

# Convexity, Subgradients and SVM

## Optimization for Machine Learning — Homework #1

Monday 8<sup>th</sup> May, 2023

The theory part can be handed-in physically during the exercise session, or digitally if typeset on Moodle. The programming part has to be sent on Moodle. *Group work is allowed (2 – 3 people), but submissions are personal.*

### Part I: Theory

12+2 points

#### I.1. Convexity

**Exercise I.1** (2+1 points). Let  $f: \mathbb{R}^d \rightarrow \mathbb{R}$  and  $g: \mathbb{R}^d \rightarrow \mathbb{R}$  be two convex functions. Show that  $f + g$  is convex.

*Bonus:* Show that  $x \mapsto \max(f(x), g(x))$  is convex.

**Answer I.1.** Let  $f: \mathbb{R}^d \rightarrow \mathbb{R}$  and  $g: \mathbb{R}^d \rightarrow \mathbb{R}$  be two convex functions. The domain of the function  $h := f + g$  is  $\mathbb{R}^d$ , which is convex. Then, let  $\lambda \in [0, 1]$ ,  $(x, y) \in \mathbb{R}^d \times \mathbb{R}^d$ , and evaluate

$$\begin{aligned} h(\lambda x + (1 - \lambda)y) &= f(\lambda x + (1 - \lambda)y) + g(\lambda x + (1 - \lambda)y) \\ &\leq \lambda f(x) + (1 - \lambda)f(y) + \lambda g(x) + (1 - \lambda)g(y) \quad \text{since } f \text{ and } g \text{ are convex} \\ &\leq \lambda(f(x) + g(x)) + (1 - \lambda)(f(y) + g(y)) \\ &\leq \lambda h(x) + (1 - \lambda)h(y), \end{aligned}$$

and  $h = f + g$  is convex.

*Bonus:* Now, let  $h: x \mapsto \max(f(x), g(x))$ . The domain of  $h$  is convex. For  $\lambda \in [0, 1]$  and  $(x, y) \in \mathbb{R}^d \times \mathbb{R}^d$ , evaluate

$$\begin{aligned} h(\lambda x + (1 - \lambda)y) &= \max \{f(\lambda x + (1 - \lambda)y), g(\lambda x + (1 - \lambda)y)\} \\ &\leq \max \{ \lambda f(x) + (1 - \lambda)f(y), \lambda g(x) + (1 - \lambda)g(y) \} \quad f, g \text{ convex} \\ &\leq \max \{ \lambda f(x), \lambda g(x) \} + \max \{ (1 - \lambda)f(y), (1 - \lambda)g(y) \} \end{aligned}$$

(since  $\max(a + b, c + d) \leq \max(a, c) + \max(b, d)$  for all  $(a, b, c, d) \in \mathbb{R}^4$ )

$$\begin{aligned} &\leq \lambda \max \{f(x), g(x)\} + (1 - \lambda) \max \{f(y), g(y)\} \quad \max(ca, cb) = c \max(a, b) \text{ for } c \geq 0 \\ &\leq \lambda h(x) + (1 - \lambda)h(y). \end{aligned}$$

**Exercise I.2** (2+1 points). Let  $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$  be a linear function (i.e.  $f(x) = Ax$ , for  $A \in \mathbb{R}^{m \times n}$ ) and  $g: \mathbb{R}^m \rightarrow \mathbb{R}$  be convex functions. Show that  $g \circ f: \mathbb{R}^n \rightarrow \mathbb{R}$  is convex.

*Bonus:* What about if  $f$  is affine, i.e.  $f(x) = Ax + b$ , with  $b \in \mathbb{R}^m$ ?

**Answer I.2.** The domain of  $g \circ f$  is convex ( $\mathbb{R}^n$ ). Let  $\lambda \in [0, 1]$ , and  $(x, y) \in \mathbb{R}^n \times \mathbb{R}^n$ . Evaluate

$$\begin{aligned} g \circ f(\lambda x + (1 - \lambda)y) &= g(A(\lambda x + (1 - \lambda)y)) = g(\lambda Ax + (1 - \lambda)Ay) \\ &\leq \lambda g(Ax) + (1 - \lambda)g(Ay) \quad (g \text{ is convex}) \\ &\leq \lambda(g \circ f)(x) + (1 - \lambda)(g \circ f)(y), \end{aligned}$$

and  $g \circ f$  is convex.

*Bonus:* If  $f$  is affine on  $\mathbb{R}^n$ , i.e.  $f(x) = Ax + b$  with  $A \in \mathbb{R}^{m \times n}$  and  $b \in \mathbb{R}^m$ , a common trick is to express  $f$  as a linear function on  $\mathbb{R}^{n+1}$ , by setting  $\tilde{x} = \begin{pmatrix} x \\ 1 \end{pmatrix} \in \mathbb{R}^{n+1}$  and  $\tilde{A} = \begin{pmatrix} A & b \end{pmatrix} \in \mathbb{R}^{m \times (n+1)}$ . One checks that  $\tilde{A}\tilde{x} = Ax + b$ , and that for  $\lambda \in [0, 1]$ , if  $x_\lambda = \lambda x_1 + (1 - \lambda)x_0$ , then  $\tilde{x}_\lambda = \lambda \tilde{x}_1 + (1 - \lambda)\tilde{x}_0$ . Therefore, the function  $\tilde{f}: x \mapsto \tilde{A}\tilde{x}$  is linear, and  $g \circ \tilde{f} = g \circ f$  is convex.

## I.2. (Sub-) Gradients

It is assumed known that, if  $E_1$  and  $E_2$  are two convex functions,

$$\partial(E_1 + E_2)(w) = \partial E_1(w) + \partial E_2(w) = \{g_1 + g_2 \mid g_1 \in \partial E_1(w), g_2 \in \partial E_2(w)\}$$

**Exercise I.3** (2 points). Let  $\phi: \mathbb{R}^d \rightarrow \mathbb{R}^p$  be a differentiable basis function. Define the model  $f: \mathbb{R}^d \times \mathbb{R}^p \rightarrow \mathbb{R}$  as, for  $b \in \mathbb{R}$ ,

$$\begin{aligned} f &: \mathbb{R}^d \times \mathbb{R}^p \longrightarrow \mathbb{R} \\ (x, w) &\longmapsto \langle w, \phi(x) \rangle + b \end{aligned}$$

1. What is  $\nabla_w f(x, w)$ ?
2. What is  $\nabla_x f(x, w)$ ?

**Answer I.3.** 1. One computes, for  $(x, w) \in \mathbb{R}^d \times \mathbb{R}^p$ ,  $\nabla_w f(x, w) = \phi(x)$ .

2. The function  $\phi: \mathbb{R}^d \rightarrow \mathbb{R}^p$  is differentiable, let  $J_\phi(x) \in \mathbb{R}^{p \times d}$  be its Jacobian at  $x \in \mathbb{R}^d$ . With the chain-rule for products (cf Ex Sheet#1), one computes for  $(x, w) \in \mathbb{R}^d \times \mathbb{R}^p$ ,  $\nabla_x f(x, w) = J_\phi(x)^\top w$

**Exercise I.4** (2 points). For  $\lambda \geq 0$ , let  $E(w) = E_s(w) + \lambda \|w\|_1$ , where  $E_s$  is assumed to be convex and everywhere differentiable, and

$$\|w\|_1 := \sum_{i=1}^p |w_i|$$

is the 1-norm of  $w$ .

1. Is  $E$  convex?
2. Where is  $w \mapsto \|w\|_1$  (not) differentiable?
3. What is  $\partial E(w)$  (as a function of  $\nabla E_s(w)$ )? (*Hint*: see what happens for  $p = 1, 2$  first.)

**Answer I.4.** 1.  $E$  is convex, as the sum of two convex functions in  $w$ :  $E_s$  and  $w \mapsto \lambda \|w\|_1$  (for  $\lambda \geq 0$ ,  $\lambda f$  is convex if  $f$  is).

The function  $w \mapsto \|w\|_1$  is convex as a sum of convex functions on the coordinates  $w_i$  ( $t \mapsto |t|$  is convex since  $\forall t \in \mathbb{R}$ ,  $|t| = \max(-t, t)$ , and  $t \mapsto \max(f(t), g(t))$  is convex when  $f$  and  $g$  are, see Ex I.1).

2. The function  $\mathbb{R} \ni t \mapsto |t|$  is differentiable on  $(-\infty, 0) \cup (0, +\infty)$ , but not on 0. Therefore,  $\mathbb{R}^p \ni w \mapsto \|w\|_1$  is differentiable at  $w \in \mathbb{R}^p$  if and only if  $\forall i \in [p]$ ,  $w_i \neq 0$ .
3.  $w \mapsto E(w)$  is convex but not differentiable everywhere (since  $w \mapsto \|w\|_1$  is not differentiable everywhere). Since  $E_s$  is differentiable, we always have

$$\partial E(w) = \{\nabla E_s(w)\} + \partial(w \mapsto \lambda \|w\|_1)(w)$$

We are looking for  $\partial(w \mapsto \lambda \|w\|_1)(w) = \lambda \partial \|w\|_1$ . We can see what happens for  $p = 1, 2$ .

- For  $p = 1$ , from Ex Sheet#1,  $\partial|w| = \begin{cases} \{-1\} & \text{if } w < 0, \\ [-1, 1] & \text{if } w = 0, \\ \{1\} & \text{if } w > 0 \end{cases}$ .
- For  $p = 2$ ,  $\partial \|w\|_1 = \partial(|w_1| + |w_2|) = \partial((w_1, w_2) \mapsto |w_1|)(w) + \partial((w_1, w_2) \mapsto |w_2|)(w)$ .  
The subdifferential  $\partial((w_1, w_2) \mapsto |w_1|)(w) \subset \mathbb{R}^2$  can be written

$$\partial((w_1, w_2) \mapsto |w_1|)(w) = \left\{ \begin{pmatrix} g_1 \\ 0 \end{pmatrix} \mid g_1 \in \begin{cases} \{-1\} & \text{if } w_1 < 0, \\ [-1, 1] & \text{if } w_1 = 0, \\ \{1\} & \text{if } w_1 > 0 \end{cases} \right\}.$$

Likewise, the subdifferential  $\partial((w_1, w_2) \mapsto |w_2|)(w) \subset \mathbb{R}^2$  can be written

$$\partial((w_1, w_2) \mapsto |w_2|)(w) = \left\{ \begin{pmatrix} 0 \\ g_2 \end{pmatrix} \mid g_2 \in \begin{cases} \{-1\} & \text{if } w_2 < 0, \\ [-1, 1] & \text{if } w_2 = 0, \\ \{1\} & \text{if } w_2 > 0 \end{cases} \right\}.$$

$$\text{Therefore, } \partial((w_1, w_2) \mapsto \|w\|_1)(w_1, w_2) = \left\{ \begin{pmatrix} g_1 \\ g_2 \end{pmatrix} \mid \forall i \in \{1, 2\}, g_i \in \begin{cases} \{-1\} & \text{if } w_i < 0, \\ [-1, 1] & \text{if } w_i = 0, \\ \{1\} & \text{if } w_i > 0 \end{cases} \right\}$$

This carries out to any dimension  $p$ , so that, for  $w \in \mathbb{R}^p$ ,

$$\partial (w \mapsto \|w\|_1) (w) = \left\{ g \in \mathbb{R}^p \mid \forall i \in [p], g_i \in \begin{cases} \{-1\} & \text{if } w_i < 0, \\ [-1, 1] & \text{if } w_i = 0, \\ \{1\} & \text{if } w_i > 0 \end{cases} \right\}$$

To conclude,

$$\begin{aligned} \partial E(w) &= \{\nabla E_s(w)\} + \lambda \partial \|w\|_1 \\ &= \left\{ \nabla E_w(w) + g \in \mathbb{R}^p \mid \forall i \in [p], g_i \in \begin{cases} \{-\lambda\} & \text{if } w_i < 0, \\ [-\lambda, \lambda] & \text{if } w_i = 0, \\ \{\lambda\} & \text{if } w_i > 0 \end{cases} \right\} \end{aligned}$$

### I.3. Support Vector Machines

**Exercise I.5** (4 points). The Support Vector Machines (SVM) algorithm solves a binary classification task. Given  $N$  couples samples / targets  $\{x_i, t_i\}_{i \in [N]}$ , with  $x_i \in \mathbb{R}^d$  and  $t_i \in \{-1, 1\}$  for each  $i \in [N] := \{1, \dots, N\}$ , the goal is to classify the samples, i.e. find the regions where the positive (resp. negative) samples lie. We will do that by finding a **hyperplane that separates** (or **splits**) the dataset, with positive samples on one side of the hyperplane and the negative on the other. We assume that **such an hyperplane exists** (the samples are said to be *linearly separable*). One can picture the case for  $d = 2$  or  $d = 3$ , where points are clustered in two groups and can be separated by a straight line ( $d = 2$ ) or a plane ( $d = 3$ ), see Figure 1a.

A hyperplane in  $\mathbb{R}^d$  is represented with a vector  $w \in \mathbb{R}^d$  and a *bias*  $b \in \mathbb{R}$  with the equation

$$y(x; w, b) = \langle w, x \rangle + b. \quad (1)$$

For  $i \in [N]$ , denote  $y_i := y(x_i; w, b) = \langle w, x_i \rangle + b$ . Note that  $y_i$  **still depends on**  $(w, b)$  even if the notation is dropped.

The hyperplane equation (1) splits  $\mathbb{R}^d$  into three regions:

- points  $x$  such that  $y(x) > 0$ ,
- points  $x$  such that  $y(x) = 0$  (the hyperplane itself),
- points  $x$  such that  $y(x) < 0$ .

Therefore, we would like to find an hyperplane such that the samples  $x_i$  that have a positive target  $t_i = 1$  all lie on the side of the hyperplane where  $y(x) > 0$ , i.e.  $t_i = 1 \implies y_i > 0$ , and reciprocally all samples  $x_i$  such that  $t_i = -1$  should be on the side where  $y(x) < 0$ , i.e.  $t_i = -1 \implies y_i < 0$ . Then, the target  $t_i$  could simply be read from  $y_i$  by looking at its sign.

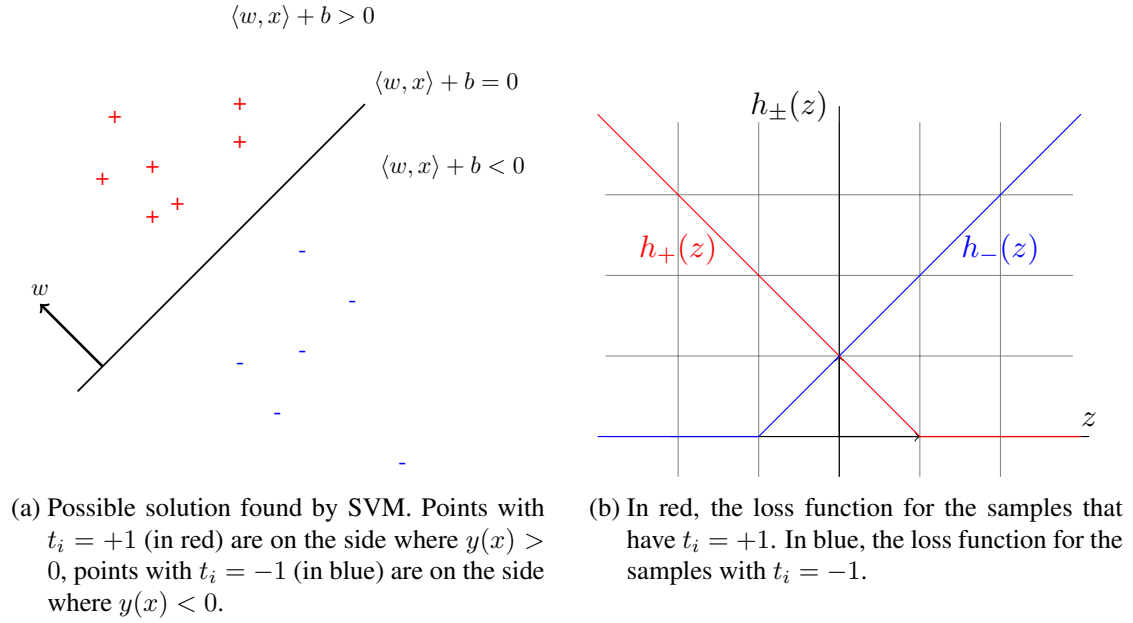


Figure 1: SVM illustration,  $d = 2$ .

With this requirement, the product  $t_i y_i$  should **always be positive**, and the loss we define is

$$\forall (w, b) \in \mathbb{R}^d \times \mathbb{R}, \quad E(w, b) = \sum_{i=1}^N \max(0, 1 - t_i y_i) = \sum_{i=1}^N \max(0, 1 - t_i (\langle w, x_i \rangle + b)), \quad (2)$$

which has the effect of pushing the product  $t_i y_i$  towards the greatest value possible, for all  $i \in [N]$ .

Let  $\mathcal{I}_+ = \{i \in [N] \mid t_i = +1\}$  and  $\mathcal{I}_- = \{i \in [N] \mid t_i = -1\}$  be the sets of the indices of the positive and negative samples. The loss can be further written as

$$\forall (w, b) \in \mathbb{R}^d \times \mathbb{R}, \quad E(w, b) = \sum_{i \in \mathcal{I}_+} \max(0, 1 - y_i) + \sum_{i \in \mathcal{I}_-} \max(0, 1 + y_i) =: \sum_{i \in \mathcal{I}_+} h_+(y_i) + \sum_{i \in \mathcal{I}_-} h_-(y_i)$$

The functions  $h_+$  and  $h_-$  are plotted in Figure 1b.

1. Why is the loss  $(w, b) \mapsto E(w, b)$  convex? (*Hint*: if  $f$  and  $g$  are convex, then  $\max(f, g)$  is convex).

Recall that the function  $h_+ : \mathbb{R} \ni z \mapsto \max(0, 1 - z)$  has the following subgradient (see Exercise Sheet #2, I.2):

$$\partial h_+(z) = \begin{cases} \{-1\} & \text{if } z < 1, \\ [-1, 0] & \text{if } z = 1, \\ \{0\} & \text{if } z > 1. \end{cases}$$

2. Show that the subgradient of the function  $h_- : \mathbb{R} \ni z \mapsto \max(0, 1 + z)$  is

$$\partial h_-(z) = \begin{cases} \{0\} & \text{if } z < -1, \\ [0, 1] & \text{if } z = -1, \\ \{1\} & \text{if } z > -1. \end{cases}$$

3. Using the chain rule for the subgradients  $\partial(g \circ A)(\tilde{w}) = A^\top \partial g(A\tilde{w})$ , for any linear operator  $A \in \mathbb{R}^{m \times p}$ , and convex function  $g : \mathbb{R}^m \rightarrow \mathbb{R}$ , and  $\tilde{w} \in \mathbb{R}^p$ , compute the subgradient of the loss  $E$  with respect to  $w$  and  $b$  at  $(w, b) \in \mathbb{R}^d \times \mathbb{R}$ .

4. What should be the (sub-) gradient descent algorithm to minimize  $E$ ?

**Answer I.5.** 1. • The model  $y_i$  is affine in  $w$ , hence convex

- The functions  $h_+$  and  $h_-$  are the maximum of convex (affine) functions of  $w$ , hence they are convex.
- The loss  $w \mapsto E(w)$  is a sum of convex functions in  $w$ , hence is itself convex in  $w$ .

2. With the chain rule formula,  $h_-(z) = h_+(-z)$ , hence  $\partial h_-(z) = -\partial h_+(-z)$  (!).

Without the chain rule formula, or to practice the definition of a subgradient, let  $h_- : \mathbb{R} \ni z \mapsto \max(0, 1 + z)$ .  $h_-$  is differentiable on  $(-\infty, -1) \cup (-1, +\infty)$ , with derivative 0 on

$$(-\infty, -1) \text{ and } 1 \text{ on } (-1, +\infty), \text{ hence } \partial h_-(z) = \begin{cases} \{0\} & \text{if } z < -1, \\ \{1\} & \text{if } z > -1. \end{cases}$$

At  $z = -1$ ,  $h_-$  is not differentiable. We argue by necessary conditions. We are looking a subgradient  $g \in \mathbb{R}$ , such that, for any  $y \in \mathbb{R}$ ,

$$\begin{aligned} h_-(y) &\geq h_-(-1) + g \cdot (y - (-1)) \\ \iff \max(0, 1 + y) &\geq g \cdot (y + 1) \end{aligned} \tag{3}$$

- At  $y = -1$ , the condition become  $0 \geq g \cdot 0$ , which is true for any  $g \in \mathbb{R}$  (condition is not discriminative).
- For  $y < -1$  (i.e.  $y + 1 < 0$ ), the condition becomes  $0 \geq g \cdot (y + 1) \implies 0 \leq g$ .
- For  $y > -1$  (i.e.  $y + 1 > 0$ ), the condition becomes  $1 + y \geq g \cdot (y + 1) \implies 1 \geq g$ .

Therefore, a subgradient  $g$  of  $h_-$  at  $z = -1$  will have to be such that  $g \in [0, 1]$  (necessary condition).

One checks that, which such a  $g \in [0, 1]$ , the subgradient condition (3) is verified.

Therefore,  $\partial h_-(-1) = [0, 1]$ , and we conclude

$$\partial h_-(z) = \begin{cases} \{0\} & \text{if } z < -1, \\ [0, 1] & \text{if } z = -1, \\ \{1\} & \text{if } z > -1. \end{cases}$$

3. For  $(w, b) \in \mathbb{R}^d \times \mathbb{R}$ , let  $\tilde{w} = \begin{pmatrix} w \\ b \end{pmatrix} \in \mathbb{R}^{d+1}$ , and for each sample  $x_i \in \mathbb{R}^d$ ,  $i \in [N]$ , let  $A_i = \begin{pmatrix} x_i^\top & 1 \end{pmatrix} \in \mathbb{R}^{1 \times (d+1)}$ . Then,

$$A_i \tilde{w} = x_i^\top w + b = y_i(w, b).$$

This is the usual trick to express an affine function as a linear function, with an increment in the dimension. Now,  $E$  can be expressed as a function of  $\tilde{w}$

$$\begin{aligned} E(w, b) = E(\tilde{w}) &= \sum_{i \in \mathcal{I}_+} h_+(y_i) + \sum_{i \in \mathcal{I}_-} h_-(y_i) \\ &= \sum_{i \in \mathcal{I}_+} h_+(A_i \tilde{w}) + \sum_{i \in \mathcal{I}_-} h_-(A_i \tilde{w}) \end{aligned}$$

We can apply the formula for the subdifferential chain rule. For  $(w, b) \in \mathbb{R}^d \times \mathbb{R}$ ,

$$\begin{aligned} \partial E(\tilde{w}) &= \sum_{i \in \mathcal{I}_+} \partial(h_+ \circ A_i)(\tilde{w}) + \sum_{i \in \mathcal{I}_-} \partial(h_- \circ A_i)(\tilde{w}) \\ &= \sum_{i \in \mathcal{I}_+} A_i^\top \partial h_+(A_i \tilde{w}) + \sum_{i \in \mathcal{I}_-} A_i^\top \partial h_-(A_i \tilde{w}) \\ &= \sum_{i \in \mathcal{I}_+} \begin{pmatrix} x_i \\ 1 \end{pmatrix} \partial h_+(y_i) + \sum_{i \in \mathcal{I}_-} \begin{pmatrix} x_i \\ 1 \end{pmatrix} \partial h_-(y_i), \end{aligned} \tag{4}$$

with the subgradients  $\partial h_+(y_i)$  and  $\partial h_-(y_i)$  derived earlier.

4. The subgradient algorithm to minimize  $E$  is as follows.

- a) Start with a zero  $\tilde{w}^{(0)} = \begin{pmatrix} w^{(0)} \\ b^{(0)} \end{pmatrix} = \mathbf{0}_{d+1}$ . Choose a learning rate  $\eta > 0$ .

- b) From a time step  $k$ , update the parameters to  $k + 1$  as

$$\tilde{w}^{(k+1)} = \tilde{w}^{(k)} - \eta \cdot \text{sg}^{(k)},$$

where  $\text{sg}^{(k)} \in \mathbb{R}^{d+1}$  is a subgradient at time  $k$ , computed as  $\text{sg}^{(k)} \in \partial E(\tilde{w}^{(k)})$ . From (4), we see that  $\text{sg}^{(k)} = \sum_{i \in [N]} \text{sg}_i^{(k)}$  is a sum of contributions over all samples  $i \in [N]$  (each  $\text{sg}_i^{(k)} \in \mathbb{R}^{d+1}$ ). More explicitly:

For all  $i \in \mathcal{I}_+$ , compute  $y_i^{(k)} = \langle w^{(k)}, x_i \rangle + b^{(k)}$ . The contribution  $\text{sg}_i^{(k)}$  depends on  $t_i$  and  $y_i^{(k)}$ :

- i. If  $i \in \mathcal{I}_+$  (i.e.  $t_i = +1$ ), then

$$\text{sg}_i^{(k)} = \alpha_i \cdot \begin{pmatrix} x_i \\ 1 \end{pmatrix}, \quad \alpha_i \in \begin{cases} \{-1\} & \text{if } y_i^{(k)} < 1, \\ [-1, 0] & \text{if } y_i^{(k)} = 1, \\ \{0\} & \text{if } y_i^{(k)} > 1. \end{cases} \tag{5}$$

ii. If  $i \in \mathcal{I}_-$  (i.e.  $t_i = -1$ ), then

$$\text{sg}_i^{(k)} = \alpha_i \cdot \begin{pmatrix} x_i \\ 1 \end{pmatrix}, \alpha_i \in \begin{cases} \{0\} & \text{if } y_i^{(k)} < -1, \\ [0, 1] & \text{if } y_i^{(k)} = -1, \\ \{1\} & \text{if } y_i^{(k)} > -1. \end{cases} \quad (6)$$

c) Once a convergence criterion is reached, output  $\tilde{w}^{(K)}$ .

*Remark I.3.1.* Notice how  $\text{sg}_i^{(k)} = t_i \alpha_i \begin{pmatrix} x_i \\ 1 \end{pmatrix}$ ,  $\alpha_i \in \begin{cases} \{-1\} & \text{if } t_i y_i^{(k)} < 1, \\ [-1, 0] & \text{if } t_i y_i^{(k)} = 1, \\ \{0\} & \text{if } t_i y_i^{(k)} > 1. \end{cases}$  one could

have worked with the expression (2) without splitting the samples into  $\mathcal{I}_+$  and  $\mathcal{I}_-$ , but one goal was to practice the definition of the subgradient with  $h_-$ . In addition, the expression with split  $\mathcal{I}_+, \mathcal{I}_-$  has a more detailed interpretation.

## Part II: Programming

8+2 points

**Exercise II.1** (8+2 points). This exercise implements some material from Exercise I.5. The data is generated with the helper function `gen_binary_data` in `ex02/Utils.py`. The dimension is set to  $d = 2$  in order to visualize the result at the end. The generation simply draws some random points on the plane, draws an hyperplane, and classify the points depending on the sign of  $\langle w, x \rangle + b$ . Therefore, the training data is linearly separable.

1. Implement the loss from (2).
2. Implement the (sub-) gradient algorithm derived in I.5.4
3. Visualize the solution found by the algorithm, as well as its convergence.
4. *Bonus*: What happens if the data is not linearly separable?