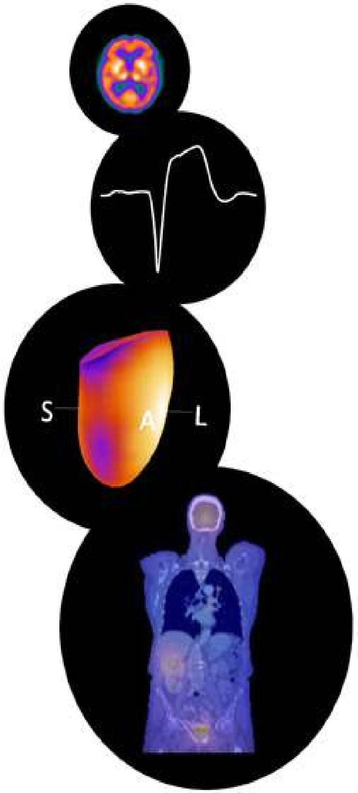


METU Informatics Institute
Min720

Pattern Classification with Bio-Medical Applications

Part 7:

Linear and Generalized Discriminant Functions

LINEAR DISCRIMINANT FUNCTIONS

Assume that the discriminant functions are linear functions of X.

$$g(X) = w_1 x_1 + w_2 x_2 + ... + w_n x_n + w_o$$

$$= W^T X + w_0$$

$$W = \left[w_1 ....w_n\right]^T$$

$$X = \left[x_1 ....x_n\right]^T$$

We may also write g in a compact form as:

$g(X) = W_a^{\ T} X_a = W_a^{\ T} Y$ where $Y = X_a = \left[x_1 .....x_n 1\right]$

$$W_a = \left[w_1 w_2 ......w_n w_0\right]^T$$

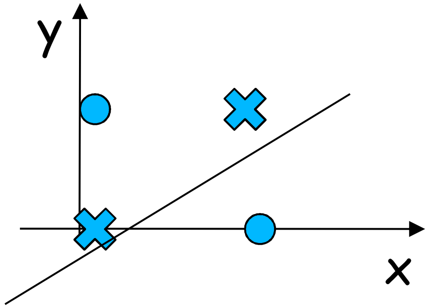
a-augmented

augmented pattern vector and weight factor.

Linear discriminant function classifier:

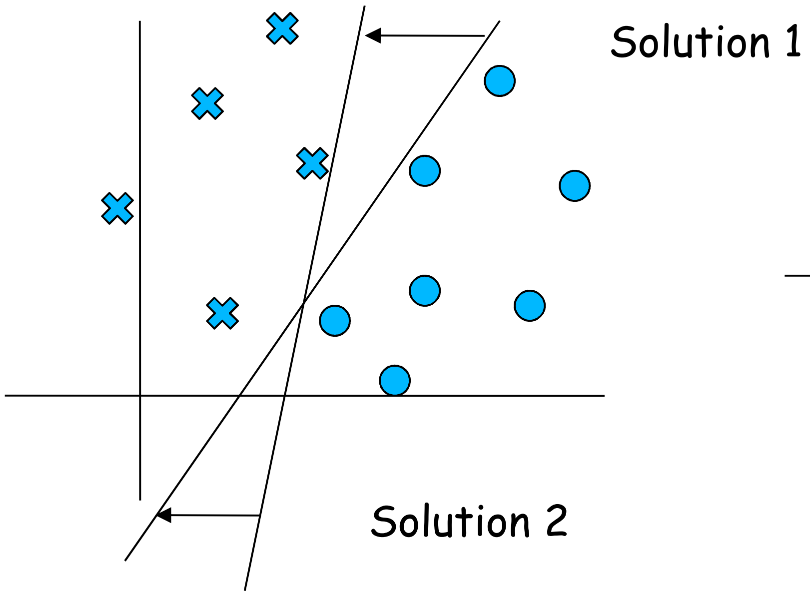
- It's assumed that the discriminant functions (g's)are linear.
- The labeled learning samples only are used to find best linear solution.
- Finding the g is the same as finding Wa.
- How do we define 'best'? All learning samples are classified correctly?
- Does a solution exist?

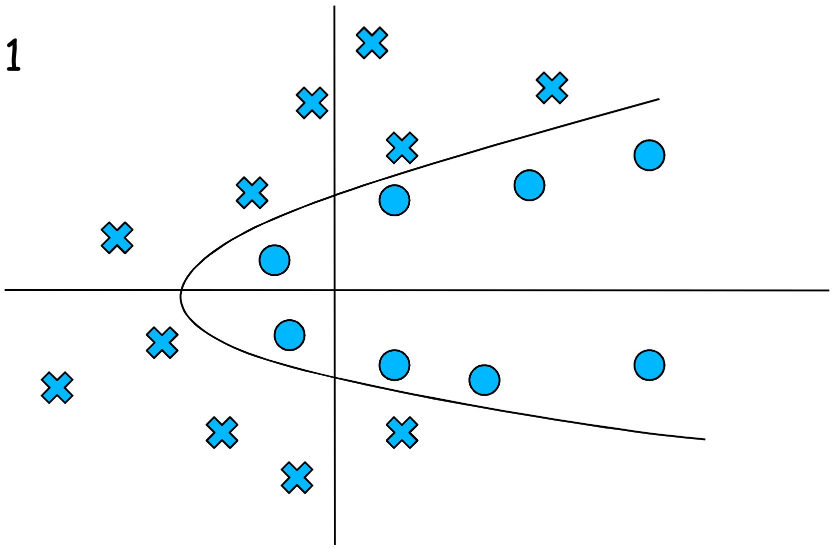
Linear Separability



XOR Problem
Not linearly separable

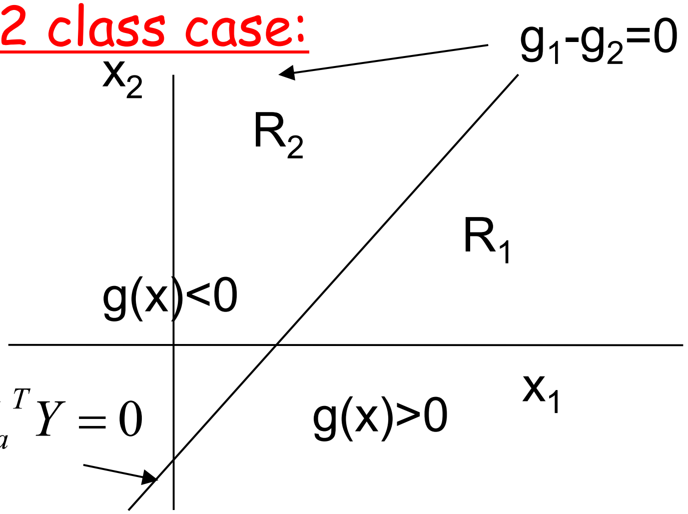
How do we determine W_a?





Many or no solutions possible

2 class case:



$$W_a^T Y = 0$$

The decision boundaries are lines, planes or hyperplanes. Our aim is to find best g.

$$\underbrace{W_{a1}^T X_a}^{g_1} = \underbrace{W_{a2}^T X_a}^{g_2} = W_{a2}^T Y$$

Where X is a point on the boundary (g_1=g_2)

$$g(X) = \underbrace{[W_{a1} - W_{a2}]}_{W_a}{}^T Y = 0$$

$$g(X) = g_1(Y) - g_2(Y)$$

a single disriminant function is enough!

$W_a^T Y = 0$ on the boundary.

or $W^T X + w_o = 0$ is the equation of the line for 2 feature problem..

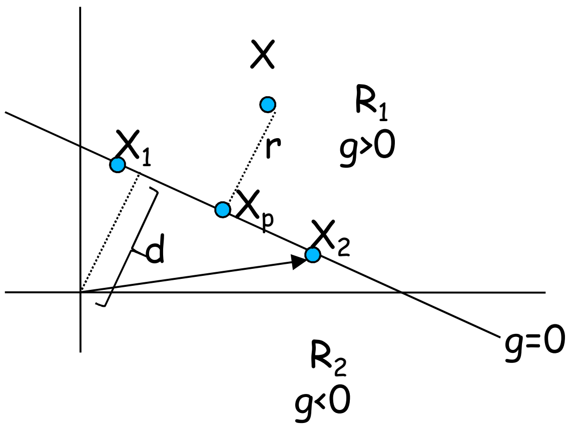
g(x)>0 in R_1 and g(x)<0 in R_2

Take two points on the hyperplane, X_1 and X_2

$$W^T Y_1 = W^T Y_2 = 0$$

$$W^T (X_1 - X_2) + \cancel{w_o} - \cancel{w_o} = 0$$

W is <u>normal</u> to the hyperplane.



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

Show!

Hint: $g(X_p)=0$

g(x) is proportional to the distance of X to the hyperplane.

$$d = \frac{w_o}{\|w\|}$$

distance from origin to the hyperplane.

- What is the criteria to find W?
    1. For all samples from c1, $g>0$, $WX>0$
    2. For all samples from c2, $g<0$ $WX<0$

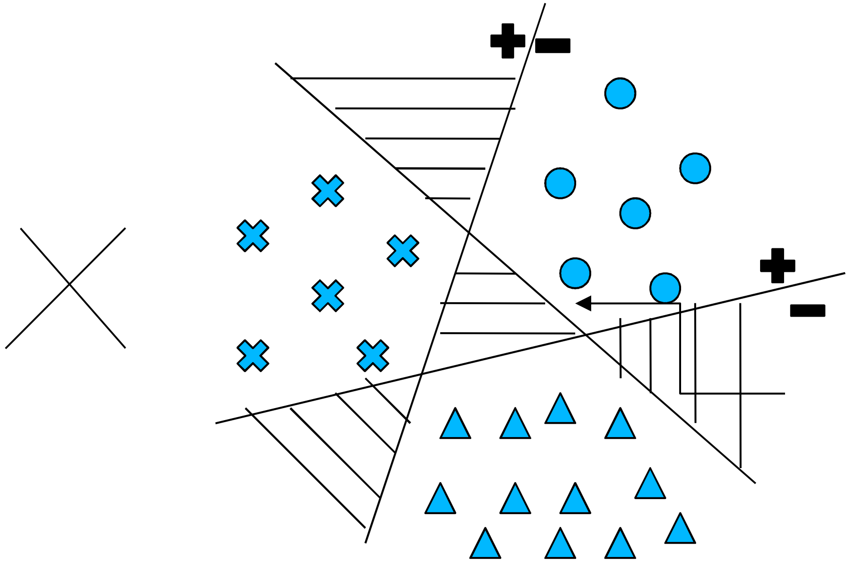
- That means, find an W that satisfies above if there is one.
- Iterative and non iterative solutions exist.

If a solution exists-the problem is called "linearly separable" and W_a is found iteratively. Otherwise "not linearly separable" piecewise or higher degree solutions are seeked.

Multicategory Problem

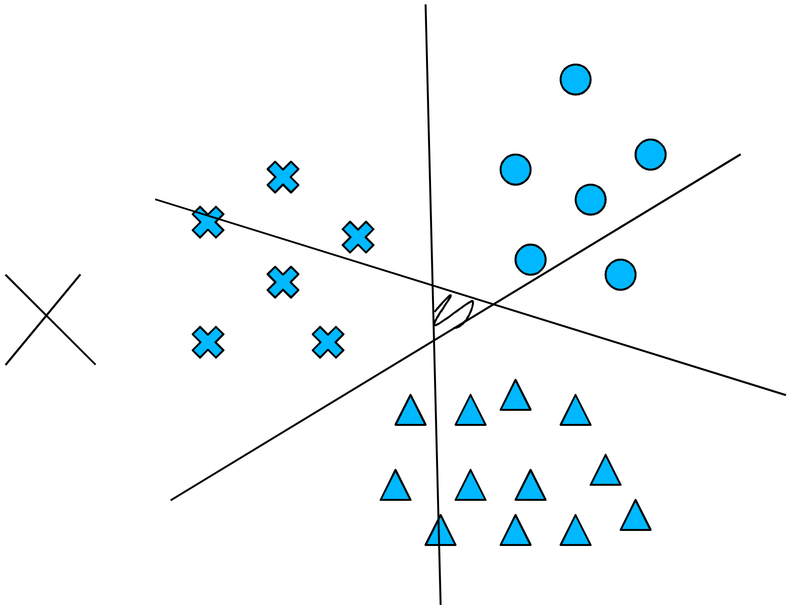
- Many to one
- Pairwise
- One function/category $\quad g_i = W_i X + w_{io}$

Many to one, Pairwise: Results with undefined regions

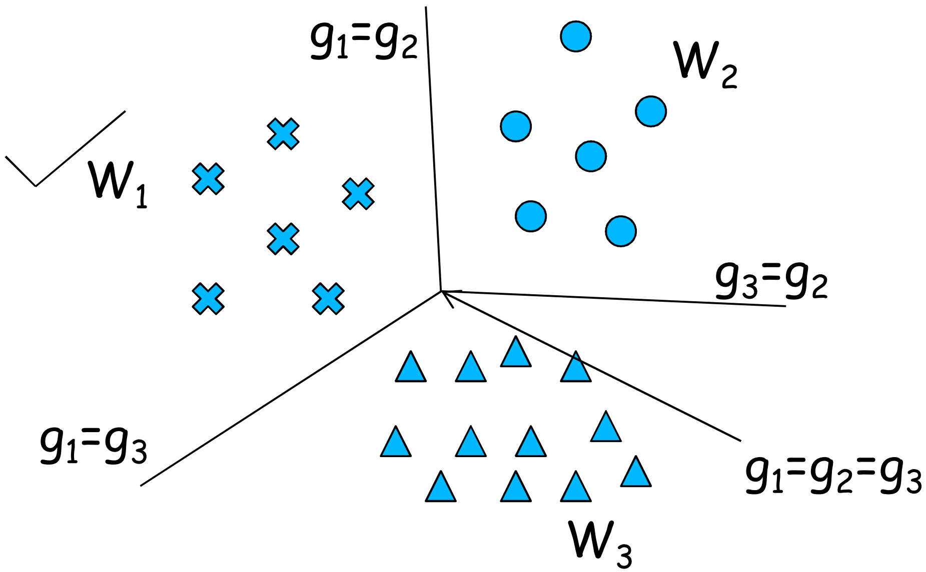


Many to one

undefined regions



pairwise



one function/category

Approaches for finding W: Consider 2-Class Problem

- For all samples in category 1, we should have

$$W_a^T X_a > 0$$

- And for all samples in category 2, we should have

$$W_a^T X_a < 0$$

- Take negative of all samples in category 2, then we need

$$W_a^T X_a > 0 \qquad \text{for all samples.}$$

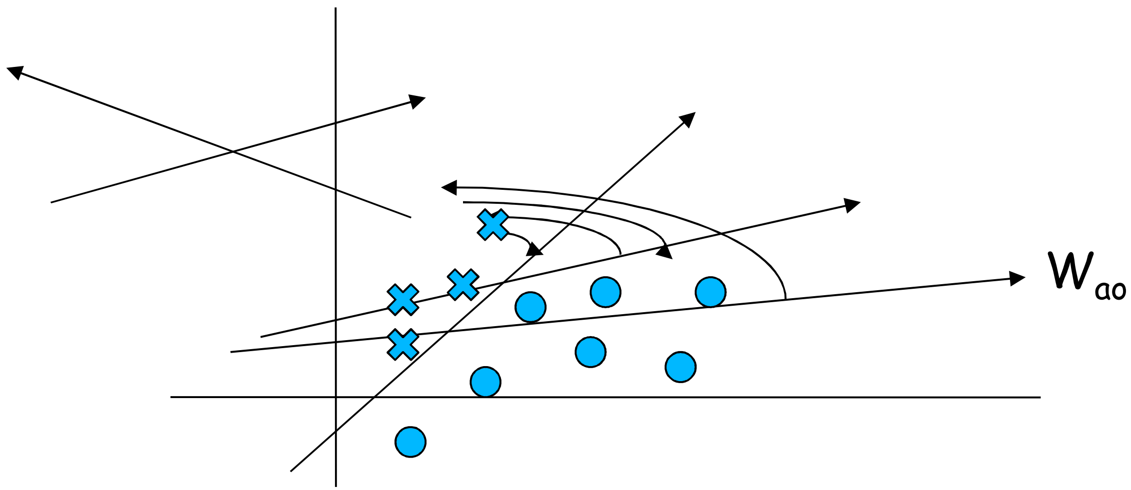
Gradient Descent Procedures and Perceptron Criterion

- Find a solution to

$$W_a^T X_a > 0$$

- If it exists for a learning set (X:labeled samples)
  - Iterative Solutions
  - Non-iterative Solutions

Iterative: start with an initial estimate and update it until a solution is found.



Gradient Descent Procedures:

Iteratively minimize a criterion function J(w).

Solutions are called "gradient descent" procedures.

- Start with an arbitrary solution.
- Find $\nabla J(w(1))$ - gradient
- Move towards the negative of the gradient

$$w(k+1) = w(k) - \eta(k) \nabla J(w(k))$$

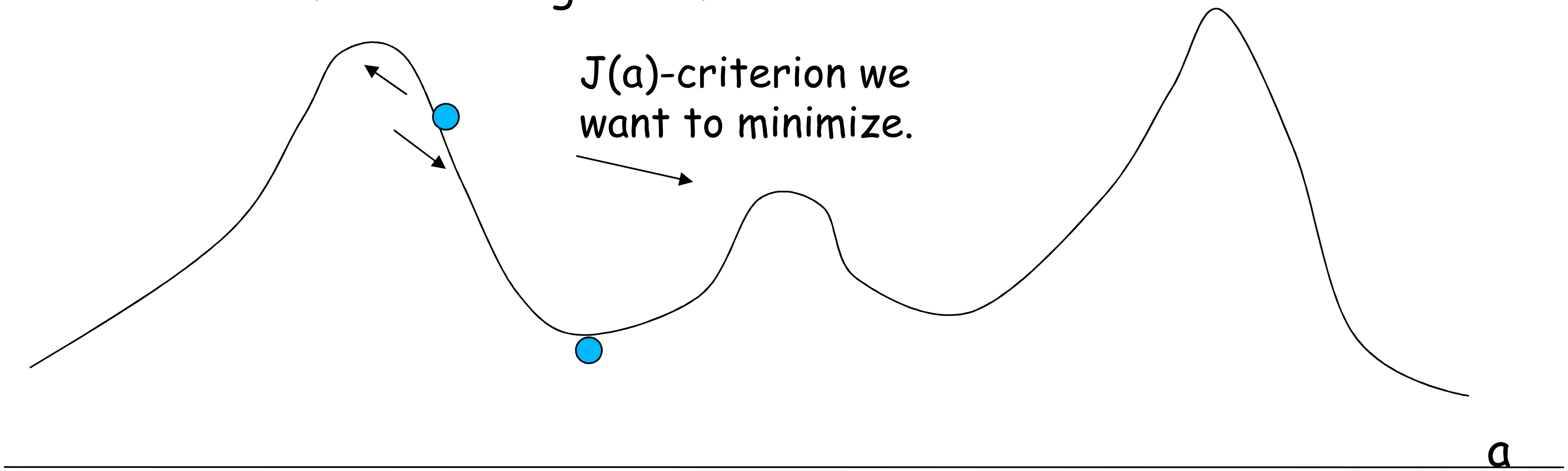
Learning rate

- Continue until $\eta(k)\nabla J(w(k) < \theta$

- What should η be so that the algorithm is fast?

  Too big: we pass the solution

  Too small: slow algorithm



Perceptron Criterion Function

$$J_p(w) = \sum_{y \in Y} (-w^T y)$$

•Where Y is the set of misclassified samples. {J_p is proportional to the sum of distances of misclassified samples to the decision boundary.}

$$\nabla J_p = \sum_{y \in Y} (-y)$$

Hence $$w_{k+1} = w_k + \eta(k) \underbrace{\sum_{y \in Y} y}_{\text{Sum of misclassified samples}}$$

Batch Perceptron Algorithm

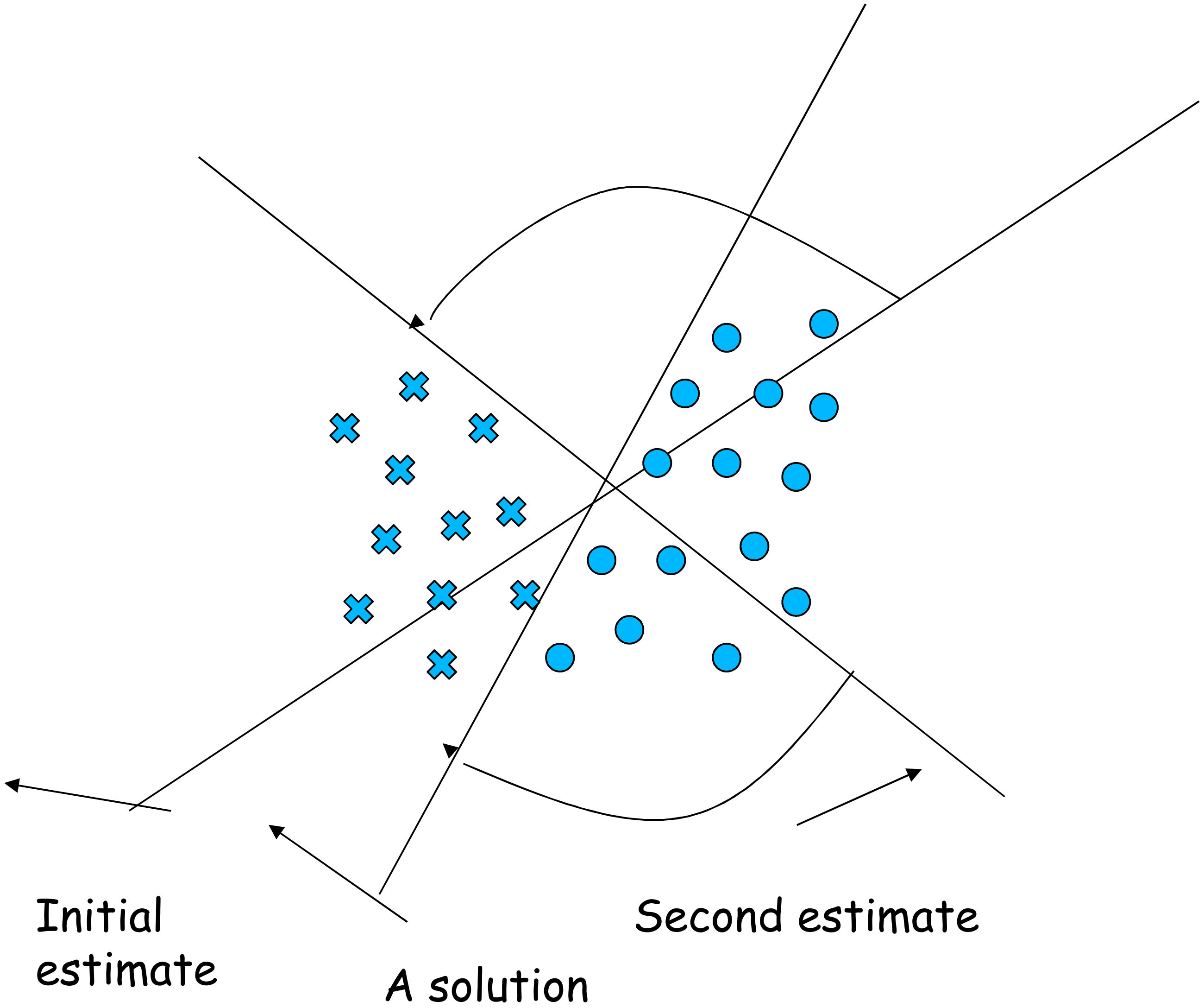
Initialize w, η , k $\leftarrow$ 0, θ

Do k $\leftarrow$ k+1

w $\leftarrow$ w+ $\eta(k) \sum Y$

Until $|\eta(k) \sum y| < \theta$

return w.

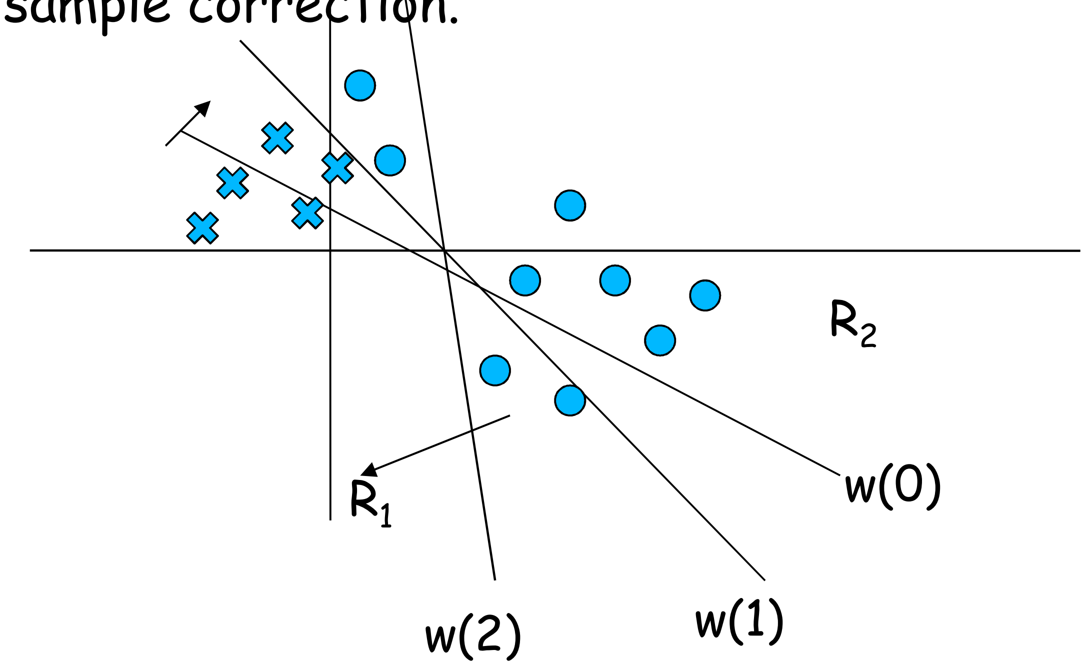


Initial estimate
A solution
Second estimate

Assume linear separability.

PERCEPTRON LEARNING

An iterative algorithm that starts with an initial weight vector and moves it back and forth until a solution is found, with a single sample correction.



Single Step Fixed-Increment Rule for 2-category Case

STEP 1. Choose an initial arbitrary $W_a(0)$.

STEP 2. Update $W_a(k)$ {k^{th} iteration} with a sample $X^{1i} \in C_1$ as follows (i'th sample from class 1)

$$W_a(k+1) = \begin{cases} W_a(k) & W_a^T(k)X_a^{1i} > 0 \\ W_a(k) + \eta X_a^{1i} & W_a^T(k)X_a^{1i} \le 0 \end{cases}$$

If the sample $X^{2i} \in C_2$, then

$$W_a(k+1) = \begin{cases} W_a(k) & W_a^T(k)X_a^{2i} < 0 \\ W_a(k) - \eta_k X_a^{2i} & W_a^T(k)X_a^{2i} \ge 0 \end{cases}$$

STEP 3. repeat step 2 for all samples until the desired inequalities are satisfied for all samples.

η_k - A positive scale factor that governs the stepsize. if $\eta_k = \eta$ fixed increment. We can show that the update of W tends to push it in the correct direction (towards a better solution).

$$W_a^T(k+1) = W_a^T(k) + \eta X_a^{1i^T}$$

Multiply both sides with X_a^{1i}

$$W_a^T(k+1)X_a^{1i} = W_a^T(k)X_a^{1i} + \underset{+ve}{\eta}\underbrace{X_a^{1i^T}X_a^{1i}}_{+ve} \geq W_a^T(k)X_a^{1i}$$

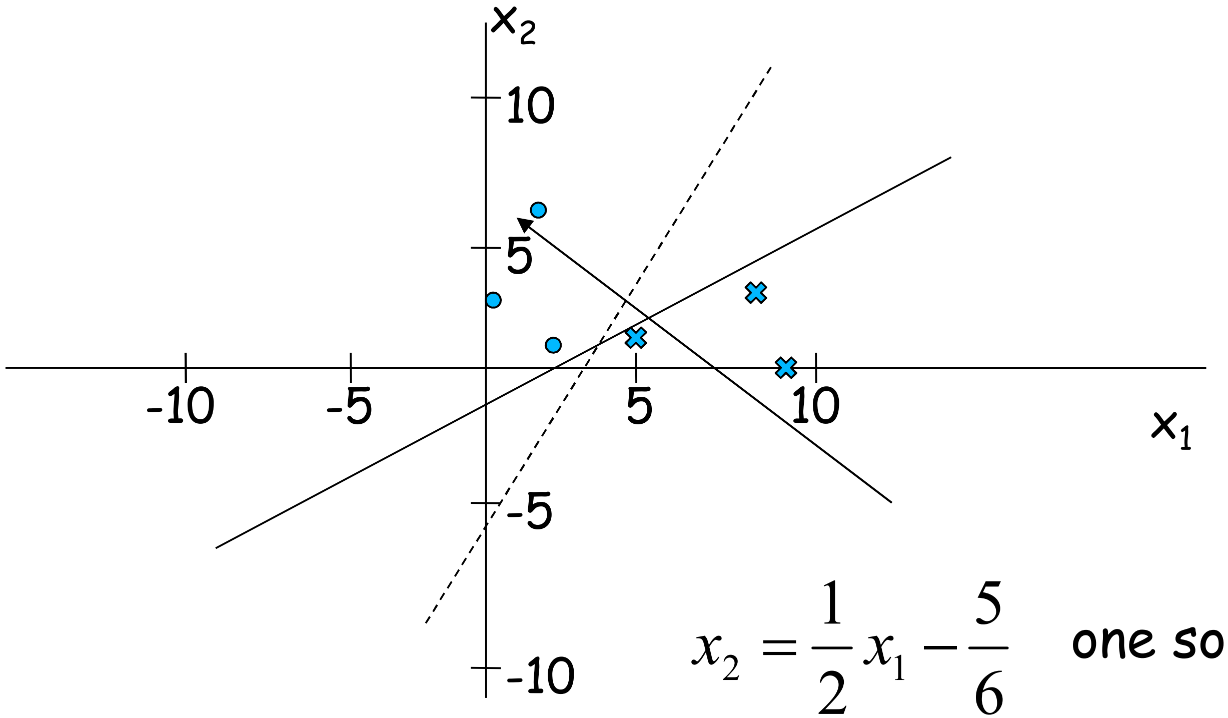
- Rosenblatt found in 50's.
- The perceptron learning rule shown by Rosenblatt is to converge in a finite number of iterations.

EXAMPLE (FIXED INCREMENT RULE)

Consider a 2-d problem where

$$X^{11} = \begin{bmatrix} 8 \\ 3 \end{bmatrix}, X^{12} = \begin{bmatrix} 5 \\ 1 \end{bmatrix}, X^{13} = \begin{bmatrix} 9 \\ 0 \end{bmatrix} \in C_1$$

$$X^{21} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}, X^{22} = \begin{bmatrix} 0 \\ 3 \end{bmatrix}, X^{23} = \begin{bmatrix} 3 \\ 6 \end{bmatrix} \in C_2$$



$$x_2 = \frac{1}{2}x_1 - \frac{5}{6}$$ one solution

Augment X's to Y's

$$X_a^{\ 21} = \begin{bmatrix} 3 \\ 1 \\ 1 \end{bmatrix} \quad X_a^{\ 12} = \begin{bmatrix} 5 \\ 1 \\ 1 \end{bmatrix} \quad X_a^{\ 13} = \begin{bmatrix} 9 \\ 0 \\ 1 \end{bmatrix}$$

$$X_a^{\ 21} = \begin{bmatrix} 3 \\ 1 \\ 1 \end{bmatrix} \quad X_a^{\ 22} = \begin{bmatrix} 0 \\ 3 \\ 1 \end{bmatrix} \quad X_a^{\ 23} = \begin{bmatrix} 3 \\ 0 \\ 1 \end{bmatrix}$$

Step 1. Assume $W_a(0) = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$

Step 2.

$$W_a(0)\begin{bmatrix} 8 \\ 3 \\ 1 \end{bmatrix} = 0$$

So update

$$W(1) = W(0) + \begin{bmatrix} 8 \\ 3 \\ 1 \end{bmatrix} = \begin{bmatrix} 8 \\ 3 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 8 & 3 & 1 \end{bmatrix}\begin{bmatrix} 5 \\ 1 \\ 1 \end{bmatrix} > 0 \qquad \begin{bmatrix} 8 & 3 & 1 \end{bmatrix}\begin{bmatrix} 9 \\ 0 \\ 1 \end{bmatrix} > 0$$

$$\begin{bmatrix} 8 & 3 & 1 \end{bmatrix}\begin{bmatrix} 3 \\ 1 \\ 1 \end{bmatrix} > 0$$ update

$$W(4) = \begin{bmatrix} 8 \\ 3 \\ 1 \end{bmatrix} - \begin{bmatrix} 3 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \\ 2 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 5 & 2 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 3 \\ 1 \end{bmatrix} > 0$$

$$\begin{bmatrix} 5 \\ 2 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 3 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \\ -1 \\ -1 \end{bmatrix}$$

$$\begin{bmatrix} 5 & -1 & -1 \end{bmatrix} \begin{bmatrix} 3 \\ 6 \\ 1 \end{bmatrix} > 0$$

$$\begin{bmatrix} 5 \\ -1 \\ -1 \end{bmatrix} - \begin{bmatrix} 3 \\ 6 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ -7 \\ -2 \end{bmatrix}$$

continue by going back to the first sample and iterate in this fashion.

SOLUTION:

$$W_a = \begin{array}{c} \begin{array}{ccc} w_1 & w_2 & w_0 \end{array} \\ \begin{bmatrix} 3 & -6 & -5 \end{bmatrix}^T \end{array}$$

Equation of the boundary:

$g(X) = 3x_1 - 6x_2 - 5 = 0$ on the boundary.

Extension to Multicategory Case

Class 1 samples { $X^{11}, X^{12}, \ldots\ldots\ldots\ldots X^{1n_1}$

Class 2 samples {$X^{21}, X^{22}, \ldots\ldots\ldots\ldots X^{2n_2}$

.

.

Class c samples {$X^{c1}, X^{c2}, \ldots\ldots\ldots\ldots X^{cn_c}$

$n_1 + n_2 + \ldots\ldots + n_c = n$ total number of samples

We want to find $g_1, \ldots\ldots, g_c$ or corresponding $W_1, W_2 \ldots\ldots, W_c$

so that $g_i = W_i^T X^{ik} > W_j^T X^{ik}$

For all $i \neq j$ and $1 \leq k \leq n_i$

The iterative single-sample algorithm

STEP 1. Start with random arbitrary initial vectors

$$W_1(0), W_2(0), \dots\dots\dots, W_c(0)$$

STEP 2. Update $W_i(k)$ and $W_j(k)$ using sample X^{is} as below:

$$W_i(k+1) = \begin{cases} W_i(k) & W_i(k)^T X^{is} > W_j(k)^T X^{is} \\ W_i(k) + \alpha(k) X^{is} & otherwise \end{cases}$$

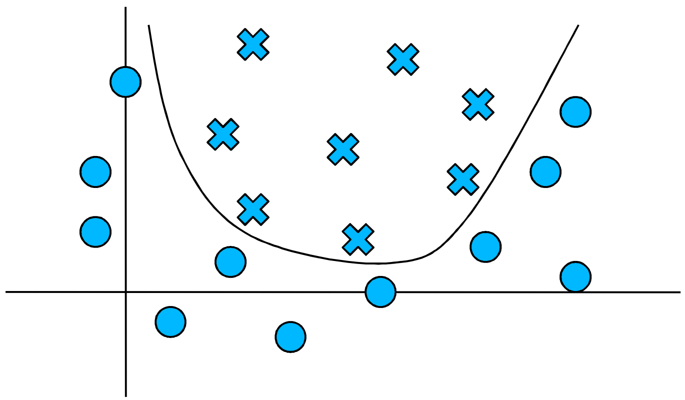
$$W_j(k+1) = \begin{cases} W_j(k) & W_i(k)^T X^{is} > W_j(k)^T X^{is} \\ W_j(k) - \alpha(k) X^{is} & otherwise \end{cases}$$

Do for all j.

STEP 3. Go to 2 unless all inequalities are satisfied <u>and repeat for all samples.</u>

GENERALIZED DISCRIMINANT FUNCTIONS

When we have nonlinear problems as below:



Then we seek for a higher degree boundary.

Ex: quadratic boundary

$$g(X) = \sum \sum w_{ij} x_i x_j + \sum w_i x_i + w_0$$

will generate hyperquadratic boundaries.

g(X) still a linear function w's.

$$g(Y_a) = W_a^T Y_a$$

$$= \underbrace{\begin{bmatrix} w_{11} & w_{12} & . & . & . & w_1 & . & . & . & w_n & w_0 \end{bmatrix}}_{W_a} \underbrace{\begin{bmatrix} x_1^2 \\ x_1 x_2 \\ . \\ x_n^2 \\ x_1 \\ . \\ x_n \\ 1 \end{bmatrix}}_{Y_a}$$

Then, use fixed increment rule as before using W_a and Y_a as above.

EXAMPLE: Consider 2-d problem,

$$g(X) = w_{11}x_1^2 + w_{22}x_2^2 + w_0$$ a general form for an ellipse.

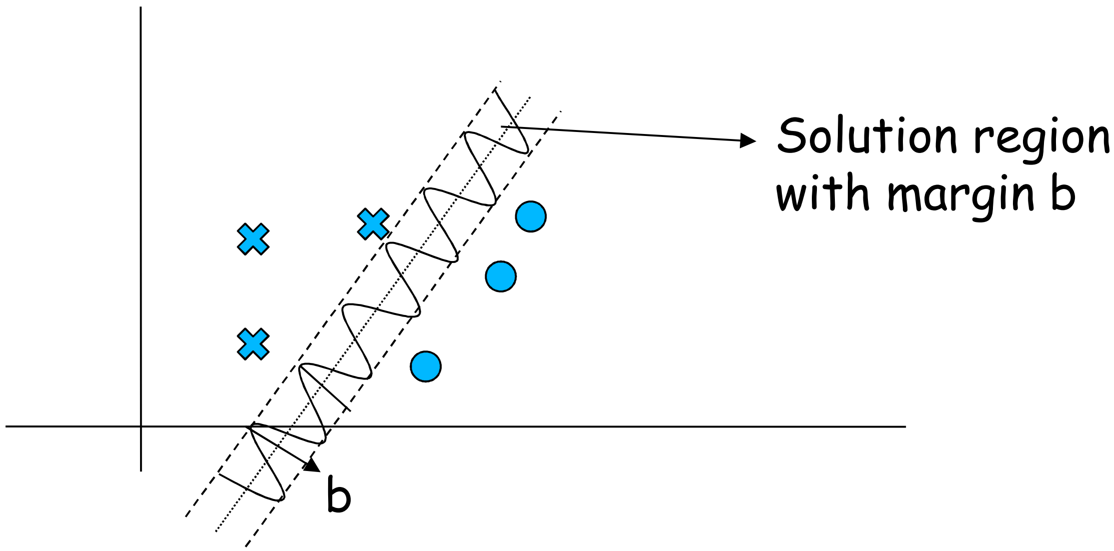
$$g(X) = \begin{bmatrix} w_{11} & w_{22} & w_0 \end{bmatrix} \begin{bmatrix} x_1^2 \\ x_2^2 \\ 1 \end{bmatrix}$$ so update your samples as Ya and iterate as before .

Variations and Generalizations of fixed increment rule

- Rule with a margin
- Non-seperable case: higher degree perceptrons
- : Neural Networks
- Non-iterative procedures: Minimum-squared Error
- Support Vector Machines

Perceptron with a margin

Perceptron finds "any" solution if there's one.



- A solution can be very close to some samples since we just check if

$$g(X_k) = W^T X_k \geq 0$$

- But we believe that a plane that is away from the nearest samples will generalize better (will give better solutions with test samples) so we put a margin b we say

$$W^T X_k \geq b \qquad\qquad (g(X) = r\|w\|)$$

- Restricts our solution region.

Distance from the separating plane

- Perceptron algorithm can be modified to replace 0 with b.
- *Gives best solutions if the learning rate* η *is made variable and*

$$\eta(k) \sim \frac{1}{k}$$

- Modified perceptron with variable increment and a margin:

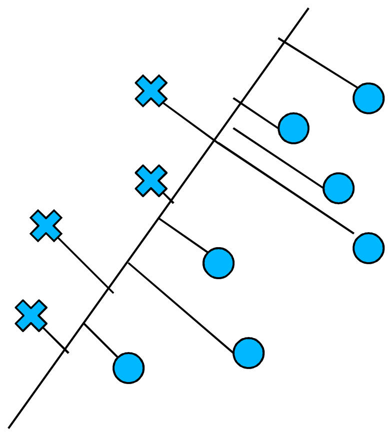
-if $W^T X_k \leq b$ then $W \longleftarrow W + \eta(k) X_k$

Shown to converge to a solution W.

NON-SEPARABLE CASE-What to do?

Different approaches

1- Instead of trying to find a solution that correctly classifies all the samples, find a solution that involves the distance of "all" samples to the seperating plane.



Instead of $W^T Y_i \geq 0$

Find a solution to $W^T Y_i = b_i$

where $b_i > 0$ (margin)

2-It was shown that we can increase the feature space dimension with a nonlinear transformation, the results are linearly separable. Then find an optimum solution.(Support Vector Machines)

3. Perceptron can be modified to be multidimensional –Neural Nets

Minimum-Squared Error Procedure

Good for both separable and non-separable problems

$W^T Y_i = b_i$ for the i^{th} sample where Y_i is the augmented feature vector.

Consider the sample matrix $A = \begin{bmatrix} Y_1^T \\ Y_2^T \\ . \\ Y_n^T \end{bmatrix}$ (For n samples A is dxn))

Then AW=B $\quad B = \begin{bmatrix} b_1 \\ b_2 \\ . \\ b_n \end{bmatrix}$

cannot be solved since many equations, but not enough unknowns.
(many solutions exist;more rows than columns)

So find a W so that

$\|AW - B\|$ is minimized.

Well-known least square solution (from the gradient and set it zero)

$$W = \underbrace{\underbrace{\left[A^T A\right]^{-1} A^T}_{A^+} B}$$

Pseudo-inverse if <u>A^TA is non-singular </u>(has an inverse)

B is usually chosen as

B= [1 1........... 1]^T (was shown to approach Bayes discriminant when $n \to \infty$)