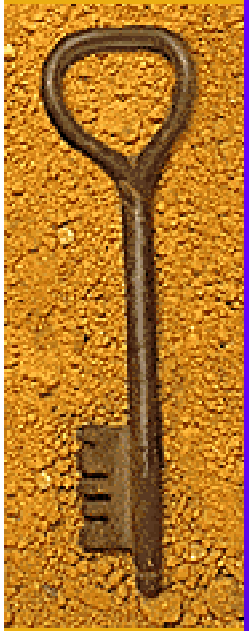
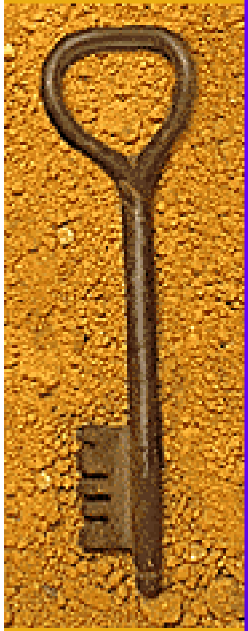


Statistical methods in recognition

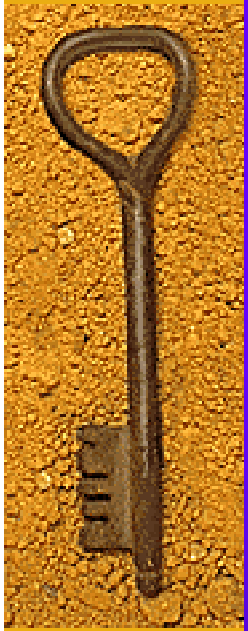
- Basic steps in classifier design
  - Collect training data
  - Choose a classification model
    - **Statistical**
    - Linguistic
  - Estimate “parameters” of classification model from training images
    - Learning
  - Evaluate model on training data and refine
  - Collect test data set
  - Apply classifier to test data



Why is classification a problem?

- Because classes overlap in our (impoverished) representations
- Example: Classify a person as a male or female based on weight
  - Male training set :{ 155, 122, 135, 160, 240, 220, 180, 145}
  - Female training set: {95, 132, 115, 124, 145, 110, 150}
  - Unknown sample has weight 125. Male or female?

Factors that should influence our decision



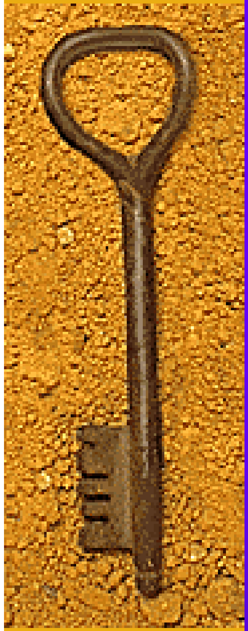
- ♦ How likely is it that a person weighs 125 pounds given that the person is a male? Is a female?
  - – **Class-conditional probabilities**

■How likely is it that an arbitrary person is a male? A female?

–**Prior class probabilities**

■What are the costs of calling a male a female? A female a male?

–**Risks**



Basic approaches to statistical classification

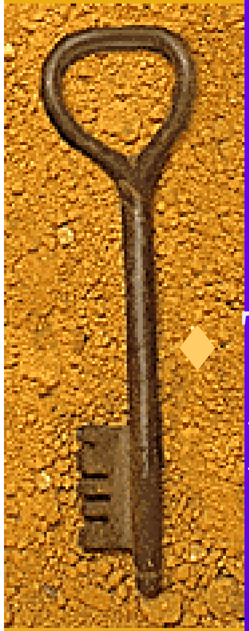
1. Build (parametric) probabilistic models of our training data, and compute the probability that an unknown sample belongs to each of our possible classes using these models.
2. Compare an unknown sample directly to each member of the training set, looking for the training element “most similar” to the unknown.
   - Nearest neighbor classification
3. Train a neural network to recognize unknown samples by “teaching it” how to correctly train the elements of the training set.

A primer on probability

- Probability spaces - models of random phenomena
- Example: a box contains s balls labeled 1, ..., s
  - Experiment: Pick a ball, note its label and then replace it in the box. Repeat this experiment n times.
  - Let $N_n(k)$ be the number of times that a ball labeled k was chosen in an experiment of length n
  - example: s = 3, n = 20

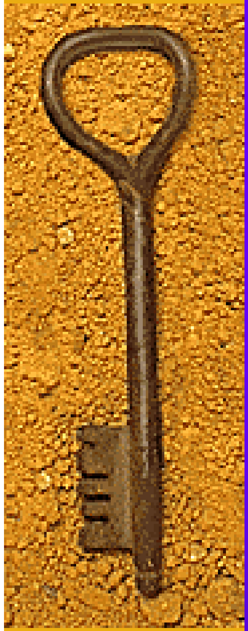
  1 1 3 2 1 2 2 3 2 3 3 2 1 2 3 3 1 3 2 2

  - $N_{20}(1) = 5$ $N_{20}(2) = 8$ $N_{20}(3) = 7$



Primer on probability

- The relative frequencies of the outcomes 1,2,3 are
  - $N_{20}(1)/20 = .25 \quad N_{20}(2)/20 = .40 \quad N_{20}(3)/20 = .35$
  - As n gets large, these numbers should settle down to fixed numbers p_1, p_2, p_3
  - We say p_i is the probability that the i'th ball will be chosen when the experiment is performed once



Primer on probability

- Suppose: we color balls 1, ..., r red and balls r+1, .., s green
  - What is the probability of choosing a red ball?
  - Intuitively it is r/s = $\Sigma\, p_k$ where the sum is over all ω_k such that the k'th ball is red
- Let A be the subset of possible outcomes, ω_k , such that k is red.
  - A has r points
  - A is called an event
  - When we say that A has occurred we mean that an experiment has been run and the outcome is represented by a point in A.
- If A and B are events, then so are $A \cap B$, $A \cup B$ and A^c

Primer on probability

- Assigning probabilities to events:

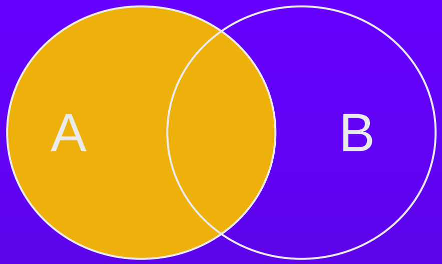
$$P(B) = \sum_{w_k \in B} p_k$$

- A probability measure on a set Ω of possible outcomes is a real valued function having domain 2^{Ω} satisfying
  - – $P(\Omega) = 1$
  - – $0 <= P(A) <= 1$, for all $A \subset \Omega$
  - – If A_n are mutually disjoint sets then

$$P(\bigcup_{n=1}^{k} A_n) = \sum_{n=1}^{k} P(A_n)$$

Primer on probability

- Simple properties of probabilities
  - $P(A^c) = 1 - P(A)$
    - $P(\emptyset) = 1 - P(\Omega) = 1 - 1 = 0$
    - if A is a subset of B, then $P(A) <= P(B)$
    - $P(A \cup B) = P(A) + P(B) - P(A \cap B)$



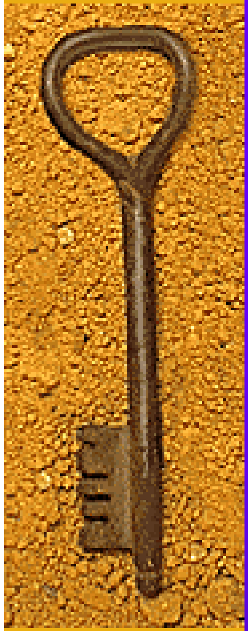
- Conditional probabilities
  - Our box has r red balls labeled 1, ..., r and b black balls labeled r+1, ..., r+b. If the ball drawn is known to be red, what is the probability that its label is 1?
    - A - event “red”
    - B - event “1”
    - interested in conditional probability of B knowing that A has occurred - $P(B|A)$

Primer on probability

- Let A and B be two events such that P(A) > 0. Then the conditional probability that B occurs given A, written P(B|A) is defined to be

$$P(B|A) = \frac{P(B \cap A)}{P(A)}$$

- Ball example: what is P(“1”| “red”)
  - Let r = 5 and b = 15
  - P(1 and red) = .05
  - P(red) = .25
  - So, P(1 | red) = .05/.25 = .20



Primer on probability

- Recognition
  - A_1, ..., A_n are mutually disjoint events with union Ω.
    - think of the A_i as the possible identities of an object
  - B is an event with $P(B) > 0$
    - think of B as an observable event, like the area of a component in an image
  - $P(B|A_k)$ and $P(A_k)$ are known, $k = 1,..., n$
    - $P(B|A_k)$ is the probability that we would observe a component with area B if the identify of the object is A_i
    - $P(A_k)$ is the prior probability that an event is in class k.
  - Question: What is $P(A_i|B)$
    - What we will really be after - the probability that the identity of the object is A_i given that we make measurements B

Primer on probability

$$P(B|A) = \frac{P(B \cap A)}{P(A)}$$

$$B = B \cap \left(\bigcup_{k=1}^{n} A_k \right) = \bigcup_{k=1}^{n} (B \cap A_k)$$

So intersections are disjoint since the A_k are and

$$P(B) = \sum_{k=1}^{n} P(B \cap A_k)$$

But

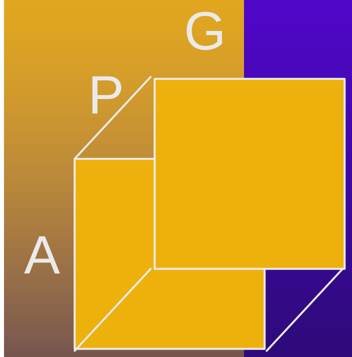
$$P(B \cap A_k) = P(A_k)P(B|A_k)$$

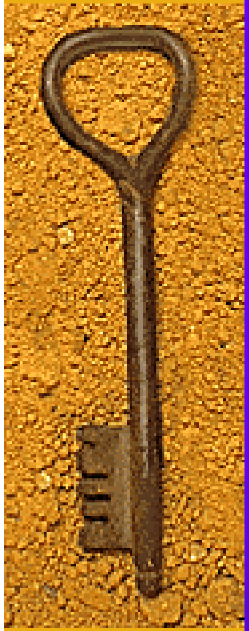
Combining all this we get Bayes Rule

$$P(A_i|B) = \frac{P(A_i \cap B)}{P(B)} = \frac{P(A_i)P(B|A_i)}{\sum_{k=1}^{n} P(A_k)P(B|A_k)}$$

Training - computing $P(B|A_i)$

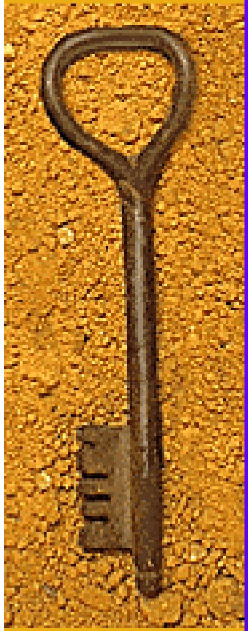
- Our training data is used to compute the $P(B|A_i)$, where B is the vector of features we plan to use to classify unknown images in the classes A_i
  - – B might be (area, perimeter, moments)
- How might we represent $P(B|A_i)$?
  - – as a table
    - quantize area, perimeter and average gray level suitably, and then use the training samples to fill in the three dimensional histogram.
    - analytically, by a standard probability density function such as the normal, uniform, ...





Primer on probability - training

- When we have many random variables it is usually impractical to create a table of the values of $P(B|A_i)$from our training set.
  - Example
    - 5 measurements
    - quantize each to 50 possible values
    - Then there are 50^5 possible 5-tuples we might observe in any element of the training set, and we would need to estimate this many probabilities to represent the conditional probability
      - too few training samples
      - too much storage required for the table



Primer on probability

- Instead, it is usually assumed that $P(B|A_i)$ has some simple mathematical form
  - uniform density function
    - each x_i takes on values only in the finite range $[a_i, b_i]$
    - $P(B|A_i)$ is constant for any realizable $(x_{1,} \ldots, x_n)$
    - for one random variable, $P(B|A_i) = 1/(b-a)$ for $a <= x <= b$ and 0 elsewhere
  - Normal distribution

$$f(x) = n(x;m,s) = \frac{1}{\sqrt{2p}\,s} e^{-(\frac{x-m}{s})^2}$$

  - In any case, once the parameters of the assumed density function are estimated, its goodness of fit should also be evaluated.

Primer on probability

- Density function is called the Gaussian function and the error function
  - μ is called the locatio parameter
  - σ is called the scale parameter
- Generalization to multivariate density functions
  - mean vector
  - covariance matrix

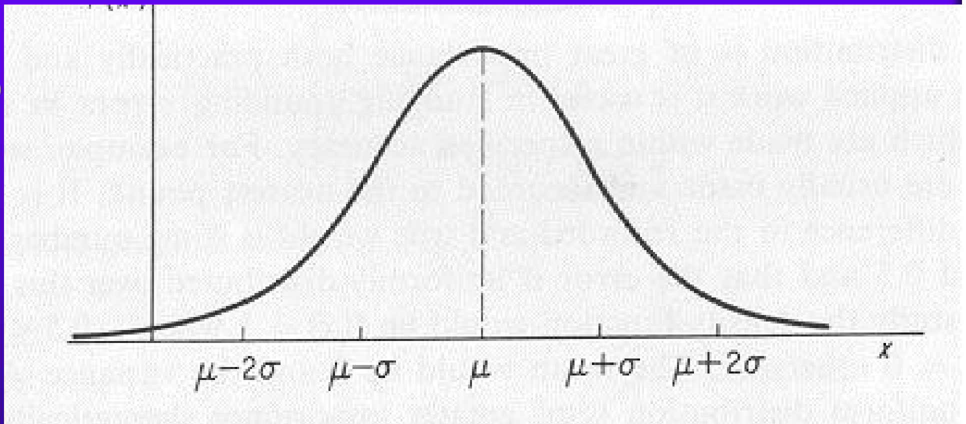
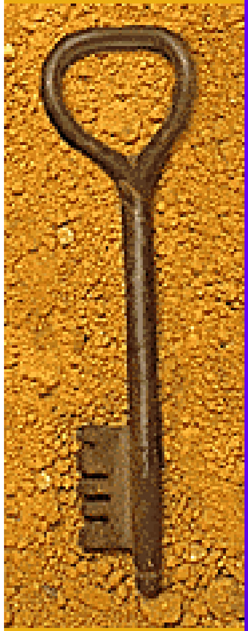


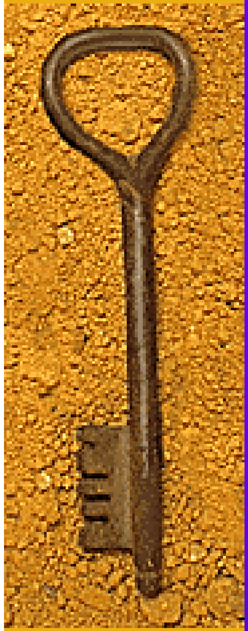
Fig. 3.7 Typical Normal Density Curve

Prior probabilities and their role in classification

- Prior probabilities of each object class
  - probabilities of the events: object is from class i ($P(A_i)$)
  - Example
    - two classes - A and B; two measurement outcomes: 0 and 1
    - prob(0|A) = .5, prob(1|A) = .5; prob(0|B) = .2 prob(1|B)=.8
  - Might guess that if we measure 0 we should decide that the class is A, but if we measure 1 we should decide B



- But suppose that P(A) = .10 and P(B) = .90
  - Out of 100 samples, 90 will be B’s and 18 of these (20% of those 90) will have measurement 0
    - We will classify these incorrectly as A’s
    - Total error is nP(B)P(0|B)
  - 10 of these samples will be A’s and 5 of them will have measurement 0 - these we’ll get right
    - Total correct is nP(A)P(0|A)



Prior probabilities

- So, how do we balance the effects of the prior probabilities and the class conditional probabilities?
- We want a rule that will make the fewest errors
  - Errors in A proportional to $P(A)P(x|A)$
  - Errors in B proportional to $P(B)P(x|B)$
  - To minimize the number of errors choose A if $P(A)P(x|A) > P(B)P(x|B)$; choose B otherwise
- The rule generalizes to many classes. Choose the C_i such that $P(C_i)P(x|C_i)$ is greatest.
- Of course, this is just Bayes’ rule again

Bayes error

- The formula for $P(C_i|x)$ is

$$P(C_i|x)=\frac{P(C_i)P(x|C_i)}{P(x)}$$

- where

$$P(x)=\sum_i P(C_i)P(x|C_i)$$

is a normalization factor that is the same for all classes.

- To evaluate the performance of our decision rule we can calculate the probability of error - probability that the sample is assigned to the wrong class.

Bayes error

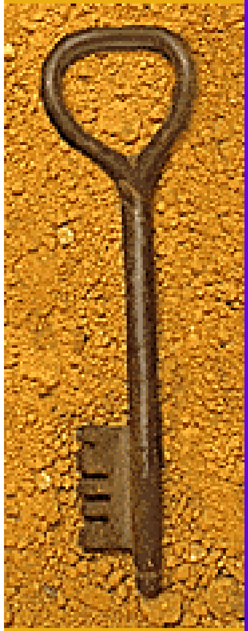
- The **total error** which is called the **Bayes error** is defined as E[r(x)] =

$$e = \int \min[P(C_1)P(x|C_1), P(C_2)P(x|C_2)]p(x)dx$$

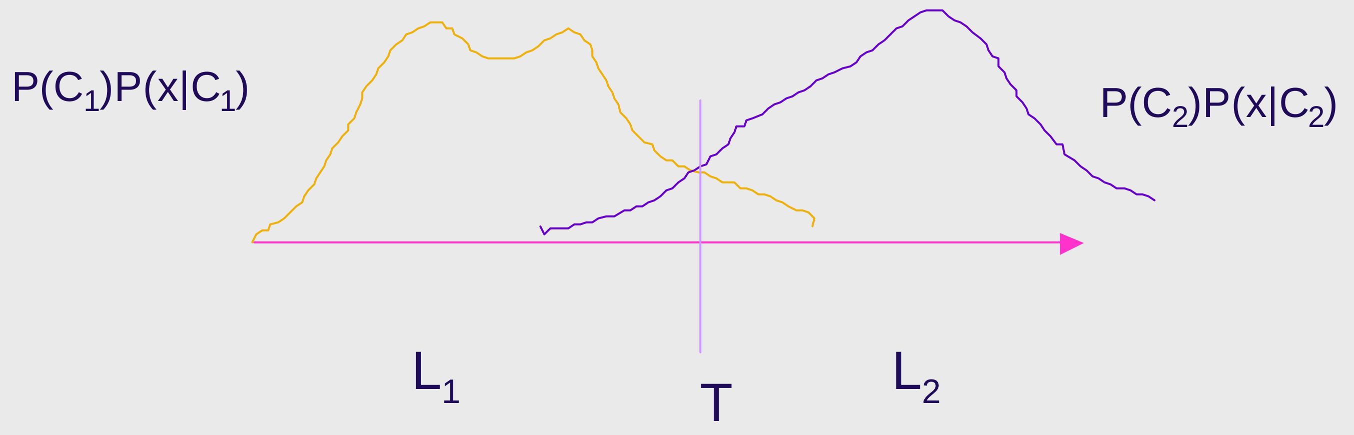
$$= P(C_1)\int_{L_1} P(C_1|x)dx + P(C_2)\int_{L_2} P(C_2|x)dx$$

$$= P(C_1)e_1 + P(C_2)e_2$$

- The regions L_1 and L_2 are the regions where x is classified as C_1 and C_2 respectively.



Example

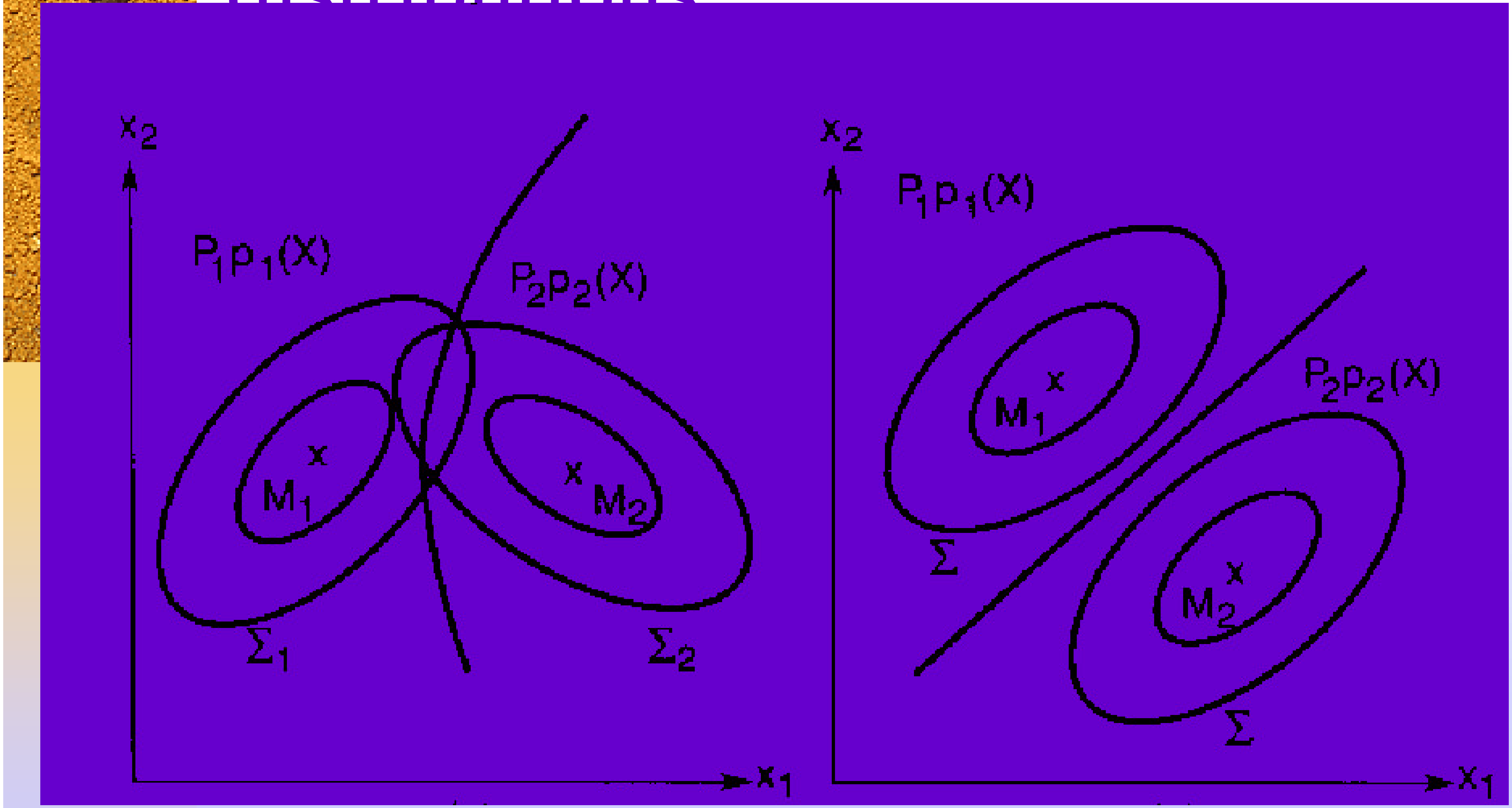


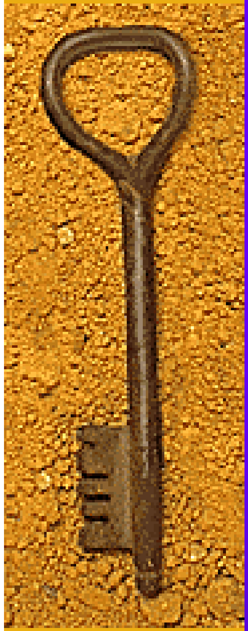
Moving T either left or right would increase the overall probability of error

Example - normal distributions

- In the case of normal distributions, the decision boundaries that provide the Bayes error can be shown to be quadratic functions - quadratic curves for two dimensional probability density functions
- In the special case where the classes have the same covariance matrix, decision boundary is a linear function - classes can be separated by a hyperplane

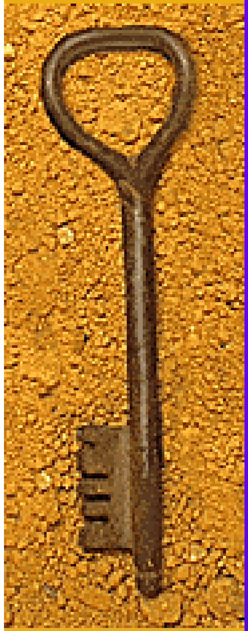
Bayes error for normal distributions





Adding risks

- Minimizing total number of errors does not take into account the cost of different types of errors
- Example: Screening X-rays for diagnosis
  - two classes - healthy and diseased
  - two types of errors
    - classifying a healthy patient as diseased - might lead to a human reviewing X-rays to verify computer classification
    - classifying diseased patient as healthy - might allow disease to progress to more threatening level
- Technically, including costs in the decision rule is accomplished by modifying the a priori probabilities



An example from image segmentation

- How do we know which groups of pixels in a digital image correspond to the objects or features to be analyzed?
  - In some simple cases, objects may be uniformly darker or brighter than the background against which they appear
    - Black characters imaged against the white background of a page
    - High gradient magnitude points tend to lie on edges

Image segmentation

- ♦ Ideally, object pixels would be black (0 intensity) and background pixels white (maximum intensity)
- ♦ But this rarely happens
  - – pixels overlap regions from both the object and the background, yielding intensities between pure black and white - edge blur
  - – cameras introduce “noise” during imaging - measurement “noise”

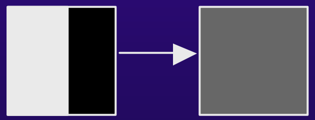
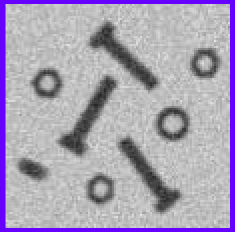
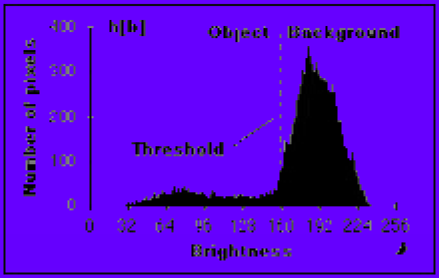


Image segmentation by thresholding

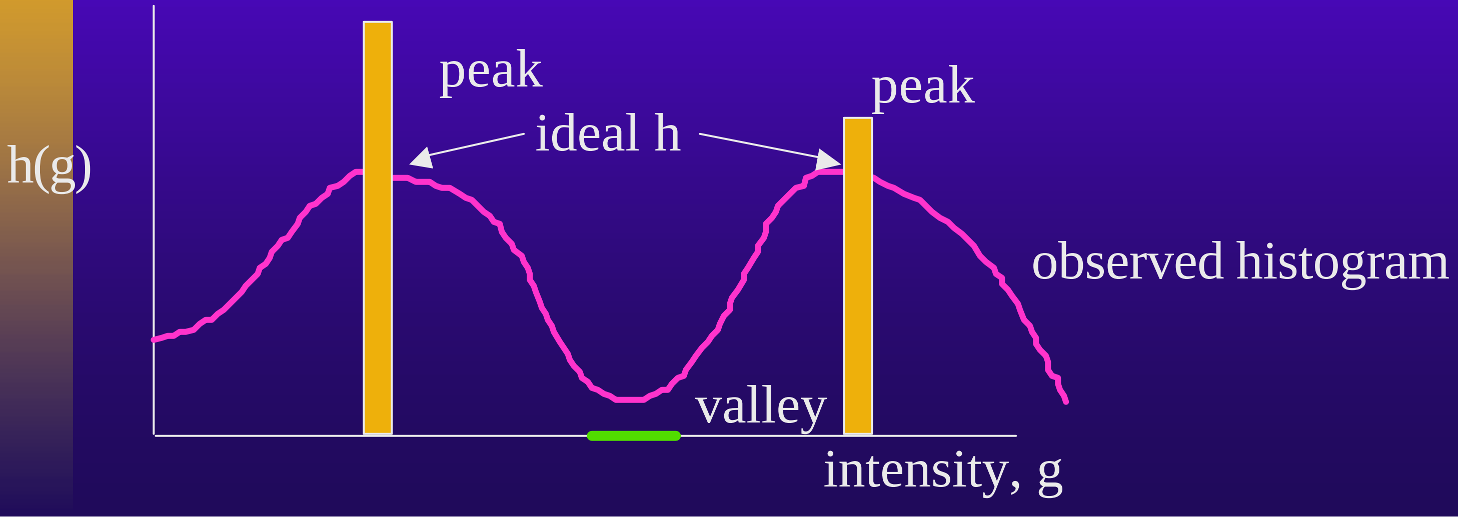
- If the objects and background occupy different ranges of gray levels, we can correctly “mark” the object pixels by a process called **thresholding:**
  - Let F(i,j) be the original, gray level image
  - B(i,j) is a binary image (pixels are either 0 or 1) created by thresholding F(i,j)
    - B(i,j) = 1 if F(i,j) < t
    - B(i,j) = 0 if F(i,j) >= t

Thresholding





- ♦ How do we choose the threshold t?
- ♦ Histogram (h) - gray level frequency distribution of the gray level image F.
  - – $h_F(g)$ = number of pixels in F whose gray level is g
  - – $H_F(g)$ = number of pixels in F whose gray level is <=g



h(g)

peak

ideal h

peak

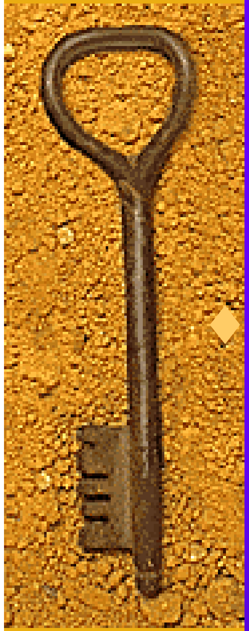
observed histogram

valley

intensity, g

Thresholding – a heuristic algorithm

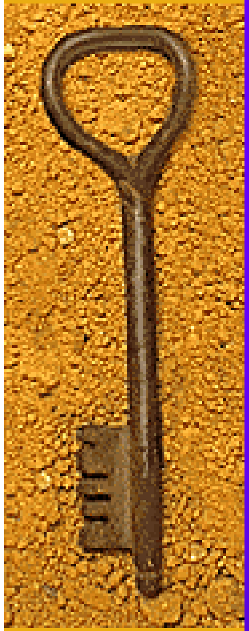
- Peak and valley method
  - Find the two most prominent peaks of h
    - g is a peak if $h_F(g) > h_F(g \pm \Delta g)$, $\Delta g = 1, ..., k$
  - Let g_1 and g_2 be the two highest peaks, with $g_1 < g_2$
  - Find the deepest valley, g, between g_1 and g_2
    - g is the valley if $h_F(g) <= h_F(g')$, g,g' in $[g_1, g_2]$
  - Use g as the threshold



A probabilistic threshold selection method - minimizing Kullback information distance

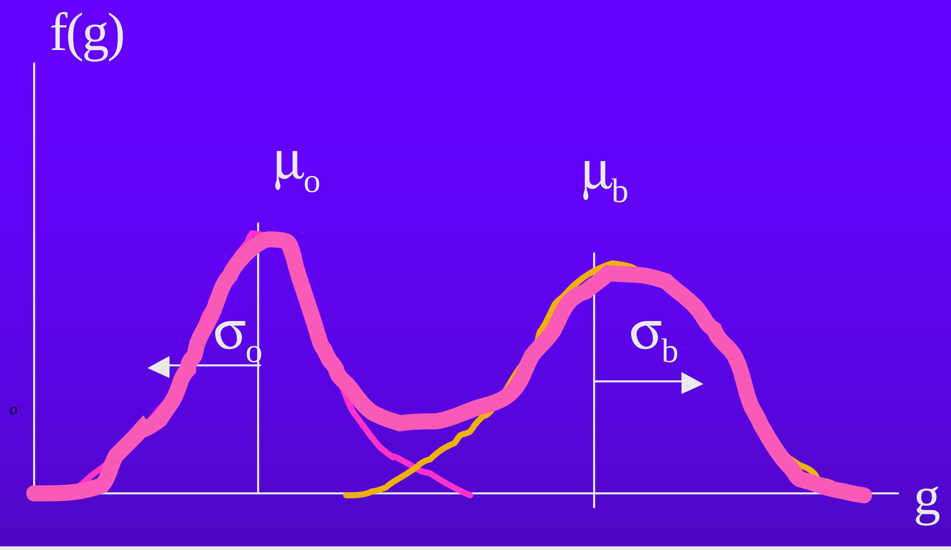
- The observed histogram, f, is a mixture of the gray levels of the pixels from the object(s) and the pixels from the background
  - in an ideal world the histogram would contain just two spikes
  - but
    - measurement noise,
    - model noise (e.g., variations in ink density within a character) and
    - edge blur (misalignment of object boundaries with pixel boundaries and optical imperfections of camera)

  spread these spikes out into hills



Kullback information distance

- Make a parametric model of the shapes of the component histograms of the objects(s) and background
- Parametric model - the component histograms are assumed to be Gaussian
  - p_o and p_b are the proportions of the image that comprise the objects and background
  - μ_o and μ_b are the mean gray levels of the objects and background
  - σ_o and σ_b- are their standard deviations



$$f_o(g) = \frac{p_o}{\sqrt{2p}\, s_o} e^{-1/2(\frac{g - m_o}{s_o})^2}$$

$$f_b(g) = \frac{p_b}{\sqrt{2p}\, s_b} e^{-1/2(\frac{g - u_b}{s_b})^2}$$

Kullback information distance

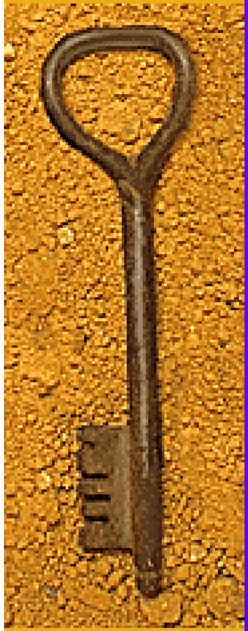
- ♦ Now, if we hypothesize a threshold, t, then all of these unknown parameters can be approximated from the image histogram.
- ♦ Let f(g) be the observed and normalized histogram
  - – f(g) = percentage of pixels from image having gray level g

$$p_o(t) = \sum_{g=0}^{t} f(g) \qquad p_b(t) = 1 - p_0(t)$$

$$m_o(t) = \sum_{g=0}^{t} f(g)g \qquad m_b(t) = \sum_{g=t+1}^{\max} f(g)g$$

Kullback information distance

- So, for any hypothesized t, we can “predict” what the total normalized image histogram **should** be if our model (mixture of two Gaussians) is correct.
  - $P_t(g) = p_o f_o(g) + p_b f_b(g)$
- The total normalized image histogram is **observed to be** f(g)
- So, the question reduces to:
  - determine a suitable way to measure the similarity of P and f
  - then search for the t that gives the highest similarity

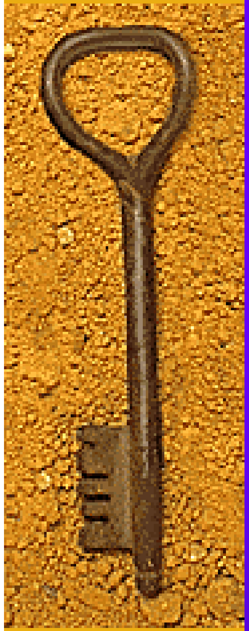


Kullback information distance

- A suitable similarity measure is the Kullback directed divergence, defined as

$$K(t) = \sum_{g=0}^{\max} f(g)\log[\frac{f(g)}{P_t(g)}]$$

- If P_t matches f exactly, then each term of the sum is 0 and K(t) takes on its minimal value of 0
- Gray levels where P_t and f disagree are penalized by the log term, weighted by the importance of that gray level (f(g))

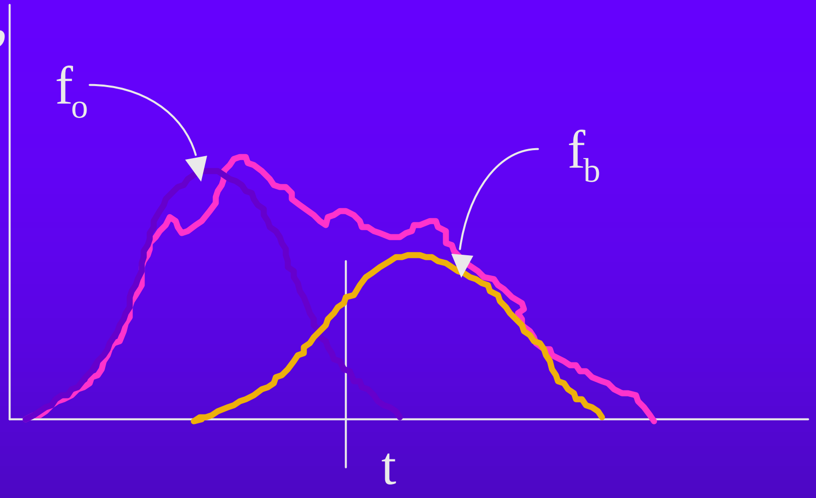


An alternative - minimize probability of error

- Using the same mixture model, we can search for the t that minimizes the predicted probability of error during thresholding
- Two types of errors
  - background points that are marked as object points. These are points from the background that are darker than the threshold
  - object points that are marked as background points. These are points from the object that are brighter than the threshold

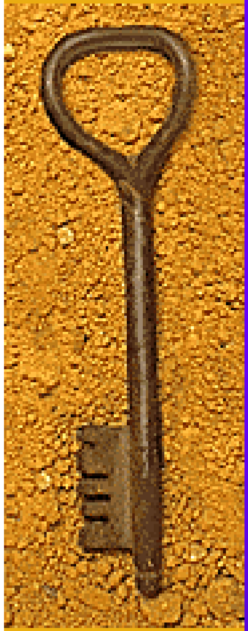
An alternative - mimimize probability of error

- For each “reasonable” threshold
  - compute the parameters of the two Gaussians and the proportions
  - compute the two probability of errors
- Find the threshold that gives
  - minimal overall error
  - most equal errors



$$e_b(t) = p_b \sum_{g=0}^{t} f_b(g)$$

$$e_o(t) = p_o \sum_{g=t+1}^{\max} f_o(g)$$



Nearest neighbor classifiers

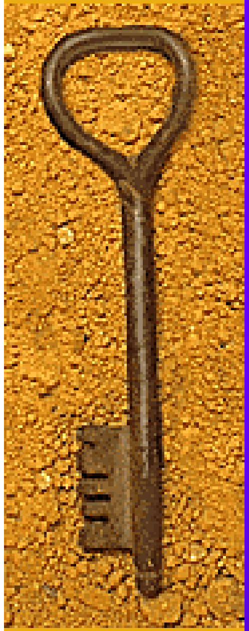
- Can use the training set directly to classify objects from the test set.
  - Compare the new object to every element of the training set
    - need a measure of closeness between an object from the training set and a test object

$$D(x,y)=\sum_{i}\frac{(x_i-y_i)^2}{s_i^2}$$

  - Choose the class corresponding to the closest element from the training set
  - Generalization - k nearest neighbors: find k nearest neighbors and perform a majority vote

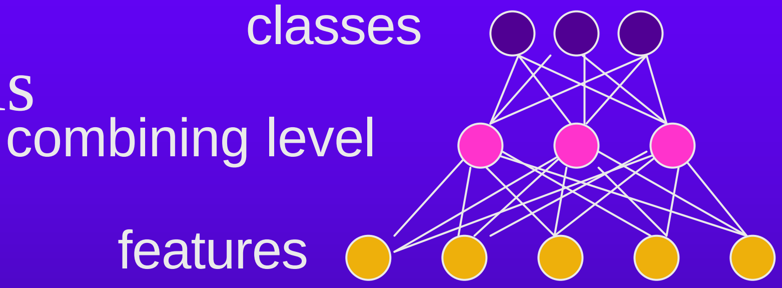
Nearest neighbor classification

- ♦ Computational problems
  - – Choosing a suitable similarity measurement
  - – Efficient algorithms for computing nearest neighbors with large measurement sets (high dimensional spaces)
    - k-d trees
    - quadtrees
    - but must use a suitable similarity measure
  - – Algorithms for “editing” the training set to produce a smaller set for comparisons
    - clustering: replace similar elements with a single element
    - removal: remove elements that are not chosen as nearest neighbors



Other classification models

- Neural networks
- Structural models
  - grammatical models
  - graph models
  - logical models
- Mixed models



classes

combining level

features