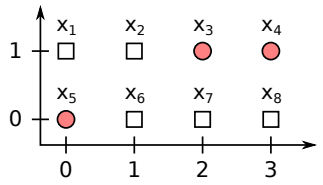
Exercises for the course
Machine Learning 1
Winter semester 2021/22

Abteilung 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 10

Exercise 1: Boosted Classifiers (25 + 25 P)

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



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 $\alpha_0, \ldots, \alpha_T \in \mathbb{R}$ 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 $p_1^{(t)}, \ldots, p_N^{(t)}$ 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 $\alpha_0, \ldots, \alpha_T$ 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 $\mathcal{D} = \{(x_1, y_1), \ldots, (x_N, y_N)\}$. The algorithm starts with uniform weighting ($\forall_{i=1}^N : \; p_i^{(1)} = 1/N$) and performs the following iteration:

for $t = 1 \ldots T$:

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.

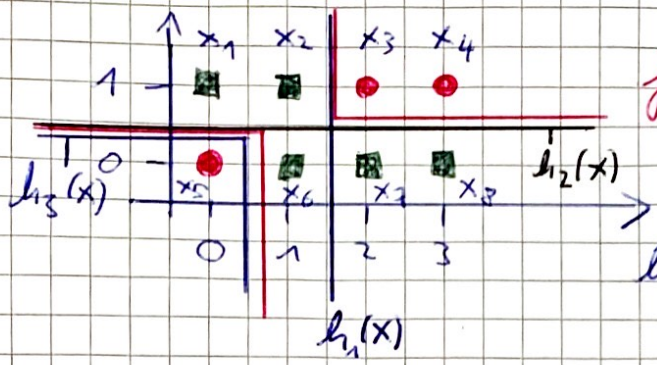
(a) *Show* that the objective can be rewritten as $\mathcal{G}(\boldsymbol{\alpha}) = N \cdot \Big(\prod_{\tau=1}^{t-1} Z_\tau\Big) \cdot \sum_{i=1}^{N} p_i^{(t)} \exp(-y_i \alpha_t h_t(x_i))$.

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

ML10

1

a)



$f(x)$: final boosted classifier

$h_t(x)$: weak classifiers

b)

For the final classifier $f(x) = \text{sign}\left(\alpha_0 + \sum_{t=1}^{T} \alpha_t h_t(x)\right)$ we need to choose proper weights α_i for each weak classifier $h_i(x)$. The first two classifiers $h_1(t)$ and $h_2(t)$ can be weighted equally:

$$\alpha_1 = \alpha_2 = 1$$

The third classifier needs a larger weight because it classifies x_5 as a single classifier (x_3 and x_4 are two classifiers). We would like values of zero to become negative such that they are correctly classified as $y_i = -1$. Therefore, we choose the bias weight negative:

$$\alpha_0 = -1$$

The data weighting terms $\rho_i^{(t)}$ need to select the correct linear decision boundary using the mean of the corresponding data points. For example, for $h_2(x)$ the correct mean is between x_2 and x_3. So our 3 weighting vectors ρ_i become:

$\rho_1 = [\,0, 1, 1, 0, 0, 0, 0, 0\,]$

$\rho_2 = [\,0, 0, 1, 1, 0, 0, 1, 1\,]$

$\rho_3 = [\,1, 0, 0, 0, 1, 1, 0, 0\,]$

Our data values are

$$X = \left\{ \begin{pmatrix}1\\0\end{pmatrix}, \begin{pmatrix}1\\1\end{pmatrix}, \begin{pmatrix}1\\2\end{pmatrix}, \begin{pmatrix}1\\3\end{pmatrix}, \begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}0\\1\end{pmatrix}, \begin{pmatrix}0\\2\end{pmatrix}, \begin{pmatrix}0\\3\end{pmatrix} \right\}$$

and the labels are

$$y = \{ -1, -1, +1, +1, +1, -1, -1, -1 \}$$

The first classifier $h_1(x)$ is:

$$h_1(x) = \begin{cases} +1 & \|x - \mu_1^+\| < \|x - \mu_1^-\| \\ -1 & \text{else} \end{cases}$$

The mean vectors are

$$\mu_1^+ = \frac{\sum_{i:y_i=+1} p_i^{(1)} x_i}{\sum_{i:y_i=+1} p_i^{(1)}} = \frac{p_2 x_2 + p_3 x_3 + p_4 x_4}{p_2 + p_3 + p_4} = \frac{1\cdot\begin{pmatrix}1\\1\end{pmatrix} + 1\cdot\begin{pmatrix}1\\2\end{pmatrix} + 0\cdot\begin{pmatrix}1\\3\end{pmatrix}}{1+1+0} = \frac{1}{2}\begin{pmatrix}2\\3\end{pmatrix}$$

and

$$\mu_1^- = \frac{\sum_{i:y_i=-1} p_i^{(1)} x_i}{\sum_{i:y_i=-1} p_i^{(1)}} = \frac{p_1 x_1 + p_5 x_5 + p_6 x_6 + p_7 x_7 + p_8 x_8}{p_1 + p_5 + p_6 + p_7 + p_8} = \begin{pmatrix}0\\0\end{pmatrix}$$

because $p_1 = p_5 = p_6 = p_7 = p_8 = 0$

The first weak classifier $h_1(x)$ for the first data point x_1 becomes

$$h_1\begin{pmatrix}1\\0\end{pmatrix} = \begin{cases} +1 & \text{if } \left\|\begin{pmatrix}1\\0\end{pmatrix} - \frac{1}{2}\begin{pmatrix}2\\3\end{pmatrix}\right\| < \left\|\begin{pmatrix}1\\0\end{pmatrix} - \begin{pmatrix}0\\0\end{pmatrix}\right\| \iff \left\|\begin{pmatrix}0\\-3/2\end{pmatrix}\right\| < \left\|\begin{pmatrix}1\\0\end{pmatrix}\right\| \\ -1 & \text{else} \end{cases}$$

$$\iff h_1\begin{pmatrix}1\\0\end{pmatrix} = \begin{cases} +1 & \text{if } \frac{3}{2} < 1 \\ -1 & \text{else} \end{cases}$$

Therefore, $h_1(x_1) = -1$.

Analogously for $x_2, \ldots, x_8$ and $h_2(x)$ and $h_3(x)$.

2 a)

Show that the objective

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

can be rewritten as

2) $$\mathcal{G}(\alpha) = N \cdot \left(\prod_{\tau=1}^{t-1} Z_\tau\right) \sum_{i=1}^{N} p_i^{(t)} \exp\left(-y_i \alpha_t h_t(x_i)\right)$$

We rewrite 1):

3) $$\mathcal{G}(\alpha) = \sum_{i=1}^{N} \exp\left(-y_i \sum_{\tau=1}^{t-1} \alpha_\tau h_\tau(x_i)\right) \cdot \exp\left(-y_i \alpha_t h_t(x_i)\right)$$

Using the AdaBoost algorithm we can iteratively calculate p_i:

$$p_i^{(1)} = \frac{1}{N}$$

$$p_i^{(2)} = Z_1^{-1} p_i^{(1)} \exp\left(-\alpha_1 y_i h_1(x_i)\right)$$

$$p_i^{(3)} = Z_2^{-1} p_i^{(2)} \exp\left(-\alpha_2 y_i h_2(x_i)\right)$$

$$= Z_2^{-1} Z_1^{-1} \frac{1}{N} \exp\left(-\alpha_1 y_i h_1(x_i)\right) \exp\left(-\alpha_2 y_i h_2(x_i)\right)$$

$$\vdots$$

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

4) $$\iff N\left(\prod_{\tau=1}^{t-1} Z_\tau\right) p_i^{(t)} = \exp\left(-y_i \sum_{\tau=1}^{t-1} \alpha_\tau h_\tau(x_i)\right)$$

Inserting 4) in 3) yields

$$\mathcal{G}(\alpha) = \sum_{i=1}^{N} N\left(\prod_{\tau=1}^{t-1} Z_\tau\right) p_i \exp\left(-y_i \alpha_t h_t(x_i)\right)$$

$$= N\left(\prod_{\tau=1}^{t-1} Z_\tau\right) \sum_{i=1}^{N} p_i \exp\left(-y_i \alpha_t h_t(x_i)\right)$$

b) The value α_t which minimizes the objective $\mathcal{G}(\alpha)$ can be obtained by setting the partial derivative to 0:

$$\alpha_t = \operatorname*{argmin}_{\alpha_t} \mathcal{G}(\alpha)$$

$$\frac{\partial \mathcal{G}(\alpha)}{\partial \alpha_t} = \frac{\partial}{\partial \alpha_t}\left(N \left(\prod_{c=1}^{t-1} Z_c\right) \sum_{i=1}^{N} p_i^{(t)} \exp\left(-y_i \alpha_t h_t(x_i)\right)\right)$$

$$= \underbrace{N\left(\prod_{c=1}^{t-1} Z_c\right)}_{\text{const.}} \sum_{i=1}^{N} p_i^{(t)} (-y_i h_t(x_i)) \exp(-y_i \alpha_t h_t(x_i)) \overset{!}{=} 0$$

y_i and $h_t(x_i)$ are $\in \{-1, +1\}$

Therefore we can split the previous term into

$$\frac{\partial \mathcal{G}(\alpha)}{\partial \alpha_t} = \underbrace{\sum_{i \in "+"}^{N} p_i^{(t)} \exp(-\alpha_t)(-1)}_{y_i \cdot h_t(x_i) = +1} + \underbrace{\sum_{i \in "-"} p_i^{(t)} \exp(\alpha_t) \cdot 1}_{y_i \cdot h_t(x_i) = -1} = 0$$

$$\Leftrightarrow \underbrace{\left(\sum_{i \in "+"} p_i^{(t)}\right)}_{= 1 - \varepsilon_t} \exp(-\alpha_t) = \underbrace{\left(\sum_{i \in "-"} p_i^{(t)}\right)}_{= \varepsilon_t} \exp(\alpha_t)$$

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

$$\Leftrightarrow \ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right) = \alpha_t + \alpha_t$$

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