MATHEMATICAL FOUNDATIONS OF MACHINE LEARNING

(NMAG 469, FALL TERM 2017-2018)

HÔNG VÂN LÊ *

Contents

Machine learning is a subfield of computer sciences which builds models for deriving a theory from empirical data under certain computational constraints and mathematical assumptions. Machine learning is typically used whenever large amounts of data are available and when one aims at a computer program that is (too) difficult to program directly. Mathematical foundation of machine learning is statistical learning theory.

In this lecture course we cover the following topics: statistical model and learning machine, learning algorithm and estimator, VC-dimension, Rademacher complexity, Fisher metric, efficient estimator, neural network and natural gradient flow, supervised learning, unsupervised learning and deep learning, support vector machine and reproducing kernel Hilbert space.

Date: October 13, 2017.

* Institute of Mathematics of ASCR, Zitna 25, 11567 Praha 1, email: hvle@math.cas.cz.

Prerequisite: basis knowledge in linear algebra, analysis and probability theory is required as well as some elementary Hilbert space theory.

Recommended textbooks:

- Foundations of Machine Learning, M. Mohri, A. Rostamizadeh, A. Talwalkar, MIT Press, 2012,
- Understanding Machine Learning: From Theory to Algorithms, Shai Shalev-Shwartz, Shai Ben-David, Cambridge University Press, 2014

For understanding the nature (philosophy) of machine learning:

- The nature of statistical learning theory, V. Vapnik, Springer, 1999,
- Estimation of Dependences Based on Empirical Data, V. Vapnik, Springer, 2006

- Probably Approximately Correct, L. Valiant, Basic Book, 2013.

The lecture course most close to our one:

- Mathematical Foundations of Machine Learning, lecture notes, M. M. Wolf, TU München, 2016, (updated 2017).

1. Introduction

Machine learning is defined by its current problems together with methodology and techniques to solve them, its history and our vision for its future. Today I shall briefly discuss history of machine learning and its main current problems. During the course, especially at the end of our lecture course, we shall discuss our vision for future of machine learning.

1.1. **A brief history of machine learning.** Machine learning is a subfield of computer science that evolved from the study of pattern recognition in artificial intelligence. Already in the early days of AI [1], after the second WW II, some researchers were interested in having machines learn from data. They attempted to approach the problem with various symbolic methods, as well as what were then termed "neural networks"; these were mostly perceptrons and other models that were later found to be reinventions of the generalized linear models of statistics. Both these methods will be considered later in our course from a modern point of view. Probabilistic reasoning was also employed, especially in automated medical diagnosis. In 1959, Arthur Samuel [2] defined machine learning as a "Field of study that gives computers the ability to learn without being explicitly programmed".

[1] the name AI is coined in 1956 according to [RN2010]

[2] (December 5, 1901 -July 29, 1990) was an American pioneer in the field of computer gaming and artificial intelligence. The Samuel Checkers-playing Program appears to be the world's first self-learning program, and as such a very early demonstration of the fundamental concept of artificial intelligence (AI)

1.2. **Current tasks and types of machine learning.** One common feature of all of tasks in machine learning is that, in contrast to more traditional uses of computers, in these cases, due to the complexity of the patterns that need to be detected, a human programmer cannot provide an explicit and detailed specification of how such tasks should be executed.

To be more specific on tasks in machine learning which do not use *explicit and detailed specification* and to give an overview of types of machine learning we need first to agree what is the essential feature of learning that lead to such specifications.

A core objective of a learner is to *generalize from its experience*. Generalization in this machine context is the ability of a learning machine to *perform prediction/decision accurately on new, unseen examples/tasks after having experienced a learning data set*. The training examples come from some generally unknown probability distribution (considered representative of the space of occurrences) and the *learner has to build a general model about this space*. The probability natural of machine learning come from incompleteness of information or randomness of events/ depending on your philosophy. We shall this random nature and its model in machine learning in the next lecture. Generalization has another name: inductive reasoning/inference. Theory of inductive reasoning is statistical learning theory.

Applications of machine learning include spam filtering (new spam come every day, the program has to recognize them from experience), optical character recognition (OCR) (hand writings and print words are not standard or completely classified), natural language processing, search engines and computer vision are all pattern recognition problems. Machine learning is sometimes conflated with data mining, where the latter subfield focuses more on exploratory data analysis and is known as unsupervised learning.

1.2.1. *Main tasks of current machine learning.* Let us give a short description of current applications of machine learning.

Classification task assigns a category to each item. For example, document classification may assign items with categories such as politics, email spam, sports, or weather while image classification may assign items with categories such as landscape, portrait, or animal. The number of categories in such tasks is often relatively small, but can be large in some difficult tasks and even unbounded as in OCR, text classification, or speech recognition. In short, a classification task is a (construction of a) function on the set of items that takes value in a *countable set*.

Regression task predicts a real value for each item. Examples of regression include prediction of stock values or variations of economic variables. In this problem, the penalty for an incorrect prediction depends on the magnitude of the *distance between the true and predicted values*, in contrast with the classification problem, where there is typically no notion of closeness between

various categories. In short, a regression task is a (construction of a) function on the set of items that takes value in $\mathbb{R}$. [3]

Ranking task orders items according to some criterion. Web search, e.g., returning web pages relevant to a search query, is the canonical ranking example. If the number of ranking is finite, then this task is close to the classification problem, but not the same, since in the ranking task we need to specify each rank during the task and not before the task as in the classification problem.

Clustering task partitions items into (homogeneous) regions. Clustering is often performed to analyze very large data sets. For example, in the context of social network analysis, clustering algorithms attempt to identify communities within large groups of people. The closeness of items is measured by a distance function o the set of items.

Dimensionality reduction or manifold learning transforms an initial representation of items in high dimensional space into a space of lower dimension while preserving some properties of the initial representation. A common example involves preprocessing digital images in computer vision tasks.

1.2.2. *Main types of machine learning.* The type of a machine learning task is defined by the type of *interaction* between *the learner* and *the environment*. More precisely we consider *types of training data* available to the learner, the outcomes and *and the test data* used to evaluate (and apply) the learning algorithm.

Main types of machine learning are supervised and unsupervised.

• In *supervised learning a learner*, also called *a learning machine*, is a device that receives *labeled training data* as input and outputs a program that predicts the label for unseen instances and thus generalizes beyond the training data, see the next lecture for more precise mathematical description. Examples of sets of labeled data are emails that are labeled "spam" or "no spam" and medical histories that are labeled with the occurrence or absence of a certain disease. In these cases the learners output would be a spam filter and a diagnostic program, respectively. Most of classification and regression problems of machine learning belong to supervised learning.

• In *unsupervised learning* there is *no additional label* attached to the data and the task is to identify and model hidden patterns in the data. Clustering and dimensionality reduction are example of unsupervised learning problems. Most important problem of unsupervised learning are problem of finding association rules that are important in market analysis, banking

[3]The term "regression" was coined by Francis Galton in the nineteenth century to describe a biological phenomenon. The phenomenon was that the heights of descendants of tall ancestors tend to regress down towards a normal average (a phenomenon also known as regression toward the mean of population). For Galton, regression had only this biological meaning, but his work was later extended by Udny Yule and Karl Pearson to a more general statistical context: movement toward the mean of a statistical population. Galton method of investigation is non-standard at that time: first he collected the data, then he guessed the relationship model of the events.

security and consists of important part of pattern recognisation, which is important for understand advanced AI.

At the current time, unsupervised learning is primarily *descriptive* and exprimental whereas supervised learning is more *predictive* (and has deeper theoretical foundation). In this course, we will deal with the supervised learning and discuss the problem of unsupervised learning only in the last lecture.

There are some immediate types between supervised learning and unsupervised learning. Here we consider two main types.

• In *semi-supervised learning* the learner receives a training sample consisting of both labeled and unlabeled data, and makes predictions for all unseen points. Semi-supervised learning is common in settings where unlabeled data is easily accessible but labels are expensive to obtain. Various types of problems arising in applications, including classification, regression, or ranking tasks, can be framed as instances of semi-supervised learning. The hope is that the distribution of unlabeled data accessible to the learner can help him achieve a better performance than in the supervised setting. The analysis of the conditions under which this can indeed be realized is the topic of much modern theoretical and applied machine learning research.

• In *reinforcement learning*. Reinforcement learning is the study of planing and learning in a scenario where a learner actively interacts with the environment to achieve a certain goal. More precisely, the learner does not passively receive a labeled data set. Instead, he collects information through a course of actions by interacting with the environment. This active interaction justifies the terminology of *agent* used to refer to the learner. The achievement of the agents goal is typically measured by the reward he receives from the environment and which he seeks to maximize.

Unlike the supervised learning scenario, in reinforcement learning *there is no fixed distribution* according to which instances are drawn; the choice of *a policy* (the best course of actions) defines the distribution.

We shall not cover these very rich areas of machine learning, especially reinforcement learning with connections to control theory, optimization, and cognitive sciences and refer the interested in to [MRT2012, Chapter 14, p. 313] for discussion.

Thus our lecture course shall cover mainly supervised learning.

1.2.3. *Representation methods of supervised learning.* A first coarse classification of supervised learners is in terms of the chosen representation, which determines the basic structure of the generated programs. Common ones are:

- Decision trees, (i.e. using graph to represent learning/decision algorithm)

- k-nearest neighbors (k-NN), (a metric method of classification/regression task),

- Neural networks,

- Support vector machines and kernel methods.

These representations (of functions we need to compute), though quite different in nature, have two important things in common: they are computable and they form universal hierarchies. The fact that their structure are computable are essentially important.

Forming a universal hierarchy means that the representation contains more and more refined levels that, in principle, are capable of representing every possibility or at least approximating every possibility to arbitrary accuracy. Only few such representations are known and the above examples (together with variations on the theme and combinations thereof) already seem to cover most of the visible universe.

The first two methods are simple so we shall not discuss in or course. You can read about them in the two textbooks I recommended. We spend more time on the two last methods, which are arguably the most sophisticated and most powerful ones.

1.3. **Basic questions in mathematical foundations of machine learning.** What I said about current problems in machine learning concern only its applications, what are they and little bit on their methods. To understand machine learning deeply and its future directions we need to understand why machine learning works, what is its limitations. In short we want to study mathematical foundation of machine learning that discuss in deep the following questions:

- Q1 What is the mathematical model of learning?
- Q2 How to quantify the difficulty/hardness/complexity of a learning problem?
- Q3 How to choose a learning algorithm?
- Q4 How to measure success of machine learning?

Clearly Q1 should provide general guidance for investigating Q2, 3, 4. The investigation of Q3 depends on our understanding of Q2 and Q4.

We also study basic mathematical methods of machine learning that illustrate the three above questions, in particular we shall learn which knowledge a machine can acquire.

1.4. **Future of machine learning and AI.** The future of machine learning depends very much on the foundation of machine learning which we discuss above and that contains also the following sub-questions: Which knowledge (under which environment) can a machine learn?[4] Is there a limitation of machine learning?

In a sense, machine learning can be viewed as a branch of AI (Artificial Intelligence), since, after all, the ability to turn experience into expertise or to detect meaningful patterns in complex sensory data is a cornerstone of

[4]Studying knowledge is something philosophers have been doing since Plato's work Theaetetus but we shall approach to the concept of knowledge from a mathematical point of views.

human (and animal) intelligence. However, one should note that, in contrast with *traditional* AI, machine learning is not trying to build automated imitation of intelligent behavior, but rather to use the strengths and special abilities of computers to complement human intelligence, often performing tasks that fall way beyond human capabilities. For example, the ability to scan and process huge databases allows machine learning programs to detect patterns that are outside the scope of human perception. I believe that AI and machine learning shall converge because the difference between them - the automated imitation of human intelligent behavior shall disappear. Every intelligent task is expressible in term of decision and computation problems which belong to the domain of mathematics, statistical learning theory and machine learning (with vision for future of machine learning, see below).

Let me state some problems of future machine learning

1) Put theoretical foundation for unsupervised learning.

2) Unify inductive reasoning with logical reasoning.

3) Understand the essential of intellect.

There are many papers discussing these questions you can find in internet. We also may discuss these problems in our next term seminar on machine learning.

1.5. **Conclusion.** Machine learning is algorithmic implementation of statistical learning (inductive inference). Machine learning has been created to address the need for good performance in computing capacity relative to investment in energy, time, money. In all tasks of machine learning, regarding as computing functions of many variables, (the size of) the variables are not explicitly specified, the computer program needs to decide what to do with new variables based on past performance/experience.

Finally I recommend few sources on machine learning that emphasize methods and applications (and not foundational questions). There are two popular textbooks for computer scientists on mathematical methods of machine learning:

- T. Hastie, R. Tibshirani and J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer 2008.

- Bishop C. M. Pattern Recognition and Machine Learning, Springer, 2006.

The field ML currently are being very quickly developed. It is useful to check sites on ML. Here is a good repository of software, data set, papers in ML:

http://mlcomp.org/

which are currently migrated to the new website

https://worksheets.codalab.org/worksheets/0x818930127c4d47de84c1ceaadf04d014/

2. Statistical learning framework for supervised learning

Last week formulated four main questions of mathematical foundation of machine learning. Today we shall discuss the first question, namely a mathematical model of machine learning.

Mathematical language and basic mathematical concepts of machine learning that are needed for modelling of machine learning stem from statistical learning theory whose founder is Vladimir Vapnik. Leslie Valiant added to Vapnik's theory computational complexity ingredients, that concern the second and the fourth questions we formulated in the last lecture and gave it the name "Probably Approximately Correct". Today we shall learn statistical setting of machine learning and in two weeks we shall learn PAC theory deeper.

According to Vapnik, at the beginning of mathematical learning theory is the following *mathematical* postulation:

"Learning is a problem of function estimation on the basis of empirical data".

In mathematical language experience is empirical data and *knowledge is function estimation.* We shall examine this mathematical postulation in today lecture and for that we also need mathematical notions formalizing the following concepts: a learner (learning machine), the set of objects we want to learn, property of objects we want to learn, generalization, training data.

Example 2.1. A ML firm wants to estimate the potential of applicants to new positions of developers of algorithms in ML of its firm based on its experience that the potential of a software developer depends on three qualities of an applicant: his/her analytical mathematical skill rated by the mark (from 1 to 20) in his graduate diploma, his/her computer sciences skill, rated by the mark (from 1 to 20) in his graduate diploma, and his/her communication skill rated by the firm test (scaled from 1 to 5). The potential of an applicant for the open position is evaluated in scale 1-10. Since the position of developer of algorithm in ML will be periodically re-opened and therefore they can design a ML program to predict the potential of applicants such that the program *automatically will be improved with time.*

Remark 2.2. 1. Functions, more precisely, we need to estimate are called *the outputs of the learner (a learning machine).* In the above example, the output is the potential of applicant, regarded as a function of the variable (mathematical skill, computer sciences, communication skill). Furthermore, the feature functions, (in the considered example: the potential, the mathematical skill and the communication skill) are measured only approximately correctly and no marks give a full information on mathematical ability of the holder. That is why we need probability assumption in learning theory. The learning theory that incorporates probability assumption is called *statistical learning theory.*

2. Feature functions are often classified by the *type of their values*: binary, finite, real etc...

3. The learning example with applicants is based on the experience that the potential of an applicant correlates with the marks in his/her diploma in mathematics, computer sciences and with his/her communication skill. It is good question to keep in mind for future examination, when we learn more advanced learning theory, what happens if the potential of the applicant does not correlates with the applicant communication skill? How to recognize this (in)correlation? Is there a notion of partial correlation and if so, how to express this notion in mathematical language? We shall address this question in stochastic scenario of statistical learning theory later.

2.1. **Setting of a statistical learning problem.**

2.1.1. *Main notions.* In the basic statistical learning setting, the learner has access to the following:

• *Domain set* (also called *input space*) is an arbitrary set $\mathcal{X}$. This is the set of objects, also called *examples, instances*, that we may wish to label, i.e. we wish to define a map (also called predictor) $h : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{Y}$ the set of possible outcomes[5]. For example, in the applicant learning problem mentioned before, the domain set will be the set of all applicants and as we know the predictor is the function of application whose value is his/her potential. Since we are interested only in the *dependence of the predictor on certain features of the object in consideration*, domain points (elements) will be represented by features F of the objects in consideration (the marks in math, CS and in communication test in the example above) and we *assume* that the output (the predictor) $h(x)$, $x \in \mathcal{X}$ depends on (correlates to) $F(x)$.

• *Label set* is the set $\mathcal{Y}$ of possible outcomes, (in geometric language: $\mathcal{Y}$ is the target space of the desired map h).

Remark 2.3. We note that the choice of a mathematical representation of the real world output is also very delicate. The taste degree may take value in a nonempty set S consisting of more than two elements. By choosing an arbitrary subset S_0 in S we can simplify our taste decision problem into the case when S has two elements.

• *Training data* is a sequence $S = \{(x_1, y_1), \cdots (x_n, y_n)\} \in (\mathcal{X} \times \mathcal{Y})^m$ of observed correlations.

• *Prediction rule* is a map $h_S : \mathcal{X} \to \mathcal{Y}$ based on S. This map is also called a *predictor*, a *hypothesis*, or *a classifier* [6]. To find prediction rule is the goal of the learner. (The learner is also called a *learning machine*). In other words the learner needs to find *an algorithm* (a "computable" map)

$$A : \bigcup_{n\in\mathbb{N}} (\mathcal{X} \times \mathcal{Y})^n \to \mathcal{Y}^{\mathcal{X}},\ S \to h_S. \tag{2.1}$$

[5]Classical, in mathematical statistic, inputs are called *independent variables* and outputs are called *dependent variables*.

[6]h stands for "hypothesis"

In our applicant example, A is a rule that our learner will employ to predict whether a future applicant he examines is going to be successful or not.

As we we have remarked in Remark 2.2.1 we can measure a feature function $F(x)$ that represents instance x in consideration only approximately correctly. (We regard $F(x)$ as coordinates of x and from now on we identify x with $F(x)$). This idea of approximately correctly representation/measurement and its consequences will be formalized in mathematical language of statistical learning theory below.

2.1.2. *Probability (and stochastic) assumption of statistical learning theory.* Recall that we can measure $x \in \mathcal{X}$ (or more precisely its representation/coordinates function $F(x)$) only approximately correctly. So we represent x as a random value on $\mathcal{X}$ and therefore x is distributed by some probability measure $\mu_{\mathcal{X}}$. In other words, the probability that x belongs to $A \in \Sigma_{\mathcal{X}}$ is $\mu_{\mathcal{X}}(A)$.

In the same way, we do not have a complete information on the true hypothesis $h : \mathcal{X} \to \mathcal{Y}$, even in the observed (labeled) pair (x_i, y_i) y_i may not be represented as a value of a function of x. We express the random nature of the pair (x_i, y_i) by a probability measure μ_h on $\mathcal{X} \times \mathcal{Y}$, i.e. the probability that a labeled pair (x, y) belongs to $B \in \Sigma_{(\mathcal{X}\times\mathcal{Y})}$ is $\mu_h(B)$.

Now let us formulate this very general assumption on incomplete information of $h(x, y)$ using probability theory.

A data-generating model describes the random nature of observable data/ training data. The pairs (x_i, y_i) in a sequence S of training data are treated as values of random variables (X_i, Y_i) that are *identically and independently distributed* according to some (unknown) probability measure μ_h on a measurable space $(\mathcal{X} \times \mathcal{Y}, \Sigma_{(\mathcal{X}\times\mathcal{Y})})$, i.e. for all i the probability of a labeled pair $(X_i, Y_i) \in A \in \Sigma_{(\mathcal{X}\times\mathcal{Y})}$ is equal to $\mu_h(A)$.

Example 2.4. We assume that the instances (e.g. the applicants in Example 2.1) are generated by some probability distribution $\mu_{\mathcal{X}}$ but the predictor h_S is deterministic, i.e. h_S is a map from $\mathcal{X}$ to $\mathcal{Y}$. (The assumption that the instances are generated by a distribution $\mu_{\mathcal{X}}$ means that the space $\mathcal{X}$ is measurable, and μ is a probability measure on it. In this case we say that the probability that $X \in A \in \Sigma_{\mathcal{X}}$ is $\mu_{\mathcal{X}}(A)$). Now we shall compute the probability distribution of the pair $(x, (h(x)))$ which is generated probability distribution by x. Since h is deterministic, the pair $(x, h(x))$ is distributed by the probability $\mu_{(\mu_{\mathcal{X}},h)}$ defined as follows

$$\mu_{(\mu_{\mathcal{X}},h)}[(x, y)|\, x \in A \,\&\, y = h(x)] = \mu_{\mathcal{X}}(A). \tag{2.2}$$

The LHS of (2.2) can be rewritten as $\mu_{(\mu_{\mathcal{X}},h)}(\Gamma_h(A))$ - the probability measure of the restriction of the graph $\Gamma_h(A)$ of h over A (i.e. $\Gamma_h(A) = \{(x, y)|\, x \in A, y = h(x)\}$). In particular $\Gamma_h(A)$ is measurable, i.e. the σ-algebra $\Sigma_{(\mathcal{X}\times\mathcal{Y})}$ on $\mathcal{X} \times \mathcal{Y}$ contains the graph $\Gamma_h(A)$ for any $A \in \Sigma_{\mathcal{X}}$,

e.g. $\Sigma_{(\mathcal{X}\times\mathcal{Y})}$ is generated by $\{\Gamma_h(A)|\ A \in \Sigma_{\mathcal{X}}\}$. By Equation (2.2) the measure $\mu_{(\mu_{\mathcal{X}},h)}$ on $\mathcal{X}\times\mathcal{Y}$ is concentrated at the graph $\Gamma_f(\mathcal{X})$. Hence for any measurable subset $Z \subset \mathcal{X}\times\mathcal{Y}$ we have

$$\mu_{(\mu_{\mathcal{X}},h)}(Z) = \mu_{\mathcal{X}}(pr_{\mathcal{X}}(Z \cap \Gamma_h(\mathcal{X}))), \tag{2.3}$$

where $pr_{\mathcal{X}} : \mathcal{X}\times\mathcal{Y} \to \mathcal{X}$ denotes the projection onto $\mathcal{X}$.

Remark 2.5. (1) The learning scenario where as in Example 2.1 the distribution $\mu_{(\mu_{\mathcal{X}},h)}$ of labelled pairs (x, y) on $\mathcal{X}\times\mathcal{Y}$ is generated by a distribution $\mu_{\mathcal{X}}$ on $\mathcal{X}$ and a true predictor h, a measurable function from $\mathcal{X}$ to $\mathcal{Y}$ is called *deterministic* or *algorithmic*. Otherwise it is called *stochastic scenario*. In the deterministic learning scenario, the probability component expresses the uncertainty of input which has been observed/measured through a channel with noise. The output depends uniquely on input. Only in stochastic scenario our output are truly random variables, i.e. the correlation h between input and output label is uncertain/stochastic. This stochastic scenario is suitable for considering the last question in Remark 2.2.

(2) We will throughout assume that the corresponding σ-algebra on the probability space $\mathcal{X}\times\mathcal{Y}$ is a product of Borel σ-algebras on topological spaces $\mathcal{X}$ and $\mathcal{Y}$ and their product $(\mathcal{X}\times\mathcal{Y})^n$ w.r.t. the usual topologies.

2.1.3. *Rules of statistical learning.* In mathematical language we describe optimal predictor as a solution of an mathematical optimization problem. So the rule of statistical learning is to formulate this problem and suggest methods to solve it. Hence the goal of the leaner is to find a good hypothesis $h = h_S$ that minimizes a *risk/error* (that is maximizes a "success") for a given event S.

• *The true risk* (also called *the true error* or *generalization error*, *probability error*) $R_\mu(h')$ [7] of an arbitrary classifier $h' : \mathcal{X} \to \mathcal{Y}$ is the probability that h' does not predict the correct label that is distributed by a probability measure μ on $\mathcal{X}\times\mathcal{Y}$. Thus

$$R_\mu(h') := \mu(\{(x, y)|\, h'(x) \neq y\}) = 1 - \mu(\{x, h'(x)\}). \tag{2.4}$$

If we have the correct predictor h then the far RHS of (2.4) is zero by (2.4). Usually the lower script μ is omitted, so we write just $R(h)$, because we dont know μ and we wish to know it.

Example 2.6. Let us consider a deterministic learning scenario . In this case we have $\mu = \mu_{(\mu_{\mathcal{X}},h)}$ and using (2.3), (2.4) we obtain

$$R_\mu(h') = \mu_{\mathcal{X}}(\{x|\, h'(x) \neq h(x)\}). \tag{2.5}$$

Remark 2.7. We don't know μ and therefore we don't know R_μ. If μ is known and $\mathcal{Y}$ is finite the Bayes classifier (the solution of the learning problem) exists and can be found explicitly (Exercises 2.8, 2.9). In other

[7] R stands for "risk". Here we follow [Wolf2017]. In [SSBD2014, p. 31] the authors use the notation L instead of R but we shall use L for "loss" in Subsection 2.3, see also the footnote in Subsection 2.3 for related terminology.

words, the goal of the learner is reached. In general case we don't know μ and therefore we have to find a deterministic predictor h_S, given a sequence S of empirical data generated by the probability distribution μ corresponding to a *true predictor* h_μ, such that the error between h_S and h_μ is smallest possible. This motivates the notion of the empirical risk discussed below.

2.2. **ERM and overfitting.**

The empirical risk, also called *the training error*, is defined as follows

$$\hat{R}_S(h) := \frac{|i \in [m] : h(x_i) \neq y_i\}|}{m} \tag{2.6}$$

for a training data $S = \{(x_1, y_1), \cdots, (x_n, y_n)\}$.

The empirical risk is a function of two variables: the "empirical data" S and the predictor h. Given S a learner can compute $\hat{R}_S(h)$ for any function $h : \mathcal{X} \to \mathcal{Y}$. An optimal predictor should have minimal empirical risk. This is the *empirical risk minimization principle*.

This principle sounds good but we do not know if it works.

Below we shall consider a predictor $\hat{h}_S$, given a training data S, whose empirical risk $\hat{R}_S(\hat{h}_S)$ is zero, nevertheless whose true risk is equal to ε for any given $\varepsilon \in (0, 1)$.

Let $S = (x_1, y_1), \cdots, (x_n, y_n)$. Then we set

$$\hat{h}_S(x) = \begin{cases} y_i & \text{if there exists } i \in [n] \text{ s.t. } x_i = x \\ 0 & \text{otherwise.} \end{cases}$$

Clearly $\hat{R}_S(\hat{h}_S) = 0$. We also note that $\hat{h}_S(x) = 0$ except finite (at most n) points x in $\mathcal{X}$.

Now, given $\varepsilon \in (0, 1)$, we shall find a true predictor $h = h(\varepsilon) : \mathcal{X} \to \mathcal{Y}$, and its associated true risk such that

$$R_{\mu_{(D,h)}}(\hat{h}_S) = \varepsilon.$$

Let $\mathcal{X}$ be an open domain in $\mathbb{R}^k$, $k \geq 1$, and $\mathcal{Y} = \mathbb{Z}_2$. We decompose $\mathcal{X}$ into a disjoint union of two measurable subsets A_1 and A_2 such that $D(A_2) = \varepsilon$. Let $h : \mathcal{X} \to \mathbb{Z}_2$ be equal χ_{A_1} - the indicator function of A_1. By (2.5) we have

$$R_{\mu_{(D,h)}}(\hat{h}_S) = D(\{x \in \mathcal{X} | \, \hat{h}_S(x) \neq \chi_{A_1}(x)\}). \tag{2.7}$$

Since $\hat{h}_S(x) = 0$ a.e. on $\mathcal{X}$ it follows from (2.7) that

$$R_{\mu_{(D,h)}}(\hat{h}_S) = D(A_2) = \varepsilon.$$

Such a predictor $\hat{h}_S$ is said to be *overfitting*, i.e. it fits well to training data but not real life. Why?

(a) Possibly we should modify the definition of risk and then the notion of empirical risk?

(b) Is there a deeper reason for overfitting and how to deal with it?

In the remainder of this section we shall consider the suggestion (a) to modify the notion of risk. In later Subsections **??**, **??**, **??** we shall discuss Question (b).

2.3. **(General) Empirical Risk Minimization.** Generalizing the notion of true risk, we replace $R_\mu(h)$, $h \in \mathcal{Y}^{\mathcal{X}}$, by the notion of a generalized risk, which may fit better to a learning problem on $\mathcal{X} \times \mathcal{Y}$ with additional structures that are more suitable for measurement of "risk". First we choose a suitable *loss function*, also called *instantaneous loss function*, $L : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ [8] that measures how far $h(x)$ is from the respective y. The degree of "far" can be measured in terms of distance on metric spaces. If $\mathcal{Y}$ has no metric, then a loss function naturally has only two values (Yes/No) and the *true* loss function is defined as follows

$$L_{true}(y, y') = 1 - \delta_{y,y'}.$$

Note that the true risk of a hypothesis h is defined as average of the true loss

$$R(h) = R_\mu(h) = \int_{\mathcal{X}\times\mathcal{Y}} L_{true}(y, h(x))\, d\mu.$$

In the same way, *the generalized risk* depending on the loss function L is defined by

$$R^L_\mu(h) := \int_{\mathcal{X}\times\mathcal{Y}} L(y, h(x)) d\mu \tag{2.8}$$

(and hence is called some time by *expected loss*, *expected cost*, *averaged risk*). If L is fixed, then we also omit the superscript L.

For a given loss function L we also define the notion of *the empirical risk*:

$$\hat{R}^L_S(h) := \frac{1}{n} \sum_{i=1}^{n} L(y_i, h(x_i)) \in \mathbb{R} \tag{2.9}$$

for $S \in (\mathcal{X} \times \mathcal{Y})^n$. If L is fixed, then we also omit the superscript L.

The general principle of statistical learning states that an optimal predictor h_μ minimizes the generalized risk, and since the probability measure μ that governs the distribution of labeled pairs (x_i, y_i) is unknown, we wish to use the ERM principle, i.e. optimal predictor h_S must minimize the (generalized) empirical risk.

The case when $\mathcal{Y}$ is finite is simple.

Exercise 2.8 (Existence of Bayes classifier)**.** If $\mathcal{Y}$ is finite set then a minimizer, also called *a Bayes classifier*, of a generalized risk exists and can be expressed in an explicit formula.

[8] L stands for "loss", some time it is called a "cost function" and then it is denoted by C

Exercise 2.9 (The Bayes Optimal Predictor)**.** ([SSBD2014, p. 46]) If $\mathcal{Y} = \mathbb{Z}_2$ there is an explicit formula for a Bayes classifier, called the Bayes optimal predictor. Given any probability distribution D over $\mathcal{X} \times \{0,1\}$, the best label predicting function from $\mathcal{X}$ to $\{0,1\}$ will be

$$f_D(x) = \begin{cases} 1 & \text{if } r(x) := D[y=1|x] \geq 1/2 \\ 0 & \text{otherwise} \end{cases}$$

Show that for every probability distribution D, the Bayes optimal predictor f_D is optimal. In other words for every classifier g we have $R_D(f_D) \leq R_D(g)$.

If $\mathcal{Y}$ is not finite, a minimizer of a generalized risk may not exists. So we have to choose a natural loss function for which the existence of a minimizer of the averaged loss function is ensured. For this purpose, we often restrict our search for a predictor/estimator $\hat{h} : \mathcal{X} \to \mathcal{Y}$ to a subclass $\mathcal{F} \subset \mathcal{Y}^{\mathcal{X}}$ on which we can define a natural and efficient to compute generalized risk.

As an example of a choice of a natural risk that is defined on a natural hypothesis class $\mathcal{F}$ we consider a regression task, i.e. when the label set $\mathcal{Y}$ is $\mathbb{R}$. Note that there are natural embeddings

$$i_1 : \mathbb{R}^{\mathcal{X}} \to \mathbb{R}^{\mathcal{X}\times\mathcal{Y}} : \, [i_1(f)](x,y) = f(x),$$

$$i_2 : \mathbb{R}^{\mathcal{Y}} \to \mathbb{R}^{\mathcal{X}\times\mathcal{Y}} : \, [i_2(f)](x,y) = f(y).$$

(These embeddings are adjoint to the projections: $\mathcal{X}\times\mathbb{R} \to \mathcal{X}$ and $\mathcal{X}\times\mathcal{Y} \to \mathcal{Y}$.) For a given probability measure μ on $\mathcal{X} \times \mathbb{R}$ we set

$$L^2(\mathcal{X},\mu) := \{f \in \mathbb{R}^{\mathcal{X}} |\, i_1(f) \in L^2(\mathcal{X}\times\mathbb{R},\mu)\},$$

$$L^2(\mathbb{R},\mu) := \{f \in \mathbb{R}^{\mathbb{R}} |\, i_2(f) \in L^2(\mathcal{X}\times\mathbb{R},\mu)\}.$$

Now we let $\mathcal{F} := L^2(\mathcal{X},\mu)$. Let Y denote the function on $\mathbb{R}$ such that $Y(y) = y$. Assume that $Y \in L^2(\mathbb{R},\mu)$. Then we can define the *averaged loss/ expected risk* w.r.t. the *quadratic loss function* $L : \mathbb{R}\times\mathbb{R} \to \mathbb{R}$

$$L(y,y')) := |y-y'|^2, \tag{2.10}$$

$$R^L_\mu(h) = \mathbb{E}_\mu(|Y - h(X)|^2) = |i_2(Y) - i_1(h)|^2_{L^2(\mathcal{X}\times\mathbb{R},\mu)}. \tag{2.11}$$

The defined generalized risk R^L_μ is called the *L_2-risk*, also known as *mean squared error* (MSE).

Theorem 2.10 (Regression theorem)**.** *Assume that $h : \mathcal{X} \to \mathbb{R}$ belongs to the hypothesis class $\mathcal{F} = L^2(\mathcal{X},\mu)$ and Y belongs to $L^2(\mathbb{R},\mu)$. Then the regression function $r(x) := \mathbb{E}_\mu(i_2(Y)|i_1(X) = x)$ belongs to $\mathcal{F}$ and minimizes the $L_2(\mu)$-risk of h.*

Proof. Let us compute $R^L_\mu(h)$ using Pythagora's theorem. Denote by $\Pi_1 : L^2(\mathcal{X}\times\mathbb{R},\mu) \to i_1(L^2(\mathcal{X},\mu))$ the orthogonal projection. Then

$$R^L_\mu(h) = |i_2(Y) - \Pi_1(i_2(Y))|^2)_{L^2(\mathcal{X}\times\mathbb{R},\mu)} + |\Pi_1(i_2(Y)) - i_1(h)|^2_{L^2(\mathcal{X}\times\mathbb{R},\mu)}. \tag{2.12}$$

Using the interpretation of the condition expectation, we obtain

$$i_1(r) = P_1(i_2(Y)). \tag{2.13}$$

(see e.g. Theorem 3.3 in Appendix for conditional expectation). In particular, $i_1(r) \in \mathcal{F}$ and this proves the first assertion of Theorem 2.10.

Using (2.13) we obtain from (2.12)

$$R^L_\mu(h) = |i_2(Y) - i_1(r)|^2_{L^2(\mathcal{X}\times\mathbb{R},\mu)} + |i_1(r) - i_1(h)|^2_{L^2(\mathcal{X}\times\mathbb{R},\mu)}.$$

This implies Theorem 2.10 immediately. □

Theorem 2.10 says that the regression function $r(x) = E_\mu(Y|X = x)$ is the optimal predictor $\mathcal{X} \to \mathcal{Y}$, if $\mathcal{Y} = \mathbb{R}$. Note that the regression function is not a deterministic function, it is an element in $L^2(\mathcal{X}, \mu)$ and therefore is defined uniquely only up to the "induced" measure μ on $\mathcal{X}$. This theorem demonstrates the Bayes principle that if probability distribution μ of labeled pairs is known, then the optimal predictor h_μ can be expressed explicitly.

Exercise 2.11 (Empirical risk minimization)**.** Let $\mathcal{X} \times \mathcal{Y} = \mathbb{R}^d \times \mathbb{R}$ and $\mathcal{F} := \{h : \mathcal{X} \to \mathcal{Y} | \, \exists v \in \mathbb{R}^d : h(x) = \langle v, x\rangle\}$ be the class of linear functions in $\mathcal{Y}^\mathcal{X}$. For $S = ((x_i, y_i))_{i=1}^n \in (\mathcal{X} \times \mathcal{Y})^n$ and the quadratic loss L (defined in (2.10)), derive the hypothesis $\hat{h} \in \mathcal{F}$ that minimizes the empirical risk $\hat{R}^L_S$.

2.4. **Conclusion.** Statistical learning theory is the main ingredient of PAC theory which represents mathematical model of machine learning. The main problem of statistical learning theory is to find a (deterministic) predictor $h : \mathcal{X} \to \mathcal{Y}$ or more general, a stochastic predictor defined as a probability distribution μ_h of i.i.d. labeled pairs (X, Y) that fits empirical data/ training data $S = \{(x_1, y_1), \cdots, (x_n, y_n)\} \in \cup_N(\mathcal{X} \times \mathcal{Y})^N$ best in a sense that y_i approximates $h(x_i)$ for all i as close as possible. To formalize the notion of "best approximation" we introduce the notion of (generalized) risk and (generalized) empirical risk. If we know the distribution μ of the labeled pairs (x_i, y_i) then the Bayes principle, in particular, the regression theorem, produces the best predictor h_μ explicitly. The main problem is to find distribution μ of random variable (X, Y) once we know i.i.d. sequence of its values (x_i, y_i). In this lecture we approach this problem by considering the ERM principle and its generalization for a given loss function L. In next lecture we shall approach this problem from a geometrical approach.

3. Appendix: Some basic notions in mathematical statistics

3.1. **Conditional measures and conditional expectations.** ([Bogachev2007, Chapter 10, p. 339], [Kallenberg1997, Chapter 5, p. 80]) Modern probability theory can be said to begin with the notions of conditioning and disintegration. Conditioning is constantly used as a basic tool to describe and analyze systems involving randomness. The notion may be thought of in terms of averaging, projection, and disintegration-view points that are all essential for a proper understanding. In all but the most elementary contexts, one

defines conditioning with respect to a σ-field rather than a single event. In general, the result of the operation is not a constant but a random variable, measurable with respect to the given σ-field.

Let (Ω, Σ, μ) be a measure space and let $\mathcal{B}$ be a sub-σ-algebra in Σ.

Definition 3.1. Let $f \in L^1(\mu)$. A *conditional expectation* of f with respect to the σ-algebra $\mathcal{B}$ and the measure μ is a $\mathcal{B}$-measurable μ-integrable function $\mathbb{E}^{\mathcal{B}}_{\mu} f$ such that

$$\int_{\Omega} gf \, d\mu = \int_{\Omega} g\mathbb{E}^{\mathcal{B}}_{\mu} f \, d\mu \tag{3.1}$$

for every bounded $\mathcal{B}$-measurable function g.

A conditional expectation of an individual integrable function f is defined as the conditional expectation of the corresponding class in $L^1(\mu)$.

Remark 3.2. 1) The defining equality (3.1) is equivalent to the following relationship obtained by the substitution$g = 1_{\mathcal{B}}$:

$$f \, d\mu = \int_B \mathbb{E}^{\mathcal{B}}_{\mu} f \, d\mu \, \forall B \in \mathcal{B}. \tag{3.2}$$

This follows from the fact that every bounded $\mathcal{B}$-measurable function is the uniform limit of simple $\mathcal{B}$-measurable functions.

2) In the case where only one measure μ is given, for simplification of notation and terminology, in place of $\mathbb{E}^{\mathcal{B}}_{\mu} f$ one uses the symbol $\mathbb{E}^{\mathcal{B}} f$. Furthermore if $\mathcal{B} = \{\emptyset, \Omega\}$, then $\mathbb{E}^{\mathcal{B}}_{\mu} f$ coincides with the integral of f over Ω. If μ is a probability measure, the integral of f over the whole space Ω is denoted sometimes by $\mathbb{E}f$ and is called *the expectation of* f. In the probabilistic literature one frequently uses the notation $\mathbb{E}(f|\mathcal{B})$.

3) If Y is an integrable function on a probability space and $\mathcal{B}$ is generated by a measurable function (or mapping) η, then one uses the notation $\mathbb{E}(Y|\eta)$, i.e., $\mathbb{E}(Y|\eta) = \mathbb{E}^{\sigma(\eta)} Y$. Specializing further, the regression function is defined as follows

$$\mathbb{E}(Y|\eta = x) := \int_{[\eta = x]} \mathbb{E}(Y|\eta). \tag{3.3}$$

4) *The conditional probability* is defined using the notion of conditional expectation as follows

$$P[A|\mathcal{B}] = \mathbb{E}[1_A|\mathcal{B}].$$

that combine into a random probability measure on Ω.

Theorem 3.3. *(*[Bogachev2007, Theorem 10.1.5, p. 341]. *Suppose that μ is a probability measure. To every function $f \in L^1(\mu)$, one can associate a $\mathcal{B}$-measurable function $\mathbb{E}^{\mathcal{B}} f$ such that*
(1) $\mathbb{E}^{\mathcal{B}} f$ is a conditional expectation of f with respect to $\mathcal{B}$;
(2) $\mathbb{E}^{\mathcal{B}} f = f$ μ-a.e. for every $\mathcal{B}$-measurable μ-integrable function f ;
(3) $\mathbb{E}^{\mathcal{B}} f \geq 0$ μ-a.e. if $f \geq 0$ μ-a.e.;

(4) if a sequence of μ-integrable functions f_n converges monotonically decreasing or increasing to a μ-integrable function f, then $\mathbb{E}^{\mathcal{B}} f_n \to \mathbb{E}^{\mathcal{B}} f$ μ-a.e.; (5) For every $p \in [1, +\infty]$, the mapping $\mathbb{E}^{\mathcal{B}}$ defines a continuous linear operator with norm 1 on the space $L^p(\mu)$. In addition, $\mathbb{E}^{\mathcal{B}}$ is the orthogonal projection of $L^2(\mu)$ to the closed linear subspace generated by $\mathcal{B}$-measurable functions.

References

[Bishop2006] C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

[Bogachev2007] , V. I. Bogachev, Measure theory, Springer, 2007.

[Borovkov1998] A. A. Borovkov, Mathematical statistics, Gordon and Breach Science Publishers, 1998. 2006.

[HTF2008] T. Hastie, R. Tibshirani and J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer 2008.

[Kallenberg1997] O. Kallenberg, Foundations of modern Probability, Springer, 1997.

[MRT2012] M. Mohri, A. Rostamizadeh, A. Talwalkar, Foundations of Machine Learning, MIT Press, 2012.

[RN2010] S. J. Russell and P. Norvig, Artificial Intelligence A Modern Approach, Prentice Hall, 2010.

[SSBD2014] S. Shalev-Shwart, and S. Ben-David, Understanding Machine Learning: From Theory to Algorithms, Cambridge University Press, 2014.

[Vapnik1999] V. Vapnik, The nature of statistical learning theory, Springer, 1999.

[Wolf2017] M. M. Wolf, Mathematical Foundations of Machine Learning, lecture notes (2017).