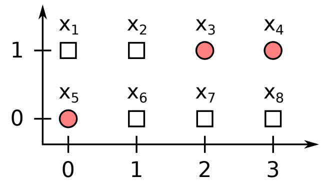
Exercises for the course
Machine Learning 1
Winter semester 2024/25

Fachgebiet Maschinelles Lernen
Institut für Softwaretechnik und theoretische Informatik
Fakultät IV, Technische Universität Berlin
Prof. Dr. Klaus-Robert Müller
Email: klaus-robert.mueller@tu-berlin.de

Exercise Sheet 11

Exercise 1: Boosted Classifiers (25 + 25 P)

We consider a two-dimensional dataset $x_1, \dots, x_8 \in \mathbb{R}^2$ with binary labels $y_1, \dots, y_8 \in \{-1, 1\}$.



Red circles denote the first class ($y_i = +1$) and white squares denote the second class ($y_i = -1$). We decide to classify this data with a boosted classifier and use the nearest mean classifier as a weak classifier. The boosted classifier is given by

$$f(x) = \mathrm{sign}\Big(\alpha_0 + \sum_{t=1}^{T} \alpha_t h_t(x)\Big)$$

where $\alpha_0, \dots, \alpha_T \in \mathbb{R}$ are the boosting coefficients. The tth nearest mean classifier is given by

$$h_t(x) = \begin{cases} +1 & \|x - \mu_t^+\| < \|x - \mu_t^-\| \\ -1 & \text{else} \end{cases} \qquad \text{with} \quad \mu_t^+ = \frac{\sum_{i:y_i=+1} p_i^{(t)} x_i}{\sum_{i:y_i=+1} p_i^{(t)}} \quad \text{and} \quad \mu_t^- = \frac{\sum_{i:y_i=-1} p_i^{(t)} x_i}{\sum_{i:y_i=-1} p_i^{(t)}}.$$

where $p_1^{(t)}, \dots, p_N^{(t)}$ are the data weighting terms for this classifier.

(a) *Draw* at hand a possible boosted classifier that classifies the dataset above, i.e. *draw* the decision boundary of the weak classifiers $h_t(x)$ and of the final boosted classifier $f(x)$. We use the convention $\mathrm{sign}(0) = 0$.

(b) *Write* the weighting terms $p_i^{(t)}$ and the coefficients $\alpha_0, \dots, \alpha_T$ associated to the classifiers you have drawn.

(Note: In this exercise, the boosted classifier does not *need to derive from a particular algorithm. Instead, the number of weak classifiers, the coefficients and the weighting terms can be picked at hand with the sole constraint that the final classifier implements the desired decision boundary.)*

Exercise 2: AdaBoost as an Optimization Problem (25 + 25 P)

Consider AdaBoost for binary classification applied to some dataset $\mathcal{D} = \{(x_1, y_1), \dots, (x_N, y_N)\}$. The algorithm starts with uniform weighting ($\forall_{i=1}^N : \; p_i^{(1)} = 1/N$) and performs the following iteration:

for $t = 1 \dots T$:

Step 1:	$\mathcal{D}, p^{(t)} \mapsto h_t$	(learn tth weak classifier using weighting $p^{(t)}$)
Step 2:	$\epsilon_t = \mathbb{E}_{p^{(t)}}[1_{(h_t(x) \neq y)}]$	(compute the weighted error of the classifier)
Step 3:	$\alpha_t = \frac{1}{2} \log\Big(\frac{1 - \epsilon_t}{\epsilon_t}\Big)$	(set its contribution to the boosted classifier)
Step 4:	$\forall_{i=1}^N : \; p_i^{(t+1)} = Z_t^{-1} p_i^{(t)} \exp(-\alpha_t y_i h_t(x_i))$	(set a new weighting for the data)

The term $\mathbb{E}_{p^{(t)}}[\cdot]$ denotes the expectation under the data weighting $p^{(t)}$, and Z_t is a normalization term. An interesting property of AdaBoost is that it can be shown to minimize some objective function

$$\mathcal{G}(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \exp(-y_i f_{\boldsymbol{\alpha},t}(x_i))$$

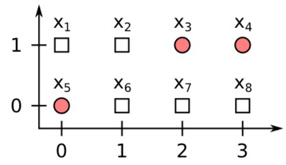
where $f_{\boldsymbol{\alpha},t}(x) = \sum_{\tau=1}^{t} \alpha_\tau h_\tau(x)$ is the output score of the boosted classifier after t iterations.

(a) *Show* that the objective can be rewritten as $\mathcal{G}(\boldsymbol{\alpha}) = N \cdot \Big(\prod_{\tau=1}^{t-1} Z_\tau\Big) \cdot \sum_{i=1}^{N} p_i^{(t)} \exp(-y_i \alpha_t h_t(x_i))$.

(b) *Show* that Step 3 of the AdaBoost procedure above is equivalent to computing $\alpha_t = \arg\min_{\alpha_t} \mathcal{G}(\boldsymbol{\alpha})$.

Exercise 1: Boosted Classifiers (25 + 25 P)

We consider a two-dimensional dataset $x_1, \dots, x_8 \in \mathbb{R}^2$ with binary labels $y_1, \dots, y_8 \in \{-1, 1\}$.



Red circles denote the first class ($y_i = +1$) and white squares denote the second class ($y_i = -1$). We decide to classify this data with a boosted classifier and use the nearest mean classifier as a weak classifier. The boosted classifier is given by

$$f(x) = \operatorname{sign}\Big(\alpha_0 + \sum_{t=1}^{T} \alpha_t h_t(x)\Big)$$

where $\alpha_0, \dots, \alpha_T \in \mathbb{R}$ are the boosting coefficients. The tth nearest mean classifier is given by

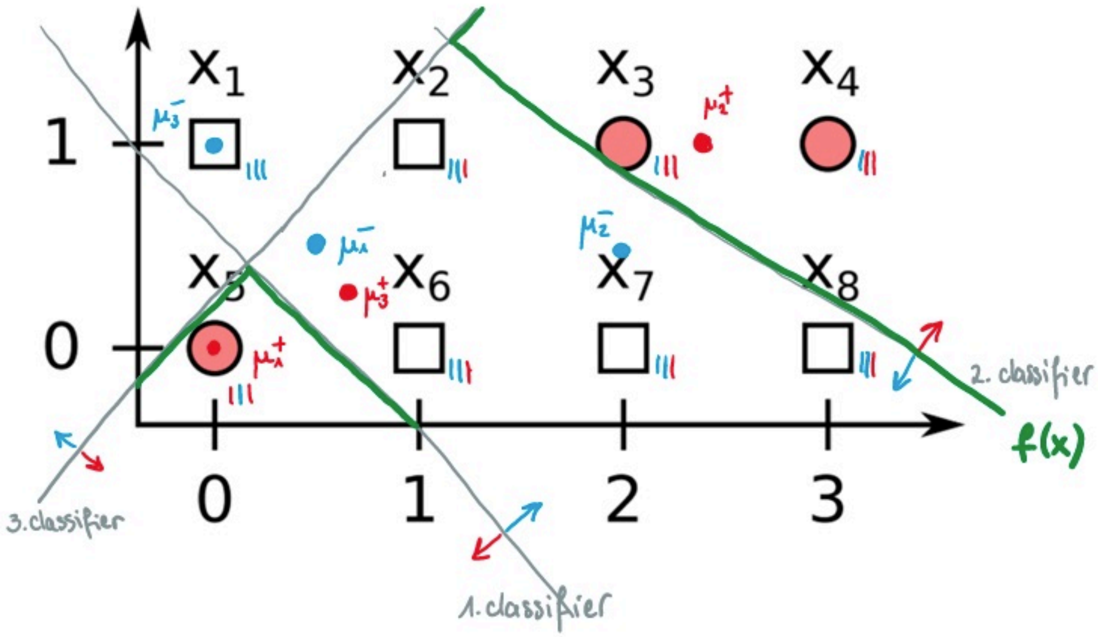
$$h_t(x) = \begin{cases} +1 & \|x - \mu_t^+\| < \|x - \mu_t^-\| \\ -1 & \text{else} \end{cases} \qquad \text{with} \quad \mu_t^+ = \frac{\sum_{i:y_i=+1} p_i^{(t)} x_i}{\sum_{i:y_i=+1} p_i^{(t)}} \quad \text{and} \quad \mu_t^- = \frac{\sum_{i:y_i=-1} p_i^{(t)} x_i}{\sum_{i:y_i=-1} p_i^{(t)}}.$$

where $p_1^{(t)}, \dots, p_N^{(t)}$ are the data weighting terms for this classifier.

(a) *Draw* at hand a possible boosted classifier that classifies the dataset above, i.e. *draw* the decision boundary of the weak classifiers $h_t(x)$ and of the final boosted classifier $f(x)$. We use the convention $\operatorname{sign}(0) = 0$.

(b) *Write* the weighting terms $p_i^{(t)}$ and the coefficients $\alpha_0, \dots, \alpha_T$ associated to the classifiers you have drawn.

(Note: In this exercise, the boosted classifier does not *need to derive from a particular algorithm. Instead, the number of weak classifiers, the coefficients and the weighting terms can be picked at hand with the sole constraint that the final classifier implements the desired decision boundary.)*



1. classifier :

$\mu_1^+ = \begin{pmatrix} 0 \\ 0 \end{pmatrix} = x_5$

$p_5^{(1)} = 1,$

$p_3^{(1)} = p_4^{(1)} = 0$

$\mu_1^- = \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix} = \frac{1}{2}\big((x_1 + x_6)\big)$

$p_1^{(1)} = p_6^{(1)} = \frac{1}{2}$

$p_2^{(1)} = p_7^{(1)} = p_8^{(1)} = 0$

2. classifier :

$\mu_1^+ = \begin{pmatrix} 2.\overline{3} \\ 0.5 \end{pmatrix} = \frac{2}{3} x_3 + \frac{1}{3} x_4$

$p_3^{(2)} = \frac{2}{3} \quad p_4^{(2)} = \frac{1}{3}$

$p_5^{(2)} = 0$

$\mu_2^- = \begin{pmatrix} 2 \\ 0.5 \end{pmatrix} = \frac{1}{2}\big((x_2 + x_8)\big)$

$p_2^{(2)} = p_8^{(2)} = \frac{1}{2}$

$p_1^{(2)} = p_6^{(2)} = p_7^{(2)} = 0$

3. classifier :

$\mu_3^+ = \begin{pmatrix} 2/3 \\ 1/3 \end{pmatrix} = \frac{1}{3} x_3 + \frac{2}{3} x_5$

$p_3^{(3)} = \frac{1}{3} \quad p_5^{(3)} = \frac{2}{3}$

$p_4^{(3)} = 0$

$\mu_3^- = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = x_1$

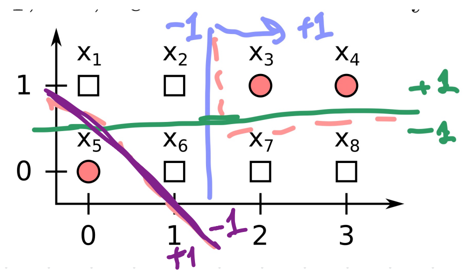
$p_1^{(3)} = 1$

$p_2^{(3)} = p_6^{(3)} = p_7^{(3)} = p_8^{(3)} = 0$

Final classifier combines all classifiers with equal weights $\alpha_1 = \alpha_2 = \alpha_3$ and $\alpha_0 = 0$.

11) Boosting

Ex 1) a)



≈ NN
do a NMC to find ±1 for x_2 & x_3
±1 for x_3 & x_7

for p:
0: other points
1: points we choose

probability vectors: $p^{(1)} = [0 \quad 1/2 \quad 1/2 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$

for ±1, only x_2 and x_3

for prob. dist: divide to 2
$P(x_2) + P(x_3) = 1$

if we use this weighting then $\mu_t^+ = x_3$ $\mu_t^- = x_2$ and the decision function will be $-1 \mid +1$

for $\frac{+1}{-1}$: $p^{(2)} = [0 \quad 0 \quad 1/2 \quad 0 \quad 0 \quad 0 \quad 1/2 \quad 0]$

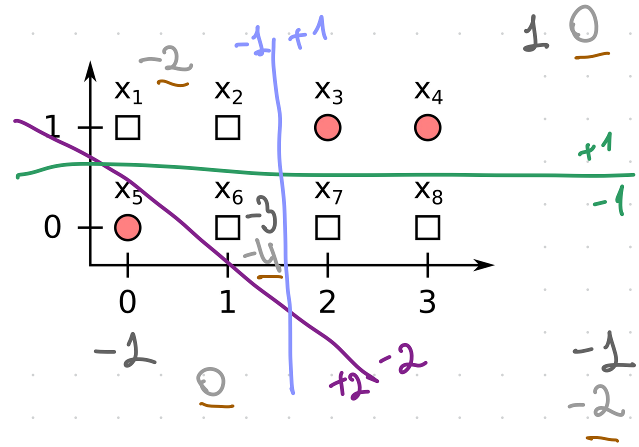
only x_3 & x_7 so that the NMC will produce in x_3 & x_7 as μ_t^+ / μ_t^-

we could also chose other points, f.e. x_3 & x_7 / x_4 & x_8 to get —— as db

for $\diagdown$ -1 / $+1$: $p^{(3)} = [0 \quad 1/3 \quad 0 \quad 0 \quad 1/3 \quad 1/3 \quad 0 \quad 0]$

we have 3 points now: x_5, x_1, x_6

Regions of our db:



3 decision boundaries
find the α such that $f(x)$ will produce the right output

there are now 2 different sections classified as -1

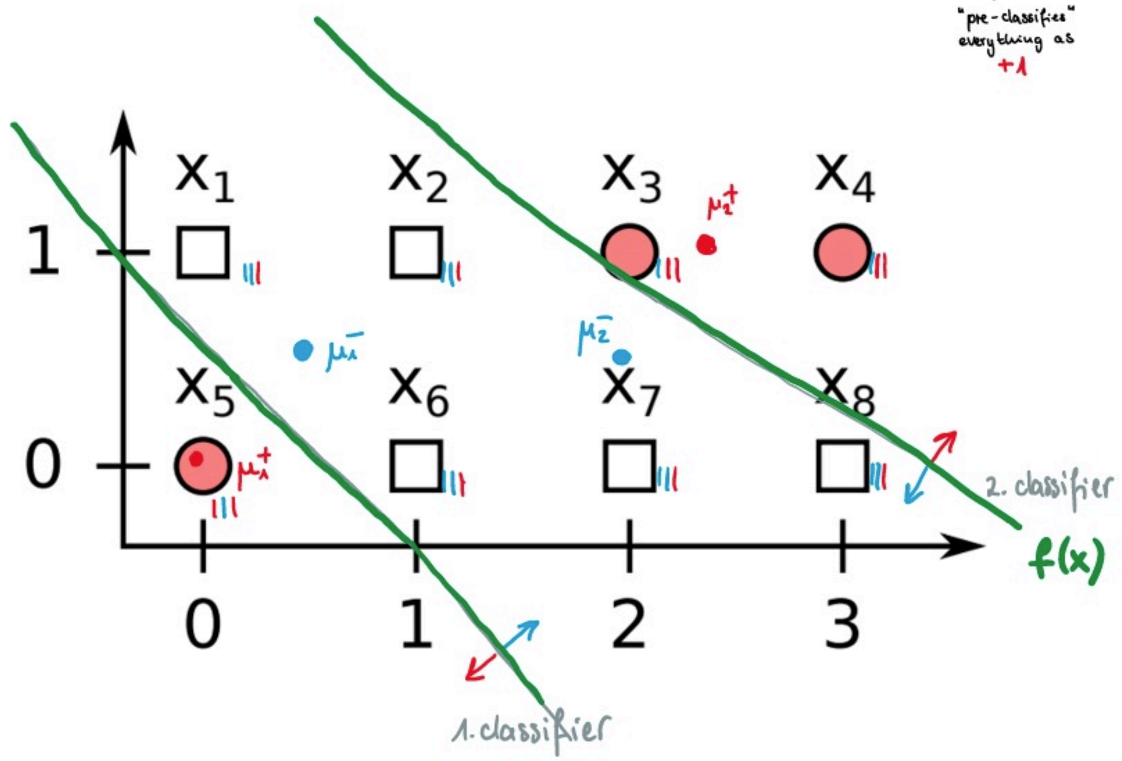
increase the α of the 3rd classifier, for ex: +2/-2 to compensate

set $\alpha_0 = -1$ → then the sign will be correct:

$\alpha_0 = -1$ $\quad \alpha_1 = 1$ $\quad \alpha_2 = 1$ $\quad \alpha_3 = 2$ → parameters of a boosted classifier to get our decision boundries

Alternative solution:

Final classifier only combines classifiers 1 and 2 with equal weights $\alpha_1 = \alpha_2 = \underbrace{\alpha_0}_{\text{"pre-classifies" everything as } +1} = 1$



Exercise 2: AdaBoost as an Optimization Problem (25 + 25 P)

Consider AdaBoost for binary classification applied to some dataset $\mathcal{D} = \{(x_1, y_1), \dots, (x_N, y_N)\}$. The algorithm starts with uniform weighting ($\forall_{i=1}^N : \ p_i^{(1)} = 1/N$) and performs the following iteration:

for $t = 1 \dots T$:

Step 1:	$\mathcal{D}, p^{(t)} \mapsto h_t$	(learn tth weak classifier using weighting $p^{(t)}$)
Step 2:	$\epsilon_t = \mathbb{E}_{p^{(t)}}[1_{(h_t(x) \neq y)}]$	(compute the weighted error of the classifier)
Step 3:	$\alpha_t = \frac{1}{2} \log \left(\frac{1 - \epsilon_t}{\epsilon_t} \right)$	(set its contribution to the boosted classifier)
Step 4:	$\forall_{i=1}^N : \ p_i^{(t+1)} = Z_t^{-1} p_i^{(t)} \exp(-\alpha_t y_i h_t(x_i))$	(set a new weighting for the data)

The term $\mathbb{E}_{p^{(t)}}[\cdot]$ denotes the expectation under the data weighting $p^{(t)}$, and Z_t is a normalization term. An interesting property of AdaBoost is that it can be shown to minimize some objective function

$$\mathcal{G}(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \exp(-y_i f_{\boldsymbol{\alpha},t}(x_i))$$

where $f_{\boldsymbol{\alpha},t}(x) = \sum_{\tau=1}^{t} \alpha_\tau h_\tau(x)$ is the output score of the boosted classifier after t iterations.

(a) *Show* that the objective can be rewritten as $\mathcal{G}(\boldsymbol{\alpha}) = N \cdot \left(\prod_{\tau=1}^{t-1} Z_\tau \right) \cdot \sum_{i=1}^{N} p_i^{(t)} \exp(-y_i \alpha_t h_t(x_i))$.

$$\boxed{\begin{aligned} &* \\ &p_i^{(t+1)} = Z_t^{-1} p_i^{(t)} \exp(-u_i^{(t)}) \\ \Leftrightarrow \ &Z_t \, p_i^{(t+1)} = p_i^{(t)} \exp(-u_i^{(t)}) \end{aligned}}$$

$$g(\alpha) = N \cdot \left(\prod_{\tau=1}^{t-1} Z_\tau \right) \cdot \sum_{i=1}^{N} p_i^{(t)} \overbrace{\underline{\exp(-y_i \alpha_t h_t(x_i))}}^{=: \exp\{-u_i^{(t)}\}}$$

$$= N \cdot \sum_{i=1}^{N} \left(\prod_{\tau=1}^{t-1} Z_\tau \right) p_i^{(t)} \exp(-u_i^{(t)})$$

$$= N \cdot \sum_{i=1}^{N} \left(\prod_{\tau=1}^{t-2} Z_\tau \right) \cdot Z_{t-1} \cdot p_i^{(t)} \cdot \exp(-u_i^{(t)})$$

$$\overset{*}{=} N \cdot \sum_{i=1}^{N} \left(\prod_{\tau=1}^{t-2} Z_\tau \right) \cdot p_i^{(t-1)} \exp(-u_i^{(t-1)}) \cdot \exp(-u_i^{(t)})$$

$$\overset{*}{=} N \cdot \sum_{i=1}^{N} \left(\prod_{r=1}^{t-3} Z_r \right) p_i^{(t-2)} \exp(-u_i^{(t-2)}) \exp(-u_i^{(t-1)}) \exp(-u_i^{(t)})$$

$$\vdots$$

$$\overset{*}{=} N \cdot \sum_{i=1}^{N} p_i^{(1)} \cdot \prod_{r=1}^{t} \exp(-u_i^{(r)})$$

$$\overset{p_i^{(1)} = \frac{1}{N}}{=} \sum_{i=1}^{N} \prod_{\tau=1}^{t} \exp(-u_i^{(\tau)})$$

$$= \sum_{i=1}^{N} \exp\left(\sum_{\tau=1}^{t} -u_i^{(\tau)} \right)$$

$$= \sum_{i=1}^{N} \exp\left(\sum_{\tau=1}^{t} -y_i \alpha_\tau h_\tau(x_i) \right)$$

$$= \sum_{i=1}^{N} \exp\left(-y_i \sum_{\tau=1}^{t} \alpha_\tau h_\tau(x_i) \right)$$

$$= \sum_{i=1}^{N} \exp\left(-y_i \cdot f_{\alpha,t}(x_i) \right)$$

(b) *Show* that Step 3 of the AdaBoost procedure above is equivalent to computing $\alpha_t = \arg\min_{\alpha_t} \mathcal{G}(\boldsymbol{\alpha})$.

$$g(\alpha) = N \cdot \left(\prod_{r=1}^{t-1} Z_r \right) \cdot \sum_{i=1}^{N} p_i^{(t)} \exp(-y_i \alpha_t h_t(x_i))$$

$$\frac{\partial g(\alpha)}{\partial \alpha_t} = N \cdot \left(\prod_{r=1}^{t-1} Z_r \right) \cdot \sum_{i=1}^{N} p_i^{(t)} \exp(-y_i \alpha_t h_t(x_i)) \cdot (-y_i h_t(x_i)) \overset{!}{=} 0$$

$$\Leftrightarrow \sum_{i=1}^{N} p_i^{(t)} \exp(-y_i \alpha_t h_t(x_i)) \cdot (-y_i h_t(x_i)) \overset{!}{=} 0$$

$h_t(x_i), y_i \in \{-1, 1\}$ so that $(-y_i h_t(x_i)) = \begin{cases} +1 & \text{if } y_i \neq h_t(x_i) \\ -1 & \text{if } y_i = h_t(x_i) \end{cases}$

$$\Leftrightarrow \sum_{i=1}^{N} p_i^{(t)} \cdot \left(1_{(y_i \neq h_t(x_i))} \cdot \exp\{\alpha_t\} - 1_{(y_i = h_t(x_i))} \cdot \exp(-\alpha_t) \right) = 0$$

$$\Leftrightarrow \exp\{\alpha_t\} \sum_{i : y_i \neq h_t(x_i)} p_i^{(t)} = \exp\{-\alpha_t\} \sum_{i : y_i = h_t(x_i)} p_i^{(t)}$$

$$\Leftrightarrow \exp\{\alpha_t\} \sum_{i : y_i \neq h_t(x_i)} p_i^{(t)} = \exp\{-\alpha_t\} \cdot \left(1 - \sum_{i : y_i \neq h_t(x_i)} p_i^{(t)} \right)$$

$$\Leftrightarrow \exp\{\alpha_t\} \mathbb{E}_{p^{(t)}}\left[\mathbb{1}_{(y_i \neq h_t(x_i))} \right] = \exp\{-\alpha_t\} \left(1 - \mathbb{E}_{p^{(t)}}\left[\mathbb{1}_{(y_i \neq h_t(x_i))} \right] \right)$$

$$\Leftrightarrow \exp\{\alpha_t\} \varepsilon_t = \exp\{-\alpha_t\} (1 - \varepsilon_t)$$

$$\Leftrightarrow \frac{\exp\{\alpha_t\}}{\exp\{-\alpha_t\}} = \frac{(1 - \varepsilon_t)}{\varepsilon_t}$$

$$\Leftrightarrow \log\{\exp\{2\alpha_t\}\} = \log\left\{ \frac{(1-\varepsilon_t)}{\varepsilon_t} \right\}$$

$$\Leftrightarrow \alpha_t = \frac{1}{2} \log\left\{ \frac{(1-\varepsilon_t)}{\varepsilon_t} \right\}$$