

# Machine Learning and Neural Networks III (MATH3431)

## Epiphany term

Georgios P. Karagiannis

georgios.karagiannis@durham.ac.uk

Department of Mathematical Sciences (Office MCS3088)  
Durham University  
Stockton Road Durham DH1 3LE UK

2024/01/11 at 16:37:17

### Concepts

- Convex learning problems
- Stochastic learning
- Support vector machines
- Artificial neural networks
- Kernel methods
- Gaussian process regression



# Reading list

These lecture Handouts have been derived based on the above reading list.

## Main texts:

- Bishop, C. M. (2006). Pattern recognition and machine learning. New York: Springer.
  - It is a classical textbook in machine learning (ML) methods. It discusses all the concepts introduced in the course (not necessarily in the same depth). It is one of the main textbooks in the module. The level on difficulty is easy.
  - Students who wish to have a textbook covering traditional concepts in machine learning are suggested to get a copy of this textbook. It is available online from the Microsoft's website <https://www.microsoft.com/en-us/research/publication/pattern-recognition>
- Shalev-Shwartz, S., & Ben-David, S. (2014). Understanding machine learning: From theory to algorithms. Cambridge university press.
  - It has several elements of theory about machine learning algorithms. It is one of the main textbooks in the module. The level on difficulty is advanced as it requires moderate knowledge of maths.
- Bishop, C. M. (1995). Neural networks for pattern recognition. Oxford university press.
  - It is a classical textbook about 'traditional' artificial neural networks (ANN). It is very comprehensive (compared to others) and it goes deep enough for the module although it may be a bit outdated. It is one of the main textbooks in the module for ANN. The level on difficulty is moderate.

## Supplementary textbooks:

- Ripley, B. D. (2007). Pattern recognition and neural networks. Cambridge university press.
  - A classical textbook in artificial neural networks (ANN) that also covers other machine learning concepts. It contains interesting theory about ANN.
  - It is suggested to be used as a supplementary reading for neural networks as it contains a few interesting theoretical results. The level on difficulty is moderate.
- Williams, C. K., & Rasmussen, C. E. (2006). Gaussian processes for machine learning (Vol. 2, No. 3, p. 4). Cambridge, MA: MIT press.

- A classic book in Gaussian process regression (GPR) that covers the material we will discuss in the course about GPR. It can be used as a companion textbook with that of (Bishop, C. M., 2006). The level on difficulty is easy.
- Murphy, K. P. (2012). Machine learning: a probabilistic perspective. MIT press.
  - A popular textbook in machine learning methods. It discusses all the concepts introduced in the module. It focuses more on the probabilistic/Bayesian framework but not with great detail. It can be used as a comparison textbook for brief reading about ML methods just to see another perspective than that in (Bishop, C. M., 2006). The level on difficulty is easy.
- Murphy, K. P. (2022). Probabilistic machine learning: an introduction. MIT press.
  - A textbook in machine learning methods. It covers a smaller number of ML concepts than (Murphy, K. P., 2012) but it contains more fancy/popular topics such as deep learning ideas. It is suggested to be used in the same manner as (Murphy, K. P., 2012). The level on difficulty is easy.
- Barber, D. (2012). Bayesian reasoning and machine learning. Cambridge University Press.
  - A textbook in machine learning methods from a Bayesian point of view. It discusses all the concepts introduced apart from ANN and stochastic gradient algorithms. It aims to be more ‘statistical’ than those of Murphy and Bishop. The level on difficulty is easy.
- Devroye, L., Györfi, L., & Lugosi, G. (2013). A probabilistic theory of pattern recognition (Vol. 31). Springer Science & Business Media.
  - Theoretical aspects about machine learning algorithms. The level on difficulty is advanced as it requires moderate knowledge of probability.

## Handout 0: Machine learning –A recap on: definitions, notation, and formulation

Lecturer &amp; author: Georgios P. Karagiannis

georgios.karagiannis@durham.ac.uk

**Aim.** To get some definitions and set-up about the learning procedure; essentially to formalize what introduced in term 1.

### Reading list & references:

- Bishop, C. M. (2006). Pattern recognition and machine learning. New York: Springer.
  - Ch. 1 Introduction
- Shalev-Shwartz, S., & Ben-David, S. (2014). Understanding machine learning: From theory to algorithms. Cambridge university press.
  - Ch. 1 Introduction

### 1. GENERAL INTRODUCTIONS AND LOOSE DEFINITIONS

**Pattern recognition** is the automated discovery of patterns and regularities in data  $z \in \mathcal{Z}$ . **Machine learning (ML)** are statistical procedures for building and understanding probabilistic methods that 'learn'. **ML algorithms**  $\mathfrak{A}$  build a (probabilistic/deterministic) model able to make predictions or decisions with minimum human interference and can be used for pattern recognition. **Learning** (or training, estimation, fitting) is called the procedure where the ML model is tuned. **Training data** (or observations, sample data set, examples) is a set of observables  $\{z_i \in \mathcal{Z}\}$  used to tune the parameters of the ML model. By  $\mathcal{Z}$  we denote the examples (or observables) domain. **Test set** is a set of available examples/observables  $\{z'_i\}$  (different than the training data) used to verify the performance of the ML model for a given a measure of success. **Measure of success** (or performance) is a quantity that indicates how bad the corresponding ML model or Algorithm performs (eg quantifies the failure/error), and can also be used for comparisons among different ML models; eg, **Risk function** or **Empirical Risk Function**. Two main problems in ML are the supervised learning (we will focus on this here) and the unsupervised learning.

**Supervised learning** problems involve applications where the training data  $z \in \mathcal{Z}$  comprise examples of the input vectors  $x \in \mathcal{X}$  along with their corresponding target vectors  $y \in \mathcal{Y}$ ; i.e.  $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ . By  $\mathcal{X}$  we denote the inputs (or instances) domain, and by  $\mathcal{Y}$  we denote the target domain. **Classification problems** are those which aim to assign each input vector  $x$  to one of a finite number of discrete categories of  $y$ . **Regression problems** are those where the output  $y$  consists of one or more continuous variables. All in all, the learner wishes to discover an unknown pattern (i.e. functional relationship) between components  $x \in \mathcal{X}$  that serves as inputs and components  $y \in \mathcal{Y}$  that act as outputs; i.e.  $x \mapsto y$ . Hence,  $\mathcal{X}$  is the input domain, and  $\mathcal{Y}$  is the output (or target) domain. The goal of learning is to discover a function which predicts (or help us make decisions about)  $y \in \mathcal{Y}$  from  $x \in \mathcal{X}$ .

**Unsupervised learning** problems involve applications where the training data  $z \in \mathcal{Z}$  consist of a set of input vectors  $x \in \mathcal{X}$  without any corresponding target values ; i.e.  $\mathcal{Z} = \mathcal{X}$ . In clustering the goal is to discover groups of similar examples within the data of it is to discover groups of similar examples within the data.

## 2. (LOOSE) NOTATION & DEFINITIONS IN LEARNING

**Definition 1.** The learner's output is a function,  $h : \mathcal{X} \rightarrow \mathcal{Y}$  which predicts  $y \in \mathcal{Y}$  from  $x \in \mathcal{X}$ . It is also called hypothesis, prediction rule, predictor, or classifier.

*Notation 2.* We often denote the set of hypothesis as  $\mathcal{H}$  ; i.e.  $h \in \mathcal{H}$ .

**Example 3.** (Linear Regression)<sup>1</sup> Consider the regression problem where the goal is to learn the mapping  $x \rightarrow y$  where  $x \in \mathcal{X} \subseteq \mathbb{R}^d$  and  $y \in \mathcal{Y} \subseteq \mathbb{R}$ . A hypothesis is a linear function  $h : \mathcal{X} \rightarrow \mathcal{Y}$  (that learner wishes to learn) to approximate mapping  $x \rightarrow y$ . The hypothesis set  $\mathcal{H} = \{x \rightarrow \langle w, x \rangle : w \in \mathbb{R}^d\}$ . We can use the loss  $\ell(h, (x, y)) = (h(x) - y)^2$ .

**Definition 4.** Training data set  $\mathcal{S}$  of size  $m$  is any finite sequence of pairs  $(z_i = (x_i, y_i) ; i = 1, \dots, m)$  in  $\mathcal{X} \times \mathcal{Y}$ . This is the information that the learner has assess.

**Definition 5.** Data generation model  $g(\cdot)$  is the probability distribution over  $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ , unknown to the learner that has generated the data. E.g.  $z \sim g$ .

**Definition 6.** We denote as  $\mathfrak{A}(\mathcal{S})$  the hypothesis (outcome) that a learning algorithm  $\mathfrak{A}$  returns given training sample  $\mathcal{S}$ .

**Definition 7.** (Loss function) Given any set of hypothesis  $\mathcal{H}$  and some domain  $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ , a loss function  $\ell(\cdot)$  is any function  $\ell : \mathcal{H} \times \mathcal{Z} \rightarrow \mathbb{R}_+$ . Loss function  $\ell(h, z)$  specified according to the purpose to quantify the “error” for a given hypothesis  $h$  and a given example  $z$  –the greater the error the greater the value of the loss.

**Example 8.** (Cont. Example 3) In regression problems  $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$  and  $\mathcal{Y} \subset \mathbb{R}$  is uncountable, a potential loss function is

$$\ell_{\text{sq}}(h, (x, y)) = (h(x) - y)^2$$

**Example 9.** In binary classification problems with  $h : \mathcal{X} \rightarrow \mathcal{Y}$  a learner where  $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$  and  $\mathcal{Y} = \{0, 1\}$  is discrete, a loss function can be

$$\ell_{0-1}(h, (x, y)) = 1(h(x) \neq y),$$

**Definition 10.** (Risk function) The risk function  $R_g(h)$  of  $h$  is the expected loss of the hypothesis  $h \in \mathcal{H}$ , w.r.t. the data generation model (which is a probability distribution)  $g$  over domain  $\mathcal{Z}$ ; i.e.

$$(2.1) \quad R_g(h) = \mathbb{E}_{z \sim g}(\ell(h, z))$$

---

<sup>1</sup> $\langle w, x \rangle = w^\top x$

*Remark 11.* In learning, an ideal way to obtain an optimal predictor  $h^*$  is to compute the minimizer of the risk; i.e.

$$h^* = \arg \min_{\forall h} (R_g(h))$$

**Example 12.** (Cont. Ex. 8) The risk function is  $R_g(h) = \mathbb{E}_{z \sim g} (h(x) - y)^2$ , and it measures the quality of the hypothesis function  $h : \mathcal{X} \rightarrow \mathcal{Y}$ , (or equiv. the validity of the class of hypotheses  $\mathcal{H}$ ) against the data generating model  $g$ , as the expected square difference between the predicted values from  $h$  and the true target values  $y$  at every  $x$ .

*Note 13.* Computing the risk minimizer may be practically challenging due to the integration w.r.t. the unknown data generation model  $g$  involved in the expectation (2.1). Sub-optimally, one may resort to the Empirical risk function.

**Definition 14.** (Empirical risk function) The empirical risk function  $\hat{R}_S(h)$  of  $h$  is the expectation of loss of  $h$  over a given sample  $S = (z_1, \dots, z_m) \in \mathcal{Z}^m$ ; i.e.

$$\hat{R}_S(h) = \frac{1}{m} \sum_{i=1}^m \ell(h, z_i).$$

**Example 15.** (Cont. Example 12) Given given sample  $S = \{(x_i, y_i); i = 1, \dots, m\}$  the empirical risk function is  $\hat{R}_S(h) = \frac{1}{m} \sum_{i=1}^m (h(x_i) - y_i)^2$ .

**Example 16.** Consider a learning problem where the true data generation distribution (unknown to the learner) is  $g(z)$ , the statistical model (known to the learner) is given by a sampling distribution  $f_\theta(y) := f(y|\theta)$  labeled by an unknown parameter  $\theta$ . The goal is to learn  $\theta$ . If we assume loss function

$$\ell(\theta, z) = \log \left( \frac{g(z)}{f_\theta(z)} \right)$$

then the risk is

$$(2.2) \quad R_g(\theta) = \mathbb{E}_{z \sim g} \left( \log \left( \frac{g(z)}{f_\theta(z)} \right) \right) = \mathbb{E}_{z \sim g} (\log(g(z))) - \mathbb{E}_{z \sim g} (\log(f_\theta(z)))$$

whose minimizer is

$$\theta^* = \arg \min_{\forall \theta} (R_g(\theta)) = \arg \min_{\forall \theta} (\mathbb{E}_{z \sim g} (-\log(f_\theta(z))))$$

as the first term in (2.2) is constant. Note that in the Maximum Likelihood Estimation technique the MLE  $\theta_{\text{MLE}}$  is the minimizer

$$\theta_{\text{MLE}} = \arg \min_{\theta} \left( \frac{1}{m} \sum_{i=1}^m (-\log(f_\theta(z_i))) \right)$$

where  $S = \{z_1, \dots, z_m\}$  is an IID sample from  $g$ . Hence, MLE  $\theta_{\text{MLE}}$  can be considered as the minimizer of the empirical risk  $R_S(\theta) = \frac{1}{m} \sum_{i=1}^m (-\log(f_\theta(z_i)))$ .

**Definition 17.** A learning problem with hypothesis class  $\mathcal{H}$ , examples domain  $\mathcal{Z}$ , and loss function  $\ell$  may be denoted with a triplet  $(\mathcal{H}, \mathcal{Z}, \ell)$ .

**Example 18.** Consider the multiple linear regression problem with regressors  $x \in \mathcal{X} \subseteq \mathbb{R}^d$  and response  $y \in \mathcal{Y} \subseteq \mathbb{R}$ . Till now, we set the learning problem  $(\mathcal{H}, \mathcal{Z}, \ell)$  in the linear regression with hypothesis class  $\mathcal{H} = \{x \rightarrow \langle w, x \rangle : w \in \mathbb{R}^d\}$ , loss  $\ell(h, (x, y)) = (h(x) - y)^2$ , and examples domain  $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$  with  $\mathcal{X} \subseteq \mathbb{R}^d$  and  $\mathcal{Y} \subseteq \mathbb{R}$ . Because learning problem involves only linear functions as predictors  $h(x) = \langle w, x \rangle$ , this learning problem could be defined equivalently (for presentation convenience) with a hypothesis class  $\mathcal{H} = \{w \in \mathbb{R}^d\}$  and loss function  $\ell(w, (x, y)) = (\langle w, x \rangle - y)^2$ . The latter will be mainly used.

## APPENDIX A. USEFUL THINGS

Below are some standard notation used as default in the notes except in cases that is defined otherwise.

- $q$ -norm: When  $x \in \mathbb{R}^d$   $\|x\|_q := \left(\sum_{j=1}^d x_j^q\right)^{1/q}$
- Manhattan norm: When  $x \in \mathbb{R}^d$   $\|x\|_1 := \sum_{j=1}^d |x_j|$
- Euclidean norm: When  $x \in \mathbb{R}^d$   $\|x\|_2 := \sqrt{\sum_{j=1}^d x_j^2}$ . When  $\|\cdot\|$  we will assume the Euclidean norm.
- Infinity norm or maximum norm:  $\|x\|_\infty := \max_{\forall j} |x_j|$
- Inner product of  $x, y$ : If  $x, y \in \mathbb{R}^d$  then  $\langle x, y \rangle = x^\top y$ . So  $\langle x, x \rangle = \|x\|^2$

Also some standard formulas.

- Jensens' inequality: If  $x \in \mathbb{R}^d$  and  $f : \mathbb{R}^d \rightarrow \mathbb{R}$  then

$$\begin{cases} f(E(x)) \leq E(f(x)) & \text{if } f \text{ is convex} \\ f(E(x)) \geq E(f(x)) & \text{if } f \text{ is concave} \end{cases}$$

- Cauchy-Schwarz inequality: If  $x, y \in \mathbb{R}^d$  then  $|\langle x, y \rangle|^2 \leq \langle x, x \rangle \langle y, y \rangle$  equiv.  $|\langle x, y \rangle| \leq \|x\| \|y\|$ .

# Handout 1: Elements of convex learning problems

Lecturer &amp; author: Georgios P. Karagiannis

georgios.karagiannis@durham.ac.uk

**Aim.** To introduce elements of convexity, Lipschitzness, and smoothness that can be used for the analysis of stochastic gradient related learning algorithms.

## Reading list & references:

- Shalev-Shwartz, S., & Ben-David, S. (2014). Understanding machine learning: From theory to algorithms. Cambridge university press.  
– Ch. 12 Convex Learning Problems

Further reading

- Bishop, C. M. (2006). Pattern recognition and machine learning. New York: Springer.

## 1. MOTIVATIONS

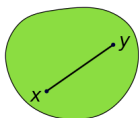
*Note 1.* We introduce concepts of convexity and smoothness that facilitate the analysis and understanding of the learning problems and their solutions that we will discuss (eg stochastic gradient descent, SVM) later on. Also learning problems with such characteristics can be learned more efficiently.

*Note 2.* Some of the ML problems discussed in the course (eg, Artificial neural networks, Gaussian process regression) are non-convex. To overcome this problem, we will introduce the concept of surrogate loss function that allows a non-convex problem to be handled with the tools introduced in the convex setting.

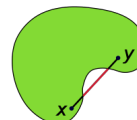
## 2. CONVEXITY

**Definition 3.** A set  $C$  is convex if for any  $u, v \in C$ , the line segment between  $u$  and  $v$  is contained in  $C$ . Namely,

- for any  $u, v \in C$  and for any  $\alpha \in [0, 1]$  we have that  $\alpha u + (1 - \alpha)v \in C$ .



A convex set

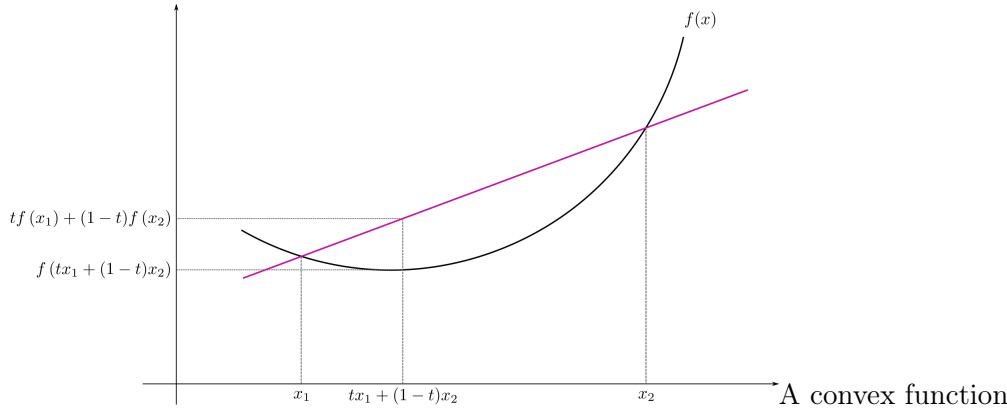


A non-convex set

**Definition 4.** Let  $C$  be a convex set. A function  $f : C \rightarrow \mathbb{R}$  is convex function if for any  $u, v \in C$  and for any  $\alpha \in [0, 1]$

$$f(\alpha u + (1 - \alpha)v) \leq \alpha f(u) + (1 - \alpha)f(v)$$





**Example 5.** The function  $f : \mathbb{R} \rightarrow \mathbb{R}_+$  with  $f(x) = x^2$  is convex function. For any  $u, v \in C$  and for any  $\alpha \in [0, 1]$  it is

$$(\alpha u + (1 - \alpha) v)^2 - \alpha (u)^2 + (1 - \alpha) (v)^2 = -\alpha (1 - \alpha) (u - v)^2 \leq 0$$

**Proposition 6.** Every local minimum of a convex function is the global minimum.

**Proposition 7.** Let  $f : C \rightarrow \mathbb{R}$  be convex function. The tangent of  $f$  at  $w \in C$  is below  $f$ , namely

$$\forall u \in C \quad f(u) \geq f(w) + \langle \nabla f(w), u - w \rangle$$

**Proposition 8.** Let  $f : \mathbb{R}^d \rightarrow \mathbb{R}$  such that  $f(w) = g(\langle w, x \rangle + y)$  for some  $x \in \mathbb{R}^d$ ,  $y \in \mathbb{R}$ . If  $g$  is convex function then  $f$  is convex function.

*Proof.* See Exercise 1 in the Exercise sheet. □

**Example 9.** Consider the regression problem with regressor  $x \in \mathbb{R}^d$ , and response  $y \in \mathbb{R}$  and predictor rule  $h(x) = \langle w, x \rangle$ . The loss function  $\ell(w, (x, y)) = (\langle w, x \rangle - y)^2$  is convex because  $g(a) = (a)^2$  is convex and Proposition 8.

**Proposition 10.** Let  $f_j : \mathbb{R}^d \rightarrow \mathbb{R}$  convex functions for  $j = 1, \dots, r$ . Then:

- (1)  $g(x) = \max_{\forall j} (f_j(x))$  is a convex function
- (2)  $g(x) = \sum_{j=1}^r w_j f_j(x)$  is a convex function where  $w_j > 0$

**Solution.**

- (1) For any  $u, v \in \mathbb{R}^d$  and for any  $\alpha \in [0, 1]$

$$\begin{aligned}
 g(\alpha u + (1 - \alpha) v) &= \max_{\forall j} (f_j(\alpha u + (1 - \alpha) v)) \\
 &\leq \max_{\forall j} (\alpha f_j(u) + (1 - \alpha) f_j(v)) && (f_j \text{ is convex}) \\
 &\leq \alpha \max_{\forall j} (f_j(u)) + (1 - \alpha) \max_{\forall j} (f_j(v)) && (\max(\cdot) \text{ is convex}) \\
 &\leq \alpha g(u) + (1 - \alpha) g(v)
 \end{aligned}$$

(2) For any  $u, v \in \mathbb{R}^d$  and for any  $\alpha \in [0, 1]$

$$\begin{aligned}
 g(\alpha u + (1 - \alpha)v) &= \sum_{j=1}^r w_j f_j(\alpha u + (1 - \alpha)v) \\
 &\leq \alpha \sum_{j=1}^r w_j f_j(u) + (1 - \alpha) \sum_{j=1}^r w_j f_j(v) \quad (f_j \text{ is convex}) \\
 &\leq \alpha g(u) + (1 - \alpha) g(v)
 \end{aligned}$$

**Example 11.**  $g(x) = |x|$  is convex according to Example 10, as  $g(x) = |x| = \max(-x, x)$ .

### 3. LIPSCHITZNESS

**Definition 12.** Let  $C \in \mathbb{R}^d$ . Function  $f : \mathbb{R}^d \rightarrow \mathbb{R}^k$  is  $\rho$ -Lipschitz over  $C$  if for every  $w_1, w_2 \in C$  we have that

$$(3.1) \quad \|f(w_1) - f(w_2)\| \leq \rho \|w_1 - w_2\|. \quad \text{Lipschitz condition}$$

*Conclusion 13.* That means: a Lipschitz function  $f(x)$  cannot change too drastically wrt  $x$ .

**Example 14.** Consider the function  $f : \mathbb{R} \rightarrow \mathbb{R}_+$  with  $f(x) = x^2$ .

(1)  $f$  is not a  $\rho$ -Lipschitz in  $\mathbb{R}$ .

(2)  $f$  is a  $\rho$ -Lipschitz in  $C = \{x \in \mathbb{R} : |x| < \rho/2\}$ .

$$|f(x_2) - f(x_1)| = |x_2^2 - x_1^2| = |(x_2 + x_1)(x_2 - x_1)| \leq 2\rho/2 (x_2 - x_1) = \rho |x_2 - x_1|$$

**Solution.**

(1) For  $x_1 = 0$  and  $x_2 = 1 + \rho$ , it is

$$|f(x_2) - f(x_1)| = (1 + \rho)^2 > \rho(1 + \rho) = \rho |x_2 - x_1|$$

(2) It is

$$|f(x_2) - f(x_1)| = |x_2^2 - x_1^2| = |(x_2 + x_1)(x_2 - x_1)| \leq 2\rho/2 (x_2 - x_1) = \rho |x_2 - x_1|$$

**Theorem 15.** Let functions  $g_1$  be  $\rho_1$ -Lipschitz and  $g_2$  be  $\rho_2$ -Lipschitz. Then  $f$  with  $f(x) = g_1(g_2(x))$  is  $\rho_1\rho_2$ -Lipschitz.

**Solution.** See Exercise 2 from the exercise sheet

**Example 16.** Let functions  $g$  be  $\rho$ -Lipschitz. Then  $f$  with  $f(x) = g(\langle v, x \rangle + b)$  is  $(\rho|v|)$ -Lipschitz.

**Solution.** It is

$$\begin{aligned}
 |f(w_1) - f(w_2)| &= |g(\langle v, w_1 \rangle + b) - g(\langle v, w_2 \rangle + b)| \leq \rho |\langle v, w_1 \rangle + b - \langle v, w_2 \rangle - b| \\
 &\leq \rho |v^\top w_1 - v^\top w_2| \leq \rho |v| |w_1 - w_2|
 \end{aligned}$$

*Note 17.* So, given Examples 14 and 16, in the linear regression setting using loss  $\ell(w, z = (x, y)) = (w^\top x - y)^2$ , the loss function is  $\rho$ -Lipschitz for a given  $z = (x, y)$  and bounded  $\|w\| < \rho$ .

#### 4. SMOOTHNESS

**Definition 18.** A differentiable function  $f : \mathbb{R}^d \rightarrow \mathbb{R}$  is  $\beta$ -smooth if its gradient is  $\beta$ -Lipschitz; namely for all  $v, w \in \mathbb{R}^d$

$$(4.1) \quad \|\nabla f(w_1) - \nabla f(w_2)\| \leq \beta \|w_1 - w_2\|.$$

**Theorem 19.** Function  $f : \mathbb{R}^d \rightarrow \mathbb{R}$  is  $\beta$ -smooth iff

$$(4.2) \quad f(v) \leq f(w) + \langle \nabla f(w), v - w \rangle + \frac{\beta}{2} \|v - w\|^2$$

*Remark 20.* If  $f : \mathbb{R}^d \rightarrow \mathbb{R}$  is  $\beta$ -smooth then (4.2) holds, and if it is convex as well then

$$f(v) \geq f(w) + \langle \nabla f(w), v - w \rangle$$

holds. Hence together these conditions imply upper and lower bounds

$$f(v) - f(w) \in \left( \langle \nabla f(w), v - w \rangle, \langle \nabla f(w), v - w \rangle + \frac{\beta}{2} \|v - w\|^2 \right)$$

**Example 21.** If  $f : \mathbb{R}^d \rightarrow \mathbb{R}$  is  $\beta$ -smooth then for  $v, w \in \mathbb{R}^d$  such that  $v = w - \frac{1}{\beta} \nabla f(w)$  then by (4.2), it is

$$\frac{1}{2\beta} \|\nabla f(w)\|^2 \leq f(w) - f(v)$$

If additionally  $f(x) > 0$  for all  $x \in \mathbb{R}^d$  then

$$\|\nabla f(w)\|^2 \leq 2\beta f(w)$$

which provides assumptions to bound the gradient.

**Theorem 22.** Let  $f : \mathbb{R}^d \rightarrow \mathbb{R}$  with  $f(w) = g(\langle w, x \rangle + y)$   $x \in \mathbb{R}^d$  and  $y \in \mathbb{R}$ . Let  $g : \mathbb{R} \rightarrow \mathbb{R}$  be a  $\beta$ -smooth function. Then  $f$  is a  $(\beta \|x\|^2)$ -smooth.

*Proof.* See Exercise 3 from the Exercise sheet □

**Example 23.** Let  $f(w) = (\langle w, x \rangle + y)^2$  for  $x \in \mathbb{R}^d$  and  $y \in \mathbb{R}$ . Then  $f$  is  $(2\|x\|^2)$ -smooth.

**Solution.** It is  $f(w) = g(\langle w, x \rangle + y)$  for  $g(a) = a^2$ .  $g$  is 2-smooth since

$$\|g'(w_1) - g'(w_2)\| = \|2w_1 - 2w_2\| \leq 2\|w_1 - w_2\|.$$

Hence from (22),  $f$  is  $(2\|x\|^2)$ -smooth.

**Example 24.** Consider the regression problem with predictor rule  $h(x) = \langle w, x \rangle$ , loss function  $\ell(w, (x, y)) = (\langle w, x \rangle - y)^2$ , feature  $x \in \mathbb{R}^d$ , and target  $y \in \mathbb{R}$ . Then  $\ell(w, \cdot)$  is  $(2\|x\|^2)$ -smooth.

**Solution.** Follows from Example 23.

#### 5. CONVEX LEARNING PROBLEMS

**Definition 25.** Convex learning problem is a learning problem  $(\mathcal{H}, \mathcal{Z}, \ell)$  that the hypothesis class  $\mathcal{H}$  is a convex set, and the loss function  $\ell$  is a convex function for each example  $z \in \mathcal{Z}$ .

**Example 26.** Consider the regression problem with predictor rule  $h(x) = \langle w, x \rangle$ , loss function  $\ell(w, (x, y)) = (\langle w, x \rangle - y)^2$ , feature  $x \in \mathbb{R}^d$ , and target  $y \in \mathbb{R}$ . This imposes a convex learning problem due to Example 10.

**Definition 27.** Convex-Lipschitz-Bounded Learning Problem  $(\mathcal{H}, \mathcal{Z}, \ell)$  with parameters  $\rho$ , and  $B$ , is called the learning problem whose the hypothesis class  $\mathcal{H}$  is a convex set, for all  $w \in \mathcal{H}$  it is  $\|w\| \leq B$ , and the loss function  $\ell(\cdot, z)$  is convex and  $\rho$ -Lipschitz function for all  $z \in \mathcal{Z}$ .

**Example 28.** Consider the regression problem with predictor rule  $h(x) = \langle w, x \rangle$ , loss function  $\ell(w, (x, y)) = (\langle w, x \rangle - y)^2$ , feature  $x \in \mathbb{R}^d$ , and target  $y \in \mathbb{R}$ . This imposes a Convex-Lipschitz-Bounded Learning Problem if  $\mathcal{H} = \{w \in \mathbb{R}^d : \|w\| \leq B\}$  due to Examples 10, and 14(2).

**Definition 29.** Convex-Smooth-Bounded Learning Problem  $(\mathcal{H}, \mathcal{Z}, \ell)$  with parameters  $\beta$ , and  $B$ , is called the learning problem whose the hypothesis class  $\mathcal{H}$  is a convex set, for all  $w \in \mathcal{H}$  it is  $\|w\| \leq B$ , and the loss function  $\ell(\cdot, z)$  is convex, nonnegative, and  $\beta$ -smooth function for all  $z \in \mathcal{Z}$ .

**Example 30.** Consider the regression problem with predictor rule  $h(x) = \langle w, x \rangle$ , loss function  $\ell(w, (x, y)) = (\langle w, x \rangle - y)^2$ , feature  $x \in \mathbb{R}^d$ , and target  $y \in \mathbb{R}$ . This imposes a Convex-Smooth-Bounded Learning Problem if  $\mathcal{H} = \{w \in \mathbb{R}^d : \|w\| \leq B\}$  due to Examples 10, and 24.

## 6. NON-CONVEX LEARNING PROBLEMS (SURROGATE TREATMENT)

*Remark 31.* A learning problem may involve non-convex loss function  $\ell(w, z)$  which implies a non-convex risk function  $R_g(w)$ . However, our learning algorithm will be analyzed in the convex setting. A suitable treatment to overcome this difficulty would be to upper bound the non-convex loss function  $\ell(w, z)$  by a convex surrogate loss function  $\tilde{\ell}(w, z)$  for all  $w$ , and use  $\tilde{\ell}(w, z)$  instead of  $\ell(w, z)$ .

**Example 32.** Consider the binary classification problem with inputs  $x \in \mathcal{X}$ , outputs  $y \in \{-1, +1\}$ ; we need to learn  $w \in \mathcal{H}$  from hypothesis class  $\mathcal{H} \subset \mathbb{R}^d$  with respect to the loss

$$\ell(w, (x, y)) = 1_{(y\langle w, x \rangle \leq 0)}$$

with  $y \in \mathbb{R}$ , and  $x \in \mathbb{R}^d$ . Here  $\ell(\cdot)$  is non-convex. A convex surrogate loss function can be

$$\tilde{\ell}(w, (x, y)) = \max(0, 1 - y\langle w, x \rangle)$$

which is convex (Example 10) wrt  $w$ . Note that:

- $\tilde{\ell}(w, (x, y))$  is convex wrt  $w$  ; because  $\max(\cdot)$  is convex
- $\ell(w, (x, y)) \leq \tilde{\ell}(w, (x, y))$  for all  $w \in \mathcal{H}$

Then we can compute

$$\tilde{w}_* = \arg \min_{\forall x} \left( \tilde{R}_g(w) \right) = \arg \min_{\forall x} \left( \mathbb{E}_{(x, y) \sim g} (\max(0, 1 - y\langle w, x \rangle)) \right)$$

instead of

$$w_* = \arg \min_{\forall x} (R_g(w)) = \arg \min_{\forall x} \left( \mathbb{E}_{(x, y) \sim g} (1_{(y\langle w, x \rangle \leq 0)}) \right)$$

Of course by using the surrogate loss instead of the actual one, we introduce some approximation error in the produced output  $\tilde{w}_* \neq w_*$ .

*Remark 33.* (Intuitions...) Using a convex surrogate loss function instead the convex one, facilitates computations but introduces extra error to the solution. If  $R_g(\cdot)$  is the risk under the non-convex loss,  $\tilde{R}_g(\cdot)$  is the risk under the convex surrogate loss, and  $\tilde{w}_{\text{alg}}$  is the output of the learning algorithm under  $\tilde{R}_g(\cdot)$  then we have the upper bound

$$R_g(\tilde{w}_{\text{alg}}) \leq \underbrace{\min_{w \in \mathcal{H}} (R_g(w))}_{\text{I}} + \underbrace{\left( \min_{w \in \mathcal{H}} (\tilde{R}_g(w)) - \min_{w \in \mathcal{H}} (R_g(w)) \right)}_{\text{II}} + \underbrace{\epsilon}_{\text{III}}$$

where term I is the approximation error measuring how well the hypothesis class performs on the generating model, term II is the optimization error due to the use of surrogate loss instead of the actual non-convex one, and term III is the estimation error due to the use of a training set and not the whole generation model.