

Overview of Pattern Recognition Methods for Speech & Speaker Recognition

Professor Marie Roch

Huang: 3.4, 4.3, 4.4.2-3, 4.5

Discriminative Training

- When training classifiers, we must have an objective function which we are trying to maximize.
- With maximum likelihood estimators (MLE), we attempt to maximize the parameters with respect to specific data.

Overview of discriminative training

- Several techniques attempt to maximize the ability of models to discriminate amongst each other.
- Examples of discriminative training techniques (others exist)
 - Minimum-error-rate estimation
 - Neural networks
 - Maximum mutual information

Minimum-error-rate classification

- Also known as minimum classification error (MCE)
- A misclassification measure is defined:

$$e_i(x) = -d_i(x, \phi) + \left[\frac{1}{s-1} \sum_{j \neq i} d_j(x, \phi)^\eta \right]^{\frac{1}{\eta}}$$

– s = number of classes, $\eta > 0$

Minimum-error-rate estimation

$$e_i(x) = -d_i(x, \phi) + \left[\frac{1}{s-1} \sum_{j \neq i} d_j(x, \phi)^\eta \right]^{\frac{1}{\eta}}$$

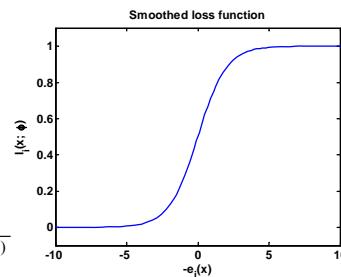
- Given x in ω_i , $e_i(x) > 0$ implies x will be misclassified.
- As $\eta \rightarrow \infty$, the largest d_j plays a more important role and the bracketed expression approaches a maximum function.

Minimum-error-rate classification

- Optimization can only be done on smooth loss functions

- Smooth with a sigmoid function:

$$l_i(x; \phi) = \text{sigmoid}(e_i(x)) = \frac{1}{1 + e^{-e_i(x)}}$$



- Large misclassification measure \rightarrow loss 1

Minimum-error-rate estimation

- The loss function can be minimized, but this is computationally intensive.

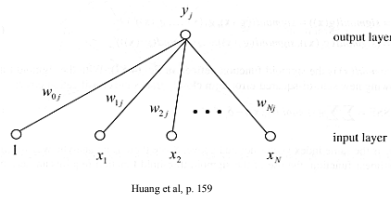
Neural networks

- Frequently referred to as connectionist models, neural networks are loosely based upon neurons which occur in the brain.
- Computational units of neural networks are perceptrons.

The perceptron

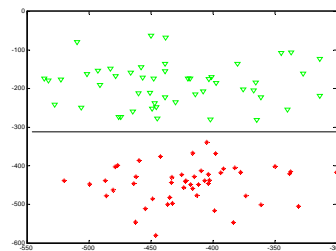
- A single layer perceptron is the simplest form of neural network:

$$y_j = w_{0j} + \sum_{i=1}^N w_{ij} x_i$$



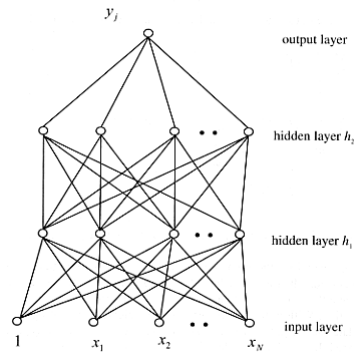
The perceptron

- The perceptron training algorithm (not covered) can be used to determine the weights w .
- A single perceptron is capable of separating classes that can be *linearly separated*.



Multilayer Perceptron

- By creating networks of perceptrons, it is possible to build classifiers which can partition regions which are more complicated.



Huang et al., p. 161

More unsupervised methods

- Expectation-Maximization (EM) algorithm
- Gaussian mixtures

The EM algorithm

- Suppose that we wish to maximize a parameter set ϕ given $Y=y$, but ϕ also depends upon random variable X .
- That is, if we had $X=x$, we could select ϕ to maximize:

$$P(X = x, Y = y | \phi)$$

The EM Algorithm

- $X=x$ is unavailable, and we will refer to it as hidden.
- Outline of the EM algorithm:
 1. Use an initial estimate of ϕ to determine $E[X]$ taking into account $Y=y$.
 2. Use $Y=y$ and $E[X]$ to determine a new ϕ .
 3. If converged, stop, otherwise goto 1.
- Convergence is guaranteed

EM: more formally...

- Goal: Find ϕ' which maximizes observed data

$$P(Y = y | \phi')$$

- But distribution dependent upon both X & Y

$$\log \Pr(x, y | \phi') = \log \Pr(x | y, \phi') + \log \Pr(y | \phi') \quad \text{Bayes rule}$$

(Note: We drop the $X=x, Y=y$ notation, but it is still implied.)

- Which we can rewrite to

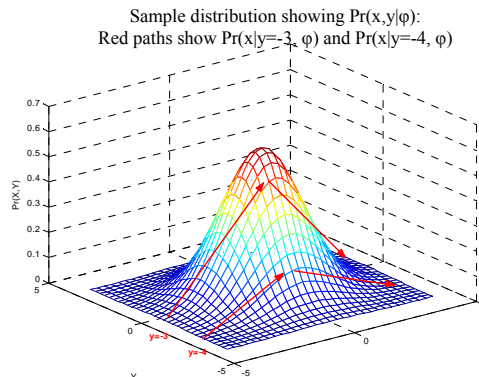
$$\log \Pr(y | \phi') = \log \Pr(x, y | \phi') - \log \Pr(x | y, \phi')$$

EM step 1: Find $E[X]$ given $Y=y$

- Suppose current parameter estimate ϕ .
- Let $E_{\phi}[f]_{X|Y=y}$ denote the expected value of function f over X with respect to:
 - distribution with parameter ϕ
 - a specific value of y .

EM Step 1: Conditional expectation

Hold at a specific y .
Find expected value
of x



$$E_{\phi}[f]_{X|Y=y} = \sum_x f(x, y) \Pr(x | y, \phi)$$

Note: Although the Gaussian is a continuous distribution, we are using the discrete notation of Huang et al. to maintain consistency with the derivation in the text. The EM algorithm is applicable to Gaussian distributions.

EM Step 1: Conditional expectation

- Recall

$$\log \Pr(y | \phi') = \log \Pr(x, y | \phi') - \log \Pr(x | y, \phi')$$

- therefore

$$E_{\phi}[\log \Pr(y | \phi')]_{X|Y=y} = E_{\phi}[\log(\Pr(X, y | \phi') - \log \Pr(X | y, \phi'))]_{X|Y=y}$$

EM Step 1

- Since

$$\begin{aligned} E_{\phi}[\log \Pr(y | \phi')]_{X|Y=y} &= \sum_x \Pr(x | y, \phi) \log(\Pr(y | \phi')) \\ &= \log \Pr(y | \phi') \end{aligned}$$

we can rewrite

$$E_{\phi}[\log \Pr(y | \phi')]_{X|Y=y} = E_{\phi}[\log(\Pr(X, y | \phi')) - \log \Pr(X | y, \phi')]_{X|Y=y}$$

to

$$\log \Pr(y | \phi') = E_{\phi}[\log(\Pr(X, y | \phi')) - \log \Pr(X | y, \phi')]_{X|Y=y}$$

EM Step 1

$$\log \Pr(y | \phi') = E_{\phi}[\log(\Pr(X, y | \phi')) - \log \Pr(X | y, \phi')]_{X|Y=y}$$

$$= \underbrace{E_{\phi}[\log(\Pr(X, y | \phi'))]_{X|Y=y}}_{\triangleq Q(\phi, \phi')} - \underbrace{E_{\phi}[\log \Pr(X | y, \phi')]_{X|Y=y}}_{\triangleq H(\phi, \phi')}$$

$$= Q(\phi, \phi') - H(\phi, \phi')$$

EM Step 1

$$\begin{aligned} Q(\phi, \phi') &= E_{\phi} [\log(\Pr(X, y | \phi'))]_{X|Y=y} \\ &= \sum_x ((\log \Pr(x, y | \phi')) \Pr(x | y, \phi)) \end{aligned}$$

$$\begin{aligned} H(\phi, \phi') &= E_{\phi} [\log(\Pr(X | y, \phi'))]_{X|Y=y} \\ &= \sum_x ((\log \Pr(x | y, \phi')) \Pr(x | y, \phi)) \end{aligned}$$

note: p171 eq. 4.94 second line $H(\phi, \phi')$ should read $P(X=x|Y=y, \phi)$, not $P(X=x|Y=y, \phi')$

EM Step 2: Maximization

- What value of ϕ' will increase $\Pr(y | \phi')$ as compared to our current estimate $\Pr(y | \phi)$?

$$\log \Pr(y | \phi') \geq \log \Pr(y | \phi)$$

$$\rightarrow Q(\phi, \phi') - H(\phi, \phi') \geq Q(\phi, \phi) - H(\phi, \phi)$$

$$\rightarrow Q(\phi, \phi') - Q(\phi, \phi) - H(\phi, \phi') + H(\phi, \phi) \geq 0$$

EM Step 2: Maximize Q Function

- Suppose we can select a ϕ' such that

$$Q(\phi, \phi') \geq Q(\phi, \phi)$$

- Clearly possible
 - Equality at $Q(\phi, \phi' = \phi) = Q(\phi, \phi)$
 - Other values may produce $Q(\phi, \phi') \geq Q(\phi, \phi)$
 - How we maximize this depends upon the application

EM Step 2: What happens to H?

- Remember, likelihood nondecreasing if

$$Q(\phi, \phi') - Q(\phi, \phi) - H(\phi, \phi') + H(\phi, \phi) \geq 0$$

- We know that we can pick ϕ' such that the contribution by Q is ≥ 0 . What about H?

EM Step 2: H function

- For any ϕ' , $H(\phi, \phi) - H(\phi, \phi') \geq 0$

$$\begin{aligned} & H(\phi, \phi) - H(\phi, \phi') \\ &= \sum_x ((\log \Pr(x|y, \phi)) \Pr(x|y, \phi)) - \sum_x ((\log \Pr(x|y, \phi')) \Pr(x|y, \phi)) \\ &= \sum_x ((\log \Pr(x|y, \phi) - \log \Pr(x|y, \phi')) \Pr(x|y, \phi)) \\ &= \sum_x \left(\left(\log \frac{\Pr(x|y, \phi)}{\Pr(x|y, \phi')} \right) \Pr(x|y, \phi) \right) \\ &= E_\phi \left[\log \frac{\Pr(x|y, \phi)}{\Pr(x|y, \phi')} \right]_{x|Y=y} = E_\phi \left[-\log \frac{\Pr(x|y, \phi')}{\Pr(x|y, \phi)} \right]_{x|Y=y} \end{aligned}$$

EM Step 2: Jensen's \neq

- Jensen has shown that for any concave function f :

$$E[f(x)] \leq f(E[x])$$

- and that for any convex function f :

$$E[f(x)] \geq f(E[x])$$

EM Step 2: H Function

- By Jensen's \neq ($-\log$ is a convex function)

$$E_{\phi} \left[-\log \frac{\Pr(x | y, \phi')}{\Pr(x | y, \phi)} \right]_{x|y=y} \geq -\log E_{\phi} \left[\frac{\Pr(x | y, \phi')}{\Pr(x | y, \phi)} \right]_{x|y=y}$$

- We will obtain a lower bound

$$-\log E_{\phi} \left[\frac{\Pr(x | y, \phi')}{\Pr(x | y, \phi)} \right]_{x|y=y} = 0$$

which implies that $H(\phi, \phi) - H(\phi, \phi') \geq 0$

EM Step 2: H Function

- Consider our lower bound:

$$\begin{aligned} &= -\log E_{\phi} \left[\frac{\Pr(x | y, \phi')}{\Pr(x | y, \phi)} \right]_{x|y=y} \\ &= -\log \sum_x \left(\frac{\Pr(x | y, \phi')}{\Pr(x | y, \phi)} \Pr(x | y, \phi) \right) \quad \text{by definition of E} \\ &= -\log \sum_x \Pr(x | y, \phi') \quad \text{by cancellation} \\ &= -\log 1 = 0 \end{aligned}$$

EM Steps 2 and 3

- So any ϕ' we pick which increases the Q function will increase or leave at the same likelihood:

$$\log \Pr(y | \phi') = E_{\phi} [\log(\Pr(X, y | \phi') - \log \Pr(X | y, \phi'))]_{X|Y=y}$$

- Step 3: Let $\phi = \phi'$. Goto step 1 unless convergence has been reached.

Notes on the EM

- Maximizing the Q function (selecting ϕ') depends upon the specific problem.
- Each iteration's maximization step produces an estimate which is *at least as good* as the previous one if not better.
- No proof on rate of convergence, but on the type of problems we will consider, a small number of iterations suffice.

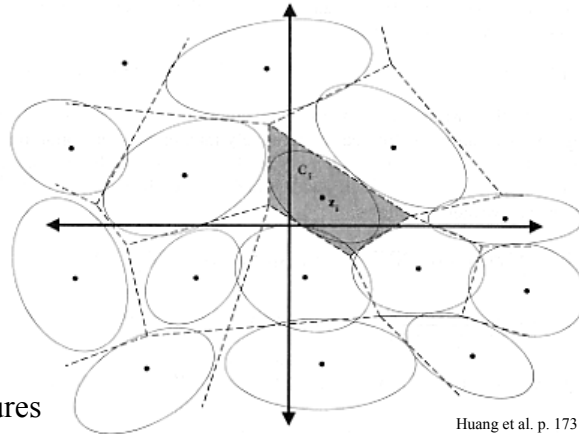
EM: Sounds familiar?

- K-Means algorithm is an approximate EM algorithm
 - The partitioning is unavailable
 - Expectation:
 - Partition training vectors by finding minimum distortion codeword.
 - Maximization:
 - Recompute centroids based upon partitioning.
 - Note: Although we min. distortion, one could think of this as maximizing the inverse of distortion.

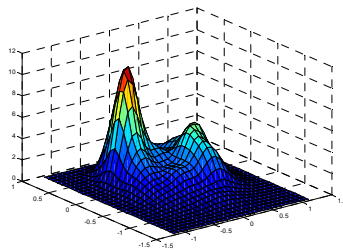
Gaussian Mixture Models (GMMs)

- Similar to vector codebooks.
- N normal distributions chosen to represent data.
 - Means: Similar to the codewords
 - $\Pr(x | \mu, \Sigma)$ used instead of distortion (more on this later)

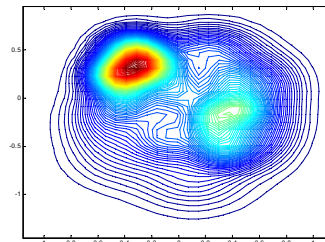
Partitioning induced by a GMM



Sample 16 mixture model Based upon R^2 cepstral speech data



Surface plot of pdfs



Equal likelihood contour lines

GMM: Parameters

- Parameters for each of K mixtures $i=1\dots K$:
 - μ_i mean
 - Σ_i variance-covariance matrix
 - c_i mixture weight where $\sum_{k=1}^K c_k = 1$
- Use Φ_i to denote (c_i, μ_i, Σ_i) and Φ to denote the entire set of parameters (Note: Authors do not include c_i in Φ_i)

GMM: Evaluation probability

- Probability of an observation:

$$\begin{aligned}\Pr(\bar{x} | \Phi) &= \sum_{k=1}^K c_k N_k(\bar{x} | \mu_k, \Sigma_k) \\ &= \sum_{k=1}^K c_k \frac{1}{(2\pi)^{d/2} |\Sigma_k|^{1/2}} e^{\left(\frac{-1}{2}(\bar{x}-\mu_k)^T \Sigma_k^{-1} (\bar{x}-\mu_k)\right)}\end{aligned}$$

GMM: Mixture weights

- Interpretation of mixture weights c_i
 - Prior probability that observation x comes from mixture i .
 - ... or ...
 - Scaling of mixtures such that all mixtures together form a pdf.

GMM: Estimation

- Application of the EM algorithm
- Expectation step
 - Determine how well each mixture models each observation
 - Determine probability of observation with respect to mixture in question.
 - Divide by probability of seeing the observation (sum across mixtures).

GMM: Expectation step

- How well does mixture k model y_i ?

$$\begin{aligned}\gamma_k^i &= \frac{\Pr(y_i \mid \text{mixture } k)}{\Pr(y_i \mid \text{all mixtures})} \\ &= \frac{\Pr(y_i \mid \Phi_k)}{\Pr(y_i \mid \Phi)} \\ &= \frac{c_k \Pr(y_i \mid \mu_k, \Sigma_k)}{\sum_{j=1}^K c_j \Pr(y_i \mid \mu_j, \Sigma_j)}\end{aligned}$$

GMM: Expectation step

- Also need to determine how well each mixture represents the training data
 - Sum the γ 's for each mixture over all observations.
 - This is not a probability. Used as a normalizing factor in the maximization step.

$$\gamma_k = \sum_{i=1}^N \gamma_k^i$$

GMM: Maximization step

- $\hat{c}_k = \frac{\text{How well mixture } k \text{ represents training data}}{\text{How well all mixtures represent training data}}$
$$= \frac{\gamma_k}{\sum_{j=1}^K \gamma_j}$$
$$= \frac{\gamma_k}{N}$$

GMM: Maximization step

- Why does $\sum_{j=1}^K \gamma_j = N$?
 - γ_k represents contribution of k^{th} mixture to probability measured for each observation.
 - The total contribution to the probability for a single observation must be one. (all probability must be accounted for).
 - As there are N observations, the sum is N .

GMM: Maximization step

- $$\begin{aligned} \hat{\mu}_k &= \frac{\sum_{i=1}^N (\text{Contribution mixture } k) y_i}{\text{Contribution all mixtures}} \\ &= \frac{\sum_{i=1}^N \gamma_k^i y_i}{\sum_{i=1}^N \gamma_k^i} \\ &= \frac{\sum_{i=1}^N \frac{c_k \Pr(y_i | \mu_k, \Sigma_k)}{\Pr(y_i | \Phi)} y_i}{\sum_{i=1}^N \frac{c_k \Pr(y_i | \mu_k, \Sigma_k)}{\Pr(y_i | \Phi)}} \end{aligned}$$

GMM: Maximization step

- $$\begin{aligned} \hat{\Sigma}_k &= \frac{\sum_{i=1}^N (\text{Contribution mixture } k) y_i \text{'s contribution to cov}}{\text{Contribution all mixtures}} \\ &= \frac{\sum_{i=1}^N \gamma_k^i (y_i - \mu_k)(y_i - \mu_k)'}{\sum_{i=1}^N \gamma_k^i} \\ &= \frac{\sum_{i=1}^N \frac{c_k \Pr(y_i | \mu_k, \Sigma_k)}{\Pr(y_i | \Phi)} (y_i - \mu_k)(y_i - \mu_k)'}{\sum_{i=1}^N \frac{c_k \Pr(y_i | \mu_k, \Sigma_k)}{\Pr(y_i | \Phi)}} \end{aligned}$$

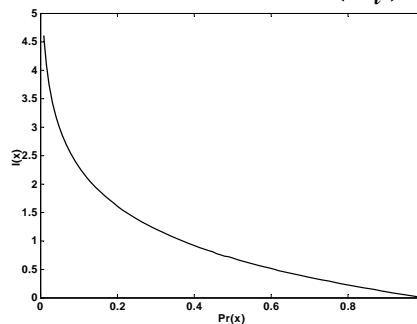
Information theory

- The next model depends on having a measure which indicates the usefulness of any given partitioning of data.
- We turn to information theory to provide such a measure.

Quantity of information

- One interpretation of the quantity of information is the amount of surprise that one sees when observing an event.
- If an event is rare, we can derive a large quantity of information from it.

$$I(x_i) = \log \frac{1}{\text{Pr}(x_i)}$$



Quantity of information

- Why use log?
 - Suppose we want to know the information in two independent events:

$$\begin{aligned} I(x_1, x_2) &= \log \frac{1}{\Pr(x_1, x_2)} \\ &= \log \frac{1}{\Pr(x_1) \Pr(x_2)} && x_1, x_2 \text{ independent} \\ &= \log \frac{1}{\Pr(x_1)} + \log \frac{1}{\Pr(x_2)} \\ &= I(x_1) + I(x_2) \end{aligned}$$

Quantity of information

- The logarithm permits us to determine the information of independent events by addition.
- When the logarithm is base 2, we call the unit of information a *bit*.
- Let us assume that all logarithms are base 2.

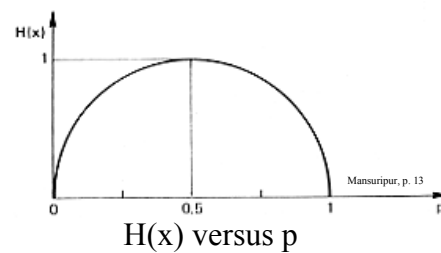
Entropy

- Entropy is defined as the expected amount of information (average amount of surprise) and is usually denoted by the symbol H .

$$\begin{aligned} H(X) &= E[I(X)] \\ &= \sum_{x_i \in S} \Pr(x_i) I(x_i) && S \text{ is all possible symbols} \\ &= \sum_{x_i \in S} \Pr(x_i) \log \frac{1}{\Pr(x_i)} && \text{definition } I(x_i) \\ &= E[-\log \Pr(X)] \end{aligned}$$

Example

- Assume
 - $X = \{0, 1\}$
 - $\Pr(X) = \begin{cases} p & X = 0 \\ 1-p & X = 1 \end{cases}$



- Then
 - $H(X) = E[I(X)]$
 - $= -p \log p - (1-p) \log(1-p)$

Conditional entropy

- Suppose we want to know the entropy of X given evidence Y .
- Recall that before observing Y

$$H(X) = \sum_x P(X = x_i) \log \frac{1}{P(X = x_i)}$$

- Once a specific y_j has been observed:

$$\begin{aligned} H(X | Y = y_j) &= \sum_x P(X = x_i | Y = y_j) \log \frac{1}{P(X = x_i | Y = y_j)} \\ &= - \sum_x P(x_i | y_j) \log P(x_i | y_j) \end{aligned}$$

Conditional entropy

- Taking the expected value of $H(X|Y=y_j)$

$$\begin{aligned} E_Y[H(X | Y = y_j)] &= \sum_Y H(X | Y = y_j) P(Y = y_j) \\ &= - \sum_Y P(y_j) \sum_X P(x_i | y_j) \log P(x_i | y_j) \quad \text{defn. } H(X | Y = y_j) \\ &= - \sum_Y \sum_X P(x_i, y_j) \log P(x_i | y_j) \quad \text{Bayes rule} \end{aligned}$$

- Tells us the average surprise in seeing X after having observed some instance of Y

Implications of conditional entropy

- Consider the joint entropy of X and Y

$$\begin{aligned} H(X, Y) &= -\sum_x \sum_y P(x_i, y_j) \log P(x_i, y_j) \quad \text{defn. } H(X, Y) \\ &= -\sum_x \sum_y P(x_i, y_j) (\log P(x_i) + \log P(y_j | x_i)) \quad \text{Bayes rule} \\ &= -\sum_x \sum_y P(x_i, y_j) \log P(x_i) - \sum_x \sum_y P(x_i, y_j) \log P(y_j | x_i) \\ &= -\sum_x P(x_i) \log P(x_i) - \sum_x \sum_y P(x_i, y_j) \log P(y_j | x_i) \quad \text{marginal} \\ &= H(X) + H(Y | X) \end{aligned}$$

Conditional entropy

- One way of considering $H(X|Y)$ is to think of X as representing a class.
- If $H(X|Y)=0$, knowing Y will always permit us to know the class.

Mutual information

- $I(X;Y) = H(X) - H(X|Y)$
 - Indicates how much information is gained by adding a variable.
 - Symmetric: $I(X;Y)=I(Y;X)$
 - If X & Y are independent, we learn nothing more: $I(X;Y) = 0$
 - $0 \leq I(X;Y) \leq \min(H(X), H(Y))$

Maximum mutual information estimation (MMIE)

- Posterior $P(\omega_i | x) = \frac{P(x | \omega_i)P(\omega_i)}{P(x)}$
- The MAP decision rule ignores the contribution of $P(x)$ as it does not effect the maximum.
- Typically, when training, we only estimate the likelihood function $P(x | \omega_i)$

MMIE

- One might argue that maximizing the posterior is a better method.
- Basic idea: maximize

$$P(\omega_i | x) = \frac{P(x | \omega_i)P(\omega_i)}{P(x | \omega_i)P(\omega_i) + \sum_{k \neq i} P(x | \omega_k)P(\omega_k)}$$

which maximizes contribution of ω_i and minimizes contributions of ω_k where $k \neq i$.

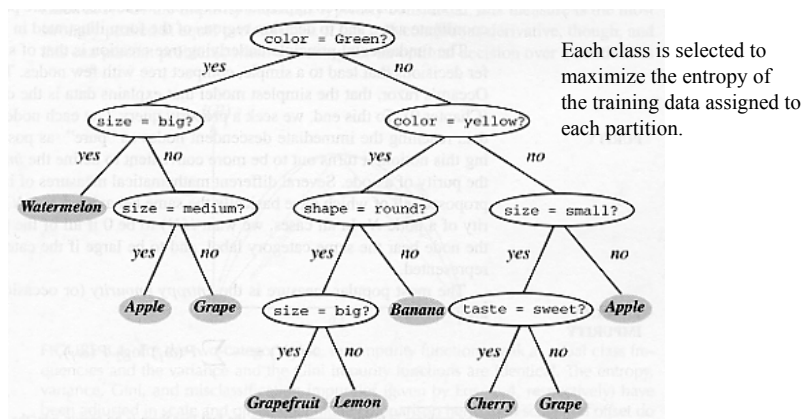
MMIE

- When the prior is considered uniform, this can be reformulated as a maximum mutual information problem.
- Gradient descent can be used to estimate the models.

Overview of CART: Classification and regression trees

- Partitions set by asking a series of questions.
- A bit like the game 20 questions
 - i.e. Are you thinking of a politician?
 - Answer allows us to prune the space.

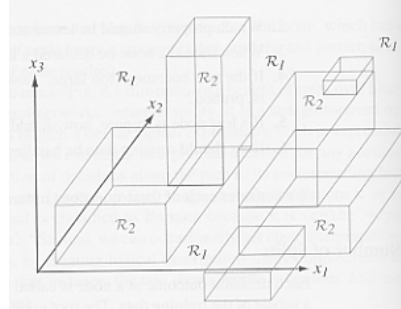
CART: Sample binary tree for classification of fruit



Duda, Hart, & Stork p. 397

CART

- CART can only partition the space into rectangular regions.
- Each question in the tree creates a new boundary.



Duda, Hart, & Stork p. 398

CART

- CART is a supervised learner.
- Trees, one must determine what question to ask at each node.
- Choice of question based upon information theoretic criteria.

CART

- It is always possible to construct a tree which will classify correctly all of its training data.
- This can lead to overtraining:
 - CART tree represents the training data too well
 - Tree fails to generalize well.
- Pruning removes the weakest subtrees as measured by a cost-complexity measure.

HMMs

- Hidden Markov models (HMMs) permit the modeling of temporal sequences of feature vectors.
- We will cover these in *detail* later in the semester.
- HMM-based recognition is the current state-of-the-art for acoustic modeling.