j \, x_j \Sigma_j \, y_j - N \Sigma_j \, x_j y_j}{\left(\Sigma_j \, x_j\right)^2 - N \Sigma_j \, x_j^2} \; ; \quad b = \left(\Sigma_j \, y_j - a \, \Sigma_j \, x_j\right) / N$$

Learning with Complete Data

Complete $\rightarrow$ Each training instance has a value for each variable in the underlying probability model.

Two main types

1. Learning the parameters of a model. E.g. the values in a Bayesian Network Conditional Probability Table.
2. Learning structure of a model. E.g. topology of a Bayesian Net.

Returning to the Earthquake example:

- Assume that we want to learn $P(\text{Mary Calls} = \text{T} \mid \text{Earthquake} = \text{T})$ from a large set of atomic events $(\mathbf{e})$.
- We can do this by simply counting the events with Earthquake $=$ T, and then finding the fraction of those with Mary Calls $=$ T.
- It turns out that this also agrees with the results of ML analysis.

Maximum-Likelihood Parameter Learning

- Let $\theta = P(\text{Mary Calls} = \text{T} \mid \text{Earthquake} = \text{T})$
- Then we have an infinite number of hypotheses, h_θ, for all possible values of θ.
- To compute the h_θ that best explains $\mathbf{e}$ (i.e. h_{ML}), we need to maximize the likelihood of $\mathbf{e}$:

$$P(\mathbf{e} \mid h_\theta) \; = \; \Pi_{i=1}^{n} P(e_i \mid h_\theta)$$

- This assumes conditional independence of the evidence, given h_θ.
- When maximizing, it is convenient to maximize the log, since that reduces a product to a sum.

$$arg \max_{h_\theta \in H} \Pi_{i=1}^{n} P(e_i \mid h_\theta) \; = \; arg \max_{h_\theta \in H} \Sigma_{i=1}^{n} \log P(e_i \mid h_\theta)$$

Maximum-Likelihood Parameter Learning (2)

- Let $f(\mathbf{e}, \theta) = \sum_{i=1}^{n} \log P(e_i \mid h_\theta)$
- Then we find the θ that maximizes $f(\mathbf{e}, \theta)$ by setting the derivative to 0:

$$\frac{df(\mathbf{e}, \theta)}{d\theta} = 0$$

- and solving for θ.
- $P(e_i \mid h_\theta)$ may or may not be easy to differentiate and solve.
- But the optimal θ, i.e. the h_{ML}, is exactly the same value as we would get by counting the atomic events!
- See section 20.2 (pp 716-718) for a more concrete example.

Naive Bayes Models

Diagnosis example from chapter 13 notes:

$$\begin{aligned} & \mathbf{P}(Cause \mid Effect_1, \ldots, Effect_n) \\ = \; & \alpha \mathbf{P}(Effect_1, \ldots, Effect_n | Cause) \mathbf{P}(Cause) \\ = \; & \alpha \mathbf{P}(Cause) \prod_i \mathbf{P}(Effect_i | Cause) * \end{aligned}$$

*By conditional independence of the effects (given the cause),

Called *naive*, since the effects are often **assumed** conditionally independent, even though their relationships may not be well understood.

Naive Bayesian Learning Methods

This also works well for concept learning. In fact, it is the most common Bayesian method for machine learning:

$$\begin{aligned} & \mathbf{P}(Class \mid Attribute_1, \ldots, Attribute_n) \\ = \ & \alpha \mathbf{P}(Attribute_1, \ldots, Attribute_n | Class) \mathbf{P}(Class) \\ = \ & \alpha \mathbf{P}(Class) \Pi_i \mathbf{P}(Attribute_i | Class) * \end{aligned}$$

*Here, the attributes are assumed conditionally independent, given the class.

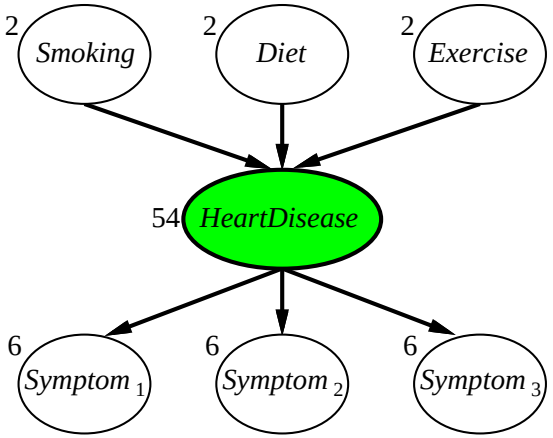
- Given attributes, choose the h_{MAP} class.
- When all classes have equal prior probs (quite common), then $h_{ML} = h_{MAP}$.
- No search needed.
- Handles noise very well
- Boosted version is one of best general-purpose learning algorithms.

Learning Bayesian Net Topologies

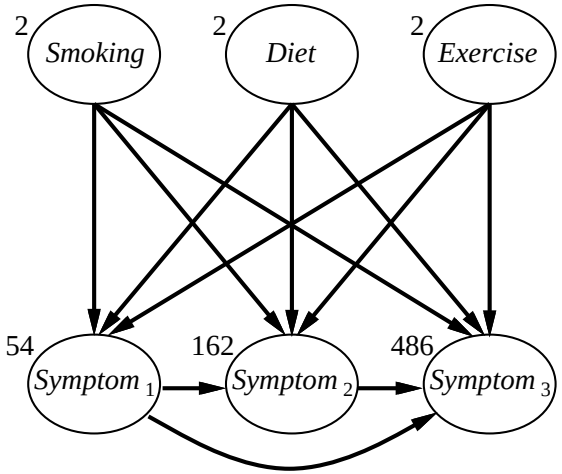
- What are the key variables?
- How are they connected?
  - What vars are independent of all others?
  - What vars are conditionally dependent upon what other vars?
  - What vars are cond indep of others, given a 3rd set of vars?
- This is a difficult search in a space of acyclic graphs.
- Testing of generated topologies:
  - Are conditional independences in topology actually true in the data set? The numbers will not work out exactly, so use statistical confidence levels.
  - Penalize overly-complex networks. The likelihood of evidence given a model never decreases if extra links are added to that model, so maximum-likelihood nets can be bloated with unecessary links

Learning from Incomplete Data

- Hidden Variables: Some key factors in a situation may not be captured by the model variables.
- Find these hidden factors, and make variables for them.
- This can GREATLY reduce the needed prior probabilities.



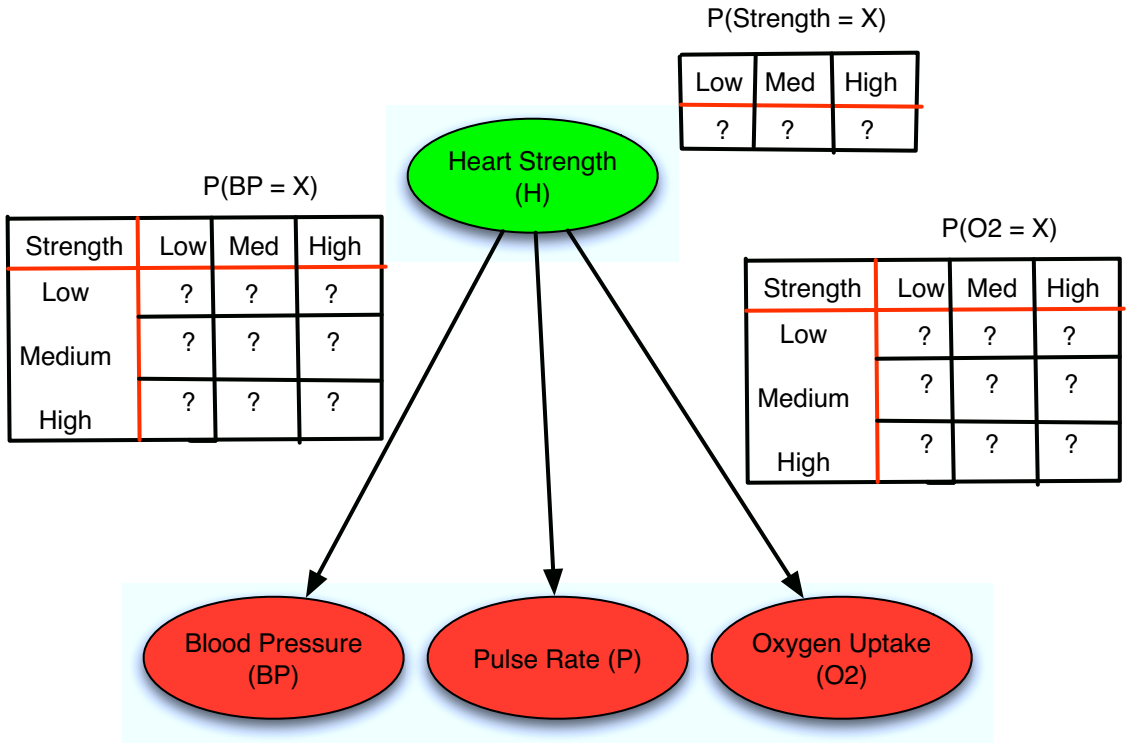
(a)



(b)

- In above network for heart diagnosis, each variable has 3 possible values.
- Finding the hidden variable $\longrightarrow$ 708 priors reduced to 78!!

Parameter Learning for Hidden Variables



Evidence

BP = high, P = low, O2 = high

BP = high, P = high, O2 = low

BP = med, P = med, O2 = high

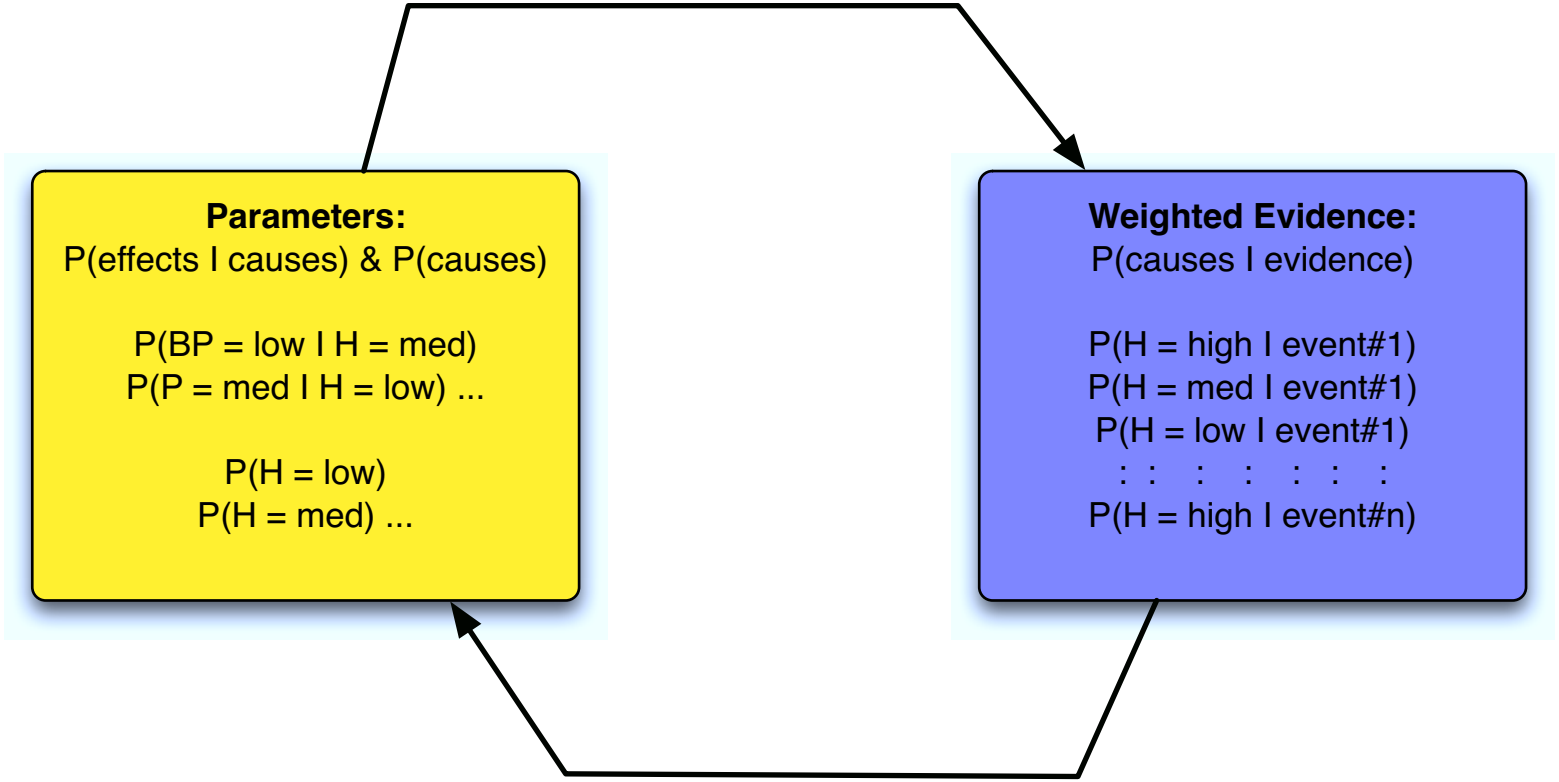
: : : : : : : : : : : : :

Given: a hypothesized hidden var + evidence (atomic events over the effects/symptoms).

Compute: The probs in tables connecting the hidden var to the effect vars.

Expectation Maximization

E Step: For each evidence event (E), use the parameters (and Bayes Rule) to compute the probability distribution over all the possible causes of E (i.e. weight the evidence w.r.t. each cause)



M Step: Update the estimates of the parameters based on the weighted evidence

The Expectation (E) Step

$$
\begin{aligned}
P(cause_i \mid \mathbf{e}_k) &= \frac{P(\mathbf{e}_k \mid cause_i)P(cause_i)}{p(\mathbf{e}_k)} \\
&= \alpha P(cause_i)\prod_j P(\mathbf{e}_{k,j} \mid cause_i)
\end{aligned}
$$

By the conditional indep of the effects, given the cause. Or, in classification tasks, the cond indep of the attributes, given the class.

In the heart example, let e_1 = (BP = high, P = med, O2 = low).

$$
\begin{aligned}
P(H = low \mid \mathbf{e}_1) &= \frac{P(\mathbf{e}_1 \mid H = low)P(H = low)}{p(\mathbf{e}_1)} \\
&= \alpha P(H = low)P(BP = high \mid H = low) \\
&\quad P(P = med \mid H = low)P(O2 = low \mid H = low)]
\end{aligned}
$$

Do same calc for each evidence event and each possible cause (i.e., 3 values of H)

The Maximization (M) Step

1. Recompute the a-priori probability estimates based on evidence.

$$P(cause_i) = \frac{1}{N}\Sigma_k P(cause_i \mid \mathbf{e}_k)$$

The values in the sum were all computed on the E step.
From the heart example:

$$P(H = high) = \frac{1}{N}\Sigma_{k=1}^{N} P(H = high \mid \mathbf{e}_k)$$
$$P(H = med) = \frac{1}{N}\Sigma_{k=1}^{N} P(H = med \mid \mathbf{e}_k)$$
$$P(H = low) = \frac{1}{N}\Sigma_{k=1}^{N} P(H = low \mid \mathbf{e}_k)$$

Sum the probabilities of the cause over all the evidence.

The Maximization (M) Step (2)

2. Recompute the conditional probability estimates based on evidence.

$$
\begin{aligned}
P(effect_j \mid cause_i) &= \frac{P(effect_j \wedge cause_i)}{P(cause_i)} \\
&= \frac{\frac{1}{N}\sum_{\mathbf{e}_k \in S} P(cause_i \mid \mathbf{e}_k)}{\frac{1}{N}\sum_k P(cause_i \mid \mathbf{e}_k)} \\
&= \frac{\sum_{\mathbf{e}_k \in S} P(cause_i \mid \mathbf{e}_k)}{\sum_k P(cause_i \mid \mathbf{e}_k)}
\end{aligned}
$$

Where $S = \{$all $\mathbf{e}_i$ in which $effect_j$ is true $\}$

M Step for Heart Example

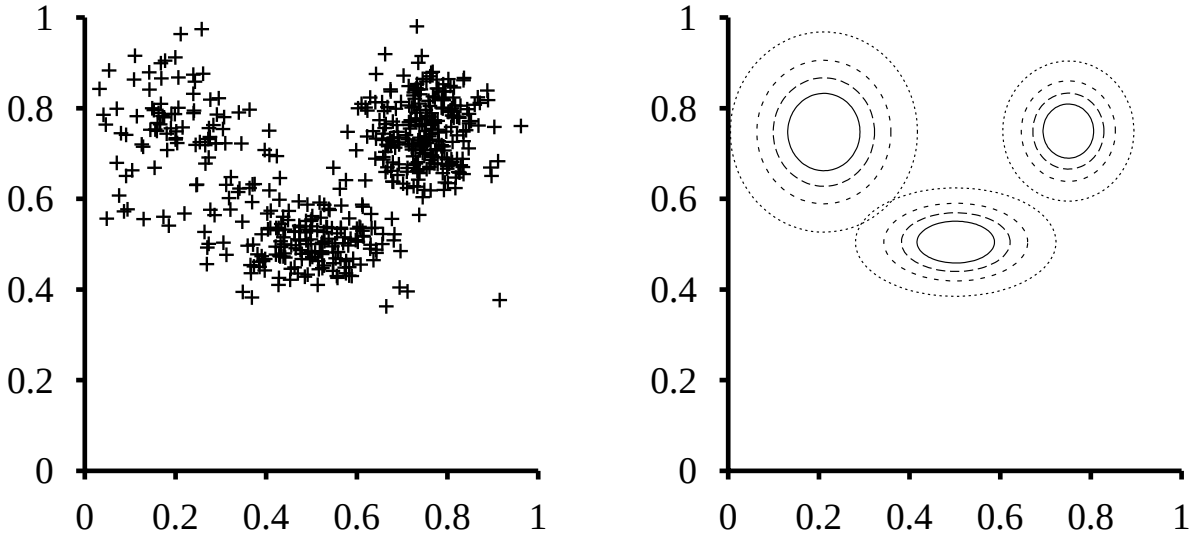
$$
\begin{aligned}
P(O2 = high \mid H = low) &= \frac{P(O2 = high \wedge H = low)}{P(H = low)} \\
&= \frac{\frac{1}{N}\sum_{\mathbf{e}_k \in S} P(H = low \mid \mathbf{e}_k)}{\frac{1}{N}\sum_k P(H = low \mid \mathbf{e}_k)} \\
&= \frac{\sum_{\mathbf{e}_k \in S} P(H = low \mid \mathbf{e}_k)}{\sum_k P(H = low \mid \mathbf{e}_k)}
\end{aligned}
$$

Where $S = \{$all $\mathbf{e}_i$ in which O2 = high $\}$

$$
P(BP = low \mid H = medium) = \frac{\sum_{\mathbf{e}_k \in S*} P(H = medium \mid \mathbf{e}_k)}{\sum_k P(H = medium \mid \mathbf{e}_k)}
$$

Where $S* = \{$all $\mathbf{e}_i$ in which BP = low $\}$
Do same calc for all conditional probabilities being estimated.

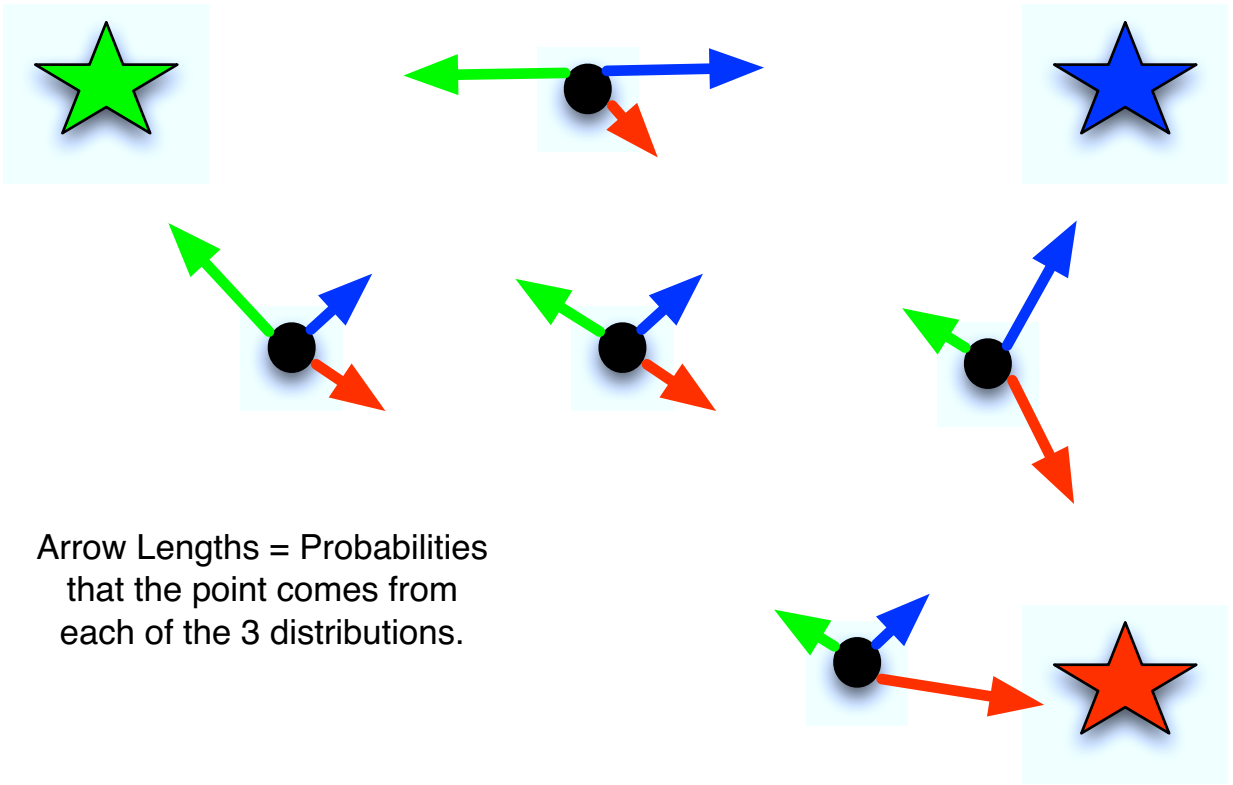
Mixtures of Distributions



Parameters: Means and Variances for the 3 unknown distributions.
Evidence: The data points

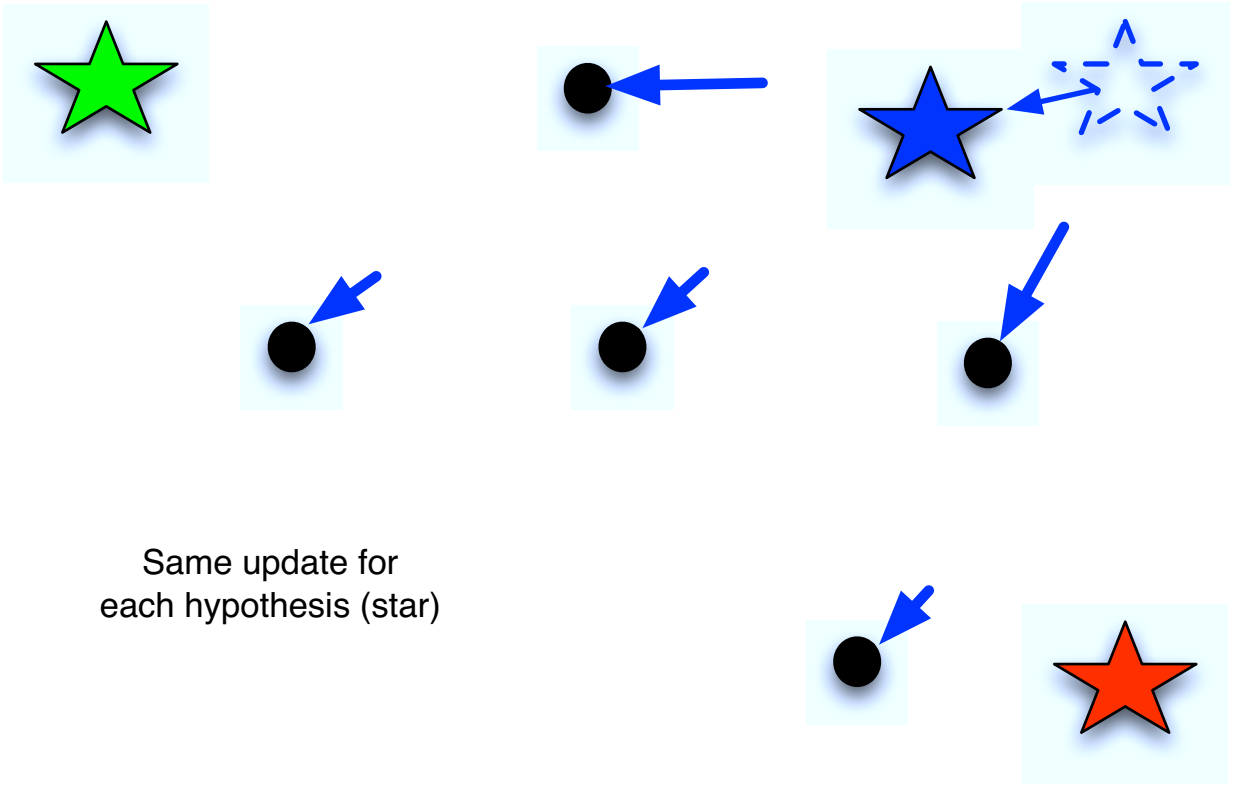
- Initialize: Assign random values to the parameters
- E-step: For each data point, compute the probability that it comes from each distribution.
- M-step: Update parameters based on probability weightings of data points.

E Step



- The hypothesized distributions are **competing** for the data points.
- Each hypothesis gives membership weights to the data points.
- Every point's weights (3 in this case) are normalized.
- So weights = the **relative likelihood** of distribution/class membership.

M Step



- Each hypothesis (distribution mean and variance) is updated based on:
  - The locations of the data points.
  - Their weighting toward that particular hypothesis.
- Hypotheses are *pulled* toward the data points with varying force.

EM Generality

EM is widely applicable in situations where there are many data points and a model with one or more hidden factors.

Typical Tasks:

- Diagnosis
  - Data = patient records of symptoms
  - Params = Prior probs for the values of a hidden factor (HF) + P(symptom I HF = v) $\forall$ symptoms and v.
- Classification
  - Data = unclassified instances (list of attributes, but without class/answer).
  - Params = Prior probs of classes (C_i) + P(attribute I C_i) $\forall$ attributes and C_i.
- Distribution Discovery
  - Data = points originating from any one of the distributions.
  - Params = mean and variance of each distribution.

Summary

Full Bayesian learning gives best possible predictions but is intractable

MAP learning balances complexity with accuracy on training data

Maximum likelihood assumes uniform prior, OK for large data sets

ML for continuous spaces using gradient (etc.) of log likelihood

Regression with Gaussian noise $\rightarrow$ minimize sum-of-squared errors