Linear Decision Rules

Robert M. Haralick

Computer Science, Graduate Center
City University of New York

Outline

Fisher Linear Discriminant (1936)

- 2-Class case
  - Can be generalized to K-Classes
- Find a direction vector w so that the ratio of the between class variance to within class variance is maximized
- Project the data on that direction
- Choose a threshold that maximizes the classification accuracy

Within and Between Class Variance

- X is a random variable
- X comes from a mixture distribution
- $\mu_i = E[X \mid c_i]$, $\Sigma_i = E[(X - \mu_i)(X - \mu_i)' \mid c_i]$, i=1,2
- Mixture Fractions: $p_1, p_2 \geq 0$, $p_1 + p_2 = 1$
- $E[X] = E[X \mid c_1]p_1 + E[X \mid c_2]p_2$
- $\mu = p_1\mu_1 + p_2\mu_2$

$$\begin{aligned}
\Sigma &= E[(X - \mu)(X - \mu)'] \\
&= E[(X - \mu)(X - \mu)' \mid c_1]p_1 + E[(X - \mu)(X - \mu)' \mid c_2]p_2
\end{aligned}$$

Within and Between Class Variance

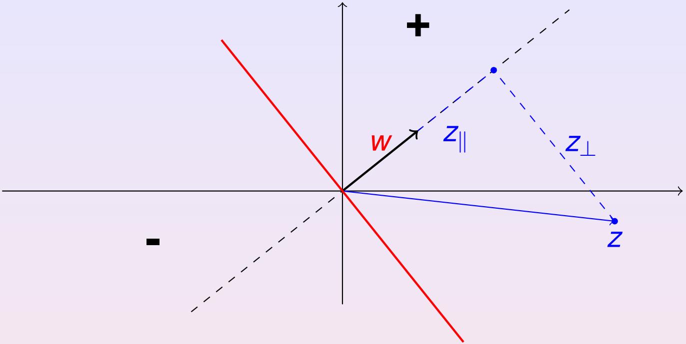
$$
\begin{aligned}
E[(X-\mu)(X-\mu)' \mid c_1] &= E[(X-\mu_1+(\mu_1-\mu))(X-\mu_1+(\mu_1-\mu))' \mid c_1] \\
&= E[(X-\mu_1)(X-\mu_1)' \mid c_1] + E[(X-\mu_1)(\mu_1-\mu)' \mid c_1] + \\
&\quad E[(\mu_1-\mu)(X-\mu_1)' \mid c_1] + E[(\mu_1-\mu)(\mu_1-\mu)' \mid c_1] \\
&= E[(X-\mu_1)(X-\mu_1)' \mid c_1] + E[(\mu_1-\mu)(\mu_1-\mu)' \mid c_1] \\
&= \Sigma_1 + (\mu_1-\mu)(\mu_1-\mu)' \\
&= \Sigma_1 + (\mu_1-(p_1\mu_1+p_2\mu_2))(\mu_1-(p_1\mu_1+p_2\mu_2))' \\
&= \Sigma_1 + p_2^2(\mu_1-\mu_2)(\mu_1-\mu_2)'
\end{aligned}
$$

Within and Between Class Variance

$$
\begin{aligned}
\Sigma &= E[(X-\mu)(X-\mu)' \mid c_1]p_1 + E[(X-\mu)(X-\mu)' \mid c_2]p_2 \\
&= p_1(\Sigma_1 + p_2^2(\mu_1-\mu_2)(\mu_1-\mu_2)') + p_2(\Sigma_2 + p_1^2(\mu_2-\mu_1)(\mu_2-\mu_1)') \\
&= p_1\Sigma_1 + p_2\Sigma_2 + (p_1p_2^2 + p_2p_1^2)(\mu_1-\mu_2)(\mu_1-\mu_2)' \\
&= p_1\Sigma_1 + p_2\Sigma_2 + p_1p_2(p_1+p_2)(\mu_1-\mu_2)(\mu_1-\mu_2)' \\
&= p_1\Sigma_1 + p_2\Sigma_2 + p_1p_2(\mu_1-\mu_2)(\mu_1-\mu_2)' \\
\Sigma_W &= p_1\Sigma_1 + p_2\Sigma_2 \\
\Sigma_B &= p_1p_2(\mu_1-\mu_2)(\mu_1-\mu_2)' \\
\Sigma &= \Sigma_W + \Sigma_B
\end{aligned}
$$

- Σ_W is called the Within Class Variance or Within Class Scatter
- Σ_B is called the Between Class Variance or Between Class Scatter

Projection



For hyperplane going through the origin, if $\| w \| = 1$, then $w'z$ is the signed length of the orthogonal projection of z onto w.

$$\begin{aligned} w'z &= w'(z_{||} + z_\perp) = w'z_{||} + w'z_\perp \\ &= w'z_{||} = w'(\pm || z_{||} || w) = \pm || z_{||} || \end{aligned}$$

$$w'z = \begin{cases} z_{||} & \text{if the angle } z \text{ makes with } w \text{ is less than } 90^\circ \\ -z_{||} & \text{if the angle } z \text{ makes with } w \text{ is more than } 90^\circ \end{cases}$$

Fisher Linear Discriminant

Let $Y = v'X$. Then

- $\mu_Y = v'\mu$
- $\Sigma_{YY} = v'\Sigma v$
- $\Sigma_{YB} = v'\Sigma_B v$
- $\Sigma_{YW} = v'\Sigma_W v$
- $J(v) = \frac{\Sigma_{YB}}{\Sigma_{YW}} = \frac{v'\Sigma_B v}{v'\Sigma_W v}$

Find v to maximize

$$J(v) = \frac{v'\Sigma_B v}{v'\Sigma_W v}$$

Fisher Linear Discriminant

$$
\begin{aligned}
J(v) &= \frac{v'\Sigma_B v}{v'\Sigma_W v} \\
\frac{\partial}{\partial v} J(v) &= \frac{v'\Sigma_W v \times 2\Sigma_B v - v'\Sigma_B v \times 2\Sigma_W v}{(v'\Sigma_W v)^2} \\
0 &= \frac{v'\Sigma_W v \times 2\Sigma_B v - v'\Sigma_B v \times 2\Sigma_W v}{(v'\Sigma_W v)^2} \\
&= \frac{2\Sigma_B v}{v'\Sigma_W v} - \frac{v'\Sigma_B v \times 2\Sigma_W v}{(v'\Sigma_W v)^2} \\
\frac{v'\Sigma_B v \times 2\Sigma_W v}{(v'\Sigma_W v)^2} &= \frac{2\Sigma_B v}{v'\Sigma_W v} \\
\frac{v'\Sigma_B v}{v'\Sigma_W v}\Sigma_W v &= \Sigma_B v = p_1 p_2(\mu_1 - \mu_2)(\mu_1 - \mu_2)' v \\
\frac{[v' p_1 p_2(\mu_1 - \mu_2)(\mu_1 - \mu_2)' v]\Sigma_w v}{v'\Sigma_w v} &= p_1 p_2(\mu_1 - \mu_2)(\mu_1 - \mu_2)' v \\
\frac{[v'(\mu_1 - \mu_2)(\mu_1 - \mu_2)' v]\Sigma_w v}{v'\Sigma_w v} &= (\mu_1 - \mu_2)[(\mu_1 - \mu_2)' v] \\
\frac{[v'(\mu_1 - \mu_2)(\mu_1 - \mu_2)' v] v}{v'\Sigma_w v} &= \Sigma_w^{-1}(\mu_1 - \mu_2)[(\mu_1 - \mu_2)' v]
\end{aligned}
$$

Fisher Linear Discriminant

$$
\begin{aligned}
\frac{[v'(\mu_1-\mu_2)(\mu_1-\mu_2)'v]v}{v'\Sigma_w v} &= \Sigma_w^{-1}(\mu_1-\mu_2)[(\mu_1-\mu_2)'v] \\
\frac{v'(\mu_1-\mu_2)(\mu_1-\mu_2)'v}{v'\Sigma_W v}v &= (\mu_1-\mu_2)'v\Sigma_W^{-1}(\mu_1-\mu_2) \\
\frac{[v'(\mu_1-\mu_2)][(\mu_1-\mu_2)'v]}{v'\Sigma_W v}v &= [(\mu_1-\mu_2)'v]\Sigma_W^{-1}(\mu_1-\mu_2) \\
\frac{v'(\mu_1-\mu_2)}{v'\Sigma_W v}v &= \Sigma_W^{-1}(\mu_1-\mu_2)
\end{aligned}
$$

Fisher Linear Discriminant

$$\frac{v'(\mu_1 - \mu_2)}{v'\Sigma_W v} v = \Sigma_W^{-1}(\mu_1 - \mu_2)$$

Now examine the left hand side under the condition that $v = \Sigma_W^{-1}(\mu_1 - \mu_2)$

$$\begin{aligned}\frac{v'(\mu_1 - \mu_2)}{v'\Sigma_W v} v &= \frac{(\mu_1 - \mu_2)'\Sigma_W^{-1}(\mu_1 - \mu_2)}{(\mu_1 - \mu_2)'\Sigma_W^{-1}\Sigma_W\Sigma_W^{-1}(\mu_1 - \mu_2)}\Sigma_W^{-1}(\mu_1 - \mu_2) \\ &= \Sigma_W^{-1}(\mu_1 - \mu_2)\end{aligned}$$

Therefore,

$$v = \Sigma_W^{-1}(\mu_1 - \mu_2)$$

Fisher Linear Discriminant

We now look at the problem from the point of view of the projection.
Let $Y = v'X$. Then

- $\mu_Y = v'\mu$
- $\Sigma_{YY} = v'\Sigma v$
- $\Sigma_{YB} = v'\Sigma_B v$
- $\Sigma_{YW} = v'\Sigma_W v$
- $J(v) = \frac{\Sigma_{YB}}{\Sigma_{YW}} = \frac{v'\Sigma_B v}{v'\Sigma_W v}$

Find v to maximize

$$J(v) = \frac{v'\Sigma_B v}{v'\Sigma_W v}$$

Find v to maximize $v'\Sigma_B v$ subject to the constraint $v'\Sigma_W v = 1$

Fisher Linear Discriminant

$$\begin{aligned}
\epsilon^2(v) &= v'\Sigma_B v - \lambda(v'\Sigma_W v - 1) \\
\frac{\partial \epsilon^2(v)}{\partial v} &= 2\Sigma_B v - \lambda 2\Sigma_W v \\
0 &= 2\Sigma_B v - \lambda 2\Sigma_W v \\
\Sigma_B v &= \lambda \Sigma_W v \\
\Sigma_W^{-1}\Sigma_B v &= \lambda v
\end{aligned}$$

v is the eigenvector of $\Sigma_W^{-1}\Sigma_B$ having non-zero eigenvalue
Plug in $\Sigma_B = p_1 p_2(\mu_1 - \mu_2)(\mu_1 - \mu_2)'$ and $v = \Sigma_W^{-1}(\mu_1 - \mu_2)$. Then

$$\begin{aligned}
\Sigma_W^{-1}\Sigma_B v &= \Sigma_W^{-1} p_1 p_2(\mu_1 - \mu_2)(\mu_1 - \mu_2)' v \\
&= \Sigma_W^{-1} p_1 p_2(\mu_1 - \mu_2)(\mu_1 - \mu_2)'\Sigma_W^{-1}(\mu_1 - \mu_2) \\
&= [p_1 p_2(\mu_1 - \mu_2)'\Sigma_W^{-1}(\mu_1 - \mu_2)]\Sigma_W^{-1}(\mu_1 - \mu_2) \\
&= [p_1 p_2(\mu_1 - \mu_2)'\Sigma_W^{-1}(\mu_1 - \mu_2)] v = \lambda v
\end{aligned}$$

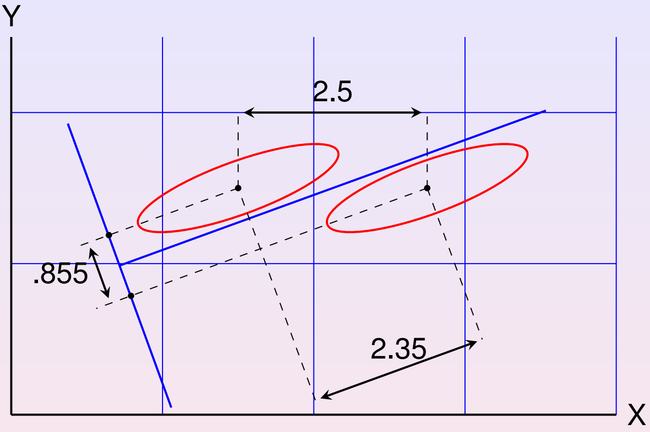


Figure: Shows two ellipsoidally symmetric distributions. The direction in which their means are furthest apart is not the direction to project them.

Invariance Under Linear Transformation

$$v = S_W^{-1}(\mu_1 - \mu_2)$$

$$\begin{aligned} v'x &= [S_W^{-1}(\mu_1 - \mu_2)]'x \\ &= (\mu_1 - \mu_2)'S_W^{-1}x \end{aligned}$$

Let A be any non-singular matrix and $y = Ax$. Then,

$$\begin{aligned} \mu_{1y} &= A\mu_1 \\ \mu_{2y} &= A\mu_2 \\ S_{Wy} &= AS_WA' \\ S_{Wy}^{-1} &= (AS_WA')^{-1} = (A')^{-1}S_W^{-1}A^{-1} \\ u &= S_{Wy}^{-1}(\mu_{1y} - \mu_{2y}) \\ u'y &= (\mu_{1y} - \mu_{2y})'S_{Wy}^{-1}y \\ &= (A\mu_1 - A\mu_2)'(A')^{-1}S_W^{-1}A^{-1}Ax \\ &= (\mu_1 - \mu_2)'A'(A')^{-1}S_W^{-1}A^{-1}Ax \\ &= (\mu_1 - \mu_2)'S_W^{-1}x = v'x \end{aligned}$$

With Samples

- Data Set: $<(x_1, k_1), \ldots, (x_N, k_N)>$
- $x_n \in R^D$ is a data vector
- $k_n \in \{1, 2\}$ is the class label of x_n
- $C_1 = \{n \mid k_n = 1\}$
- $C_2 = \{n \mid k_n = 2\}$
- $N_1 = |C_1|$
- $N_2 = |C_2|$
- $N = N_1 + N_2$

$$
\begin{aligned}
\hat{\mu} &= \frac{1}{N}\sum_{n=1}^{M} x_n \\
S &= \sum_{n=1}^{N} (x_n - \hat{\mu})(x_n - \hat{\mu})'
\end{aligned}
$$

With Samples

$$
\begin{aligned}
\hat{\mu}_1 &= \frac{1}{N_1} \sum_{n \in C_1} x_n \\
\hat{\mu}_2 &= \frac{1}{N_2} \sum_{n \in C_2} x_n \\
S_B &= \frac{N_1 N_2}{N} (\hat{\mu}_1 - \hat{\mu}_2)(\hat{\mu}_1 - \hat{\mu}_2)' \\
S_1 &= \sum_{n \in C_1} (x_n - \hat{\mu}_1)(x_n - \hat{\mu}_1)' \\
S_2 &= \sum_{n \in C_2} (x_n - \hat{\mu}_2)(x_n - \hat{\mu}_2)' \\
S_W &= S_1 + S_2 \\
S &= S_B + S_W
\end{aligned}
$$

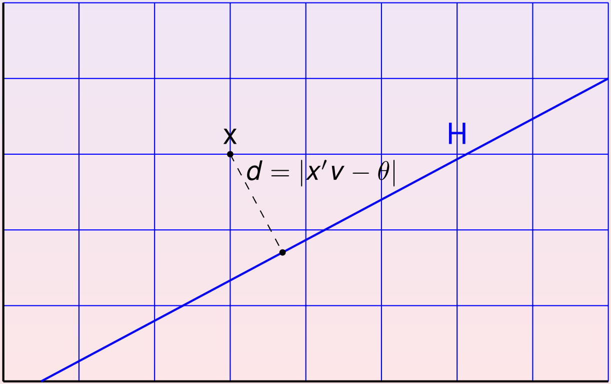
$$
v = S_W^{-1} (\hat{\mu}_1 - \hat{\mu}_2)
$$

The Geometry Of The Hyperplane

Suppose $||v|| = 1$. Define the Hyperplane H by

$$H = \{y \mid v'y = \theta\}$$

Consider determining the distance of x to the hyperplane H.



The Geometry Of The Hyperplane

Given x, determine y to minimize $||y-x||^2$ subject to the constraint that $v'y=\theta$. Define

$$f(y)=(y-x)'(y-x)+\lambda(v'y-\theta)$$

$$\begin{aligned}
0=\frac{\partial}{\partial y}f(y) &= 2(y-x)+\lambda v \\
y-x &= -\frac{1}{2}\lambda v \\
y &= x-\frac{1}{2}\lambda v \\
(y-x)'(y-x) &= \frac{1}{4}\lambda^2 v'v=\frac{1}{4}\lambda^2 \\
0=v'y-\theta &= v'(x-\frac{1}{2}\lambda v)-\theta \\
0 &= v'x-\frac{1}{2}\lambda v'v-\theta \\
\frac{1}{2}\lambda &= v'x-\theta \\
\lambda &= 2(v'x-\theta) \\
(y-x)'(y-x) &= \frac{1}{4}4(v'x-\theta)^2=(v'x-\theta)^2
\end{aligned}$$

Hyperplane

$$H = \left\{x \in \mathbb{R}^N \mid \sum_{n=1}^{N} x_n w_n = \theta\right\} = \{x \in \mathbb{R}^N \mid x' w = \theta\}$$

Proposition

Let $\{b_1, \ldots, b_{N-1}, w\}$ be an orthogonal basis for $\mathbb{R}^N$ and $x_0 \in H$. Then,

$$\begin{aligned} H &= \{x \in \mathbb{R}^N \mid \text{for some } \alpha_n \in R, n = 1, \ldots, N-1, \\ &\qquad x = x_0 + \sum_{n=1}^{N-1} \alpha_n b_n\} \end{aligned}$$

Proof.

$$(x_0 + \sum_{n=1}^{N-1} \alpha_n b_n)' w = x_0' w + \sum_{n=1}^{N-1} \alpha_n b_n' w = \theta$$

□

Hyperplane

Proposition

Let

$$H = \{x \in R^N \mid x' w = \theta\}$$

Let $x, x_0 \in H$. *Then* w *is normal to* $x - x_0$: $w'(x - x_0) = 0$

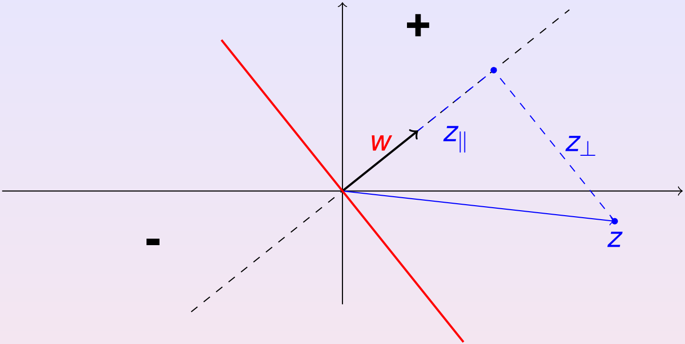
Proof.

Consider $w'(x - x_0)$.

$$w'(x - x_0) = w'x - w'x_0$$

Since both x *and* x_0 *are in H,* $w'x = \theta$ *and* $w'x_0 = \theta$. *Hence,* $w'(x - x_0)$*=0* $\square$

Projection

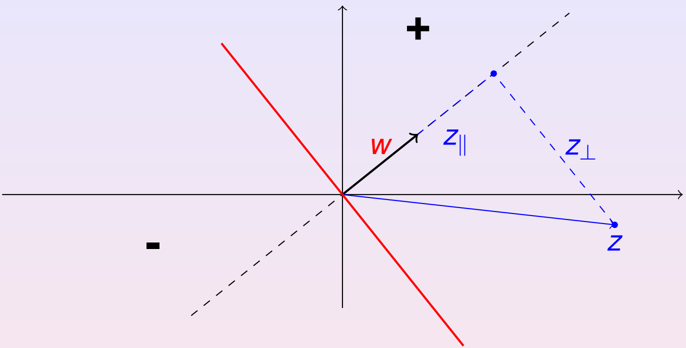


For a hyperplane going through the origin, if $\| w \| = 1$, then $w'z$ is the signed length of the orthogonal projection of z onto v.

$$
\begin{aligned}
w'z &= w'(z_{\|} + z_{\perp}) = w'z_{\|} + w'z_{\perp} \\
&= w'z_{\|} = w'(\pm \| z_{\|} \| w) = \pm \| z_{\|} \|
\end{aligned}
$$

$$
w'z = \begin{cases} z_{\|} & \text{if the angle } z \text{ makes with } w \text{ is less than } 90^\circ \\ -z_{\|} & \text{if the angle } z \text{ makes with } w \text{ is more than } 90^\circ \end{cases}
$$

Projection



$$sgn(w'z) = \begin{cases} +1 & \text{if } z \text{ is on the + side of the hyperplane} \\ -1 & \text{if } z \text{ is on the - side of the hyperplane} \end{cases}$$

Discrimination

Classify x as class 1 if

$$v'x \geq \theta$$

otherwise as class 2

Expected Gain vs Misdetect, False Alarm Rate

$P_M(\theta)$	Misdetect rate at θ
$P_F(\theta)$	False Alarm rate at θ
$P_T(c^1)$	Class c^1 prior probabilities
$P_T(c^2)$	Class c^2 prior probability

P_{TA}	Assigned		P_T
True	c^1	c^2	c
c^1	$P_T(c^1)(1-P_M(\theta))$	$P_T(c^1)P_M(\theta)$	$P_T(c^1)$
c^2	$P_T(c^2)P_F(\theta)$	$P_T(c^2)(1-P_F(\theta))$	$P_T(c^2)$

e	Assigned	
True	c^1	c^2
c^1	α	β
c^2	γ	δ

$$\begin{aligned} E[e] &= \alpha P_T(c^1)[1-P_M(\theta)] + \beta P_T(c^1)P_M(\theta) + \\ &\quad \gamma P_T(c^2)P_F(\theta) + \delta P_T(c^2)[1-P_F(\theta)] \\ &= -P_M(\theta)[P_T(c^1)(\alpha-\beta)] - P_F(\theta)[P_T(c^2)(\delta-\gamma)] + \\ &\quad \alpha P_T(c^1) + \delta P_T(c^2) \end{aligned}$$

Example

Economic Gain

True	c^1	c^2	Prior Prob
c^1	2	-3	.6
c^2	-4	4	.4

P_F	P_M	Gain
0.000	1.000	-.200
0.100	0.790	.109
0.200	0.619	.304
0.300	0.478	.405
0.400	0.363	.431
0.500	0.269	.399
0.600	0.192	.305
0.700	0.129	.174
0.800	0.077	.009
0.900	0.035	-.184
1.000	0.000	-.400

Example

Economic Gain

True	c^1	c^2
c^1	2	-3
c^2	-4	4

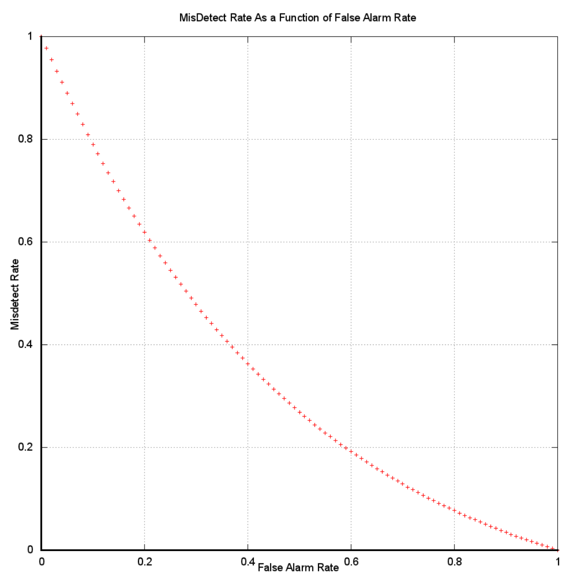
$$\begin{aligned} P_F &= .500 \\ P_M &= .269 \\ P_T(c^1)(1-P_M) &= .6(1-.269) = .4386 \\ P_T(c^1)P_M &= .6(.269) = .1614 \end{aligned}$$

Confusion Matrix

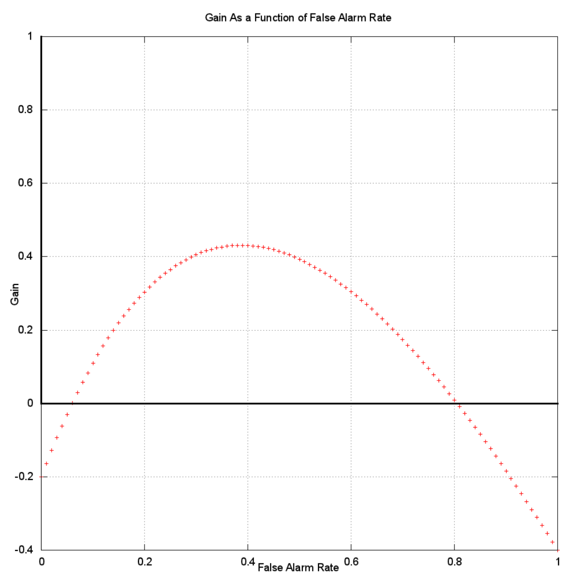
True	c^1	c^2	Prior Prob
c^1	.4386	.1614	.6
c^2	.2	.2	.4

$$G = .4386 \times 2 + .1614 \times (-3) - 4 \times .2 + 4 \times .2 = .393$$

Misdetect and False Alarm Rate



Gain and False Alarm Rate



The Naive Bayes Classifier

Definition

The Naive Bayes approach makes the assumption that the economic gain is the identity matrix, the class prior probabilities are all equal and conditioned on class, the joint probabilities are the product of the first order marginals.

$$P(x_1, \ldots, x_N \mid c) = P(x_1 \mid c)P(x_2 \mid c) \cdots P(x_N \mid c)$$

The Naive Bayes

With the identity gain matrix and equal class priors, the Naive Bayes classifier assigns a measurement tuple $(x_1, \ldots, x_N)$ to class c^*, satisfying

$$P(x_1 \mid c^*)P(x_2 \mid c^*) \cdots P(x_N \mid c^*) > P(x_1 \mid c)P(x_2 \mid c) \cdots P(x_N \mid c)$$

for any other class c.

The Naive Bayes Classifier: General Case

Taking logs on both sides of the inequality produces

$$\sum_{n=1}^{N} \log P(x_n \mid c^*) > \sum_{n=1}^{N} \log P(x_n \mid c)$$

for any other c.

The Naive Bayes Classifier: Binary Variates

In the case of binary variates, let

$$w_{nc} = P(x_n = 1 \mid c)$$

then, $P(x_n \mid c)$ can be written as

$$P(x_n \mid c) = w_{nc}^{x_n}(1 - w_{nc})^{1-x_n}$$

Hence,

$$\begin{aligned} \log P(x_n \mid c) &= \log w_{nc}^{x_n}(1 - w_{nc})^{1-x_n} \\ &= x_n log w_{nc} + (1 - x_n)\log(1 - w_{nc}) \end{aligned}$$

Letting

$$
\begin{aligned}
a_{nc} &= \log w_{nc} \\
b_{nc} &= \log(1 - w_{nc})
\end{aligned}
$$

then $(x_1, \ldots, x_N)$ is assigned to class c^* when

$$\sum_{n=1}^{N} a_{nc^*} x_n + b_{nc^*}(1 - x_n) > \sum_{n=1}^{N} a_{nc} x_n + b_{nc}(1 - x_n)$$

for any other class c

The Naive Bayes Classifier: Binary Variates

Collecting terms, assign $(x_1, \ldots, x_N)$ to class c^* when

$$\sum_{n=1}^{N}(a_{nc^*} - b_{nc^*})x_n + b_{nc^*} > \sum_{n=1}^{N}(a_{nc} - b_{nc})x_n + b_{nc}$$

for any other class c

The Naive Bayes Classifier
For Binary Variables is a Linear Classifier

The Brain Is A Mechanism

- The essential features of the brain can be derived in principle from a knowledge of the connections and states of the neurons which comprise it
- The information-handling capabilities of biological networks do not depend upon any specifically vitalistic powers which could not be duplicated by man-made devices

Frank Rosenblatt *Principles of Neurodynamics*, 1962

Network

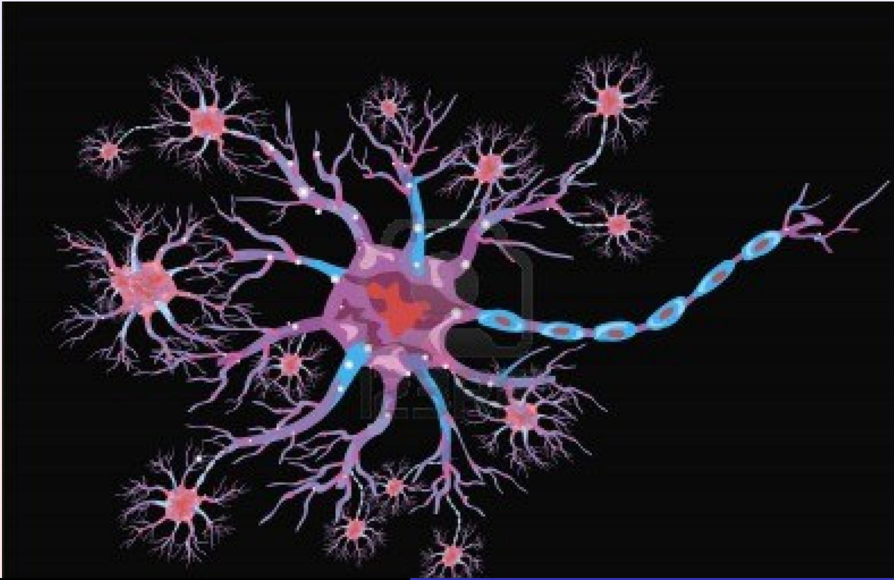
- Individual elements, or cells, of a neural network have never been demonstrated to possess any specifically psychological functions, such as
  - Memory
  - Awareness or
  - Intelligence
- Such properties reside in the organization and function of the network as a whole

Frank Rosenblatt *Principles of Neurodynamics*, 1962

The Neuron

- The nervous system consists of a network of neurons
- Each neuron has a cell body with one or more
  - Dendrites, communicating afferent (incoming) signals
  - Axons, communicating efferent (outgoing) signals

The Neuron

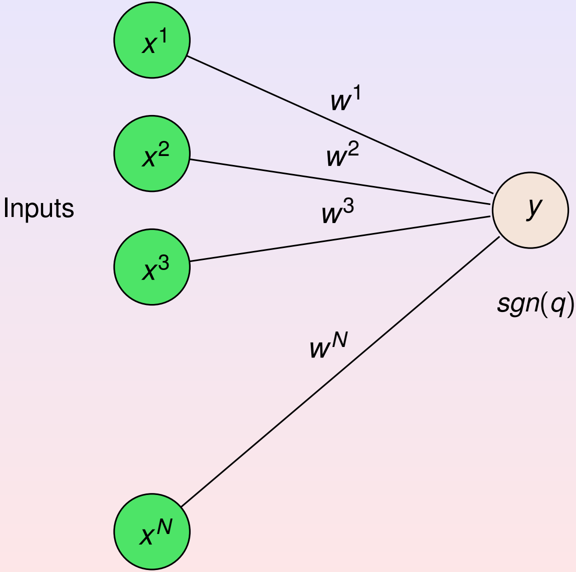


McCulloch and Pitts

- The activity of the neuron is an all-or none process
- A certain fixed number of synapses must be excited within the period of latent addition in order to excite a neuron at any time, and this number is independent of previous activity and position on the neuron
- The only significant delay within the nervous system is synaptic delay
- The activity of any inhibitory synapse absolutely prevents excitation of the neuron at that time
- The structure of the net does not change with time

McCulloch and Pitts, *A Logical Calculus of the Ideas Immanent in Nervous Activity*, 1943

Threshold Logic Unit



$$y = sgn\left(\sum_{n=1}^{N} w^n x^n\right)$$

$$sgn(q) = \begin{cases} 1 & \text{when } q \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

The Perceptron: Frank Rosenblatt

Let $x_1, \ldots, x_Z \in R^N$ be the training data N-dimensional vectors.

$$x_z = \begin{pmatrix} x_z^1 \\ x_z^2 \\ \vdots \\ x_z^{N-1} \\ 1 \end{pmatrix}$$

The last component is always 1 so that the threshold θ is automatically included in the weight vector $w = (w^1, \ldots, w^N)'$.

Perceptron

Assign class c^1 to vector x if

$$w'x > 0$$

else assign class c^2

Associate class c^1 with -1 and class c^2 with $+1$.

Assign class y to vector x where

$$y = sgn(w'x)$$

The Perceptron: The Cost Function

Let $c_1, \ldots, c_Z \in \{-1, 1\}$ specify the corresponding classes.

$< (x_1, c_1), \ldots, (x_Z, c_Z) >$ is the training data.

For any w, define the set of error indexes

$$M(w) = \{z \mid sgn(w'x_z) \neq c_z\}$$

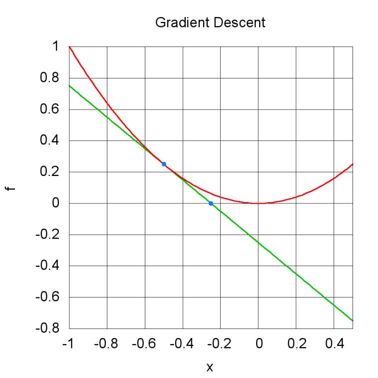
$w'x_z$ has the opposite sign of c_z

Cost Function

$$J(w) = \sum_{z \in M(w)} -c_z w' x_z$$

Notice that $J(x)$ is always positive.
Find the w to minimize $J(w)$

Gradient Descent



Find w to minimize $f(w)$

$$\begin{aligned} y &= f(w) \qquad \frac{\partial y}{\partial w} = f'(w) \\ w_t &= -.5 \qquad \delta > 0 \\ w_{t+1} &= w_t - \delta f'(w_t) \end{aligned}$$

Cost Function

$$
\begin{aligned}
J(w) &= \sum_{z \in M(w)} -c_z w' x_z \\
\frac{\partial}{\partial w} J(w) &= \sum_{z \in M(w)} -c_z x_z \\
w_{new} &= w_{old} + \delta \sum_{z \in M(w)} c_z x_z
\end{aligned}
$$

Iterate

Iteratively changes the weight vector to make it produce the correct class for each training vector.

Weight vector at iteration t.

$$w(t) = \begin{pmatrix} w^1(t) \\ w^2(t) \\ \vdots \\ w^N(t) \end{pmatrix}$$

The Perceptron

Iteratively changes the weight vector to make it produce the correct class for each training vector.

- $w^n(0)$, $n = 1, \ldots, N$ set at random
- t is the iteration index
- (t) as a subscript means $t \mod Z$
- δ a small positive constant

$$
\begin{aligned}
y(t) &= sgn(w(t)' x_{(t)}), \text{ the assigned class} \\
w(t+1) &= w(t) + \delta[c_{(t)} - y(t)] x_{(t)} \\
w(t+1) &= \begin{cases} w(t) & \text{if } c_{(t)} = y(t) \\ w(t) + 2\delta x_{(t)} & \text{if } c_{(t)} = 1 \text{ and } y(t) = -1 \\ w(t) - 2\delta x_{(t)} & \text{if } c_{(t)} = -1 \text{ and } y(t) = 1 \end{cases}
\end{aligned}
$$

Will converge in a finite number of steps if the vectors of one class are linearly separable from the vectors in the other class.

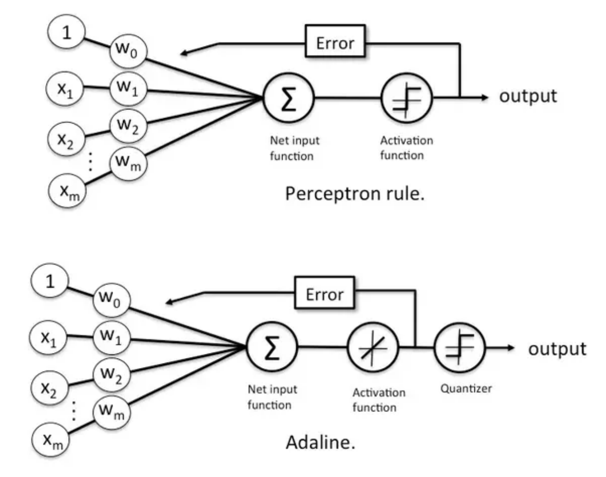
Pocket Algorithm for Non-linearly Separable Classes

Keep the best weight vector in the pocket.
At the end of the iterations report the weight vector in the pocket.

- Initialize $w(0) = w_s$ randomly
- h_s is history counter for w_s
- Initialize $h_s = 0$
- t^{th} iteration
  - Update $w(t+1)$ using the perceptron update rule
  - Use $w(t+1)$ to determine the number h of training vectors that are correctly classified
  - If $h > h_s$, define
    - $w_s = w(t+1)$
    - $h_s = h$

Final weight vector is w_s

Perceptron and Adaline



Gradient Descent for Adaline

$$
\begin{aligned}
e(w) &= \frac{1}{2}\sum_{z=1}^{Z}(c_z - w'x_z)^2 \\
\frac{\partial e}{\partial w} &= \sum_{z=1}^{Z}(c_z - w'x_z)(-x_z) \\
\delta_t &> 0 \\
w(t+1)^{N\times 1} &= w(t)^{N\times 1} - \delta_t[\frac{\partial e}{\partial w}(w(t))]^{N\times 1} \\
&= w(t) + \delta_t \sum_{z=1}^{Z}[c_z - w(t)'x_z]x_z
\end{aligned}
$$

Adaline

Widrow and Hopf, 1960

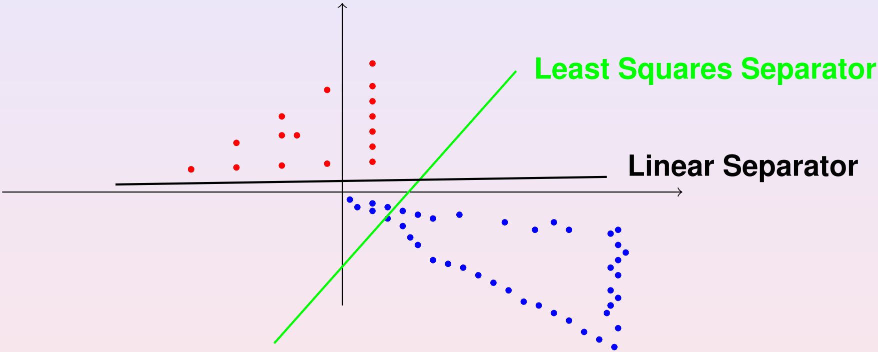
$$
\begin{aligned}
y(t) &= w(t)' x_{(t)} \\
w(t+1) &= w(t) + \delta(t)[c_{(t)} - y(t)] x_{(t)} \\
\sum_{t=0}^{\infty} \delta(t) &= \infty \\
\sum_{t=0}^{\infty} \delta^2(t) &< \infty
\end{aligned}
$$

w is the least squares solution minimizing:

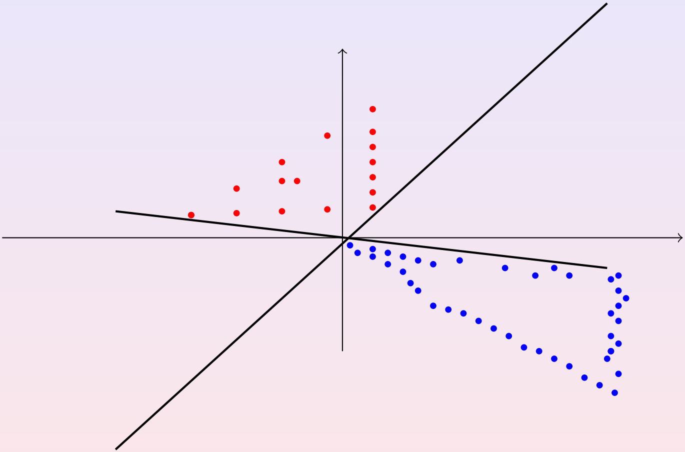
$$\sum_{z=1}^{Z} (w' x_z - c_z)^2 = \| Xw - c \|^2$$

where the rows of X are $x_1', \ldots, x_Z'$ and the rows of c are $c_1, \ldots, c_Z$.

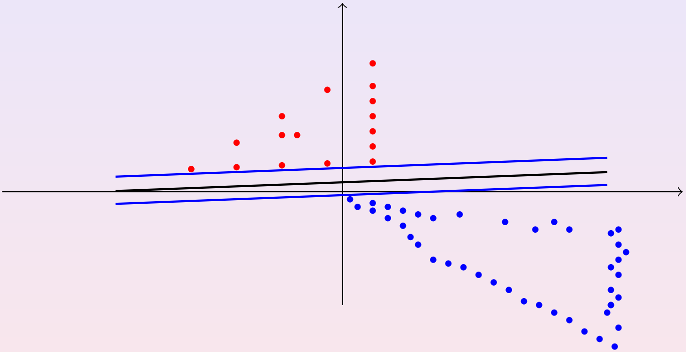
Least Squares and Linear Separator



Linear Separators



Largest Margin Linear Separators



Maximal Margin Hyperplane Separator

Distance of point x to hyperplane whose normal is w is

$$\frac{|w'x - \theta|}{||w||}$$

$$\max_{w} C$$

subject to

$$\frac{c_z(w'x_z - \theta)}{||w||} \geq C, \; z = 1, \ldots, Z$$

Maximal Margin Hyperplane Separator

Since for any w satisfying the inequalities, any positively scaled multiple will also satisfy the inequalities, we can arbitrarily set $||w|| = \frac{1}{C}$ Our optimization problem then is equivalent to

$$\min_{w} \frac{1}{2} ||w||^2$$

subject to

$$c_z(w'x_z - \theta) \geq 1, \; z = 1, \ldots, Z$$

Note that the closest points x_z to the hyperplane will satisfy $c_z(w'x_z - \theta) = 1$
In this case

$$\frac{c_z(w'x_z - \theta)}{||w||} = \frac{1}{||w||}$$

is the distance of the point x_z to the hyperplane

Lagrange Multiplier Formulation

$$\min_{w} \frac{1}{2}||w||^2$$

subject to

$$c_z(w'x_z - \theta) \geq 1,\ z = 1, \ldots, Z$$

Define

$$f(w, \theta) = \frac{1}{2}||w||^2 - \sum_{z=1}^{Z} \alpha_z [c_z(w'x_z - \theta) - 1]$$

$$\begin{aligned}
\frac{\partial}{\partial w} f(w, \theta) &= w - \sum_{z=1}^{Z} \alpha_z [c_z x_z] \\
0 &= w - \sum_{z=1}^{Z} \alpha_z [c_z x_z] \\
w &= \sum_{z=1}^{Z} \alpha_z [c_z x_z]
\end{aligned}$$

Lagrange Formulation

$$
\begin{aligned}
f(w,\theta) &= \frac{1}{2}||w||^2 - \sum_{z=1}^{Z} \alpha_z [c_z(w' x_z - \theta) - 1] \\
\frac{\partial}{\partial\theta} f(w,\theta) &= \sum_{z=1}^{Z} \alpha_z c_z \\
0 &= \sum_{z=1}^{Z} \alpha_z c_z
\end{aligned}
$$

Support Vector Machine

The Lagrange multipliers can be either 0 or positive.
Define $S = \{z \mid \alpha_z > 0\}$
S is the set of indices of the support vectors.

$$w' x_z - \theta = \pm 1, z \in S$$

$$w = \sum_{z \in S} \alpha_z c_z x_z = \sum_{z=1}^{Z} \alpha_z c_z x_z$$

The largest margin classifier is known as the Support Vector Machine

Dual Form

$$
\begin{aligned}
f(w,\theta) &= \frac{1}{2}||w||^2 - \sum_{z=1}^{Z} \alpha_z [c_z(w' x_z - \theta) - 1] \\
w &= \sum_{z \in S} \alpha_z c_z x_z \\
0 &= \sum_{z \in S} \alpha_z c_z
\end{aligned}
$$

Dual Form

$$
\text{Minimize } L(\alpha_1, \ldots, \alpha_Z) = \sum_{z=1}^{Z} \alpha_z - \frac{1}{2} \sum_{i=1}^{Z} \sum_{j=1}^{Z} \alpha_i \alpha_j c_i c_j x_i' x_j
$$

subject to

$$
\begin{aligned}
\alpha_z &\geq 0,\ z = 1, \ldots, Z \\
\sum_{z=1}^{Z} \alpha_z c_z &= 0
\end{aligned}
$$

Dual Form

$$
\begin{aligned}
f(w,\theta) &= \frac{1}{2}||w||^2 - \sum_{z=1}^{Z} \alpha_z [c_z(w' x_z - \theta) - 1] \\
w &= \sum_{z=1}^{Z} \alpha_z c_z x_z \\
0 &= \sum_{z=1}^{Z} \alpha_z c_z \\
\frac{1}{2} w' w &= \frac{1}{2} \left(\sum_{i=1}^{Z} \alpha_i c_i x_i \right)' \sum_{j=1}^{Z} \alpha_j c_j x_j \\
&= \frac{1}{2} \sum_{i=1}^{Z} \sum_{j=1}^{Z} \alpha_i \alpha_j c_i c_j x_i' x_j
\end{aligned}
$$

Dual Form

$$
\begin{aligned}
0 &= \sum_{z=1}^{Z} \alpha_z c_z \\
w' x_z &= \left(\sum_{i=1}^{Z} \alpha_i c_i x_i \right)' x_z \\
\sum_{z=1}^{Z} \alpha_z [c_z (w' x_z - \theta) - 1] &= \sum_{z=1}^{Z} \alpha_z [c_z (\sum_{i=1}^{Z} \alpha_i c_i x_i' x_z - \theta) - 1] \\
&= \sum_{z=1}^{Z} \alpha_z c_z \sum_{i=1}^{Z} \alpha_i c_i x_i' x_z - \sum_{z=1}^{Z} \alpha_z c_z \theta - \sum_{z-1}^{Z} \alpha_z \\
&= \sum_{i=1}^{Z} \sum_{j=1}^{Z} \alpha_i \alpha_j c_i c_j x_i' x_j - \sum_{i=1}^{Z} \alpha_z
\end{aligned}
$$

Dual Form

$$
\begin{aligned}
f(w,\theta) &= \frac{1}{2}||w||^2 - \sum_{z=1}^{Z}\alpha_z[c_z(w'x_z-\theta)-1] \\
\frac{1}{2}w'w &= \frac{1}{2}\sum_{i=1}^{Z}\sum_{j=1}^{Z}\alpha_i\alpha_j c_i c_j x_i' x_j \\
\sum_{z=1}^{Z}\alpha_z[c_z(w'x_z-\theta)-1] &= \sum_{i=1}^{Z}\sum_{j=1}^{Z}\alpha_i\alpha_j c_i c_j x_i' x_j - \sum_{i=1}^{Z}\alpha_z
\end{aligned}
$$

$$
\begin{aligned}
f(\alpha_1,\ldots,\alpha_Z) &= \frac{1}{2}\sum_{i=1}^{Z}\sum_{j=1}^{Z}\alpha_i\alpha_j c_i c_j x_i' x_j - \left\{\sum_{i=1}^{Z}\sum_{j=1}^{Z}\alpha_i\alpha_j c_i c_j x_i' x_j - \sum_{i=1}^{Z}\alpha_z\right\} \\
&= \sum_{i=1}^{Z}\alpha_z - \frac{1}{2}\sum_{i=1}^{Z}\sum_{j=1}^{Z}\alpha_i\alpha_j c_i c_j x_i' x_j
\end{aligned}
$$

Support Vector Machine

- $\alpha_z > 0$
  - $c_z(w'x_z - \theta) = 1$
  - x_z is on the boundary of the slab
  - $S = \{z \mid \alpha_z > 0\}$
- $\alpha_z = 0$
  - $c_z(w'x_z - \theta) > 1$
  - x_z is outside of the slab

$$\begin{aligned} c_z(w'x_z - \theta) &= 1, z \in S \\ w &= \sum_{z \in S} \alpha_z c_z x_z \end{aligned}$$

Support Vector Machine

$$c_z(w'x_z - \theta) = 1, z \in S$$

If $z \in S$ and $c_z = 1$

$$\begin{aligned} w'x_z - \theta &= 1 \\ \theta &= w'x_z - 1 \end{aligned}$$

If $z \in S$ and $c_z = -1$

$$\begin{aligned} w'x_z - \theta &= -1 \\ \theta &= w'x_z + 1 \end{aligned}$$

Support Vector Machine

For $z \in S$

$$\begin{aligned} c_z(w'x_z - \theta) &= 1 \\ w'x_z - \theta &= c_z \\ \theta &= w'x_z - c_z \end{aligned}$$

Therefore,

$$\theta = \frac{1}{|S|} \sum_{z \in S} (w'x_z - c_z)$$

Capacity

How many randomly chosen points in an N-dimensional space can be guaranteed to be linearly separable?

As N gets large Cover,1965, showed the answer is $2N$

Ripley, 1996, showed that the probability that Z points randomly chosen from any continuous distribution in R^N can be randomly divided into two groups that are linearly separable is approximately

$$\Phi\left(\frac{2N-Z}{\sqrt{Z}}\right)$$

$$\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}y^2} dy$$