Machine Learning and Neural Networks III (MATH3431)

Epiphany term

Georgios P. Karagiannis

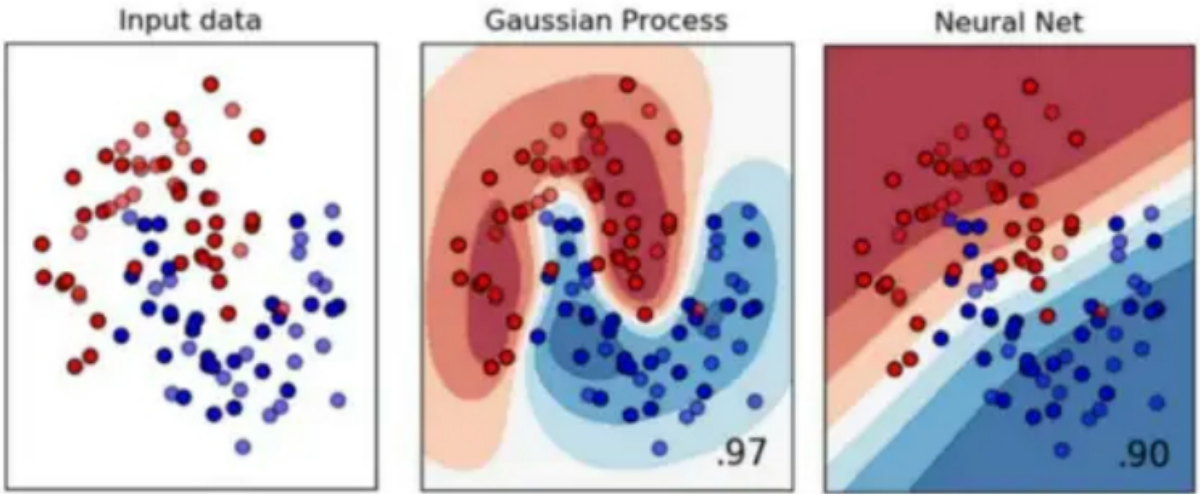
georgios.karagiannis@durham.ac.uk

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

2024/01/22 at 02:07:47

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-recogniti`
- 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 It is very comprehensive (compared to others) and it goes deep enough for the module although it may be a bit outdated.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.

Contents

Machine Learning and Neural Networks (MATH3431) Epiphany term, 2024

Handout 1: Machine learning –A recap on: definitions, notation, and formalism

Lecturer & 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. $x \longmapsto y$. 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, $h : \mathcal{X} \to \mathcal{Y}$ 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)[1] Consider the regression problem where the goal is to learn the mapping $x \to y$ where $x \in \mathcal{X} \subseteq \mathbb{R}^d$ and $y \in \mathcal{Y} \subseteq \mathbb{R}$. A hypothesis is a linear function $h : \mathcal{X} \to \mathcal{Y}$ (that learner wishes to learn) with $h(x) = \langle w, x \rangle$ approximating the mapping $x \to y$. The hypothesis set $\mathcal{H} = \left\{ x \to \langle w, x \rangle : w \in \mathbb{R}^d \right\}$.

Example 4. (Binary Classification) Consider the classification problem where the goal is to learn the mapping $x \to y$ where $x \in \mathcal{X} \subseteq \mathbb{R}^d$ and $y \in \mathcal{Y}\{-1, +1\}$. A hypothesis can be a function $h : \mathcal{X} \to \mathcal{Y}$ with $h(x) = \text{sign}(\langle w, x \rangle)$ approximating the mapping $x \to y$. The hypothesis set $\mathcal{H} = \left\{ x \to \text{sign}(\langle w, x \rangle) : w \in \mathbb{R}^d \right\}$.

Definition 5. Training data set $\mathcal{S}$ of size m is any finite sequence of pairs $(z_i = (x_i, y_i) ; i = 1, ..., m)$ in $\mathcal{X} \times \mathcal{Y}$; i.e. $\mathcal{S} = \{(x_i, y_i) ; i = 1, ..., m\}$. This is the information that the learner has assess.

Definition 6. 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 7. We denote as $\mathfrak{A}(\mathcal{S})$ the hypothesis (outcome) that a learning algorithm $\mathfrak{A}$ returns given training sample S.

Definition 8. (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 $\ell : \mathcal{H} \times \mathcal{Z} \to \mathbb{R}_+$. Loss function $\ell(h, z)$ for $h \in \mathcal{H}$ and $z \in \mathcal{Z}$ is specified according to the purpose the machine learning algorithm. It reflects how the "error" is quantified for a given hypothesis h and a given example z. The rule is "the greater the error the greater the value of the loss".

Example 9. (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 10. (Cont. Example 4) In binary classification problems with hypothesis $h : \mathcal{X} \to \mathcal{Y}$ where $\mathcal{Y} = \{0, 1\}$ is discrete, a loss function can be

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

[1] $\langle w, x \rangle = w^\top x$

Definition 11. (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 Z; i.e.

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

Remark 12. In learning, an ideal way to obtain an optimal predictor h^* is to compute the minimizer of the risk; i.e.

$$h^* = \arg\min_{\forall h}(R_g(h)) \tag{2.2}$$

Example 13. (Cont. Ex. 9) The risk function is $R_g(h) = \mathrm{E}_{z\sim g}(h(x)-y)^2$, and it measures the quality of the hypothesis function $h : \mathcal{X} \to \mathcal{Y}$, (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 form h and the true target values y at every x.

Note 14. 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 use the Empirical risk function instead of the Risk function in (2.2).

Definition 15. (Empirical risk function) The Empirical Risk Function (ERF) $\hat{R}_S(h)$ of h is the expectation of loss of h over a given sample $S = (z_1, ..., z_m) \in \mathcal{Z}^m$; i.e.

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

Remark 16. Given Empirical Risk Function (ERF) $\hat{R}_S(h)$ of h the optimal predictor h^* is the minimizer of the ERF; i.e.

$$h^* = \arg\min_{\forall h}\left(\hat{R}_S(h)\right) \tag{2.3}$$

Example 17. (Cont. Example 13) Given given sample $S = \{(x_i, y_i); i = 1, ..., m\}$ the empirical risk function is $\hat{R}_S(h) = \frac{1}{m}\sum_{i=1}^{m}(h(x_i)-y_i)^2$.

Example 18. (Cont. Example 10) Given given sample $S = \{(x_i, y_i); i = 1, ..., m\}$ the empirical risk function is $\hat{R}_S(h) = \frac{1}{m}\sum_{i=1}^{m}1(h(x_i) \neq y_i)$.

Remark 19. If the Hypothesis set $\mathcal{H}$ is a known parametric family of functions; i.e. $\mathcal{H} = \{h_w(\cdot); w \in \mathcal{W}\}$ parameterized by unknown $w \in \mathcal{W}$, then we can equivalently consider $\mathcal{H} = \{w \in \mathcal{W}\} = \mathcal{W}$ keeping in mind that the learner's output is restricted to $h_w(\cdot)$.

Example 20. Consider the multiple linear regression problem with regressors $x \in \mathcal{X} \subseteq \mathbb{R}^d$ and response $y \in \mathcal{Y} \subseteq \mathbb{R}$. Because it involves only linear functions as predictors $h_w(x) = \langle w, x\rangle$, we could consider a hypothesis class $\mathcal{H} = \left\{w \in \mathbb{R}^d\right\} = \mathbb{R}^d$ and loss function loss $\ell(w,(x,y)) = (\langle w, x\rangle - y)^2$ for computational simplicity. The latter will be mainly used.

Example 21. 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 θ. The goal is to learn θ. If we assume loss function

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

then the risk is

$$R_g(\theta) = \mathrm{E}_{z\sim g}\left(\log\left(\frac{g(z)}{f_\theta(z)}\right)\right) = \mathrm{E}_{z\sim g}(\log(g(z))) - \mathrm{E}_{z\sim g}(\log(f_\theta(z))) \tag{2.4}$$

whose minimizer is

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

as the first term in (2.4) is constant. Note that in the Maximum Likelihood Estimation technique the MLE θ_{MLE} is the minimizer

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

where $S = \{z_1, ..., z_m\}$ is an IID sample from g. Hence, MLE θ_{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 22. A learning problem with hypothesis class $\mathcal{H}$, examples domain $\mathcal{Z}$, and loss function ℓ may be denoted with a triplet $(\mathcal{H}, \mathcal{Z}, \ell)$.

Example 23. The standard multiple linear regression problem with regressors $x \in \mathcal{X} \subseteq \mathbb{R}^d$ and response $y \in \mathcal{Y} \subseteq \mathbb{R}$, is a learning problem with examples domain $\mathcal{Z} = \mathcal{X} \times \mathcal{Y} = \mathbb{R}^{d+1}$, hypothesis class $\mathcal{H} = \left\{x \to \langle w, x\rangle : w \in \mathbb{R}^d\right\}$, and loss function $\ell_{\mathrm{sq}}(h, (x, y)) = (h(x) - y)^2$.

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 If $x \in \mathbb{R}^d$ and $f : \mathbb{R}^d \to \mathbb{R}$ then

$$\begin{cases} f(\mathrm{E}(x)) \le \mathrm{E}(f(x)) & \text{if } f \text{ is convex} \\ f(\mathrm{E}(x)) \ge \mathrm{E}(f(x)) & \text{if } f \text{ is concave} \end{cases}$$

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

Machine Learning and Neural Networks (MATH3431) Epiphany term, 2024

Handout 2: Elements of convex learning problems

Lecturer & 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

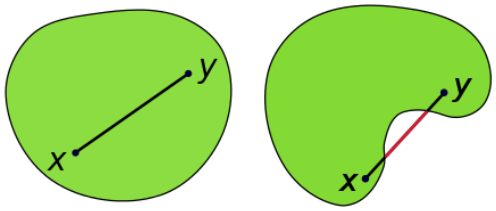
Note 1. We introduce concepts of convexity and smoothness that facilitate the analysis and understanding of the learning problems and their solutions that we will discuss (eg stochastic gradient descent, SVM) later on. Also learning problems with such characteristics can be learned more efficiently.

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 int he convex setting.

2. Convexity

Definition 3. A set C is convex if for any $u, v \in C$ and for any $\alpha \in [0,1]$ we have that $\alpha u + (1-\alpha)\, v \in C$.

Note 4. Namely, a set C is convex if for any $u, v \in C$, the line segment between u and v is contained in C.



(A) (B)

FIGURE 2.1. (2.1a) is a Convex set ; (2.1b) is a non-convex set

Example 5. For instance $\mathbb{R}^d$ for $d \geq 1$ is a convex set.

Definition 6. Let C be a convex set. A function $f : C \to R$ is convex function if for any $u, v \in C$ and for any $\alpha \in [0, 1]$

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

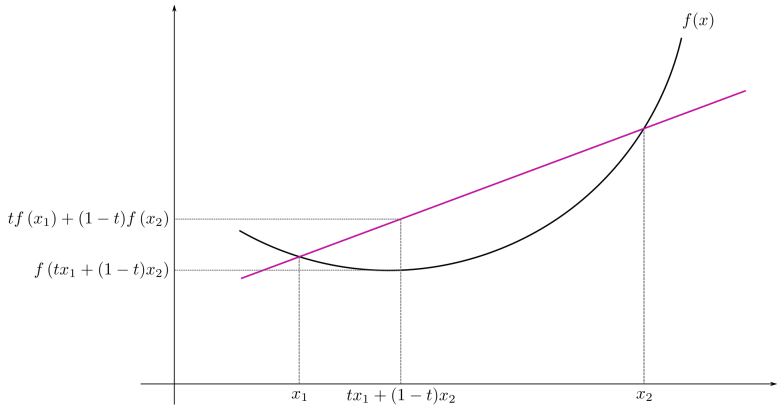


FIGURE 2.2. A convex function

Example 7. The function $f : \mathbb{R} \to \mathbb{R}_+$ with $f(x) = x^2$ is convex function. For any $u, v \in C$ and for any $\alpha \in [0, 1]$ it is

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

Proposition 8. *Every local minimum of a convex function is the global minimum.*

Proposition 9. *Let $f : C \to \mathbb{R}$ be convex function. The tangent of f at $w \in C$ is below f, namely*

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

Proposition 10. *Let $f : \mathbb{R}^d \to \mathbb{R}$ such that $f\left(w\right) = g\left(\left\langle w, x\right\rangle + y\right)$ 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 11. 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 10.

Proposition 12. *Let $f_j : \mathbb{R}^d \to \mathbb{R}$ convex functions for $j = 1, ..., r$. 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 $\alpha \in [0, 1]$

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

Example 13. $g(x) = |x|$ is convex according to Example 12, as $g(x) = |x| = \max(-x, x)$.

3. Lipschitzness

Definition 14. Let $C \in \mathbb{R}^d$. Function $f : \mathbb{R}^d \to \mathbb{R}^k$ is ρ-Lipschitz over C if for every $w_1, w_2 \in C$ we have that

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

Conclusion 15. That means: a Lipschitz function $f(x)$ cannot change too drastically wrt x.

Example 16. Consider the function $f : \mathbb{R} \to \mathbb{R}_+$ with $f(x) = x^2$.

(1) f is not a ρ-Lipschitz in $\mathbb{R}$.
(2) f is a ρ-Lipschitz in $C = \{x \in \mathbb{R} : |x| < \rho/2\}$.

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

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

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

Theorem 17. *Let functions g_1 be ρ_1-Lipschitz and g_2 be ρ_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 18. Let functions g be ρ-Lipschitz. Then f with $f(x) = g(< v, x > +b)$ is $(\rho\,|v|)$-Lipschitz.

Solution. It is

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

Note 19. So, given Examples 16 and 18, in the linear regression setting using loss $\ell(w, z = (x, y)) = \left(w^\top x - y\right)^2$, the loss function is -Lipschitz for a given $z = (x, y)$ and and bounded $\|w\| < \rho$.

4. Smoothness

Definition 20. A differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ is β-smooth if its gradient is β-Lipschitz; namely for all $v, w \in \mathbb{R}^d$

$$\|\nabla f(w_1) - \nabla f(w_2)\| \leq \beta\,\|w_1 - w_2\|\,. \tag{4.1}$$

Theorem 21. *Function $f : \mathbb{R}^d \to \mathbb{R}$ is β-smooth iff*

$$f(v) \leq f(w) + < \nabla f(w), v - w > + \frac{\beta}{2}\,\|v - w\|^2 \tag{4.2}$$

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

Proof. See Exercise 3 from the Exercise sheet □

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

Solution. It is $f(w) = g(< w, x > +y)$ for $g(a) = a^2$. g is 2-smooth since

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

Hence from Theorem 22, f is $\left(2\,\|x\|^2\right)$-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 $\left(2\,\|x\|^2\right)$-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 ℓ 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 Examples 5 and 12.

Definition 27. Convex-Lipschitz-Bounded Learning Problem $(\mathcal{H}, \mathcal{Z}, \ell)$ with parameters ρ, 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 ρ-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 $\mathcal{H} = \left\{w \in \mathbb{R}^d : \|w\| \leq B\right\}$ due to Examples 12, and 16(2).

Definition 29. Convex-Smooth-Bounded Learning Problem $(\mathcal{H}, \mathcal{Z}, \ell)$ with parameters β, 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 β-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 $\mathcal{H} = \left\{w \in \mathbb{R}^d : \|w\| \leq B\right\}$ due to Examples 12, and 24.

Proposition 31. *If ℓ is a convex loss function and the class $\mathcal{H}$ is convex, then the ERM$_\mathcal{H}$ problem, of minimizing the empirical risk $\hat{R}_\mathcal{S}(w)$ over $\mathcal{H}$, is a convex optimization problem (that is, a problem of minimizing a convex function over a convex set).*

Proof. The ERM$_\mathcal{H}$ problem is

$$w^* = \arg\min_{w \in \mathcal{H}} \left\{\hat{R}_\mathcal{S}(w)\right\}$$

given a sample $\mathcal{S} = \{z_1, ..., z_m\}$ for $\hat{R}_\mathcal{S}(w) = \frac{1}{m}\sum_{i=1}^m \ell(w, z_i)$. $\hat{R}_\mathcal{S}(w)$ is a convex function from Proposition (12). Hence ERM rule is a problem of minimizing a convex function subject to the constraint that the solution should be in a convex set. □

Example 32. Multiple linear regression 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 $y \in \mathbb{R}$ where

$$w^* = \arg\min_w \mathrm{E}\,(\langle w, x\rangle - y)^2$$

or

$$w^{**} = \arg\min_w \frac{1}{m}\sum_{i=1}^m (\langle w, x_i\rangle - y_i)^2$$

is a convex learning problem –from Proposition 31.

Note 33. Problems like that in Proposition 31 can be efficiently solved with algorithms such as Stochastic Gradients Descent to be introduced later.

6. Non-convex learning problems (surrogate treatment)

Remark 34. 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 35. 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 12) 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

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

instead of

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

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 36. (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

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

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.

7. Strong convexity

Note 37. Strong convexity is a central concept in regularization, e.g. Ridge, as it makes a convex loss function strongly convex by adding a shrinkage term.

Definition 38. (Strongly convex functions) A function f is λ-strongly convex function is for all w, u, and $\alpha \in (0,1)$ we have

$$f(aw + (1-\alpha)u) \leq af(w) + (1-\alpha)f(u) - \frac{\lambda}{2}\alpha(1-\alpha)\|w-u\|^2 \tag{7.1}$$

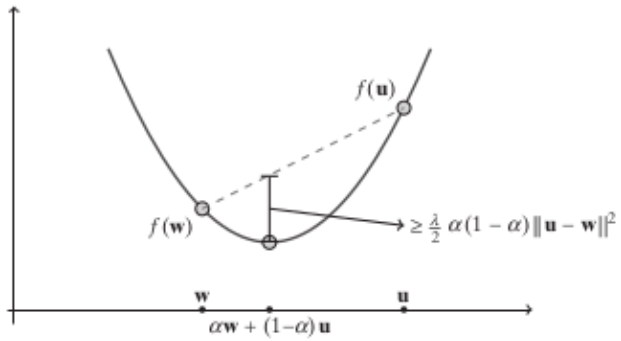


FIGURE 7.1. Strongly convex function

Proposition 39.

(1) *The function* $f(w) = \lambda\|w\|^2$ *is* 2λ-strongly convex
(2) *If* f *is* λ-strongly convex and g is convex then $f+g$ is λ-strongly convex

Proof. Both can be checked from the definition by substitution. □

Lemma 40. *If* f *is* λ-strongly convex and $w^* = \arg\min_w f(w)$ is a minimizer of f then for any w

$$f(w) - f(w^*) \geq \frac{\lambda}{2}\|w-w^*\|^2$$

Proof. Exercise 8 in the Exercise sheet. □

Proposition 41. *If* ℓ *is a convex loss function and the class* $\mathcal{H}$ *is convex, then the Ridge ERM*$_\mathcal{H}$ *problem, with learning rule*

$$\mathfrak{A}(\mathcal{S}) = \arg\min_{w\in\mathcal{H}} \left(\hat{R}_\mathcal{S}(w) + \lambda\|w\|_2^2\right)$$

is a 2λ-strongly convex learning problem.

Proposition 42. *(*ERM *with Ridge regularization) If* ℓ *is a convex loss function, the class* $\mathcal{H}$ *is convex, and* $J(\cdot;\lambda) = \lambda\|\cdot\|_2^2$ *with* $\lambda > 0$ *then* $\hat{R}_\mathcal{S}(w) + \lambda\|w\|_2^2$ *is a* 2λ*-strongly convex function, and the ERM*$_\mathcal{H}$ *problem*

$$w^* = \arg\min_{w\in\mathcal{H}} \left\{\hat{R}_\mathcal{S}(w) + \lambda\|w\|_2^2\right\}$$

is a strongly convex optimization problem (i.e. the learning rule is the minimizer of a strongly convex function over a convex set).

Proof. $\hat{R}_\mathcal{S}(\cdot)$ is a convex function from Proposition 31, $\lambda\|\cdot\|_2^2$ is 2λ-strongly convex, hence $\hat{R}_\mathcal{S}(w) + \lambda\|w\|_2^2$ is a 2λ-strongly convex function. Hence the above ERM$_\mathcal{H}$ problem is a strongly convex optimization problem. □

Machine Learning and Neural Networks (MATH3431) Epiphany term, 2024

Handout 3: Learnability and stability in learning problems

Lecturer & author: Georgios P. Karagiannis georgios.karagiannis@durham.ac.uk

Aim. To introduce concepts PAC, fitting vs stability trade off, stability, and their implementation in regularization problems and convex problems.

Reading list & references:

- Shalev-Shwartz, S., & Ben-David, S. (2014). Understanding machine learning: From theory to algorithms. Cambridge university press.
  - Ch. 2, 3, 13
- Bousquet, O., Boucheron, S., & Lugosi, G. (2003). Introduction to statistical learning theory. In Summer school on machine learning (pp. 169-207). Berlin, Heidelberg: Springer Berlin Heidelberg. (Suitable for PG students)

1. Learnable Probably Approximately Correct (PAC) Learning

Note 1. We formally define the broad learning problem we will work on.

Note 2. Learning algorithms $\mathfrak{A}$ use training data sets $\mathcal{S}$ which may miss characteristics of the unknown data-generating process g. Essentially, approximations (aka errors) are inevitable; some characteristics of data generating process g will be missed even if we use a very representative training set $\mathcal{S}$. We cannot hope that the learning algorithm will find a hypothesis whose error is smaller than the minimal possible error.

Note 3. The PAC learning problem requires no prior assumptions about the data-generating process g. It requires that the learning algorithm $\mathfrak{A}$ will find a predictor $\mathfrak{A}(\mathcal{S})$ whose error is not much larger than the best possible error of a predictor in some given benchmark hypothesis class. So essentially, in practice, the researcher's effort falls on the hypothesis class $\mathcal{H}$.

Definition 4. (Agnostic PAC Learnability for General Loss Functions) A hypothesis class $\mathcal{H}$ is agnostic PAC learnable with respect to a domain $\mathcal{Z}$ of size m and a loss function $\ell : \mathcal{H} \times \mathcal{Z} \to \mathbb{R}_+$, if there exist a function $m_{\mathcal{H}} : (0,1)^2 \to \mathbb{N}$ and a learning algorithm $\mathfrak{A}$ with the following property:

- for every $\epsilon \in (0,1)$, $\delta \in (0,1)$, and distribution g over $\mathcal{Z}$, when running algorithm $\mathfrak{A}$ given training set $\mathcal{S}_m = \{z_1, ..., z_m\}$ with $z_i \overset{\text{IID}}{\sim} g$ for $m \geq m_{\mathcal{H}}(\epsilon, \delta)$ then $\mathfrak{A}$ returns $\mathfrak{A}(\mathcal{S}) \in \mathcal{H}$ such as

$$\Pr_{\mathcal{S} \sim g} \left(R_g\left(\mathfrak{A}\left(\mathcal{S}_m\right)\right) - \min_{h' \in \mathcal{H}} \left(R_g\left(h'\right)\right) \leq \epsilon \right) \geq 1 - \delta \tag{1.1}$$

Note 5. It may be easier to work with expectations: by using Markov inequality (1.1) becomes

$$\Pr_{\mathcal{S}\sim g}\left(R_g\left(\mathfrak{A}\left(\mathcal{S}_m\right)\right)-\min_{h'\in\mathcal{H}}\left(R_g\left(h'\right)\right)\leq\epsilon\right)\geq 1-\underbrace{\frac{1}{\epsilon}\mathrm{E}_{\mathcal{S}\sim g}\left(R_g\left(\mathfrak{A}\left(\mathcal{S}_m\right)\right)-\min_{h'\in\mathcal{H}}\left(R_g\left(h'\right)\right)\right)}_{\delta} \tag{1.2}$$

and hence we need to work with expectations and bounded above.

Hint: Markov inequality $\Pr\left(X\geq a\right)\leq\frac{1}{a}\mathrm{E}\left(X\right)$ for $X>0$.

Remark 6. The accuracy parameter ϵ determines how far the output rule h can be from the optimal one (this corresponds to the 'approximately correct'). The confidence parameter δ indicates how likely the classifier is to meet that accuracy requirement (corresponds to the "probably" part of "PAC").

2. Analysis of the risk based on the trade-off fitting vs stability

Note 7. Let $R^*=\min_{\forall h}\left(R\left(h\right)\right)$ be an ideal/optimal (hence minimum) Risk, $\mathfrak{A}$ be a learning algorithm, and $\mathfrak{A}\left(\mathcal{S}\right)$ the learnign rule from $\mathfrak{A}$ under training dataset $\mathcal{S}$. The Risk of a learning algorithm $\mathfrak{A}$ can be decomposed as

$$R_g\left(\mathfrak{A}\left(\mathcal{S}\right)\right)-R^*=\underbrace{\hat{R}_{\mathcal{S}}\left(\mathfrak{A}\left(\mathcal{S}\right)\right)-R^*}_{\text{(I)}}+\underbrace{R_g\left(\mathfrak{A}\left(\mathcal{S}\right)\right)-\hat{R}_{\mathcal{S}}\left(\mathfrak{A}\left(\mathcal{S}\right)\right)}_{\text{(II)}} \tag{2.1}$$

Note 8. Over-fitting can be represented by $R_g\left(\mathfrak{A}\left(\mathcal{S}\right)\right)-\hat{R}_{\mathcal{S}}\left(\mathfrak{A}\left(\mathcal{S}\right)\right)$. However, $\hat{R}_{\mathcal{S}}\left(\cdot\right)$ is a random variable and for our computational convenience here we focus on its expectation w.r.t. $\mathcal{S}\sim g$. Hence, we provide the following (arguable) definition of over-fitting on which we base our analysis .

Definition 9. For a learning algorithm $\mathfrak{A}$, as a measure of over-fitting we consider the expected difference between true Risk and empirical Risk

$$\mathrm{E}_{\mathcal{S}\sim g}\left(R_g\left(\mathfrak{A}\left(\mathcal{S}\right)\right)-\hat{R}_{\mathcal{S}}\left(\mathfrak{A}\left(\mathcal{S}\right)\right)\right) \tag{2.2}$$

Definition 10. We say that learning algorithm $\mathfrak{A}$ suffers from over-fitting when (2.2) is 'too' large.

Note 11. The expected Risk of a learning algorithm $\mathfrak{A}\left(\mathcal{S}\right)$ can be decomposed as

$$\mathrm{E}_{\mathcal{S}\sim g}\left(R_g\left(\mathfrak{A}\left(\mathcal{S}\right)\right)\right)=\underbrace{\mathrm{E}_{\mathcal{S}\sim g}\left(\hat{R}_{\mathcal{S}}\left(\mathfrak{A}\left(\mathcal{S}\right)\right)\right)}_{\text{(I)}}+\underbrace{\mathrm{E}_{\mathcal{S}\sim g}\left(R_g\left(\mathfrak{A}\left(\mathcal{S}\right)\right)-\hat{R}_{\mathcal{S}}\left(\mathfrak{A}\left(\mathcal{S}\right)\right)\right)}_{\text{(II)}} \tag{2.3}$$

(by applying expectations in (2.1) and ignoring R^*), where (I) indicates how well $\mathfrak{A}\left(\mathcal{S}\right)$ fits the training set $\mathcal{S}$, and (II) indicates the discrepancy between the true and empirical risks of $\mathfrak{A}\left(\mathcal{S}\right)$. In Section 3, we argue that (II) measures the over-fitting and a certain type of stability of $\mathfrak{A}\left(\mathcal{S}\right)$.

Note 12. The following result connects the task of upper bounding (and minimizing this bound) the Expected Risk in (2.3) with PAC learning (4).

Theorem 13. *Let $\mathfrak{A}$ be a learning algorithm that guarantees the following:*

- *If* $m \geq m_{\mathcal{H}}(\epsilon)$ *then for every distribution* g*, it is*

$$E_{\mathcal{S}\sim g}\left(R_g\left(\mathfrak{A}\left(\mathcal{S}_m\right)\right) - \min_{h'\in\mathcal{H}}\left(R_g\left(h'\right)\right)\right) \leq \epsilon$$

Then $\mathfrak{A}$ *it satisfied Definition 4:*

- *for every* $\delta \in (0,1)$*, if* $m \geq m_{\mathcal{H}}(\epsilon\delta)$ then

$$\Pr_{\mathcal{S}\sim g}\left(R_g\left(\mathfrak{A}\left(\mathcal{S}_m\right)\right) - \min_{h'\in\mathcal{H}}\left(R_g\left(h'\right)\right) \leq \epsilon\right) \geq 1-\delta$$

Proof. Let $\xi = R_g\left(\mathfrak{A}\left(\mathcal{S}_m\right)\right) - \min_{h'\in\mathcal{H}}\left(R_g\left(h'\right)\right)$. From Markov's inequality $\Pr\left(\xi \geq \mathrm{E}\left(\xi\right)/\delta\right) \leq \frac{1}{\mathrm{E}(\xi)/\delta}\mathrm{E}\left(\xi\right) = \delta$. Namely, $\Pr\left(\xi \leq \mathrm{E}\left(\xi\right)/\delta\right) = 1-\delta$. But it is given that if $m \geq m_{\mathcal{H}}\left((\epsilon\delta)\right)$ for every distribution g it is $\mathrm{E}\left(\xi\right) \leq (\epsilon\delta)$. So by substitution $\Pr\left(\xi \leq (\epsilon\delta)/\delta\right) = 1-\delta$ implies $\Pr\left(\xi \leq \epsilon\right) = 1-\delta$. Now substitute back ξ and we conclude the proof. □

Note 14. We aim to design a learning algorithm $\mathfrak{A}\left(\mathcal{S}\right)$ that both fits the training set and is stable; i.e. minimizing both terms in (2.3). As seen later, there may be a trade-off between (I) and (II); expected empirical risk term and stability term. Hence, we aim to upper bound (2.3) by upper bounding (I) and (II) individually.

3. Stability and over-fitting

Notation 15. Let $\mathcal{S} = \{z_1, ..., z_m\}$ be a training sample, and $\mathfrak{A}$ be a learning algorithm with output $\mathfrak{A}\left(\mathcal{S}\right)$.

Note 16. A learning algorithm can be stable if a small change of the input to the algorithm does not change the output of the algorithm much. Formalizing this in maths, we can say that if $\mathcal{S}^{(i)} = \{z_1, ..., z_{i-1}, z', z_{i+1}, ..., z_m\}$ is another training dataset equal to $\mathcal{S}$ but the ith element which is replaced by another $z' \sim g$, then a good learning algorithm $\mathfrak{A}$ would produce a small value of

$$\ell\left(\mathfrak{A}\left(\mathcal{S}^{(i)}\right), z_i\right) - \ell\left(\mathfrak{A}\left(\mathcal{S}\right), z_i\right) \geq 0$$

Definition 17. We say that a learning algorithm $\mathfrak{A}$ is **on-average-replace-one-stable** with rate $\epsilon : \mathbb{N} \to \mathbb{R}$ if for every distribution g

$$\mathrm{E}_{\substack{\mathcal{S}\sim g \\ z'\sim g, i\sim U\{1,...,m\}}}\left(\ell\left(\mathfrak{A}\left(\mathcal{S}^{(i)}\right), z_i\right) - \ell\left(\mathfrak{A}\left(\mathcal{S}\right), z_i\right)\right) \leq \epsilon\left(m\right)$$

where $\epsilon\left(\cdot\right)$ has to be a decreasing function.

Note 18. Following we discuss the association of stability and over-fitting, based on the over-fitting Definition 10.

Theorem 19. *For any learning algorithm* $\mathfrak{A}$

$$E_{\mathcal{S}\sim g}\left(R_g\left(\mathfrak{A}\left(\mathcal{S}\right)\right) - \hat{R}_{\mathcal{D}}\left(\mathfrak{A}\left(\mathcal{S}\right)\right)\right) = E_{\substack{\mathcal{S}\sim g,\, z'\sim g \\ i\sim U\{1,...,m\}}}\left(\ell\left(\mathfrak{A}\left(\mathcal{S}^{(i)}\right), z_i\right) - \ell\left(\mathfrak{A}\left(\mathcal{S}\right), z_i\right)\right)$$

where g is a distribution, $\mathcal{S} = \{z_1, ..., z_m\}$ and $\mathcal{S}^{(i)} = \{z_1, ..., z_{i-1}, z', z_{i+1}, ..., z_m\}$ are training datasets with $z', z_1, ..., z_m \overset{\text{iid}}{\sim} g$.

Proof. As $z', z_1, ..., z_m \overset{\text{iid}}{\sim} g$, then for every i

$$\mathrm{E}_{\mathcal{S}\sim g}\left(R_g\left(\mathfrak{A}\left(\mathcal{S}\right)\right)\right) = \mathrm{E}_{\substack{\mathcal{S}\sim g \\ z'\sim g}}\left(\ell\left(\mathfrak{A}\left(\mathcal{S}\right), z'\right)\right) = \mathrm{E}_{\substack{\mathcal{S}\sim g \\ z'\sim g}}\left(\ell\left(\mathfrak{A}\left(\mathcal{S}^{(i)}\right), z_i\right)\right)$$

and

$$\mathrm{E}_{\mathcal{S}\sim g}\left(\hat{R}_{\mathcal{S}}\left(\mathfrak{A}\left(\mathcal{S}\right)\right)\right) = \mathrm{E}_{\substack{\mathcal{S}\sim g \\ i\sim U\{1,...,m\}}}\left(\ell\left(\mathfrak{A}\left(\mathcal{S}^{(i)}\right), z_i\right)\right)$$

□

4. Implementation in regularized loss learning problems

Definition 20. Assume $(\mathcal{H}, \mathcal{Z}, \ell)$. Regularized Loss Minimization (RLM) learning rule is the one that results as the output of jointly minimizing the empirical risk $\hat{R}_{\mathcal{Z}}(h)$ and a regularization function $J : \mathcal{H} \to \mathbb{R}$ that is

$$h^* = \underbrace{\arg\min}_{h\in\mathcal{H}}\left(\hat{R}_{\mathcal{S}}\left(h\right) + J\left(h\right)\right) \tag{4.1}$$

Remark 21. The motivation for considering the regularization function J in (4.1) is to: (1.) control complexity and (2.) improve stability; as we will see later.

Note 22. We make our example more specific and narrow it to the Ridge RLM learning problem (could be LASSO, Elastic Net, etc.).

Definition 23. The Ridge RLM learning problem $(\mathcal{H}, \mathcal{Z}, \ell)$, here $\mathcal{H} = \mathcal{W} \subset \mathbb{R}^d$, uses regularization function $J\left(w; \lambda\right) = \lambda \left\|w\right\|_2^2$ with $\lambda > 0$, $w \in \mathcal{W}$ and produces learning rule

$$\mathfrak{A}\left(\mathcal{S}\right) = \underset{w\in\mathcal{W}}{\arg\min}\left(\hat{R}_{\mathcal{S}}\left(w\right) + \lambda \left\|w\right\|_2^2\right) \tag{4.2}$$

Note 24. Recall (Term 1) that the regularization function in Ridge RLM learning problem penalizes complexity. Essentially, implies a sequence of hypothesis $\mathcal{H}_1 \subset \mathcal{H}_2 \subset ...$ with $\mathcal{H}_i = \left\{w \in \mathbb{R}^d : \left\|w\right\|_2 < i\right\}$.

Note 25. Below, we will try to analyze the behavior of Ridge RLM learning rule (4.2) w.r.t. the Risk decomposition (2.3). In particular, to upper bounded w.r.t. the the shrinkage term λ,training sample size m, and other characteristics.

4.1. Bounding the empirical risk (I) in (2.3).

Note 26. From (4.2), we have

$$\begin{aligned}\hat{R}_{\mathcal{S}}\left(\mathfrak{A}\left(\mathcal{S}\right)\right) &\leq \hat{R}_{\mathcal{S}}\left(\mathfrak{A}\left(\mathcal{S}\right)\right) + \lambda \left\|\mathfrak{A}\left(\mathcal{S}\right)\right\|_2^2 \\ &\leq \hat{R}_{\mathcal{S}}\left(w'\right) + \lambda \left\|w'\right\|_2^2; \quad \forall w' \in \mathcal{W}\end{aligned}$$

and by taking expectations w.r.t. $\mathcal{S}$, it is

$$\mathrm{E}_{\mathcal{S}\sim g}\left(\hat{R}_{\mathcal{S}}\left(\mathfrak{A}\left(\mathcal{S}\right)\right)\right) \leq R_g\left(w'\right) + \lambda\left\|w'\right\|_2^2; \quad \forall w' \in \mathcal{W} \tag{4.3}$$

because $\mathrm{E}_{\mathcal{S}\sim g}\left(\hat{R}_{\mathcal{S}}\left(\cdot\right)\right) = \frac{1}{m}\sum_{i=1}^{m}\mathrm{E}_{\mathcal{S}\sim g}\left(\ell\left(\cdot, z_i\right)\right) = R_g\left(\cdot\right)$.

Note 27. We observe that part (I) in the expected risk decomposition (2.3), (aka the upper bound of the expected empirical risk) increases with the regularization term $\lambda > 0$. (!!!) –Although expected, it did not start well.

4.2. **Bounding the empirical risk (II) in (2.3).**

Note 28. We do that by constraining the loss function to be convex and Lipstchitz.

Assumption 29. *The loss function $\ell\left(\cdot, z\right)$ in (4.2) is convex for any $z \in \mathcal{Z}$.*

Note 30. Let $\tilde{R}_{\mathcal{S}}\left(w\right) = \hat{R}_{\mathcal{S}}\left(w\right) + \lambda\left\|w\right\|_2^2$. $\tilde{R}_{\mathcal{S}}\left(\cdot\right)$ is 2λ-strongly convex as the sum of a convex function $\hat{R}_{\mathcal{S}}\left(\cdot\right)$ (Assumption 29) and a 2λ-strongly convex function $J\left(\cdot;\lambda\right) = \lambda\left\|\cdot\right\|_2^2$ (results directly from Definition 43).

Note 31. Because $\tilde{R}_{\mathcal{S}}\left(\cdot\right)$ is 2λ-strongly convex and $\mathfrak{A}_{\text{Ridge}}\left(\mathcal{S}\right)$ is its minimizer, according to Lemma 45 in Handout (...), for $\mathfrak{A}\left(\mathcal{S}\right)$ and any $w \in \mathcal{W}$, it is

$$\tilde{R}_{\mathcal{S}}\left(w\right) - \tilde{R}_{\mathcal{S}}\left(\mathfrak{A}\left(\mathcal{S}\right)\right) \geq \lambda\left\|w - \mathfrak{A}\left(\mathcal{S}\right)\right\|^2, \quad \forall w \in \mathcal{W} \tag{4.4}$$

Note 32. Also, for any $w, u \in \mathcal{W}$, it is

$$\begin{aligned}
\tilde{R}_{\mathcal{S}}\left(w\right) - \tilde{R}_{\mathcal{S}}\left(u\right) &= \left(\hat{R}_{\mathcal{S}}\left(w\right) + \lambda\left\|w\right\|_2^2\right) - \left(\hat{R}_{\mathcal{S}}\left(u\right) + \lambda\left\|u\right\|_2^2\right) \\
&= \left(\hat{R}_{\mathcal{S}^{(i)}}\left(w\right) + \lambda\left\|w\right\|_2^2\right) - \left(\hat{R}_{\mathcal{S}^{(i)}} + \lambda\left\|u\right\|_2^2\right) \\
&\quad + \frac{\ell\left(w, z_i\right) - \ell\left(u, z_i\right)}{m} + \frac{\ell\left(w, z'\right) - \ell\left(u, z'\right)}{m} \\
&= \tilde{R}_{\mathcal{S}^{(i)}}\left(w\right) - \tilde{R}_{\mathcal{S}^{(i)}}\left(u\right) + \frac{\ell\left(w, z_i\right) - \ell\left(u, z_i\right)}{m} + \frac{\ell\left(w, z'\right) - \ell\left(u, z'\right)}{m}
\end{aligned}$$

Choosing $w = \mathfrak{A}\left(\mathcal{S}^{(i)}\right)$ and $u = \mathfrak{A}\left(\mathcal{S}\right)$, and the fact that $\tilde{R}_{\mathcal{S}^{(i)}}\left(\mathfrak{A}\left(\mathcal{S}^{(i)}\right)\right) \leq \tilde{R}_{\mathcal{S}^{(i)}}\left(\mathfrak{A}\left(\mathcal{S}\right)\right)$ it is

$$\tilde{R}_{\mathcal{S}}\left(w\right) - \tilde{R}_{\mathcal{S}}\left(u\right) \leq \frac{\ell\left(\mathfrak{A}\left(\mathcal{S}^{(i)}\right), z_i\right) - \ell\left(\mathfrak{A}\left(\mathcal{S}\right), z_i\right)}{m} + \frac{\ell\left(\mathfrak{A}\left(\mathcal{S}^{(i)}\right), z'\right) - \ell\left(\mathfrak{A}\left(\mathcal{S}\right), z'\right)}{m} \tag{4.5}$$

Note 33. Then (4.5) and (4.4) imply

(4.6)

$$\lambda\left\|\tilde{R}_{\mathcal{S}}\left(\mathfrak{A}\left(\mathcal{S}^{(i)}\right)\right) - \tilde{R}_{\mathcal{S}}\left(\mathfrak{A}\left(\mathcal{S}\right)\right)\right\| \leq \frac{\ell\left(\mathfrak{A}\left(\mathcal{S}^{(i)}\right), z_i\right) - \ell\left(\mathfrak{A}\left(\mathcal{S}\right), z_i\right)}{m} + \frac{\ell\left(\mathfrak{A}\left(\mathcal{S}^{(i)}\right), z'\right) - \ell\left(\mathfrak{A}\left(\mathcal{S}\right), z'\right)}{m}$$

Note 34. Now that we brought it in that form, we can use an additional assumption on the loss to bound it.

Assumption 35. *The loss function $\ell\left(\cdot, z\right)$ in (4.2) is convex for any $z \in \mathcal{Z}$ and ρ-Lipschitz.*

Note 36. Given ρ-Lipschitzness in Assumption 35, it is

$$\ell\left(\mathfrak{A}\left(\mathcal{S}^{(i)}\right), z_i\right) - \ell\left(\mathfrak{A}\left(\mathcal{S}\right), z_i\right) \leq \rho \left\|\mathfrak{A}\left(\mathcal{S}^{(i)}\right) - \mathfrak{A}\left(\mathcal{S}\right)\right\| \tag{4.7}$$
$$\ell\left(\mathfrak{A}\left(\mathcal{S}^{(i)}\right), z'\right) - \ell\left(\mathfrak{A}\left(\mathcal{S}\right), z'\right) \leq \rho \left\|\mathfrak{A}\left(\mathcal{S}^{(i)}\right) - \mathfrak{A}\left(\mathcal{S}\right)\right\|$$

and hence (4.6) yields

$$\left\|\mathfrak{A}\left(\mathcal{S}^{(i)}\right) - \mathfrak{A}\left(\mathcal{S}\right)\right\| \leq 2\frac{\rho}{\lambda m} \tag{4.8}$$

Note 37. Plugging (4.8) in (4.7) yields

$$\ell\left(\mathfrak{A}\left(\mathcal{S}^{(i)}\right), z_i\right) - \ell\left(\mathfrak{A}\left(\mathcal{S}\right), z_i\right) \leq 2\frac{\rho^2}{\lambda m}$$

Note 38. Using Theorem 19, we get an upper bound for the stability / over-fitting

$$\mathrm{E}_{\mathcal{S}\sim g}\left(R_g\left(\mathfrak{A}\left(\mathcal{S}\right)\right) - \hat{R}_{\mathcal{D}}\left(\mathfrak{A}\left(\mathcal{S}\right)\right)\right) = \mathrm{E}_{\substack{\mathcal{S}\sim g,\, z'\sim g \\ i\sim U\{1,...,m\}}}\left(\ell\left(\mathfrak{A}\left(\mathcal{S}^{(i)}\right), z_i\right) - \ell\left(\mathfrak{A}\left(\mathcal{S}\right), z_i\right)\right) \leq 2\frac{\rho^2}{\lambda m}$$

Note 39. After this saga, the researcher could come to the conclusion that: In a Ridge regularization learning problem with loss function which is convex and ρ-Lipschitz, and the regularizer is $J\left(\cdot;\lambda\right) = \lambda\left\|\cdot\right\|^2$ with $\lambda > 0$, the learning rule trained against iid sample $\mathcal{S} = \{z_i\}_{i=1}^m$ is on-averave-replace-one-stable with rate $\epsilon\left(m\right) = 2\frac{\rho^2}{\lambda m}$; i.e.

$$\mathrm{E}_{\mathcal{S}\sim g}\left(R_g\left(\mathfrak{A}\left(\mathcal{S}\right)\right) - \hat{R}_{\mathcal{D}}\left(\mathfrak{A}\left(\mathcal{S}\right)\right)\right) \leq 2\frac{\rho^2}{\lambda m} \tag{4.9}$$

Note 40. From (4.9), we see that stability improves (and over-fitting decreases) as the shrinkage parameter λ increases.

4.3. **Bounding the Risk (2.3).**

Note 41. Given the bounds (4.3) and (4.9), the decomposition of the expected Risk in (2.3) yields that: In a Ridge regularization learning problem with loss function which is convex and ρ-Lipschitz, and the regularizer is $J\left(\cdot;\lambda\right) = \lambda\left\|\cdot\right\|^2$ with $\lambda > 0$, the learning rule trained against iid sample $\mathcal{S} = \{z_i\}_{i=1}^m$ has expected Risk bound

$$\mathrm{E}_{\mathcal{S}\sim g}\left(R_g\left(\mathfrak{A}\left(\mathcal{S}\right)\right)\right) \leq \underbrace{R_g\left(w'\right) + \lambda\left\|w'\right\|_2^2}_{\text{(I)}} + \underbrace{2\frac{\rho^2}{\lambda m}}_{\text{(II)}}; \quad \forall w' \in \mathcal{W} \tag{4.10}$$

Note 42. From 4.10, we see that there is a trade-off between Empirical Risk (I) and stability (II) with regards the regularization parameter λ. We wish to use the optimal $\lambda > 0$ corresponding to the smallest bound in (4.10); it has to both fit the training data well (but perhaps not too well) and be very stable to different training data from the same g (but perhaps not too stable)!

Definition 43. (Strongly Convex functions) A function f is λ-strongly convex function is for all w, u, and $\alpha \in (0,1)$ we have

$$f(aw + (1-\alpha)u) \leq af(w) + (1-\alpha)f(u) - \frac{\lambda}{2}\alpha(1-\alpha)\|w-u\|^2 \tag{4.11}$$

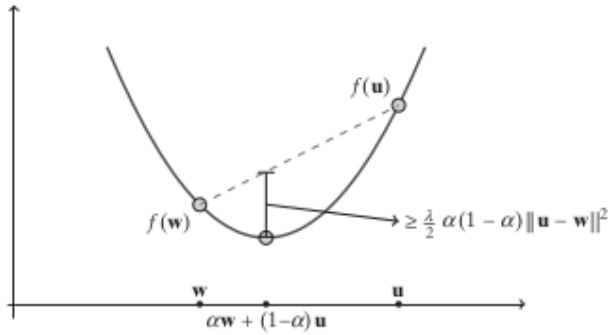


FIGURE 4.1. Strongly convex function

Proposition 44.

(1) *The function* $f(w) = \lambda\|w\|_2^2$ *is* 2λ-strongly convex
(2) *If* f *is* λ-strongly convex and g is convex then $f+g$ is λ-strongly convex

Lemma 45. *If* f *is* λ-strongly convex and u is a minimizer of f then for any w

$$f(w) - f(u) \geq \frac{\lambda}{2}\|w-u\|^2$$

Proof. Exercise 8 in the Exercise sheet. □