13\1

CS6640 Chapter 11 Week 13

classification: assign a set of pixels to a semantic class

binary: yes/no in one class

multi: choose + assign across k classes

Necessary items

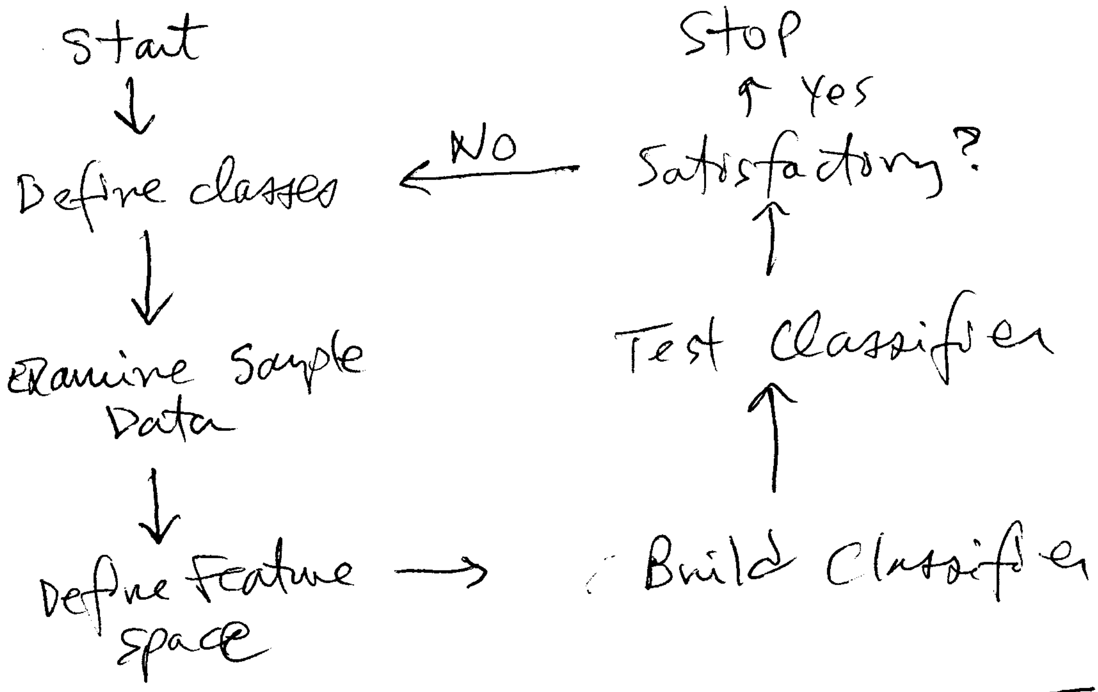
* Task specification
* Labelled data (to use for training)

Supervised: give pixels + their classes

Unsupervised: cluster "similar" pixels

May not use pixel intensity values, but rather compute features from pixel neighborhoods

Classification System Design



Issues (and terms)

Feature vector

feature space

training data

test data

pattern class

discriminant function

Simple Classifiers

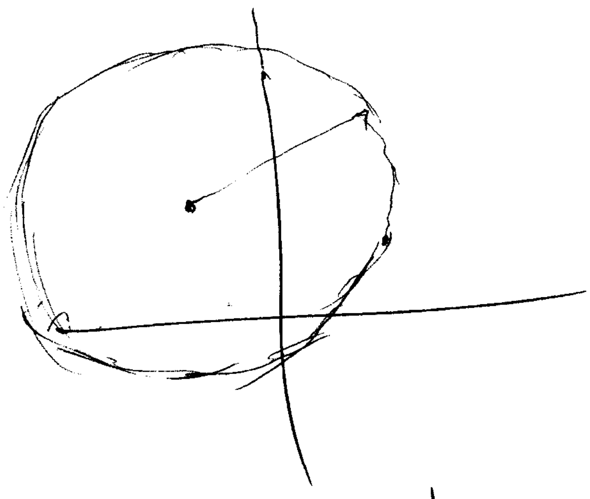
prototypes & min distance

distance functions:

Euclidean: $L(x,y) = \left[\sum_{i=1}^{N} (x_i - y_i)^2 \right]^{1/2}$

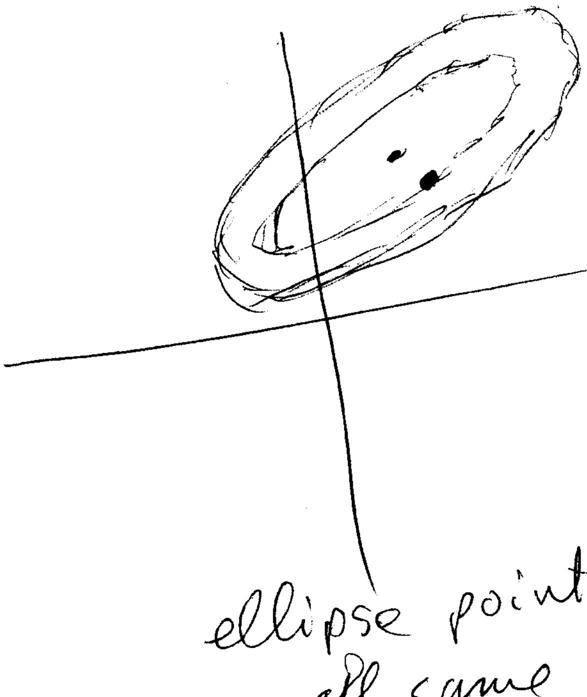
Mahalanobis: $L(x,y) = \left[(x-y)^T \Sigma^{-1} (x-y) \right]^{1/2}$

Euclidean



circular points are all same distance

Mahalanobis

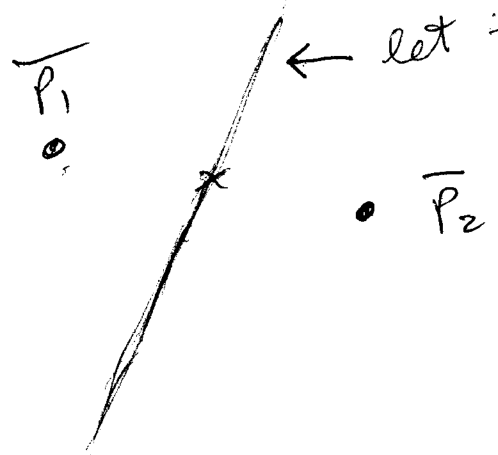


ellipse points are all same distance : takes covariance into account

Linear Discriminant Functions

Given 2 prototype vectors, $\bar{p}_1$ and $\bar{p}_2$ (means of samples) (say in 2D), consider line passing through midpoint between $\bar{p}_1$ + $\bar{p}_2$ and is orthogonal to the line containing $\bar{p}_1$ and $\bar{p}_2$

← let this be: $ax+by+c=0$



then $g(x,y) = ax+by+c$

gives the signed distance of the point from the line

So for a pair of prototypes $\bar{p}_i$ and $\bar{p}_j$

define $g_{ij}(x,y) = a_{ij}x + b_{ij}y + c$

such that g_{ij} is positive if closer to j, negative if closer to i

assign class: **if** $g_{kj}(x,y) > g_{jk}(x,y)$ $\forall j \neq k$

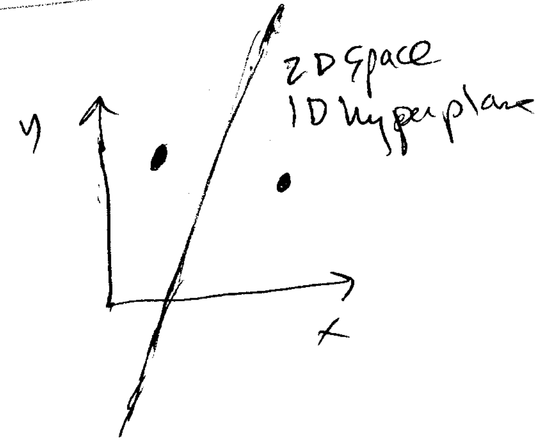
then assign (x,y) to class j

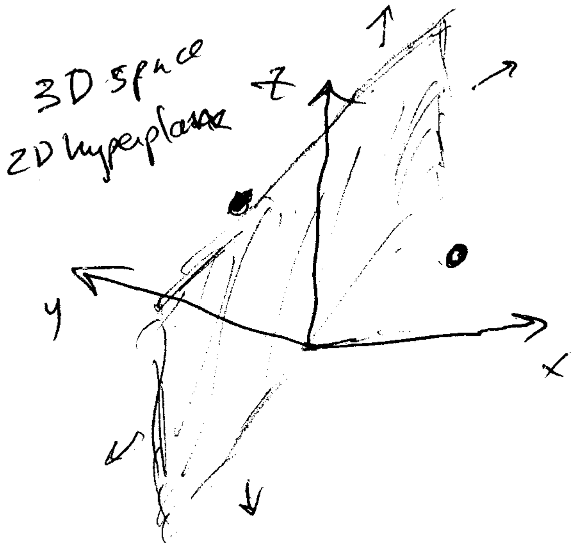
N dimensions

$$g(\bar{x}) = g(x_1, x_2, \ldots, x_N)$$
$$= \sum_{k=0}^{N} w_k x_k = w_0 + w_1 x_1 + \ldots + w_N x_N = 0$$

$$g(\bar{x}) = \bar{w} \cdot \bar{x} + w_0 = 0$$

represents a hyperplane





Note!: any two distinct points in a hyperplane define a vector that is orthogonal to the normal to the hyperplane

E.g., 2D

$[-4;0]$ $[-1;0]$ $y=0 \equiv 0x + 1y + 0 = 0$

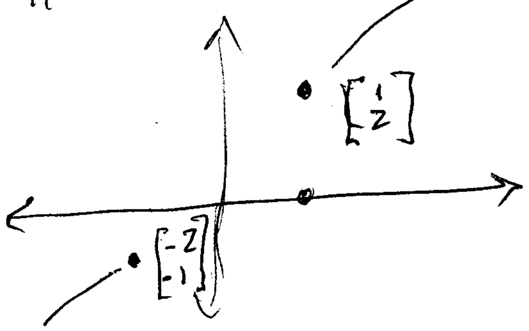
normal $[0;1] = \bar{n}$

$$\bar{d} = \begin{bmatrix} -3 \\ 0 \end{bmatrix} = \begin{bmatrix} -4 \\ 0 \end{bmatrix} - \begin{bmatrix} -1 \\ 0 \end{bmatrix} \qquad \bar{d} \cdot \bar{n} = \begin{bmatrix} -3 \\ 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 0$$

Note 2: equation gives distance (perpendicular), s, from arbitrary point to hyperplane

$$s = \frac{g(\bar{x})}{\|\bar{w}\|}$$

$0 \cdot 1 + 1 \cdot 2 + 0 = 2$



$0 \cdot x + 1y + 0 = g(x,y)$

$0 \cdot (-2) + 1 \cdot (-1) + 0 = -1$

and sign tells which side of line

To incorporate statistical info about features, use means and variance

use Mahalanobis distance

see CS6640 - Week 13

Bayesian classification

assign unknown feature vector
to most probable class

Assume

* N-dimensional feature vectors
* C classes
* Probability of class C, $P(C)$, is known (also called $P(w_j)$: class priors)
* $\exists$ M samples $\{\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_m\}$ whose corresponding classes are known

Bayes Decision Rule

if $P(w_j \mid \bar{x}) > P(w_k \mid \bar{x}) \quad \forall k \neq j$
then assign $\bar{x}$ to w_j

$\bar{x}$ is a feature vector
w_j is a class

$$P(w_j|\bar{x}) = \frac{P(\bar{x}|w_j)\,P(w_j)}{P(\bar{x})}$$

Since all $P(w_j|\bar{x})$ have $P(\bar{x})$ as denominator rule becomes:

<u>if</u> $p(\bar{x}|w_j)\,p(w_j) > p(\bar{x}|w_k)\,P(w_k) \quad \forall j \neq k$

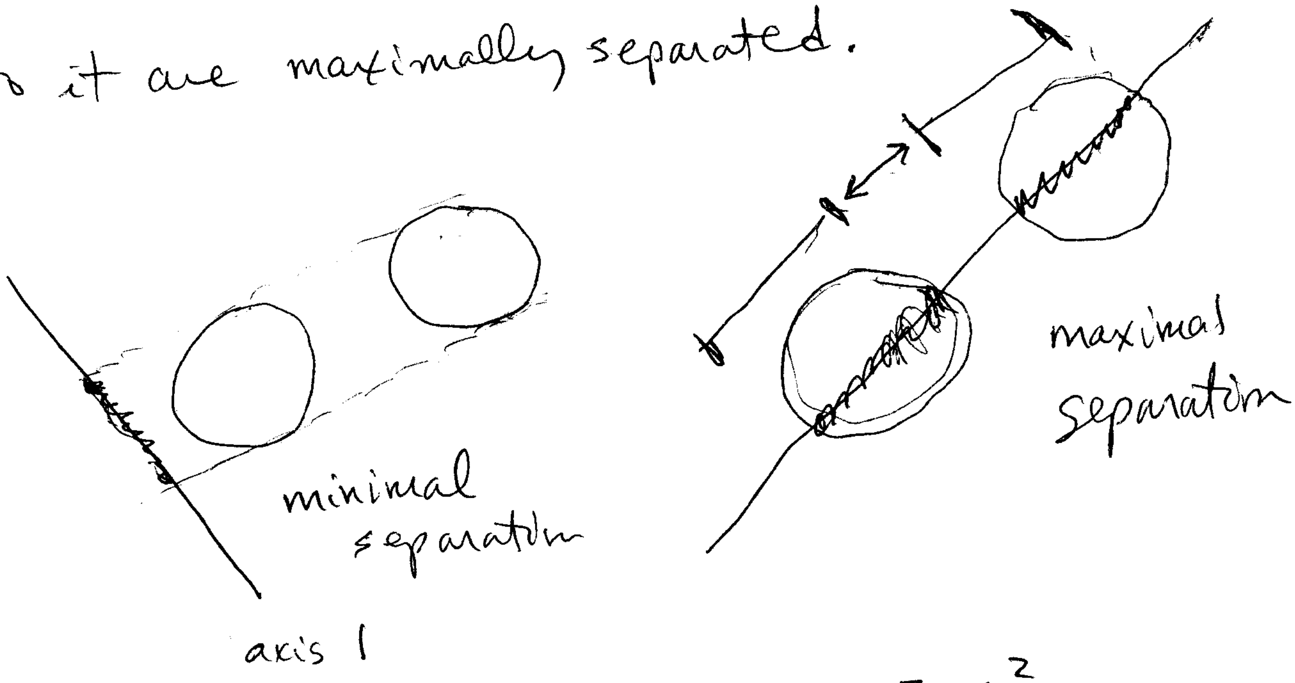
<u>then</u> assign $\bar{x}$ to ω_j

if we use the Multivariate Normal Distribution, then the iso-contours (of probability) are hyper-ellipsoids.

Text gives ways to reduce computation and take advantage of special properties of the covariance matrix.

Fisher discriminant function

Find an axis (1D) such that the points projected onto it are maximally separated.



Criterion

$$J(\bar{w}) = \frac{|\bar{m}_1 - \bar{m}_2|^2}{s_A^2 + s_B^2}$$

where $\bar{w}$ is the projection vector

$\bar{m}_1$ is mean of point set 1

$\bar{m}_2$ is mean of point set 2

s_1^2 is sample variance of projected points 1

s_2^2 is sample variance of projected points 2

Solution $\bar{w} = S_w^{-1} [M_A^x - M_B^x]$

M_A^x, M_B^x means of original vectors

$S_w = S_A + S_B$ sum of covariance matrices