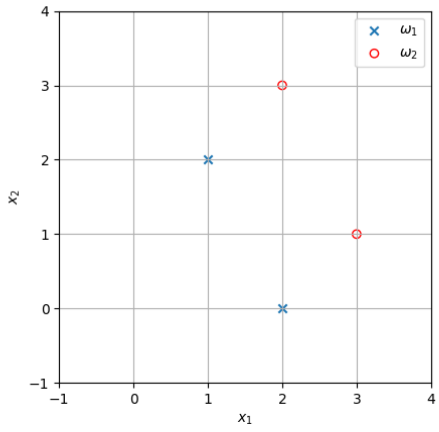
Machine Learning - Theoretical exercise 4

Téo Bouvard

February 21, 2020

Problem 1

a) We have the same number of training samples for classes ω_1 and ω_2, thus the prior probabilities are equal for both classes i.e. $P(\omega_1) = 0.5$ and $P(\omega_2) = 0.5$



b) We compute θ according to the LS-method.

$$\theta = (X^T X)^{-1} X^T y \tag{1}$$

where

$$X = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 0 & 1 \\ 3 & 1 & 1 \\ 2 & 3 & 1 \end{bmatrix}^T \qquad y = \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix}$$

The steps to solve this equation are shown below.

$$X^T X = \begin{bmatrix} 18 & 11 & 8 \\ 11 & 14 & 6 \\ 8 & 6 & 4 \end{bmatrix}$$

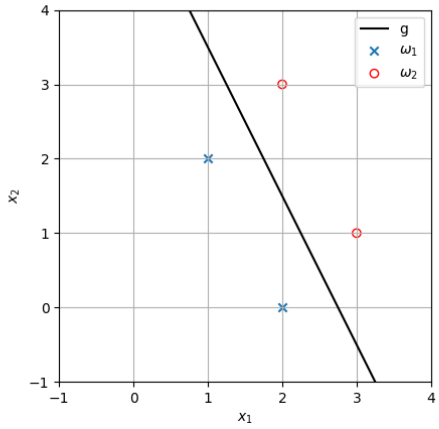
$$(X^T X)^{-1} = \begin{bmatrix} -\frac{5}{9} & \frac{1}{9} & -\frac{23}{18} \\ \frac{1}{9} & \frac{2}{9} & -\frac{5}{9} \\ -\frac{23}{18} & -\frac{5}{9} & \frac{131}{36} \end{bmatrix}$$

$$(X^T X)^{-1} X^T = \begin{bmatrix} -\frac{1}{2} & -\frac{1}{6} & \frac{1}{2} & \frac{1}{6} \\ 0 & -\frac{1}{3} & 0 & \frac{1}{3} \\ \frac{5}{4} & -\frac{13}{12} & -\frac{3}{4} & -\frac{7}{12} \end{bmatrix}$$

$$\boldsymbol{\theta} = \begin{bmatrix} -\frac{4}{3} \\ -\frac{2}{3} \\ \frac{11}{3} \end{bmatrix}$$

To determine the decision boundary, we find the root of the discriminant function.

$$\begin{aligned} g(x) &= 0 \\ -\frac{4}{3}x_1 - \frac{2}{3}x_2 + \frac{11}{3} &= 0 \\ x_2 &= -2x_1 + \frac{11}{2} \end{aligned}$$

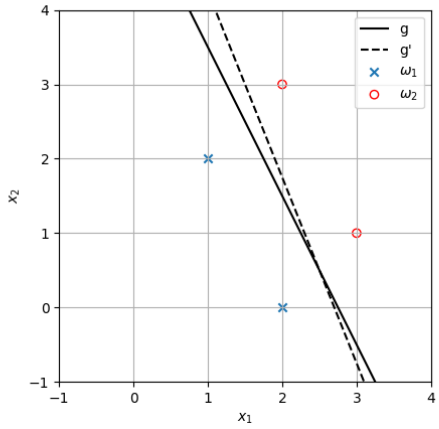


c) If we set $y_4 = -0.5$, we reduce the weight of the fourth training sample, which modifies θ.

$$\theta' = \begin{bmatrix} -\frac{5}{4} \\ -\frac{1}{2} \\ \frac{27}{8} \end{bmatrix}$$

Because $\theta' \neq \theta$, the decision boundary also changes.

$$\begin{aligned} g'(x) &= 0 \\ -\frac{5}{4}x_1 - \frac{1}{2}x_2 + \frac{27}{8} &= 0 \\ x_2 &= -\frac{5}{2}x_1 + \frac{27}{4} \end{aligned}$$



We can see that decreasing the weight of a training sample moves the decison boundary closer to it.

d) We now compute θ with the LMS-method. We use μ to denote the the learning rate. The descent vector at each iteration is denoted ∇.

Initialization

$$\mu^{(0)} = 0.5 \qquad \boldsymbol{\theta}^{(0)} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \qquad \theta = 1$$

Iteration 1

$$\begin{aligned} \nabla &= \mu^{(1)}(y_1 - \boldsymbol{\theta}^{(0)T} y_1 x_1) y_1 x_1 \\ &= \frac{0.5}{1} \left([1] - \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} [1] \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \right) [1] \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \\ &= -\frac{3}{2} \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \end{aligned}$$

$$\left|\nabla\right| = \frac{3\sqrt{6}}{2} \approx 3.7 > \theta$$

$\left|\nabla\right|$ is greater than the threshold, so $\boldsymbol{\theta}$ is updated, and we will need to loop through all samples again for the next epoch.

$$\begin{aligned} \boldsymbol{\theta}^{(1)} &= \boldsymbol{\theta}^{(0)} + \nabla \\ &= -\frac{1}{2} \begin{bmatrix} 1 \\ 4 \\ 1 \end{bmatrix} \end{aligned}$$

Iteration 2

$$\begin{aligned} \nabla &= \mu^{(2)}(y_2 - \boldsymbol{\theta}^{(1)T} y_2 x_2) y_2 x_2 \\ &= \frac{0.5}{2} \left([1] - \begin{bmatrix} -\frac{1}{2} & -2 & -\frac{1}{2} \end{bmatrix} [1] \begin{bmatrix} 2 \\ 0 \\ 1 \end{bmatrix} \right) [1] \begin{bmatrix} 2 \\ 0 \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} \frac{5}{4} \\ 0 \\ \frac{5}{8} \end{bmatrix} \end{aligned}$$

$$\left|\nabla\right| = \frac{5\sqrt{5}}{8} \approx 1.4 > \theta$$

$\left|\nabla\right|$ is greater than the threshold, so $\boldsymbol{\theta}$ is updated.

$$\begin{aligned} \boldsymbol{\theta}^{(2)} &= \boldsymbol{\theta}^{(1)} + \nabla \\ &= \begin{bmatrix} \frac{3}{4} \\ -2 \\ \frac{1}{8} \end{bmatrix} \end{aligned}$$

Iteration 3

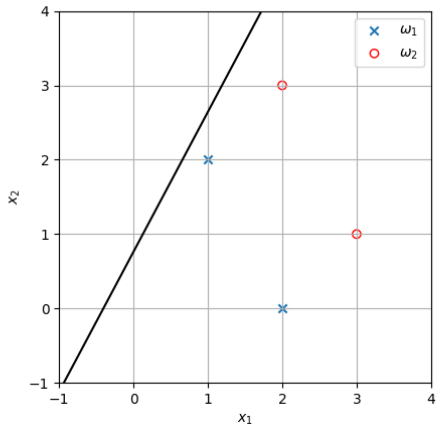
$$
\begin{aligned}
\nabla &= \mu^{(3)}(y_3 - \boldsymbol{\theta}^{(2)T} y_3 x_3) y_3 x_3 \\
&= \frac{0.5}{3}\left([-1] - \begin{bmatrix} \frac{3}{4} & -2 & \frac{1}{8} \end{bmatrix} [-1] \begin{bmatrix} 3 \\ 1 \\ 1 \end{bmatrix} \right) [-1] \begin{bmatrix} 3 \\ 1 \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} \frac{5}{16} \\ \frac{5}{48} \\ \frac{5}{48} \end{bmatrix}
\end{aligned}
$$

$$
|\nabla| = \frac{5\sqrt{11}}{48} \approx 0.3 < \theta
$$

$|\nabla|$ is smaller than the threshold, so $\boldsymbol{\theta}$ is not updated and we continue with the third sample. The rest of the computations were done programmaticaly. After the 12^{th} iteration, the algorithm converges and exits with the following result.

$$
\boldsymbol{\theta} \approx \begin{bmatrix} 0.79 \\ -0.42 \\ 0.36 \end{bmatrix}
$$

If we plot the resulting decision boundary, we observe that the converged value of $\boldsymbol{\theta}$ does not discriminate between the classes.



Problem 2

a) An analytical solution to the equation $\boldsymbol{X\theta} = \boldsymbol{y}$ would be to find the inverse of $\boldsymbol{X}$ and compute $\boldsymbol{\theta} = \boldsymbol{X^{-1}y}$ directly. However, this solution assume that $\boldsymbol{X}$ is invertible. In practice $\boldsymbol{X}$ is often rectangular, with more rows (samples) than columns (features). Trying to find an exact solution would lead to have more equations than unknowns which is not solvable in general.

b) In order to find $\boldsymbol{\theta}$ minimizing the function $\|\boldsymbol{X\theta} - \boldsymbol{y}\|^2$, we can differentiate this function with respect to $\boldsymbol{\theta}$ and find its root.

$$
\frac{\partial \|\boldsymbol{X\theta} - \boldsymbol{y}\|^2}{\partial \boldsymbol{\theta}} = 2\boldsymbol{X}^T(\boldsymbol{X\theta} - \boldsymbol{y})
$$

We now set the derivative to zero.

$$
\begin{aligned}
2\boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{\theta} - \boldsymbol{y}) &= 0 \\
\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\theta} &= \boldsymbol{X}^T\boldsymbol{y} \\
\boldsymbol{\theta} &= (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}
\end{aligned}
$$

In this case, $\boldsymbol{X}^T\boldsymbol{X}$ is guaranteed to be invertible because it is a square symmetric matrix.

c) Let $\boldsymbol{\theta}_*$ be the value of $\boldsymbol{\theta}$ minimizing the squared error. We can rewrite the distance function as such.

$$\|\boldsymbol{X}\boldsymbol{\theta}_* - \boldsymbol{y}\|^2 = \|\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y} - \boldsymbol{y}\|^2$$

d) For the problem to be lineraly separable, the distance function should be equal to zero.

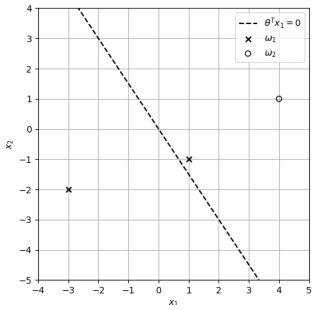
$$\|\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y} - \boldsymbol{y}\|^2 = 0 \implies \boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y} = \boldsymbol{y} \implies \boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T = \boldsymbol{I}$$

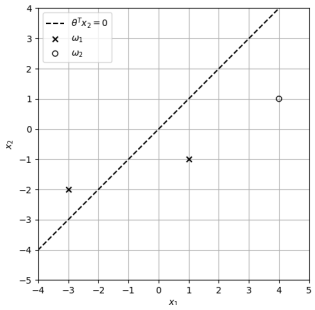
Problem 3

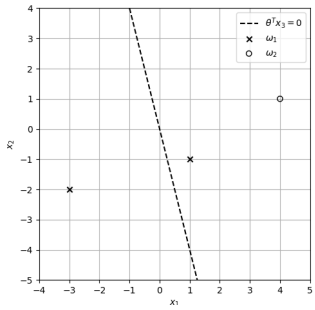
a) Samples are labeled as +1 if they belong to ω_1 and -1 if they belong to ω_2.

$$y_1 = +1 \qquad\qquad y_2 = +1 \qquad\qquad y_3 = -1$$

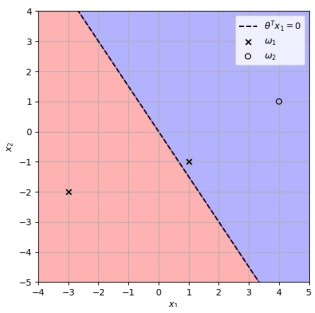
b) For each sample, we draw the decision boundary $y_i\boldsymbol{\theta}_T x_i = 0$. This "normalization" of each sample by its class label simplifies the analysis as we only have to check that $y_i\boldsymbol{\theta}_T x_i > 0$ to conclude that the classification is correct.

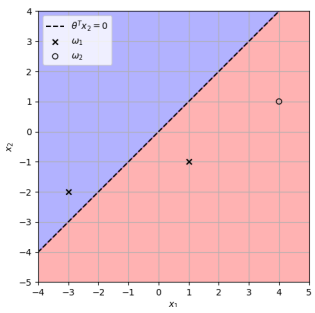


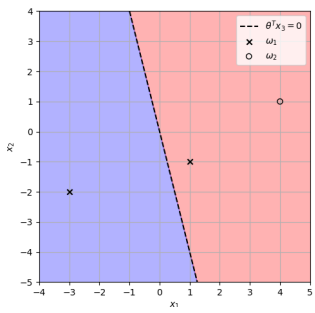




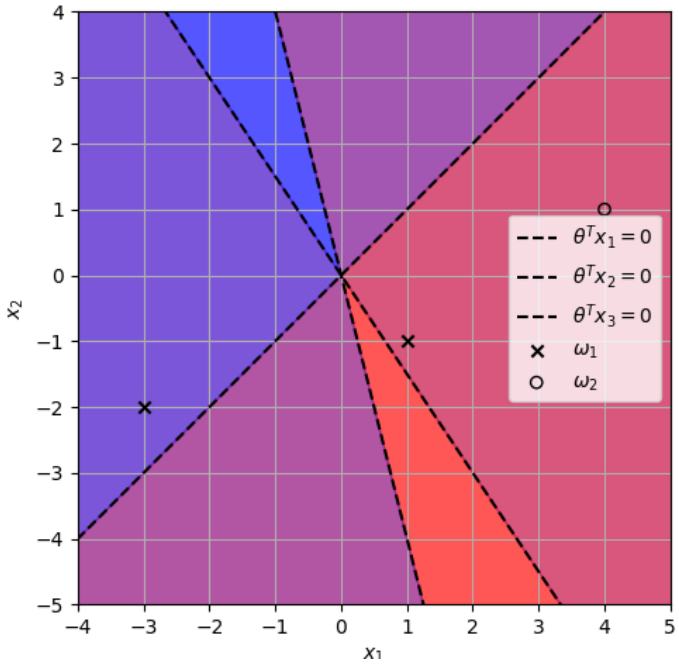
c) The normalization we used also simplifies the visualization as we only need to consider the intersection of all positive regions to find the solution region.If we colour the positive side of the discriminant function in red and the negative side in blue, we get the following regions.







If we superpose the three previous plots, we can identify the solution region by color addition. The solution region is the one having the most vivid red color.



d) We apply the Batch Perceptron algorithm with a constant learning rate. Let X_i be the set of misclassified samples at iteration i. We identify the misclassified samples by checking if $y_i \boldsymbol{\theta}^{(i)^T} > \theta$

Initialization

$$\boldsymbol{\theta}^{(1)} = \begin{bmatrix} 0 & 0 \end{bmatrix}^T \qquad \mu = 1 \qquad \theta = 0$$

Iteration 1

$$\boldsymbol{\theta}^{(1)^T} x_1 = 0 \qquad \boldsymbol{\theta}^{(1)^T} x_2 = 0 \qquad \boldsymbol{\theta}^{(1)^T} x_3 = 0 \qquad \Longrightarrow \qquad X = \{x_1, x_2, x_3\}$$

X is not empty, so we should update $\boldsymbol{\theta}$

$$\begin{aligned} \boldsymbol{\theta}^{(2)} &= \boldsymbol{\theta}^{(1)} + \mu(y_1 x_1 + y_2 x_2 + y_3 x_3) \\ &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -3 \\ -2 \end{bmatrix} + \begin{bmatrix} 1 \\ -1 \end{bmatrix} - \begin{bmatrix} -4 \\ -1 \end{bmatrix} \\ &= \begin{bmatrix} 2 \\ -2 \end{bmatrix} \end{aligned}$$

Iteration 2

$$\boldsymbol{\theta}^{(2)^T} x_1 = -2 \qquad \boldsymbol{\theta}^{(2)^T} x_2 = 4 \qquad \boldsymbol{\theta}^{(2)^T} x_3 = 6 \qquad \Longrightarrow \qquad X = \{x_1\}$$

X is not empty, so we should update $\boldsymbol{\theta}$

$$\begin{aligned} \boldsymbol{\theta}^{(3)} &= \boldsymbol{\theta}^{(2)} + \mu(y_1 x_1) \\ &= \begin{bmatrix} 2 \\ -2 \end{bmatrix} + \begin{bmatrix} -3 \\ -2 \end{bmatrix} \\ &= \begin{bmatrix} -1 \\ -4 \end{bmatrix} \end{aligned}$$

Iteration 3

$$\boldsymbol{\theta}^{(3)^T} x_1 = 11 \qquad \boldsymbol{\theta}^{(3)^T} x_2 = 3 \qquad \boldsymbol{\theta}^{(3)^T} x_3 = -8 \qquad \Longrightarrow \qquad X = \{x_3\}$$

X is not empty, so we should update $\boldsymbol{\theta}$

$$\begin{aligned} \boldsymbol{\theta}^{(4)} &= \boldsymbol{\theta}^{(3)} + \mu(y_3 x_3) \\ &= \begin{bmatrix} -1 \\ -4 \end{bmatrix} - \begin{bmatrix} -4 \\ -1 \end{bmatrix} \\ &= \begin{bmatrix} 3 \\ -3 \end{bmatrix} \end{aligned}$$

At iteration 8, we end up with $\boldsymbol{\theta}^{(8)} = \begin{bmatrix} 2 & -7 \end{bmatrix}^T$.

$$\boldsymbol{\theta}^{(8)^T} x_1 = 8 \qquad \boldsymbol{\theta}^{(3)^T} x_2 = 9 \qquad \boldsymbol{\theta}^{(3)^T} x_3 = 1 \qquad \Longrightarrow \qquad X = \emptyset$$

We can see that all samples are correctly classified with this value of $\boldsymbol{\theta}$.

e) We now apply the Batch Perceptron algorithm, but we decrease decrease the learning rate at each iteration. We get the following results.

i	$\boldsymbol{\theta}^T$	X	$\lvert\mu \sum_{x \in X} y_i x_i\rvert$
1	$\begin{bmatrix} 0 & 0 \end{bmatrix}$	$\{x_1, x_2, x_3\}$	2.82
2	$\begin{bmatrix} 2 & -2 \end{bmatrix}$	$\{x_1\}$	1.80
3	$\begin{bmatrix} 0.5 & -3 \end{bmatrix}$	$\{x_3\}$	1.37
4	$\begin{bmatrix} 1.8 & -2.7 \end{bmatrix}$	$\{x_1\}$	0.90
5	$\begin{bmatrix} 1.1 & -3.2 \end{bmatrix}$	$\emptyset$	0

We can see that decreasing the learning rate at each step reduces the steps needed to converge to the solution region.