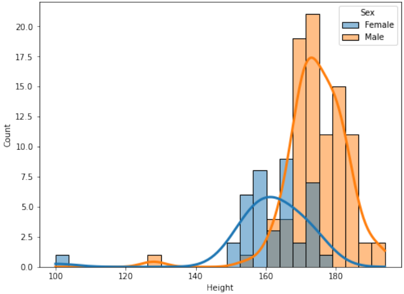
03

MAC 0460 / 5832
Introduction to Machine Learning

• decision surfaces • binary classification • perceptron •
• Caltech's Lecture 01 •

IME/USP

Recalling last class ...



$$p(y = F|x) = \frac{p(y = F)p(x|y = F)}{p(x)}$$

$$p(y = M|x) = \frac{p(y = M)p(x|y = M)}{p(x)}$$

$$p(y = F|x) \overset{?}{\lesseqgtr} p(y = M|x)$$

$$p(y = F)p(x|y = F) \overset{?}{\lesseqgtr} p(y = M)p(x|y = M)$$

Suppose two classes $y \in \{-1, +1\}$

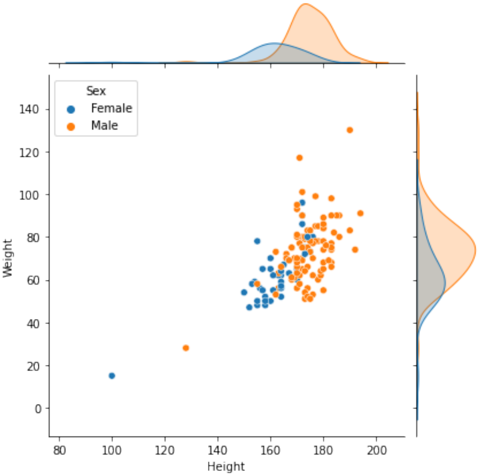
Suppose $p(y = -1) = p(y = +1) = 0.5$

Decision surface

Is the curve formed by the set of points $\mathbf{x}$ such that

$$p(y = -1)p(x|y = -1) = p(y = +1)p(x|y = +1)$$

$$P(\mathbf{x} \,|\, y = -1) = P(\mathbf{x} \,|\, y = +1)$$


Sex
Female
Male
140
120
100
80
60
40
20
0
Weight
80
100
120
140
160
180
200
Height

Example: When $\mathbf{x}$ follows a **Multivariate Gaussian distribution**

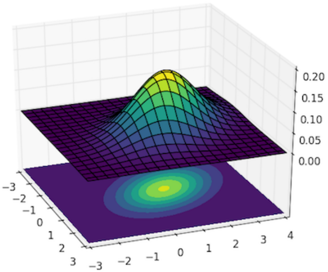
$N(\mu, \Sigma)$ (μ mean vector, Σ covariance matrix)

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left\{ -\frac{1}{2}(\mathbf{x}-\mu)^t\Sigma^{-1}(\mathbf{x}-\mu)\right\}, \quad \mathbf{x} \in \mathbb{R}^d$$

Covariance matrix Σ

$\sigma_{ii} = E[(x_i - \mu_i)^2]$ (variance of component x_i)

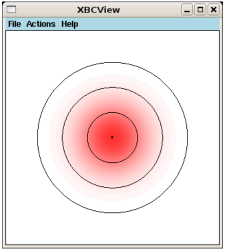
$\sigma_{ij} = E[(x_i - \mu_i)(x_j - \mu_j)]$, $i \neq j$, covariance between x_i and x_j

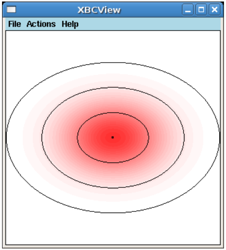


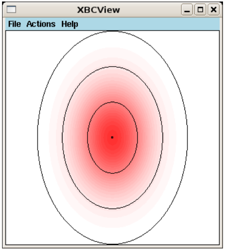
https://scipython.com/blog/visualizing-the-bivariate-gaussian-distribution/

2D Gaussian examples – contour lines

$$\Sigma = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} 4 & 0 \\ 0 & 2 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} 2 & 0 \\ 0 & 4 \end{bmatrix}$$

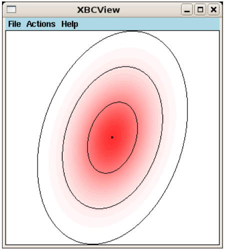




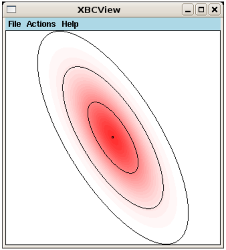


2D Gaussian examples – contour lines

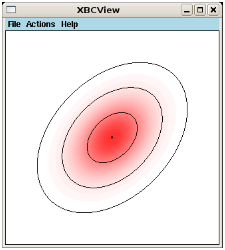
$$\Sigma = \begin{bmatrix} 2 & .75 \\ .75 & 2 \end{bmatrix}$$



$$\Sigma = \begin{bmatrix} 2 & .75 \\ .75 & 4 \end{bmatrix}$$

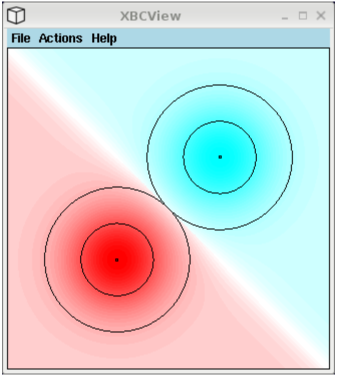


$$\Sigma = \begin{bmatrix} 2 & -2 \\ -2 & 4 \end{bmatrix}$$



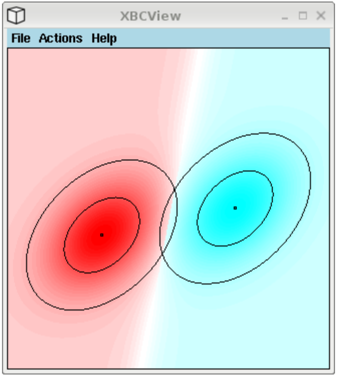
Decision surface

$\Sigma_j = \sigma^2 I$: both classes have the same covariance matrix, null covariance



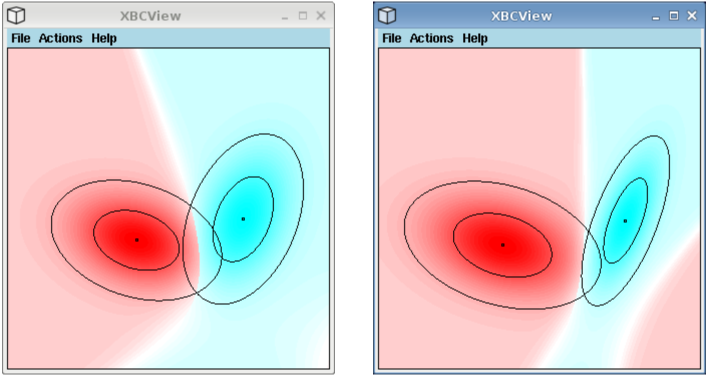
Decision surface

$\Sigma_j = \Sigma$: both classes have the same covariance matrix



Decision surface

Σ_j arbitrary: classes have distinct covariance matrix



In binary classification, if the two classes have Gaussian distribution with the same covariance matrix, then the optimal decision surface is a hyperplane

Perceptron

Find a separating hyperplane when classes are linearly separable

Input: $\mathbf{x} = (x_1, x_2, \ldots, x_d) \in \mathbb{R}^d$

Output: $y \in \{-1, +1\}$

Hipothesis:

$$h(\mathbf{x}) = \mathrm{sign}\Big(\big(\sum_{i=1}^{d} w_i x_i\big) + b\Big),\ b \in \mathbb{R}, w_i \in \mathbb{R}, i = 1, 2, \ldots, d$$

Artificial component, just to simplify notatiton:

$$\mathbf{x} = (1, x_1, x_2, \ldots, x_d) \in \mathbb{R}^{d+1}$$
$$\mathbf{w} = (w_0, w_1, w_2, \ldots, w_d) \in \mathbb{R}^{d+1}$$

Thus

$$h(\mathbf{x}) = \mathrm{sign}\big(\mathbf{w}^T \mathbf{x}\big)$$

Input: $\mathbf{x} = (x_1, x_2, \ldots, x_d) \in \mathbb{R}^d$

Output: $y \in \{-1, +1\}$

Hipothesis:

$$h(\mathbf{x}) = \mathrm{sign}\Big(\big(\sum_{i=1}^{d} w_i x_i \big) + b \Big), \; b \in \mathbb{R}, w_i \in \mathbb{R}, i = 1, 2, \ldots, d$$

Artificial component, just to simplify notatiton:

$$\mathbf{x} = (1, x_1, x_2, \ldots, x_d) \in \mathbb{R}^{d+1}$$
$$\mathbf{w} = (w_0, w_1, w_2, \ldots, w_d) \in \mathbb{R}^{d+1}$$

Thus

$$h(\mathbf{x}) = \mathrm{sign}\big(\mathbf{w}^T \mathbf{x}\big)$$

($\mathbf{w}^T \mathbf{x} = 0$ defines a hyperplane)

Perceptron Algorithm

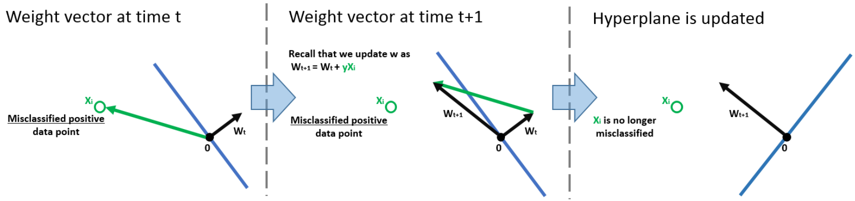
Let $\mathbf{w}$ be the **"current" weight**
Let $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)}), i = 1, \ldots, N\}$ be the **training set**

Repeat

1. Let $(\mathbf{x}^{(i)}, y^{(i)}) \in \mathcal{D}$ such that $\mathrm{sign}(\mathbf{w}^T \mathbf{x}^{(i)}) \neq y^{(i)}$
   If there is no such pair, then stop.
2. Update current weight as follows:

$$\mathbf{w} \leftarrow \mathbf{w} + y^{(i)}\mathbf{x}^{(i)} \qquad\qquad y^{(i)} \in \{-1, +1\}$$

Return $\mathbf{w}$



http://www.cs.cornell.edu/courses/cs4780/2015fa/web/lecturenotes/lecturenote03.html

A simple learning algorithm - PLA

The perceptron implements

$$h(\mathbf{x}) = \text{sign}(\mathbf{w}^{\intercal}\mathbf{x})$$

Given the training set:

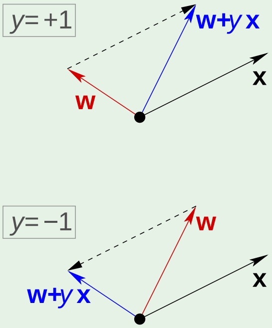
$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_N, y_N)$$

pick a misclassified point:

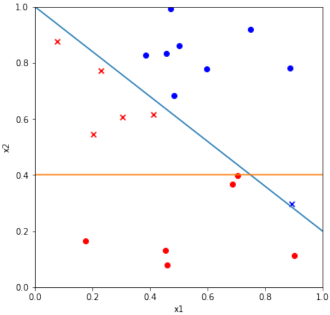
$$\text{sign}(\mathbf{w}^{\intercal}\mathbf{x}_n) \neq y_n$$

and update the weight vector:

$$\boxed{\mathbf{w} \leftarrow \mathbf{w} + y_n\mathbf{x}_n}$$



Perceptron - step by step example

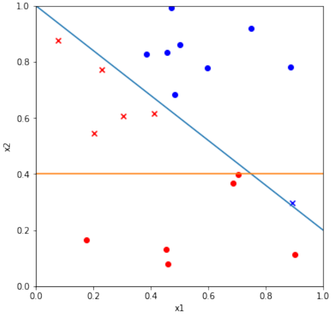


Target
$f(x_1, x_2) = -1 + 0.8x_1 + x2$

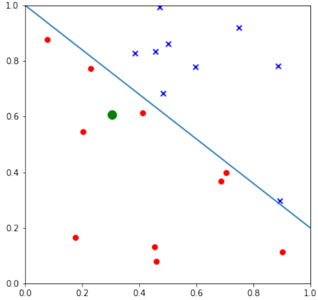
Starting hypothesis
$h_0(x_1, x_2) = -0.4 + x_2$

Performance of h_0:

- ● true positive $\surd$
- ● true negative $\surd$
- $\times$ false negative X
- $\times$ false positive X



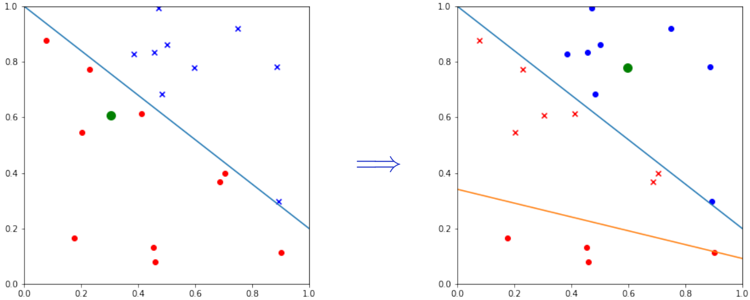
$\Longrightarrow$



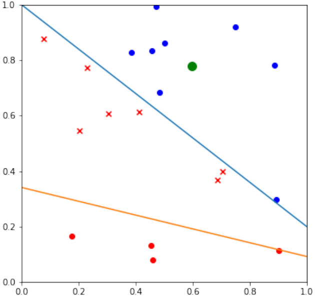
$$\mathbf{w}(0) = (-0.4,\ 0,\ 1) \xRightarrow{-(1, 0.3, 0.6)} \mathbf{w}(1) = (-1.4,\ -0.3,\ 0.4)$$

iteration 1

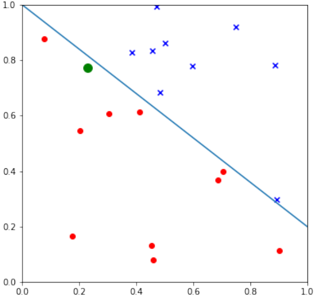
green dot indicates the example used to adjust the weight



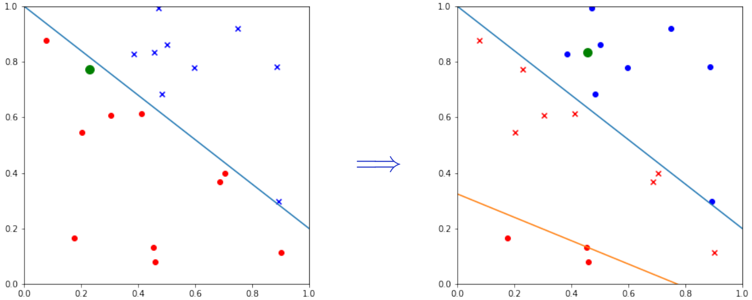
iteration 2



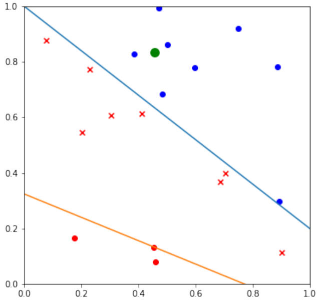
$\Longrightarrow$



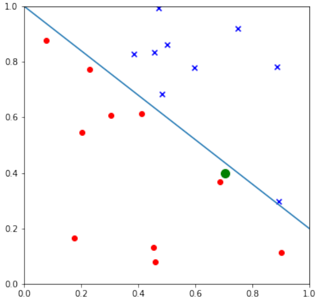
iteration 3



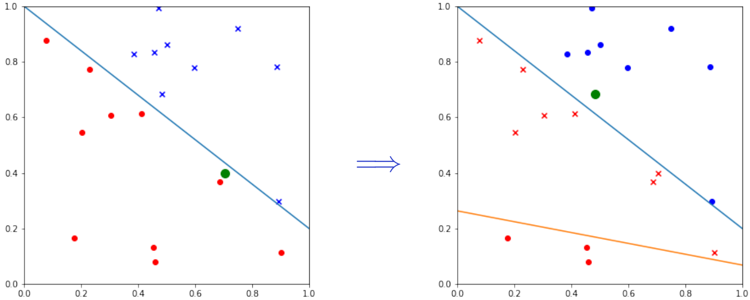
iteration 4



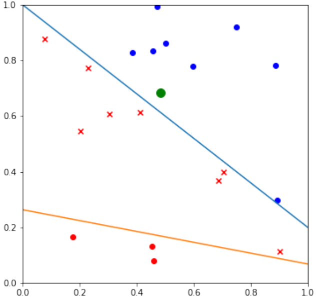
$\Longrightarrow$



iteration 5


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
$\Longrightarrow$
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

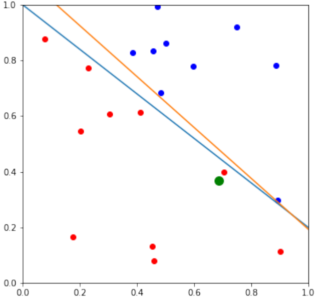
iteration 6



$\Longrightarrow \cdots$

iteration 7

$\Longrightarrow$



It ends after 77 iterations

Convergence of the Perceptron algorithm

Assumptions

The two classes are linearly separable: there exists a separating hyperplane with margin γ — there exists a weight vector $\mathbf{w}$ with norm 1 (i.e., $||\mathbf{w}|| = 1$) such that $y^{(i)}\mathbf{w}^T\mathbf{x}^{(i)} > \gamma$, $\forall i$.

Let also R be the maximum norm of the examples $\mathbf{x}^{(i)} \in \mathcal{D}$.

Let k denote the number of iterations of the algorithm.

Convergence of the Perceptron algorithm

The proof consists in showing that k is bounded by $\mathcal{O}(R^2/\gamma^2)$

(1) $\| \mathbf{w}^{k+1} \| > k\gamma$

(2) $\| \mathbf{w}^{k+1} \|^2 \leq kR^2$

(1)+(2) $k^2\gamma^2 < \| \mathbf{w}^{k+1} \|^2 \leq kR^2 \Longrightarrow k < \frac{R^2}{\gamma^2}$

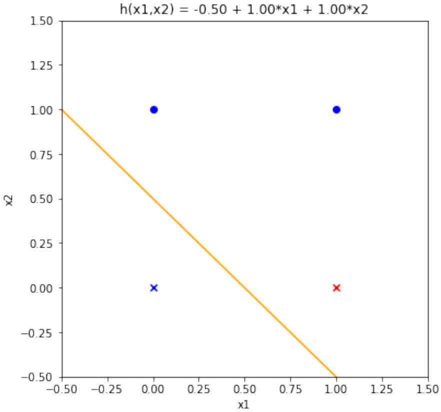
More details:

- https://www.cse.iitb.ac.in/~shivaram/teaching/old/cs344+386-s2017/resources/classnote-1.pdf
- Book: Marvin Minsky and Seymour Papert, Perceptrons

Homework

$\mathbf{w}(0) = (-0.5,\ 1,\ 1)$

x_1 x_2	y
0 0	1
0 1	1
1 0	-1
1 1	1



Process the examples cyclically, starting with 0 0

Answer: $(0.5,\ -1,\ 2)$ (though it may differ if you scan the examples in a different ordering)

Example: Predicting how a viewer will rate a movie

10% improvement = **1 million dollar prize**

The essence of machine learning:

- A pattern exists.
- We cannot pin it down mathematically.
- We have data on it.

Solution components

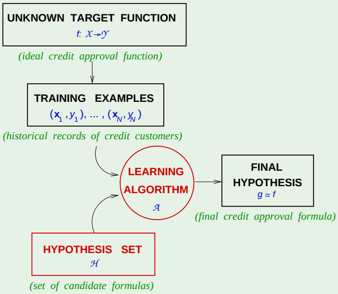
The 2 solution components of the learning problem:

- The Hypothesis Set

  $\mathcal{H} = \{h\} \qquad g \in \mathcal{H}$

- The Learning Algorithm

Together, they are referred to as the *learning model*.



Basic premise of learning

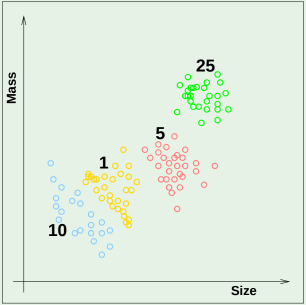
"using a set of observations to uncover an underlying process"

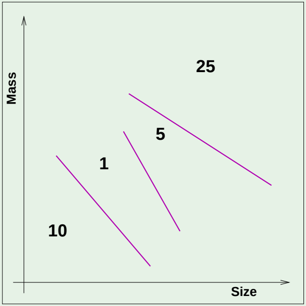
broad premise $\Longrightarrow$ many variations

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

Supervised learning

Example from vending machines – **coin recognition**





Unsupervised learning

Instead of (input,correct output), we get (input, ?)