MATHEMATICAL FOUNDATIONS OF MACHINE LEARNING

(NMAG 469, FALL TERM 2017-2018)

HÔNG VÂN LÊ *

Contents

Date: February 20, 2018.

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

It is not knowledge, but the act of learning ... which grants the greatest enjoyment.
Carl Friedrich Gauss

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.

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, 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".

1.2. **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 see 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 comes 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] the name AI was 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)

1.2.1. *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 on 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 pre-processing 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.

[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.

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

At the current time, unsupervised learning is primarily *descriptive* and experimental 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 intermediate 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.

• *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 *an agent* used to refer to the learner. The achievement of the agent's 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 a 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 structures are computable is 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 our needs for representation power in computing.

The first two methods are simple so we shall not discuss them 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 they are and little bit their methods. To understand machine learning deeply and its future directions we need to understand why machine learning works and what is its limitations. In short we want to study mathematical foundation of machine learning that discuss in deep the following questions:

Problem 1.1. *What is the mathematical model of learning?*

Problem 1.2. *How to quantify the difficulty/hardness/complexity of a learning problem?*

Problem 1.3. *How to choose a learning algorithm?*

Problem 1.4. *How to measure success of machine learning?*

Clearly Problem 1.1 should provide general guidance for investigating Problems 1.2, 1.3, 1.4. The investigation of Problem 1.3 depends on our understanding of Problems 1.2 and 1.4.

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 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 essence 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.

[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 view.

Finally I recommend some 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 is 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 is currently migrated to a new website.

2. Statistical learning framework for supervised learning

Last week I 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 Leslie Valiant 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/her graduate diploma, his/her computer sciences skill, rated by the mark (from 1 to 20) in his/her 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, correlations 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 correlate 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 inputs are the followings.

- *Domain set* (also called *input space*) - 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) $F(x)$,

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

- *Label set* - 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. For example, the applicant potential may take value in a nonempty set S consisting of more than ten elements. By choosing an arbitrary subset S_0 in S we can simplify our decision problem into the case when S has two elements. Thus binary classification problem is the simplest and most basic problem of machine learning. The choice of a mathematical representation of real world outputs in machine learning is called *feature learning*, which is extremely important in deep learning, see Section 12.

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

The learner output is *a prediction rule* - 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].

In other words the learner needs to find *an algorithm* (a "computable" map)

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

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 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 correct 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/coordinate/feature 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 $A \in \Sigma_{\mathcal{X}}$ - the sigma-algebra of the measurable space $(\mathcal{X}, \Sigma_{\mathcal{X}})$, 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) the element y_i may not be represented as the value of a function of x at x_i. We express the random nature of the labeled 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 $\mu_h(B)$.

[6] h stands for "hypothesis"

Now let us formulate this very general assumption on incomplete information of labeled pairs (x_i, y_i) 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 $\mu_h(A)$. (In this general case only the distribution measure of the labeled pairs (x, y) is important: (x_i, y_i) are in *a correlation* defined on the product space $\mathcal{X} \times \mathcal{Y}$.)

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 $\mathcal{Y}$ [7]. (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 $\mu_{\mathcal{X}}(A)$). Since h is deterministic, the pair $\big(x, h(x)\big)$ is distributed by the probability $(\Gamma_h)_*\mu$ defined as follows (cf. [MRT2012, (2.1), p. 12], [SSBD2014, (2.1), p. 34][8])

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

Clearly the LHS of (2.2) is equal to the value of the push-forwarded measure of μ via the map $\Gamma_h : \mathcal{X} \to \mathcal{X} \times \mathcal{Y}$, $x \mapsto (x, h(x))$ applied to the graph $\Gamma_h(A) \subset \mathcal{X} \times \mathcal{Y}$.

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 $\{\Gamma_h(A)|\, A \in \Sigma_{\mathcal{X}}\}$.

In what follows we shall show that (2.2) defines the probability measure $(\Gamma_h)_*\mu$ on $\mathcal{X} \times \mathcal{Y}$ uniquely.

Since $[(\Gamma_h)_*\mu]\big(\Gamma_f(\mathcal{X})\big) = 1$ by (2.2), the support of $(\Gamma_h)_*\mu$ is $\Gamma_f(\mathcal{X})$. It follows that for any measurable subset $Z \subset \mathcal{X} \times \mathcal{Y}$ we have

$$[(\Gamma_h)_*\mu](Z) = [(\Gamma_h)_*\mu]\big(Z \cap \Gamma_h(\mathcal{X})\big).$$

Hence, by (2.2) we have

$$[(\Gamma_h)_*\mu](Z) = \mu_{\mathcal{X}}\Big(\Gamma_h^{-1}\big(Z \cap \Gamma_h(\mathcal{X})\big)\Big). \tag{2.3}$$

Equation (2.3) shows that the value of the measure $(\Gamma_h)_*\mu$ applied to any measurable subset $Z \subset \mathcal{X} \times \mathcal{Y}$ is equal to the value of the push-forwarded measure of μ via Γ_h applied to Z.

---

[7] for any given $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ the probability that $y = h_S(x)$ is equal to either 1 or 0.

[8] The cited authors did not compute the measure $(\Gamma_h)_*\mu$ on $\mathcal{X} \times \mathcal{Y}$, they defined the generalization risk by using the measure $\mu_{\mathcal{X}}$ and the RHS of (2.2), see also Subsection 2.1.3 in the next page.

Remark 2.5. (1) The learning scenario where as in Example 2.4 the distribution $(\Gamma_h)_*\mu$ of labeled 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 topology.

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*) $R_\mu(h')$ [9] 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 definition. Usually the lower script μ is omitted, so we write just $R(h)$, because we don't know μ and we wish to know it.

Example 2.6. Let us consider a deterministic learning scenario. In this case we have $\mu = (\Gamma_h)_*\mu_{\mathcal{X}}$ and using (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 2.11, 2.12). In other 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.

[9] 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.

2.2. **ERM and overfitting.**

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

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

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 $R_S(h)$ 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 h_S, given a training data S, whose empirical risk $R_S(h_S)$ is zero, nevertheless whose true risk is equal to ε for any given $\varepsilon \in (0, 1)$.

Example 2.8. Let $S = \{(x_1, y_1), \cdots, (x_n, y_n)\}$. Then we set

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

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

Now, given a probability D on $\mathbb{R}^2$ whose support is $\mathbb{R}^2$ and a positive number $\varepsilon \in (0, 1)$, we shall find a true predictor $h = h(\varepsilon) : \mathcal{X} \to \mathcal{Y}$, and its associated true risk such that

$$R_{[(\Gamma_h)_* D]}(h_S) = \varepsilon.$$

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 1_{A_1} - the indicator function of A_1. By (2.5) we have

$$R_{[(\Gamma_h)_* D]}(h_S) = D(\{x \in \mathcal{X} | \, h_S(x) \neq 1_{A_1}(x)\}). \tag{2.7}$$

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

$$R_{[(\Gamma_h)_* D]}(h_S) = D(A_2) = \varepsilon.$$

Such a predictor h_S is said to be *overfitting*, i.e. it fits well to training data but not real life.

Why does overfitting happen?

Problem 2.9. *Possibly we should modify the definition of risk and then the notion of empirical risk?*

Problem 2.10. *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 Section 6 we shall discuss Problem 2.10.

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*, $L : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ [10] 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

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

(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*:

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

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

Exercise 2.12 (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

[10] L stands for "loss", some time it is called a "cost function" and then it is denoted by C, see also (5.3) for a more general form of a loss function.

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 exist. 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 $h : \mathcal{X} \to \mathcal{Y}$ 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* $L : \mathbb{R}\times\mathbb{R} \to \mathbb{R}$

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

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

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

Theorem 2.13 (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 $r(x) := \mathbb{E}_\mu\big(i_2(Y)|i_1(X) = x\big)$ 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 $\Pi_1 : L^2(\mathcal{X}\times\mathbb{R},\mu) \to i_1\big(L^2(\mathcal{X},\mu)\big)$ the orthogonal projection. Then

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

Using the interpretation of the conditional expectation (see e.g. Theorem 13.4 in Appendix), we obtain

$$i_1(r) = \Pi_1\big(i_2(Y)\big). \tag{2.13}$$

It follows that $r \in \mathcal{F}$. This proves the first assertion of Theorem 2.13.

Using (2.13) we obtain from (2.12)

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

This implies Theorem 2.13 immediately. □

Theorem 2.13 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 2.14 (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 $S = \big((x_i, y_i)\big)_{i=1}^n \in (\mathcal{X} \times \mathcal{Y})^n$ and the quadratic loss L (defined in (2.10)), derive the hypothesis $\hat{h} \in \mathcal{F}$ that minimizes the empirical risk $\hat{R}_S^L$.

Remark 2.15. In Subsection 5.1 we generalize further the notion of a loss function and the notion of an expected risk/error function, see (5.2), (5.3).

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 the next lecture we shall approach this problem from a geometrical approach.

3. Statistical models and the Cramér-Rao inequality

Let us recall that in machine learning, uncertainty comes in many forms. Since training data can be measured only approximately correctly and we do not have certain information on the hypothesis class $\mathcal{F}$ where a best predictor should be search for, we replace a determined predictor $h : \mathcal{X} \to \mathcal{Y}$ by its stochastic version: a probability measure μ_h on $\Omega = \mathcal{X} \times \mathcal{Y}$. It is also necessary to find a "best" subset $\mathcal{F}$ of possible predictors/hypotheses from the whole space $\mathcal{Y}^\mathcal{X}$ of all predictors, so that on $\mathcal{F}$ we can define a reasonable (generalized) risk function and irregularity like overfitting does not take place. Today we shall discuss the notion of a good class $\mathcal{F}$ of

stochastic hypotheses from a geometer point of view. More concrete, we consider the following two problems:

Problem 3.1. *To find a mathematical model/representation for a subset $P_{\mathcal{F}}$ of probability distributions on a measurable space (Ω, Σ) where we can do analysis and geometry.*

Problem 3.2. *Using geometric and analytic methods derive best possible estimation of the underlying probability measure $\mu_h \in P_{\mathcal{F}}$, given an observable (data) $S \in \Omega$.*

To find a best possible estimation of the underlying probability measure $\mu_h \in P_{\mathcal{F}}$, given an observable (data) $S \in \Omega$, is one main part of theoretical statistic, which is called *estimation of unknown parameter*.

Remark 3.3. Murphy in his book [Murphy2012, §1.3, p. 9] regards many unsupervised learning problems as *unconditional density estimation* problems, and supervised problems as a *conditional density estimation* problems. Regarding labeled data as data, we consider supervised learning problems as particular cases of parameter estimation problem. The parameter in consideration is the unknown hypotheses h_S in a hypothesis class, or more general, a parameter in a subset $P_{\mathcal{F}}$ of the set $\mathcal{P}(\Omega, \Sigma)$ of all probability measures on Ω.

In statistics, a family $\mathcal{P}_{\mathcal{F}}$ of probability measures on (Ω, Σ) is also called *a statistical model*. Our first aim is to find a good analytic and geometric properties of the set $\mathcal{P}(\Omega, \Sigma)$ of all probability measures on (Ω, Σ).

In this section we fix Σ and write Ω instead of (Ω, Σ).

3.1. **The space of all probability measures and total variation norm.** It is a general idea in mathematics that we need to "represent" mathematical objects of interest as a subset S (or subcategory) of some "regular" space (resp. category) E with nice structure, so that S inherits the nice structure and we have more mathematical technique to analyze properties of S under investigation. For example, in algebra we represent groups as subgroups of linear transformations of a linear (finite or infinite dimensional) vector space. This is called *representation theory*. In algebraic geometry we represent a system of polynomial equations with the set of it solutions in an *affine or projective space*. The associated set will be called an algebraic set and we can define *a topology* on it, which is a good tool for analyzing the system of polynomial equations. In PDE theory we embed a space of smooth functions into the Hilbert space of L^2-functions, etc. where we have an additional structure to analyze the solutions. Later in our lecture course we shall apply similar ideas in the theory of support vector machines.

Let us return back to the space $\mathcal{P}(\Omega)$ for an arbitrary set Ω. We shall show that this space can be regarded as a convex subset of an (possibly infinite dimensional) Banach space, where we can do standard analysis.

Let us fix some notations. Recall that a signed finite measure μ on Ω is a function $\mu : \Sigma \to \mathbb{R}$ which satisfies all axioms of a measure except that μ needs not take non-negative value.

$$\begin{aligned}
\mathcal{P}(\Omega) &:= \{\mu \ : \ \mu \text{ a probability measure on } \Omega\} \\
\mathcal{M}(\Omega) &:= \{\mu \ : \ \mu \text{ a finite measure on } \Omega\} \\
\mathcal{S}(\Omega) &:= \{\mu \ : \ \mu \text{ a signed finite measure on } \Omega\} \\
\mathcal{S}_0(\Omega) &:= \{\mu \in \mathcal{S}(\Omega) \ : \ \int_\Omega d\mu = 0\}.
\end{aligned}$$

Clearly, $\mathcal{P}(\Omega) \subset \mathcal{M}(\Omega) \subset \mathcal{S}(\Omega)$. It is known that $\mathcal{S}_0(\Omega) \subset \mathcal{S}(\Omega)$ are Banach spaces whose norm is given by the total variation of a signed measure, defined as

$$\|\mu\|_{TV} \ := \ \sup \sum_{i=1}^{n} |\mu(A_i)|$$

where the supremum is taken over all finite partitions $\Omega = A_1 \dot\cup \ldots \dot\cup A_n$ with disjoint sets $A_i \in \Sigma$ (see e.g. [Halmos1950]). Here, the symbol $\dot\cup$ stands for the disjoint union of sets.

Let me describe the total variation norm using the Jordan decomposition theorem for signed measures, which is an analogue of the decomposition theorem for a measurable function. For a measurable function $\phi : \Omega \to [-\infty, \infty]$ we define $\phi_+ := \max(\phi, 0)$ and $\phi_- := \max(-\phi, 0)$, so that $\phi_\pm \geq 0$ are measurable with disjoint support, and

$$\phi = \phi_+ - \phi_- \qquad\qquad |\phi| = \phi_+ + \phi_-. \tag{3.1}$$

Similarly, by the *Jordan decomposition theorem*, each measure $\mu \in \mathcal{S}(\Omega)$ can be decomposed uniquely as

$$\mu = \mu_+ - \mu_- \qquad \text{with } \mu_\pm \in \mathcal{M}(\Omega),\ \mu_+ \perp \mu_-. \tag{3.2}$$

That is, there is a *Hahn decomposition* $\Omega = P \dot\cup N$ with $\mu_+(N) = \mu_-(P) = 0$ (in this case the measures μ_+ and μ_- are called *mutually singular*). Thus, if we define

$$|\mu| := \mu_+ + \mu_- \in \mathcal{M}(\Omega),$$

then (3.2) implies

$$|\mu(A)| \leq |\mu|(A) \qquad \text{for all } \mu \in \mathcal{S}(\Omega) \text{ and } A \in \Sigma, \tag{3.3}$$

so that

$$\|\mu\|_{TV} = \| \ |\mu| \ \|_{TV} = |\mu|(\Omega).$$

In particular,

$$\mathcal{P}(\Omega) = \{\mu \in \mathcal{M}(\Omega) \ : \ \|\mu\|_{TV} = 1\}.$$

Next let us consider important subsets of dominated measures and equivalent measures in the Banach space $\mathcal{S}(\Omega)$ which are most frequently used subsets in statistics and ML.

Definition 3.4. Let μ be a finite measure and ν be a finite signed measure on a measurable space Ω. We say that ν is dominated by μ (or equivalently, ν is absolutely continuous w.r.t. μ), if each μ-null set is a $|\nu|$-null set. Two finite measures are called *equivalent*, if they dominate each other and hence have the same null sets.

Given a measure $\mu_0 \in \mathcal{M}(\Omega)$, we let

$$\begin{aligned} \mathcal{P}(\Omega,\mu_0) &:= \{\mu \in \mathcal{P}(\Omega) \,:\, \mu \text{ is dominated by } \mu_0\} \\ \mathcal{M}(\Omega,\mu_0) &:= \{\mu \in \mathcal{M}(\Omega) \,:\, \mu \text{ is dominated by } \mu_0\} \\ \mathcal{P}_+(\Omega,\mu_0) &:= \{\mu \in \mathcal{P}(\Omega,\mu_0) \,:\, \mu \text{ is equivalent to } \mu_0\} \\ \mathcal{M}_+(\Omega,\mu_0) &:= \{\mu \in \mathcal{M}(\Omega,\mu_0) \,:\, \mu \text{ is equivalent to } \mu_0\} \\ \mathcal{S}(\Omega,\mu_0) &:= \{\mu \in \mathcal{S}(\Omega) \,:\, \mu \text{ is dominated by } \mu_0\} \\ \mathcal{S}_0(\Omega,\mu_0) &:= \mathcal{S}(\Omega,\mu_0) \cap \mathcal{S}_0(\Omega). \end{aligned} \tag{3.4}$$

By the Radon-Nikodym theorem (see Appendix Theorem 13.1), we may canonically identify $\mathcal{S}(\Omega,\mu_0)$ with $L^1(\Omega,\mu_0)$ by the correspondence

$$\imath_{can} : L^1(\Omega,\mu_0) \longrightarrow \mathcal{S}(\Omega,\mu_0), \qquad \phi \longmapsto \phi\,\mu_0. \tag{3.5}$$

With this, $\mathcal{M}(\Omega,\mu_0) = \{\phi\mu_0 \;:\; \phi \geq 0\}$ and $\mathcal{M}_+(\Omega,\mu_0) = \{\phi\mu_0 \;:\; \phi > 0\}$ and the corresponding descriptions apply to $\mathcal{P}(\Omega,\mu_0)$ and $\mathcal{P}_+(\Omega,\mu_0)$, respectively. Observe that $\imath_{can}$ is an isomorphism of Banach spaces, since evidently

$$\|\phi\|_{L^1(\Omega,\mu_0)} = \int_\Omega |\phi|\, d\mu_0 = \|\phi\,\mu_0\|_{TV}.$$

Example 3.5. Let $\Omega_n := \{\omega_1, \cdots, \omega_n\}$ be a finite set of n elementary events. Let δ_{ω_i} denote the Dirac measure concentrated at ω_i. Then

$$\mathcal{S}(\Omega_n) = \{\mu = \sum_{i=1}^n x_i\delta_{\omega_i}\} = \mathbb{R}^n(x_1, \cdots, x_n)$$

and

$$\mathcal{M}(\Omega_n) = \{\sum_{i=1}^n x_i\delta_{\omega_i} |\, x_i \geq 0\} = \mathbb{R}^n_{\geq 0}.$$

For $\mu \in \mathcal{M}(\Omega_n)$ of the form

$$\mu = \sum_{i=1}^k c_i\delta_i,\ c_i > 0$$

we have $||\mu||_{TV} = \sum c_i$. Thus the space $L^1(\Omega_n,\mu)$ with the total variation norm is isomorphic to $\mathbb{R}^k$ with the l^1-norm. The space $\mathcal{P}(\Omega_n)$ with the induced total variation topology is homeomorphic to a $(n-1)$-dimensional simplex $\{(c_1, \cdots, c_n) \in \mathbb{R}^n_+ |\ \sum_i c_i = 1\}$.

Exercise 3.6. ([JLS2017]) For any countable family of signed measures $\{\mu_n \in \mathcal{S}(\omega)\}$ show that there exists a measure $\mu \in \mathcal{M}(\Omega)$ dominating all measures μ_n.

On (possibly infinite dimensional) Banach spaces we can do analysis, since we can define the notion of differentiable mappings. Let V and W be Banach spaces and $U \subset V$ an open subset. A map $\phi : U \to W$ is called *differentiable at* $x \in U$, if there is a bounded linear operator $d_x\phi \in Lin(V, W)$ such that

$$\lim_{h\to 0} \frac{\|\phi(x+h) - \phi(x) - d_x\phi(h)\|_W}{\|h\|_V} = 0. \tag{3.6}$$

In this case, $d_x\phi$ is called the *(total) differential of* ϕ *at* x. Moreover, ϕ is called *continuously differentiable* or shortly a C^1*-map*, if it is differentiable at every $x \in U$, and the map $d\phi : U \to Lin(V, W)$, $x \mapsto d_x\phi$ is continuous. Furthermore, a differentiable map $c : (-\varepsilon, \varepsilon) \to W$ is called a *curve in* W.

3.2. **Statistical model, predictor and estimator.** In this subsection we shall induce the geometry of the Banach space $\mathcal{S}(\Omega)$ to it statistical models.

3.2.1. *A statistical model and its logarithmic tangent space.*

Definition 3.7. A subset $P \subset \mathcal{P}(\Omega)$ is called *a statistical model.*

Usually a statistical model P is described with a help of a *parameterization*, i.e. P is the image of a map $\mathbf{p} : M \to \mathcal{P}(\Omega)$ where M is a "nice" geometric object, e.g. M is an open subset in a Banach space (in general case P may not be an open subset in $\mathcal{P}(\Omega) \subset \mathcal{S}(M)$), or more general, M is a differentiable Banach manifold. A differentiable Banach manifold is a topological space which locally is homeomorphic to an open subset in a fixed Banach space B and there is a rule, called *a transition map*, that allow us to patch (analytic) calculations on local coordinates on M (i.e. on each model open subset in the fixed Banach space). A good book on Banach manifolds is [Lang2002]. We need only very elementary knowledge of Banach manifolds for today lecture. You can assume that M is an open subset of a Banach space and the general case follows without any difficulty.

Definition 3.8. (cf. [AJLS2017]) A statistical model $P \subset \mathcal{P}(\Omega)$ is called *a parameterized statistical model*, if $P = \mathbf{p}(M)$ where M is a differentiable Banach manifold and $\mathbf{p} : M \to \mathcal{S}(\Omega)$ is a differentiable map. A parameterized statistical model P together with its parametrization $p : M \to \mathcal{S}(\Omega)$ will be denoted by $(M, \Omega, \mathbf{p})$.

Exercise 3.9. Using Exercise 3.6, show that any parameterized statistical model $(M, \Omega, \mathbf{p})$ is dominated by a measure μ_0, i.e. $\mathbf{p}(x)$ is dominated by μ_0 for each $x \in m$, if M is finite dimensional manifold.

Remark 3.10. 1. Usually in statistics and machine learning we consider only statistical models that are dominated by some (probability) measure. There are two reasons for considering only such statistical models: firstly, most important statistical models are finite dimensional and by Exercise 3.9 they are dominated by a measure, secondly - most analytic tools applied in statistics are developed for finite dimensional families of measures. In

our works [AJLS2016], [AJLS2017] we developed new language and methods that are well suitable for analyzing infinite dimensional parameterized statistical models which need not be families of dominated measures.

2. A family of probability measures $\{\mu \in \mathcal{P} \subset \mathcal{P}(\Omega)\}$ that are dominated by a measure μ_0 is represented by the *density functions* $\frac{d\mu}{d\mu_0}$ - the Radon-Nikodym derivative of μ wrt to μ_0 (or the Radon-Nikodym density of μ w.r.t. μ_0). We also write $\mu = \frac{d\mu}{d\mu_0}\mu_0$. Once the base measure μ_0 is fixed, the statistical model P is represented by the corresponding set of density functions $\frac{d\mu}{d\mu_0}$. If P is a parameterized measure model we shall write $P = (M, \Omega, \mu_0, p)$ where $p(x) = \frac{d\mathbf{p}(x)}{d\mu_0}$. In this case, the problem of parameter estimation is called a density estimation problem.

Example 3.11. The most widely used distribution in statistics and machine learning is the Gaussian or normal (family of) distributions. It is a 2-dimensional parametrized statistical model $(H^2_+, \mathbb{R}, dx, \mathcal{N}(x))$ of dominated probability measures on $\mathbb{R}$ whose density function $\mathcal{N}$ is given by

$$\mathcal{N}(x|\,\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\big(-\frac{1}{2\sigma^2}(x-\mu)^2\big). \tag{3.7}$$

Here the parameter (μ, σ^2) is in the non-negative half-plane H^2_+ and $x \in \mathbb{R}$.

Example 3.12. The multivariate Gaussian (family of) probability distributions, also denoted by MVN (multivariate normal), is a muti-dimensional generalization of the 2-dimensional family of normal distributions. It is a $\frac{n(n+3)}{2}$-dimensional parameterized statistical model of dominated measures on $\mathbb{R}^n$ whose density function $\mathcal{N}$ is given as follows

$$\mathcal{N}(x|\,\mu, \Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^2} \exp\big(-\frac{1}{2}\Sigma^{-1}(x-\mu, x-\mu)\big) \tag{3.8}$$

Here the couple (μ, Σ) is the parameter of the family, where $\mu \in \mathbb{R}^n$ and Σ is a positive definite quadratic form (symmetric bilinear form) on $\mathbb{R}^n$. Using the canonical Euclidean metric on $\mathbb{R}^n$, we identify $\mathbb{R}^n$ with its dual space $(\mathbb{R}^n)^*$. Thus the inverse Σ^{-1} is regarded as a quadratic form on $\mathbb{R}^n$ and the norm $|\Sigma|$ is also well-defined.

To do analysis on statistical models we need to introduce the notion of the tangent fibration of a statistical model. We study a geometric object S by investigating the space of functions on S (which is a linear infinite dimensional vector space) and by investigating its dual version: the space of all curves on S. The tangent fibration of S describes the first order approximation of the later space.

Definition 3.13. ([AJLS2017, Definition 3.2, p. 141]) Let $(V, \|\cdot\|)$ be a Banach space, $X \subset V$ an arbitrary subset and $x_0 \in X$. Then $v \in V$ is called *a tangent vector of X at x_0*, if there is a curve $c : (-\varepsilon, \varepsilon) \to X \subset V$ such that $c(0) = x_0$ and $\dot{c}(0) = v$.

The tangent (double) cone C_xX at a point $x \in X$ is defined as the subset of the tangent space $T_xV = V$ that are tangent to a curve lying in X. The *tangent space* T_xX is the linear hull of the tangent cone.

If $P = (M, \Omega, \mathbf{p})$ is a parametrized statistical model then we should look at not only the tangent space TM but only the image of TM under the differential $d\mathbf{p}$ of the map $\mathbf{p}$.

Definition 3.14. *The reduced tangent space* $\hat{T}_mM$ at the point $m \in M$ of a parametrized statistical model $(M, \Omega, \mathbf{p})$ is defined to be the quotient $T_mM/\ker d_m\mathbf{p}$.

Example 3.15. Let us consider the normal mixture family $(W, \mathbb{R}, dx, p)$ where

$$W = \{(a, b) \in \mathbb{R}^2 |\, a \in [0, 1], b \in \mathbb{R}\}$$

$$p(x|a, b) := \frac{(1-a)e^{-x^2/2} + ae^{-(x-b)^2/2}}{\sqrt{2\pi}}.$$

This family is a typical example of Gaussian mixture models which comprise also the changing time model (the Nile River model) and the ARMA model in time series [Amari2016, §12.2.6, p. 311]. We compute

$$\partial_a p(x|a, b) = \frac{-e^{-x^2/2} + e^{-(x-b)^2/2}}{\sqrt{2\pi}},$$

$$\partial_b p(x|a, b) = \frac{a(x-b)e^{-(x-b)^2/2}}{\sqrt{2\pi}}.$$

Hence $\partial_a p(x|a, b) = 0\, \forall x$ iff $b = 0$ and $\partial_b p(x|a, b) = \forall x$ iff $a = 0$. Furthermore it is not hard to see that ($\partial_a p(x|a, b)$ and $\partial_b p(x|a, b)$) are linearly independent. Thus the singularity of $(W, \mathbb{R}, dx, p)$ is $\{a = 0\} \cup \{b = 0\}$. Furthermore $\hat{T}_{(0,0)}W = \{pt\}$, $\hat{T}_{(a,0)}W = \mathbb{R}^2/(\mathbb{R}, 0)$ for $a \neq 0$, $\hat{T}_{(0,b)}W = \mathbb{R}^2/(0, \mathbb{R})$ for $b \neq 0$.

Exercise 3.16. (cf. [AJLS2016, Theorem 2.1]) Let P be a statistical model. Show that for all $v \in T_\xi P \subset \mathcal{S}(\Omega)$ v is dominated by ξ. Hence *the logarithmic tangent vector* $\log v := dv/d\xi$ is an element of $L^1(\Omega, \xi)$.

The set of all logarithmic tangent vectors ξ at a point $\xi \in P$ will be denoted by $T_{\log \xi}P$ and called *the logarithmic tangent cone of* P *at* ξ whose linear hull will be called *the logarithmic tangent space of* P *at* ξ and denoted by $T_{\log \xi}P$.

Example 3.17. Assume that a statistical model P consists of measures dominated by a measure μ_0 and therefore P is regaded as a family of density functions on Ω, namely

$$P = \{f \cdot \mu_0 |\, f \in L^1_P(\Omega, \mu_0) \subset L^1(\Omega, \mu_0)\}. \tag{3.9}$$

Then a tangent vector $v \in T_\xi P$ has the form $v = \dot{f}(0) \cdot \mu_0$ and the logrithmic tangent vector

$$\frac{dv}{d\xi} = \frac{\dot{f}(0)}{f(0)} = \frac{d}{dt}_{|t=0}(\log f(t)). \tag{3.10}$$

The RHS of (3.10) motivates the name "logarithmic tangent vector" for the far LHS of (3.10).

Using the notion of the tangent fibration TP, we extend the notion of C^1-maps between Banach spaces to the case when the domain of a C^1-map is a statistical model $P \subset \mathcal{P}(\Omega)$.

Definition 3.18. Let $P \subset V := \mathcal{P}(\Omega)$ be a statistical model and W a Banach space. A map $\phi : P \to W$ is called *differentiable at* $x \in P$, if there exists a bounded linear operator $d_x\phi \in Lin(T_xP, W)$ such that (3.6) holds. A function $\phi : P \to \mathbb{R}$ is called *Gatêau-differentiable* at $x \in P$, if its is differentiable at x on each curve on P through x.

3.2.2. *Predictor and estimators.* Now we reformulate the notion of a stochastic predictor in a slightly more general terms of an estimator.

Definition 3.19. Given a (possibly parameterized) statistical model P, a map $\hat{\sigma} : \Omega \to P$ is called *an estimator*.

Remark 3.20. The notion of a C^1-map is well-defined for a map between parameterized statistical models. It is not clear how to define the notion of a C^1-function on a unparameterized statistical model $P \subset \mathcal{P}(\Omega)$.

Next, we are going to investigate the dependence of the efficiency of an arbitrary estimator on P from the geometry of P.

3.3. **The Fisher metric, MSE and variance of estimator.** To do analysis on a statistical model P we need a metric on P, a Riemannian metric, if we want to employ analytic methods. As we learned last week with the regression problem, it is important to narrow the hypothesis class to define a good generalized risk, namely the L_2-risk, which is also called MSE.

We also wish to measure the efficiency of our estimator $\hat{\sigma} : \Omega \to P$ via MSE. For this purpose we need further formalization. In general case P is a subset of an infinite dimensional space $\mathcal{P}(\Omega)$ and to define a point $\xi \in P$ we need its coordinates, or certain features of $\xi \in P$ which is formalized as a vector valued map $\varphi : P \to V$, where V is a vector space (resp. a map $\varphi : M \to V$ in the case of parameterized statistical model $(M, \Omega, \mathbf{p})$).

Definition 3.21. *A φ-estimator* is a composition of an estimator $\hat{\sigma} : \Omega \to P$ and a map $\varphi : P \to V$, where V is a topological vector space.

Next we want to put a Riemannian metric on P i.e., to put a positive quadratic form $\mathfrak{g}$ on each tangent space $T_\xi P$ (respectively we want to put a positive quadratic form on each reduced tangent space $\hat{T}_m M$ of a parameterized statistical model $(M, \Omega, \mathbf{p})$). By Exercise 3.16, the logarithmic tangent

space $T_{\log\xi}P$ is a subspace in $L^1(\Omega,\xi)$. The space $L^1(\Omega,\xi)$ does not have a natural metric but its subspace $L^2(\Omega,\xi)$ is a Hilbert space. So we put the following restriction on P

$$T_{\log\xi}P \subset L^2(\Omega,\xi). \tag{3.11}$$

Now we define *the Fisher metric* on $T_\xi P$ as follows. For each $v,w \in C_\xi P$ we set

$$\mathfrak{g}(v,w) := \langle \log v, \log w\rangle_{L^2(\Omega,\xi)} = \int_\Omega \log v \cdot \log w \, d\xi. \tag{3.12}$$

This formula is well-defined, since $\log v \cdot \log w \in L^1(\Omega,\xi)$ by (3.11). Since $T_\xi P$ is a linear hull of $C_\xi P$, the formula (3.12) extends uniquely to a positive quadratic form on $T_\xi P$.

Example 3.22. 1. Let Ω_n be a finite sample space of n elementary events. As in Example 3.5, let δ_{ω_i} denote the Dirac measure concentrated at ω_i. Then any $\mu \in \mathcal{M}(\Omega_n)$ can be written as

$$\mu = \sum_{i=1}^n \mu_i \delta_{\omega_i} \tag{3.13}$$

where $\mu_i \geq 0$. Then, for all $k \geq 1$,

$$L^k(\Omega_n,\mu) = \{f \in \mathbb{R}^{\Omega_n} \mid f = \sum_{i=1}^n a_i (sign\,\mu_i)\delta_i\,, a_i \in \mathbb{R}\} = \mathcal{S}(\Omega[\mu]),$$

where $sign\,0 = 0$, δ_i is the Dirac function on Ω_n, i.e., $\delta_i(\omega_i) = \delta^i_j$, and $\Omega[\mu] = \{\omega_k \in \Omega_n | \, \mu_k > 0\}$.

It follows that for any subset $M \subset \mathcal{M}(\Omega_n)$ and any $\xi \in M$ we have

$$T_{\log\xi}M \subset L^1(\Omega_n,\xi) = L^2(\Omega_n,\xi).$$

Hence the Fisher metric is well-defined on M, since by Example 3.22.1 Formula (3.11) is well-defined on

$$T_\xi \mathcal{M}_+(\Omega_n,\mu_n) = \mathcal{S}(\Omega_n) = L^2(\Omega_n,\mu_n) \supset T_{\log\xi}M.$$

2. Let us compute the Fisher metric on the statistical model $\mathcal{P}_+(\Omega_n) = \mathcal{P}_+(\Omega_n, \mu_n := \sum_{i=1}^n \delta_{\omega_i})$. For any $\mu \in \mathcal{M}_+(\Omega_n)$ let $\bar\mu := d\mu/d\mu_n$ - the density function of the measure μ w.r.t. the counting measure μ_n. Then

$$\bar\mu(\omega_i) = \mu(\omega_i) = \mu_i. \tag{3.14}$$

Set

$$\Lambda_\mu : T_\mu \mathcal{M}_+(\Omega_n) \to L_2(\Omega_n,\mu),\ \Lambda_\mu(v) = \partial_v \log\bar\mu.$$

Then, using (3.14), the Fisher metric defined in (3.11) can be rewritten as follows for any $v,w \in T_\mu\mathcal{M}_+(\Omega_n)$

$$\mathfrak{g}_\mu(v,w) = \int_{\Omega_n} \Lambda_\mu(v)\cdot\Lambda_\mu(w)\, d\mu = \sum_{i=1}^n \partial_v\mu_i \cdot \partial_w\mu_i \cdot (\mu_i)^{-1} \tag{3.15}$$

Since $\mathcal{P}_+(\Omega_n) \subset \mathcal{M}_+(\Omega_n)$ and hence $T_\mu(\mathcal{P}_+(\Omega_n)) \subset T_\mu(\mathcal{M}_+(\Omega_n))$ Formula (3.15) also defines the Fisher metric on $T_\mu(\mathcal{P}_+(\Omega_n))$.

Exercise 3.23. Let Ω_n denote a sample set of n-elements $\omega_1, \cdots, \omega_n$. As before, δ_{ω_i} denotes the Dirac measure concentrated at ω_i. Let $p : (0,1) \to \mathcal{P}(\Omega_n)$ denote the parametrized statistical curve defined by

$$p(t) = \frac{1-t^n}{1-t} \sum_{i=1}^{n} t^{i-1} \delta_{\omega_i}.$$

Compute the Fisher metric on $p(t)$.

Remark 3.24. (1) Originally the Fisher metric has been defined on a finite dimensional parametrized statistical model $(M, \Omega, \mathbf{p})$ in the same way. By Exercise 3.9, all the measure $\mathbf{p}(m), m \in M$, is dominated by a measure μ_0. So we write $\mathbf{p}(m) = \bar{p}(m) \cdot \mu_0$. For $v, w \in T_m M$ we set (cf. (3.12))

$$\mathfrak{g}(v,w) := \int_\Omega \partial_v \bar{p}(m) \cdot \partial_w \bar{p}(m) \cdot \big(\bar{p}(m)\big)^{-1} d\mu_0.$$

This formula is a generalization of Formula (3.15). Moreover it is obtained from the Fisher metric on the subset $\mathbf{p}(M) \subset \mathcal{P}(\Omega)$ via the map $d\mathbf{p}$. Hence it defines a non-degenerate (Riemannian) metric, if and only if $\ker d\mathbf{p} = 0$.

(2) The Fisher metric has been defined by Fisher in 1925 as an "information" quantity of a parametrized statistical model. It was Rao who recognised that the Fisher information quantity is a Riemannian metric on parametrised statistical models. We refer [AJLS2017] for a historical account. The Fisher metric is also called information metric, or information Fisher metric. It enjoys many invariance properties that make the Fisher metric good structure on statistical models satisfying (3.11). One of most notable applications of the Fisher metric is the Cramér-Rao inequality which measures our ability to have a good estimator in terms of geometry of the underlying statistical model, see Theorem 3.32 below.

Using the Fisher metric we shall define a MSE, a generalized risk function, for a φ-estimator.

As in the case of L_2-risk for the regression problem in the previous lecture, we also need to restrict the class of estimators $\hat{\sigma}$ for a correct definition of the generalized MSE corresponding to a given "feature function" φ. Since $\varphi \circ \hat{\sigma}$ takes value in V, it is useful to look at each coordinate of $\varphi \circ \hat{\sigma}$, i.e. the function of the form

$$\varphi^l \circ \hat{\sigma} := \langle l, \varphi \circ \hat{\sigma} \rangle$$

for each $l \in V^*$. Here $\varphi^l = l \circ \varphi$ is the "l-th-coordinate of φ.

Set

$$L^2_\varphi(P, \Omega) := \{\hat{\sigma} : \Omega \to P | \, \varphi^l \circ \hat{\sigma} \in L^2(\Omega, \xi) \text{ for all } \xi \in P\}.$$

If $\hat{\sigma} \in L^2_{\varphi}(P, \Omega)$, then the following L_2-risk function (or MSE) is well-defined for any $l, k \in V^*$

$$MSE^{\varphi}_{\xi}[\hat{\sigma}](l,k) := \mathbb{E}_{\xi}[(\varphi^l \circ \hat{\sigma} - \varphi^l \circ \xi) \cdot (\varphi^k \circ \hat{\sigma} - \varphi^k \circ \xi)]. \tag{3.16}$$

Thus the $MSE^{\varphi}_{\xi}[\hat{\sigma}](l,l)$ is the L_2-risk of the quadratic lost function

$$L : \mathbb{R} \to \mathbb{R},\ L(\hat{\sigma}) = |\varphi^l \circ \hat{\sigma} - \varphi^l \circ \xi|^2.$$

In the formula (3.16) we adopt a point of view that for a general vector space V (the value space of a general estimator) we should consider the L_2-risk, the MSE, as a quadratic form on the dual space V^*, i.e. $MSE^{\varphi}_{\xi}[\hat{\sigma}]$ is a quadratic function of $l \in V^*$.

Next we introduce a new quantity - the mean value $\varphi_{\hat{\sigma}}$ of a φ-estimator $\varphi \circ \hat{\sigma}$. We consider $\varphi_{\hat{\sigma}}$ as a V^{**}-valued function on P as follows

$$\langle \varphi_{\hat{\sigma}}(\xi), l \rangle := \mathbb{E}_{\xi}(\varphi^l \circ \hat{\sigma}) = \int_{\Omega} \varphi^l \circ \hat{\sigma}\, d\xi \tag{3.17}$$

for any $l^* \in V^*$.

Definition 3.25. The difference

$$b^{\varphi}_{\hat{\sigma}} := \varphi_{\hat{\sigma}} - \varphi \in V^P \tag{3.18}$$

will be called the *bias of the estimator $\hat{\sigma}$ w.r.t. the map φ.*

Definition 3.26. Given an estimator $\hat{\sigma} \in L^2_{\varphi}(P, \Omega)$ the estimator $\hat{\sigma}$ will be called *φ-unbiased*, if $\varphi_{\hat{\sigma}} = \varphi$, equivalently, $b^{\varphi}_{\hat{\sigma}} = 0$.

Using the mean value $\varphi_{\hat{\sigma}}$, we define the *variance of $\hat{\sigma}$ w.r.t. φ* as the derivation of $\varphi \circ \hat{\sigma}$ from its mean value $\varphi_{\hat{\sigma}}$. We set for all $l \in V^*$

$$V^{\varphi}_{\xi}[\hat{\sigma}](l) := \mathbb{E}_{\xi}[(\varphi^l \circ \hat{\sigma} - \varphi^l_{\hat{\sigma}}) \cdot (\varphi^l \circ \hat{\sigma} - \varphi^l_{\hat{\sigma}})]. \tag{3.19}$$

The RHSs of (3.19) is well-defined, since $\hat{\sigma} \in L^2_{\varphi}(P, \Omega)$. It is a quadratic form on V^* and will be denoted by $V^{\varphi}_{\xi}[\hat{\sigma}]$.

Exercise 3.27. ([JLS2017]) Prove the following formula

$$MSE^{\varphi}_{\xi}[\hat{\sigma}](l,k) = V^{\varphi}_{\xi}[\hat{\sigma}](l,k) + \langle b^{\varphi}_{\hat{\sigma}}(\xi), l \rangle \cdot \langle b^{\varphi}_{\hat{\sigma}}(\xi), k \rangle \tag{3.20}$$

for all $\xi \in P$ and all $l, k \in V^*$.

3.4. **A general Cramér-Rao inequality.** So far we have found a good condition on a statistical model $P \subset \mathcal{P}(\Omega)$ so that we can define the MSE for a φ-estimator $\hat{\sigma} : \Omega \to V$ that generalizes the L_2-risk for the regression problem considered in the previous lecture. To get a more refined theory we need a little bit more regularity of a statistical model P and a "regularity" condition on estimators $\hat{\sigma}$.

Definition 3.28. (cf. [AJLS2017]) A statistical model P is called *2-integrable*, if (1) the Fisher metric $\mathfrak{g}$ is well-defined on the tangent fibration TP, i.e. the relation (3.11) holds at each point $\xi \in P$, and (2) the metric $\mathfrak{g}$ is weakly continuous, i.e., the map $v \mapsto ||v||_{\mathfrak{g}}$ is continuous on TP.

Remark 3.29. If P is finite dimensional, then it is not hard to see that the weak continuity of a Riemannian metric on P is equivalent to the continuity of the metric $\mathfrak{g}$. For a general case of (possibly infinite dimensional) P we refer to [JLS2017, Theorem 2.7] for interpretation of the weak continuity of the Fisher metric.

Definition 3.30. [JLS2017] An estimator $\hat{\sigma} \in L^2_\varphi(P, \Omega)$ is called *φ-regular* if for all $l \in V^*$ the function $\xi \mapsto ||\varphi^l \circ \hat{\sigma}||_{L^2(\Omega,\xi)}$ is locally bounded, i.e. for all $\xi_0 \in P$

$$\lim_{\xi \to \xi_0} \sup ||\varphi^l \circ \hat{\sigma}||_{L^2(\Omega,\xi)} < \infty.$$

Proposition 3.31. *(cf.* [JLS2017]*) Let $P \subset \mathcal{P}(\Omega)$ be a 2-integrable statistical model, φ - a V-valued function on P and $\hat{\sigma} \in L^2_\varphi(P, \Omega)$ - a φ-regular estimator. Then $\varphi^l_{\hat{\sigma}}$ is Gatêau-differentiable function on P for any $l \in V^*$. Moreover for any $\xi \in P$ we have*

$$V^\varphi_\xi[\hat{\sigma}](l, l) := \mathbb{E}_\xi(\varphi^l \circ \hat{\sigma} - \mathbb{E}_\xi(\varphi^l \circ \hat{\sigma}))^2 \geq \|d\varphi^l_{\hat{\sigma}}\|^2_{\mathfrak{g}^{-1}}(\xi).$$

Regarding $\|d\varphi^l_{\hat{\sigma}}\|^2_{\mathfrak{g}^{-1}}(\xi)$ as a quadratic form on V^* and re-denoting this quadratic form by $(\mathfrak{g}^\varphi_{\hat{\sigma}})^{-1}$ we obtain

Theorem 3.32 (The Cramér-Rao inequality)**.** *(cf.* [JLS2017]*) Let P be a statistical model, φ a V-valued function on P and $\hat{\sigma} \in L^2_\varphi(P, \Omega)$ a φ-regular estimator. Then the difference $V^\varphi_\xi[\hat{\sigma}] - (\mathfrak{g}^\varphi_{\hat{\sigma}})^{-1}(\xi)$ is a positive semi-definite quadratic form on V^* for any $\xi \in P$.*

3.5. **Conclusion.** In this lecture we have considered the problem of defining a good hypothesis class and a good predictor from the most general point of views of theory of parameter estimations in mathematical statistics. We define the tangent fibration on each statistical model P - that is a subset in the set of all probability measures on a measurable space Ω. With a further mild restriction of P we define the Fisher metric on the tangent fibration of P and the general L_2-risk function - the MSE which is a quadratic form on the dual vector space V^* where V is the target vector space of a φ-estimator $\varphi \circ \hat{\sigma} : \Omega \to P \to V$. Under a mild regularity assumption on the φ-estimator $\varphi \circ \hat{\sigma} : \Omega \to P \to V$ we derive the Cramér-Rao inequality giving a lower bound for the variance (and hence for the MSE) of the (possibly biased) φ-estimator $\varphi \circ \hat{\sigma}$. This provides a simple solution to Problem 1.4 in the framework of classical statistics at the beginning of our lecture: an unbiased φ-estimator is successful, also called efficient, if it reaches the equality in the Cramér-Rao inequality. We shall investigate efficient φ-estimators deeper in the next lecture.

4. Efficient estimators

In the last lecture we derived the general Cramér-Rao inequality (Theorem 3.32). We shall clarify the importance of the general Cramér-Rao inequality in the first part of today lecture. First we shall derive from

the general Cramér-Rao inequality classical Cramér-Rao inequality (4.9) in which the relation between the MSE of an unbiased estimator and the Fisher metric is expressed. Then we shall reconsider the maximum likelikhood estimator (Example 4.5) using the Cramér-Rao inequality (Theorem 4.7).

In the second part of today lecture we shall discuss three important questions related to the Cramér-Rao equality. First we introduce the following

Definition 4.1. A φ-regular estimator $\hat{\sigma} : \Omega \to P$ is called *efficient*, if the Cramér-Rao inequality in (4.3) holds for any $\xi \in M$ and any $l \in V^*$.

Problem 4.2. *How to find an efficient φ-estimator $\varphi \circ \hat{\sigma}$, i.e., $\varphi \circ \hat{\sigma}$ reaches the equality in the Cramér-Rao inequality?*

Problem 4.3. *Which statistical models admit efficient φ-estimators?*

Problem 4.4. *Is there a sequence of estimators $\hat{\sigma}_n : (\Omega)^n \to P$ such that their MSE tend to zero as n go to infinity?*

If Problem 4.4 has a positive answer this implies that the learning problem with the generalized risk MSE has a solution in the hypothesis class/statistical model P. The estimators satisfying the conditions in Problem 4.4 are called *asymptotic Fisher efficient*.

Before discussing these questions let me present an important class of estimators that have been recommended, analyzed and widely popularized by Ronald Fisher between 1912 and 1922 (although it had been used earlier by Carl Friedrich Gauss, Pierre-Simon Laplace etc.).

Example 4.5 (MLE)**.** A natural way to construct an estimator is the *maximum likelihood method*, also denoted by MLE. Assume that $P = (M, \Omega, \mathbf{p})$ is a parameterized statistical model where M is a finite dimensional manifold, e.g., M is an open subset in $\mathbb{R}^n$. By Exercise 3.9 we can write $P = (M, \Omega, \mathbf{p} = p \cdot \mu_0)$ for some based dominating measure μ_0. We also assume that the map $\mathbf{p} : M \to \mathcal{P}(\Omega)$ is injective, i.e. the map $p : M \to L^1(\Omega, \mu_0)$ is injective. Given $x \in \Omega$, one selects $\hat{\sigma}(x) := p_{ML}(x) \in P$ to be the element of P with the highest density at the observed point x

$$p\big(x; p_{ML}(x)\big) := \max_{\xi \in M} p(x; \xi). \tag{4.1}$$

Note that a *maximum likelihood* solution $p_{ML} \in P^\Omega$ of the equation (4.1) may not exist if M is not compact.

If the RHS of (4.1) is well-defined, then its value does not depend on the choice of the base μ_0 since changing μ_0 only introduces a factor that does not depend on $\xi \in M$.

The equation (4.1) yields the necessary condition at every element $x \in \Omega$ expressed in term of the vanishing of the first variation of the density function p w.r.t. the variable $\xi \in M$ at $\xi = p_{ML}(x)$:

$$\frac{d}{d\xi}_{|\xi = p_{ML}(x)} p(x, \xi) = 0. \tag{4.2}$$

A solution $p_{ML} : \Omega \to P, x \mapsto p_L(x)$, of (4.2) is called *a (relaxed) maximum likelihood estimator (MLE).*[11]

4.1. **Consequences of the Cramér-Rao inequality.** From the Cramér-Rao inequality for (unparameterized) statistical models P (Theorem 3.32) we obtain immediately the following Cramér-Rao inequality for 2-integrable parameterized statistical models $(M, \Omega, \mathbf{p})$ [JLS2017]

$$V^{\varphi}_{\mathbf{p}(\xi)}[\hat{\sigma}](l,l) := \mathbb{E}_{\mathbf{p}(\xi)}(\varphi^l \circ \hat{\sigma} - \mathbb{E}_{\mathbf{p}(\xi)}(\varphi^l \circ \hat{\sigma}))^2 \geq \|d\varphi^l_{\hat{\sigma}}\|^2_{\hat{\mathfrak{g}}^{-1}}(\xi). \tag{4.3}$$

Here $\varphi : M \to V$ is a feature function, $\hat{\mathfrak{g}}$ is the reduced Fisher metric on the reduced tangent space $\hat{T}M$ obtained from the Fisher metric $\mathfrak{g}$ by the following formula

$$\hat{\mathfrak{g}}(\hat{v}, \hat{w}) := \mathfrak{g}(v, w) := \mathfrak{g}\big(d\mathbf{p}(v), d\mathbf{p}(w)\big). \tag{4.4}$$

for any $\hat{v}, \hat{w} \in \hat{T}P$ and their representatives $v, w \in TP$.

We shall derive classical Cramér-Rao inequalities from the Cramér-Rao inequality (4.3) by restricting ourself to the case P is a 2-integrable statistical model which is bijectively parameterized by an open subset in $\mathbb{R}^n$, i.e, $P = (D, \Omega, \mathbf{p})$ where $D \subset \mathbb{R}^n$ is an open set and $\mathbf{p} : D \to \mathcal{P}(\Omega)$ is an injective map and the Fisher metric is well-defined and continuous on D, and furthermore, φ is a coordinate mapping on P, i.e., its components φ^l are the coordinate functions of P.

(A) Assume that $V = \mathbb{R}^n$, $P \subset \mathbb{R}^n$ and $\varphi : P \to \mathbb{R}^n$ is a coordinate mapping, i.e. φ is the natural embedding. Then $d\varphi^l = d\xi^l$, and with (3.18), abbreviating $b^{\varphi}_{\hat{\sigma}}$ as b, we write

$$(\mathfrak{g}^{\varphi}_{\hat{\sigma}})^{-1}(\xi)(l,k) = \langle \sum_i^n (\frac{\partial \xi^l}{\partial \xi^i} + \frac{\partial b^l}{\partial \xi^i}) d\xi^i, \sum_{j=1}^n (\frac{\partial \xi^k}{\partial \xi^j} + \frac{\partial b^k}{\partial \xi^j}) d\xi^j \rangle_{\mathfrak{g}^{-1}}(\xi). \tag{4.5}$$

Let $D(\xi)$ be the linear transformation of V whose matrix coordinates are

$$D(\xi)^l_k := \frac{\partial b^l}{\partial \xi^k}.$$

With (4.5), the Cramér-Rao inequality in Theorem 3.32 becomes

$$V_{\xi}[\hat{\sigma}] \geq (\mathbb{E} + D(\xi))\mathfrak{g}^{-1}(\xi)(\mathbb{E} + D(\xi))^T. \tag{4.6}$$

The inequality (4.6) coincides with the Cramér-Rao inequality in [Borovkov1998, Theorem 1.A, p. 147]. The condition (R) in [Borovkov1998, p. 140, 147] for the validity of the Cramér-Rao inequality is essentially equivalent to the 2-integrability of the (finite dimensional) statistical model with positive regular density function under consideration, more precisely Borokov ignores/excludes the points $x \in \Omega$ where the density function vanishes for computing the Fisher metric. Since we do not assume the existence of a positive regular density function, our set-up is more general than that

[11] In statistics and ML the method of transformation of the original ML equation (4.1) into the relaxed version (4.2) is called *the variational method* [Bishop2006, Chapter 10].

by Borokov. Borovkov also uses the φ-regularity assumption, written as $\mathbb{E}_\theta((\theta^*)^2) < c < \infty$ for $\theta \in \Theta$, see also [Borovkov1998, Lemma 1, p. 141] for a more precise formulation.

(B) Specializing further and assuming that $V = \mathbb{R}$ and φ is a coordinate mapping. Then

$$\mathbb{E} + D(\xi) = 1 + b'_{\hat{\sigma}} \tag{4.7}$$

where $b_{\hat{\sigma}}$ is short for $b^{\varphi}_{\hat{\sigma}}$. Using (4.7) and (3.20), we derive from (4.6)

$$\mathbb{E}_\xi(\hat{\sigma} - \xi)^2 \geq \frac{[1 + b'_{\hat{\sigma}}(\xi)]^2}{\mathfrak{g}(\xi)} + b_{\hat{\sigma}}(\xi)^2. \tag{4.8}$$

(4.8) is identical with the Cramér-Rao inequality with a bias term in [CT2006, (11.290) p.396,(11.323) p.402].

(C) Assume that V is finite dimensional, φ is a coordinate mapping and $\hat{\sigma}$ is φ-unbiased. Then the terms involving $b_{\hat{\sigma}}$ vanish, and the Cramér-Rao inequality in Theorem 3.32 becomes the well-known Cramér-Rao inequality for an unbiased estimator which is written in almost all books on mathematical statistics.

$$V_\xi[\hat{\sigma}] \geq \mathfrak{g}^{-1}(\xi). \tag{4.9}$$

The RHS of (4.9) is the inverse of the Fisher metric on a statistical model P. The Fisher metric on a parameterized statistical model $(M, \Omega, \mathbf{p})$ does not depend on the parameterization as we know that it is the pull back of the Fisher metric on the image $\mathbf{p}(M)$. The Fisher metric measures the information of the underlying statistical model [AJLS2015, AJLS2017, Le2016] and is an important ingredient of information geometry. The Cramér-Rao inequality plays an important role in the information theory and we refer the reader to [JLS2017] for a survey of the state of art of generalizations and applications of the Cramér-Rao inequality.

4.2. **Efficient estimators and MLE.** In this subsection we study Problem 4.2 of finding efficient φ-estimators on finite dimensional parameterized statistical models (M, Ω, μ_0, p), i.e, M is a finite dimensional manifold. Let $\varphi : M \to V$ be a feature map. Denote the projection $TM \to TM/\ker \mathbf{p}$ by pr. We shall prove the following

Proposition 4.6. *Assume that $\hat{\sigma}$ is φ-regular. Then $d\varphi^l_{\hat{\sigma}}$ descends to a linear function on the quotient $\hat{T}_\xi M = T_\xi M/\ker \mathbf{p}$.*

As a consequence, for any $V \in T_\xi M$ there exists a unique vector $\nabla_{\hat{\mathfrak{g}}}\varphi^l_{\hat{\sigma}} \in \hat{T}_\xi M$ such that the following equality holds

$$d\varphi^l_{\hat{\sigma}}(V) = \hat{\mathfrak{g}}(pr(V), \nabla_{\hat{\mathfrak{g}}}\varphi^l_{\hat{\sigma}}). \tag{4.10}$$

We think of $\nabla_{\hat{\mathfrak{g}}}\varphi^l_{\hat{\sigma}}$ as the gradient of the function $\varphi^l_{\hat{\sigma}}$, cf. (10.3).

Theorem 4.7. *Assume that (M, Ω, μ, p) is a 2-integrable parameterized statistical model which admits a φ-unbiased efficient estimator $\hat{\sigma}$. Assume the linear mapping $V \to \hat{T}_\xi M$, $l \mapsto \nabla_{\hat{g}} \varphi^l_{\hat{\sigma}}$ is surjective for all $\xi \in M$. Then $\varphi \circ \hat{\sigma}$ is an MLE.*

Theorem 4.7 justifies the popular choice of MLE as a best estimator for MSE. [12] The MSE-risk is a function on the space of estimators and MLE is point-wise on Ω minimization of the likelihood function $\log(x, \xi(x))$, $x \in \Omega$.

Proof of Proposition 4.6. We recall the following

Lemma 4.8 (Differentiation under integral sign)**.** *(*[JLS2017]*) Assume that $\hat{\sigma}$ is φ-regular. Then for any $l \in V^*$ the function $\varphi^l \circ \hat{\sigma}$ is Gateau-differentiable and for all $V \in T_\xi M$ we have*

$$\partial_V(\varphi^l_{\hat{\sigma}}) = \varphi^l \circ \hat{\sigma} \cdot \partial_V\big(\log p(\xi)\big)\, \mathbf{p}(\xi). \tag{4.11}$$

Consequently, for a constant function φ and a constant estimator $\hat{\sigma} : \Omega \to M$ we obtain

$$\int_\Omega \partial_V \log p(x, \xi)\, p(x, \xi) = 0 \tag{4.12}$$

for all $\xi \in M$ and $V \in T_\xi M$.

It follows from (4.11) and (4.12)

$$\partial_V \varphi^l_{\hat{\sigma}} = \int_\Omega (\varphi^l \circ \hat{\sigma} - \mathbb{E}_{\mathbf{p}(\xi)}(\varphi^l \circ \hat{\sigma}) \cdot \partial_V\big(\log p(\xi)\big)\, \mathbf{p}(\xi). \tag{4.13}$$

From (4.13) it follows immediately Poroposition 4.6. □

Proof of a weak version of Theorem 4.7. We shall give a proof of Theorem 4.7 under a mild condition that p is a positive regular density function, i.e. we have ([JLS2017])

$$\frac{d\big(\partial_V \mathbf{p}(\xi)\big)}{d\mathbf{p}(\xi)} = \partial_V \log p. \tag{4.14}$$

Assume that $\hat{\sigma} : \Omega \to M$ is φ-regular and φ-efficient. Then the Cramér-Rao inequality turns to an equality iff for all $\xi \in M$ and for all $l \in V^*$ we have

$$\mathbb{E}_{\mathbf{p}(\xi)}\big(\varphi^l \circ \hat{\sigma} - \mathbb{E}_{\mathbf{p}(\xi)}(\varphi^l \circ \xi)\big) = ||d\varphi^l_{\hat{\sigma}}||^2_{\hat{\mathfrak{g}}^{-1}}(\xi). \tag{4.15}$$

For any $\xi \in M$ the map

$$e : \hat{T}_\xi M \to L^2\big(\Omega, \mathbf{p}(\xi)\big),\ V \mapsto \partial_V \log p(\xi)$$

is an embedding. This embedding is an isometric embedding w.r.t. the reduced Fisher metric $\hat{\mathfrak{g}}$ on $\hat{T}_\xi M$ defined (3.12) and the L_2-metric on $L^2(\Omega, \mathbf{p}(\xi))$.

[12] The converse of Theorem 4.7 is not correct, i.e., there is MLE which is not unbiased and underestimates the variance [Bishop2006, p. 27-28].

Since $e(\hat{T}_\xi M)$ is a closed (finite dimensional) subspace we have the orthogonal decomposition

$$L^2\big(\Omega, \mathbf{p}(\xi)\big) = e(\hat{T}_\xi M) \oplus e(\hat{T}_\xi M)^\perp.$$

Denote by $\Pi_{e(\hat{T}_\xi M)}$ the orthogonal projection of $L^2(\Omega, \mathbf{p}(\xi))$ to $e(\hat{T}_\xi M)$. Then it follows from (4.13)

$$d\varphi^l_{\hat{\sigma}}(V) = \langle \varphi^l \circ \hat{\sigma} - \mathbb{E}_{\mathbf{p}(\xi)}(\varphi^l \circ \hat{\sigma}), e(V)\rangle_{L^2(\Omega, \mathbf{p}(\xi))}.$$

Compare this equation with (4.10), we obtain immediately

$$\Pi_{e(\hat{T}_\xi M)}\big(\varphi^l \circ \hat{\sigma} - \mathbb{E}_{\mathbf{p}(\xi)}(\varphi^l \circ \hat{\sigma})\big) = e(\nabla_{\hat{\mathfrak{g}}}\varphi^l_{\hat{\sigma}}). \tag{4.16}$$

Since $p : M \to \mathbb{R}$ is a C^1-function and $\ker dp = \ker d\mathbf{p}$ the derivation $\partial_V p$ depend only on the projection $pr(V)$. Hence we also write

$$\partial_{pr(V)} p := \partial_V p.$$

Using this new notation, we obtain from the definition of the Fisher metric

$$\partial_{\nabla_{\hat{\mathfrak{g}}}\varphi^l_{\hat{\sigma}}} \log p(\cdot, \xi) = e(\nabla_{\hat{\mathfrak{g}}}\varphi^l_{\hat{\sigma}}). \tag{4.17}$$

Note that the RHS of (4.17) is equal to the LHS of (4.16) which vanishes at $\xi = \hat{\sigma}(x)$ since $\hat{\sigma}$ is φ-unbiased. Hence the LHS of (4.17) vanishes at $\xi = \hat{\sigma}(x)$ for all l. By the surjectivity assumption in Theorem 4.7, this implies that $\partial_V \log p(x, \xi) = 0$ at $\xi = \hat{\sigma}(x)$ for all x. This implies that $\varphi \circ \hat{\sigma}$ is MLE. □

Exercise 4.9. Assume that $V = \mathbb{R}^n$, $P \subset \mathbb{R}^n$ and $\varphi : P \to \mathbb{R}^n$ is a coordinate mapping, i.e. φ is the natural embedding. Using Theorem 4.7, show that $\hat{\sigma} : M \to P$ is MLE if it is (φ)-unbiased and efficient.

Remark 4.10. 1. The proof of Theorem 4.7 is carried in the same way but with more complicated notations.[13]

2. The unbiasedness seems a good condition but unbiased estimators may not exist, see e.g. [LC1998, Example 1.2, p. 83]. We shall discuss this problem in the last part of our lecture.

Exercise 4.11. Explain the interpretation of maximum likelihood estimators as empirical risk minimizers for stochastic predictors in [SSBD2014, 24.1.2, p. 345], using strict mathematical language.

[13] In his classical book "Homology theory and cohomology theory" appeared in 1978, W. Massey commented to the proof of Proposition A.18 that the main difficulty in the proof is to use certain complicated notations.

4.3. **Efficient estimators and exponential models.** The MLE condition gives us a receipt how to define an efficient estimator σ. Now we shall investigate the second question on the relation between the existence of an efficient estimator and the geometry of the underlying statistical model.

First I shall reformulate the classical definition of an exponential family, slightly generalizing this notion.

First we introduce the notion of *parameterized measure model* $(M, \Omega, \mathbf{p})$ ([AJLS2017]) which is the same as the notion of a parameterized statistical model in Definition 3.8, but we relax the condition that $\mathbf{p}(M) \subset \mathcal{P}(\Omega)$ and instead we pose the condition that $\mathbf{p}(M) \subset \mathcal{M}(\Omega)$.

Definition 4.12. A parameterized measure model (D, Ω, μ, p) is called *an un-normalized exponential family*, if there are continuous mappings to a topological vector space V and its dual V^*

$$F : D \to V \text{ and } T : \Omega \to V^*$$

such that

$$p(\theta) := g(\theta) \cdot e^{\langle F(\theta), T(x)\rangle}. \tag{4.18}$$

We shall denote by $(D, \Omega, V, F, T, g, \mu)$ the un-normalized exponential family defined by (4.18).

If M is an open subset of V, F is the canonical embedding and $g(\theta) = 1$, then (4.18) is called *the canonical form* of the exponential family.

An un-normalized exponential family will be called *an exponential family*, if $\mathbf{p}(D) \subset \mathcal{P}(\Omega)$.

Example 4.13. Examples of classical exponential family are the multivariate Gaussian family of probability distributions (Example 3.12) and many important families of distributions e.g. the gamma distribution, chi-squared distributions, ect., [Borovkov1998, LC1998].

Other *infinite dimensional exponential families* considered in [PS1995], see also [AJLS2015, AJLS2017], are also examples of exponential families in sense of Definition 4.12.

Theorem 4.14. *Assume that (M, Ω, μ, p) is a finite dimensional open connected 2-integrable parameterized statistical model with regular density function p (i.e., 4.14 holds), moreover* $\ker d\mathbf{p} : TM \to \mathcal{S}(\Omega)$ *is zero. Let $\hat{\sigma} : \Omega \to P$ be a φ-efficient estimator for a V-valued function φ on P. Suppose that the map*

$$I : V^* \times M \to T_\xi M,\ (l, \xi) \mapsto \nabla_{\mathfrak{g}} \varphi^l_{\hat{\sigma}}(\xi)) \cdot \mathbf{p}(\xi)$$

is surjective and for any $l \in V^*$ and any $\xi \in M$. Then (M, Ω, μ, p) is an exponential family.

Proof. The idea of the proof of Theorem 4.14 consists in the following. Since $\hat{\sigma}$ is φ-efficient, by (4.16) the image of the gradient of $\varphi^l \circ \hat{\sigma}$ via the embedding e at every point $\xi \in M$ is a given function on Ω. Using this observation we

study the gradient lines of $\varphi^l_{\hat\sigma}$ on M and show that these lines define an affine coordinate system on M in which we can find an explicit *canonical form* of an exponential family for M.

By (4.16), and recalling that $\ker d\mathbf{p} = 0$ and $\varphi \in L^2_{\hat\sigma}(\Omega)$, for each $l \in V' \setminus \{0\}$, we have

$$e(\nabla_{\mathfrak{g}}\varphi^l_{\hat\sigma}(\xi)) = \varphi^l \circ \hat\sigma - \mathbb{E}_{\mathbf{p}(\xi)}(\varphi^l \circ \hat\sigma) \in L^2_P(\Omega). \tag{4.19}$$

We abbreviate $\nabla\varphi^l_{\hat\sigma}$ as $X(l)$. Using (4.17) we have

$$\partial_{X(l)} \log p(\cdot;\xi) = \varphi^l \circ \hat\sigma - \mathbb{E}_{\mathbf{p}(\xi)}(\varphi^l \circ \hat\sigma). \tag{4.20}$$

Note that $\varphi^l \circ \hat\sigma$ does not depend on ξ and the last term on the RHS of (4.20), the expectation value $\mathbb{E}_{\mathbf{p}(\xi)}(\varphi^l \circ \hat\sigma)$ is differentiable in ξ by Lemma 4.8. Hence the vector field $X(l)$ on P is locally Lipschitz continuous, since $\mathbf{p} : P \to \mathcal{S}(\Omega)$ is a C^1-immersion. For a given point $\xi \in P$ we consider the unique integral curve $\alpha_l[\xi](t) \subset P$ for $X(l)$ starting at ξ; this curve satisfies

$$\frac{d\alpha_l[\xi](t)}{dt} = X(l) \text{ and } \alpha_l[\xi](0) = \xi. \tag{4.21}$$

We abbreviate

$$f_l(x) := \varphi^l \circ \hat\sigma(x),\ \eta_l(\xi) := \mathbb{E}_{\mathbf{p}(\xi)}(\varphi^l \circ \hat\sigma), \tag{4.22}$$

and set

$$\psi_l[\xi](t) := \int_0^t \eta_l(\alpha_l[\xi](\tau))\, d\tau. \tag{4.23}$$

Using (4.21) and (4.22), the restriction of the equation (4.20) to the curve $\alpha_l[\xi](t)$ has the following form

$$\frac{d}{dt} \log p(x, \alpha_l[\xi](t)) = f_l(x) - \eta_l(\alpha_l[\xi](t)). \tag{4.24}$$

Using (4.23), (4.21), and regarding (4.24) as an ODE for the function $F(t) := \log p(x, \alpha_l[\xi](t))$ with the initial value $F(0) = \log p(x, \xi)$, we write the unique solution to (4.24) as

$$\log p(x, \alpha_l[\xi](t)) = f_l(x) \cdot t + \log p(x, \xi_0) - \psi_l[\xi](t). \tag{4.25}$$

We shall now express the function $\psi_l[\xi](t)$ in a different way. Using the equation

$$\int_\Omega p(x, \alpha_l[\xi](t)) d\mu = 1 \text{ for all } t$$

which by (4.25) is equivalent to

$$\int_\Omega \exp(f_l(x) \cdot t - \psi_l[\xi](t)) \cdot p(x, \xi) d\mu = 1,$$

we obtain

$$\psi_l[\xi](t) = \log \int_\Omega \exp(f_l(x) \cdot t) \cdot p(x, \xi)\, d\mu. \tag{4.26}$$

Now we define a map $\Phi : V' \times P \to P$ by

$$\Phi(l, \xi) = \alpha_l[\xi](1).$$

We compute

$$d_{(0,\xi)}\Phi(l, 0) = \frac{d}{dt}_{|t=0} \alpha_{tl}[\xi](1) = X(l)_{|\xi}. \tag{4.27}$$

The map Φ may not be well defined on the whole space $V' \times P$. For a given $\xi \in P$ we denote by $V'(\xi)$ the maximal subset in V' where the map $\Phi(l, \xi)$ is well-defined for $l \in V'(\xi)$. Since ξ is an interior point of P, $V'(\xi)$ is open.

Lemma 4.15. *Given a point $\xi \in P$, $l \in V'(\xi)$, assume that $l' \in V'(\Phi(l, \xi))$ and $l + l' \in V'(\xi)$. Then we have*

$$\Phi(l', \Phi(l, \xi)) = \Phi(l' + l, \xi). \tag{4.28}$$

Proof. Choose a small neighborhood $U(\xi) \subset P$ of ξ such that the restriction of $\mathbf{p}$ to $U(\xi)$ is an embedding. Now we choose $V'_\varepsilon \subset V'(\xi)$ such that the LHS and the RHS of (4.28) belong to $\mathbf{p}(U(\xi))$ for any $l, l' \in V'_\varepsilon$ such that $l + l' \in V'_\varepsilon$. Since $\mathbf{p} : U(\xi) \to \mathcal{M}(\Omega)$ is an embedding, to prove (4.28), it suffices to show that for all $x \in \Omega$ we have

$$\log p(x, \Phi(l', \Phi(l, \xi))) = \log p(x, \Phi(l' + l, \xi)). \tag{4.29}$$

Using (4.25) we have

$$\log p(x, \Phi(l', \Phi(l, \xi))) = \log p(x, \Phi(l, \xi)) + f_{l'}(x) + N(l', l, \xi), \tag{4.30}$$

where $N(l', l, \xi)$ is the normalizing factor such that p in the LHS of (4.30) is a probability density, cf. (4.26). (This factor is also called the cumulant generating function or the log-partition function.) Expanding the RHS of (4.30), using (4.25), we obtain

(4.31)

$$\log p(x, \Phi(l', \Phi(l, \xi))) = \log p(x, \xi) + f_l(x) + N(l, \xi) + f_{l'}(x) + N(l', l, \xi).$$

Since $f_l(x) = \langle l, \varphi \circ \hat{\sigma}(x) \rangle$ we obtain

$$f_l(x) + f_{l'}(x) = f_{l+l'}(x).$$

Using this, we deduce from (4.31)

$$\log p(x, \Phi(l', \Phi(l, \xi)) = \log p(x, \xi) + f_{l+l'}(x) + N_1(l, l', \xi), \tag{4.32}$$

where $N_1(l, l', \xi)$ is the log-partition function. By (4.25) the RHS of (4.32) coincides with the RHS of (4.29). This proves (4.29) and hence (4.28). This proves Lemma 4.15 for the case $l, l', l + l' \in V'_\varepsilon \subset V'(\xi)$.

Now assume that $l, l' \in V'(\varepsilon)$ but $l + l' \in V'(\xi) \setminus V'_\varepsilon$. Let c be the supremum of all positive numbers c' in the interval $[0, 1]$ such that (4.28) holds for $c' \cdot l, c' \cdot l'$. Since Φ is continuous, it follows that c is the maximum. We just proved that (4.28) holds for an open small neighborhood of any point ξ. Hence $c = 1$, i.e. (4.28) holds for any $l, l' \in V'(\varepsilon)$. Repeating this argument, we complete the proof of Lemma 4.15. □

Completion of the proof of Theorem 4.14.
Choosing a local isomorphism

$$\Phi_\xi : V' \to P,\ l \mapsto \Phi(l, \xi)$$

around a given point $\xi \in P$, using (4.22), we conclude from (4.25)

$$p(x, \Phi_\xi(l)) = p(x, \xi) \cdot \exp(l \circ \varphi \circ \hat{\sigma}(x)) \cdot \tilde{N}(l, \xi), \tag{4.33}$$

where $\tilde{N}(l, \xi)$ is the log-partition function. From (4.33), and remembering that the RHS of (4.20) does not vanish, we conclude that the map $\Phi_\xi : V'(\xi) \to P$ is injective. By (4.27), Φ_ξ is an immersion at ξ. Hence Φ_ξ defines affine coordinates on $V(\xi)$ where by (4.33) $\Phi_\xi(V'(\xi))$ is represented as an exponential family.

We shall show that Φ_ξ can be extended to Φ_D on some open subset $D \subset V'$ such that Φ_D provides global affine coordinates on $P = \Phi_D(D)$ in which P is represented as an exponential family.

Assume that $\xi' \in V'(\xi) \setminus \{0\}$. Denote by $V'(\xi') + \xi'$ the translation of the subset $V(\xi')$ in the affine space V' by the vector ξ'. Let $W := V'(\xi) \cup [V'(\xi') + \xi']$. We extend Φ_ξ to Φ_W on W via

$$\Phi_W(l) = \Phi_\xi(l) \text{ for } l \in V'(\xi), \tag{4.34}$$

$$\Phi_W(l' + \xi') = \Phi_{\xi'}(l') \text{ for } l' \in' (\xi'). \tag{4.35}$$

We claim that Φ_W is well defined. It suffices to show that if $l' + \xi' \in V'(\xi)$ then

$$\Phi_\xi(l' + \xi') = \Phi_{\xi'}(l'). \tag{4.36}$$

Clearly (4.36) is a consequence of Lemma 4.15. This shows that Φ_W provides a global affine coordinate system on $\Phi_W(W') \subset P$. Repeating this argument, we provide a global affine coordinate system on P provided by the map $\Phi_D : D \to P$.

Finally we need to show that P is an exponential family in the constructed affine coordinates, i.e. for any $\theta \in D$ we have

$$p(x, \Phi_W(\theta)) = p(x, \xi) \cdot \exp(\theta, \varphi \circ \hat{\sigma}(x)) \cdot \exp(N(\theta)) \tag{4.37}$$

where $N(\theta)$ is the log-partition function. Representing

$$\Phi_W(\theta) = \Phi(l_n, \Phi(l_{n_1}, \cdots, \Phi(l_1, \xi))$$

and using Lemma 4.15 and (4.33), we obtain immediately (4.37). This completes the proof of Theorem 4.14. □

4.4. **Asymptotic efficient estimators.** In what follows we shall consider the classical case, when $\varphi : M \to \mathbb{R}^n$ is the natural embedding of an open domain $M \subset \mathbb{R}^n$. Hence $\varphi \circ \hat{\sigma} = \hat{\sigma}$ and "φ" will be omitted.

So far we consider efficiency of an estimator $\hat{\sigma} : \Omega \to P$. In statistics as well as in machine learning, we are given a sequence of data $\{x_k \in (\Omega)^k\}$ and therefore we are interested in a sequence of estimators $\hat{\sigma}_n : (\Omega)^n \to P$ which is asymptotic efficient. On the product measure space $(\Omega^n, p_n\mu) = (\Omega, p\mu)^n$ the density function p_n is defined as follows. For $(x_1, \cdots, x_n) \in (\Omega)^n$ we have

$$\log p_n(x_1, \cdots, x_n) = \sum_{i=1}^{n} \log p(x_i).$$

Denote by $\mathfrak{g}_n$ the Fisher metric on a statistical model $(M, (\Omega)^n, \mu^n, p_n)$. We assume that p is a regular density estimation. Then p_n is also a regular density function. It follows that [14]

$$\mathfrak{g}_n(V, V) = n \cdot \mathfrak{g}_1(V, V). \tag{4.38}$$

This implies that the lower bound in the Cramér-Rao inequality for unbiased estimators $\hat{\sigma}_n : (\Omega)^n \to M$ tends to zero.

Thus we could get asymptotic efficient estimators σ_n if they are asymptotically unbiased and they asymptotically satisfy the Cramér-Rao inequality.' The imprecise notion of asymptotically unbiased estimators is replaced by the notion of *consistent* estimators. More precisely we assume that there exists a number $N_0 \in \mathbb{N}$ such that the following space

$$L^2_\infty(P, V) = \bigcap_{i=N_0}^{\infty} L^2_{\hat{\sigma}_i}(P, V)$$

is non-empty.

Definition 4.16. Let $\varphi \in L^2_\infty(P, V)$. A sequence of estimators $\{\hat{\sigma}_N : \Omega^N \to P |\, N = 1, 2 \cdots\}$ is called φ-*consistent*, if for all $\xi \in P$, for all $l \in V'$ and for all $\varepsilon > 0$ we have

$$\lim_{N\to\infty} \mathbf{p}(\xi)(\{x^N \in \Omega^N |\, |l \circ \tilde{b}_{\varphi|\hat{\sigma}_N}(x^N, \xi)| > \varepsilon\}) = 0 \tag{4.39}$$

where

$$\tilde{b}_{\varphi|\hat{\sigma}_N}(x, \xi) := \varphi(\hat{\sigma}_N(x)) - \varphi(\xi)$$

is called the *error function*.

We recommend the reader to [LC1998, Chapter 6] and [IH1981] for further discussions on theory of asymptotic efficient estimators.

[14] The equality (4.38) also holds without regularity assumption on the density function p, even without assumption of the existence of dominating measure

4.5. **Conclusion.** In this lecture we discussed further the relation between the existence of efficient estimators and the geometry of the underlying parameterized statistical model. We also discussed the notion of asymptotic efficiency of estimators which can be relaxed when we are given not only one estimator but a large sequence of estimators. The asymptotic efficiency means that in the limit the MSE of asymptotic efficient estimators goes to zero, and this is sufficient for learning if we have enough time. The later condition is related to the notion of complexity of computations and complexity of samples which will be discussed in the next lecture.

5. Sample complexity and PAC-learning

In the last two lectures we considered the general stochastic learning scenario where the correlation between an observed element $x_i \in \mathcal{X}$ and its observed label $y_i \in \mathcal{Y}$ is expressed by a probability measure μ_h on a measurable space $\Omega := \big(\mathcal{X} \times \mathcal{Y}, \Sigma_{(\mathcal{X}\times\mathcal{Y})}\big)$. We investigate general unparametrized statistical models P - a more common name for hypothesis classes in most general statistical consideration - as well as parameterized statistical models. Using analytic and geometric methods we derived the general Cramér-Rao inequality that gives a lower bound for the MSE of an unbiased estimator $\hat{\sigma} : \Omega \to P$. We also know that the bound in the Cramér-Rao inequality goes to zero when *the size n of a sample $x_n \in (\Omega)^n$* goes to infinity. Asymptotic theory of estimation investigates asymptotic efficient estimators in order to know if the MSE of asymptotic efficient estimators $\hat{\sigma}_n$ goes to zero when n goes to infinity. If it is the case, learning based on empirical data is possible modulo time resource.

In today lecture we consider a hypothesis class $\mathcal{F} \subset \mathcal{Y}^{\mathcal{X}}$ from the classical point of view, i.e. no analytic-geometric property of $\mathcal{F}$ is taken into account. Recall that the goal of the learner is to find a hypothesis h_S that minimizes a (generalized) risk $R_\mu(h)$ over the hypothesis class $\mathcal{F} \ni h$ for each given data $S \in (\mathcal{X} \times \mathcal{Y})^n$. The (generalized) risk R_μ is unknown to the learner since μ is unknown, and the ERM principle/rule suggests that the "best" hypothesis h_S must minimize the generalized empirical risk $\hat{R}_S(h)$. In order to make the ERM principle work, i.e., in order to ensure that h_S also minimizes the (true) risk R_μ we need further assumptions on a hypothesis class $\mathcal{F}$. For this purpose we introduce PAC-learning concepts in the statistical learning framework of machine learning. Using PAC-learning concepts, we analyze the convergence of the generalization error $R_\mu(h_S)$ of a minimizer h_S of empirical risk $\hat{R}_S$, when the size of the underlying sample S goes to infinity, taking into account the cost of the *computational representations and the time complexity of the learning algorithm.* In particular we want to address the following problem.

Problem 5.1. *How many random examples does a learning algorithm need to draw before it has sufficient information to learn unknown target hypothesis/concept from the concept class $\mathcal{F}$?*

The answer to Problem 5.1 led Vladimir Vapnik to his notion of *sample complexity* which distinguishes statistical learning theory from classical mathematical statistics.

The sample complexity of a machine learning algorithm represents the number of training-samples that it needs in order to *successfully learn* a target function. The notion of the sample complexity is inseparable from the notion of PAC-learnability of a hypothesis class.

Before going further let us introduce some short notations accepted in ML community.
• $P_{S\sim D^m}[f(S)]$ denotes the probability of a sample $S \in \Omega^m$ of m i.i.d. random variables distributed on Ω with probability measure D such that S satisfies relation $f(S)$,
• $\mathbb{E}_{S\sim D^m}(f_S)$ denotes the expectation of the function f_S on the measurable space where $S \in \Omega^m$ is a sequence of m i.i.d. random variables distributed on Ω with probability measure D.
• We write $P[f(S)]$, $\mathbb{E}(f_S)$ if $m = 1$ and D is given (and hence can be omitted).

Now we want to analyze the idea " successful learning from training data". Since we consider the training data to be random, we have to work in PAC-setting.

5.1. **PAC-learnable hypothesis class.** There are many variants/refinements of the concept of a PAC-learnable hypothesis class. Each variant depends on the parameter/specification of the learnability of a (class of) learning problem(s) we wish to take into account. Below we present two variants of the concept of PAC-learnability.

5.1.1. *Deterministic PAC-learning with true risk (and realizability assumption).* Problems is machine learning are classified under assumptions we pose on them. One of most frequently used assumption is the realizability assumption.

Definition 5.2. (cf. [SSBD2014, Definition 2.1, p. 38]) A hypothesis class $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ is said to satisfy *the realizability assumption (w.r.t. a risk function R^L)*, if for all probability distribution D on $\mathcal{X} \times \mathcal{Y}$ there exists an element $h_D \in \mathcal{H}$ such that $R^L_D(h_D) = 0$.

Recall that by (2.1) a learning algorithm A turns each data S into a hypothesis h_S, i.e., $h_S = A(S)$. Denote by $size(h)$ the maximal cost of the computational representation of $h \in \mathcal{H}$.

Definition 5.3. (cf. [MRT2012, Definition 2.3, p. 13] [15]) Assume that a hypothesis class $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ is deterministic, i.e., its elements are deterministic

[15] I omit one variable of the function $m_{\mathcal{H}}$ in Definition 2.3 of [MRT2012] since the omitted variable does not enter in the definition of $m_{\mathcal{H}}$.

(see Remark 2.5). We say that $\mathcal{H}$ is *PAC-learnable* if there exists an algorithm A and a polynomial function $m_{\mathcal{H}}\big(1/\varepsilon, 1/\delta, size(h_{true})\big)$ such that for any $\varepsilon > 0$ and $\delta > 0$, for all distributions D on $\mathcal{X}$ and for any true hypothesis $h_{true} \in \mathcal{H}$ the following holds for $m \geq m_{\mathcal{H}}\big(1/\varepsilon, 1/\delta, n, size(h_{true})\big)$:

$$P_{S\sim[(\Gamma_f)_* D]^m}[R_{(\Gamma_f)_* D}(h_S) \leq \varepsilon] \geq 1 - \delta. \tag{5.1}$$

If A further runs in $poly(1/\varepsilon, 1/\delta, size(h_{true}))$, then $\mathcal{H}$ is said to be *efficiently PAC-learnable*. When such an algorithm A exists, it is called a *PAC-learning algorithm for* $\mathcal{H}$.

Definition 5.4. *The sample complexity* $m_{\mathcal{H}} : (0,1)^2 \to \mathbb{N}$ *of a hypothesis class* $\mathcal{H}$ is the "minimal function" of the *accuracy* (ε) and *confidence* (δ) parameters that appears in the definition of PAC-learnability of $\mathcal{H}$, i.e., for each (ε, δ) the sample complexity $m_{\mathcal{H}}(\varepsilon, \delta)$ is the minimal number that appears in Definition 5.4.

Remark 5.5. (1) In the definition of PAC-learnability (Definition 5.3) one considers only the true risk. Clearly, any true hypothesis h_{true} satisfies the realizability assumption w.r.t. the true risk.

(2) In Definition 5.3 the sample complexity $m_{\mathcal{H}}$ has further one variable - the size of the true hypothesis h_{true} (recall that h_{true} is unknown so we have to guess the upper bound of $size(h_{true})$). Furthermore the function $m_{\mathcal{H}}$ is required to be polynomial in the variables specified above. This requirement is crucial in any practical application of machine learning, where we have limited computational resource. For further reading on computational complexity I recommend [SSBD2014, Chapter 8], [KV1994] and [Pudlak2013].

In many cases, in particular when the computational representation of the concepts is not explicitly discussed or is straightforward, we may omit the polynomial dependency on $size(h_{true})$ in the PAC-definition and focus only on the sample complexity $m_{\mathcal{H}}$ with two variables ε and δ.

5.1.2. *Agnostic PAC-learning with general loss function and without realizability assumption.* Now we shall define a PAC-learning concept for general case where the learning scenario is stochastic with a general risk of the following form

$$R_D^L(h) := \mathbb{E}_{z\sim D}(L(h, z)), \tag{5.2}$$

where $h \in \mathcal{H} \subset \mathcal{P}(\Omega)$, $z \in \Omega$ is a sample distributed by $D \in \mathcal{P}(\Omega)$ and

$$L : \mathcal{H} \times \Omega \to \mathbb{R}_+ \tag{5.3}$$

is a general loss function, which will be also called *a loss function*.

It is not hard to see that the generalized risk function defined in (2.8) can be written in the form (5.2). From now on we shall consider general risk functions of the form (5.2).

Definition 5.6. (cf. [SSBD2014, Definitions 3.3, 3.4, p. 46, 49]) Assume that a hypothesis class $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ is agnostic, i.e., its elements are stochastic.

We say $\mathcal{H}$ is *agnostic PAC-learnable* w.r.t. a loss function L (or a risk function R^L) if there exist a sample complexity function $m_{\mathcal{H}}(\varepsilon, \delta)$ and a learning algorithm A with the following property. For every $(\varepsilon, \delta) \in (0,1)^2$, for every distribution D over $\mathcal{X} \times \mathcal{Y}$, if $m \geq m_{\mathcal{H}}(\varepsilon, \delta)$ then

$$P_{S \sim D^m}[\big(R^L_D(h_S) - \inf_{h' \in \mathcal{H}} R^L_D(h')\big) \leq \varepsilon] \geq 1 - \delta. \tag{5.4}$$

Remark 5.7. (1) In Definition 5.6 we do not assume the realizability condition.

(2) In both Definitions 5.3 and 5.6 the PAC-framework is a distribution-free model: no particular assumption is made about the distribution D from which examples are drawn.

Remark 5.8 (Error decomposition)**.** Given a distribution D on $\mathcal{X} \times \mathcal{Y}$

$$R^L_{b,D} := \inf_h R^L_D(h)$$

be *the Bayes risk*, computed with the measure D and a loss function L, where the infimum is taken over all measurable functions $h : \mathcal{X} \to \mathcal{Y}$, and let

$$R^L_{\mathcal{H},D} := \inf_{h \in \mathcal{H}} R^L_D(h)$$

quantify the optimal performance of a learner capable of representing $\mathcal{H}$. Then we can decompose the difference between the risk of a hypothesis and the optimal Bayes risk as

$$R^L_D(h) - R^L_{b,D} = (R^L_D(h) - R^L_{\mathcal{H},D}) + (R^L_{\mathcal{H},D} - R^L_{b,D}). \tag{5.5}$$

The first term in the RHS of (5.5) is called *the estimation error* and the second term is called *the approximation error*.

The approximation error does neither depend on the hypothesis nor on the data. It quantifies how well the hypothesis class $\mathcal{F}$ is suited for the problem under consideration. The estimation error measures how well the hypothesis h performs relative to best hypotheses in $\mathcal{F}$.

In the PAC-bound (5.4) we measure the set of all samples S of size m whose estimation error is less than ε.

Exercise 5.9. Let $\mathcal{X} = \mathbb{R}^2$ and $\mathcal{Y} = \{0, 1\}$. We define the concept class C to be the set of all axis-aligned rectangles lying in $\mathbb{R}^2$. Each concept $c \in C$ defines a binary function $h_c : \mathcal{X} \to \mathcal{Y}$ as follows

$$h_c(x) = 1_c(x),$$

where 1_c is the indicate function of the axis-aligned rectangle c. The learning problem consists of determining with small error a target axis-aligned rectangle using the labeled training sample. Show that the concept class C is PAC-learnable.

5.2. **ERM, uniform convergence property and PAC-learnability.** In this subsection we shall find a sufficient condition, called *the uniform convergence property* (Definition (5.12), for the PAC-learnability of a hypothesis class provided with a general loss function L and with a learning algorithm A defined by the ERM rule.

Definition 5.10 (ε-representative sample)**.** ([SSBD2014, Definition 4.1]) A training set S is called *ε-representative* (w.r.t. domain Ω, hypothesis class $\mathcal{H}$, loss function L, and distribution D) if

$$\forall h \in \mathcal{H},\ |R_S^L(h) - R_D^L(h)| \leq \varepsilon.$$

The next simple lemma states that whenever the sample is ε-representative, an empirical risk minimizer, also called ERM predictor, approximates a general risk minimizer with ε-accuracy.

Lemma 5.11. *Assume that a training set S is $\varepsilon/2$-representative. Then any empirical risk minimizer h_S satisfies*

$$R_D^L(h_S) \leq \min_{h\in\mathcal{H}} R_D^L(h) + \varepsilon.$$

Proof. For every $h \in \mathcal{H}$ Definition 5.10 implies

$$R_D^L(h_S) \leq R_S^L(h_S) + \varepsilon/2 \leq R_S^L(h) + \varepsilon/2$$

$$\leq R_D^L(s) + \varepsilon.$$

where the second inequality holds since h_S is an ERM predictor. □

Lemma 5.11 implies that to ensure that the ERM rule is an agnostic PAC-learner for a hypothesis class $\mathcal{H}$, it suffices to show that with probability of at least $1-\delta$ a sample S is ε-representative (w.r.t. $\mathcal{H}$, Ω, L and *all* D).

We shall formulate the discussed sufficient condition as a uniform convergence property of a hypothesis class in the following definition.

Definition 5.12 (Uniform Convergence)**.** A hypothesis class $\mathcal{H}$ is said to have *the uniform convergence property* (w.r.t. a domain Ω and a loss function L), if there exists a function $m_{\mathcal{H}}^{UC} : (0,1)^2 \to \mathbb{N}$ such that for every $\varepsilon, \delta \in (0,1)$ and for every probability distribution D on Ω, if S is a sample of $m \geq m_{\mathcal{H}}^{UC}$ examples drawn i.i.d. according to D, then with probability at least $1-\delta$, S is ε-representative.

Using the definition of uniform convergence property we summarize our discussion on a sufficient condition for PAC-leaner as follows.

Corollary 5.13. *If a class $\mathcal{H}$ has the uniform convergence property with a function $m_{\mathcal{H}}^{UC}$ then the class is agnostic PAC-learnable with the sample complexity $m_{\mathcal{H}}(\varepsilon;\delta) \leq m_{\mathcal{H}}^{UC}(\varepsilon/2,\delta)$. Furthermore, in that case, an ERM-er is an agnostic PAC-learning algorithm for $\mathcal{H}$.*

5.3. **Finite classes are agnostic PAC-learnable.** Using the results obtained in the previous subsection we shall prove the following main theorem of today lecture.

Theorem 5.14. *Let Ω be a domain, $\mathcal{H}$ a finite hypothesis class, and $L : \mathcal{H} \times \Omega \to [0,1]$ a loss function. Then, $\mathcal{H}$ is agnostic PAC-learnable using the ERM algorithm with sample complexity*

$$m_{\mathcal{H}}(\varepsilon, \delta) \leq \frac{2\log\big(2\#(\mathcal{H})/\delta\big)}{\varepsilon^2}.$$

Proof. To prove Theorem 5.14 it suffices to find for each (ε, δ) a number $m = m_{\mathcal{H}}(\varepsilon, \delta)$ such that

$$D^m\big(\bigcap_{h\in\mathcal{H}} \{S : |R_S^L(h) - R_D^L(h)| \leq \varepsilon\}\big) \geq 1 - \delta. \tag{5.6}$$

Clearly (5.6) is a consequence of the following inequality

$$\sum_{h\in\mathcal{H}} D^m\big(\{S : |R_S^L(h) - R_D^L(h)| > \varepsilon\}\big) < \delta. \tag{5.7}$$

Since $\#H < \infty$, it suffices to find m such that each summand in RHS of (5.7) is small enough. For this purpose we shall apply the well-known Hoeffding Inequality [Hoeffding1963].

Lemma 5.15 (Hoeffding's Inequality)**.** *Let $\theta = (\theta_1, \cdots, \theta_n)$ be a sequence of i.i.d. random variables and assume that for all i $\mathbb{E}_{\theta_i\sim D}(\theta_i) = \mu$ and $P_{\theta_i\sim D}[a_i \leq \theta_i \leq b] = 1$. Then for any $\varepsilon > 0$ we have*

$$P_{\theta\sim D^m}\big[\big|\frac{1}{m}\sum_{i=1}^m \theta_i - \mu\big| > \varepsilon\big] \leq 2\exp\big(\frac{-2m\varepsilon^2}{(b-a)^2}\big).$$

To apply Hoeffding's inequality to the proof of Theorem 5.14 we observe that for each $h \in \mathcal{H}$ and for all $i \in [1, m]$ the random variables

$$\theta_i := L(h, z_i) \in [0, 1]$$

are i.i.d. Furthermore we have for any $h \in \mathcal{H}$

$$R_S^L(h) = \frac{1}{m}\sum_{i=1}^m \theta_i,$$

$$R_D^L(h) = \int_\Omega L(h, z)\, dD = \mu.$$

Hence the Hoeffding inequality implies for each $h \in \mathcal{H}$

$$D^m\big(\{S : |R_S^L(h) - R_D^L(h)| > \varepsilon\}\big) \leq 2\exp(-2m\varepsilon^2). \tag{5.8}$$

Now pluging

$$m \geq \frac{\log(2\#(\mathcal{H})/\delta)}{2\varepsilon^2}$$

in (5.8) we obtain (5.7). This completes the proof of Theorem 5.14. □

Exercise 5.16. Let $S, S' \in (\mathcal{X} \times \mathcal{Y})^n$ be n-tuples of i.i.d. random variables distributed by a probability measure D. Consider a loss function L whose image is contained in a finite interval of length $c > 0$. Using the Hoeffding inequality derive an upper bound for the probability

$$P_{(S,S')\sim D^{2n}}\left[\left|R_S(h) - R_{S'}(h)\right| > \varepsilon\right]$$

that depends on n, ε and c.

5.4. **Conclusion.** In this section we introduce the notion of sample complexity of a hypothesis class $\mathcal{H}$ which measures the success of learning a best hypothesis in $\mathcal{H}$. The sample complexity $m_{\mathcal{H}}$ is a function of accuracy ε and confidence δ and possibly of variables measuring other computational complexities of $\mathcal{H}$. We show that the ERM principle is successful for learning problem when $\mathcal{H}$ is finite. Thus the overfitting phenomenon happens only in infinite hypothesis classes. In the next lecture we shall study PAC-learnability of infinite hypothesis classes.

6. PAC-learning in infinite hypothesis classes

In this section under *PAC-learning* (resp. *agnostic PAC-learning*) we understand the deterministic learning scenario (resp. the stochastic learning scenario, i.e. a probability measure on the space $\mathcal{X} \times \mathcal{Y}$ of labeled pairs is not generated by a probability measure on the space $\mathcal{X}$ of instances and a true predictor $h : \mathcal{X} \to \mathcal{Y}$) with true risk, with PAC-sample complexity and without PAC-computational complexity.

In the previous lecture we learn that finite classes are agnostic PAC-learnable w.r.t. any general loss function (Theorem 5.14). We also know that some infinite hypothesis classes with the true risk are not PAC-learnable, if we use ERM principle, (Subsection 2.2), and some infinite classes are PAC-learnable (Exercise 5.9). In this lecture we shall analyze the PAC-learnability of infinite hypothesis classes. More concrete we want to address the following two questions/problems.

Problem 6.1. *Which infinite hypothesis classes are (agnostic) PAC-learnable?*

Problem 6.2. *Are there (agnostic) PAC-learnable classes which is not PAC-learnable using ERM principle?*

We begin analyzing infinite hypothesis classes with No-Free-Lunch Theorem, which implies that infinite hypothesis classes are not PAC-learnable (Corollary 6.6). Hence we need a prior information for the PAC-learnability of a hypothesis class. A prior information of a hypothesis class results in its capacity (complexity, expressive power, richness, or flexibility) which can be measured by the VC-dimension (Definition 6.9).

Using the notion of VC-dimension we shall solve Problem 6.1 for binary classification classes (Theorem 6.15). The solution of Problem 6.1 also provides an answer to Problem 6.2.

6.1. **No-Free-Lunch and PAC-learnability.** Rephrasing, No-free-lunch theorem is a mathematical statement of the wisdom that there is no universal learner that can succeed on all learning tasks (Exercise 6.5, Corollary 6.6).

• *Notations.* For a finite set $\mathcal{X}$ let us denote by $U_{\mathcal{X}}$ the uniform probability measure on $\mathcal{X}$, i.e.,

$$U_{\mathcal{X}}(A) = \frac{\#A}{\#\mathcal{X}} \text{ for any } A \subset \mathcal{X}.$$

Using the uniform probability measure, for $f, h : \mathcal{X} \to \mathcal{Y}$ we let

$$R_f(h) := P_{U_{\mathcal{X}}}[h(x) \neq f(x)] = \mathbb{E}_{U_{\mathcal{X}}}(1 - \delta^{h(x)}_{f(x)}).$$

denote the true risk of h defined by the true predictor f and $U_{\mathcal{X}}$ (see Example 2.6).

Theorem 6.3 (No-free-lunch)**.** *(*[Wolf2017, Theorem 1.5]*) Assume that $\mathcal{X}$, $\mathcal{Y}$ are finite sets and $\#\mathcal{X} \geq n$. Then for any learning algorithm $A : S \mapsto h_S$, $S \in (\mathcal{X} \times \mathcal{Y})^n$, we have*

$$\mathbb{E}_{f \sim U_{\mathcal{Y}^{\mathcal{X}}}} \Big(\mathbb{E}_{S \sim [(\Gamma_f)_* U_{\mathcal{X}}]^n} \big(R_f(h_S) \big) \Big) \geq (1 - \frac{1}{\#\mathcal{Y}})(1 - \frac{n}{\#\mathcal{X}}) \tag{6.1}$$

Remark 6.4. The average error probability of random guessing of the true predictor f in the above scenario is $1 - (\#\mathcal{Y})^{-1}$. Theorem 6.3 only leaves little room for improvement beyond this by the factor $1 - n \cdot (\#\mathcal{X})^{-1}$. This factor reflects the fact that the learner - a learning algorithm A - has already seen the training data, which is at most $n \cdot (\#\mathcal{X})^{-1}$ of all cases. Regarding the unseen cases, however, all learning algorithms A are the same on average and perform no better than the random guessing. This result also implies that there is no order among learners. Hence there is No Free Lunch.

Proof of Theorem 6.3. We set for $S \in (\mathcal{X} \times \mathcal{Y})^n$

$$Pr_i : (\mathcal{X} \times \mathcal{Y})^n \to \mathcal{X},\ (x_1, y_1), \cdots, (x_n, y_n) \mapsto x_i \in \mathcal{X},$$

$$\mathcal{X}_S := \bigcup_{i+1}^{n} Pr_i(S).$$

Let us compute and estimate the double integral in the LHS of (6.1) using the Fubini theorem:

$$\begin{aligned}
\mathbb{E}_{f\sim U_{\mathcal{Y}^{\mathcal{X}}}}\Big(\mathbb{E}_{S\sim[(\Gamma_f)_*U_{\mathcal{X}})]^n}\big(R_f(h_S)\big)\Big) &= \frac{1}{\#\mathcal{X}}\mathbb{E}_{f\sim U_{\mathcal{Y}^{\mathcal{X}}}}\Big(\mathbb{E}_{S\sim[(\Gamma_f)_*U_{\mathcal{X}})]^n}\big(\sum_{x\in\mathcal{X}}(1-\delta_{f(x)}^{h_S(x)})\big)\Big)\\
&\geq \frac{1}{\#\mathcal{X}}\mathbb{E}_{f\sim U_{\mathcal{Y}^{\mathcal{X}}}}\Big(\mathbb{E}_{S\sim[(\Gamma_f)_*U_{\mathcal{X}})]^n}\big(\sum_{x\notin\mathcal{X}_S}(1-\delta_{f(x)}^{h_S(x)})\big)\Big)\\
&= \frac{1}{\#\mathcal{X}}\mathbb{E}_{S\sim[(\Gamma_f)_*U_{\mathcal{X}})]^n}\Big(\sum_{x\notin\mathcal{X}_S}\mathbb{E}_{f\sim U_{\mathcal{Y}^{\mathcal{X}}}}(1-\delta_{f(x)}^{h_S(x)})\Big)\\
&= \frac{1}{\#\mathcal{X}}\mathbb{E}_{S\sim[(\Gamma_f)_*U_{\mathcal{X}})]^n}\Big(\#[\mathcal{X}\setminus\mathcal{X}_S]\cdot\big(1-\frac{1}{\#\mathcal{Y}}\big)\Big)\\
&\overset{\text{since } \#[\mathcal{X}\setminus\mathcal{X}_S]\geq\#\mathcal{X}-n}{\geq} (1-\frac{1}{\#\mathcal{Y}})(1-\frac{n}{\#\mathcal{X}}). \qquad (6.2)
\end{aligned}$$

This completes the proof of Theorem 6.3. □

The following Exercise states that for every learner, there exists a task on which it fails.

Exercise 6.5. Derive from Theorem 6.3 the following version of No-Free-Lunch Theorem ([SSBD2014, Theorem 5.1, p. 61]). Let A be any learning algorithm for the task of binary classification with respect to the true loss and risk. Let m be any number smaller than $(1/2)\#\mathcal{X}$, representing a training set size. Then there exist a distribution D over $\mathcal{X}$ and $f\in\{0,1\}^{\mathcal{X}}$ such that

$$P_{S\sim[(\Gamma_f)_*D]^m}[R_{(\Gamma_f)_*D}\big(A(S)\big)\geq 1/8]\geq 1/7.$$

The No-Free-Lunch Theorem implies the non-existence of a universal learner in PAC-learning, which is expressed in the following.

Corollary 6.6. *Let $\mathcal{X}$ be an infinite domain set. Then the hypothesis class $\mathcal{H}:=\{0,1\}^{\mathcal{X}}$ is not PAC-learnable.*

Proof. Assume the opposite. By Definition of PAC-learnability 5.3 for any (ε,δ) and any $m>m_{\mathcal{H}}(\varepsilon,\delta)$ and any $h_{true}\in\mathcal{H}$ we have for any distribution D over $\mathcal{X}$ such that

$$P_{S\sim[(\Gamma_f)_*D]^m}[R_{(\Gamma_f)_*D}\big(A(S)\big)\geq\varepsilon]\leq\delta. \qquad (6.3)$$

Let us choose D to be a distribution concentrated over a subset $C\subset\mathcal{X}$ such that $\#C\geq 2m$. Then (6.3) contradicts to the assertion of Exercise 6.5. □

6.2. **The VC-dimension and PAC-learnability.** We shall show in the remainder of this lecture that the VC-dimension [16] of a hypothesis class gives the correct characterization of its PAC-learnability.

To motivate the definition of the VC-dimension, let us recall the proof of Corollary 6.6. There, we have shown that if there is a subset $C\subset\mathcal{X}$ of size

[16] for Vapnik-Chervonenkis dimension

$2m$ and the restriction of $\mathcal{H}$ to C is the full set of functions in $\{0,1\}^C$ then the PAC-bound (5.1) does not hold for sample size m.

Definition 6.7 (Shattering)**.** A hypothesis class $\mathcal{H} \subset \{0,1\}^{\mathcal{X}}$ *shatters* a finite subset $C \subset \mathcal{X}$ if $\#\mathcal{H}|_C = 2^{\#C}$.

Remark 6.8. If a hypothesis $\mathcal{H} \subset \{0,1\}^{\mathcal{X}}$ shatters some set C of size $2m$ then we cannot learn $\mathcal{H}$ using m examples, by Exercise 6.5.

Definition 6.9 (VC-dimension)**.** *The VC-dimension* of a hypothesis class $\mathcal{H} \subset \{0,1\}^{\mathcal{X}}$, denoted by $VC\dim(H)$, is the maximal size of a set $C \subset \mathcal{X}$ that can be shattered by $\mathcal{H}$. If $\mathcal{H}$ can shatter sets of arbitrarily large size we say that $\mathcal{H}$ has infinite VC-dimension.

Example 6.10. Assume that C is the concept class of all axis-aligned rectangle lying in $\mathbb{R}^2$ (see Exercise 5.9). Since for an arbitrary subset S_1 of a set S of 4 points on $\mathbb{R}^2$ we can always find a rectangle in C that separates S_1 from $S \setminus S_1$ the VC-dimension of C is at least 4. On the other hand, it is not hard to see that for any set S_5 of 5 distinct points on $\mathbb{R}^2$ there is a partition of S_5 into two subsets S_5^+ and S_5^- such that they cannot be separated by any rectangle $c \in C$. Hence the VC-dimension of C is 4.

Exercise 6.11 (VC-Threshold functions)**.** Consider the set of all threshold functions $\mathcal{F} \subset \{-1,1\}^{\mathbb{R}}$ defined by

$$\mathcal{F} := \{x \mapsto sgn(x-b)\}_{b\in\mathbb{R}}.$$

Show that $VC\dim(\mathcal{F}) = 1$.

The finiteness of the VC-dimension of a hypothesis class $\mathcal{H}$ is a necessary condition for the PAC-learnability of $\mathcal{H}$. We shall show that the finiteness of the VC-dimension of $\mathcal{H}$ is also a sufficient condition for its PAC-learnability, since this condition ensures the polynomial asymptotic behavior of the growth function of $\mathcal{H}$, which we shall define below.

Definition 6.12 (Growth function)**.** Let $\mathcal{F} \subset \mathcal{Y}^{\mathcal{X}}$ be a class of functions with finite target space $\mathcal{Y}$. *The growth function* $\Gamma_{\mathcal{F}}$ assigned to $\mathcal{F}$ is then defined for all $n \in \mathbb{N}$ as

$$\Gamma_{\mathcal{F}}(n) := \max_{\Sigma\subset\mathcal{X}|\,\#\Sigma=n} \#\mathcal{F}|_\Sigma.$$

We also set $\Gamma(0) = 1$.

The idea of the growth function is ubiquitous in mathematics. Whenever we have to deal with an infinite set (resp. an infinite dimensional space), it is useful to introduce a finite (resp. finite dimensional) filtration on the set (resp. space) and to investigate the property in consideration on each level of the filtration (the growth function describes the maximal size of $\mathcal{F}$ when restricted to a domain of n points). As we shall see later, the notion of growth function is the most important concept in proving various PAC-bounds, which also leads to VC-dimension and Rademacher complexity.

Example 6.13. Consider the set of all threshold functions $\mathcal{F} \subset \{1,1\}^{\mathbb{R}}$ defined by

$$\mathcal{F} := \{x \mapsto sgn(x-b)\}_{b\in\mathbb{R}}.$$

Given a set of distinct points $\{x_1, \cdots, x_n\} = \Sigma \subset \mathbb{R}$, there are $n+1$ functions in $\mathcal{F}|_\Sigma$ corresponding to $n+1$ possible ways of placing b relative to the x_i s. Hence, in this case $\Gamma(n) = n+1$.

The following Lemma, which has been first proved by Vapnik-Chervonenkis but known as Sauer's Lemma in learning machine community [Vapnik2006, p. 427], [MRT2012, p. 35], relates the notion of VC-dimension with the growth function.

Lemma 6.14. *Let $\mathcal{H} \subset \{0,1\}^{\mathcal{X}}$ be a hypothesis class with $VC\dim(H) = d < \infty$. Then, for all $n \in \mathbb{N}$ we have*

$$\Gamma_{\mathcal{H}}(n) \leq \sum_{i=0}^{d} \binom{n}{i}.$$

In particular, if $n > d+1$ then $\Gamma_{\mathcal{H}}(n) \leq (en/d)^d$.

Proof. To prove the first inequality in Lemma 6.14, it suffices to that for every subset $C \subset \mathcal{X}$ with $\#C = n$ we have

$$\#\mathcal{H}|_C \leq \#\{B \subset C | \, \mathcal{H}|_B = \{0,1\}^B\}. \tag{6.4}$$

We shall prove (6.4) by induction on the size $\#C$. Clearly (6.4) holds if $\#C = 1$. Now assume that the induction hypothesis holds for any $A \subset \mathcal{X}$ with $\#A = n-1$. Let $C \subset \mathcal{X}$ with $\#C = n$. Pick an element $c \in C$. We set

$$\mathcal{H}' : \{h \in \mathcal{H}|_C \,|\, \exists g \in \mathcal{H}|_C : h(c) \neq g(c) \text{ and } h|_{\{C\setminus c\}} = g|_{\{C\setminus c\}}\},$$

$$\mathcal{H}_c := \mathcal{H}'|_{\{C\setminus c\}}.$$

Then

$$\#\mathcal{H}|_C = \#\mathcal{H}|_{\{C\setminus c\}} + \#\mathcal{H}_c. \tag{6.5}$$

By induction hypothesis we have

$$\#\mathcal{H}|_{\{C\setminus c\}} \leq \#\{B \subset C \,|\, \mathcal{H}|_B = \{0,1\}^B \text{ and } c \notin B\}. \tag{6.6}$$

Now let us estimate the second term in RHS of ((6.5)

$$\begin{aligned}
\#\mathcal{H}_c = \#\mathcal{H}'|_{\{C\setminus c\}} &\leq \#\{B \subset \{C \setminus c\} \,|\, \mathcal{H}'|_B = \{0,1\}^B\} \\
&= \#\{B \subset \{C \setminus c\} \,|\, \mathcal{H}'|_{\{B\cup c\}} = \{0,1\}^{\{B\cup c\}}\} \\
&= \#\{B \subset C \,|\, \mathcal{H}'|_B = \{0,1\}^B \text{ and } c \in B\} \\
&\leq \#\{B \subset C \,|\, \mathcal{H}|_B = \{0,1\}^B \text{ and } c \in B\}. \qquad (6.7)
\end{aligned}$$

Adding (6.6) with (6.7) we get (6.4) for C.

Now let us prove the second inequality in Lemma 6.14. We have

$$\sum_{i=1}^{d}\binom{n}{i} \leq \sum_{i=1}^{d}\binom{n}{i}(\frac{n}{d})^{d-i}$$
$$= (\frac{n}{d})^d(1+\frac{d}{n})^n \overset{\text{since } (1+x)\leq e^x}{\leq} (\frac{n}{d})^d e^d.$$

This completes the proof of Lemma 6.14. □

6.3. **Fundamental theorem of binary classification.** In this Subsection we prove the simplest variance of Fundamental theorem of binary classification (Theorem 6.15) and discuss various quantitative variances of it in Remark 6.17.

Theorem 6.15 (Fundamental theorem of binary classification)**.** *Let $\mathcal{H} \subset \{0,1\}^{\mathcal{X}}$ be a hypotheses class with true risk. Then the following are equivalent:*

1. *$\mathcal{H}$ has the uniform convergence property.*
2. *Any ERM rule is a successful agnostic PAC-learner for $\mathcal{H}$.*
3. *$VC\dim(\mathcal{H}) < \infty$.*

Proof. 1. Note that $(1 \Longrightarrow 2)$ is a part of Corollary 5.13.

2. We note that the implication $(2 \Longrightarrow 3)$ follows from the No-Free-Lunch Theorem. Indeed, if the VC-dimension of $\mathcal{H}$ is infinite we cannot learn $\mathcal{H}$ by Remark 6.8.

3. The proof of $(3 \Longrightarrow 1)$ is based on a PAC bound via growth function (Lemma 6.16), which is one of essential assertions in in PAC-learning theory.

Lemma 6.16. *(*[SSBD2014, Lemma 6.11, p. 75]*) For every $D \in \mathcal{P}(\mathcal{X}\times\mathcal{Y})$ and every $\delta \in (0,1)$ we have*

$$P_{S\sim D^m}[|R_D(h) - R_S(h)| \leq \frac{4+\sqrt{\log\big(\Gamma_{\mathcal{H}}(2m)\big)}}{\delta\sqrt{2m}}] \geq 1-\delta. \tag{6.8}$$

Lemma 6.16 says that if the growth function $\Gamma_{\mathcal{H}}(m)$ is polynomial, then $\mathcal{H}$ enjoys the uniform convergence property. What is really important is the asymptotic behavior of the upper bound for $|R_D^L(h) - R_S^L(h)|$. The best available upper bound for $|R_D^L(h) - R_S^L(h)|$ is in [Wolf2017, Theorem 1.6], where Wolf considered arbitrary bounded loss function.

Proof of Lemma 6.16. To prove Lemma 6.16 first we use the Markov inequality that gives a lower bound of the LHS of (6.8) as follows

$$P_{S\sim D^m}[|R_D(h) - R_S(h)| \geq a] \leq \frac{\mathbb{E}_{S\sim D^m}\big(|R_D(h) - R_S(h)|\big)}{a} \tag{6.9}$$

for any positive number $a \in \mathbb{R}_+$. Using (6.9), we reduce the proof of Lemma 6.16 to a proof of the following inequality

$$\mathbb{E}_{S\sim D^m}\big(\sup_{h\in\mathcal{H}}|R_D(h)-R_S(h)|\big) \le \frac{4+\sqrt{\log\big(\Gamma_{\mathcal{H}}(2m)\big)}}{\sqrt{2m}}. \tag{6.10}$$

Using the following relation between the average empirical risk and the true risk for any m

(6.11)

$$\mathbb{E}_{S'\sim D^m}R_{S'}(h) = \frac{1}{m}\sum_{i=1}^{m}\mathbb{E}_{S\sim D^m}\big(1_{[h(x_i)\neq y_i)]}\big) = \mathbb{E}_{S\sim D}\big(1_{[h(x)\neq y]}\big) = R_D(h)$$

we rewrite the LHS of (6.10) as follows

$$\begin{aligned}
\text{LHS of (6.10)} &= \mathbb{E}_{S\sim D^m}\Big(\sup_{h\in\mathcal{H}}|\mathbb{E}_{S'\sim D^m}\big(R_{S'}(h)\big)-R_S(h)|\Big).\\
&\le \mathbb{E}_{S\sim D^m}\Big(\sup_{h\in\mathcal{H}}\mathbb{E}_{S'\sim D^m}\big(|R_{S'}(h)-R_S(h)|\big)\Big)\\
&\le \mathbb{E}_{S\sim D^m}\big(\mathbb{E}_{S'\sim D^m}\sup_{h\in\mathcal{H}}|R_{S'}(h)-R_S(h)|\big)\\
&= \mathbb{E}_{S,S'\sim D^m}\Big(\sup_{h\in\mathcal{H}}|\sum_{i=1}^{m}\frac{1}{m}\big(L_{true}(h,z_i')-L_{true}(h,z_i)\big)|\Big)\\
&= \mathbb{E}_{S,S'\sim D^m}\Big(\sup_{h\in\mathcal{H}}|\sum_{i=1}^{m}\frac{1}{m}\sigma_i\big(L_{true}(h,z_i')-L_{true}(h,z_i)\big)|\Big) \qquad (6.12)
\end{aligned}$$

for any $\sigma_i \in \mathbb{Z}_2$. Hence, setting $\sigma := (\sigma_1, \cdots, \sigma_m) \in \mathbb{Z}_2^m$, we obtain

$$\begin{aligned}
\text{RHS of (6.12)} &= \mathbb{E}_{\sigma\sim U_{\mathbb{Z}_2}^m}\mathbb{E}_{S,S'\sim D^m}\Big(\sup_{h\in\mathcal{H}}\frac{1}{m}|\sum_{i=1}^{m}\sigma_i\big(L_{true}(h,z_i')-L_{true}(h,z_i)\big)|\Big)\\
&(6.13)= \mathbb{E}_{S,S'\sim D^m}\mathbb{E}_{\sigma\sim U_{\mathbb{Z}_2}^m}\Big(\max_{h\in\mathcal{H}|_{\mathcal{X}_{(S\cup S')}}}\frac{1}{m}|\sigma_i\big(L_{true}(h,z_i')-L_{true}(h,z_i)\big)|\Big).
\end{aligned}$$

Fixing training data S, S' of size m and $h \in \mathcal{H}|_{\mathcal{X}_{(S\cup S')}}$, regarding $\sigma \in \mathbb{Z}_2^m$ as a variable, we set

$$\theta_{h,S,S'}(\sigma) := \frac{1}{m}\sum_{i+1}^{m}\sigma_i\big(L_{true}(h,z_i')-L_{true}(h,z_i)\big).$$

Noting that

$$\mathbb{E}_{\sigma\sim U_{\mathbb{Z}_2}^m}\big(\theta_{h,S,S'}(\sigma)\big)=0 \text{ and } \theta_{h,S,S'}(\sigma)\in[-1,1],$$

the Hoeffding inequality yields for every $\rho > 0$

$$P_{\sigma\sim U_{\mathbb{Z}_2}^m}[\max_{h\in\mathcal{H}|_{\mathcal{X}_{(S\cup S')}}}|\theta_{h,S,S'}(\sigma)|>\rho] \le 2\exp(-2m\rho^2). \tag{6.14}$$

Letting h run on the finite set $\mathcal{H}|_{\mathcal{X}_{(S\cup S')}}$, we obtain from (6.14)

$$P_{\sigma\sim U^m_{\mathbb{Z}_2}}[\max_{h\in\mathcal{H}|_{\mathcal{X}_{(S\cup S')}}} |\theta_{h,S,S'}(\sigma)| > \rho] \leq 2\#(\mathcal{H}|_{\mathcal{X}_{(S\cup S')}})\cdot\exp(-2m\rho^2). \tag{6.15}$$

To complete the proof of (6.10), using (6.12) and (6.12), it suffices to show that for any $S, S' \in (\mathcal{X}\times\mathcal{Y})^m$ we have

$$\mathbb{E}_{\sigma\sim U^m_{\mathbb{Z}_2}}\big(\max_{h\in\mathcal{H}|_{\mathcal{X}_{(S\cup S')}}} |\theta_{h,S,S'}(\sigma)|\big) \leq \frac{4+\sqrt{\log\big(\Gamma_{\mathcal{H}}(2m)\big)}}{\sqrt{2m}}. \tag{6.16}$$

Now we are going to derive (6.16) from (6.15). For $i\in\mathbb{N}$ we set

$$\rho_i = \frac{i+\sqrt{\log\#(\mathcal{H}|_{\mathcal{X}_{(S\cup S')}})}}{\sqrt{2m}}.$$

Since t_i is monotone increasing, abbreviating

$$\Theta_{S,S'}(\sigma) := \max_{h\in\mathcal{H}|_{\mathcal{X}_{(S\cup S')}}} |\theta_{h,S,S'}(\sigma)|,$$

we have

$$\begin{aligned}
\mathbb{E}_{\sigma\sim U^m_{\mathbb{Z}_2}}\big(\Theta_{S,S'}(\sigma)\big) &\leq \frac{\sqrt{\log\#\big(\mathcal{H}|_{\mathcal{X}_{(S\cup S')}}\big)}}{\sqrt{2m}} + \sum_{i=1}^{\infty} t_i P_{\sigma\sim U^m_{\mathbb{Z}_2}}[\Theta_{S,S'}(\sigma) > t_{i-1}] \\
&\overset{(6.15)}{\leq}\Longrightarrow \frac{\sqrt{\log\Gamma_{\mathcal{H}}(2m)}}{\sqrt{2m}} + 2\frac{\Gamma_{\mathcal{H}}(2m)}{\sqrt{2m}}\sum_{i=1}^{\infty}\frac{i+\log\Gamma_{\mathcal{H}}(2m)}{\exp\big((i-1+\sqrt{\log\Gamma_{\mathcal{H}}(2m)})^2\big)} \\
&\leq \frac{\sqrt{\log\Gamma_{\mathcal{H}}(2m)}}{\sqrt{2m}} + 2\frac{\Gamma_{\mathcal{H}}(2m)}{\sqrt{2m}}\int_{1+\sqrt{\log\Gamma_{\mathcal{H}}(2m)}}^{\infty} xe^{-(x-1)^2}\,dx \\
&= \frac{\sqrt{\log\Gamma_{\mathcal{H}}(2m)}}{\sqrt{2m}} + 2\frac{\Gamma_{\mathcal{H}}(2m)}{\sqrt{2m}}\int_{\sqrt{\log\Gamma_{\mathcal{H}}(2m)}}^{\infty}(y+1)e^{-y^2}\,dy \\
&\leq \frac{\sqrt{\log\Gamma_{\mathcal{H}}(2m)}}{\sqrt{2m}} + 4\frac{\Gamma_{\mathcal{H}}(2m)}{\sqrt{2m}}\int_{\sqrt{\log\Gamma_{\mathcal{H}}(2m)}}^{\infty} ye^{-y^2}\,dy \\
&= \frac{\sqrt{\log\Gamma_{\mathcal{H}}(2m)}}{\sqrt{2m}} + 2\frac{\Gamma_{\mathcal{H}}(2m)}{\sqrt{2m}}[-e^{-y^2}]_{\log\Gamma_{\mathcal{H}}(2m)}^{\infty} \\
&= \frac{\sqrt{\log\Gamma_{\mathcal{H}}(2m)}}{\sqrt{2m}} + \frac{2}{\sqrt{2m}}. \qquad (6.17)
\end{aligned}$$

Clearly (6.17) implies (6.16). This completes the proof of Lemma 6.16. □

Completion of the proof of Theorem 6.15: (3 $\Longrightarrow$ 1). Assume that $VC\dim(\mathcal{H}) = d < \infty$. By Lemma 6.14

$$\Gamma_{\mathcal{H}}(2m) \leq (\frac{2em}{d})^d.$$

Using this, Lemma 6.16 implies the following *PAC-bound via VC-dimension* [SSBD2014, p. 75] [17]

$$P_{S\sim D^m}[|R_S(h) - R_D(h)| \leq \frac{4 + \sqrt{d\log(2em/d)}}{\delta\sqrt{2m}}] \geq 1 - \delta. \tag{6.18}$$

Since m growth much quicker than $\log m$, it is not hard to see that, for any given $\varepsilon, \delta \in (0,1)$ there exists $m_{\mathcal{H}}(\varepsilon,\delta) \in \mathbb{R}$ such that

$$m \geq \frac{2d\log m}{(\delta\cdot\varepsilon)^2} + \frac{2d\log(2e/d)}{(\delta\cdot\varepsilon)^2} \text{ if } m \geq m_{\mathcal{H}}(\varepsilon,\delta). \tag{6.19}$$

Once (6.19) holds, the value $|R_S(h) - R_D(h)|$ in (6.18) is bounded by ε, and the uniform convergence holds. This completes the proof of theorem 6.15. □

Remark 6.17. (1) A starting observation in the proof of the fundamental theorem for binary classifications is the relation between the true risk and the average empirical risk (6.11).

(2) Important techniques for proving PAC-bounds in the fundamental theorem for binary classifications are measure concentration inequalities as the Hoeffding inequality, the Markov inequality, which together with other tricks relates required bounds in probability with required bounds in expectations.

(3) There are many advanced variances of the fundamental theorem for binary classification, see e.g. [SSBD2014, Theorem 6.8, p. 72]. An easy improvement of Theorem 6.15 is to show that the function $m_{\mathcal{H}}(\varepsilon,\delta)$ can be chosen to be polynomial in $(1/\varepsilon, 1/\delta)$ (cf. [Wolf2017, Theorem 1.1, p. 24]).

6.4. **Conclusion.** In this lecture we learn that a correct characterization of the PAC-learnability of a hypothesis class $\mathcal{H}$ is the VC-dimension $VC\dim(\mathcal{H})$, whose motivation is the proof of the No-Free-Lunch theorem, which implies that the finiteness of the VC-dimension is a necessary condition for the PAC-learnability. It turns out that the finiteness of the VC-dimension is also a sufficient condition for the PAC-learnability of $\mathcal{H}$. To prove this assertion Vapnik and Chervonenkis invented the notion of the growth function which measures the complexity of $\mathcal{H}$ better than $VC\dim\mathcal{H}$, when $\#\mathcal{H} = \infty$. The main property that makes the growth function interesting is that it can be either polynomial or exponential - nothing in-between ($\Gamma_{\mathcal{H}}(n) = 2^n$ for all $n \leq VC\dim(\mathcal{H})$ holds by definition of the VC-dimension). The polynomial growth of $\Gamma_{\mathcal{H}}(n)$ plays the main role in the proof of PAC-learnability of a hypothesis class $\mathcal{H}$ with finite VC-dimension (Lemma 6.16).

[17] there are many versions of PAC-bounds via VC-dimension, see. e.g. [MRT2012, (3.31), p. 48], what important is the asymptotic behavior of the upper bound for $|R_D(h) - R_S(h)$ in (6.18), see also [Wolf2017, (1.25), (1.36)]

7. Basic methods in PAC-learning

In this lecture we shall revisit and refine central concepts of PAC-learning: sample complexity, PAC-bound, ERM and introduce the method of adaptive boost in order to solve more complicated problems in machine learning. We end the lecture with discuss of methods for model selection.

7.1. **Distribution dependent PAC-bounds and Rademacher complexity.** In the previous two lectures we consider PAC-bounds which are distribution independent. We obtained distribution free PAC-bounds via the growth function in inequality (6.8) and via the VC-dimension in Lemma 6.16 (Inequality (6.18)). Note that the VC-dimension and the growth function are distribution free concepts. The VC-dimension is a property of a hypothesis class in binary classification problem and the growth function is a feature of a hypothesis class in classification problem with finite label set. We cannot extend non-trivially these concept to regression problem.

Our claim that PAC-bounds in previous lectures are distribution free is only conditionally correct. The PAC-bound in Definition 5.3 is only distribution free inside the deterministic class, i.e., we require that the distribution entering in the PAC bound (5.1) is of the form $(\Gamma_f)_* D$, where $D \in \mathcal{P}(\mathcal{X})$. In general, as we shall see, a PAC-bound need not be distribution free. A class of successful distribution dependent PAC-bounds are obtained via a distribution dependent sample complexity - the Rademacher complexity. The approach based on Rademacher complexity offers in certain cases a better bound than a distribution free approach.

Consider a set of real-valued functions $\mathcal{G} \subset \mathbb{R}^{\mathcal{Z}}$ and a vector $z = (z_1, \cdots, z_n) \in \mathcal{Z}^n$. For $\sigma = (\sigma_1, \cdots, \sigma_n) \in \{-1, 1\}^n$ and $g \in \mathcal{G}$ we set

$$\langle \sigma, g^{\bullet n}(z) \rangle := \sum_{i=1}^{n} \sigma_i g(z_i).$$

Definition 7.1 (Rademacher complexity)**.** *The empirical Rademacher complexity* of $\mathcal{G}$ w.r.t. $z \in \mathcal{Z}^n$ is defined as

$$\mathcal{R}_z(\mathcal{G}) := \mathbb{E}_{\sigma \sim U_{\mathbb{Z}_2}^n} [\sup_{g \in \mathcal{G}} \frac{1}{n} \langle \sigma, g^{\bullet n}(z) \rangle].$$

If the z_i's are i.i. distributed according to a probability measure P on $\mathcal{Z}$, then *the Rademacher complexity* of $\mathcal{G}$ w.r.t. P are given by

$$\mathcal{R}_{n,P}(\mathcal{G}) := \mathbb{E}_{z \sim P^n} [\mathcal{R}_z(\mathcal{G})].$$

Remark 7.2. The uniformly distributed $\sigma_i' s$ are called Rademacher variables. We met them and a Rademacher complexity type quantity in the proof of Lemma 6.16.

The following Lemma implies that the Rademacher complexity can be estimated reliably from the data z and that no additional knowledge about P is required.

Lemma 7.3 (Rademacher vs. empirical Rademacher complexity)**.** *Let $\mathcal{G} \subset [a,b]^{\mathcal{Z}}$ be a set of real-valued functions. Then for every $\varepsilon > 0$ and any product measure P^n on $\mathcal{Z}^n$ it holds that*

$$P_{z\sim P^n}[\big(\mathcal{R}_{n,P}(\mathcal{G}) - \mathcal{R}_z(\mathcal{G})\big) \geq \varepsilon] \leq \exp\big(-\frac{2n\varepsilon^2}{(b-a)^2}\big). \tag{7.1}$$

We refer the reader to [Wolf2017, p. 27] for the proof of Lemma 7.3.

Using Lemma 7.3 and McDiarmid's inequality, which is a refinement of Hoeffding inequality we obtain the following distribution dependent PAC-bound.

Theorem 7.4 (PAC-bound via Rademacher complexities)**.** *(*[Wolf2017, Theorem 1.12]*) Consider arbitrary spaces $\mathcal{X}, \mathcal{Y}$, a hypotheses class $\mathcal{F} \subset \mathcal{Y}^{\mathcal{X}}$, a loss function $L : \mathcal{Y} \times \mathcal{Y} \to [0,c]$ and define*

$$\mathcal{Z} := \mathcal{X} \times \mathcal{Y} \text{ and } \mathcal{G}_{\mathcal{F}}^L := \{(x,y) \mapsto L(y,h(x)) |\, h \in \mathcal{F}\} \subset [0,c]^{\mathcal{Z}}.$$

For any $\delta > 0$ and any probability measure P on $\mathcal{Z}$ we have

$$P_{S\sim P^n}[R_P^L(h) - R_S^L(h) \leq \mathcal{R}_{n,P}(\mathcal{G}_{\mathcal{F}}^L) + c\sqrt{\frac{\log\frac{1}{\delta}}{2n}}] \geq 1-\delta, \tag{7.2}$$

$$P_{S\sim P^n}[R_P^L(h) - R_S^L(h) \leq \mathcal{R}_S(\mathcal{G}_{\mathcal{F}}^L) + 3c\sqrt{\frac{\log\frac{2}{\delta}}{2n}}] \geq 1-\delta. \tag{7.3}$$

We refer the reader to [Wolf2017] for the proof of Theorem 7.4.

Remark 7.5. 1) The Rademacher complexity approach allows us to go beyond binary classification and treat more general function classes that appear in classification or regression problems on an equal footing.

2) The Rademacher complexity can be upper bounded by the VC-dimension [Wolf2017, Corollary 1.18] and hence upper bounds by the Rademacher complexity are stronger than upper bounds by VC-dimension.

3) PAC-type bounds as in Exercise 6.5 are important for proving the non-existence of PAC-learning. There are refinements of the PAC-type bound in Exercise 6.5 or proving a lower bound of sample complexity (if the sample complexity is infinite, then the hypothesis class is not PAC-learnable), see e.g. [Wolf2017, Theorem 3.7, p. 51].

4) Once we introduce more structure on hypothesis classes $\mathcal{H}$ we could consider more related complexities, e.g. the packing number, etc. and use them for proving new PAC-bounds, see e.g. [Wolf2017].

7.2. **Algorithm dependent PAC-bounds and algorithmic stability.** The PAC-bounds in previous lectures ignore the specific algorithm used, that is, they hold for any algorithm using $\mathcal{H}$ as a hypothesis set. One may ask if an analysis of the properties of a specific algorithm could lead to finer guarantees. The property of a specific algorithm which relates to the PAC-bound is *the algorithmic stability* of a concrete learning algorithm.

Definition 7.6 (Stability)**.** Consider a loss function $L : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$. A learning algorithm A is said to be *uniformly stable with rate* $\varepsilon : \mathbb{N} \to \mathbb{R}$ if for all $n \in \mathbb{N}$, $i \in \{1, \cdots, n\}$, $(x, y) \in \mathcal{X} \times \mathcal{Y}$, the following inequality holds for all $S, S' \in (\mathcal{X} \times \mathcal{Y})^n$ that differs in only one element

$$|L\big(y, h_S(x)\big) - L\big(y, h_{S'}(x)\big)| \leq \varepsilon(n).$$

Furthermore, A is said to be *on-average stable with rate* $\varepsilon : \mathbb{N} \to \mathbb{R}$ if for all $n \in \mathbb{N}$ and all probability measure P on $\mathcal{X} \times \mathcal{Y}$ we have

$$|\mathbb{E}_{S\sim P^n}\mathbb{E}_{(x,y)\sim P}\mathbb{E}_{i\sim U_{\{1,\cdots,n\}}}(L\big(y_i, h_S(x_i)\big) - L\big(y_i, h_{S^i}(x_i)\big)]| \leq \varepsilon(n) \tag{7.4}$$

where $S = \big((x_i, y_i)\big)_{i=1}^n$, and S^i is obtained from S by replacing the i'th element with (x, y).

Exercise 7.7. (1) Introduce a metric on the space of samples $\mathcal{S} := \cup_{n=1}^{\infty}(\mathcal{X}\times\mathcal{Y})^n$ and construct a function $\tilde{A} : \mathcal{S} \to \mathbb{R}\cup\infty$ such that the uniform stability of a learning algorithm A is equivalent to the absolute continuity of A.

(2) Reformulate the on-average stability of A in terms of "on average-continuity" of the function $\tilde{A}$.

We note that the uniform stability implies the on-average stability with the same rate. The on-average stability is used for upper bound of average of the difference between true risk and empirical risk over training data S, see [Wolf2017, p. 37]. The uniform stability can be used for PAC-bound as follows.

Theorem 7.8. *(*[Wolf2017, Theorem 1.16, p. 38]*) Consider a loss function with range in* $[-c; c]$ *and any learning algorithm* A *that is uniformly stable with rate* $\varepsilon_1 : N \to \mathbb{R}$*. Then the following holds w.r.t. repeated sampling of training data sets of size* n*. For all* $\varepsilon > 0$ *and all probability measures over* $\mathcal{X} \times \mathcal{Y}$

$$P_{S\sim P^n}[|R_S(A(S)) - R(A(S))| \geq \varepsilon + \varepsilon_1(n)] \leq 2\exp[-\frac{n\varepsilon^2}{2\big(n\varepsilon_1(n) + c\big)^2}].$$

7.3. **Weak learnability and adaptive boost.** Another refinement of the notion of PAC-learning is the notion of a γ-weak learnability. M. Kearn and L. Valiant suggested that we begin with an arbitrary learning algorithm, which maybe "weak" but it can be iterated via a "boosting" algorithm to deliver a "strong" learning algorithm. Such an idea of iteration mapping is popular in analysis and topology for producing solution of a PDE equation, which could be consider of a fixed point of a continuous mapping.

To formalize the notion of weak-learner in mathematical language we first recall that the strong learnability of $\mathcal{H}$ requires the existence of a sample complexity $m_{\mathcal{H}}$ in variable ε, δ such that if the sample size of a training data is at least $m_{\mathcal{H}}(\varepsilon, \delta)$ then we have a PAC-bound with ε-accuracy and confidence δ. To weaken this strong learnability we drop the variable ε from the sample complexity function and replace it by a constant $1/2 - \gamma$. The

newly weakened notion of PAC-learnability is called *γ-weak learnable*, and the corresponding learning algorithm is called *a γ-weak learner*.

Once we have the precise notion of a γ-weak learner we can estimate the success of a boosting algorithm which amplifies the accuracy of a weak learner, see e.g. [SSBD2014, Theorem 10.2, p. 135], [Wolf2017, Theorem 1.18, p. 42].

7.4. **Structural risk minimization (SRM).** In many cases the performance of the ERM algorithm is typically very poor in practice. Additionally, determining the ERM solution is computationally intractable. Therefore, ERM is often modified in practice.

One possibility, called *structural risk minimization*, is to consider a sequence of hypotheses classes $\mathcal{H}_1 \subset \mathcal{H}_2 \subset \mathcal{H}_3 \subset \cdots$ of increasing complexity and to optimize the empirical error plus a complexity term $c(\mathcal{H}_n, m)$ for each class $\mathcal{H}_n$ and each sample size m, or plus a regularization term, which is typically defined as $\lambda||h||^2$ for some norm $||,||$ when $\mathcal{H}$ is a subset in a vector space and $\lambda \geq 0$ is a regularization parameter. In SRM paradigm we search the minimizer

$$h_S^{SRM} = argmin_{h\in\mathcal{H}_n,\, n\in\mathbb{N}}\big(R_S^L(h) + c(H_n, m)\big).$$

In regularization paradigm we search the minimizer

$$h_S^{REG} = argmin\big(R_S(h) + \lambda||h||^2\big).$$

There are also many variations of SRM and regularization scheme, see e.g. [SSBD2014].

Note that the idea to use a filtration of finite increasing complexity subsets $\mathcal{H}_1 \subset \mathcal{H}_2 \subset \cdots$ with a perturbation/regularization term is ubiquitous in analysis, geometry and topology.

7.5. **Model selection.** The choice of a right prior information in machine learning is often interpreted as the choice of a right class of hypotheses, which is also called *a model selection*.

In model selection tasks, we try to find the right balance between approximation and estimation errors. An important empirical approach in model selection is called *cross-validation*. The basic idea is to partition the training set into two sets. One is used for training each of the candidate models, and the second is used for deciding which of them yields the best results.

Validation is used for model selection as follows. We first train different algorithms (or the same algorithm with different parameters) on the given training set. Let $\mathcal{H} := \{h_1, \cdots, h_r\}$ be the set of all output predictors of the different algorithms. Now, to choose a single predictor from $\mathcal{H}$ we sample a fresh validation set and choose the predictor that minimizes the error over the validation set. In other words, we apply ERM over the validation set.

The *n-cross validation* is a refinement of cross-validation by partition of the training set into n-subsets and use one of them for testing the and repeat the procedure $(n-1)$-time for other testing subsets.

Remark 7.9. Model selection and cross-validation also belong to numerical experimental testing. There are several machine learning repositories of data for testing, the most famous of them probably are

http://archive.ics.uci.edu/ml,

https://www.kaggle.com

7.6. **Conclusion.** In this section we refine the notion of PAC-learning for the cases when we have prior information on underlying distribution, or on stability of learning algorithm, generally, when we have more specific properties of the underlying hypothesis classes equipped with a learning algorithm. In the next lectures we shall develop more sophisticate methods for most popular learning problems/classes.

8. Support vector machines

Today we shall consider a very simple hypothesis class for binary classification problems and see how general ideas of PAC-learning outlined in the previous lectures are implemented here. The class consists of linear classifiers and the PAC-learning algorithm, called the SVM, is more successful than the ERM rule (Remark 8.12). The original SVM algorithm, also called hard SVM, was invented by Vapnik and Chervonenkis in 1963. The current standard incarnation (soft margin) was proposed by Cortes and Vapnik in 1993 and published in 1995.

8.1. **Linear classifier and hard SVM.** We begin with a linear machine for classifying patterns. Given input pattern $x \in \mathcal{X} = V$, where V is a real Hilbert space, a parameter $\xi = (w, b) \in V \times \mathbb{R}$, we consider an affine function

$$f_\xi(x) = \langle w, x\rangle + b. \tag{8.1}$$

A linear classifier h_ξ divides patterns into two classes as follows

$$h_\xi(x) = sign f_\xi(x) \in \{-1, 1\} = \mathcal{Y}.$$

Let $\mathcal{H}_{lin} \subset \{\pm 1\}^V$ be the hypothesis class consisting of linear classifiers h_ξ, which will be identified with the zero set $H_\xi \subset V$ of f_ξ, since rescaling f_ξ by a positive factor does not change h_ξ. Clearly, the zero set H_ξ is a hyper-plane in V.

Now let $S = (x_1, y_1), \cdots, (x_m, y_m) \in (V \times \{\pm 1\})^m$ be a training sample with $y_i = f(x_i)$ for some unknown target function $f \in \mathcal{H}$ and x_i drawn i.i.d. from V according to some unknown distribution D on V.

Definition 8.1. A training sample S is called *separable*, if there is a hyper-plane $H_\xi \subset V$ such that the corresponding classifier h_ξ correctly classifies S.

Write $S = S_+ \cup S_-$ where

$$S_{\pm 1} := \{(x, y) \in S |\, y = \pm 1\}.$$

Let $Pr : (V \times \{\pm 1\})^m \to V^m$ denote the canonical projection. Then H_ξ correctly separates S iff it separates $\Pr(S_+)$ and $Pr(S_-)$.

We note that a training sample S is separable then the separating hyperplane is not unique and the question arises, which separating hyperplane to choose. The standard approach in the SVM framework is to choose the one that maximizes the distance to the closest points on both sides. This approach is called *the hard SVM rule*. To formulate the hard SVM rule we need a formula for the distance of a point to a hyperplane.

Lemma 8.2 (Distance to a hyperplane)**.** *Let V be a real Hilbert space and $H_\xi := \{z \in V | \langle z, w\rangle + b = 0\}$ be the zero set of f_ξ. The distance of a point $x \in V$ to H_ξ is given by*

$$d(x, H_\xi) := \inf_{z \in H_\xi} ||x - z|| = \frac{|\langle x, w\rangle + b|}{||w||}. \tag{8.2}$$

Proof. Since $H_\xi = H_{\xi/\lambda}$ for all $\lambda > 0$, it suffices to prove (8.2) for the case $||w|| = 1$ and hence we can assume that $w = e_1$. Now formula (8.2) follows immediately, noting that $H_{(e_1,b)} = H_{(e_1,0)} - be_1$. □

Let H_ξ separate $S = \{(x_1, y_1), \cdots, (x_m, y_m)\}$ correctly. Then

$$y_i = sign f_\xi(x_i) = sign(\langle x_i, w\rangle + b),$$

where $\xi = (w, b)$. Hence, by Lemma 8.2, the distance between H_ξ and S is

$$\rho(S, H_\xi) := \min_i d(x_i, H_\xi) = \frac{\min_i y_i(\langle x_i, w\rangle + b)}{||w||} \tag{8.3}$$

Definition 8.3. The distance $\rho(S, H_\xi)$ is called *the margin of the hyperplane H_ξ (or classifier h_ξ) w.r.t. S*. The hyperplanes, which are the parallel to the separating hyperplane and passing through the closest points on the negative or positive sides are called *marginal*.

Now we formulate a learning algorithm A for the hypothesis class $\mathcal{H}_{lin}$. Denote by $\mathcal{H}_S$ the subset of $\mathcal{H}_{lin}$ consisting of hyperplanes that separate S. Then

$$A(S) := arg \max_{H_{\xi'} \in \mathcal{H}_S} \rho(S, H_{\xi'}) \tag{8.4}$$

The domain of the optimization problem in (8.4) is $\mathcal{H}_S$, which is not easy to determine. So we replace this problem by another optimization problem over a convex domain as follows.

Lemma 8.4. *We have*

$$A(S) = arg \max_{(w,b):||w|| \leq 1} \min_i y_i(\langle w, x_i\rangle + b). \tag{8.5}$$

Proof. If $H_{(w,b)}$ separates S then $\rho(S, H_{\xi'}) = \min_i y_i(\langle w, x_i\rangle + b)$. Since the constrain $||w|| \leq 1$ does not effect on H_ξ, this implies that the RHS of (8.5) is less than or equal to the RHS of (8.4). To complete the proof of Lemma 8.4 we observe that if $H_\xi \notin \mathcal{H}_S$ then

$$\min_i y_i(\langle w, x_i\rangle + b) < 0$$

and hence

$$\max_{(w,b):||w||\leq 1} \min_i y_i(\langle w, x_i\rangle + b) = \max_{(w,b):H_{(w,b)}\in H_S} \min_i y_i(\langle w, x_i\rangle + b).$$

This completes the proof of Lemma 8.4. □

Note that the constraint $||w|| \leq 1$ in (8.5) is obtained by fixing the dominator of the far RHS of (8.3) and maximizing the enumerator of the far RHS of (8.3). Equivalently we can fix the enumerator of the far RHS of (8.3) such that

$$\min_i y_i(\langle w, x_i\rangle + b) = 1$$

and minimizing the dominator of the RHS (8.3). Thus (8.4) is equivalent to the following optimization problem which is called *Hard-SVM*

$$(w_0, b_0) = arg \min_{w,b} ||w||^2 \text{ s.t. } \forall i\, y_i(\langle w, x_i\rangle + b) \geq 1. \tag{8.6}$$

Exercise 8.5. Show that the vector w_0 of the solution (w_0, b_0) in (8.6) of the SVM problem is a linear combination of the training set vectors $x_1, \cdots, x_m$. Show that x_i lies on the marginal hyperplane $\langle w, x\rangle + b = \pm 1$.

A vector x_i appears in the linear expansion of the weight vector w_0 in Exercise 8.5 is called *a support vector*.

Remark 8.6. (1) The optimization problem of (8.4) is a specific instance of quadratic programming (QP), a family of problems extensively studied in optimization. A variety of commercial and open-source solvers are available for solving convex QP problems. It is well-known that there is a unique solution of (8.4).

(2) In practice, when we have a large sample set $S \subset \mathbb{R}^n$, the set S is not linearly separable, thus the application of hard SVM is limited.

8.2. **Soft SVM.** Now we consider the case when the sample set S is not linearly separable. There are at least two possibilities to overcome this difficulty. The first one is to find a nonlinear embedding of patterns into a high-dimensional space. To realize this approach we use a kernel trick that embeds the patterns in an infinite dimensional Hilbert space space, which we shall learn in the next lecture. The second way is to seek a predictor h_ξ whose zero set, the hyperplane H_ξ, still has maximal margin in some sense. More precisely, we shall relax the hard SVM rule (8.6) by replacing the constraint

$$y_i(\langle w, x_i\rangle + b) \geq 1 \tag{8.7}$$

by the relaxed constraint

$$y_i(\langle w, x_i\rangle) + b \geq 1 - \xi_i \tag{8.8}$$

where $\xi_i \geq 0$ are called *the slack variables*. The slack variables are commonly used in optimization to define relaxed versions of some constraints. In our

case a slack variable ξ_i measures the distance by which vector x_i violates the original inequality in the LHS of (8.7).

The relaxed hard SVM rule is called *the soft SVM rule*. It jointly minimizes the norm of $||w||$ (corresponding to the margin) and the average of ξ_i (corresponding to the violations of the constraints). More precisely we have to solve the following optimization problem with a parameter $\lambda > 0$ that controls the tradeoff between two terms: the norm $||w||$ and the average of ξ_i:

$$(w_0, b_0, \xi_0) = arg\min_{w,b,\xi}(\lambda||w||^2 + \frac{1}{m}\sum_{i=1}^{m}\xi_i) \tag{8.9}$$

$$\text{s. t. } \forall i,\ y_i(\langle w, x_i\rangle + b) \geq 1 - \xi_i \text{ and } \xi_i \geq 0. \tag{8.10}$$

Remark 8.7. (1) For a hyperplane $H_{(w,b)}$ a vector x_i with $\xi > 0$ can be viewed as *an outlier*. If we omit the outlier, the training data is correctly separated by $H_{(w,b)}$ with margin $\rho = 1/||w||$.

(2) There is some arbitrariness in how one penalizes large ξ. In Equation (8.9) we have essentially chosen the l_1-norm of ξ. Another common choice would be the l_2-norm. It is also possible to choose l_p-norm, see e.g. [MRT2012, (4.23)].

In order to apply the ideas developed in the previous lectures for solving the equation defining the soft SVM rule, we shall refomulate the equation (8.9) as a regularized ERM rule discussed in the framework of SRM.

First we define the hinge loss function $L^{hinge} : \mathcal{H}_{lin} \times (V \times \{\pm 1\}) \to \mathbb{R}$ for the hypothesis class H_{lin} as follows

$$L^{hinge}(h_{(w,b)}, (x, y)) := \max\{0, 1 - y(\langle w, x\rangle + b)\}. \tag{8.11}$$

We abbreviate the empirical hinge risk function

$$R_S^{L^{hinge}}(h_{(w,b)}) = \frac{1}{m}\sum \max\{0, 1 - y_i(\langle w, x_i\rangle + b)$$

where $S = \{(x_1, y_1), \cdots, (x_m, y_m)\}$ as R_S^{hinge}. [18]

Lemma 8.8. *The Equation (8.9) with constraint (8.10) for the soft SVM is equivalent to the following regularized risk minimization problem, which does not depend on the slack variables ξ:*

$$arg\min_{w,b}\Big(\lambda||w||^2 + R_S^{hinge}\big((w, b)\big)\Big). \tag{8.12}$$

Proof. The idea of the proof is that a slack variable ξ_0 of a solution (w_0, b_0, ξ_0) of the minimization problem (8.9) must be a function of (w_0, b_0). To see that we fix (w_0, b_0) and minimize the RHS of (8.9) under the constraint (8.10). Now it is straightforward to see that $\xi_i = L^{hinge}\big((w, b), (x_i, y_i)\big)$ which completes the proof. □

[18] It is not hard to see that R_S^{hinge} can be written using another hinge loss $L_{hinge} : \mathbb{R} \times \mathbb{R} \to \mathbb{R} : (y, y') \mapsto \max\{0, 1 - yy'\}$ in the same way as in Subsection 5.1, see also Remark 8.12.

8.3. **PAC-learnability of SVM.** In [MRT2012] and [SSBD2014] the authors proved different PAC-bounds for hard and soft SVMs. In this subsection we present PAC-bounds for SVM given in [SSBD2014, §26.3].

To simplify the exposition we shall assume that the linear classifier $h_\xi : V \to \{\pm 1\}$ is defined by a linear function $f_{w'}$ on a Hilbert space $V' = \langle e_1\rangle_{\otimes\mathbb{R}} \oplus V$, i.e. we incorporate the bias term b of f_ξ in (8.1) into the term w as an extra coordinate. More precisely for $\xi = (w, b)$ we set

$$w' := be_1 + v \text{ and } x' := e_1 + x.$$

Then

$$f_{(w,b)}(x) = f_{w'}(x').$$

Clearly the projection of the zero set $H_{w'}$ of $f_{w'}$ to $\mathbb{R}^d$ is the zero set H_ξ of f_ξ. Thus learning using the SVM with affine functions in $\mathbb{R}^d$ is the same as learning using the SVM with linear functions in $\mathbb{R}^{d+1}$.

All the PAC-bounds for SVMs given in [MRT2012] and [SSBD2014] assume a condition on the underlying distribution on the instance space $\mathbb{R}^d$. In the PAC bounds below the dimension d of the instance space V does not play any role.

Definition 8.9. ([SSBD2014, Definition 15.3]) Let D be a distribution on $\mathbb{R}^d \times \{\pm 1\}$. We say that D *is separable with a* (γ, ρ)*-margin* if there exists $(w^*, b^*) \in \mathbb{R}^d \times \mathbb{R}$ such that $\|w\| = 1$ and such that

$$P_{(x,y)\sim D}[y(\langle w^*, x\rangle + b^*) \geq \gamma \text{ and } ||x|| \leq \rho] = 1.$$

Similarly, we say that D is separable with a (γ, ρ)-margin using a homogeneous half-space if the preceding holds with a half-space defined by a vector $(w^*, 0)$.

Theorem 8.10. *(*[SSBD2014, Theorem 15.4]*) Let D be a distribution over $\mathbb{R}^d \times \{\pm 1\}$ that satisfies the (γ, ρ) separability with margin assumption using a homogeneous halfspace. Let A denote the hard SVM. Then we have*

$$P_{S\sim D^m}[R_D(A(S)) \leq \sqrt{\frac{4(\rho/\gamma)^2}{m}} + \sqrt{\frac{2\log(2/\delta)}{m}}] \geq 1 - \delta.$$

Thus the PAC-learnability of hard SVM depends only on the ratio ρ/γ. Since γ measures the margin of the separating hyperplane, this PAC-bounds justifies the choice of maximizing margin separating hyperplane.

Next, using the Markov inequality, instead of a PAC-bound we present a distribution dependent bound on the expectation of the true risk/error probability of the soft SVM.

Theorem 8.11. *(*[SSBD2014, Corollary 15.7]*) Let D be a distribution over the ball $B(0, \rho)$ of radius ρ centered at the origin of V. Let A denote the*

soft SVM rule. Then for every $r > 0$ we have

$$\mathbb{E}_{S\sim D^m}\Big(R_D\big(A(S)\big)\Big) \leq \mathbb{E}_{S\sim D^m}\Big(R_D^{hinge}\big(A(S)\big)\Big)$$

$$\leq \min_{w\in B_r(0)} R_D^{hinge}(h_w) + \sqrt{\frac{8\rho^2 r^2}{m}}. \tag{8.13}$$

Remark 8.12. (1) Theorem 8.11 follows from a general PAC-bound for regularized empirical risk minimization [SSBD2014, Corollary 13.8], taking into account that the hinge loss function,

$$L^{hinge} : \mathbb{R}\times\mathbb{R}\to\mathbb{R}_+,\ (y,y')\mapsto \max\{0, 1-yy'\},$$

upper bounds the true loss function, i.e.,

$$\text{if } y\in\{\pm 1\} \text{ then } 1-\delta^y_{sign\, h(x)} \leq L^{hinge}(y; h(x)).$$

(2) In [SSBD2014, p. 209] the authors showed that the PAC-bound in (8.13) is much better than the VC bound for the ERM rule for the true probability error on the same class of linear classifiers.

Exercise 8.13. 1) Do the hard-and soft SVM work on infinite dimensional Hilbert space of instances?

2) Compare the PAC-bound given in [MRT2012, Corollary 4.1, p. 82] with the PAC-bound in Theorem 8.11.

8.4. **Conclusion.** In this section we consider the hard- and soft SVM for binary linear classifiers. The hard SVM is reduced to a convex optimization problem with affine constraint, which is a classical problem in optimization. The soft SVM is equivalent to a regularized ERM for the hinge loss function, which can be analyzed using SRM theory. The PAC learnability of hard and soft SVM is general better than ERM rule (for the true risk function), which also works for binary classifiers on finite dimensional spaces since the VC-dimension of the hypothesis class in this case is finite by Radon's theorem.

9. Kernel methods and RKHS

In the previous lecture we considered the hypothesis class $\mathcal{H}_{lin}$ of linear classifiers, whose zero sets are hyperplanes. A linear classifier h_ξ correctly classifies a training sample S in a labeled sample space $V\times\{\pm 1\}$ iff the zero set H_ξ of h_ξ separates the subsets $Pr(S_-)$ and $Pr(S_+)$ of the patterns in S. By Radon's theorem any set of distinct $(d+2)$ points in $\mathbb{R}^d$ can be partitioned into two subsets that cannot be separated by a hyperplane in $\mathbb{R}^d$. Thus it is reasonable to enlarge the hypothesis class $\mathcal{H}_{lin}$ by adding polynomial classifiers. Note that any (polynomial) function $y = f(x)$ on $\mathbb{R}^d$ can be regarded as the restriction of the linear function y, which is the new coordinate, on $\mathbb{R}^d\times\mathbb{R}$ to the image of the graph Γ_f of f : $f(x) = y(\Gamma_f(x)) = [\Gamma_f^*(y)](x)$.

However, the computational complexity of learning by polynomial embedding in higher dimension may be computationally expensive. The common

solution to this concern is kernel based learning. The term "kernels" is used in this context to describe inner products in the feature space. The kernel trick simplifies computational aspect of learning by polynomial classifiers, or more geometric, by applying the SVM to the image of input data via a mapping, not necessary polynomial, into a real Hilbert space of higher dimension.

9.1. **Kernel trick.** In the previous lecture we learned that a solution of a hard SVM can be expressed as a linear combination of support vectors (Exercise 8.5). If the number of support vectors is less than the dimension of the instance space, then this property simplifies the search for a solution of the hard SVM. Below we shall show that this property is a consequence of the representer theorem concerning solutions of a special optimization problem. The optimization problem we are interested in is of the following form:

$$w_0 = arg\min_{w\in W}\Big(f\big(\langle w,\psi(x_1)\rangle,\cdots,\langle w,\psi(x_m)\rangle\big) + R(\|w\|)\Big) \tag{9.1}$$

where w and $\psi(x_i)$ are elements of a Hilbert space W, $f : \mathbb{R}^m \to \mathbb{R}$ is an arbitrary function and $R : \mathbb{R}_+ \to \mathbb{R}$ is a monotonically non-decreasing function. The map ψ is often called *the feature map*, and W is called *the feature space*.

Example 9.1. (1) The equation (8.12) for a solution of soft SVM is an instance of Equation (9.1).

(2) The Equation (8.6) for a solution of hard SVM can be derived from Equation (9.1) by letting $R(a) = a^2$ and letting $f(a_1,\cdots,a_m)$ be 0 if there exists b such that $y_i(a_i + b) \geq 1$ for all i, and $f(a_1,\cdots,a_m) = \infty$ otherwise.

Theorem 9.2 (Representer theorem)**.** *Let $\psi : \mathcal{X} \to W$ is a mapping from an instance space $\mathcal{X} = V$ to a Hilbert space W and w_0 a solution of (9.1). Then the projection of w_0 to the subspace $\langle\psi(x_1),\cdots,\psi(x_m)\rangle_{\otimes\mathbb{R}}$ in W is also a solution of (9.1).*

Proof. Assume that w_0 is a solution of (9.1). Then we can write

$$w_0 = \sum_{i=1}^{m}\alpha_i\psi(x_i) + u$$

where $\langle u,\psi(x_i)\rangle = 0$ for all i. Set $\bar{w}_0 := w_0 - u$. Then

$$\|\bar{w}_0\| \leq \|w_0\| \tag{9.2}$$

and since $\langle\bar{w}_0,\psi(x_i)\rangle = \langle w_0,\psi(x_i)\rangle$ we have

$$f\big(\langle\bar{w}_0,\psi(x_1)\rangle,\cdots,\langle\bar{w}_0,\psi(x_m)\rangle\big) = f\big(\langle w_0,\psi(x_1)\rangle,\cdots,\langle w_0,\psi(x_m)\rangle\big). \tag{9.3}$$

From (9.2), (9.3) and taking into account the monotonicity of R, we conclude that $\bar{w}_0$ is also a solution of (9.1). This completes the proof of Theorem 9.2. □

The representer theorem implies that it suffices to find a minimizer w_0 of (9.1) that lies on the smaller subspace $W_1 := \langle \psi(x_i) \rangle_{\otimes \mathbb{R}} \subset W$. If R is strictly monotone, then any minimizer of (9.1) lies on W_1. In what follows we shall describe the method to solve this "smaller" minimization problem, which is called *the kernel trick*.

Let

- $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, $K(x, x') := \langle \psi(x), \psi(x') \rangle$ be *a kernel function*.
- $G = (G_{ij})$ be the Gram matrix, where $G_{ij} := K(x_i, x_j)$.

Since a solution w_0 of the minimization problem (9.1) has the form $w_0 = \sum \alpha_j \psi(x_j)$, the coefficients α_i are a solution of the following minimization problem

$$arg \min_{\alpha \in \mathbb{R}^m} f\big(\sum_{j=1}^{m} \alpha_j G_{j1}, \cdots, \sum_{j=1}^{m} \alpha_j G_{jm}\big) + R\Big(\sqrt{\sum_{i,j=1}^{m} \alpha_i \alpha_j G_{ji}}\Big). \tag{9.4}$$

Once we know a solution α of (9.4) we can predict any sample $x \in W$ using "nonlinear" classifier w_0 as follows

$$\langle w_0, \psi(x) \rangle = \sum_{i=1}^{m} \alpha_j \langle \psi(x_j), \psi(x) \rangle = \sum_{j=1}^{m} \alpha_j K(x_j, x). \tag{9.5}$$

To compute (9.5) we need to know only the kernel function K and not the mapping ψ, nor the inner product $\langle , \rangle$ on the Hilbert space W.

This motivates the following question.

Problem 9.3. *Find a sufficient and necessary condition for a function, also called a kernel, $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that K can be written as $K(x, x') = \langle \psi(x), \psi(x') \rangle$ for a mapping $\psi : \mathcal{X} \to W$, where W is a real Hilbert space.*

If K satisfies the condition in Problem 9.3 we shall say that K is induced from the inner product in a Hilbert space W via a (feature) mapping ψ. The target Hilbert space is also called *a feature space*.

9.2. **PSD kernels and reproducing kernel Hilbert spaces.**

9.2.1. *Positive semi-definite kernel.* We note that a necessary condition for $K(x, x')$ to be written as $\langle \psi(x), \psi(x') \rangle$ is the positive semi-definite (PSD) [19] of K.

Definition 9.4. Let $\mathcal{X}$ be an arbitrary set. A map $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called *positive semi-definite kernel* (PSD kernel) iff for all $x_1, \cdots, x_m$ the Gram matrix $G_{ij} = K(x_i, x_j)$ is positive semi-definite.

[19] In [MRT2012, p. 92] the authors also use PDS for "positive definite symmetric" but the expression "positive" matrix always means a property of the associated quadratic form, which is derived from a symmetric bilinear form. Furthermore the positive definiteness is a very strong condition, it implies that for any m-tuples of points $x_1, \cdots, x_m \in \mathcal{X}$ the vectors $\psi(x_1), \cdots, \psi(x_m)$ are linear independent.

Theorem 9.5. *A function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is induced from the inner product in some Hilbert space if and only if it is positive semi-definite.*

Proof. 1) Let us prove the "only if" assertion of Theorem 9.5. Assume that $K(x, x') = \langle \psi(x), \psi(x') \rangle$ for a mapping $\psi : \mathcal{X} \to W$, where W is a Hilbert space. Given m points $x_1, \cdots, x_m \in \mathcal{X}$ we consider the subspace $W_m \subset W$ generated by $\psi(x_1), \cdots, \psi(x_m)$. Using the positive definite of the inner product on W_m, we conclude that the Gram matrix $G_{ij} = K(x_i, x_j)$ is positive semi-definite. This proves the "only if" part of Theorem 9.5

2) Let us prove the "if" part. Assume that $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is positive semidefinite. For each $x \in \mathcal{X}$ let $K_x \in \mathbb{R}^{\mathcal{X}}$ be the function defined by

$$K_x(y) := K(x, y).$$

Denote by

$$W := \{f \in \mathbb{R}^{\mathcal{X}} \mid f = \sum_i a_i K_{x_i},\ a_i \in \mathbb{R}\}.$$

Then W is equipped with the following inner-product

$$\langle \sum_i \alpha_i K_{x_i}, \sum_j \beta_j K_{y_j} \rangle := \sum_{i,j} \alpha_i \beta_j K(x_i, y_j).$$

The PSD property of K implies that the inner product is positive semi-definite, i.e.

$$\langle \sum_i \alpha_i K_{x_i}, \sum_j \alpha_i K_{x_i} \rangle \geq 0.$$

Since the inner product is positive semi-definite, the Cauchy-Schwarz inequality implies for $f \in W$ and $x \in \mathcal{X}$

$$\langle f, K_x \rangle^2 \leq \langle f, f \rangle \langle K_x, K_x \rangle. \tag{9.6}$$

Since for all x, y we have $K_y(x) = K(y, x) = \langle K_y, K_x \rangle$, it follows that for all $f \in W$ we have

$$f(x) = \langle f, K_x \rangle. \tag{9.7}$$

Using (9.7), we obtain from (9.6) for all $x \in \mathcal{X}$

$$|f(x)|^2 \leq \langle f, f \rangle K(x, x).$$

This proves that the inner product on W is positive definite and hence W is a pre-Hilbert space. Let $\mathcal{H}$ be the completion of W. The map $x \mapsto K_x$ is the desired mapping from $\mathcal{X}$ to $\mathcal{H}$. This completes the proof of Theorem 9.5. □

Example 9.6. (1) (Polynomial kernels). On $\mathcal{X} = \mathbb{R}^d$ any polynomial in $\langle x, y \rangle$ with non-negative coefficients is a PSD kernel since a sum of PDS kernels is a kernel by the convexity of the space of symmetric positive semi-definite matrix (we could also use Theorem 9.5 to prove this fact), and the product of PDS kernels is also a PDS kernel by the Schur Product Theorem. In particular, $K(x, y) := (1 + \langle x, y \rangle)^2$ is a PSD kernel.

(2) (Exponential kernel). For any $\gamma > 0$ the function (also called kernel) $K(x, y) := \exp\big(\gamma \cdot \langle x, y\rangle\big)$ is a PDS kernel, since it is the limit of a polynomials in $\langle x, y\rangle$ with non-negative coefficients.

Exercise 9.7. (1) Show that the Gaussian kernel $K(x, y) := \exp\big(-\frac{\gamma}{2}||x - y||^2$ is a PDS kernel.

(2) Let $\mathcal{X} = B(0, 1)$ - the open ball of radius 1 centered at the origin $0 \in \mathbb{R}^d$. Show that $K(x, y) := (1 - \langle x, y\rangle)^{-p}$ is a PDS kernel for any $p > 0$.

9.2.2. *Reproducing kernel Hilbert space.* For a given PSD kernel, the corresponding feature map and feature space are not unique (e.g., we compose the feature map with an isometric embedding of the feature space into another Hilbert space). However, there is a canonical choice for the feature space, a so-called reproducing kernel Hilbert space.

Definition 9.8 (Reproducing kernel Hilbert space)**.** Let $\mathcal{X}$ be a set and $\mathcal{H} \subset \mathbb{R}^{\mathcal{X}}$ a real Hilbert space of functions on $\mathcal{X}$ with the unique vector space structure such that for $x \in \mathcal{X}$ the evaluation map

$$ev_x : \mathcal{H} \to \mathbb{R}\,, ev_x(f) := f(x)$$

is a linear map. Then $\mathcal{H}$ is called a *reproducing kernel Hilbert space* (RKHS) on $\mathcal{X}$ if for all $x \in \mathcal{X}$ the linear map ev_x is bounded i.e., $\sup_{f \in B(0,1) \subset \mathcal{H}} ev_x(f) < \infty$.

Remark 9.9. Let $\mathcal{H}$ be a RKHS on $\mathcal{X}$ and $x \in \mathcal{X}$. Since ev_x is bounded, by the Riesz representation theorem there is a function $k_x \in \mathcal{H}$ so that $f(x) = \langle f, k_x\rangle$ for all $f \in \mathcal{H}$. Then the kernel

$$K(x, y) := \langle k_x, k_y\rangle$$

is a PSD kernel. K is called *the reproducing kernel of* $\mathcal{H}$.

Theorem 9.10. *Let $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a PSD kernel. There there exists a unique RKHS $\mathcal{H}$ such that K is the reproducing kernel of $\mathcal{H}$.*

Proof. By Theorem 9.5 there exists a RKHS $\mathcal{H}$ and a mapping $\varphi : \mathcal{X} \to \mathcal{H}$ such that

$$\forall x, x' \in \mathcal{X} \text{ we have } K(x, x') = \langle K_x, K_{x'}\rangle. \tag{9.8}$$

From (9.7) and (9.8) we conclude that $\mathcal{H}$ has the reproducing property. To show the uniqueness of a RKHS $\mathcal{H}$ we assume that there exists another RKHS $\mathcal{H}'$ such that $K(x, y) = \langle k_x, k_y\rangle$ where $f(x) = \langle f, k_x\rangle$. We define a map $g : \mathcal{H} \to \mathcal{H}'$ by setting $g(K_x) = k_x$. It is not hard to see that g is an isometric embedding. To show that g extends to an isometry it suffices to show that the set k_x is dense in $\mathcal{H}'$. Assume the opposite, i.e. there exists $f \in \mathcal{H}'$ such that $\langle f, k_x\rangle = 0$ for all x. But this implies that $f(x) = 0$ for all x and hence $f = 0$. This completes the proof of Theorem 9.10. □

9.3. **Generalization property of kernel methods.** The kernel method when implemented is also ERM with a special form of loss function. The main tool to prove a generalization bound for such a general ERM is to use the Rademacher complexity [MRT2012, §5.3.3], [SSBD2014, §263, p. 383]. For a sample $S = (x_1, \cdots, x_m)$ we denote by $K[S]$ the matrix in $Mat_{m\times m}$ whose entries $K[S]_{ij}$ are equal to $K(x_i, x_j)$.

Theorem 9.11 (Rademacher complexity of kernel-based hypotheses)**.** *(*[MRT2012, Theorem 5.5, p.102]*) Let $K : \mathcal{X}\times\mathcal{X} \to \mathbb{R}$ be a PSD kernel and let $\tilde{K} : \mathcal{X} \to \mathcal{H}$ be a feature mapping associated to K, i.e., $\tilde{K}(x) = K_x$ for all $x \in \mathcal{X}$. Let $S \subset \{x \in \mathcal{X} : K(x, x) \leq r^2\}$ be a sample of size m, and let $\mathcal{H}_K := \{x \mapsto \langle w, K_x\rangle,$ where $||w||_{\mathcal{H}} \leq \Lambda\} \subset \mathbb{R}^{\mathcal{X}}$ for some $\Lambda > 0$. Then*

$$\mathcal{R}_S(\mathcal{H}_K) \leq \frac{\Lambda\sqrt{TrK[S]}}{m} \leq \sqrt{\frac{r^2\Lambda^2}{m}}.$$

From this lower bound for the Rademacher complexity, using Talagrand's lemma (see [MRT2012, Lemma 4.2, p. 78]), one derives the PAC-type bound for the true risk of a ERM predictor with special loss form see [MRT2012, Corollary 5.1, p. 103], which proves the generalization property of the kernel algorithm.

9.4. **Conclusion.** In this section we learn the kernel trick which simplifies the algorithm of hard SVM using embedding of patterns into a Hilbert space. The kernel trick is based on the theory of RKHS. The learnability of the kernel algorithm is ensured by an upper bound for the empirical Rademacher complexity of the hypothesis class in consideration. The main difficulty of the kernel method is that we still have no general method of selecting a suitable kernel for a concrete problem.

10. Neural networks

In the previous two lectures we learned basic techniques of machine learning: SVMs and the related kernel tricks, also called kernelized SVMs, that have been invented by Vapnik and Chervonenskis at the beginning of statistical learning theory and developed further into main techniques in machine learning till recently. We examined their generalization properties, expressed via PAC-bounds, using the Rademacher complexities.

Today most powerful learning machines are artificial neural networks, shortened as neural networks (otherwise, non-artifical neural networks are (called) biological neural networks). Neural networks achieve outstanding performance on many important problems in computer vision, speech recognition, and natural language processing. They're being deployed on a large scale by companies such as Google, Microsoft, and Facebook.

In today lecture we shall study neural networks, concentrating on most known about them: their expressive power, i.e. how good they can be used

to represent computable functions. Regarding neural networks as a hypothesis class of functions the networks represent we think of the expressive power of neural networks as the approximation error of the hypothesis class in consideration. We shall also discuss two methods of training a neural network using gradient descents: *the back-propagation* and *the stochastic gradient descent (SGD)*.

In this lecture we shall not consider the sample complexity of neural networks, since we don't know any generalization property of neural networks that can be proved using the VC-dimension or the Rademacher complexity [Wolf2017], [SSBD2014] , [ZBHRV2016].

10.1. **Neural networks as computing circuits.** Neural networks are computing circuits. They are similar to Boolian and arithmetic circuits in CS and therefore many results concerning computational complexities of Boolian circuits or arithmetic circuits have analogues in neural networks. Let me introduce main notations and concepts.

• *A neural network* is a quadruple (V, E, σ, w) where V and E are the sets of nodes and directed edges connecting nodes of the network.

• The graph (V, E) is called *the underlying graph of the network*.

• Each node in V is modeled as a function $\sigma : \mathbb{R} \to \mathbb{R}$, which is also called *the activation function*.

Most common activation functions are:
- the sign function $\sigma(x) = sign(x)$,
- the threshold function $\sigma(x) = 1_{\mathbb{R}^+}(x)$,
- the sigmoid function $\sigma(x) := \frac{1}{1+e^{-x}}$, which is a smooth approximation to the threshold function.

• $w : E \to \mathbb{R}$ is called *the weight function of the network*.

• *The networks architecture* of a neural network is the triple $G = (V, E, \sigma)$.

• *The input* $I(\mathfrak{n})$ *of a neuron* $\mathfrak{n}$ is equal to the weighted sum of the outputs of all the neuron connected to it: $I(\mathfrak{n}) = \sum w(\mathfrak{n}'\mathfrak{n})O(\mathfrak{n}')$, where $\mathfrak{n}'n \in E$ is a directed edge and $O(\mathfrak{n}')$ is the output of the neuron $\mathfrak{n}'$ in the network.

• *The output* $O(\mathfrak{n})$ *of a neuron* $\mathfrak{n}$ is obtained from the input $I(\mathfrak{n})$ as follows: $O(\mathfrak{n}) = \sigma(I(\mathfrak{n}))$.

• *The i-th input nodes* give the output x_i. If the input space is $\mathbb{R}^n$ then we have $n + 1$ input-nodes, one of them is the "constant" neuron, whose output is 1.

Thus each neural network (E, V, w, σ) represents a multivariate multivariable function which we denote by $h_{V,E,\sigma,w}$. For each architecture (V, E, σ) we denote by

$$\mathcal{H}_{V,E,\sigma} = \{h_{V,E,\sigma,w} : \ w \in \mathbb{R}^E\}$$

the underlying hypothesis class of functions from the input space to the output space of the network.

Remark 10.1. Neural networks are abstraction of biological neural networks, and the activation function is usually an abstraction representing

the rate of action potential firing in the cell. In its simplest form, this function is binary, that is, either the neuron is firing or not. We can consider activation function as a filter of relevant information, or introducing the non-linearlity in regression problems.

Neural networks are classified by type of their underlying graph.

- *A feedforward network* has underlying acyclic directed graph. Otherwise, it is called *a recurrent network*.

- *A layered feedforward neural network* FN has vertices arranged in a disjoint union of layers $V = \cup_{l=0}^{n} V_i$ such that every edge in E connects nodes in neighboring layers V_i, V_{i+1}. The *depth* of the network FN is m. V_0 is called *the input layer*, V_n is called *the output layer*, the other layer is called *hidden*.

Example 10.2. A binary linear classifier is a neuron network with a single neuron (perceptron). A perceptron has been introduced by Rosenblatt m 1958 for pattern recognition problem. A key difference compared to the perceptron, however, is that the neural network uses continuous sigmoidal nonlinearities in the hidden units, whereas the perceptron uses step-function nonlinearities.

10.2. **The expressive power of neural networks.** In this section we want to address the following problems.

Problem 10.3. *Which functions can be represented/implemented or approximated using a neural network?*

Rephrasing, we study the approximation error of the hypothesis classes defined by all functions computable by neural networks.

Now assume that a function f (resp. a function class $\mathcal{F}$) is represented by a neural network. As in classical computational complexity, we define *a size of a neural network* the number of its edges. Then we define *the neural network size of* f (resp. of $\mathcal{F}$) as the minimal size of the neural network representing f.

Problem 10.4. *Find a lower bound of the neural network size of f (resp. $\mathcal{F}$).*

As in classical computational complexity, the versions of Problems 10.3 and 10.4 where networks are supposed to have a given depth are also interesting to study.

Problem 10.3 has positive answers.

Theorem 10.5. *Every continuous function $f : [0,1]^n \to \mathbb{R}$ can be represented by a neural network of depth 2.*

Proof. We shall show that Theorem 10.5 follows from the following

Proposition 10.6 (Kolgomorov's superposition theorem)**.** *Every continuous function f on $[0,1]^n$ can be written as follows*

$$f(x_1, x_2, \cdots, x_n) = \sum_{k=1}^{2n+1} h_k\Big(\sum_{i=1}^{n} \varphi_{ik}(x_i)\Big) \tag{10.1}$$

where h_k and φ_k are continuous functions and φ_{ik} are chosen independent from f.

It is not hard to see that the RHS of (10.1) can be represented by a neural network with one hidden layer with $2n+1$ neurons. This completes the proof of Theorem 10.5. □

Remark 10.7. The function h_k depends on f. Thus Theorem 10.5 is proved using an activation function h_k of possibly high computational complexity.

Generalizing Theorem 10.5 to the class of multivariate functions, we have to replace the representability of a function f by a neural network by the approximability of f by a neural network.

Theorem 10.8. *(*[Wolf2017, Theorem 2.6]*) Let $d, d' \in \mathbb{N}$, $K \subset \mathbb{R}^d$ be a compact, and $\sigma : \mathbb{R} \to \mathbb{R}$ any activation function that is*
(i) continuous and non-polynomial or
(ii) bounded and so that the limits $\lim_{z\to\omega\infty} \sigma(z)$ exists in $\mathbb{R}$ and different from each other.

Then the set of functions representable by a feedforward neural network with a single hidden layer of neuron with activation function σ is dense in the space of continuous functions $K \to \mathbb{R}^{d'}$ in the topology of uniform convergence.

For the proof see [Wolf2017].

Problem 10.4 has been discussed widely, see e.g. [Wolf2017], [SSBD2014, §20.3, p. 271].

Proposition 10.9 (Representation of Boolean functions)**.** *(* [SSBD2014, Claim 20.1, p. 271]*) Every Boolean function $f : \{0,1\}^d \to \{0,1\}$ can be represented exactly by a feedforward neural network with a single hidden layer containing at most $2^d + 1$ neurons, if $\sigma(x) = sign(x)$ is used as activation function.*

Outline of the proof. The idea of the proof is to choose the "maximal" underlying graph of feedforward neural networks in Proposition 10.9 and show that we can adjust the weight function to obtain a desired function $f : \{0,1\}^d \to \{0,1\}$. The maximal graph has $|V_0| = d+1$, $|V_1| = s^d + 1$ and all possible edges between adjacent layers. Let $u_i \in f^{-1}(1)$. Set

$$g_i(x) := sign(\langle x, u_i\rangle - d + 1).$$

Then $\{g_i\}$ can be implemented by the neurons in V_1. Now set

$$f(x) := sign(\sum_{i=1}^{k} g_i(x) + k - 1),$$

where $k = \#f^{-1}(1)$. □

Remark 10.10. The preceding claim shows that neural networks can implement any Boolean function. However, this seems a very weak property, as the size of the resulting network might be exponentially large. It turns out that this "weak" property is a result of high computational complexity of the class of Boolian functions in consideration. More precisely we have a theorem stating that the neuron network circuit size of a Boolian function is at most a quadartic function of its time computational complexity [SSBD2014, Theorem 20.3, p. 272]. This theorem is a consequence of the relation between the neuron network circuit size, the Boolian circuit size, the time complexity of the program.

The following exercise give a lower bound for the neural network size of a function.

Exercise 10.11. Let $f : \mathbb{R}^d \to \{-1, 1\}$ be given as follows

$$f(x) = sign(-1 + \sum_{i=1}^{d} \max\{0, x_i\}).$$

Show that the size of a neural network with a single hidden layer and activation function $\sigma = sign$ that represent f is larger or equal to 2^{d-1}.

10.3. **Training neural networks.** Training a neural network is the popular name for running a learning algorithm for a hypothesis class $\mathcal{H}_{V,E,\sigma}$. We shall consider only the case where the input space and the output space of the networks are euclidean spaces $\mathbb{R}^n$ and $\mathbb{R}^m$ respectively.

Recall that, given a training set S of labeled pairs $(x_1, y_1), \cdots, (x_N, y_N)$, where $x_i \in \mathbb{R}^n$ and $y_i \in \mathbb{R}^m$, a learning algorithm assigns a weight function $w_S : E \to \mathbb{R}$ that minimizes a risk function. The risk function we use train our network is the MSE that is the expected error of the squared error/loss L (cf. (2.10) [20] defined as follows for $h_w \in \mathcal{H}_{V,E,\sigma}$

$$L(h_w(x), y) = \frac{1}{2}||h_w(x) - y||^2. \tag{10.2}$$

In what follows we rewrite the loss function $\tilde{L}$ in the form of (5.3), i.e. we consider $\tilde{L}$ as a function on $\mathbb{R}^E \times (\mathcal{X} \times \mathcal{Y})$

$$\tilde{L}(w, z) := L(h_w(x), y)$$

for $z = (x, y)$. Hence for $D \in \mathcal{P}(\mathbb{R}^n \times \mathbb{R}^m)$ the risk function is defined as follows

$$R_D^L(h_w) = \mathbb{E}_{z\sim D}\big(\tilde{L}(w, z)\big).$$

Since D is unknown the classical learning algorithm minimizes the empirical risk R_S^L and using sample complexity to prove that if h_S minimizes R_S^L then it also minimizes with ε-accuracy and δ-confidence if $\#(S) \geq m(\varepsilon, \delta)$.

[20] we use the notation of the loss function defined in (5.3)

Current training methods of neural networks uses gradient flows to find approximate minimizers of R_S^L, which in certain cases are also approximate minimizers of R_D^L. The methods are based on the following observation. For any differentiable function f on a Riemannian manifold (M, g) denote by $\nabla_g f$ the gradient of f w.r.t. g , i.e., for any $x \in M$ and any $V \in T_x M$ we have

$$df(V) = \langle \nabla_g f, X \rangle. \tag{10.3}$$

If g is fixed we just write ∇f instead of $\nabla_g f$.

The negative gradient flow of f on M is a dynamic system on M defined by the following ODE

$$x(0) = x_0 \in M \text{ and } \dot{x}(t) = -\nabla f(x(t)). \tag{10.4}$$

If $x(t)$ is a solution of (10.4) then $f(x(t)) < f(x(t'))$ for any $t' > t$ unless $\nabla f(x(t)) = 0$, i.e., $x(t)$ is a critical point of f.

If f is not differentiable we modify the notion of the gradient of f as follows.

Definition 10.12. Let $f : S \to \mathbb{R}$ be a function on an open convex set $S \subset \mathbb{R}^n$. A vector $v \in \mathbb{R}^n$ is called *a subgradient of f at $w \in S$* if

$$\forall u \in S,\ f(u) \geq f(w) + \langle u - w, v \rangle. \tag{10.5}$$

The set of subgradients of f at w is called *the differential set* and denoted by $\partial f(w)$.

Remark 10.13. It is known that a subgradient of a function f on a convex open domain S exists at every point $w \in S$ iff f is convex, see e.g. [SSBD2014, Lemma 14.3].

Gradient descent algorithms discretize the solution of the gradient flow equation (10.4). As a result we begin with arbitrary point x_0 on M and set

$$x_{n+1} = x_n - \gamma_n \nabla f(x_n), \tag{10.6}$$

where $\gamma_n \in \mathbb{R}_{\geq 0}$ is a constant, called a "learning rate" in machine learning, to be optimized. The algorithm works if ∇f is Lipschitz [Wolf2017, Theorem 2.10], see also [SSBD2014, Corollary 14.2, p. 188].

• *Back-propagation*, short for "backward propagation of errors", is a trick/algorithm to compute the gradient of the risk function R_S^L on the Euclidean space $\mathbb{R}^{|E|}$ of all possible weight functions $w : E \to \mathbb{R}$, see [Bishop2006, §5.3, p. 241] for a very clear and simple exposition of this method. [21]

[21] According to [Bishop2006, p. 241] the term "backpropogation" is used in the neural computing literature to mean a variety of different things. Here we use the term "backpropagation" for (computing) the gradient of a sum of squares error function of the network weight function $w \in \mathbb{R}^{|E|}$.

• *Stochastic gradient descent* searches for a approximate minimizer of R_D^L using the following formula of "differentiation under integration"

$$\nabla_w R_D^L(h(w)) = \int_{\mathbb{R}^n \times \mathbb{R}^m} \nabla_w L(h(w), z)\, D(dz) \tag{10.7}$$

if L and $h(w)$ are differentiable.

There are several versions of SGD, see e.g. [SSBD2014, §14.4, p. 193, §20.6, p. 277], [Wolf2017]. We present here a simple version of SGD, following [SSBD2014, §14.5.1, p. 196]. The algorithm works as follows.
1) Choose any labeled pair $z \in \mathbb{R}^n \times \mathbb{R}^m$.
2) Set $w_1 = 0 \in \mathbb{R}^E$.
3) $w_{t+1} := w_t - \eta \nabla_w \tilde{L}(w_t, z)$.
4) Fix T, set the output $\bar{w}_T := \frac{1}{T}\sum_{t=1}^T w_t$.

Note that the output $\bar{w}_T$ depends on z and hence is a function of z, so we write $\bar{w}_T := \bar{w}_T(z)$.

Proposition 10.14. *(*[SSBD2014, Corollary 14.12, p. 197]*) Assume that $R_D^L(h(w))$ is a convex function in w (e.g., $\tilde{L}$ is a convex function in variable w). Assume that $B, \rho \in \mathbb{R}_+$ are given with the following properties.*

1) $w^* \in arg\min_{w:||w||\leq B} R_D^L\big(h(w)\big)$.

2) The SGD is run for T iterations with $\eta = \sqrt{\frac{B^2}{\rho^2 T}}$.

3) For all $t \in [1, T]$ we have $\mathbb{E}_{z\sim D}\big(||\nabla_w \tilde{L}(w_t, z)||\big) \leq \rho$ (e.g., $||\nabla_w \tilde{L}(w_t, z)|| \leq \rho$ for all z).

4) Assume that $T \geq \frac{B^2\rho^2}{\varepsilon^2}$.

Then

$$\mathbb{E}_{z\sim D}\Big(R_D^L\big(h(\bar{w}_T(z))\big)\Big) \leq R_D^L\big(h(w)\big) + \varepsilon. \tag{10.8}$$

Exercise 10.15. Find δ such that the following PAC-bound for the SGD algorithm in Proposition 10.14 holds

$$P_{z\sim D}[||R_D^L\Big(h\big(\bar{w}_T(z)\big) - R_D^L\big(h(w^*)\big)|| \leq 2\varepsilon] \geq 1 - \delta. \tag{10.9}$$

10.4. **Conclusion.** Neural networks have excellent expressive power. They can be trained by back-propagation and SGD. Under certain regularity condition the SGD works up to expectation by measure D (with a given accuracy ε), which implies a PAC-bound for the result of the algorithm after T rounds.

11. Amari's natural gradient stochastic descent learning

In the PAC learning framework, a learning algorithm for supervised learning associates to each training data S an approximate minimizer h_S of a risk function R_D^L on the set of hypothesis class $\mathcal{H}$, where L is the risk/error function and D is the unknown distribution of the labeled data $(x_i, y_i = h(x_i))$ where h is the true (or optimal) hypothesis. This is the discriminative approach in machine learning where our goal is not to learn the underlying

distribution D of an optimal predictor h_S but rather to learn h_S from the hypothesis class $\mathcal{H}$.

There is also a generative approach in statistical learning/and machine learning where we learn the correct joint distribution D of the labeled pairs (x_i, y_i) on a statistical model of possible joint distributions $p(x, y)$. The hypothesis class h generated by D is then regarded as a function of D. That is the Bayes principle we learned from the Bayes optimal predictor (Exercise 2.12) and from Theorem 2.13. Finding an optimal (correct) distribution D is the classical problem of density estimation or parameter estimation in statistics. We have discussed the part of this theory that concerns MSE and the Cramér-Rao inequality. Today we shall discuss generative approach in machine learning and gradient descent methods on statistical models, following [Amari2016, Chapter 12], and compare this approach with the discriminative approach by translating concepts in the discriminative approach to the ones in the generative approach.

11.1. **Parametrized statistical model for a hypothesis class of predictors.** We begin our discussion with assigning a statistical model to a hypothesis class of predictors, following [Amari2016] and [Bishop2006].

We consider a labeled training pair (x_i, y_i) as a noisy version of the desired pair $(x, f(x))$ where $f(x) \in \mathcal{Y}$ is a "response" to $x \in \mathcal{X}$. This is expressed as follows

$$y_i = f(x_i) + \varepsilon \tag{11.1}$$

where ε is a random noise. [22] Further we assume that the probability distributions D_ξ of $(x, f(x, \xi))$ are dominated measures on $\mathcal{X} \times \mathcal{Y}$ for a hypothesis class $\mathcal{H} := \{f(x, \xi)\}$ of functions parametrized by ξ. (As we know this is the case if the corresponding parametrized statistical model is finite dimensional.) Denoting the density of the distribution D by $p_\xi(x, y)$, we write (cf. Theorem 13.8)

$$p_\xi(x, y) = q(x) Prob_\xi(y|x) = q(x) p_\varepsilon\big(y - f(x, \xi)\big). \tag{11.2}$$

In this formula the second identity simplifies the form of the conditional density $Prob(y|x)$ saying that it is equal to probability density $p_\varepsilon(\varepsilon)$ of the noise ε. Here we assume that $\mathcal{Y}$ is a subset of a vector space. For general case we need to use the formalism of feature functions with value in a topological vector space developed in Section 3.

Example 11.1. We consider a neural network (V, E, σ, w) whose parameter is a function $w \in \mathbb{R}^{|E|}$ and the input-and output spaces are of the same dimension m. In the statistical model for (V, E, σ, w) the conditional probability $Prob(y|x)$ has parameter w and usually is assumed to have the

[22] Random noise, also called statistical noise, is is a term for recognized amounts of (unexplained) variation in a sample (and we explain it by probability measure).

following form [Bishop2006, (5.12), p. 233]

$$Prob(y|x,w) = \mathcal{N}\big(y|f(x,w),\beta^{-1}\big)$$

$$= \frac{|\beta|^2}{(2\pi)^{m/2}} \exp\big(-\frac{1}{2}\beta\langle y-f(x,w), y-f(x,w)\rangle\big) \tag{11.3}$$

where $f(x,w) : \mathbb{R}^m \to \mathbb{R}^m$ is the function represented by the neural network (V,E,σ,w) and $\beta \in \mathbb{R}_+$ is the precision (inverse variance) of the Gaussian noise. The definition of β depends on the "optimal" value of w, see [Bishop2006, p. 234]. The formula for $Prob(y|x,w)$ has the form in (11.2).

Exercise 11.2. Find statistical model for a neural networks (V,E,σ,w) whose parameter is a function $w \in \mathbb{R}^{|E|}$ and the input-and output spaces are of different dimension, using Example 11.1.

As we learned in Section 3, the translation of the notion "learning algorithm" in machine learning from a discriminative model $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ is the notion "estimator" $\hat{\sigma} : \Omega^m \to P$ in generative model $P \subset \mathcal{P}(\Omega = \mathcal{X} \times \mathcal{Y})$.

The next question we want to discuss is how to choose a risk function for a generative model/statistical model P. Previously, we considered the natural MSE function on the space of estimators $\hat{\sigma} : \Omega \to P$ and conclude that an estimator $\hat{\sigma}_0$ that turns the Crámer-Rao inequality to an equality is a minimizer of MSE function. Furthermore, if the minimizing estimator is unbiased then it is MLE.

Remark 11.3. The equation MLE for and estimator $\hat{\sigma}_m : (\mathcal{X} \times \mathcal{Y})^m \to P$ has the following form (see (4.2)

$$\frac{d}{dw}_{|w=\hat{\sigma}(z)} p(z,w) = 0. \tag{11.4}$$

Recall that $z = \{(x_1,y_1),\cdots,(x_m,y_m)\}$ is a sequence of i.i.d. labeled pairs and hence

$$\log p(z,w) = \sum_{i=1}^{m} \log p\big((x_i,y_i),w\big).$$

Thus, choosing an instantaneous loss function $l_m : (\mathcal{X} \times \mathcal{Y})^m \times P \to \mathbb{R}$ [23] to be the negative loglikehood function, i.e.,

$$l_m(z,w) := -\log\big(p(z,w)\big), \tag{11.5}$$

and compare it with (11.4), we conclude that the MLE equation for $\hat{\sigma}_m(z)$ is equivalent to the equation for the minimality of $l_m(z,w)_{|w=\hat{\sigma}_m(z)}$ regarding as a function on P of variable w. Since z is a (training) sample of size m, w minimizes the empirical loss/error function l_1 defined on $(\mathcal{X} \times \mathcal{Y}) \times P$ (cf. (2.9)).

[23] we use "l " for instantaneous loss function in generative models, to distinguish it with instantanenous loss function L in discriminative models.

Example 11.4. Let us consider the MLE equation for the statistical model (11.3) of the neural network in Example 11.1, using Remark 11.3. Given a sample $z = \{(x_1, y_1), \cdots, (x_N, y_N)\}$, the negative log-likelikehood function

$$l_m(z, w) = \frac{\beta}{2} \sum_{n=1}^{N} \big(f(x_n, w) - y_n\big)^2 - \frac{N}{2} \log \beta + \frac{N}{2} \log(2\pi). \tag{11.6}$$

Thus w is a minimizer of the quadratic empirical risk function $R_z^L(h_w)$ used in training neural networks that we considered in the previous lecture (here the usual notation S for "sample" is replaced by z). This fact confirms that our translation of concepts in discriminative models into concepts in generative models is correct/consistent.

11.2. **Online learning and batch learning.** Let us look at gradient descent methods for generative models from the algorithmic point of view. In computer science there are two main machine learning procedures: the online learning and the batch learning. In the online learning data becomes available in a sequential order and is used to update our best predictor for future data at each step, see below for more details. Batch learning techniques generate the best predictor by learning on the entire training data set at once.

Example 11.5. We consider the problem of parameter estimation ξ of a statistical model $P = \{p(z, \xi)\}$ with the instantaneous loss function $l = -\log p(z, \xi)$ whose risk function/generalization error is the function on P defined as follows

$$R^l_{\xi_0}(\xi) := \mathbb{E}_{\xi_0}\big(l(z, \xi)\big). \tag{11.7}$$

Here the true distribution ξ_0 we search for belongs to P (or lies not so far from P). We do not have separation between functions/hypotheses/predictors and their distributions as in discriminative aproach and this is an advantage of generative/statisticl model.

Now assume that P is a Riemannian manifold.

• *The online learning model of the gradient descent* algorithm in (10.6) to update our best estimator is defined as follows

$$\xi_{t+1} = \xi_t - \eta_t \nabla l(z_t, \xi_t). \tag{11.8}$$

Since training data are given one by one, the change

$$\triangle \xi_t = \eta_t \nabla l(z_t, \xi_t) \tag{11.9}$$

is a random variable depending on z_t. Under a regularity condition, the expectation $\mathbb{E}_{\xi_0}\big(\eta_t \nabla l(\xi_t)\big)$, which is $\eta_t \mathbb{E}_{\xi_0}\big(\triangle \xi_t\big)$, is equal to $-\eta_t \nabla R^l_{\xi_0}(\xi_t)$. Thus, under certain conditions, e.g., as in [Wolf2017, Theorem 2.12], the online learning algorithm (11.9) leads to a minimizer ξ of the generalization error $R^l_{\xi_0}(\xi)$.

• *The batch learning procedure for gradient descent* of the empirical/training loss function is defined as follows

$$\xi_{t+1} = \xi_t - \eta_t \frac{1}{T}\sum_{i=1}^{T} l(z_t, \xi_t). \tag{11.10}$$

Since a batch learning algorithm associates a sample S of size n an approximate (local) minimizer of the empirical error, the batch learning can be studied within framework of PAC-learning if the local minimizer is also a global minimizer, see e.g. [MRT2012, §7.4, p. 171]. Proposition 10.14 is an example of a successful batch learning algorithm in PAC-framework.

Exercise 11.6. (*) Show that Proposition 10.14 also holds, if we on the Step 3 of the algorithm replace z by arbitrary z_i and replace in the LHS of (10.8) the distribution $z \sim D$ by $(\{z_i\} \sim D^T)$.

Online learning is an important area with a rich literature in machine learning. In the remainder of this subsection we consider online learning from aspect of (stochastic) gradient descent and we compare online-learning with PAC-learning, following [MRT2012, Chapter 7, p. 147], [SSBD2014, Chapter 21, p. 287].

• Let $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ be a hypothesis class and A an online learning algorithm searching for an "optimal" hypothesis in $\mathcal{H}$.

11.2.1. *Setting of online-learning.* The general on-line learning setting involves T rounds. At the t-th round, $1 \leq t \leq T$, the algorithm A receives an instance $x_t \in \mathcal{X}$ and makes a prediction $A(x_t) \in \mathcal{Y}$. It then receives the true label $y_t \in \mathcal{Y}$ and computes a loss $L(A(y_t), y_t)$, where $L : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ is a loss function. The goal of A is to minimize the *cumulative loss*, which is an analogue of the notion of empirical risk in PAC-learning $R_A(T) := \sum_{i=1}^{T} L(\hat{y}_t, y_t)$ over T rounds [MRT2012, p. 148].

Remark 11.7. Formally speaking, the online learning model for the gradient descent algorithm in (10.6) satisfies the condition of minimizing the cumulative loss function only locally, if there are possible many critical points of the loss function l.

11.2.2. *Complexity of online-learning.* In PAC-learning we define the notion of sample complexity motivated by Problem 5.1. Similarly in online-learning we define the notion of mistake, motivated by the following

Question 11.8. *How many mistakes before we learn a particular concept?*

Definition 11.9 (Mistake Bounds, Online Learnability)**.** ([SSBD2014, Definition 21.1, p. 288]) Given any sequence $S = (x_1, h^*(y_1)), \cdots, (x_T, h^*(y_T))$, where T is any integer and $h^* \in \mathcal{H}$, let $M_A(S)$ be the number of mistakes A makes on the sequence S. We denote by $M_A(\mathcal{H})$ the supremum of $M_A(S)$ over all sequences of the above form. A bound of the form $M_A(\mathcal{H}) \leq B < \infty$

is called a *mistake bound*. We say that a hypothesis class $\mathcal{H}$ is *online learnable* if there exists an algorithm A for which $M_A(\mathcal{H}) \leq B < \infty$.

Remark 11.10. 1) In the online learning theory there is a notion of Littlestone's dimension of a hypothesis class $\mathcal{H}$, abbreviated as $L\dim(\mathcal{H})$, to measure the complexity of a hypothesis class $\mathcal{H}$ w.r.t. to online-learning. This notion is similar to the notion of VC-dimension in PAC-learning theory. One shows that the $L\dim$ is the maximum for $M_A(\mathcal{H})$ for any online learning algorithm A on $\mathcal{H}$ [SSBD2014, Corollary 21, p. 293]. In other words, a hypothesis class $\mathcal{H}$ is online-learnbal, iff $L\dim(\mathcal{H}) < \infty$.

2) In the online learning setting the notion of certainty and therefore the notion of probability measure are absent. In particular we do not have the notion of true risk. So one may wonder how it fits into the framework of statistical learning theory. In [MRT2012, §7.4, p. 171] the authors consider online learning on a hypothesis class $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ similar to the setting of online gradient in [Wolf2017, Theorem 2.12]. If the loss function L is bounded and convex w.r.t. to the first argument, using Azuma's inequality, the authors obtain PAC-bounds for stochastic gradient descent of batch type [MRT2012, Theorem 7.13, p. 172-173].

11.3. **Amari's natural gradient descent.** Amari's natural gradient descent is the gradient descent learning method, in online learning procedure or in batch learning procedure, using the Fisher metric on statistical models of hypothesis classes. It has been proposed by Amari in [Amari1967].

• *Advantage of Amari's natural gradient descent.* The natural gradient achieves invariance with respect to parameter re-encoding because the Fisher metric does not depend on parametrization of a statistical model.[24] In particular, learning become insensitive to the characteristic scale of each parameter direction.

• *Disadvantage of Amari's natural gradient descent.* Amari's method requires large computational cost for large-dimensional models: just storing the Fisher matrix already costs approximately $\dim P^2$. This cost is in the same order as the Newton second order iteration method. Furthermore the Fisher metric on a parameterized statistical model may have singularities because of ineffective parameterizations.

• *Newton second order iteration method and Amari's natural gradient descent.* Recall that the Newton iteration method for solving equation $f(x) = 0$, where $f \in C^1(\mathbb{R})$, is defined as follows

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \Longleftrightarrow f(x_n) + f'(x_n)(x_{n+1} - x_n) = 0.$$

Now assume that $n \geq 2$. Using the Taylor formula

$$df(x_{n+1}) = df(x_n) + Ddf(x_n)(x_{n+1} - x_n) + o(||x_{n+1} - x_n||)$$

[24]Moreover the Fisher metric is invariant under sufficient statistics that formalize the notion of data processing preserving information of a given statistical model [Le2016].

we define the Newton iteration method for finding a stationary point of $f \in C^2(\mathbb{R}^n)$ as follows

$$x_{n+1} = x_n - Ddf^{-1}(x_n)(df(x_n)). \tag{11.11}$$

Here $Ddf = \partial^2 f/\partial x_i \partial x_j$, the Hessian of f, is assumed to have maximal rank and hence invertible. Thus, computationally, the online gradient descent is similar to the Newton iteration method.

Problem 11.11. *The (online-)learnability/complexity of Amari's online/batch gradient descents has not been analysed and understood.*

Remark 11.12. (1) Formally Amari's stochastic gradient descent takes place on statistical models which e.g., is associated to a hypothesis class $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$. Its geometry is defined by the intrinsic Fisher metric on a statistical model. Generally speaking, a hypothesis $h \in \mathcal{H}$ must be considered as a function of its distribution. Thus Amari's stochastic gradient descent induces a stochastic gradient-like descent on $\mathcal{H}$. Amari's stochastic gradient descent may offer advantage in situation when it is easier to computer the gradient defined by the Fisher metric and we need the invariance of the gradient descent w.r.t. the change of parameter.

(2) A hypothesis class $\mathcal{H}_{V,E,\sigma}$ with parameter $w \in \mathbb{R}^E$ is another way of parametrization of functions $h \in \mathcal{H}_{V,E,\sigma}$. An invariant metric on $\mathbb{R}^E$ must be the pullback of a "natural metric" on the space of $\mathbb{R}^m$-valued functions on $\mathbb{R}^n$.

11.4. **Conclusion.** In this section we study Amari's gradient descent methods on statistical models which parameterise the probability distributions of hypothesis classes we wish to learn. This approach is more general, i.e., it also works for un-supervised learning, where we do not require labeled pair but want to maximize reward, e.g. in reinforcement learning [Amari2016, §12.1.5, p. 287]. We also translate concepts in discriminative models into the ones in generative models. The biggest advantage of the Amari natural gradient is its invariance under the change of parameters. Why the Amari natural gradient descent works has not been understood.

12. Deep learning and Bayesian neural networks

The latest important and dominating techniques in machine learning is deep learning. Deep learning is a set of techniques in machine learning with huge success [Schmidhuber2015] but also with problems [NYC2015], since most techniques are derived empirically without firm mathematical foundations.

In deep learning (also known as deep structured learning or hierarchical learning) learning can be supervised, partially supervised or unsupervised. Technically, deep machine learning utilizes neural networks or probabilistic

graphical models, which are generative models with graph structure like neural networks,[25] and uses gradient descents to find solutions of optimization problems, more precisely, to find (local) minimizers of a certain function on the underlying hypothesis class (resp. the underlying statistical model) that plays similar role as empirical- or generalization error in supervised learning.

Remark 12.1 (Representation learning in deep learning)**.** In [BCV2012] and [Schmidhuber2015] the authors emphasise that the success of machine learning algorithms generally depends on data representation, also called feature learning, cf. Remark 2.3. Hence, from algorithmic point of view we need unsupervised learning as a method of feature learning, and from mathematical point of view we need a data representation theory. As an example, the notion of a good representation of a class of generative models encompasses the notion of expressivness of the class in consideration. Namely we are interested the question: "Which probability distributions can be represented by a given class of generative models" (cf. Subsection 10.2). Probablistic graphic models are generative models with excellent expressive powers. We refer the reader to [Bishop2006, Chapter 8.1, p. 360] and [Montufar2012] for discussion of this question in deep.

There is also a trend in deep learning to add more probabilistic components to neural networks. Plain feedforward neural networks are prone to overfitting. When applied to supervised or reinforcement learning problems these networks are also often incapable of correctly assessing the uncertainty in the training data and so make overly confident decisions about the correct class, prediction or action. To address this problem Bayesian neural networks and the associated inference methods are suggested [Neal1996], [BCKW2015], see also [HTF2008, Chapter 11.9, p. 409].

12.1. **Probabilistic graphic models.** ([Bishop2006, WJ2008]).

• Let $G = (V, E)$ be a graph. *A clique C of G* is a fully connected subset of the vertex set V.

• For each vertex $s \in V$ we associate a random variable X_s taking value in some space X_s. We use lower-case letters (e.g., $x_s \in X_s$) to denote particular elements of X_s, so that the notation $\{X_s = x_s\}$ corresponds to the event that the random variable X_s takes the value $x_s \in X_s$.

• Given a *directed acyclic graph* $G = (V, E)$, for each vertex s and its parent set $\pi(s)$, i.e., the set of all vertex t such that $\vec{ts}$ is a directed edge, let $p_s(x_s|x_{\pi(s)})$ denote a nonnegative function over the variables $(x_s, x_{\pi(s)})$, normalized such that $p_s(x_s|x_{\pi(s)})\, dx_s = 1$.

• *A directed graphical model*, also called *a Bayesian network*, consists of a collection of probability distributions (densities or mass functions) that

[25](weighted) graphical representation is a classical way to represent a category of discret mathematical objects, where nodes encode the objects of the category and (weighted) edges encode morphisms of the category

factorize in the following way:

$$p(x_1, x_2, \cdots, x_m) = \Pi_{s\in V} p_s(x_s|x_{\pi(s)}).$$

• Given a *undirected graph* $G = (V, E)$, we associate with each clique C a *compatibility function* $\psi_C : (\otimes_{s\in C} X_s) \to \mathbb{R}_+$.

• *An undirected graphical model*, also known as a *Markov random field (MRF),* or *a Gibbs distribution*, is a collection of distributions that factorize as

$$p(x_1, \cdots, x_m) = \frac{1}{Z}\Pi_{C\in\mathcal{C}}\psi_C(x_C)$$

where $\mathcal{C}$ is (often) the set of all maximal cliques of G.

Remark 12.2. ([WJ2008, Chapter 2.4.1, p. 14]) Directed graphical models are good for Bayesian computations of quantities such as marginal likelihoods and posterior probabilities of parameters. Although Bayesian models can be represented using either directed or undirected graphs, it is the directed formalism that is most commonly encountered in practice.

12.2. **Bayesian neural networks.** ([BCKW2015])

We view a neural network as a probabilistic model $P(y|x, w)$: given an input $x \in \mathbb{R}^p$ a neural network assigns a probability to each possible output $y \in Y$, using the set of parameters or weights w, see also Example 11.1.

In Bayesian neural network weights w are represented by probability distributions over possible values, rather than having a single fixed value as is the norm. All weights in our neural networks are represented by probability distributions over possible values, rather than having a single fixed value as is the norm. Bayesian inference for neural networks calculates the pos- terior distribution of the weights given the training data S, $P(w|S)$.

Remark 12.3. A generalization property and success of a Bayesian learning algorithm has been analysed using PAC-formalism and Kullback-Leibler divergence in [SSBD2014, Chapter 32, p. 415].

12.3. **Conclusions.** At the moment we observe gradual deeper use of probabilistic methods in neural networks, which is also called a Bayesian approach, as well as the use computable and geometric statistical models that are similar to neural networks. Most methods in deep learning require sound mathematical foundations.

13. Appendix: Some basic notions in mathematical statistics

In this Appendix we assume that (Ω, Σ, μ) is a measure space and let $\mathcal{B}$ be a sub-σ-algebra in Σ.

13.1. **Dominating measures and the Radon-Nikodym theorem.** The following result concerning dominating measures, called the Radon-Nikodym theorem, is one of the key facts in measure theory.

Theorem 13.1. *(cf.* [Bogachev2007, Theorem 3.2.2, p. 177]*) Let μ and ν be two finite measures on a measurable space (Ω, Σ). The measure ν is dominated by the measure μ precisely when there exists a μ-integrable function f such that ν is given by*

$$\nu(A) = \int_A f \, d\mu \tag{13.1}$$

for each $A \in \Sigma$.

We denote ν by $f \cdot \mu$ for μ, ν, f satisfying the equation (13.1). The function f is called the (Radon-Nikodym) density (or the Radon-Nikodym derivate) of ν w.r.t. μ. The function f is denoted by $d\nu/d\mu$.

13.2. **Conditional expectation, (regular) conditional measure and joint distribution.** ([Bogachev2007, Chapter 10, p. 339], [Kallenberg1997, Chapter 5, p. 80], [Borovkov1998, §20, p. 106])

13.2.1. *Conditional expectation.*

Definition 13.2. 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 $\mathbb{E}_\mu^{\mathcal{B}} f$ such that

$$\int_\Omega gf \, d\mu = \int_\Omega g \mathbb{E}_\mu^{\mathcal{B}} f \, d\mu \tag{13.2}$$

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 13.3. 1) The defining equality (13.2) is equivalent to the following relationship obtained by the substitution $g = 1_{\mathcal{B}}$:

$$\int_\Omega f \, d\mu = \int_B \mathbb{E}_\mu^{\mathcal{B}} f \, d\mu \, \forall B \in \mathcal{B}. \tag{13.3}$$

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 one uses:

$$\mathbb{E}^{\mathcal{B}} f := \mathbb{E}_\mu^{\mathcal{B}} f.$$

In the probabilistic literature one uses the notation

$$\mathbb{E}(f|\mathcal{B}) := \mathbb{E}_\mu^{\mathcal{B}} f.$$

3) If Y is an integrable function on a probability space and $\mathcal{B}$ is generated by a measurable function (or mapping) $\eta : (\Omega, \Sigma) \to (\Omega', \Sigma')$ [26] then one uses the notation

$$\mathbb{E}(Y|\eta) := \mathbb{E}^{\sigma(\eta)} Y,$$

[26] One can take η to be the identity mapping, and $\Sigma' = \mathcal{B}$.

where $\sigma(\eta)$ is the sigma-algebra generated by η.

Theorem 13.4. *(*[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.

13.2.2. *Conditional measure. The condition measure* (or *conditional probability* in the case of probability measure) *of $A \in \Sigma$ w.r.t. $\mathcal{B}$*, is defined as follows

$$\mu(A|\mathcal{B}) := \mathbb{E}_\mu(1_A|\mathcal{B}). \tag{13.4}$$

In probabilistic literature one omits μ in (13.4) and writes instead

$$P(A|\mathcal{B}) := \mu(A|\mathcal{B}).$$

If $B = \xi^{-1}(\Sigma')$ where $\xi : (\Omega, \Sigma) \to (\Omega', \Sigma')$ is a measurable map, on uses the notation (cf. Remark 13.6.3)

$$P(A|\xi) := \mu(A|\xi) := \mu(A|\mathcal{B}).$$

13.2.3. *Regular conditional measure.*

Definition 13.5. (cf. [Kallenberg1997, (2), p. 84], [Bogachev2007, Definition 10.4.1, p. 357]) Suppose we are given a sub-sigma-algebra $\mathcal{B} \subset \Sigma$. A function

$$\mu^{\mathcal{B}}(.,.) : \Sigma \times \Omega \to \mathbb{R}$$

is called *a regular conditional measure* on Σ w.r.t. $\mathcal{B}$ if
1) for every $x \in \Omega$ the function $A \mapsto \mu^{\mathcal{B}}(A, x)$ is a measure on Σ,
(2) for every $A \in \Sigma$ the function $x \mapsto \mu^{\mathcal{B}}(A, x)$ is measurable w.r.t. $\mathcal{B}$ and μ-integrable,
(3) For all $A \in \Sigma$, $B \in \mathcal{B}$ the following formula for joint probability holds [27]

$$\mu(A \cap B) = \int_B \mu^{\mathcal{B}}(A, x) d\mu(x). \tag{13.5}$$

Remark 13.6. The function $\mu^{\mathcal{B}}(A, x)$ is called *a transitional measure* (in the first argument), or *conditional measure/conditional probability* (if no confusion arises), or *probability kernel*. [28]

[27] in literature one also uses $\mu(dx)$ for the notation $d\mu(x)$.

[28] in [Bogachev2007, p. 384] and [AJLS2015] the termonology "transitional measure" is used and in [Kallenberg1997, p. 84], [AJLS2016, AJLS2017] the terminology "probability kernel" is used.

(2) The relation between regular condition measures and conditional expectations (and hence conditional measures) is clarified in [Bogachev2007, Proposition 10.4.8, p. 367]) and motivated by the in [Bogachev2007, p. 356-357].

(3) The existence and uniqueness of regular conditional measure is proved under certain conditions [Bogachev2007, §10.4].

13.2.4. *(Regular) conditional probability and joint distribution.* In many practical statistical problems a joint distribution can be expressed via conditional density.

Definition 13.7. ([Borovkov1998, Definition 2, §20, p. 107]) Assume that a (regular) condition probability $P(B|y)$ for each $y \in \Omega$ is absolutely continuous w.r.t. some measure μ in Ω, i.e.,

$$P(\xi \in B|\eta = y) = \int_B f(x|y)d\mu(x). \tag{13.6}$$

Then the density $f(x|y)$ is called *the conditional density of ξ subject to the condition that $\eta = y$.*

Theorem 13.8. *(*[Borovkov1998, Theorem 2, §20, p. 108]*) If the joint distribution of ξ and η in $\Omega_1 \times \Omega_2$ has density function $f(x, y)$ w.r.t. the product of measures $\mu \in \mathcal{P}(\Omega_1)$ and $\lambda \in \mathcal{P}(\Omega_2)$ then the function*

$$f(x|y) := \frac{f(x,y)}{q(y)} \text{ where } q(y) = \int f(x,y)\mu(dx)$$

is the conditional density of ξ given $\eta = y$ and the function $q(y)$ is the density of η w.r.t. the measure λ.

References

[Amari1967] S. Amari, Theory of adaptive pattern classifiers, IEE transactions of Electronic Computers, 16, 299-307, 1967.

[Amari2016] S. Amari, Information Geometry, Springer, 2016.

[AJLS2016] N. Ay, J. Jost, H. V. Lê and L. Schwachhöfer, Parametrized measure models, arXiv:1510.07305.

[AJLS2015] N. Ay, J. Jost, H. V. Lê, and L. Schwachhöfer, Information geometry and sufficient statistics, Probability Theory and related Fields, 162 (2015), 327-364, arXiv:1207.6736.

[AJLS2017] N. Ay, J. Jost, H. V. Lê and L. Schwachhöfer, Information Geometry, Springer, 2017.

[BCV2012] Y. Bengio, A. Courville, and P. Vincent, Representation Learning: A Review and New Perspectives, arXic:1206.5538.

[BCKW2015] C. Blundell, J. Cornebise, K. Kavukcuoglu and D. Wierstra, Weight Uncertainty in Neural Networks, Proceedings of the 32 nd International Conference on Machine Learning, Lille, France, 2015. JMLR: W& CP volume 37, arXiv:1505.05424.

[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.

[CT2006] T. M. Cover and J. A. Thomas, Elements of Information theory, Wiley and Sons, second edition, 2006.

[DLWK2017] Wikipedia, Deep learning, https://en.wikipedia.org/wiki/Deep_learning.

[GB2010] X. Glorot and Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, In International Conference on Artificial Intelligence and Statistics, pages 249-256, 2010.

[JLS2017] J. Jost, H. V. Lê and L. Schwachhöfer, The Cramér-Rao inequality on singular statistical models, arXiv:1703.09403.

[Halmos1950] P.R. Halmos, Measure theory, Van Nostrand 1950.

[HOT2006] G. E. Hinton, S. Osindero, and Y. Te, A fast learning algorithm for deep belief nets, Neural Computation 18 (2006), 1527-1554.

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

[Hoeffding1963] W. Hoeffding, Probability inequalities for sums of bounded random variables, J. Am. Stat. Assoc., 58(301):13-30, 1963.

[IH1981] I. A. Ibragimov and R. Z. Has'minskii, Statistical Estimation: Asymptotic Theory, Springer, 1981.

[Janssen2003] A. Jansson, A nonparametric Cramér-Rao inequality, Statistics & Probability letters, 64(2003), 347-358.

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

[KV1994] M. J. Kearns and U. V. Vazirani, An Introduction to Computational Learning Theory, The MIT Press, 1994.

[Lang2002] S. Lang, Introduction to differentiable manifolds, Springer, 2002.

[Le2016] H.V. Lê, The uniqueness of the Fisher metric as information metric, AISM, 69 (2017), 879-896, arXiv:math/1306.1465.

[LJS2017] H. V. Lê, J. Jost and L. Schwachhöfer, The Cramér-Rao inequality on singular statistical models, Proceedings of GSI 2017, LNCS 10589, p. 552-560, Springer, 2017.

[LC1998] E. L. Lehmann and G. Casella, Theory of Point Estimation, Springer, 1998.

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

[Montufar2012] G. Montufar, On the Expressive Power of Discrete Mixture Models, Restricted Boltzmann Machines, and Deep Belief Networks - A Unified Mathematical Treatment, PhD Thesis, University Leipzig, 2012.

[Murphy2012] K. P. Murphy, Machine Learning: A Probabilistic Perspective, MIT Press, 2012.

[Neal1996] R. M. Neal, Bayesian Learning for Neural Networks, Springer, 1996.

[NYC2015] A. Nguyen, J. Yosinski, and J. Clune, Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images, in Computer Vision and Pattern Recognition (CVPR 15), IEEE, 2015, arXiv:1412.1897.

[Ollivier2015] Y. Ollivier, Riemannian metrics for neural networks I: feedforward networks. Information and Inference, 4(2):108-153, 2015.

[Ollivier2017] Y. Ollivier, Online Natural Gradient as a Kalman Filter, arXiv:1703.00209.

[PB2014] R. Pascanu and Y. Bengio, Rivisiting natural gradient for deep networks, arXiv:1301.3584.

[PS1995] G. Pistone and C. Sempi, An infinite-dimensional geometric structure on the space of all the probability measures equivalent to a given one, The Annals of Statistics 1995, Vol. 23, No. 5, 1543-1561.

[Pudlak2013] P. Pudlak, Logical Foundations of Mathematics and Computational Complexity, Springer 2013.

[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.

[Schmidhuber2015] J. Schmidhuber, Deep learning in neural networks: An overview, Neural Networks 61 (2015), 85-117.

[Valiant1984] L. Valiant, A theory of the learnable, Communications of the ACM, 27, 1984

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

[Vapnik2006] V. Vapnik, Estimation of Depedences Based on Empirical Data, Springer, 2006.

[Vorontsov2017] K. V. Vorontsov, Mathematical methods of supervised learnings (in Russian), http://www.ccas.ru/voron http://www.machinelearning.ru/wiki/

[WJ2008] M. J. Wainwright and M. I. Jordan, Graphical Models, Exponential Families, and Variational Inference, Foundations and Trends in Machine Learning Vol. 1, Nos. 1-2 (2008) 1-305.

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

[ZBHRV2016] C. Zhang, S. Bengio, M. Hardt, B. Recht, O. Vinyals, Understanding deep learning requires rethinking generalization, arXiv:1611.03530.