Linear Classifier: Linear Discriminant Function

Compiled by Lakshmi Manasa, CED16I033
Guided by
Dr Umarani Jayaraman

Department of Computer Science and Engineering
Indian Institute of Information Technology Design and Manufacturing
Kancheepuram

April 18, 2022

Discriminant Function

- We know the proper forms for the discriminant functions and use the samples to estimate the values of parameters of the discriminant function
- Although it estimates the parameters of the discriminant function, it is said to be **non-parametric form** as it does require the knowledge about the probability distributions.
- Linear Discriminant function will be formulated as a problem of minimizing a criterion function.
- **Criterion function:** the obvious criterion function for classification purpose is the sample risk or training error.

Discriminant function

- **Training error:** The average loss incurred in classifying the set of training samples.
- **No probability form is assumed:** If the parametric form of the class-density function is not known; then we have to design the decision boundary using samples which are available with us.
- Here, we don't assume any parametric form of any probability distribution function.
- But, what we know is that, the classes are linearly separable

Linear Discriminant Function

- Non parametric form
- Supervised Learning
- Classes are linearly separable
- Classes : ω_1 and ω_2
- Using this information, as the classes are linearly separable, we can formulate the linear equation as $g(x) = W^t X + w_0$ X - d-dimensional vector W - d-dimensional weight vector $W^t X$ - Inner product of two vectors w_0 - bias/threshold weight

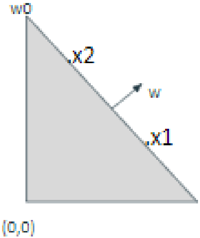
Decision criteria

- $g(x) > 0; x \epsilon \omega_1$
- $g(x) < 0; x \epsilon \omega_2$
- $g(x) = 0$; then x on the decision boundary

Now let us analyze the significance of each attribute in the equation, $g(x) = W^t X + w_0$

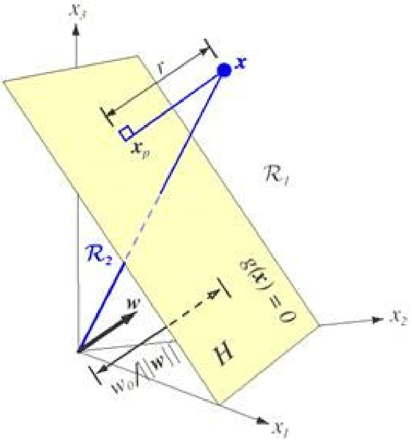
- Nature of weight vector W
- What does g(x) represents?

1. Nature of weight vector w



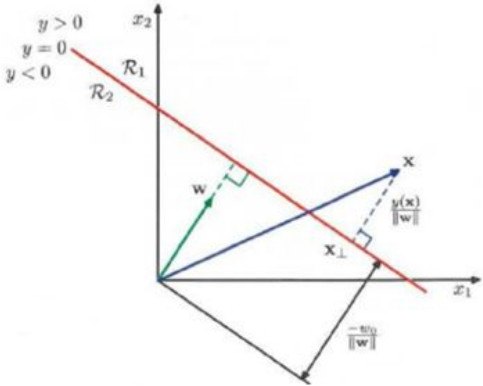
- $g(X_1) = g(X_2)$
- $W^t X_1 + w_0 = W^t X_2 + w_0$
- $W^t(X_1 - X_2) = 0$
- We know that, $A.B = |A|.|B| cos\Theta$;
- If $A.B = 0$, then A is perpendicular to B
- Likewise, $W^t(X_1 - X_2)$ is the inner product of weight vector W with $(X_1 - X_2)$.
- As it is zero, it indicates that vector 'W' is orthogonal to any vector lying on decision surface.
- In d-dimensional space, this surface is called as Hyper plane 'H'.

2. What does g(x) represents?



- Draw a perpendicular line from a point x to the Hyper plane 'H' which is X_p
- Let the distance of X and Xp is 'r' Then, $X = X_p + r.\frac{W}{||W||}$

2. What does g(x) represents?



- As seen earlier, W is orthogonal to the hyper plane 'H'.
- So, the direction of 'W' is same direction of from X_p to X.
- Hence, Both X_p to X and 'W' is orthogonal to hyper plane 'H'

2. What does g(x) represents?

$$\boxed{\frac{W}{||W||} = \frac{\sum_{i=1}^{d} w_i}{\sqrt{\sum_{i=1}^{d}(w_i)^2}}}$$

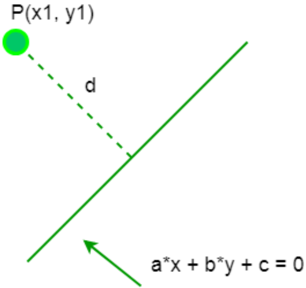
$$\boxed{X = X_P + r.\frac{w}{||W||}}$$

- $g(X) = W^t X + w_0$
- $g(X) = W^t[X_p + r.\frac{w}{||W||}] + w_0$
- $g(X) = W^t X_p + w_0 + r.\frac{W^t.W}{||W||}$

The point X_p that lies on the decision surface so $W^t X_p + w_0$ is zero.

- $g(X) = 0 + r.\frac{W^t.W}{||W||}$
- $g(X) = 0 + r.\frac{||W||^2}{||W||}$
- $g(X) = r.||W||$

2. Why g(x) is algebraic measure?

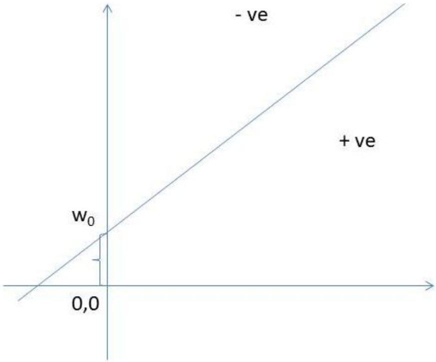


- If $ax + by + c = 0$ is the equation of the straight line and (x_1, y_1) is a point, then distance of (x_1, y_1) to the line is nothing but
- $d = \frac{ax_1+by_1+c}{\sqrt{a^2+b^2}}$ In, 2-dimension.
- $r = \frac{g(x)}{||w||^2}$ In, d-dimension.

3. Distance of origin from the hyperplane H

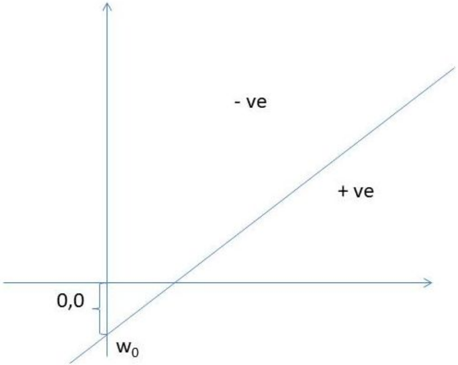
- Distance of origin from the huperplane H is $\frac{w_0}{||W||}$; w_0 is Bias/Threshold.
- If w_0 is $+$ve, then origin lies on the $+$ve side of the hyper plane 'H'.
- If w_0 is -ve, then origin lies on the -ve side of the hyper plane 'H'.
- If w_0 is zero, then the hyper plane passes through origin. And also,

3. Distance of origin from the hyperplane H



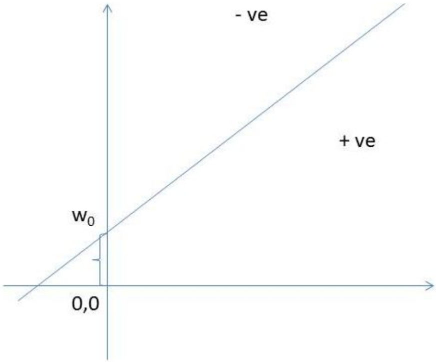
If w_0 is positive then origin lies in positive side

3. Distance of origin from the hyperplane H



If w_0 is negative then origin lies in negative side

3. Distance of origin from the hyperplane H



If w_0 is positive then origin lies in positive side

3. Distance of origin from the hyperplane H

- If w_0 is zero, discriminant function $g(x)$ takes the particular form $g(x) = W^T X$; in this case we don't have any bias because $w_0 = 0$
- $g(x) = W^T X$ is said to be in Homogeneous form
- In mathematics, It is convenient, If we represent the equation in Homogeneous form.
- So, in order to design a linear classifier we should estimate two parameters such as weight vector W and bias w_0.
- Since it is supervised learning W and w_0 are supposed to be estimated based on the samples that are available.

Design of weight vector W

- **Assumption: Two classes and linearly separable case**
- We have two classes and it is linearly separable
- We should have the discriminant function which separates these two classes
- It is of the form $g(X) = W^t X + w_0$
- This expression is not in homogeneous form.
- Hence, converting this homogeneous form makes the analysis easier.

Converting to Homogeneous form

- $g(X) = W^t X + w_0$
- $g(X) \approx a^t y$
- $g(X) \approx \begin{bmatrix} w_1 & w_2 & \dots & \dots & w_d & w_0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ .. \\ x_d \\ 1 \end{bmatrix}$
- $g(X) \approx \sum_{i=1}^{d} w_i x_i + w_0$
- $g(X) \approx W^t X + w_0$

Decision rule in Homogeneous form

- **The decision rule remains the same, for** $a^t y$
- If $a^t y > 0$ then decide $y \epsilon \omega_1$
- If $a^t y < 0$ then decide $y \epsilon \omega_2$
- If $a^t y = 0$ then no decision can be taken.

How to design weight vector 'W' and w_0 using the samples?

- We have n- no of samples (or) training samples $y_1, y_2, ..., y_n$
- $y_1 = \begin{bmatrix} x_{11} \\ x_{12} \\ .. \\ .. \\ x_1 d \\ 1 \end{bmatrix}$ $y_2 = \begin{bmatrix} x_{21} \\ x_{22} \\ .. \\ .. \\ x_{2d} \\ 1 \end{bmatrix}$ $y_3 = \begin{bmatrix} .. \\ .. \\ .. \\ .. \\ .. \\ .. \end{bmatrix}$ $y_4 = \begin{bmatrix} .. \\ .. \\ .. \\ .. \\ .. \\ .. \end{bmatrix}$ $y_n = \begin{bmatrix} x_{n1} \\ x_{n2} \\ .. \\ .. \\ x_{nd} \\ 1 \end{bmatrix}$
- These are the samples which are useful to train the classifier.
- Some of the samples are labeled as ω_1 and some are labeled as ω_2.
- Let's consider the i^{th} sample as y_i.

Two Criterion Decision rule in Homogeneous form

- **The decision rule remains the same, for** $a^t y_i$
- If $a^t y_i > 0$ then decide $y_i \epsilon \omega_1$
- If $a^t y_i < 0$ then decide $y_i \epsilon \omega_2$
- If $a^t y_i = 0$ then no decision can be taken.

Two Criterion Decision rule in Homogeneous form

- Given a weight vector 'a'; If we take all the samples which are labelled as ω_1
- If for each of the samples, $a^t y_i > 0$; then that weight vector 'a' is correctly classifying all the samples which are labelled as ω_1
- If we also find, for the same weight vector 'a' all the samples belonging to class ω_2;
- If $a^t y_i < 0$; then the weight vector 'a' is also classified correctly for all samples belongs to class ω_2
- That particular weight vector 'a' is the correct weight vector, because it is correctly classified all the samples labelled as ω_1, also it is correctly classified all the samples labelled as ω_2.

Single Criterion

- Instead of two conditions $a^t y_i > 0$ and $a^t y_i < 0$, Can't we have a single criterion to classify correctly.
- $a^t y_i > 0$ true, irrespective of class label.
- We can say that, y_i is correctly classified if $a^t y_i > 0$. Otherwise, y_i is mis-classified.
- Otherwise, include < 0 and $= 0$.

Single Criterion: How can we do that?

- Samples belonging to class ω_1, we can take them as it is.
- For the samples belonging to class ω_2, we augment them by appending 1 and then take negative of it.
- Take all the samples which are labelled as ω_2 and then negate it.
- Instead of considering y_i, consider $-y_i$
- If we take negative, this $a^t y_i$ which is supposed to be < 0, now it will be > 0.
- So, we get single (uniform) decision criterion which is $a^t y_i > 0$ for both the classes.

Single Criterion: How can we do that?

- If $a^t y_i > 0$, all samples are correctly classified, irrespective of class labels.
- Now, for this what will be the weight vector 'a'?
- We take some criteria Function, $J(a)$.
- $J(a)$ has to be minimized, if 'a' is a solution (correct weight) vector.
- $J(a)$ will be minimum, If it classifies all the training samples correctly, for the weight vector 'a' which is obtained.
- For minimization of $J(a)$, we can make use of **Gradient Descent Procedure**.

Gradient Descent Procedure

Gradient Descent

Gradient descent is an iterative optimization algorithm for finding the **minimum** of a function.

4

next position

opposite direction

gradient at current position

-10

$$x_{i+1} = x_i - \alpha \nabla f(x_i)$$

current position

-5

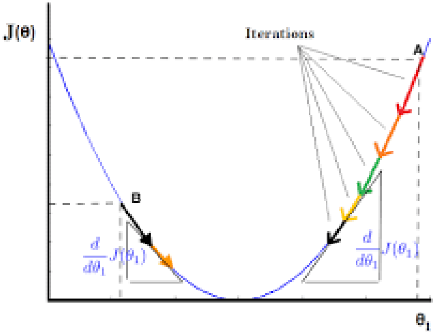
step size (learning rate)

set to 0.9 here, usually much smaller

Gradient Descent Procedure

- Initialize with weight vector $a(k)$ with some random values and try to minimize the training error for every iteration.
- At the k^{th} iteration, we know the values of a(k).
- We should update the weight vector for a(k+1).
- $a(k+1) = a(k) - \eta(k) \bigtriangledown J(a(k))$
- This is called **Gradient Descent Procedure** or **Steepest Descent Procedure**.

Algorithm: Gradient Descent



- Initialize a, threshold $\theta, \eta(.), k \leftarrow 0$
- **do** $k \leftarrow k + 1$
- $a \leftarrow a - \eta(k) \bigtriangledown J(a)$
- **until** $\eta(k) \bigtriangledown J(a) < \theta$
- return a

Perceptron Criterion Function

- Our aim will be to find out weight vector 'a' which will classify all the training samples correctly.
- So, we can try to design a criterion function which will make use of samples which are **not correctly classified**.
- If the samples are not correctly classifies by a(k), then update weight vector 'a' in a(k+1)
- So, accordingly we can define the criterion function
- **Criterion function** can be defined as,
- $J_p(a) = \sum_{\forall y \text{ misclassified}} (-a^t y)$, here p refers to perceptron criterion.

Perceptron Criterion Function

- $J_p(a) = \sum_{\forall y \text{ misclassified}} (-a^t y)$,
- Here, $(-a^t y)$ is positive.
- As a result, The criterion for $J_p(a)$, never have a negative value.
- It can always have a positive value.
- The minimum value can be 0.
- So, we have a global minimum for $J_p(a)$ and this can be find by Gradient Decent Procedure.

Perceptron Criterion Function

- According to Gradient Descent Procedure, take gradient of $J_p(a)$ w.r.t weight vector a.
- $J_p(a) = \sum_{\forall y \text{ misclassified}}(-a^t y)$,
- $\triangledown .\ J_p(a) = \sum_{\forall y \text{ misclassified}}(-y)$
- The update rule is,

  $\boxed{a(0) \Rightarrow \text{Initial weight vector; arbitrary}}$.

  $\boxed{a(k+1) = a(k) + \eta(k) \sum_{\forall y \text{ misclassified}}(y)}$
- This is the algorithm to design weight vector 'a' if the samples are linearly separable.

THANK YOU