MATHEMATICAL FOUNDATIONS OF MACHINE LEARNING

(NMAG 469, FALL TERM 2018-2019)

HÔNG VÂN LÊ *

Contents

Date: February 1, 2019.

* 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 an interdisciplinary field in the intersection of mathematical statistics and computer sciences. Machine learning studies statistical models and algorithms for deriving predictors or meaningful patterns from empirical data. Machine learning techniques are applied in search engine, speech recognition and natural language processing, image detection, robotics etc.. In our course we address the following questions: What is the mathematical model of learning? How to quantify the difficulty/hardness/complexity of a learning problem? How to choose a learning algorithm? How to measure success of machine learning?

The syllabus of our course:

1. Supervised learning and unsupervised learning.
2. Generalization ability of machine learning.
3. Fisher metric and stochastic gradient descend.
4. Support vector machine, Kernel machine and Neural network.

Recommended Literature.

1. F. Cucker and S. Smale, On mathematical foundations of learning, Bulletin of AMS, 39(2001), 1-49.

2. K. P. Murphy, Machine learning: a probabilistic perspective (MIT press, 2012).

3. M. Sugiyama, Introduction to Statistical Machine Learning, Elsevier, 2016.

4. S. Shalev-Shwartz, and S. Ben-David, Understanding Machine Learning: From Theory to Algorithms, Cambridge University Press, 2014.

1. Learning, machine learning and artificial intelligence

Machine learning is the foundation of countless important applications including speech recognition, image detection, self-driving car and many thing more which I shall discuss today in my lecture. Machine learning techniques are developed using many mathematical theories. In my lecture course I shall explain the mathematical model of machine learning and how do we design a machine which shall learn successfully.

In my today lecture I shall discuss the following topics.

1. What are learning, inductive learning and machine learning.
2. History of machine learning and artificial intelligence.
3. Current tasks and main types of machine learning.
4. Basic questions in mathematical foundation of machine learning.

1.1. **Learning, inductive learning and machine learning.**

To start our discussion on machine learning let us begin first with the notion of learning. Every one from us know what is learning from our experiences at the very early age.

(a) Small children learn to speak by observing, repeating and mimicking adults' phrases. At the beginning their language is very simple and often erroneous. Gradually they speak freely with less and less mistakes. Their way of learning is *inductive learning*: from examples of words and phrases they learn the rules of combinations of these words and phrases into meaningful sentences.

(b) In school we learn mathematics, physics, biology, chemistry by following the instructions of our teachers and those in textbooks. We learn general rules and apply them to particular cases. This type of learning is *deductive learning*. Of course we also learn inductively in school by searching similar patterns in new problems and then apply the most appropriate methods possibly with modifications for solving the problem.

(c) Experimental physicists design experiments and observe the outcomes of the experiments to validate/support or dispute/refute a statement/conjecture on the nature of the observables. In other words experimental physicists learn about the dependence of certain features of the observables from empirical data which are outcomes of the experiments. This type of learning is *inductive learning*.

In mathematical theory of machine learning, or more general, in mathematical theory of learning we consider only *inductive learning*. Deductive learning is not very interesting; essentially it is equivalent to performing a set of computations using a finite set of rules and a knowledge database. Classical computer programs learn or gain some new information by deductive learning.

Let me suggest a definition of learning, that will be updated later to be more and more precise.

Definition 1.1. *A learning* is a process of gaining new knowledge, more precisely, new correlations of features of observable by examination of empirical data of the observable. Furthermore a learning is successful if the correlations can be tested in examination of new data and will be more precise with the increase of data.

The above definition is an expansion of Vapnik's mathematical postulation: "Learning is a problem of function estimation on the basis of empirical data".

Example 1.2. A classical example of learning is that of learning a physical law by curve fitting to data. In mathematical terms, a physical law is expressed by a function f, and data are the value y_i of f at observable points x_i. Usually we also know that (or assume that) the desired function belongs to a finite dimensional space. The goal of learning in this case is to estimate the unknown f from a set of pairs $(x_1, y_1), \cdots, (x_m, y_m)$. For instance, if f is assumed to be a polynomial of degree d over $\mathbb{R}$, then f belongs to a N-dimensional linear space $\mathbb{R}^N$, where $N = d + 1$, and to

estimate f is the same as to estimate the unknown coefficients $w_0, \cdots, w_d$ of monomial components in f, observing the data (x_i, y_i).

The most popular method of curve fitting is the least square method which quantifies *the error of the estimation* of the coefficients $(w_0, \cdots, w_d)$ in terms of the value

$$\sum_{i=1}^{m} (f_w(x_i) - y_i)^2 \text{ with } f_w(x) = \sum_{j=0}^{d} w_j x^j \tag{1.1}$$

which the desired function f should minimize. If the measurements generating the data (x_i, y_i) were exact, then $f(x_i)$ would be equal to y_i and the learning problem is an interpolation problem. But in general one expects the values y_i to be affected by noise.

The least square technique, going back to Gauss and Legendre [1], which is computational efficient and relies on numerical linear algebra, solves this minimization problem.

In the case of measurement noise, which is the reality according to quantum physics, we need to use the language of probability theory to model the noise and therefore to use tools of mathematical statistics in learning theory. That is why statistical learning theory is important part of machine learning theory.

1.2. **A brief history of machine learning.** Machine learning was born as a domain of artificial intelligence and it was reorganized as a separated field only in the 1990s. Below I recall several important events when the concept of machine learning has been discussed by famous mathematicians and computer scientists.

• In 1948 John von Neumann suggested that machine can do any thing that peoples are able to do.

• In 1950 Alan Turing asked "Can machines think?" in "Computing Machine and Intelligence" and proposed the famous Turing test. The Turing test is carried out as imitation game. On one side of a computer screen sits a human judge, whose job is to chat to an unknown gamer on the other side. Most of those gamers will be humans; one will be a chatbot with the purpose of tricking the judge into thinking that it is the real human.

• In 1956 John McCarthy coined the term "artificial intelligence".

• In 1959, Arthur Samuel, the American pioneer in the field of computer gaming and artificial intelligence, defined machine learning as a field of study that gives computers the ability to learn without being explicitly programmed. 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]The least-squares method is usually credited to Carl Friedrich Gauss (1809), but it was first published by Adrien-Marie Legendre (1805)

In the early days of AI, statistical and probabilistic methods were employed. Perceptrons which are simple models used in statistics were used for classification problems in machine learning. Perceptrons were later developed into more complicated neural networks. Because of many theoretical problems and because of small capacity of hardware memory and slow speed of computers statistical methods were out of favour. By 1980, expert systems, which were based on knowledge database, and inductive logic programming had come to dominate AI. Neural networks returned back to machine learning with success in the mid-1980s with the reinvention of a new algorithm and thanks to increasing speed of computers and increasing hardware memory.

Machine learning, reorganized as a separate field, started to flourish in the 1990s. The current trend is benefited from Internet.

In the book by Russel and Norvig "Artificial Intelligence a modern Approach" (2010) AI encompass the following domains:
- natural language processing,
- knowledge representation,
- automated reasoning to use the stored information to answer questions and to draw new conclusions;
- machine learning to adapt to new circumstances and to detect and extrapolate patterns,
- computer vision to perceive objects,
- robotics.

All the listed above domains of artificial intelligence except knowledge representation and robotics are now considered domains of machine learning. Pattern detection and recognition were and are still considered to be domain of data mining but they become more and more part of machine learning. Thus AI = knowledge representation + ML + robotics.

• representation learning, a new word for knowledge representation but with a different flavor, is a part of machine learning.

• Robotics = ML + hardware.

Why did such a move from artificial intelligence to machine learning happen?

The answer is that we are able to formalize most concepts and model problems of artificial intelligence using mathematical language and represent as well as unify them in such a way that we can apply mathematical methods to solve many problems in terms of algorithms that machine are able to perform.

As a final remark on the history of machine learning I would like to note that data science, much hyped in 2018, has the same goal as machine learning: Data science seeks actionable and consistent pattern for predictive uses. [2].

[2]according to Dhar, V. (2013). "Data science and prediction". Communications of the ACM. 56 (12): 64. doi:10.1145/2500499, see also wiki site on data science

1.3. **Current tasks and types of machine learning.** Now I shall describe what current machine learning can perform and how they do it.

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

Classification task assigns a "category" [3] to each item. In mathematical language, a category is an element in a countable set. 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 can be 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 of categories*.

As we have remarked in the classical example of learning (Example 1.2), usually we have ambiguous/incorrect measurement and we have to add a "noise" to our measurement. If every thing would be exact, the classification task is the classical interpolation function problem in mathematics.

Regression task predicts a real value, i.e., a value in $\mathbb{R}$, for each item. Examples of regression tasks include learning physical law by curve fitting to data (Example 1.2) with application to predictions of stock values or variations of economic variables. In this problem, the error of the prediction, which is also called estimation in Example 1.2, 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 $\mathbb{R}$. As in the classification task, in regression problems we also need to take into account a "noise" from incorrect measurement for the regression problem. [4]

Density estimation task finds the distribution of inputs in some space. Over one hundred year ago Karl Pearson (1980-1962), the founder of the modern statistics, [5] proposed that all observations come from some probability distribution and the purpose of sciences is to estimate the parameter

[3]the term "category" used in machine learning has another meaning than the term "category" in mathematics. In what follows we use the term "category" accepted in ML community without bracket.

[4]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's method of investigation is non-standard at that time: first he collected the data, then he guessed the relationship model of the events.

[5] He founded the world's first university statistics department at University College London in 1911, the Biometrical Society and Biometrika, the first journal of mathematical statistics and biometry.

of these distributions. A particular case of parameter estimation is density estimation problem. Density estimation problem has been proposed by Ronald Fisher (1980-1962), the father of modern statistics and experiment designs, [6] as a key element of his simplification of statistical theory, namely he assumed the existence of a density function $p(\xi)$ that governs the randomness (the noise) of a problem of interest.

Digression. The measure ν is called *dominated by* μ (or *absolutely continuous with respect to* μ), if $\nu(A) = 0$ for every set A with $\mu(A) = 0$. Notation: $\nu << \mu$. By Radon-Nykodym theorem, see Appendix, Subsection A.1, we can write

$$\nu = f \cdot \mu$$

and f is *the density function of* ν *w.r.t.* μ.

For example, the Gaussian distribution on the real line is dominated by the canonical measure dx and we express the standard normal distribution in terms of its density

$$f(x) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}x^2).$$

The classical problem of density estimation is formulated as follows. Let a statistical model A be a class of densities subjected to a given dominant measure. Let the unknown density $p(x, \xi)$ we need to estimate belong to the statistical model A, which is parameterized by ξ. The problem is to estimate the parameter ξ of $p(x, \xi)$ using i.i.d. data $X_1, \cdots, X_l$ distributed according to this unknown density.

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. Clustering is one of the most widely used techniques for exploratory data analysis. In all disciplines, from social sciences to biology to computer science, people try to get a first intuition about their data by identifying meaningful groups among the data points. For example, computational biologists cluster genes on the basis of similarities in their expression in different experiments; retailers cluster customers, on the basis of their customer profiles, for the purpose of targeted marketing; and astronomers cluster stars on the basis of their spacial proximity.

[6] Fisher introduced the main models of statistical inference in the unified framework of parametric statistics. He described different problems of estimating functions from given data (the problems of discriminant analysis, regression analysis, and density estimation) as the problems of parameter estimation of specific (parametric) models and suggested the maximum likelihood method for estimating the unknown parameters in all these models.

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. Many of dimensional reduction techniques are linear. When the technique is non-linear we speak about manifold learning technique. We can regard clustering as dimension reduction too.

1.3.2. *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*, i.e., the data available to the learner before making decision and prediction, the outcomes *and the test data* that are used to evaluate and apply the learning algorithm.

Main types of machine learning are supervised, unsupervised and reinforcement.

• In *supervised learning a learning machine* is a device that receives *labeled training data*, i.e., the pair of a known instance and its feature, also called label. Examples 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. We also interpret a learning machine in supervised learning as a student who gives his supervisor a known instance and the supervisor answers with the known feature.

• In *unsupervised learning* there is *no additional label* attached to the data and *the task is to describe structure* of data. Since the examples (the available data) given to the learning algorithm are unlabeled, there is no straightforward way to evaluate the accuracy of the structure that is produced by the algorithm. Density estimation, clustering and dimensionality reduction are examples of unsupervised learning problems. Most important applications of unsupervised learning are 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. Regarding a learning machine in unsupervised learning as a student, then the student has to learn by himself without teacher. This learning is harder but happens more often in life. At the current time, except few tasks, which I shall consider in the next lecture, unsupervised learning is primarily *descriptive* and experimental whereas supervised learning is more *predictive* (and has deeper theoretical foundation).

• *Reinforcement learning* is the type of machine learning where a learner actively interacts with the environment to achieve a certain goal. More precisely, the learner 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. For examples, reinforcement learning is used in self-driving car. Reinforcement learning is aimed at acquiring the generalization ability in the same way as supervised learning, but the supervisor does not directly give answers to the students questions. Instead, the supervisor evaluates the students behavior and gives feedback about it.

1.4. **Basic questions in mathematical foundations of machine learning.** Let me recall that a learning is a process of gaining knowledge on a feature of observables by examination of partially available data. The learning is successful if we can make a prediction on unseen data, which improves when we have more data. For example, in classification problem, the learning machine has to predict the category of a new item from a specific set, after seeing a lot of labeled data consisting of items and their categories. The classification task is a typical task in supervised learning where we can explain how and why a learning machine works and how and why machine learns successfully. Mathematical foundations of machine learning aim to answer these questions in mathematical language.

Question 1.3. *What is the mathematical model of learning?*

To answer Question 1.3 we need to specify our definition of learning in a mathematical language which can be used to build instructions for machines.

Question 1.4. *How to quantify the difficulty/complexity of a learning problem?*

We quantify the difficulty of a problem in terms of its time complexity, which is the minimum time needed for performing computer program to solve a problem, and in term of its resource complexity which measure the capacity of data storage and energy resource needed to solve the problem. If the complexity of a problem is very large then we cannot not learn it. So Question 1.4 contains the sub-question " why can we learn a problem?"

Question 1.5. *How to choose a learning algorithm?*

Clearly we want to have a best learning algorithm, once we know a model of a machine learning which specifies the set of possible predictors (decisions) and the associated error/reward function.

By Definition 1.1, a learning process is successful, if its prediction/estimation improves with the increase of data. Thus the notion of success of learning process requires a mathematical treatment of asymptotic rate of error/reward in the presence of complexity of the problem.

Question 1.6. *Is there a mathematical theory underlying intelligence?*

I shall discuss this speculative question in the last lecture.

1.5. **Conclusion.** Machine learning is automatized learning, whose performance is improves with increasing volume of empirical data. Machine learning uses mathematical statistics to model incomplete information and the random nature of the observed data. Machine learning is the core part of artificial intelligence. Machine learning is very successful experimentally and there are many open questions concerning its mathematical foundations. Mathematical foundations of machine learning is necessary for building general purpose artificial intelligence, also called Artificial General Intelligence (AGI), or Universal Artificial Intelligence (UAI). The importance of mathematical foundations for AGI shall be clarified in the third lecture.

Finally I recommend some sources for further reading.

- F. Cucker and S. Smale, On mathematical foundations of learning, Bulletin of AMS, 39(2001), 1-49.
- B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman, Building machines that learn and think like people. Behavioral and Brain Sciences,(2016) 24:1-101, arXiv:1604.00289.
- S. J. Russell and P. Norvig, Artificial Intelligence A Modern Approach, Prentice Hall, 2010.

2. Statistical models and frameworks for supervised learning

Last week we discussed the concept of learning and examined several examples. Today I shall specify the concept of learning by presenting basic mathematical models of supervised learning.

A model is simply a compact representation of possible data one could observe. Modeling is central to the sciences. Models allow one to make predictions, to understand phenomena, and to quantify, compare and falsify hypotheses. A model for machine learning must be able to make predictions and improves their ability to make predictions in light of new data.

The model of supervised learning I present today is based on Vapnik's statistical learning theory, which starts from the following concise concept of learning.

Definition 2.1. ([Vapnik2000, p. 17]) Learning is a problem of function estimation on the basis of empirical data.

There are two main model types for machine learning: discriminative models and generative models. They are distinguished by the type of functions we want to estimate for understanding the feature of observable.

2.1. **Discriminative model of supervised learning.** Let us consider a toy example of a classification task, which like regression tasks (Example 1.2), is a typical example of supervised learning.

Example 2.2 (Toy example)**.** 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 a developer of algorithm in ML will be periodically re-opened and therefore they *want to design a ML program to predict* the potential of applicants such that the program *automatically will be improved with time*.

A discriminative model of supervised learning consists of the following components.

• *A domain set* $\mathcal{X}$ (also called *an input space*) consists of elements, whose features we like to learn. Elements $x \in \mathcal{X}$ are called *random inputs (or random instances)* [7] which are distributed by an unknown probability measure $\mu_{\mathcal{X}}$. In other words, the probability that x belongs to a subset $A \subset \mathcal{X}$ is $\mu_{\mathcal{X}}(A)$. The probability distribution $\mu_{\mathcal{X}}$ models our incomplete information about elements $x \in \mathcal{X}$. In general we don't know the distribution $\mu_{\mathcal{X}}$.

(In the toy example of a ML firm the domain set $\mathcal{X}$ is the set of all applicants, more precisely, their representing features: the marks in math, in CS, and in communication test. Hence $\mathcal{X} = [1, 20] \times [1, 20] \times [1, 5]$. In the regression example of learning a physical law (Example 1.2) the domain set $\mathcal{X}$ is the set of all polynomials of degree at most d, hence $\mathcal{X}$ is identified with $\mathbb{R}^d$.)

• *An output space* $\mathcal{Y}$, also called *a label set*, consists of possible features (also called labels) y of inputs $x \in \mathcal{X}$. We are interested in finding a predictor/mapping $h : \mathcal{X} \to \mathcal{Y}$ such that a feature of x is $h(x)$. If such a mapping h exists and is measurable, the feature $h(x)$ is distributed by the measure $h_*(\mu_{\mathcal{X}})$. In general such a function does not exist, and we assume that there exists only a probability measure $\mu_{\mathcal{X}\times\mathcal{Y}}$ on the space $(\mathcal{X} \times \mathcal{Y})$ that defines the probability that y is a feature of x, i.e., the probability of $(x, y) \in A \subset \mathcal{X} \times \mathcal{Y}$ being a labeled pair is equal to $\mu_{\mathcal{X}\times\mathcal{Y}}(A)$. In general we don't know $\mu_{\mathcal{X}\times\mathcal{Y}}$.

(In the toy example the label set $\mathcal{Y} = [1, 10]$ is the set of all possible potentials scaled from 1 to 10. In the example of learning a physical law (Example 1.2) the label set is the set $\mathbb{R}$ of all possible value of $f(x)$.)

• *A training data* is a sequence $S = \{(x_1, y_1), \cdots, (x_n, y_n)\} \in (\mathcal{X} \times \mathcal{Y})^n$ of observed labeled pairs, which are usually assumed to be i.i.d. (independently

[7] classically, elements of $\mathcal{X}$ are considered as (values of) *random variables*, where the word "variable" means "unknown". When $\mathcal{X}$ is an input space (resp. an output space) its elements are also called independent (resp. dependent) variables. Since nowadays the word variable has a different meaning, like [Ghahramani2013, p. 4], I would avoid "random variable" in this situation. Some authors, e.g. [Billingsley1999, p.24] use the terminology "random elements" for measurable mappings.

identically distributed). In this case S is distributed by the product measure $\mu^n_{\mathcal{X}\times\mathcal{Y}}$ on $(\mathcal{X}\times\mathcal{Y})^n$. The number n is called *the size of* S. S is thought as given by a "supervisor".

• A *hypothesis space* $\mathcal{H}\subset\mathcal{Y}^{\mathcal{X}}$ of possible predictors $h:\mathcal{X}\to\mathcal{Y}$.
(In Example 2.2 we may wish to choose
$\mathcal{H}:=\{h:\mathcal{X}\to\mathcal{Y}|\, h(x,y,z)=ax+by+cz$ for some $a,b,c\in\mathbb{Z}_{\geq 0}\}$
and in Example 1.2 we choose
$\mathcal{H}:=\{h:\mathbb{R}\to\mathbb{R}|\ h$ is a polynomial of degree at most $d\ \}\cong\mathbb{R}^{d+1}$
to simplify our search for a best prediction.)

• *The aim of a learner* is to find *a best prediction rule* A that assigns a training data S to a prediction $h_S\in\mathcal{H}$. In other words the learner needs to find a rule, more precisely, *an algorithm*

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

such that $h_S(x)$ predicts the label of (unseen) instance x with the less error.

• *The error function*, also called *a risk function*, measures the discrepancy between a hypothesis $h\in\mathcal{H}$ and an ideal predictor. The error function is a central notion in learning theory. This function should be defined as the averaged discrepancy of $h(x)$ and y, where (x,y) runs over $\mathcal{X}\times\mathcal{Y}$. The averaging is calculated using the probability measure $\mu:=\mu_{\mathcal{X}\times\mathcal{Y}}$ that governs the distribution of labeled pair (x,y). Thus a risk function R must depend on μ, so we denote it by R_μ. It is accepted that the risk function $R_\mu:\mathcal{H}\to\mathbb{R}$ is defined as follows.

$$R^L_\mu(h):=\int_{\mathcal{X}\times\mathcal{Y}}L(x,y,h)\,d\mu \tag{2.2}$$

where $L:\mathcal{X}\times\mathcal{Y}\times\mathcal{H}\to\mathbb{R}$ is *an instantaneous loss function* that measures the discrepancy between the value of a prediction/hypothesis h at x and the possible feature y:

$$L(x,y,h):=d(y,h(x)). \tag{2.3}$$

Here $d:\mathcal{Y}\times\mathcal{Y}$ is a non-negative function that vanishes at the diagonal $\{(y,y)|\, y\in\mathcal{Y}\}$ of $\mathcal{Y}\times\mathcal{Y}$. For example $d(y,y')=|y-y'|^2$. By taking averaging over $(\mathcal{X}\times\mathcal{Y})$ using μ, we effectively count only the points (x,y) which are correlated as labeled pairs.

Note the expected risk function is well defined on $\mathcal{H}$ only if $L(x,y,h)\in L^1(\mathcal{X}\times\mathcal{Y},\mu)$ for all $h\in\mathcal{H}$.

• The main question of learning theory is to find necessary and sufficient conditions for the existence of a prediction rule A in (2.1) such that the error of h_S converges to the error of an ideal predictor, or more precisely, to the infimum of the error of h over $h\in\mathcal{H}$, and then to construct such A.

Remark 2.3. (1) In our discriminative model of supervised learning we model the random nature of training data $S \in (\mathcal{X} \times \mathcal{Y})^n$ via a probability measure μ_n on $(\mathcal{X} \times \mathcal{Y})^n$, where $\mu_n = \mu_1^n$ is the training data are i.i.d.. We don't need a probability measure on $\mathcal{X}$ to model the random nature of $x \in \mathcal{X}$. The main difficulty in search for the best prediction rule A is that we don't know μ^n, we know only training data S distributed by μ^n.

(2) Note the expected risk function is well defined on $\mathcal{H}$ only if $L(x, y, h) \in L^1(\mathcal{X} \times \mathcal{Y}, \mu)$ for all $h \in \mathcal{H}$. Since we don't know μ, we should assume that $L \in L^1(\mathcal{X} \times \mathcal{Y}, \nu)$ for any $\nu \in \mathcal{P}_0$, where $\mathcal{P}_0$ is a family of probability measures on $\mathcal{X} \times \mathcal{Y}$ that contains the unknown μ.

(3) The quasi-distance function $d : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ induces a quasi-distance function $d^n : \mathcal{Y}^n \times \mathcal{Y}^n \to \mathbb{R}$ as follows

$$d^n([y_1, \cdots, y_n], [y'_1, \cdots, y'_n]) = \sum_{i=1}^{n} d(y_i, y'_i), \tag{2.4}$$

and therefore it induces the expected loss function $R_{\mu^n}^{L(d^n)} : \mathcal{H} \to \mathbb{R}$ as follows

$$\begin{aligned} R_{\mu^n}^{L(d^n)}(h) &= \int_{(\mathcal{X}\times\mathcal{Y})^n} d^n([y_1, \cdots, y_n], [h(x_1), \cdots, h(x_n)]) d\mu^n \\ &= n \int_{\mathcal{X}\times\mathcal{Y}} L(x, y, h) d\mu. \end{aligned} \tag{2.5}$$

Thus it suffices to consider only $R_\mu(h)$, if S is a sequence of i.i.d. observables.

(4) Now we show that the classical case of learning a physical law by fitting to data, assuming exact measurement, is a "classical limit" of our discriminative model of supervised learning. In the classical learning problem, since we know the exact position $S := \{(x_1, y_1), \cdots, (x_n, y_n)\} \in (\mathcal{X} \times \mathcal{Y})^n$, we assign the Dirac (probability measure) $\mu_S := \delta_{x_1,y_1} \times \cdots \times \delta_{x_n,y_n}$ to the space $(\mathcal{X} \times \mathcal{Y})^n$ [8]. Now let $d(y, y') = |y - y'|^2$, it is not hard to see that

$$R_{\mu_S}^{L(d^n)}(h) = \sum_{i=1}^{n} |h(x_i) - y_i|^2 \tag{2.6}$$

coincides with the error of estimation in (1.1).

Example 2.4 (0-1 loss)**.** Let us take $\mathcal{H} = \mathcal{Y}^{\mathcal{X}}$ - the subset of all mapping $\mathcal{X} \to \mathcal{Y}$. The *0-1 instantaneous loss function* $L : \mathcal{X} \times \mathcal{Y} \times \mathcal{H} \to \{0, 1\}$ is defined as follows: $L(x, y, h) := d(y, h(x)) = 1 - \delta^y_{h(x)}$. The corresponding expected 0-1 loss determines the probability of the answer $h(x)$ that does not correlate with x:

$$R^{(0-1)}_{\mu_{\mathcal{X}\times\mathcal{Y}}}(h) = \mu_{\mathcal{X}\times\mathcal{Y}}\{(x, y) \in \mathcal{X} \times \mathcal{Y} |\, h(x) \neq y\} = 1 - \mu_{\mathcal{X}\times\mathcal{Y}}(\{x, h(x)\}). \tag{2.7}$$

Example 2.5. Assume that $x \in \mathcal{X}$ is distributed by a probability measure $\mu_{\mathcal{X}}$ and its feature y is defined by $y = h(x)$ where $h : \mathcal{X} \to \mathcal{Y}$ is a measurable

[8] the probability that A contains S is $\delta_S(A)$

mapping. Denote by $\Gamma_h : \mathcal{X} \to \mathcal{X} \times \mathcal{Y}$, $x \mapsto (x, y)$, the graph of h. Then (x, y) is distributed by the push-forward measure $\mu_h := (\Gamma_h)_*(\mu_{\mathcal{X}})$, where

$$(\Gamma_h)_*\mu_{\mathcal{X}}(A) = \mu_{\mathcal{X}}\big(\Gamma_h^{-1}(A)\big) = \mu_{\mathcal{X}}\Big(\Gamma_h^{-1}(A \cap \Gamma_h(\mathcal{X}))\Big). \tag{2.8}$$

Let us compute the expected 0-1 loss function for a mapping $f \in \mathcal{H} = \mathcal{Y}^{\mathcal{X}}$ w.r.t. the measure μ_h. By (2.7) and by (2.8) we have

$$R^{(0-1)}_{\mu_h}(f) = 1 - \mu_{\mathcal{X}}(x|f(x) = h(x)). \tag{2.9}$$

Hence $R^{(0-1)}_{\mu_h}(f) = 0$ iff $f = h$ $\mu_{\mathcal{X}}$-a. e..

2.2. **Generative model of supervised learning.** In many cases a discriminative model of supervised learning may not yield a successful learning algorithm because the hypothesis space $\mathcal{H}$ is too small and cannot approximate a desired prediction for a feature $\in \mathcal{Y}$ of instance $x \in \mathcal{X}$ with a satisfying accuracy, i.e., *the optimal performance error of the class* $\mathcal{H}$

$$R^L_{\mu,\mathcal{H}} := \inf_{h \in \mathcal{H}} R^L_{\mu}(h) \tag{2.10}$$

that represents the optimal performance of a learner using $\mathcal{H}$ is quite large.

One of possible reasons of this failure is that, a feature $y \in Y$ of x cannot be accurately approximated (using an instantaneous loss function L) by any function $h : \mathcal{X} \to \mathcal{Y}$.

In general case we may wish to *estimate the probability that* $y \in \mathcal{Y}$ *is a feature of* x. This is expressed in term of the conditional probability $P(y \in B|x)$ - the probability that a feature y of $x \in \mathcal{X}$ belongs to $B \subset \Sigma_{\mathcal{Y}}$.

Digression. *Conditional probability* is one of most basic concepts in probability theory. In general we always have a prior information before taking decision, e.g. before estimating the probability of a future event. Conditional probability $P(A|B)$ formalizes the probability of an event A given the knowledge that event B happens. Here we assume that A, B are elements of the sigma-algebra $\Sigma_{\mathcal{X}}$ of a measurable space $(\mathcal{X}, \Sigma_{\mathcal{X}})$. If $\mathcal{X}$ is countable, the concept of conditional probability can be defined straightforward:

$$P(A|B) := \frac{P(A \cap B)}{P(B)}. \tag{2.11}$$

It is not hard to see that, given B, the conditional probability $P(\cdot|B)$ defined in (2.11) is a probability measure on $\mathcal{X}$, $A \mapsto P(A|B)$, which is called *the conditional probability measure given* B. In its turn, by taking integration over $\mathcal{X}$ using the conditional probability $P(\cdot|B)$, we obtain the notion of *conditional expectation*, given B, which shall be denoted by $\mathbb{E}_{P(\cdot|B)}$Therefore the conditional expectation given B is a function on $\Sigma_{\mathcal{X}}$.

In general case when $\mathcal{X}$ is not countable the definition of conditional probability is more subtle, especially when we have to define $P(A|B)$, where B has null-measure. A typical situation is the case $B = h^{-1}(z_0)$, where

$h : \mathcal{X} \to \mathcal{Z}$ is a random variable (a measurable mapping). To treat this important case we need to define first the notion of conditional expectation, see Subsection A.2 in Appendix. What is important for our applications in many case is the notion of conditional distribution $P(A|h(x) = z_0)$, which can be expressed by a function on $\mathcal{Z}$ moreover we also require that $P(\cdot|h(x) = z_0)$ is dominated by a measure $\mu_{\mathcal{X}}$ for all $z_0 \in \mathcal{Z}$, i.e., there exists a density function $f(x|z_0)$ on $\mathcal{X}$ such that by (A.5) we have

$$P(A|h(x) = z_0) = \int_A f(x|z_0)\mu_{\mathcal{X}}.$$

We may also wish to estimate the joint distribution $\mu := \mu_{\mathcal{X}\times\mathcal{Y}}$ of i.i.d. labeled pairs (x, y). By Formula (A.7) the joint distribution $\mu_{\mathcal{X}\times\mathcal{Y}}$ can be recovered from conditional probability $\mu(y|x)$, see also Subsection A.3. Once we know μ we know the expected risk R^L_μ for an instantaneous loss function L, and hence a minimizing sequence $\{h_i \in \mathcal{H}\}$ of R^L_μ

$$\lim_{n\to\infty} R^L_\mu(h_i) = R^L_{\mu,\mathcal{H}}$$

can be determined. In many cases we can find an explicit formula for the Bayes optimal predictor that minimizes the expected risk value R^L_μ, once μ is known.

Exercise 2.6 (The Bayes Optimal Predictor)**.** ([SSBD2014, p. 46]) If $\mathcal{Y} = \mathbb{Z}_2$ there is an explicit formula for a Bayes classifier, called the Bayes optimal predictor. Given any probability distribution D over $\mathcal{X} \times \{0, 1\}$, the best label predicting function from $\mathcal{X}$ to $\{0, 1\}$ will be

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

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

Exercise 2.7 (Regression optimal Bayesian estimator)**.** In regression problem the output space $\mathcal{Y}$ is $\mathbb{R}$. Let us define the following embedding

$$i_1 : \mathbb{R}^{\mathcal{X}} \to \mathbb{R}^{\mathcal{X}\times\mathcal{Y}} : [i_1(f)](x, y) := f(x),$$
$$i_2 : \mathbb{R}^{\mathcal{Y}} \to \mathbb{R}^{\mathcal{X}\times\mathcal{Y}} : [i_2(f)](x, y) := f(y).$$

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

$$L^2(\mathcal{X}, (\Pi_{\mathcal{X}})_*\mu) = \{f \in \mathbb{R}^{\mathcal{X}} |\, i_1(f) \in L^2(\mathcal{X} \times \mathbb{R}, \mu)\},$$
$$L^2(\mathbb{R}, (\Pi_{\mathbb{R}})_*\mu) = \{f \in \mathbb{R}^{\mathbb{R}} |\, i_2(f) \in L^2(\mathcal{X} \times \mathbb{R}, \mu)\}.$$

Now we let $\mathcal{F} := L^2(\mathcal{X}, \Pi_*(\mu))$. Let Y denote the function on $\mathbb{R}$ such that $Y(y) = y$. Assume that $Y \in L^2(\mathbb{R}, (\Pi_{\mathbb{R}})_*\mu)$ and define the quadratic loss function $L : \mathcal{X} \times \mathcal{Y} \times \mathcal{F} \to \mathbb{R}$

$$L(x, y, h) := |y - h(x)|^2, \tag{2.12}$$

$$R_\mu^L(h) = \mathbb{E}_\mu\big(|Y(y) - h(x)|^2\big) = |i_2(Y) - i_1(h)|^2_{L^2(\mathcal{X}\times\mathbb{R},\mu)}. \tag{2.13}$$

The expected risk R_μ^L is called the L_2-*risk*, also known as *mean squared error* (MSE). Show that the *regression function* $r(x) := \mathbb{E}_\mu\big(i_2(Y)|X = x\big)$ belongs to $\mathcal{F}$ and minimizes the $L_2(\mu)$-risk.

Definition 2.8. A model of supervised learning with the aim to estimate the conditional distribution $P(y \in B|x)$, in particular, a conditional density function $p(y|x)$, or joint distribution of (x, y) is called *a generative model of supervised learning*.

Remark 2.9. Generative models give us more complete information of the correlation between a feature y and an instance x but they are more complicated, since even in the regular case, a conditional density function is a function of two variables x and y and we cannot express this correlation as a dependence of y from x. In fact, we could interpret a density function $p(y|x)$ as a probabilistic mapping from $\mathcal{X}$ to $\mathcal{Y}$: $p(y|x)$ indicates the probability that the value of a mapping in consideration at x is equal to y. In many practical cases, following Fisher suggestion, [Vapnik2006, p. 481], [Sugiyama2016, p. 236], we often assume that y can be expressed in terms of a function of x *up to a white noise*, i.e.

$$y = f(x) + \varepsilon \tag{2.14}$$

where ε is a random error (a measurable function on $\mathcal{X}$) with zero expectation i.e., $\mathbb{E}_\mu(\varepsilon) = 0$.

This simplified setting of a supervised learning is a discriminative model.

2.3. **Empirical Risk Minimization and overfitting.** In a discriminative model of supervised learning our aim is to construct a prediction rule A that assigns a predictor h_S to each sequence

$$S = \{(x_1, y_1), \cdots, (x_n, y_n)\} \in (\mathcal{X} \times \mathcal{Y})^n$$

of i.i.d. labeled data such that the expected error $R_\mu^L(h_S)$ tends to the optimal performance error $R_{\mu,\mathcal{H}}^L$ of the class $\mathcal{H}$. One of most popular ways to find a prediction rule A is to use the Empirical Risk Minimization.

For a loss function

$$L : \mathcal{X} \times \mathcal{Y} \times \mathcal{H} \to \mathbb{R},$$

and a training data $S \in (\mathcal{X} \times \mathcal{Y})^n$ we define *the empirical risk* of a predictor h as follows

$$\hat{R}_S^L(h) := \frac{1}{n}\sum_{i=1}^{n} L(x_i, y_i, h) \in \mathbb{R}. \tag{2.15}$$

If L is fixed, then we also omit the superscript L.

The empirical risk is a function of two variables: the "empirical data" S and the predictor h. Given S a learner can compute $\hat{R}_S(h)$ for any function

$h : \mathcal{X} \to \mathcal{Y}$. A minimizer of the empirical risk should have also "approximately" minimize the expected risk. This is the *empirical risk minimization principle*, abbreviated as ERM.

Remark 2.10. We note that

$$\hat{R}_S^{L(d)}(h) = \frac{1}{n} R_{\mu_S}^{L(d^n)}(h) \tag{2.16}$$

where μ_S is the Dirac measure on $(\mathcal{X} \times \mathcal{Y})^n$ associated to S, see (2.6). If h is fixed, by the weak law of large numbers, the RHS of (2.16) converges in probability to the expected risk $R_\mu^L(h)$, so we could hope to find a condition under which the RHS of (2.16) for a sequence of h_S, instead of h, converges to $R_{\mu,\mathcal{H}}^L$.

Example 2.11. In this example we shall show the failure of ERM in certain cases. The 0-1 empirical risk corresponding to 0-1-loss function $L : \mathcal{X} \times \mathcal{Y} \times \mathcal{Y}^{\mathcal{X}} \to \{0, 1\}$ is defined as follows

$$\hat{R}_S^{0-1}(h) := \frac{|i \in [n] : \, h(x_i) \neq y_i|}{n} \tag{2.17}$$

for a training data $S = \{(x_1, y_1), \cdots, (x_n, y_n)\}$ and a function $h : \mathcal{X} \to \mathcal{Y}$. We also often call $R_S^{0-1}(h)$ - *the training error* or *the empirical error*.

Now we assume that labeled data (x, y) is generated by a map $f : \mathcal{X} \to \mathcal{Y}$, i.e., $y = f(x)$, and further more, x is distributed by a measure $\mu_{\mathcal{X}}$ on $\mathcal{X}$ as in Example 2.5. Then $(x, f(x))$ is distributed by the measure $\mu_f = (\Gamma_f)_*(\mu_{\mathcal{X}})$. Let $\mathcal{H} = \mathcal{Y}^{\mathcal{X}}$. Then $f \in \mathcal{H}$ and $R_{\mu_f}^{0-1}(f) = 0$. For any given $\varepsilon > 0$ and any n we shall find a map f, a measure $\mu_{\mathcal{X}}$, and a predictor h_{S_n} such that $\hat{R}_{S_n}^{0-1}(h_{S_n}) = 0$ and $R_{\mu_f}^{0-1}(h_{S_n}) = \varepsilon$, which shall imply that the ERM is invalid in this case.

Set

$$h_{S_n}(x) = \begin{cases} f(x_i) & \text{if there exists } i \in [n] \text{ s.t. } x_i = x \\ 0 & \text{otherwise.} \end{cases} \tag{2.18}$$

Clearly $\hat{R}_{S_n}^{0-1}(h_{S_n}) = 0$. We also note that $h_{S_n}(x) = 0$ except finite (at most n) points x in $\mathcal{X}$.

Let $\mathcal{X}$ be the unit cube I^k in $\mathbb{R}^k$ and $\mathcal{Y} = \mathbb{Z}_2$. Let μ_0 be the Lebesgue measure on I^k, $k \geq 1$.We decompose $\mathcal{X}$ into a disjoint union of two measurable subsets A_1 and A_2 such that $\mu_{\mathcal{X}}(A_1) = \varepsilon$. Let $f : \mathcal{X} \to \mathbb{Z}_2$ be equal 1_{A_1} - the indicator function of A_1. By (2.7) we have

$$R_{\mu_f}(h_{S_n}) = \mu_{\mathcal{X}}(\{x \in \mathcal{X} | \, h_{S_n}(x) \neq 1_{A_1}(x)\}). \tag{2.19}$$

Since $h_{S_n}(x) = 0$ a.e. on $\mathcal{X}$ it follows from (2.19) that

$$R_{\mu_f}(h_{S_n}) = \mu_{\mathcal{X}}(A_1) = \varepsilon.$$

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

Exercise 2.12 (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 (2.12)), find the hypothesis $h_S \in \mathcal{F}$ that minimizes the empirical risk $\hat{R}_S^L$.

The phenomenon of overfitting suggests the following questions concerning ERM principle [Vapnik2000, p.21]

1) Can we learn in discriminative model of supervised learning using the ERM principle?

2) If we can learn, we would like to know the rate of convergence of the learning process as well as construction method of learning algorithms.

We shall address these questions later in our course and recommend the books by Vapnik on statistical learning theory for further reading.

2.4. **Conclusion.** In this lecture we learn a discriminative model of supervised learning which consists of a hypothesis space $\mathcal{H}$ of functions $\mathcal{X} \to \mathcal{Y}$ and an expected risk function R_μ^L on $\mathcal{H}$ where L is an instantaneous loss function and $\mu \in \mathcal{P}(\mathcal{X} \times \mathcal{Y})$ is a unknown probability distribution of labeled pairs on $\mathcal{X} \times \mathcal{Y}$. The aim of a learner is to find a prediction rule $A : S \mapsto h_S \in \mathcal{H}$ such that $(x, h_S(x))$ approximates the labeled training data best, assuming that S is a sequence of i.i.d. labeled training data. The ERM principle suggests that we could choose h_S to be the minimizer of the empirical risk $\hat{R}_S$ and we hope that as the size of S increases the expected error $R_\mu^L(h_S)$ converges to the optimal performance error $R_{\mu,\mathcal{H}}^L$. Without further condition on $\mathcal{H}$ and L the ERM principle does not work.

3. Statistical models and frameworks for unsupervised learning and reinforcement learning

Last week we learned discriminative and generative models of supervised learning. The starting point of our models is Vapnik's postulate: learning is a problem of function estimation on the basis of empirical data. In supervised learning we are given i.i.d. labeled data and the problem is to predict the label of a new/unseen instance. If we regard this prediction as a function $\mathcal{X} \to \mathcal{Y}$ (up to a negligible noise) then we have to find/estimate a function from a hypothesis space $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ such that its expected error is as small as possible, using labeled data. If we wish instead to estimate the conditional probability $p(y|x)$ or the joint distribution of the labeled data then our model is generative. The error function is a central notion of learning theory that specifies the idea of "best approximation", "best predictor".

Today we shall study statistical models of machine learning for several important tasks in unsupervised learning: density estimation, clustering, dimension reduction, manifold learning and a mathematical model for reinforcement learning. The key problem is to specify the error function that measures the accuracy of an estimator or the fitness of a decision.

3.1. **Statistical models and frameworks for density estimation.** Let $\mathcal{X}$ be a measurable space and denote by $\mathcal{P}(\mathcal{X})$ the space of all probability measures on $\mathcal{X}$. In a density estimation problem we are given a sequence of observables $S_n = (x_1, \cdots, x_n) \in \mathcal{X}^n$, which are i.i.d. by unknown probability measure μ_u. We have to estimate the measure $\mu_u \in \mathcal{P}(\mathcal{X})$. Furthermore, having a prior information, we assume that μ_u belongs to a subset $P \subset \mathcal{P}(\mathcal{X})$, which is also called *a statistical model*. Simplifying further, we assume that P consists of probability measures that are dominated by a measure $\mu_0 \in \mathcal{P}(\mathcal{X})$. Thus we regard P as a family of density functions on $\mathcal{X}$. If P is finite dimensional, then estimating $\mu_u \in P$ is called *a parametric problem of density estimation*, otherwise it is called *a nonparametric problem*. The density estimation problem encompasses the problem of estimating the joint distribution in the generative model of supervised learning as particular case.

• *In the parametric density estimation problem* we assume that $P \subset \mathcal{P}(\mathcal{X})$ is parameterized by a nice parameter set Θ, e.g. Θ is an open set of $\mathbb{R}^n$. That is, there exists a surjective map $\mathbf{p} : \Theta \to P$, $\theta \mapsto p_\theta \mu_0$, which is usually (in classical statistics) assumed to be a 1-1 map. [9] In this lecture we shall assume that Θ is an open subset of $\mathbb{R}^n$ and $\mathbf{p}$ is a 1-1 map. Thus we shall identify P with Θ and the parametric density estimation in this case is equivalent to estimating the parameter $\mathbf{p}^{-1}(\mu_u) \in \Theta$. As in mathematical models for supervised learning, we define an expected risk function $R_\mu : \Theta \to \mathbb{R}$ by averaging an *instantaneous loss function* $L : \mathcal{X} \times \Theta \to \mathbb{R}$ using the unknown probability measure μ_u which we have to estimate. Usually this setting of density estimation L given by the minus log-likelihood function

$$L(x, \theta) = -\log p_\theta(x). \tag{3.1}$$

Hence the expected risk function $R_\mu : \Theta \to \mathbb{R}$ is *the expected log-likelihood function*:

$$R_\mu(\theta) = R^L_{\mu_u}(\theta) = -\int_{\mathcal{X}} \log p_\theta(x) p_u(x) d\mu_0 \tag{3.2}$$

where $\mu_u = p_u \mu_0$. Given a data $S_n = (x_1, \cdots, x_n) \in \mathcal{X}^n$, by (3.1), the corresponding empirical risk function is

$$\hat{R}^L_{S_n}(\theta) = -\sum_{i=1}^n \log p_\theta(x_i) = -\log[p^n_\theta(S_n)], \tag{3.3}$$

where $p^n_\theta(S_n)$ is the density of the probability measure μ^n_θ on $\mathcal{X}^n$. It follows that the minimizer θ of the empirical risk $\hat{R}^L_{S_n}$ is the maximizer of the log-likelihood function $\log[p^n_\theta(S_n)]$. According to ERM principle, the minimizer θ of $\hat{R}^L_{S_n}$ should provide an "approximation" of the density p_u of the unknown probability measure μ_u.

[9] For many important statistical models in machine learning the condition 1-1 map does not hold and we refer to [AJLS2017] for a general treatment.

Remark 3.1. (1) For $\mathcal{X} = \mathbb{R}$ the ERM principle for the expected log-likelihood function holds. Namely one can show that the minimum of the risk functional in (3.2), if exists is attained at a function p_u^* which may differ from p_u only on a set of zero measure, see [Vapnik1998, p.30] for a proof.

(2) Note that minimizing the expected log-likelihood function $R_\mu(\theta)$ is the same as minimizing the following modified risk function [Vapnik2000, p.32]

$$R_\mu^*(\theta) := R_\mu(\theta) + \int_{\mathcal{X}} \log p_u(x) p_u(x) d\mu_0 = -\int_{\mathcal{X}} \log \frac{p_\theta(x)}{p_u(x)} p_u(x) d\mu_0. \tag{3.4}$$

The expression on the RHS of (3.4) is the Kullback-Leibler divergence $KL(p_\theta\mu_0|\mu_u)$ that is used in statistics for measuring the divergence between $p_\theta\mu_0$ and $\mu_u = p_u\mu_0$. The Kullback-Leibler divergence $KL(\mu|\mu')$ is dedined for probability measures $(\mu, \mu') \in \mathcal{P}(\mathcal{X}) \times \mathcal{P}(\mathcal{X})$ such that $\mu << \mu'$, see also Remark 4.9 below. It is a quasi-distance, i.e., it satisfies the following properties:

$$KL(\mu|\mu') \geq 0 \text{ and } KL(\mu|\mu') = 0 \text{ iff } \mu = \mu'. \tag{3.5}$$

Thus a maximizer of the expected log-likelihood function minimizes the KL-divergence. This justifies the choice of the expected risk function $R_{\mu_u}^L$.

(3) It is important to find quasi-distance functions on $\mathcal{P}(\Omega)$ that satisfying certain natural statistical requirement. This problem has been considered in information geometry, see [Amari2016, AJLS2017] for further reading.

(4) Traditionally in statistics people consider only measures that can be expressed as a density function w.r.t. a given (dominant) measure. This assumption holds in classical situations, when we consider only finite dimensional families of probability measures. Currently in machine learning one also uses infinite dimensional family of probability measures on $\mathcal{X}$ that cannot be dominated by any measure on $\mathcal{X}$, for examples, the infinite dimensional family of posterior distributions of Dirichlet processes which are used in clustering.

• A popular *nonparametric technique* for estimating density functions on $\mathcal{X} = \mathbb{R}^{m\iota}$ using empirical data $S_n \in \mathcal{X}^n$ is *the kernel density estimation* (KDE) [Tsybakov2009, p. 2]. For understanding the idea of KDE we shall consider only the case $\mathcal{X} = \mathbb{R}$ and $\mu_0 = dx$. Let

$$F(t) = \int_{-\infty}^{t} p_u(x) dx$$

be the corresponding cumulative distribution function. Consider *the empirical distribution function*

$$\hat{F}_{S_n}(t) = \frac{1}{n} \sum_{i=1}^{n} 1_{(x_i \leq t)}.$$

By the strong law of large numbers we have

$$\lim_{n\to\infty} F_{S_n}(t) \overset{a.s.}{=} F(t).$$

How can we estimate the density p_u? Note that for sufficiently small $h > 0$ we can write an approximation

$$p_u(t) \cong \frac{F(t+h) - F(t-h)}{2h}.$$

Replacing F by $\hat{F}_{S_n}$ we define *the Rosenblatt estimator*

$$\hat{p}^R_{S_n}(t) := \frac{\hat{F}_{S_n}(t+h) - \hat{F}_{S_n}(t-h)}{2h}, \tag{3.6}$$

which can be rewritten in the following form

$$\hat{p}^R_{S_n}(t) = \frac{1}{2nh}\sum_{i=1}^{n} 1_{(t-h\leq x_i\leq t+h)} = \frac{1}{nh}\sum_{i=1}^{n} K_0(\frac{x_i - t}{h}) \tag{3.7}$$

where $K_0(u) := \frac{1}{2}1_{(-1\leq u\leq 1)}$. A simple generalization of the Rosenblatt estimator is given by

$$\hat{p}^{PR}_{S_n}(t) = \frac{1}{nh}\sum_{i=1}^{n} K(\frac{x_i - t}{h}) \tag{3.8}$$

where $K : \mathbb{R} \to \mathbb{R}$ is an integrable function satisfying $\int K(u)du = 1$. Such a function K is called *kernel* and the parameter h is called *bandwidth* of the *kernel density estimator* (3.8), also called *the Parzen-Rosenblatt estimator*.

To measure the accuracy of the estimator $\hat{p}^{PR}_{S_n}$ we use a trick, namely instead of using the L_2-estimation

$$MSE(\hat{f}^{PR}_{S_n}) := \int_{\mathbb{R}} |\hat{p}^{PR}_{S_n}(x) - p_u(x)|^2 dx,$$

we consider $MSE(\hat{f}^{PR}, x_0)$ of $\hat{f}^{PR}_{S_n}$ w.r.t. a given point $x_0 \in \mathbb{R}$, averaging over the population of all possible data $S_n \in \mathbb{R}^n$:

$$MSE(\hat{f}^{PR}, x_0) := \mathbb{E}_{p^n_u}[(\hat{p}^{PR}_{S_n}(x_0) - p_u(x_0))^2]dS_n. \tag{3.9}$$

Note that the RHS measures the accuracy of $\hat{p}^{PR}_{S_n}(x_0)$ *probably w.r.t.* $S_n \in \mathbb{R}^n$. This is an important concept of accuracy in the presence of uncertainty.

It has been proved that under certain condition on the kernel function K and the infinite dimensional statistical model P of densities the $MSE(\hat{f}^{PR}, x_0)$ converges to zero uniformly on $\mathbb{R}$ as h goes to zero [Tsybakov2009, Theorem 1.1, p. 9].

Remark 3.2. In this Subsection we discuss two popular models of machine learning for density estimation using ERM principle, which works under certain conditions. We postpone important Bayesian model of machine learning and stochastic approximation method for finding minimizer of the expected risk function using i.i.d. data to later parts of our course.

3.2. **Statistical models and frameworks for clustering.** Clustering is the process of grouping similar objects $x \in \mathcal{X}$ together. There are two possible types of grouping: partitional clustering, where we partition the objects into disjoint sets; and hierarchical clustering, where we create a nested tree of partitions. To formalize the notion of similarity we introduce a quasi-distance function on $\mathcal{X}$. That is, a function $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$ that is symmetric, satisfies $d(x, x) = 0$ for all $x \in \mathcal{X}$.

A popular approach to clustering starts by defining a cost function over a parameterized set of possible clusterings and the goal of the clustering algorithm is to find a partitioning (clustering) of minimal cost. Under this paradigm, the clustering task is turned into an optimization problem. The function to be minimized is called *the objective function*, which is a function G from pairs of an input $(\mathcal{X}, d)$, and a proposed clustering solution $C = (C_1, \cdots, C_k)$, to positive real numbers. Given G, the goal of a clustering algorithm is defined as finding, for a given input $(\mathcal{X}, d)$, a clustering C so that $G((X, d), C)$ is minimized. In order to reach that goal, one has to apply some appropriate search algorithm. As it turns out, most of the resulting optimization problems are NP-hard, and some are even NP-hard to approximate.

Example 3.3. The *k-means objective function* is one of the most popular clustering objectives. In k-means the data is partitioned into disjoint sets $C_1, \cdots, C_k$ where each C_i is represented by a centroid $\mu_i := \mu_i(C_i)$. It is assumed that the input set $\mathcal{X}$ is embedded in some larger metric space $(\mathcal{X}', d)$ and $\mu_i \subset \mathcal{X}'$. We define μ_i as follows

$$\mu_i(C_i) := \arg\min_{\mu \in \mathcal{X}'} \sum_{x \in C_i} d(x, \mu).$$

The k-means objective function G_k is defined as follows

$$G_k((X, d), (C_1, \cdots, C_k)) = \sum_{i=1}^{k} \sum_{x \in C_i} d(x, \mu_i(C_i)). \tag{3.10}$$

The k-means objective function G is used in digital communication tasks, where the members of $\mathcal{X}$ may be viewed as a collection of signals that have to be transmitted. For further reading on k-means algorithms, see [SSBD2014, p. 313].

Remark 3.4. The above formulation of clustering is deterministic. We consider also more complicated probabilistic clustering, where the output is a function assigning to each domain point $x \in \mathcal{X}$, a vector $(p_1(x), \cdots, p_k(x))$, where $p_i(x) = P[x \in C_i]$ is the probability that x belongs to cluster C_i.

3.3. **Statistical models and frameworks for dimension reduction and manifold learning.** A central cause of the difficulties with unsupervised learning is the high dimensionality of the random variables being modeled. As the dimensionality of input x grows, any learning problem significantly gets harder and harder. Handling high-dimensional data is cumbersome in practice, which is often referred to as *the curse of dimensionality*. Hence various methods of dimensionality reduction are introduced. Dimension reduction is the process of taking data in a high dimensional space and mapping it into a new space whose dimension is much smaller.

• *Classical (linear) dimension reduction methods.* Given original data $S_m := \{x_i \in \mathbb{R}^d | \, i \in [1, m]\}$ we want to embed it into $\mathbb{R}^n$, $n < d$, then we would like to find a linear transformation $W \in Hom(\mathbb{R}^d, \mathbb{R}^n)$ such that

$$W(S_m) := \{W(x_i)\} \subset \mathbb{R}^n.$$

To find the "best" transformation $W = W_{(S_m)}$ we define an error function on the space $Hom(\mathbb{R}^d, \mathbb{R}^n)$ and solve the associated optimization problem.

Example 3.5. A popular linear method for dimension reduction is called *Principal Component Analysis* (PCA), which has another name SVD (singular value decomposition.) Given $S_m \subset \mathbb{R}^d$, we use a linear transformation $W \in Hom(\mathbb{R}^d, \mathbb{R}^n)$, where $n < d$, to embed S_m into $\mathbb{R}^d$. Then, a second linear transformation $U \in Hom(\mathbb{R}^n, \mathbb{R}^d)$ can be used to (approximately) recover S_m from its compression $W(S_m)$. In PCA, we search for W and U to be a minimizer of the following *reconstruction error* function $R_{S_m} : Hom(\mathbb{R}^d, \mathbb{R}^n) \times Hom(\mathbb{R}^n, \mathbb{R}^d) \to \mathbb{R}$

$$\hat{R}_{S_m}(W, U) = \sum_{i=1}^{m} ||x_i - UW(x_i)||^2 \tag{3.11}$$

where $||, ||$ denotes the quadratic norm.

Exercise 3.6. ([SSBD2014, Lemma 23.1, p.324]) Let (W, U) be a minimizer of $\hat{R}_{S_m}$ defined in (3.11). Show that U can be chosen as an orthogonal embedding and $W \circ U = Id_{|\mathbb{R}^n}$.

Hint. First we show that if a solution (W, U) of (3.11) exists, then there is a solution (W', U') of (3.11) such that $\dim \ker(U'W') = d - n$.

Let $Hom_g(\mathbb{R}^d, \mathbb{R}^n)$ denotes the set of all ort hogonal projections from $\mathbb{R}^d$ to $\mathbb{R}^n$ and $Hom_g(\mathbb{R}^n, \mathbb{R}^d)$ the set of all orthogonal embeddings from $\mathbb{R}^m$ to $\mathbb{R}^d$. Let $\mathcal{F} \subset Hom_g(\mathbb{R}^d, \mathbb{R}^n) \times Hom_g(\mathbb{R}^n, \mathbb{R}^d)$ be the subset of all pairs (W, U) of transformations such that $W \circ U = Id_{|\mathbb{R}^n}$. Exercise 3.6 implies that any minimizer (W, U) of $\hat{R}_{S_m}$ is an element of $\mathcal{F}$.

Exercise 3.7. ([SSBD2014, Theorem 3.23, p. 325]) Let $C(S_m) \in End(\mathbb{R}^d)$ be defined as follows

$$C(S_m)(v) := \sum_{i=1}^{m} \langle x_i, v \rangle x_i.$$

Assume that $\xi_1, \cdots, \xi_d \in \mathbb{R}^d$ are eigenvectors of $C(S_m)$ with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_d \geq 0$. Show that any $(W, U) \in \mathcal{F}$ with $W(x_j) = 0$ for all $j \geq m + 1$ is a solution of (3.11).

Thus a PCA problem can be solved using linear algebra method.

• *Manifold learning and autoencoder.* In real life data are not concentrated on a linear subspace of $\mathbb{R}^d$ but around a submanifold $M \subset \mathbb{R}^d$. The current challenge in ML community is that to reduce representation of data in $\mathbb{R}^d$ using all the data in $\mathbb{R}^d$ but only use only data concentrated around M. For that purpose we use autoencoder, which is a non-linear analogue of PCA.

In an auto-encoder we learn a pair of functions: an *encoder function* $\psi : \mathbb{R}^d \to \mathbb{R}^n$, and a *decoder function* $\varphi : \mathbb{R}^n \to \mathbb{R}^d$. The goal of the learning process is to P find a pair of functions (ψ, φ) such that *the reconstruction error*

$$R_{S_m}(\psi, \varphi) := \sum_{i=1}^{m} ||x_i - \varphi(\psi(x_i))||^2$$

is small. We therefore must restrict ψ and φ in some way. In PCA, we constrain $k < d$ and further restrict ψ and φ to be linear functions.

Remark 3.8. Modern autoencoders have generalized the idea of an encoder and a decoder beyond deterministic functions to *stochastic mappings* $p_{stoch} : \mathcal{X} \to \mathcal{Y}$, $(x, y) \mapsto p(y|x)$. For further reading I recommend [SSBD2014], [GBC2016] and [Bishop2006, §12.2, p.570].

3.4. **Statistical model and framework for reinforcement learning.** A reinforcement learning agent interacts with its environment in discrete time steps. At each time t, the agent receives an observation o_t, which typically includes the reward r_t. It then chooses an action a_t from a set A of available actions, which is subsequently sent to the environment. The environment moves to a new state s_{t+1} in a set S of available states and the reward r_{t+1} associated with the transition $o_{t+1} := (s_t, a_t, s_{t+1})$ is determined. The goal of a reinforcement learning agent is to collect as much reward as possible. The agent can (possibly randomly) choose any action as a function of the history. The uncertainly in reinforcement learning is expressed in terms of a transition probability $Pr[s'|s, a]$ - distribution over destination states $s' = \delta(s, a)$ and in terms of a reward probability $Pr[r'|s, a]$ - distribution over rewards returned $r' = r(s, a)$. Thus the mathematical model of reinforcement learning is a Markov decision process. For further reading, see [MRT2012, chapter 14].

3.5. **Conclusion.** In this lecture we learned mathematical models for unsupervised learning, where for each empirical data $S_n \in \mathcal{X}^n$ we choose an empirical risk/error function $\hat{R}_{S_n}$ on a space of possible hypotheses as a quasi-distance between a hypothesis and the true (desired) hypothesis. In some case (e.g. in parametric density estimation) we can interpret this empirical risk function to be derived from an expected risk/error function as in the models of supervised learning. The main problem is to show the convergence of minimizers of $\hat{R}_{S_n}$ to the desired hypothesis as n goes to zero and S_n are i.i.d. by an unknown measure on $\mathcal{X}^n$.

4. Fisher metric and maximum likelihood estimator

In the last lecture we considered several mathematical models in unsupervised learning. The most important problem among them is the problem of density estimation, which is also a problem in generative models of supervised learning and an important problem of classical statistics. The error function in density estimation problem can be defined as the expected log-likelihood function, which combining with the ERM principle leads to the well known maximum likelihood estimator. The popularity of this estimator stems from its asymptotic accuracy, also called consistency, which holds under mild conditions. Today we shall study MLE using the Fisher metric, the associated MSE function and the Cramér-Rao inequality.

We also clarify the relation between the Fisher metric and the Kullback-Leibler divergence.

4.1. **The space of all probability measures and total variation norm.** We begin today lecture with our investigation of natural geometry of $\mathcal{P}(\mathcal{X})$ for an arbitrary measurable space $(\mathcal{X}, \Sigma)$. This geometry induces the Fisher metric on any statistical model $P \subset \mathcal{P}(\mathcal{X})$ satisfying a mild condition.

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

$$\mathcal{M}(\mathcal{X}) := \{\mu \ : \ \mu \text{ a finite measure on } \mathcal{X}\},$$

$$\mathcal{S}(\mathcal{X}) := \{\mu \ : \ \mu \text{ a signed finite measure on } \mathcal{X}\}.$$

It is known that $\mathcal{S}(\mathcal{X})$ is a Banach space 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 $\mathcal{X} = A_1 \dot{\cup} \ldots \dot{\cup} A_n$ with disjoint sets $A_i \in \Sigma(\mathcal{X})$ (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 $\phi : \mathcal{X} \to$

$[-\infty,\infty]$ we define $\phi_+ := \max(\phi,0)$ and $\phi_- := \max(-\phi,0)$, so that $\phi_\pm \geq 0$ are measurable with disjoint support, and

$$\phi = \phi_+ - \phi_- \qquad\qquad |\phi| = \phi_+ + \phi_-. \tag{4.1}$$

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

$$\mu = \mu_+ - \mu_- \qquad \text{with } \mu_\pm \in \mathcal{M}(\mathcal{X}),\ \mu_+ \perp \mu_-. \tag{4.2}$$

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

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

then (4.2) implies

$$|\mu(A)| \leq |\mu|(A) \qquad \text{for all } \mu \in \mathcal{S}(\mathcal{X}) \text{ and } A \in \Sigma(\mathcal{X}), \tag{4.3}$$

so that

$$\|\mu\|_{TV} = \| \, |\mu| \, \|_{TV} = |\mu|(\mathcal{X}).$$

In particular,

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

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

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

$$\mathcal{S}(\mathcal{X},\mu_0) := \{\mu \in \mathcal{S}(\mathcal{X}) \ : \ \mu \text{ is dominated by } \mu_0\}.$$

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

$$\imath_{can} : L^1(\mathcal{X},\mu_0) \longrightarrow \mathcal{S}(\mathcal{X},\mu_0), \qquad \phi \longmapsto \phi\,\mu_0. \tag{4.4}$$

Observe that $\imath_{can}$ is an isomorphism of Banach spaces, since evidently

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

Example 4.1. Let $\mathcal{X}_n := \{\omega_1,\cdots,\omega_n\}$ be a finite set of n elementary events. Let δ_{ω_i} denote the Dirac measure concentrated at ω_i. Then

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

and

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

For $\mu \in \mathcal{M}(\mathcal{X}_n)$ 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 $L^1(\mathcal{X}_n, \mu)$ with the total variation norm is isomorphic to $\mathbb{R}^k$ with the l^1-norm. The space $\mathcal{P}(\mathcal{X}_n)$ 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 4.2. ([JLS2017]) For any countable family of signed measures $\{\mu_n \in \mathcal{S}(\mathcal{X})\}$ show that there exists a measure $\mu \in \mathcal{M}(\mathcal{X})$ dominating all measures μ_n.

Remark 4.3. 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. Denote by $Lin(V, W)$ the space of all continuous linear map from V to W. 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

$$\lim_{h \to 0} \frac{\|\phi(x+h) - \phi(x) - d_x\phi(h)\|_W}{\|h\|_V} = 0. \tag{4.5}$$

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.

A map ϕ from an open subset Θ of a Banach space V to a subset $\mathcal{X}$ of a Banach space W is called *differentiable*, if the composition $i \circ \phi : \Theta \to W$ is differentiable.

4.2. **Fisher metric on a statistical model.** Given a statistical model $P \subset \mathcal{P}(\mathcal{X})$ we shall show that P is endowed with a nice geometric structure induced from the Banach space $(\mathcal{S}(\mathcal{X}), ||, ||_{TV})$. Under a mild condition this implies the existence of the Fisher metric on P. Then we shall compare the Fisher metric and the Kullback-Leibler divergence.

We study P by investigating the space of functions on P (which is a linear infinite dimensional vector space) and by investigating its dual version: the space of all curves on P. The tangent fibration of P describes the first order approximation of the later space.

Definition 4.4. ([AJLS2017, Definition 3.2, p. 141]) (1) Let $(V, \|\cdot\|)$ be a Banach space, $\mathcal{X} \subset V$ an arbitrary subset and $x_0 \in \mathcal{X}$. Then $v \in V$ is called *a tangent vector of* $\mathcal{X}$ *at* x_0, if there is a curve $c : (-\varepsilon, \varepsilon) \to \mathcal{X} \subset V$ such that $c(0) = x_0$ and $\dot{c}(0) = v$.

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

(3) *The tangent cone fibration* $C\mathcal{X}$ (resp. *the tangent fibration* $T\mathcal{X}$) is the union $\cup_{x \in \mathcal{X}} C_x\mathcal{X}$ (resp. $\cup_{x \in \mathcal{X}} T_x\mathcal{X}$) is a subset of $V \times V$ and therefore is endowed with the induced topology.

Exercise 4.5. (cf. [AJLS2018, Theorem 2.1]) Let P be a statistical model. Show that any $v \in T_\xi P$ is dominated by ξ. Hence *the logarithmic representation of* v

$$\log v := dv/d\xi$$

is an element of $L^1(\mathcal{X}, \xi)$.

Example 4.6. Assume that a statistical model P consists of measures dominated by a measure μ_0 and therefore P is regarded as a family of density functions on $\mathcal{X}$, namely

$$P = \{f \cdot \mu_0 |\, f \in L^1(\mathcal{X}, \mu_0)\}. \tag{4.6}$$

Then a tangent vector $v \in T_\xi P$ has the form $v = \dot{f}(0) \cdot \mu_0$, where $\xi = f(0)\mu_0$, and its logarithmic representation is expressed as follows

$$\log v = \frac{dv}{d\xi} = \frac{\dot{f}(0)}{f(0)} = \frac{d}{dt}_{|t=0} \log f(t). \tag{4.7}$$

Next we want to put a Riemannian metric on P i.e., to put a positive quadratic form $\mathfrak{g}$ on each tangent space $T_\xi P$. By Exercise 4.5, the logarithmic representation $\log(T_\xi P)$ of $T_\xi P$ is a subspace in $L^1(\mathcal{X}, \xi)$. The space $L^1(\mathcal{X}, \xi)$ does not have a natural metric but its subspace $L^2(\mathcal{X}, \xi)$ is a Hilbert space.

Definition 4.7. (1) A statistical model P that satisfies

$$\log(T_\xi P) \subset L^2(\mathcal{X}, \xi) \tag{4.8}$$

for all $\xi \in P$ is called *almost 2-integrable*.

(2) Assume that P is an almost 2-integrable statistical model. For each $v, w \in C_\xi P$ *the Fisher metric on* P is defined as follows

$$\mathfrak{g}(v, w) := \langle \log v, \log w \rangle_{L^2(\mathcal{X}, \xi)} = \int_{\mathcal{X}} \log v \cdot \log w \, d\xi. \tag{4.9}$$

(3) An almost 2-integrable statistical model P is called *2-integrable*, if the function $v \mapsto |v|_{\mathfrak{g}}$ is continuous on CP.

Since $T_\xi P$ is a linear hull of $C_\xi P$, the formula (4.9) extends uniquely to a positive quadratic form on $T_\xi P$, which is also called *the Fisher metric*.

Example 4.8. Let $P \subset \mathcal{P}(\mathcal{X})$ be a 2-integrable statistical model that is parameterized by a differentiable map $\mathbf{p} : \Theta \to P$, $\theta \mapsto p_\theta \mu_0$, where Θ is an open subset in $\mathbb{R}^n$. It follows from (4.7) that the Fisher metric on P has the following form

$$\mathfrak{g}_{|\mathbf{p}(\theta)}(d\mathbf{p}(v), d\mathbf{p}(w)) = \int_{\mathcal{X}} \frac{\partial_v p_\theta}{p_\theta} \cdot \frac{\partial_w p_\theta}{p_\theta} p_\theta d\mu_0, \tag{4.10}$$

for any $v, w \in T_\theta \Theta$.

Remark 4.9. (1) The Fisher metric has been defined by Fisher in 1925 to characterize "information" of a statistical model. One of most notable applications of the Fisher metric is the Cramér-Rao inequality which measures our ability to have a good density estimator in terms of geometry of the underlying statistical model, see Theorem 4.13 below.

(2) The Fisher metric $\mathfrak{g}_{\mathbf{p}(\theta)}(d\mathbf{p}(v), d\mathbf{p}(v))$ of a parametrized statistical model P of dominated measures in Example 4.8 can be obtained from the Taylor expansion of the Kullback-Leibler divergence $I(\mathbf{p}(\theta), \mathbf{p}(\theta + \varepsilon v))$, assuming that $\log p_\theta$ is continuously differentiable in all partial derivative in θ up to order 3. Indeed we have

$$I(\mathbf{p}(\theta), \mathbf{p}(\theta + \varepsilon v)) = \int_{\mathcal{X}} p_\theta(x) \log \frac{p_\theta(x)}{p_{\theta+\varepsilon v}(x)} d\mu_0$$

$$= -\varepsilon \int_{\mathcal{X}} p_\theta(x) \partial_v \log p_\theta(x) d\mu_0 \tag{4.11}$$

$$- \varepsilon^2 \int_{\mathcal{X}} p_\theta(x) (\partial_v)^2 \log p_\theta(x) d\mu_0 + O(\varepsilon^3). \tag{4.12}$$

Since $\log_\theta(x)$ is continuously differentiable in θ up to order 3, we can apply differentiation under the integral sign, see e.g. [Jost2005, Theorem 16.11, p. 213] to (4.11), which then must vanish, and integration by part to (4.12). Hence we obtain

$$I(\mathbf{p}(\theta), \mathbf{p}(\theta + \varepsilon v)) = \varepsilon^2 \mathfrak{g}_{\mathbf{p}(\theta)}(d\mathbf{p}(v), d\mathbf{p}(v)) + O(\varepsilon^3)$$

what is required to prove.

4.3. **The Fisher metric, MSE and Cramér-Rao inequality.** As we learned with the regression problem in Exercise 2.7, it is important to narrow a hypothesis class to define a good loss/risk function, 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 $\varphi : P \to \mathbb{R}^n$. [10].

Definition 4.10. *A φ-estimator* is a composition of an estimator $\hat{\sigma} : \Omega \to P$ and a map $\varphi : P \to \mathbb{R}^n$.

Set $\varphi^l : l \circ \varphi$ for any $l \in (\mathbb{R}^n)^*$ and

$L^2_\varphi(P, \mathcal{X}) := \{\hat{\sigma} : \mathcal{X} \to P | \, \varphi^l \circ \hat{\sigma} \in L^2(\mathcal{X}, \xi)$ for all $\xi \in P$ and for all $l \in (\mathbb{R}^n)^*\}$.

[10] see [MFSS2016] for examples of $\varphi : \mathcal{P}(\mathcal{X}) \to \mathcal{H}$, where $\mathcal{H}$ is a RKHS (see also Definition 8.9) which is a generalization of the method of moment, see e.g. [Borovkov1998, p. 56]

If $\hat{\sigma} \in L^2_\varphi(P, \mathcal{X})$, then the following L_2-risk function, also called MSE, is well-defined for any $l, k \in (\mathbb{R}^n)^*$

$$MSE^\varphi_\xi[\hat{\sigma}](l,k) := \mathbb{E}_\xi[(\varphi^l \circ \hat{\sigma} - \varphi^l \circ \xi) \cdot (\varphi^k \circ \hat{\sigma} - \varphi^k \circ \xi)]. \tag{4.13}$$

Thus the function $MSE^\varphi_\xi[\hat{\sigma}](l,l)$ on P is the expected risk of the quadratic instantaneous loss function

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

Next we define the mean value $\varphi_{\hat{\sigma}}$ of a φ-estimator $\varphi \circ \hat{\sigma}$ as a $\mathbb{R}^n$-valued function on P:

$$\langle \varphi_{\hat{\sigma}}(\xi), l\rangle := \mathbb{E}_\xi(\varphi^l \circ \hat{\sigma}) = \int_{\mathcal{X}} \varphi^l \circ \hat{\sigma}\, d\xi \tag{4.14}$$

for any $l^* \in (\mathbb{R}^n)^*$.

Definition 4.11. (1) The difference

$$b^\varphi_{\hat{\sigma}} := \varphi_{\hat{\sigma}} - \varphi \in (\mathbb{R}^n)^P \tag{4.15}$$

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

(2) Given an estimator $\hat{\sigma} \in L^2_\varphi(P, \mathcal{X})$ 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 $l \in (\mathbb{R}^n)^*$

$$V^\varphi_\xi[\hat{\sigma}](l,l) := \mathbb{E}_\xi[(\varphi^l \circ \hat{\sigma} - \varphi^l_{\hat{\sigma}}) \cdot (\varphi^l \circ \hat{\sigma} - \varphi^l_{\hat{\sigma}})]. \tag{4.16}$$

The RHS of (4.16) is well-defined, since $\hat{\sigma} \in L^2_\varphi(P, \mathcal{X})$. It is a quadratic form on $(\mathbb{R}^n)^*$ and will be denoted by $V^\varphi_\xi[\hat{\sigma}]$.

Exercise 4.12. ([JLS2017]) Prove the following formula

$$MSE^\varphi_\xi[\hat{\sigma}](l,k) = V^\varphi_\xi[\hat{\sigma}](l,k) + \langle b^\varphi_{\hat{\sigma}}(\xi), l\rangle \cdot \langle b^\varphi_{\hat{\sigma}}(\xi), k\rangle \tag{4.17}$$

for all $\xi \in P$ and all $l, k \in (\mathbb{R}^n)^*$.

By Proposition 3.3 in [JLS2017], since P is 2-integrable, the function $\varphi^l_{\hat{\sigma}} := \langle \varphi_{\hat{\sigma}}(\xi), l\rangle$ is *differentiable*, i.e., there is differential $d\varphi^l_{\hat{\sigma}} \in T^*_\xi P$ for any $\xi \in P$ such that $\partial_v \varphi^l_{\hat{\sigma}}(\xi) = d\varphi^l_{\hat{\sigma}}(v)$ for all $v \in T_\xi P$. Here for a function f on P we define $\partial_v f(\xi) := \dot{f}(c(t))$ where $c(t) \subset P$ is a curve with $c(0) = \xi$ and $\dot{c}(0) = v$. The differentiability of $\varphi^l_{\hat{\sigma}}$ is proved by differentiation under integral

$$\partial_v \varphi^l_{\hat{\sigma}} = \int_{\mathcal{X}} \partial_v(\varphi^l \circ \hat{\sigma}\, d\xi) = \int_{\mathcal{X}} (\varphi^l \circ \hat{\sigma}(x) - E_\xi(\varphi^l \circ \hat{\sigma})) \cdot \log v\, d\xi, \tag{4.18}$$

see [AJLS2017, JLS2017] for more detail.

Recall that the Fisher metric on any 2-integrable statistical model is non-degenerate. [11] We now regard $\|d\varphi^l_{\hat\sigma}\|^2_{\mathfrak{g}^{-1}}(\xi)$ as a quadratic form on $(\mathbb{R}^n)^*$.

Theorem 4.13 (The Cramér-Rao inequality)**.** *(cf.* [AJLS2017, JLS2017]*) Let P be a finite dimensional 2-integrable statistical model with non-degenerate Fisher metric, φ a $\mathbb{R}^n$-valued function on P and $\hat\sigma \in L^2_\varphi(P, \mathcal{X})$ a φ-estimator. Then for all $l \in (\mathbb{R}^n)^*$ we have*

$$V^\varphi_\xi[\hat\sigma](l,l) - \|d\varphi^l_{\hat\sigma}\|^2_{\mathfrak{g}^{-1}}(\xi) \geq 0.$$

If $\hat\sigma$ is unbiased then its MSE is equal to its variance. In this case the Cramér-Rao inequality asserts that we can never construct an exact estimator.

Remark 4.14. Let us consider the following quadratic instantaneous loss function for $\varphi : P \to \mathbb{R}^n$

$$L_\varphi : \mathcal{X} \times L^2_\varphi(P, \mathcal{X}) \times P \to \mathbb{R} : (x, \hat\sigma, \xi) \mapsto ||\varphi(\hat\sigma(x)) - \varphi(\xi)||^2 \tag{4.19}$$

which defines the expected risk on $L^2_\varphi(P, \mathcal{X})$ a follows

$$R^{L_\varphi}_\xi(\hat\sigma) := \mathbb{E}_\xi L(x, \hat\sigma, \xi) = \mathbb{E}_\xi ||\varphi \circ \hat\sigma(x) - \varphi(\xi)||^2. \tag{4.20}$$

Clearly $\hat R^{L_\varphi}_\xi(\hat\sigma)$ is equal to the mean square error $MSE^\varphi_\xi(\hat\sigma) := \sum_{i=1}^n MSE^\varphi_\xi[\hat\sigma](e^*_i, e^*_i)$, where $\{e^*_i\}$ is an orthonormal dual basis of $(\mathbb{R}^n)^*$.

Outline of the proof of the Cramér-Rao inequality. Since P is 2-integrable and finite dimensional, the logarithmic representation $log(T_\xi P) := \{\log v |\, v \in T_\xi P\}$ is a closed subspace of the Hilbert space $L^2(\mathcal{X}, \xi)$. Denote by $\Pi_{\log(T_\xi P)}$ the orthogonal projection of $L^2(\mathcal{X}, \xi)$ to $\log(T_\xi P)$ and by $\nabla_{\mathfrak{g}} f$ the gradient of a function f on P w.r.t. the Fisher metric $\mathfrak{g}$. To prove the Cramér-Rao inequality it suffices to show the following geometric identity

$$\Pi_{\log(T_\xi P)}(\varphi^l \circ \hat\sigma - \mathbb{E}_\xi(\varphi^l \circ \hat\sigma)) = \log(\nabla_{\mathfrak{g}} \varphi^l_{\hat\sigma}) \in L_2(\mathcal{X}, \xi), \tag{4.21}$$

since the square of the L_2-norm of the term in the RHS of (4.21) is equal to $||d\varphi^l_{\hat\sigma}||^2_{\mathfrak{g}^{-1}}$ and the square of the L_2-norm of the term in the LHS is equal to $V^\varphi_\xi[\hat\sigma](l,l)$.

Next, we reduce the proof of equality (4.21) to the proof of the following equality for all $v \in T_\xi P$

$$\langle \varphi^l \circ \hat\sigma - \mathbb{E}_\xi(\varphi^l \circ \hat\sigma), \log v\rangle_{L^2} = \langle \nabla_{\mathfrak{g}} \varphi^l_{\hat\sigma}(\xi), v\rangle_{\mathfrak{g}} \tag{4.22}$$

which is obtained easily from (4.18).

[11] In literature, e.g., [Amari2016, AJLS2017, Borovkov1998], one considers the Fisher metric on a parametrized statistical model, i.e., the metric obtained by pull-back the Fisher metric on P via the parameterization map $\mathbf{p} : \Theta \to P$. This "parameterized" Fisher metric may be degenerate.

Remark 4.15. The Cramér-Rao inequality in Theorem 4.13 is non-parametric, i.e. P is not assumed to be parameterized by a smooth manifold, as in Janssen's version of Cramér-Rao inequality [Janssen2003]. The condition of 2-integrability of P is an adaptation of the 2-integrability condition of parametrized measure models in [AJLS2017, JLS2017], which is equivalent to the differentiability condition in [Janssen2003], see [JLS2017] for comments and history of the Cramér-Rao inequality.

Example 4.16. Assume that φ is a differentiable coordinate mapping, i.e., there is a differentiable parameterization $\mathbf{p}$ from an open subset Θ of $\mathbb{R}^n$ such that $\varphi \circ \mathbf{p} = Id$. Assume that $\hat{\sigma}$ is an unbiased estimator. Then the terms involving $b_{\hat{\sigma}} := b^{\varphi}_{\hat{\sigma}}$ vanishes. Since $\varphi_{\hat{\sigma}} = \varphi$ we have $||d\varphi^l||_{\mathfrak{g}^{-1}(\xi)} = ||d\varphi^l_{\hat{\sigma}}||^2_{\mathfrak{g}^{-1}(\xi)}$. Hence the Cramér-Rao inequality in Theorem 4.13 becomes the well-known Cramér-Rao inequality for an unbiased estimator

$$V_\xi := V_\xi[\hat{\sigma}] \geq \mathfrak{g}^{-1}(\xi). \tag{4.23}$$

4.4. **Efficient estimators and MLE.**

Definition 4.17. Assume that $P \subset \mathcal{P}(\mathcal{X})$ is a 2-integrable statistical model and $\varphi : P \to \mathbb{R}^n$ is a feature map. An estimator $\hat{\sigma} \in L^2_\varphi(P, \mathcal{X})$ is called *efficient*, if the Cramér-Rao inequality for $\hat{\sigma}$ becomes an equality, i.e., $V^{\varphi}_\xi[\hat{\sigma}] = ||d\varphi_{\hat{\sigma}}||^2_{\mathfrak{g}^{-1}(\xi)}$ for any $\xi \in P$.

Theorem 4.18. *Let P be a 2-integrable statistical model parameterized by a differentiable map $\mathbf{p} : \Theta \to \mathcal{P}(\mathcal{X})$, $\theta \mapsto p_\theta \mu_0$, where Θ is an open subset of $\mathbb{R}^n$, and $\varphi : P \to \mathbb{R}^n$ a differentiable coordinate mapping, i.e., $\mathbf{p} \circ \varphi = Id$. Assume that the function $p(x, \theta) := p_\theta(x)$ has continuous partial derivatives up to order 3. If $\hat{\sigma} : \mathcal{X} \to P$ is an unbiased efficient estimator then $\hat{\sigma}$ is a maximum likelihood estimator (MLE), i.e.,*

$$\partial_v \log p(\theta, x)|_{\theta=\varphi\circ\hat{\sigma}(x)} = 0 \tag{4.24}$$

for all $x \in \mathcal{X}$ and all $v \in T_\theta\Theta$.

Proof. Assume that $\hat{\sigma}$ is efficient. Since φ is a coordinate mapping, we obtain from (4.21)

$$\log(\nabla_{\mathfrak{g}}\varphi^l_{\hat{\sigma}})_{|\xi=\hat{\sigma}(x)} = \varphi^l \circ \hat{\sigma} - \mathbb{E}_{\hat{\sigma}(x)}(\varphi^l \circ \hat{\sigma} = \langle l, b^{\varphi}_{\hat{\sigma}}(x)\rangle = 0$$

since $\hat{\sigma}$ is unbiased. Comparing the LHS of the above equality with the LHS of (4.24) for $d\mathbf{p}(v) = \nabla_{\mathfrak{g}} d\varphi^l = \nabla_{\mathfrak{g}} d\varphi^l_{\hat{\sigma}}$ we obtain immediately Theorem 4.18. □

4.5. **Consistency of MLE.** Assume that $P_1 = P \subset \mathcal{P}(\mathcal{X})$ is a 2-integrable statistical model that contains an unknown probability measure μ_u governing distribution of random instance $x_i \in \mathcal{X}$. Then $P_n = \{\mu^n | \, \mu \in P\} \subset \mathcal{P}(\mathcal{X}^n)$ is a 2-integrable statistical model containing probability measure μ^n_u that governs the distribution of i.i.d. of random instances $(x_1, \cdots, x_n) \in \mathcal{X}^n$. Denote by $\mathfrak{g}_n$ the Fisher metric on the statistical model P_n. The map

$\lambda_n : P_1 \to P_n$, $\mu \mapsto \mu^n$, is a 1-1 map, it is the restriction of the differentiable map, also denoted by λ_n

$$\lambda_n : \mathcal{S}(\mathcal{X}) \to \mathcal{S}(\mathcal{X}^n),\ \lambda_n(\mu) = \mu^n.$$

It is easy to see that for any $v \in T_\mu P$ we have

$$\mathfrak{g}_n(d\lambda_n(v), d\lambda_n(v)) = n \cdot \mathfrak{g}_1(v, v). \tag{4.25}$$

This implies that the lower bound in the Cramér-Rao inequality (4.23) for unbiased estimators $\hat{\sigma}_n^* := \lambda_n^{-1} \circ \hat{\sigma}_n : (\mathcal{X})^n \to P$ converges to zero and there is a hope that MLE is asymptotically accurate as n goes to infinity. Now we shall give a concept of a consistent sequence φ-estimators that formalizes the notion of asymptotically accurate sequence of estimators $\hat{\sigma}_k^* : \mathcal{X}^k \to P_1$ and using it to examine MLE. [12].

Definition 4.19. (cf. [IH1981, p. 30], [Borovkov1998, Definition 1, p. 53]) Let $P \subset \mathcal{P}(\mathcal{X})$ be a statistical model and φ a $\mathbb{R}^n$-valued function on P. A sequence of φ-estimators $\hat{\sigma}_k^* : (\mathcal{X})^k \to P \to \mathbb{R}^n$ is called *a consistent sequence of φ-estimators for the value* $\varphi(\mu_u)$ if for all $\delta > 0$ we have

$$\lim_{k\to\infty} \mu_u^k(\{\mathbf{x} \in \mathcal{X}^{\mathbf{k}} :\ |\varphi \circ \hat{\sigma}_{\mathbf{k}}^*(\mathbf{x}) - \varphi(\mu_{\mathbf{u}})| > \delta\}) = \mathbf{0}. \tag{4.26}$$

Under quite general conditions on the density functions of a statistical model $P \subset \mathcal{P}(\mathcal{X})$ of dominated measures, see e.g. [IH1981, Theorem 4.3, p. 36] the sequence of MLE's is consistent.

4.6. **Conclusion.** In this lecture we derive a natural geometry on a statistical model $P \subset \mathcal{P}(\mathcal{X})$ regarding it as a subset in the Banach space $\mathcal{S}(\mathcal{X})$ with the total variation norm. Since P is non-linear, we linearize estimator $\hat{\sigma}_k^* : \mathcal{X}^k \to P_k = \lambda_k(P)$ by composing it with a map $\varphi_k := \lambda_k^{-1} \circ \varphi : P_k \to \mathbb{R}^n$. Then we define the MSE and variance of φ_k-estimator $\varphi_k \circ \hat{\sigma}_k$, which can be estimated using the Cramér-Rao inequality. It turns out that the efficient unbiased estimator w.r.t. MSE is MLE. The notion of a φ-estimator allows to define the notion of a consistent sequence of φ-estimators that formalizes the notion of asymptotically accurate estimators.

5. Consistency of a learning algorithm

In the last lecture we learned the important concept of a consistent sequence of φ-estimators, which formalizes the notion of the asymptotic accuracy of a sequence of estimators, and applied this concept to MLE.

In this lecture we extend the concept of a consistent sequence of estimators to the notion of consistency of a learning algorithm in a unified learning model that encompasses models for density estimation and discriminative models of supervised learning. The concept of consistency of a learning algorithm formalizes the notion of learnability of a learning machine. In

[12] the notion of a consistent sequence of estimators that is asymptotically accurate has been suggested by Fisher in [Fisher1925]

particular we relate this notion with the problem of overfitting. [13] Then we shall apply this concept to examine the success of ERM algorithm in binary classification problems.

5.1. **Consistent learning algorithm and its sample complexity.**

In *a unified learning model* $(\mathcal{Z}, \mathcal{H}, L, P)$ we are given a measurable space $\mathcal{Z}$ (e.g. $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ in classification and regression problems), a hypothesis class $\mathcal{H}$ (e.g., the domain Θ in the density estimation problem in Subsection 3.1) together with an instantaneous loss function $L : \mathcal{Z} \times \mathcal{H} \times P \to \mathbb{R}_+$, where $P \subset \mathcal{P}(\mathcal{Z})$ is a statistical model. We define the expected risk function as follows

$$R^L : \mathcal{H} \times P \to \mathbb{R}_+,\ (h, \mu) \mapsto \mathbb{E}_\mu L(z, h, \mu).$$

We also set for $\mu \in P$

$$R^L_\mu : \mathcal{H} \to \mathbb{R},\ h \mapsto R^L(h, \mu).$$

A learning algorithm is a map

$$A : \bigcup_n \mathcal{Z}^n \to \mathcal{H},\ S \mapsto h_S$$

where S is distributed by some unknown $\mu^n \in P_n = \lambda_n(P)$, cf. (2.1).

Example 5.1. In a unified learning model $(\mathcal{Z}, \mathcal{H}, L, P)$ there is always a learning algorithm using ERM, if $\mathcal{H}$ is compact. For an element $S_n = (z_1, \cdots, z_n) \in \mathcal{Z}^n$ we denote by $[S_n]$ the subset $\{z_1, \cdots, z_n\} \subset \mathcal{Z}$. For $S_n \in \mathcal{Z}^n$ we define the empirical risk $\hat{R}^L_{S_n} : \mathcal{H} \to \mathbb{R}$ by the formula: $\hat{R}^L_{S_n}(h) = \sum_{z \in [S_n]} L(z, h)$. The ERM algorithm A_{erm} for $(\mathcal{Z}, \mathcal{H}, L, P)$ is defined as follows

$$A_{erm}(S_n) := \arg_{h \in \mathcal{H}} \min \hat{R}^L_{S_n}(h). \tag{5.1}$$

Observe that $\arg_{h \in \mathcal{H}} \min \hat{R}^L_{S_n}(h)$ may not exist if $\mathcal{H}$ is not compact. In this case we denote by $A_{erm}(S_n)$ any hypothesis that satisfies the inequality $\hat{R}^L_S(A_{erm}(S_n)) - \inf_{\mathfrak{h} \in \mathcal{H}} \hat{R}^L_{S_n}(h) = O(n^{-k})$, where $k \geq 1$ depends on the computational complexity of defining A_{erm}. In other words, A_{erm} is only asymptotically ERM. [14]

Note that the ERM algorithm A_{erm} does not depend on P but the knowledge of P is important for defining and understanding the expected loss function R^L_μ. In general we expect a close relationship between $\mathcal{H}$, a representation of a discriminative model, and P, a representation of a generative model.

Recall that $R^L_{\mu,\mathcal{H}} = \inf_{h \in \mathcal{H}} R^L_\mu(h)$, see (2.10).

[13] In ML community, one says that a learning algorithm is consistent if it may commit no error on the example of the training data [MRT2012, p.5].

[14] Vapnik considered ERM algorithm also for the case that $\hat{R}^L_{S_n}$ may not reach infimum, using slightly different language than ours. In [SSBD2014] the authors assumed that a minimizer of $\hat{R}^L_{S_n}$ always exists.

Definition 5.2. A learning algorithm A in a model $(\mathcal{Z}, \mathcal{H}, L, P)$ is called *consistent*, if for any $\varepsilon \in (0,1)$ and for every probability measure $\mu \in P$,

$$\lim_{n\to\infty} \mu^n\{S \in \mathcal{Z}^n : |R^L_\mu(A(S)) - R^L_{\mu,\mathcal{H}}| \geq \varepsilon\} = 0 \tag{5.2}$$

(2) A learning algorithm A is called *uniformly consistent*, if (5.2) converges to zero uniformly on P, i.e., for any $(\varepsilon, \delta) \in (0,1)^2$ there exists a number $m_A(\varepsilon, \delta)$ such that for any $\mu \in P$ and any $m \geq m_A(\varepsilon, \delta)$ we have

$$\mu^m\{S \in \mathcal{Z}^m : |R^L_\mu(A(S)) - R^L_{\mu,\mathcal{H}}| \leq \varepsilon\} \geq 1 - \delta. \tag{5.3}$$

If(5.3) holds we say that A predicts with accuracy ε and confidence $1-\delta$ using m samples.

We characterize the uniform consistency of a learning algorithm A via the notion of *the sample complexity function of* A.

Definition 5.3. Let A be an algorithm on $(\mathcal{Z}, \mathcal{H}, L, P)$ and $m_A(\varepsilon, \delta)$ the minimal number $m_0 \in \mathbb{R}_+ \cup \infty$ such that (5.3) holds for any $m \geq m_0$. Then the function $m_A : (\varepsilon, \delta) \mapsto m_A(\varepsilon, \delta)$ is called *the sample complexity function of algorithm* A.

Clearly a learning algorithm A is uniformly consistent if and only if m_A takes value in $\mathbb{R}_+$. Furthermore, A is consistent if and only if the sample function of A on the sub-model $(\mathcal{Z}, \mathcal{H}, L, \mu)$ takes values in $\mathbb{R}_+$ for all $\mu \in P$.

Example 5.4. Let $(\mathcal{Z}, \mathcal{H}, L, P)$ be a unified learning model. Denote by $\pi_n : \mathcal{Z}^\infty \to \mathcal{Z}^n$ the map $(z_1, \cdots, z_\infty) \mapsto (z_1, \cdots, z_n)$. A sequence $\{S_\infty \in \mathcal{Z}^\infty\}$ of i.i. instances distributed by $\mu \in P$ is called *overfitting*, if there exist $\varepsilon \in (0,1)$ such that for all n we have

$$|R^L_\mu(A_{erm}(\pi_n(S_\infty)) - R^L_{\mu,\mathcal{H}}| \geq \varepsilon. \tag{5.4}$$

Thus A_{erm} is consistent, if and only if the set of all overfitting sequences $S_\infty \in \mathcal{Z}^\infty$ has μ^∞-zero measure [15], equivalently, if (5.2) holds, for any $\mu \in P$. In Example 2.11 we showed the existence of a measure μ_f on $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ such that any sequence $S_\infty \in \mathcal{Z}^\infty$ distributed by μ_f^∞ is overfitting. Hence the unified learning model in Example 2.11) for any P containing μ_f is not consistent using A_{erm}.

The following simple Lemma reduces a proof of the uniform consistency of A_{erm} to the proof of the convergence in probability of $R^L_S(h)$ to $R^L_\mu(h)$ (the weak law of large numbers) that is uniform on $\mathcal{H}$.

Lemma 5.5. *Assume that for any $(\varepsilon, \delta) \in (0,1)^2$ there exists a function $m_{\mathcal{H}}(\varepsilon, \delta)$ taking value in $\mathbb{R}_+$ such that for all $m \geq m_{\mathcal{H}}(\varepsilon, \delta)$ for all $\mu \in P$ and for all $h \in \mathcal{H}$ we have*

$$\mu^m\{S \in \mathcal{Z}^m : |\hat{R}^L_S(h) - R^L_\mu(h)| < \varepsilon\} \geq 1 - \delta \tag{5.5}$$

then A_{erm} is uniformly consistent.

[15] we refer to [AJLS2017, p. 293] and [Bogachev2007, p. 188] for definition of μ^∞

Proof. For the simplicity of the exposition we assume first that $A_{erm}(S_n) = \arg\min_{h\in\mathcal{H}} R^L_{S_n}(h)$. The argument for the general case can be easily adapted from this simple case.

Given $m \geq m_{\mathcal{H}}(\varepsilon/2, \delta/2)$, $\mu \in P$ and $h_\varepsilon \in \mathcal{H}$ such that $R^L_\mu(h_\varepsilon) \leq R^L_{\mu,\mathcal{H}} + \varepsilon$ we have

$$\mu^m\{S \in \mathcal{Z}^m : |R^L_\mu(A_{erm}(S)) - R_{\mu,\mathcal{H}}| \leq 2\varepsilon\} \geq$$
$$\mu^m\{S \in \mathcal{Z}^m : R^L_\mu(A_{erm}(S)) \leq R^L_\mu(h_\varepsilon) + \varepsilon\} \geq$$
$$\mu^m\{S \in \mathcal{Z}^m : R^L_\mu(A_{erm}(S)) \leq \hat{R}^L_S(h_\varepsilon) + \frac{\varepsilon}{2} \,\&\, |R^L_S(h_\varepsilon) - R^L_\mu(h_\varepsilon)| < \frac{\varepsilon}{2}\} \geq$$

$$\mu^n\{S \in \mathcal{Z}^m : R^L_\mu(A_{erm}(S)) \leq \hat{R}^L_S(h_\varepsilon) + \varepsilon\} - \frac{\delta}{2} \geq$$
$$\mu^n\{S \in \mathcal{Z}^n : |R^L_\mu(A_{erm}(S)) - \hat{R}^L_S(A(S))| \leq \frac{\varepsilon}{2}\} - \frac{\delta}{2} \geq 1 - \delta$$

since $\hat{R}^L_S(A(S)) < \hat{R}^L_S(h_\varepsilon)$. This completes the proof of Lemma 5.5. □

Theorem 5.6. *(cf.* [SSBD2014, Corollary 4.6, p.57]*) Let $(\mathcal{Z}, \mathcal{H}, L, \mathcal{P}(\mathcal{Z}))$ be a unified learning model. If $\mathcal{H}$ is finite and $L(\mathcal{Z}\times\mathcal{H}) \subset [0, c] \not\ni \infty$ then the ERM algorithm is uniformly consistent.*

Proof. By Lemma 5.5, it suffices to find for each $(\varepsilon, \delta) \in (0,1)\times(0,1)$ a number $m_{\mathcal{H}}(\varepsilon, \delta)$ such that for all $m \geq m_{\mathcal{H}}(\varepsilon, \delta)$ and for all $\mu \in \mathcal{P}(\mathcal{Z})$ we have

$$\mu^m\big(\bigcap_{h\in\mathcal{H}} \{S \in \mathcal{Z}^m : |\hat{R}^L_S(h) - R^L_\mu(h)| \leq \varepsilon\}\big) \geq 1 - \delta. \tag{5.6}$$

In order to prove (5.6) it suffices to establish the following inequality

$$\sum_{h\in\mathcal{H}} \mu^m\big(\{S \in \mathcal{Z}^m | \, |\hat{R}^L_S(h) - R^L_\mu(h)| > \varepsilon\}\big) < \delta. \tag{5.7}$$

Since $\#H < \infty$, it suffices to find $m_{\mathcal{H}}(\varepsilon, \delta)$ such that when $m \geq m_{\mathcal{H}}(\varepsilon, \delta)$ each summand in RHS of (5.7) is small enough. For this purpose we shall apply the well-known Hoeffding inequality, which specifies the rate of convergence in the week law of large numbers, see Subsection B.2.

To apply Hoeffding's inequality to the proof of Theorem 5.6 we observe that for each $h \in \mathcal{H}$

$$\{\theta^h_i(z) := L(h, z) \in [0, c]\}$$

are i.i.d. $\mathbb{R}$-valued random variables on $\mathcal{Z}$. Furthermore we have for any $h \in \mathcal{H}$ and $S = (z_1, \cdots, z_m)$

$$\hat{R}^L_S(h) = \frac{1}{m}\sum_{i=1}^m \theta^h_i(z_i),$$
$$R^L_\mu(h) = \bar{\theta}^h.$$

Hence the Hoeffding inequality implies

$$\mu^m\{S \in \mathcal{Z}^m : |\hat{R}^L_S(h) - R^L_\mu(h)| > \varepsilon\} \leq 2\exp(-2m\varepsilon^2 c^{-2}). \tag{5.8}$$

Now pluging

$$m \geq m_{\mathcal{H}}(\varepsilon, \delta) := \frac{\log(2\#(\mathcal{H})/\delta)}{2\varepsilon^2 c^{-2}}$$

in (5.8) we obtain (5.7). This completes the proof of Theorem 5.6.

□

Definition 5.7. The function $m_{\mathcal{H}} : (0,1)^2 \to \mathbb{R}$ defined by the requirement that $m_{\mathcal{H}}(\varepsilon, \delta)$ is the least number for which (5.6) holds is called *the sample complexity of a (unified) learning model* $(\mathcal{Z}, \mathcal{H}, L, P)$.

Remark 5.8. (1) We have proved that the sample complexity of the algorithm A_{erm} is upper bounded by the sample complexity $m_{\mathcal{H}}$ of $(\mathcal{Z}, \mathcal{H}, L, P)$.

(2) The definition of the sample complexity $m_{\mathcal{H}}$ in our lecture is different from the definition of the sample complexity $m_{\mathcal{H}}$ in [**?**, Definition 3.1, p. 43], which is equivalent to the notion of the sample complexity m_A of a learning algorithm in our lecture.

5.2. **Uniformly consistent learning and VC-dimension.** In this subsection we shall examine sufficient and necessary conditions for the existence of a uniformly consistent learning algorithm on a unified learning model with infinite hypothesis class $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$. First we prove a version of No-Free-Lunch theorem which asserts that there is no uniformly consistent learning algorithm on a learning model with a very large hypothesis class and a very large statistical model. Denote by $\mathcal{P}_{\mathcal{X}}(\mathcal{X} \times \mathcal{Y})$ the set of all probability measures $(\Gamma_f)_*(\mu_{\mathcal{X}}) \in \mathcal{P}(\mathcal{X} \times \mathcal{Y})$ where $f : \mathcal{X} \to \mathcal{Y}$ is a measurable map and $\mu_{\mathcal{X}} \in \mathcal{P}(\mathcal{X})$.

Theorem 5.9. *Let $\mathcal{X}$ be an infinite domain set, $\mathcal{Y} = \{0, 1\}$ and $L^{(0-1)}$ the 0-1 loss function. Then there is no uniformly consistent learning algorithm on a unified learning model $(\mathcal{X} \times \mathcal{Y}, \mathcal{H}, L^{(0-1)}, \mathcal{P}_{\mathcal{X}}(\mathcal{X} \times \mathcal{Y}))$.*

Proof. To prove Theorem 5.9 it suffices to show that $m_A(\varepsilon, \delta) = \infty$ for $(\varepsilon, \delta) = (1/8, 1/8)$ and any learning algorithm A. Assume the opposite, i.e., $m_A(1/8, 1/8) = m < \infty$. Then we shall find $\mu(m) \in \mathcal{P}_{\mathcal{X}}(\mathcal{X} \times \mathcal{Y})$ such that Equation (5.3) violates for m, $\mu(m)$ and $(\varepsilon, \delta) = (1/8, 1/8)$.

To describe the measure $\mu(m)$ we need some notations. For a subset $C[k] \subset \mathcal{X}$ of k-elements let $\mu_{\mathcal{X}}^{C[k]} \in \mathcal{P}(\mathcal{X})$ be defined by

$$\mu_{\mathcal{X}}^{C[k]}(B) := \frac{\#(B \cap C[k])}{k} \text{ for any } B \subset \mathcal{X}. \tag{5.9}$$

For a map $f : \mathcal{X} \to \mathcal{Y}$ we set

$$\mu_f^{C[k]} := (\Gamma_f)_* \mu_{\mathcal{X}}^{C[k]} \in \mathcal{P}_{\mathcal{X}}(\mathcal{X} \times \mathcal{Y}).$$

Lemma 5.10. *Assume that $\mathcal{X}$, $\mathcal{Y}$ are finite sets and $\#\mathcal{X} \geq n + 1$. For $f \in \mathcal{Y}^{\mathcal{X}}$ set $\mu_f := \mu_f^{\mathcal{X}} \in \mathcal{P}_{\mathcal{X}}(\mathcal{X} \times \mathcal{Y})$. Then for any learning algorithm $A : S \mapsto A_S$, any $f \in \mathcal{Y}^{\mathcal{X}}$ we have*

$$\int_{\mathcal{Y}^{\mathcal{X}}} \int_{(\mathcal{X} \times \mathcal{Y})^n} R_{\mu_f}^{(0-1)}(A_S)\, d(\mu_f^n)(S) d\mu_{\mathcal{Y}^{\mathcal{X}}}^{\mathcal{Y}^{\mathcal{X}}}(f) \geq (1 - \frac{1}{\#\mathcal{Y}})(1 - \frac{n}{\#\mathcal{X}}) \tag{5.10}$$

Proof of Lemma 5.10. We set for $S \in (\mathcal{X} \times \mathcal{Y})^n$

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

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

Note that S is distributed by μ_f^n means that $S = \{(x_1, f(x_1)), \cdots, (x_n, f(x_n))$, so S is essentially distributed by the uniform probability measure $(\mu_{\mathcal{X}}^{\mathcal{X}})^n$. Let us compute and estimate the double integral in the LHS of (5.10) using (2.9) and the Fubini theorem.

$$\begin{aligned}
\mathbb{E}_{\mu_{\mathcal{Y}^{\mathcal{X}}}^{\mathcal{Y}^{\mathcal{X}}}}\Big(\mathbb{E}_{\mu_f^n}\big(R_{\mu_f}^{(0-1)}(A_S)\big)\Big) &= \frac{1}{\#\mathcal{X}}\mathbb{E}_{\mu_{\mathcal{Y}^{\mathcal{X}}}^{\mathcal{Y}^{\mathcal{X}}}}\Big(\mathbb{E}_{(\mu_{\mathcal{X}}^{\mathcal{X}})^n}\big(\sum_{x\in\mathcal{X}}(1-\delta_{f(x)}^{A_S(x)})\big)\Big) \\
&\geq \frac{1}{\#\mathcal{X}}\mathbb{E}_{\mu_{\mathcal{Y}^{\mathcal{X}}}^{\mathcal{Y}^{\mathcal{X}}}}\Big(\mathbb{E}_{(\mu_{\mathcal{X}}^{\mathcal{X}})^n}\big(\sum_{x\notin\mathcal{X}_S}(1-\delta_{f(x)}^{A_S(x)})\big)\Big) \\
&= \frac{1}{\#\mathcal{X}}\mathbb{E}_{(\mu_{\mathcal{X}}^{\mathcal{X}})^n}\Big(\sum_{x\notin\mathcal{X}_S}\mathbb{E}_{\mu_{\mathcal{Y}^{\mathcal{X}}}^{\mathcal{Y}^{\mathcal{X}}}}(1-\delta_{f(x)}^{A_S(x)})\Big) \\
&= \frac{1}{\#\mathcal{X}}\mathbb{E}_{(\mu_{\mathcal{X}}^{\mathcal{X}})^n}\Big(\#[\mathcal{X}\setminus\mathcal{X}_S]\cdot\big(1-\frac{1}{\#\mathcal{Y}}\big)\Big) \\
&\overset{\text{since } \#[\mathcal{X}\setminus\mathcal{X}_S]\geq\#\mathcal{X}-n}{\geq} (1-\frac{1}{\#\mathcal{Y}})(1-\frac{n}{\#\mathcal{X}}). \qquad (5.11)
\end{aligned}$$

This completes the proof of Lemma 5.10. □

Continuation of the proof of Theorem 5.9. It follows from Lemma 5.10 that there exists $f \in \mathcal{Y}^{\mathcal{X}}$ such that, denoting $\mu := \mu_f^{C[2m]}$, we have

$$\int_{(\mathcal{X}\times\mathcal{Y})^m} R_\mu^{(0-1)}(A_S)\, d\mu^m = \int_{(C[2m]\times\mathcal{Y})^m} R_\mu^{(0-1)}(A_S)\, d(\mu^m)(S) \geq \frac{1}{4}. \qquad (5.12)$$

Since $0 \leq R_\mu^{(0-1)} \leq 1$ we obtain from (5.12)

$$\mu^m\{S \in (\mathcal{X}\times\mathcal{Y})^m |\, R_\mu^{(0-1)}\big(A(S)\big) \geq \frac{1}{8}\} > \frac{1}{8}.$$

This implies that (5.3) does not hold for $(\varepsilon, \delta) = (1/8, 1/8)$, for any m and $\mu(m) = \mu_f^{C[m]}$. This proves Theorem 5.9. □

Remark 5.11. In the proof of Theorem 5.9 we showed that if there is a subset $C \subset \mathcal{X}$ of size $2m$ and the restriction of $\mathcal{H}$ to C is the full set of functions in $\{0,1\}^C$ then (5.3) does not hold for any learning algorithm A, $m \in \mathbb{N}$ and $(\varepsilon, \delta) = (1/8, 1/8)$. In other words we cannot predict a hypothesis in $\mathcal{H} = \mathcal{Y}^{\mathcal{X}}$ of a learning model $(\mathcal{X}, \mathcal{H}, L^{(0-1)}, \mathcal{P}_{\mathcal{X}}(\mathcal{X}\times\mathcal{Y}))$ with accuracy ε and confidence $1-\delta$ using sample of size m if $\#\mathcal{X} \geq 2m$. This motivates the following Definition.

Definition 5.12. (1) A hypothesis class $\mathcal{H} \subset \{0,1\}^{\mathcal{X}}$ *shatters* a finite subset $C \subset \mathcal{X}$ if $\#\mathcal{H}|_C = 2^{\#C}$.

(2) *The VC-dimension* of a hypothesis class $\mathcal{H} \subset \{0,1\}^{\mathcal{X}}$, denoted by $VC\dim(\mathcal{H})$, 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 5.13. Let $\mathcal{H}$ be the class of intervals in the real line, namely,

$$\mathcal{H} = \{1_{(a,b)} : a < b \in \mathbb{R}\},$$

where $1_{(a,b)} : \mathbb{R} \to \{0,1\}$ is the indicate function of the interval (a,b). Take the set $C = \{1,2\}$. Then, $\mathcal{H}$ shatters C, since all all the functions in the set $\{1,2\}^{(0,1)}$ can be obtained as the restriction of some function from $\mathcal{H}$ to C. Hence $VC\dim(H) \geq 2$. Now take an arbitrary set $C = \{c_1 < c_2 < c_3\}$ and the corresponding labeling $(1,0,1)$. Clearly this labeling cannot be obtained by an interval: Any interval $h_{(a,b)}$ that contains c_1 and c_3 (and hence labels c_1 and c_3 with the value 1) must contain c_2 (and hence it labels c_2 with 0). Hence $\mathcal{H}$ does not shatter C. We therefore conclude that $VC\dim(\mathcal{H}) = 2$. Note that $\mathcal{H}$ has infinitely many elements.

Exercise 5.14 (VC-Threshold functions)**.** Consider the hypothesis class $\mathcal{F} \subset \{-1,1\}^{\mathbb{R}}$ of all threshold functions $sgn^b : \mathbb{R} \to \mathbb{R}$, where $b \in \mathbb{R}$, defined by

$$sgn^b(x) := sgn(x-b)$$

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

In Remark 5.11 we observed that the finiteness of $VC\dim\mathcal{H}$ is a necessary condition for the existence of a uniformly consistent learning algorithm on a unified learning model $(\mathcal{X}, \mathcal{H}, L^{(0-1)}, \mathcal{P}_{\mathcal{X}}(\mathcal{X} \times \mathcal{Y}))$. In the next section we shall show that the finiteness of $VC\dim\mathcal{H}$ is also a sufficient condition for the uniform consistency of A_{erm} on $(\mathcal{X}, \mathcal{H}, L^{0-1}, \mathcal{P}_{\mathcal{X}}(\mathcal{X} \times \mathcal{Y}))$.

5.3. **Fundamental theorem of binary classification.**

Theorem 5.15 (Fundamental theorem of binary classification)**.** *A learning model $(\mathcal{X}, \mathcal{H} \subset \{0,1\}^{\mathcal{X}}, L^{(0-1)}, \mathcal{P}(\mathcal{X} \times \{0,1\}))$ has a uniformly consistent learning algorithm, if and only if $VC\dim(\mathcal{H}) < \infty$.*

Outline of the proof. Note that the "only if" assertion of Theorem 5.15 follows from Remark 5.11. Thus we need only to prove the "if" assertion. By Lemma 5.5 it suffices to show that if $VC\dim(\mathcal{H}) = k < \infty$ then $m_{\mathcal{H}}(\varepsilon,\delta) < \infty$ for all $(\varepsilon,\delta) \in (0,1)^2$. In other words we need to find a lower bound for the LHS of (5.5) in terms of the VC-dimension, which is an upper bound of the RHS of (5.5), when $\varepsilon \in (0,1)$ and m is sufficiently large. This shall be done in three steps.

In step 1, setting $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, and omitting superscript of the risk function R, we use the Markov inequality to obtain

$$\mu^m\{S \in \mathcal{Z}^m : |R_\mu(h) - \hat{R}_S(h)| < a\} \leq \frac{\mathbb{E}_{\mu^m}|R_\mu(h) - R_S(h)|}{a} \tag{5.13}$$

for any $a > 0$ and any $h \in \mathcal{H}$.

In step 2 we define *the growth function* $\Gamma_{\mathcal{H}}(m) : \mathbb{N} \to \mathbb{N}$ and use it to upper bound the RHS of (5.13). Namely we shall prove the following

$$\mathbb{E}_{\mu^m}(\sup_{h\in\mathcal{H}} |R_\mu(h) - \hat{R}_S(h)| \leq \frac{4 + \sqrt{\log\big(\Gamma_{\mathcal{H}}(2m)\big)}}{\delta\sqrt{2m}}) \geq 1 - \delta \tag{5.14}$$

for every $\mu \in \mathcal{P}(\mathcal{X} \times \mathcal{Y})$ and any $\delta \in (0, 1)$. The proof of (5.14) is delicate and can be found in [SSBD2014, p. 76-77].

Definition 5.16 (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 $\Gamma(0) = 1$.

Example 5.17. Consider the set $\mathcal{F} : \{sign^b | \, b \in \mathbb{R}\}$ of all threshold functions. 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 $\Gamma_{\mathcal{F}}(n) \geq n + 1$.

Exercise 5.18. Show that $\Gamma_{\mathcal{F}}(n) = n + 1$ for the set $\mathcal{F}$ of all threshold functions.

In step 3 we use the following Vapnik-Chervonenski-Lemma, also known as Sauer's Lemma, whose proof can be found in [SSBD2014, p. 74-75].

Lemma 5.19. *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 $\Gamma_{\mathcal{H}}(n) \leq (en/d)^d$.

It follows from Lemma 5.19 that for any $(\varepsilon, \delta) \in (0, 1)^2$ there exists m such that

$$\frac{4 + \sqrt{\log\big(\Gamma_{\mathcal{H}}(2m)\big)}}{\delta\sqrt{2m}} < \varepsilon$$

and therefore by (5.14) for any $(\varepsilon, \delta) \in (0, 1)^2$ the value $m_{\mathcal{H}}(\varepsilon, \delta)$ is finite. This completes the proof of Theorem 5.15.

5.4. **Conclusions.** In this lecture we define the notion of the (uniform) consistency of a learning algorithm A on a unified learning model $(\mathcal{Z}, \mathcal{H}, L, P)$ and characterize this notion via the sample complexity of A. We relate the consistency of the ERM algorithm with the uniform convergence of the law of large numbers over the parameter space $\mathcal{H}$ and use it to prove the uniform consistency of ERM in the binary classification problem $(\mathcal{X}, \mathcal{H} \subset \{0,1\}^{\mathcal{X}}, L^{(0-1)}, \mathcal{P}(\mathcal{X} \times \{0,1\})$. We show that the finiteness of the VC-dimension of $\mathcal{H}$ is a necessary and sufficient condition for the existence of a uniform consistent learning algorithm on a binary classification problem $(\mathcal{X} \times \{0,1\}, \mathcal{H} \subset \{0,1\}^{\mathcal{X}}, L^{(0-1)}, \mathcal{P}_{\mathcal{X}}(\mathcal{X} \times \{0,1\})$.

6. Generalization ability of a learning machine and model selection

In the last lecture we measured the consistency of a learning algorithm A in a unified learning model $(\mathcal{Z}, \mathcal{H}, L, P)$ via the sample complexity function $m_A : (0,1)^2 \to \mathbb{R}_+ \cup \infty$. The sample complexity $m_A(\varepsilon, \delta)$ is the less number of samples that A requires in order to make a prediction with ε accuracy and $(1-\delta)$ confidence. In 1984 Valiant suggested a PAC-model of learning, which corresponds to the notion of uniform (w.r.t. P) consistency of A, which has moreover to be efficiently computable, i.e. the function $m_A(\varepsilon, \delta)$ must be polynomial in ε^{-1} and δ^{-1}. Furthermore Valiant also requires that A is efficiently computable, which can be expressed in terms of the computational complexity of A, see [SSBD2014, Chapter 8] for discussion on running time of A.

Thus, given a learning machine $(\mathcal{Z}, \mathcal{H}, L, P, A)$, where A is a learning algorithm, the generalization ability of $(\mathcal{Z}, \mathcal{H}, L, P, A)$ is measured in terms of the family of sample complexities $\{m_{A,\mu}(\varepsilon, \delta) |\ \mu \in P\}$ of the learning machines $(\mathcal{Z}, \mathcal{H}, L, \mu \in P, A)$ and the computational complexity of A. In the previous lecture, Lemma 5.5, we gave upper bounds for the sample complexity $m_{A_{erm}}(\varepsilon, \delta)$ in terms of the sample complexity $m_{\mathcal{H}}(\varepsilon/2, \delta/2)$ of the learning model. Then we showed that in a binary classification problem the sample complexity $m_{\mathcal{H}}$ takes finite values if and only if the VC-dimension of $\mathcal{H}$ is finite.

Today we shall discuss two further methods of upper bounds for the sample complexities $m_{\mathcal{H}}$ and $m_{A_{erm}}$ of some important learning models. Then we discuss the problem of learning model selection.

6.1. **Covering number and sample complexity.** In the binary classification problem of supervised learning the VC-dimension is a combinatorial characterization of the hypothesis class $\mathcal{H}$, which carries no topology, since the domain $\mathcal{X}$ and the target space $\mathcal{Y}$ are discrete. The expected zero-one loss function is therefore the preferred choice of a risk function. In [CS2001] Cucker-Smale estimated the sample complexity $m_{\mathcal{H}}$ of a discriminative model for a regression problem with the MSE as the expected loss function.

Before stating Cucker-Smale's results we introduce necessary notations.

• $\mathcal{X}$ is a topological space and ρ denotes a Borel measure on $\mathcal{X} \times \mathbb{R}^n$.

• Let $C_n(\mathcal{X})$ be the Banach space of continuous bounded $\mathbb{R}^n$-valued functions on $\mathcal{X}$ with the norm [16]

$$||f||_{C^0} = \sup_{x \in \mathcal{X}} ||f(x)||.$$

• For $f \in C_n(\mathcal{X})$ let $f_Y \in C_n(\mathcal{X} \times \mathbb{R}^n)$ denote the function

$$f_Y(x, y) := f(x) - y.$$

Then $MSE_\rho(f) = \mathbb{E}_\rho(||f_Y||^2)$, see (2.2) and (2.3).

• For a function $g \in C_n(\mathcal{X} \times \mathbb{R}^n)$ denote by $V_\rho(g)$ its variance, see (4.16),

$$V_\rho(g) = \mathbb{E}_\rho(||g - \mathbb{E}_\rho(g)||^2) = \mathbb{E}_\rho(||g||^2) - ||\mathbb{E}_\rho g||^2.$$

• For a compact hypothesis class $\mathcal{H} \subset C_n(\mathcal{X})$ define the following quantities

$$MSE_{\rho,\mathcal{H}}(f) := MSE_\rho(f) - \min_{f \in \mathcal{H}} MSE_\rho(f), \tag{6.1}$$

which is called *the estimation error of* f, or *the sample error of* f [CS2001], and

$$V_\rho(\mathcal{H}) := \sup_{f \in \mathcal{H}} V_\rho(f_Y),$$

$\mathcal{N}(\mathcal{H}, s) := \min\{l \in \mathbb{N} |$ there exists l disks in $\mathcal{H}$ with radius s covering $\mathcal{H}\}$.

• For $S = (z_1, \cdots, z_m) \in \mathcal{Z}^m := (\mathcal{X} \times \mathcal{Y})^m$, where $z_i = (x_i, y_i)$, denote by f_S the minimizer of the function $MSE_S : \mathcal{H} \to \mathbb{R}$ such that

$$MSE_S(f) := \frac{1}{m} \sum_{i=1}^{m} ||f(x_i) - y_i||^2.$$

The existence of f_S follows from the compactness of $\mathcal{H}$ and the continuity of the functional MSE_S on $\mathcal{H}$.

Theorem 6.1. *([CS2001, Theorem C]) Let $\mathcal{H}$ be a compact subset of $C(\mathcal{X}) := C_1(\mathcal{X})$. Assume that for all $f \in \mathcal{H}$ we have $|f(x) - y| \leq M$ ρ-almost everywhere, where ρ is a probability measure on $\mathcal{X} \times \mathbb{R}$. Then for all $\varepsilon > 0$*

$$\rho^m\{S \in \mathcal{Z}^m |\, MSE_{\rho,\mathcal{H}}(f_S) \leq \varepsilon\} \geq 1 - \mathcal{N}(\mathcal{H}, \frac{\varepsilon}{16M}) 2e^{-\frac{m\varepsilon^2}{8(4V_\rho(\mathcal{H}) + \frac{1}{3}M^2\varepsilon)}}. \tag{6.2}$$

Theorem 6.1 implies that for any $n < \infty$ the ERM algorithm is consistent on the unified learning model $(\mathcal{X}, \mathcal{H} \subset C_n(\mathcal{X}), L_2, \mathcal{P}_B(\mathcal{X} \times \mathbb{R}^n))$, where L_2 is the quadratic loss function, and $\mathcal{P}_B(\mathcal{X} \times \mathbb{R}^n)$ denotes the space of Borel measures on the topological space $\mathcal{X} \times \mathbb{R}^n$, if $\mathcal{H}$ is compact. To increase the accuracy of the estimate in (6.2) we need to decrease the radius of the covering balls and therefore increase the number of the covering ball.

[16] In [CS2001, p.8] the authors used the L_∞-norm, but they considered only the subspace of continuous functions

The proof of Theorem 6.1 is based on Lemma 5.5. To prove that the sample complexity of A_{erm} is bounded it suffices to prove that the sample complexity $m_{\mathcal{H}}$ of the learning model $(\mathcal{X}, \mathcal{H} \subset C(\mathcal{X}), L_2, \mathcal{P}_B(\mathcal{X} \times \mathbb{R}^n))$ is bounded.

Outline of the proof of the Cucker-Smale theorem. The strategy of the proof of Theorem 6.1 is similar to that of Fundamental Theorem of binary classification (Theorem 5.15).

In the first step we prove the following Lemma, which gives a lower bound on the rate of the convergence in probability of empirical risk $MSE_S(f)$ to the expected risk $MSE_\rho(f)$ for a given $f \in \mathcal{H}$.

Lemma 6.2. *(*[CS2001, Theorem A, p.8]*) Let $M > 0$ and $f \in C_0(\mathcal{X})$ such that $|f(x) - y| \leq M$ ρ-a.e.. Then for all $\varepsilon > 0$ we have*

$$\rho^m\{S \in \mathcal{Z}^m : |MSE_\rho(f) - MSE_S(f)| \leq \varepsilon\} \geq 1 - 2e^{(-\frac{m\varepsilon^2}{2(\sigma^2 + \frac{1}{3}M^2\varepsilon)})}$$

where $\sigma^2 = V_\rho(f_Y^2)$.

Lemma 6.2 is a version of the inequality (5.8), for which we used the Hoeffding inequality. The Hoeffding inequality does not involve the variance and Cucker-Smale used the Bernstein inequality instead of the Hoeffding inequality, see Appendix B.

In the second step, we reduce the problem of estimating upper bound for the sample complexity $m_{\mathcal{H}}$ to the problem of estimating upper bound for the sample complexities m_{D_j}, where $\{D_j | \ j \in [1, l]\}$ is a cover of $\mathcal{H}$, and using the covering number. Namely we have the following easy inequality

$$\rho^m\{S \in \mathcal{Z}^m | \sup_{f \in \mathcal{H}} |MSE_\rho(f) - MSE_S(f)| \geq \varepsilon\} \leq$$

$$\sum_{j=1}^{l} \rho^m\{S \in \mathcal{Z}^m | \sup_{f \in D_j} |MSE_\rho(f) - MSE_S(f)| \geq \varepsilon\}. \tag{6.3}$$

In the last third step we proof the following

Lemma 6.3. *(*[CS2001, Proposition 3, p. 12]*) Let $f_1, f_2 \in C(\mathcal{X})$. If $|f_j(x) - y| \leq M$ on a set $U \subset \mathcal{Z}$ of full measure for $j = 1, 2$ then for any $S \in U^m$ we have*

$$|MSE_S(f_1) - MSE_S(f_2)|| \leq 4M||f_1 - f_2||_{C_0}.$$

Lemma 6.3 implies that for all $S \in U^m$

$$\sup_{f \in D_j} |MSE_\rho(f) - MSE_S(f)| \geq 2\varepsilon \Longrightarrow |MSE_\rho(f_j) - MSE_S(f_j)| \geq \varepsilon.$$

Combining the last relation with (6.3), we derive the following desired upper estimate for the sample complexity $m_{\mathcal{H}}$.

Proposition 6.4. *Assume that for all $f \in \mathcal{H}$ we have $|f(x)-y| \leq M$ ρ-a.e.. Then for all $\varepsilon > 0$ we have*

$$\rho^m\{\mathbf{z} \in \mathcal{Z}^m : \sup_{f\in\mathcal{H}} |MSE_\rho(f)-MSE_{\mathbf{z}}(f)| \leq \varepsilon\} \geq 1-\mathcal{N}(\mathcal{H}, \frac{\varepsilon}{8M})2e^{(-\frac{m\varepsilon^2}{4(2\sigma^2+\frac{1}{3}M^2\varepsilon)}}$$

where $\sigma^2 = \sup_{f\in\mathcal{H}} V_\rho(f_Y^2)$.

This completes the proof of Theorem 6.1.

Exercise 6.5. Let L_2 denote the instantaneous quadratic loss function in (2.12). Derive from Theorem 6.1 an upper bound for the sample complexity $m_{\mathcal{H}}(\varepsilon, \delta)$ of the learning model $(\mathcal{X}, \mathcal{H} \subset C_n(\mathcal{X}), L_2, \mathcal{P}_B(\mathcal{X}\times\mathbb{R}^n))$, where $\mathcal{H}$ is compact and $\mathcal{P}_B(\mathcal{X}\times\mathbb{R}^n)$ is the space of Borel measures on the topological space $\mathcal{X}\times\mathbb{R}^n$.

Remark 6.6. If the hypothesis class $\mathcal{H}$ in Theorem 6.1 is a convex subset in $\mathcal{H}$ then Cucker-Smale got an improved estimation of the sample complexity $m_{A_{erm}}$ [CS2001, Theorem C*].

6.2. **Rademacher complexities and sample complexity.** Rademacher complexities are more sophisticated complexities that can be used in upper bounds for a "half" of the sample complexity $m_{\mathcal{H}}$ of a learning model but they are sufficient for estimating upper bounds of the sample complexity $m_{A_{erm}}$ of the ERM algorithm.

The *Rademacher complexity of a learning model* $(\mathcal{Z}, \mathcal{H}, L, \mu)$ is defined as the Rademacher complexity of the family $\mathcal{G}_{\mathcal{H}}^L$ of functions

$$\{g_h : \mathcal{Z} \to \mathbb{R},\ g_h(z) = L(z,h) |\, h \in \mathcal{H}\}. \tag{6.4}$$

Definition 6.7 (Rademacher complexity)**.** *The empirical Rademacher complexity* of $\mathcal{G}$ w.r.t. a sample $S = (z_1, \cdots, z_n) \in \mathcal{Z}^n$ is defined as follows

$$\hat{\mathcal{R}}_S(\mathcal{G}) := \mathbb{E}_{(\mu_{\mathbb{Z}_2}^{\mathbb{Z}_2})^n}[\sup_{g\in\mathcal{G}} \frac{1}{n}\sum_{i=1}^n \sigma_i g(z_i)],$$

where $\{\sigma_i \in \mathbb{Z}_2 |\, i \in [1,n]\}$ and $\mu_{\mathbb{Z}_2}^{\mathbb{Z}_2}$ is the counting measure on $\mathbb{Z}_2$, see (5.9). If S is distributed according to a probability measure μ^n on $\mathcal{Z}^n$, then *the Rademacher complexity* of $\mathcal{G}$ w.r.t. μ are given by

$$\mathcal{R}_{n,\mu}(\mathcal{G}) := \mathbb{E}_{\mu^n}[\mathcal{R}_S(\mathcal{G})].$$

The Rademacher n-complexity $\mathcal{R}_n(\mathcal{Z}, \mathcal{H}, L, \mu)$ (resp. the Rademacher empirical n-complexity $R_S(\mathcal{Z}, \mathcal{H}, L)$) is defined to be the complexity $\mathcal{R}_{n,\mu}(\mathcal{G}_{\mathcal{H}}^L)$ (resp. the empirical complexity $\hat{\mathcal{R}}_S(\mathcal{G}_{\mathcal{H}}^L)$), where $\mathcal{G}_{\mathcal{H}}^L$ is the family associated to the model $(\mathcal{Z}, \mathcal{H}, L, \mu)$ by (6.4).

Example 6.8. Let us consider a learning model $(\mathcal{X}\times\mathbb{Z}_2, \mathcal{H} \subset (\mathbb{Z}_2)^{\mathcal{X}}, L^{(0-1)}, \mu)$. For a sample $S = \{(x_1, y_1), \cdots, (x_m, y_m)\} \in (\mathcal{X}\times\mathbb{Z}_2)^m$ denote by $Pr(S)$ the sample $(x_1, \cdots, x_m) \in \mathcal{X}^m$. We shall show that

$$\hat{R}_S(\mathcal{G}_{\mathcal{H}}^{(0-1)}) = \frac{1}{2}\hat{R}_{Pr(S)}(\mathcal{H}). \tag{6.5}$$

Using the identity

$$L^{(0-1)}(x, y, h) = 1 - \delta_y^{h(x)} = \frac{1}{2}(1 - y_i h(x_i))$$

we compute

$$\begin{aligned}
\hat{R}_S(\mathcal{G}_{\mathcal{H}}^{(0-1)}) &= \mathbb{E}_{(\mu_{\mathbb{Z}_2}^{\mathbb{Z}_2})^m}[\sup_{h\in\mathcal{H}} \frac{1}{m}\sum_{i=1}^m \sigma_i \delta_{y_i}^{h(x_i)}] \\
&= \mathbb{E}_{(\mu_{\mathbb{Z}_2}^{\mathbb{Z}_2})^m}[\sup_{h\in\mathcal{H}} \frac{1}{m}\sum_{i=1}^m \sigma_i \frac{1 - y_i h(x_i)}{2}] \\
&\overset{\mathbb{E}_{(\mu_{\mathbb{Z}_2}^{\mathbb{Z}_2})^m}\sigma_i=0}{=} \frac{1}{2}\mathbb{E}_{(\mu_{\mathbb{Z}_2}^{\mathbb{Z}_2})^m}[\sup_{h\in\mathcal{H}} \frac{1}{m}\sum_{i=1}^m -\sigma_i y_i h(x_i)] \\
&= \frac{1}{2}\mathbb{E}_{(\mu_{\mathbb{Z}_2}^{\mathbb{Z}_2})^m}[\sup_{h\in\mathcal{H}} \frac{1}{m}\sum_{i=1}^m \sigma_i h(x_i)] = \frac{1}{2}\hat{R}_{Pr(S)}(\mathcal{H})
\end{aligned}$$

which is required to prove.

We have the following relation between the empirical Rademacher complexity and the Rademacher complexity, using the McDiarmid concentration inequality, see (B.4) and [MRT2012, (3.14), p.36]

$$\mu^n\{S \in \mathcal{Z}^n | \mathcal{R}_{n,\mu}(\mathcal{G}_{\mathcal{H}}^L) \leq \mathcal{R}_S(\mathcal{G}_{\mathcal{H}}^L) + \sqrt{\frac{\ln(2/\delta)}{2m}}\} \geq 1 - \delta/2. \tag{6.6}$$

Theorem 6.9. *(see e.g.* [SSBD2014, Theorems 26.3, 26.5, p. 377- 378]*) Assume that $(\mathcal{Z}, \mathcal{H}, L, \mu)$ is a learning model with $|L(z, h)| < c$ for all $z \in \mathcal{Z}$ and all $h \in \mathcal{H}$. Then for any $\delta > 0$ and any $h \in \mathcal{H}$ we have*

$$\mu^n\{S \in \mathcal{Z}^n |\, R_\mu^L(h) - R_S^L(h) \leq \mathcal{R}_{n,\mu}(\mathcal{G}_{\mathcal{H}}^L) + c\sqrt{\frac{2\ln(2/\delta)}{n}}\} \geq 1 - \delta, \tag{6.7}$$

$$\mu^n\{S \in \mathcal{Z}^n |\, R_\mu^L(h) - R_S^L(h) \leq \mathcal{R}_S(\mathcal{G}_{\mathcal{H}}^L) + 4c\sqrt{\frac{2\ln(4/\delta)}{n}}\} \geq 1 - \delta. \tag{6.8}$$

(6.9)

$$\mu^n\{S \in \mathcal{Z}^n | R_\mu^L(A_{erm}(S)) - R_\mu^L(h) \leq 2Rr_S(\mathcal{G}_{\mathcal{H}}^L) + 5c\sqrt{\frac{2\ln(8/\delta)}{\delta}}\} \geq 1 - \delta.$$

$$\mathbb{E}_{\mu^n}(R_\mu^L(A_{erm}(S)) - R_{\mu,\mathcal{H}}^L) \leq 2Rr_{n,\mu}(\mathcal{G}_{\mathcal{H}}^L). \tag{6.10}$$

It follows from (6.10), using the Markov inequality, the following bound for the sample complexity $m_{A_{erm}}$ in terms of Rademacher complexity

$$\mu^n\{S \in \mathcal{Z}^n |\, R_\mu^L(A_{erm}) - R_{\mu,\mathcal{H}}^L \leq \frac{2\mathcal{R}_{n,\mu}^L(\mathcal{G}_{\mathcal{H}}^L)}{\delta}\} \geq 1 - \delta. \tag{6.11}$$

Remark 6.10. (1) The first two assertions of Theorem 6.9 give an upper bound of a "half" of the sample complexity $m_{\mathcal{H}}$ of a unified learning model $(\mathcal{Z}, \mathcal{H}, L, \mu)$ by the (empirical) Rademacher complexity $\mathcal{R}_n(\mathcal{G})$ of the associated family $\mathcal{G}$. The last assertion of Theorem 6.9 is derived from the second assertion and the Hoeffding inequality.

(2) For the binary classification problem $(\mathcal{X}\times\{0,1\}, \mathcal{H} \subset \{0,1\}^{\mathcal{X}}, L^{(0-1)}, \mathcal{P}(\mathcal{X}\times\{0,1\})$ there exists a close relationship between the Rademacher complexity and the growth function $\Gamma_{\mathcal{H}}(m)$, see [MRT2012, Lemma 3.1, Theorem 3.2, p. 37] for detailed discussion.

6.3. **Model selection.** The choice of a right prior information in machine learning is often interpreted as the choice of a right class $\mathcal{H}$ in a learning model $(\mathcal{Z}, \mathcal{H}, L, P)$ which is also called *a model selection*. A right choice of $\mathcal{H}$ should make balance between the approximation error and the estimation error of $\mathcal{H}$ defined in the error decomposition of $\mathcal{H}$.

6.3.1. *Error decomposition.* We assume that the maximum domain of the expected loss function R^L_μ is a subspace $\mathcal{H}_{L,\mu} \supset \mathcal{H}$, given L and a probability measure $\mu \in P$.

We define *the Bayes risk* of the learning problem R^L_μ on the maximal domain $\mathcal{H}_{L,\mu}$ as follows

$$R^L_{b,\mu} := \inf_{h\in\mathcal{H}_{L,\mu}} R^L_\mu(h)$$

Recall that $R^L_{\mu,\mathcal{H}} := \inf_{h\in\mathcal{H}} R^L_\mu(h)$ quantify the optimal performance of a learner in $\mathcal{H}$. Then we decompose the difference between the expected risk of a predictor $h \in \mathcal{H}$ and the Bayes risk as follows:

$$R^L_\mu(h) - R^L_{b,\mu} = (R^L_\mu(h) - R^L_{\mu,\mathcal{H}}) + (R^L_{\mu,\mathcal{H}} - R^L_{b,\mu}). \tag{6.12}$$

The first term in the RHS of (6.12) is called *the estimation error* of h, cf. (6.1), and the second term is called *the approximation error*. If $h = A_{erm}(S)$ is a minimizer of the empirical risk $\hat{R}^L_S$, then the estimation error of $A_{erm}(S)$ is also called *the sample error* [CS2001, p. 9].

The approximation error quantifies how well the hypothesis class $\mathcal{H}$ is suited for the problem under consideration. The estimation error measures how well the hypothesis h performs relative to best hypotheses in $\mathcal{H}$. Typically, the approximation error will decrease when enlarging $\mathcal{H}$ but the sample error will increase as demonstrated in No-Free-Lunch Theorem 5.9, because P should be enlarged as $\mathcal{H}$ will be enlarged.

Example 6.11. (cf. [Vapnik2000, p. 19], [CS2001, p. 4, 5]) Let us compute the error decomposition of a discriminative model $(\mathcal{X} \times \mathbb{R}, \mathcal{H} \subset \mathbb{R}^{\mathcal{X}}, L_2, \mathcal{P}_B(\mathcal{X} \times \mathbb{R})$ for regression problem. Let $\pi : \mathcal{X} \times \mathbb{R} \to \mathcal{X}$ denote the natural projection. Then the measure ρ, the push-forward measure $\pi_*(\rho) \in \mathcal{P}(\mathcal{X})$ and the conditional probability measure $\rho(y|x)$ on each fiber

$\pi^{-1}(x) = \mathbb{R}$ are related as follows

$$\int_{\mathcal{X}\times\mathbb{R}} \varphi(x,y)d\rho = \int_{\mathcal{X}}(\int_{\mathbb{R}} \varphi(x,y)d\rho(y|x))d\pi_*(\rho)$$

for $\varphi \in L^1(\rho)$, similar to the Fubini theorem, see Subsection A.4.

Let us compute the Bayes risk $R^L_{b,\mu}$ for $L = L_2$. The maximal subspace $\mathcal{H}_{L,\mu}$ where the expected loss R^L_μ is well defined is the space $L^2(\mathcal{X}, \pi_*(\rho))$. We claim that the regression function of ρ, see Exercise 2.7,

$$r_\rho(x) := E_\rho(i_2(Y)|X = x) = \int_{\mathbb{R}} y d\rho(y|x)$$

minimizes the $MSE_{\pi_*(\rho)}$ defined on the space $L^2(\mathcal{X}, \pi_*(\rho))$. Indeed, for any $f \in L^2(\mathcal{X}, \pi_*(\rho))$ we have

$$MSE_{\pi_*(\rho)}(f) = \int_{\mathcal{X}} (f(x) - r_\rho(x))^2 d\pi_*(\rho) + MSE_{\pi_*(\rho)}(r_\rho). \tag{6.13}$$

The equation (6.13) implies that the Bayes risk $R^L_{b,\pi_*(\rho)}$ is $MSE_{\pi_*(\rho)}(r_\rho)$. It follows that the approximation error of a hypothesis class $\mathcal{H}$ is equal

$$MSE_{\pi_*(\rho)}(f_{min}) = \int_{\mathcal{X}} (f_{min} - r_\rho(x))^2 d\pi_*(\rho) + MSE_{\pi_*(\rho)}(r_\rho),$$

where f_{min} is a minimizer of MSE in $\mathcal{H}$. Since $MSE_{\pi_*(\rho)}(r_\rho)$ is a constant, if $\mathcal{H}$ is compact, f_{min} exists and satisfies the condition

$$f_{min} = arg\min_{g\in\mathcal{H}} d(g, r_\rho), \tag{6.14}$$

where d is the L^2-distance on $L^2(\mathcal{X}, \pi_*(\rho))$.

6.3.2. *Validation and cross-validation.* An important empirical approach in model selection is validation and its refinement - (k-fold) cross-validation. 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 S. 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 $S' \in \mathcal{Z}^m$ and choose the predictor h_i that minimizes the error over the validation set.

The basic idea of *cross-validation* is to partition the training set $S = (z_1, \cdots, z_n) \in \mathcal{Z}^n$ into two sets $S = S_1 \cup S_2$ where $S_1 \in \mathcal{Z}^k$ and $S_2 \in \mathcal{Z}^{n-k}$. The set S_1 is used for training each of the candidate models, and the second set S_2 is used for deciding which of them yields the best results.

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.

6.4. **Conclusion.** In this lecture we use several complexities of a learning model $(\mathcal{Z}, \mathcal{H}, L, P)$ for obtaining upper bounds of the sample complexity of the ERM algorithm A_{erm}. Among them Rademacher complexities are the most sophisticated ones that measure the capacity of a hypothesis class on a specific sample, which can be used to bound the difference between empirical and expected error, and thus the excess generalization error of empirical risk minimization. To find an ideal hypothesis class $\mathcal{H}$ for a learning problem we have to take into account the error decomposition of a learning model and the resulting bias-variance trade-off and use empirical cross validation methods.

7. Support vector machines

In this lecture we shall consider a class of simple supervised learning machines for binary classification problems and apply results in the previous lectures on consistent learning algorithms. Our learning machines are $(V \times \mathbb{Z}_2, \mathcal{H}_{lin}, L, \mathcal{P}(V \times \mathbb{Z}_2), A)$ where V is a real Hilbert space, $\mathcal{H}_{lin}$ consists of linear classifiers, defined below, L is a $(0-1)$ loss function (resp. regularized $(0-1)$ loss function) and A is a hard SVM algorithm (resp. a soft SVM algorithm), which we shall learn in today lecture. The original SVM algorithm is the hard SVM algorithm, which 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.

7.1. **Linear classifier and hard SVM.** For $(w, b) \in V \times \mathbb{R}$ we set

$$f_{(w,b)}(x) := \langle w, x \rangle + b. \tag{7.1}$$

Definition 7.1. *A linear classifier* is a function $sign\, f_{(w,b)} : V \to \mathbb{Z}_2$, $x \mapsto sign f_{(w,b)}(x) \in \{-1, 1\} = \mathbb{Z}_2$.

We identify each linear classifier with the half space $H^+_{(w,b)} := sign f^{-1}_{(w,b)}(1) = f^{-1}_{(w,b)}(\mathbb{R}_{\geq 0}) \subset V$ and set $H_{(w,b)} := f^{-1}_{(w,b)}(0) \subset V$. Note that each hyperplane $H_{(w,b)} \subset V$ defines $H^+_{(w,b)}$ up to a reflection of V around $H_{(w,b)}$ and therefore defines the affine function $f_{(w,b)}$ up to a multiplicative factor $\lambda \in \mathbb{R}^*$. Denote by $\mathcal{H}_A(V)$ the set of of all hyperplanes in the affine space V. Then $\mathcal{H}_{lin}$ is a double cover of $\mathcal{H}_A(V)$ with the natural projection $\pi : \mathcal{H}_{lin} \to \mathcal{H}_A(V)$ defined above.

Definition 7.2. A training sample $S = (x_1, y_1), \cdots, (x_m, y_m) \in (V \times \{\pm 1\})^m$ is called *separable*, if there is a half space $H^+_{(w,b)} \subset V$ that correctly classifies S, i.e. for all $i \in [1, m]$ we have $x_i \in H^+_{(w,b)}$ iff $y_i = 1$. In other words, the linear classifier $sign f_{(w,b)}$ is a minimizer of the empirical risk function $R^{0-1}_S : \mathcal{H}_{lin} \to \mathbb{R}$ associated to the zero one loss function $L^{(0-1)}$.

Remark 7.3. (1) A half space $H^+_{(w,b)}$ correctly classifies S if and only if the empirical risk function $\hat{R}^{(0-1)}_S(f_{(w,b)}) = 0$ and $sign\, f_{(w,b)}$ is a linear classifier associated to $H^+_{(w,b)}$.

(2) Write $S = S_+ \cup S_-$ where

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

Let $Pr : (V \times \{\pm 1\})^m \to V^m$ denote the canonical projection. Then S is separable if and only if there exists a hyper-plane $H_{(w,b)}$ that separates $[\mathrm{Pr}(S_+)]$ and $[Pr(S_-)]$, where recall that $[(x_1, \cdots, x_m)] = \cup_{i=1}^m \{x_i\} \subset V$. In this case we say that $H_{(w,b)}$ *correctly separates* S.

(3) If a training sample S is separable then the separating hyperplane is not unique, and hence there are many minimizers of the empirical risk function $\hat{R}^{(0-1)}_S$. Thus, given S, we need to find a strategy for selecting one of these ERM's, or equivalently for selecting a separating hyperplane $H_{(w,b)}$, since the associated half-space $H^+_{(w,b)}$ is defined by $H_{(w,b)}$ and any training value (x_i, y_i). The standard approach in the SVM framework is to choose $H_{(w,b)}$ that maximizes the distance to the closest points $x_i \in [Pr(S)]$. 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 $H_{(w,b)}$.

Lemma 7.4 (Distance to a hyperplane)**.** *Let V be a real Hilbert space and $H_{(w,b)} := \{z \in V |\, \langle z, w\rangle + b = 0\}$. The distance of a point $x \in V$ to $H_{(w,b)}$ is given by*

$$\rho(x, H_{(w,b)}) := \inf_{z \in H_{(w,b)}} ||x - z|| = \frac{|\langle x, w\rangle + b|}{||w||}. \tag{7.2}$$

Proof. Since $H_{(w,b)} = H_{(w,b)/\lambda}$ for all $\lambda > 0$, it suffices to prove (7.2) for the case $||w|| = 1$ and hence we can assume that $w = e_1$. Now formula (7.2) follows immediately, noting that $H_{(e_1,b)} = H_{(e_1,0)} - be_1$. □

Let $H_{(w,b)}$ separate $S = \{(x_1, y_1), \cdots, (x_m, y_m)\}$ correctly. Then we have

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

$$\Longrightarrow |\langle x_i, w\rangle + b| = y_i(\langle x, w\rangle + b).$$

Hence, by Lemma 7.4, the distance between $H_{(w,b)}$ and S is

$$\rho(S, H_{(w,b)}) := \min_i \rho(x_i, H_{(w,b)}) = \frac{\min_i y_i(\langle x_i, w\rangle + b)}{||w||}. \tag{7.3}$$

The distance $\rho(S, H_{(w,b)})$ is also called *the margin of a hyperplane $H_{(w,b)}$ w.r.t. S*. The hyperplanes, that are parallel to the separating hyperplane and passing through the closest points on the negative or positive sides are called *marginal*.

Denote by $\mathcal{H}_S$ the subset of $\pi(\mathcal{H}_{lin}) = \mathcal{H}_A(V)$ that consists of hyperplanes separating S. Set

$$A^*_{hs}(S) := arg \max_{H_{(w,b)'} \in \mathcal{H}_S} \rho(S, H_{(w,b)'}). \tag{7.4}$$

Now we define a map $A_{hs} : \cup_m (V \times \mathbb{Z}_2)^m \to \mathcal{H}_{lin}$ by letting $A_{hs}(S) \in \pi^{-1} A^*_{hs}(S)$ to be the linear classifier that correctly classifies S.

Definition 7.5. *A hard SVM* is a learning machine $(V \times \mathbb{Z}_2, \mathcal{H}_{lin}, L^{(0-1)}, \mathcal{P}(V \times \mathbb{Z}_2), A_{hs})$.

The domain of the optimization problem in (7.4) is $\mathcal{H}_S$, which is not easy to determine. So we replace this problem by another optimization problem over a larger convex domain as follows.

Lemma 7.6. *For $S = \{(x_1, y_1), \cdots, (x_m, y_m)\}$ we have*

$$A_{hs}(S) = H^+_{(w,b)} \text{ where } (w,b) = arg \max_{(w,b):||w|| \leq 1} \min_i y_i(\langle w, x_i \rangle + b). \tag{7.5}$$

Proof. If $H_{(w,b)}$ separates S then $\rho(S, H_{(w,b)}) = \min_i y_i(\langle w, x_i \rangle + b)$. Since the constraint $||w|| \leq 1$ does not effect on $H_{(w,b)}$, which is invariant under a positive rescaling, (7.3) implies that

$$\max_{(w,b):||w|| \leq 1} \min_i y_i(\langle w, x_i \rangle + b) \geq \max_{H_{(w,b)'} \in \mathcal{H}_S} \rho(S, H_{(w,b)'}). \tag{7.6}$$

Next we observe that if $H_{(w,b)} \notin \mathcal{H}_S$ then

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

Combining this with (7.6) we obtain

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

This completes the proof of Lemma 7.6. □

A solution $A_{hs}(S)$ of the equation (7.5) maximizes the enumerator of the far RHS of (7.3) under the constraint $||w|| \leq 1$. In the Proposition below we shall show that $A_{hs}(S)$ can be found as a solution to the dual optimization problem of minimizing the dominator of the RHS (7.3) under the constraint that the enumerator of the far RHS of (7.3) has to be fixed.

Proposition 7.7. *A solution to the following optimization problem, which is called Hard-SVM,*

$$(w_0, b_0) = arg \min_{w,b} \{||w||^2 : \ y_i(\langle w, x_i \rangle + b) \geq 1 \text{ for all } i\} \tag{7.7}$$

produces a solution $(w, b) := (w_0/||w_0||, b_0/||w_0||)$ of the optimization problem (7.5).

Proof. Let (w_0, b_0) be a solution of (7.7). We shall show that $(w_0/||w_0||, b_0/||w_0||)$ is a solution of (7.5). It suffices to show that the margin of the hyperplane $H_{(w_0,b_0)}$ is greater than or equal to the margin of the hyperplane associated to a (and hence any) solution of (7.5).

Let (w^*, b^*) be a solution of Equation (7.5). Set

$$\gamma^* := \min_i y_i(\langle w^*, x_i\rangle + b^*)$$

which is the margin of the hyperplane $H_{(w^*,b^*)}$ by (7.3). Therefore for all i we have

$$y_i(\langle w^*, x_i\rangle + b^*) \geq \gamma^*$$

or equivalently

$$y_i^*(\langle \frac{w^*}{\gamma^*}, x_i\rangle + \frac{b^*}{\gamma^*}) \geq 1.$$

Hence the pair $(\frac{w^*}{\gamma^*}, \frac{b^*}{\gamma^*})$ satisfies the condition of the quadratic optimization problem in (7.7). It follows that

$$||w_0|| \leq ||\frac{w^*}{\gamma^*}|| = \frac{1}{\gamma^*}.$$

Hence for all i we have

$$y_i(\frac{w_0}{||w_0||} + \frac{b_0}{||w_0||}) = \frac{y_i(\langle w_0, x_i\rangle + b_0)}{||w_0||} \geq \frac{1}{||w_0||} \geq \gamma^*.$$

This implies that the margin of $H_{(w_0,b_0)}$ satisfies the required condition. This completes the proof of Proposition 7.7. □

Remark 7.8. (1) The optimization problem of (7.4) 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 (7.4).

(2) In practice, when we have a sample set S of large size, then S is not separable, thus the application of hard SVM is limited.

Exercise 7.9. (1) Show that the vector w_0 of the solution (w_0, b_0) in (7.7) of the SVM problem is a linear combination of the training set vectors $x_1, \cdots, x_m$.

(2) Show that x_i lies on the marginal hyperplanes $\langle w_0, x\rangle + b_0 = \pm 1$.

A vector x_i appears in the linear expansion of the weight vector w_0 in Exercise 7.9 is called *a support vector*.

7.2. **Soft SVM.** Now we consider the case when the sample set S is not 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 $sign\, f_{(w,b)}$ such that

$H_{(w,b)} = f^{-1}_{(w,b)}(0)$ still has maximal margin in some sense. More precisely, we shall relax the hard SVM rule (7.7) by replacing the constraint

$$y_i(\langle w, x_i\rangle + b) \geq 1 \tag{7.8}$$

by the relaxed constraint

$$y_i(\langle w, x_i\rangle) + b \geq 1 - \xi_i \tag{7.9}$$

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 (7.8).

The relaxed hard SVM rule is called *the soft SVM rule*.

Definition 7.10. *The soft SVM algorithm* $A_{ss} : (\mathbb{R}^d \times \mathbb{Z}_2)^m \to H_{lin}$ *with slack variables* $\{\xi \in \mathbb{R}^m_{\geq 0}\}$ is defined as follows

$$A_{ss}(S) = sign\, f_{(w_0,b_0)}(S)$$

where (w_0, b_0) satisfies the following equation with $\xi = (\xi_1, \cdots, \xi_m)$

$$(w_0, b_0, \xi) = arg\min_{w,b,\xi}(\lambda||w||^2 + \frac{1}{m}||\xi||_{l_1}) \tag{7.10}$$

$$\text{s. t. } \forall i,\ y_i(\langle w, x_i\rangle + b) \geq 1 - \xi_i \text{ and } \xi_i \geq 0. \tag{7.11}$$

In what follows we shall show that the soft SVM algorithm A_{ss} is a solution of a regularized loss minimization rule, which is a refinement of the ERM rule.

Digression *Regularized Loss Minimization* (RLM) is a learning algorithm on a learning model $(\mathcal{Z}, \mathcal{H}, L, P)$ in which we jointly minimize the empirical risk $\hat{R}^L_S$ and a regularization function. Formally, a regularization function is a mapping $R : \mathcal{H} \to \mathbb{R}$ and the regularized loss minimization rule is a map $A_{rlm} : \mathcal{Z}^n \to \mathcal{H}$ such that $A_{rlm}(S)$ is a minimizer of the empirical regularized loss function $\tilde{R}^L_S := \hat{R}^L_S + R : \mathcal{H} \to \mathbb{R}$.

As the ERM algorithm works under certain condition, the RLM algorithm also works under certain conditions, see e.g. [SSBD2014, Chapter 13] for detailed discussion.

The loss function for the soft SVM learning machine is the hinge loss function $L^{hinge} : \mathcal{H}_{lin} \times (V \times \{\pm 1\}) \to \mathbb{R}$ defined as follows

$$L^{hinge}(h_{(w,b)}, (x, y)) := \max\{0, 1 - y(\langle w, x\rangle + b)\}. \tag{7.12}$$

Hence the empirical hinge risk function is defined as follows for $S = \{(x_1, y_1) \cdots, (x_m, y_m)\}$

$$R^{hinge}_S(h_{(w,b)}) = \frac{1}{m}\sum_{i=1}^{m} \max\{0, 1 - y_i(\langle w, x_i\rangle + b\}.$$

Lemma 7.11. *The Equation (7.10) with constraint (7.11) for A_{ss} is equivalent to the following regularized risk minimization problem, which does not depend on the slack variables ξ:*

$$A_{ss}(S) = arg \min_{f_{(w,b)}} \big(\lambda||w||^2 + R_S^{hinge}(f_{(w,b)})\big) \in \mathcal{H}_{lin}. \tag{7.13}$$

Proof. Let us fix (w_0, b_0) and minimize the RHS of (7.10) under the constraint (7.11). It is straightforward to see that $\xi_i = L^{hinge}\big((w,b),(x_i,y_i)\big)$. Using this and comparing (7.10) with (7.13), we complete the proof of Lemma 7.11. □

From Lemma 7.11 we obtain immediately the following

Corollary 7.12 (Definition)**.** *A soft SVM is a learning machine* $(V \times \mathbb{Z}_2, \mathcal{H}_{lin},$
$L^{hinge}, \mathcal{P}(V \times \mathbb{Z}_2), A_{rlm})$.

Remark 7.13. The hinge loss function L^{hinge} enjoys several good properties that justify the preference of L^{hinger} as a loss function over the zero-one loss function $L^{(0-1)}$, see [SSBD2014, Subsection 12.3, p. 167] for discussion.

7.3. **Sample complexities of SVM.**

Exercise 7.14. Prove that $VC\dim H_{lin} = \dim V + 1$.

From the Fundamental Theorem of binary classification 5.15 and Exercise (7.14) we obtain immediately the following

Proposition 7.15. *The binary classification problem* $(V \times \mathbb{Z}_2, \mathcal{H}_{lin}, L^{(0-1)}, \mathcal{P}(V \times \mathbb{Z}_2))$ *ha a uniformly consistent learning algorithm if and only if V is finite dimensional.*

In what follow we shall show the uniform consistent of hard SVM and soft SVM replacing the statistical model $\mathcal{P}(\mathbb{R}^d \times \mathbb{Z}_2)$ by a smaller class.

Definition 7.16. ([SSBD2014, Definition 15.3]) Let μ be a distribution on $V \times \mathbb{Z}_2$. We say that μ *is separable with a (γ, ρ)-margin* if there exists $(w^*, b^*) \in V \times \mathbb{R}$ such that $\|w\| = 1$ and such that

$$\mu\{(x,y) \in V \times \mathbb{Z}_2 |\, y(\langle w^*, x\rangle + b^*) \geq \gamma \text{ and } ||x|| \leq \rho\} = 1.$$

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

Definition 7.16 means that the set of labeled pairs $(x, y) \in V \times \mathbb{Z}_2$ that satisfy the condition

$$y(\langle w^*, x\rangle + b^*) \geq \gamma \text{ and } ||x|| \leq \rho$$

has a full μ-measure, where μ is a separable measure on with (γ, ρ)-margin.

Remark 7.17. (1) We regard an affine function $f_{(w,b)} : V \to \mathbb{R}$ as a linear function $f_{w'} : V' \to \mathbb{R}$ where $V' = \langle e_1 \rangle_{\otimes \mathbb{R}} \oplus V$, i.e., we incorporate the bias term b of $f_{(w,b)}$ in (7.1) into the term w as an extra coordinate. More precisely we set

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

Then

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

Note that the natural projection of the zero set $f_{w'}^{-1}(0) \subset V'$ to V is the zero set $H_{(w,b)}$ of $f_{(w,b)}$.

(2) By Remark 7.17 (1), we can always assume that a separable measure with (γ, ρ)-margin is a one that uses a homogeneous half-space by enlarging the instance space V.

Denote by $\mathcal{P}_{(\gamma,\rho)}(V \times \mathbb{Z}_2)$ the subset of $\mathcal{P}(V \times \mathbb{Z}_2)$ that consists of separable measures with a (γ, ρ)-margin using a homogeneous half-space. Using the Rademacher complexity, see [SSBD2014, Theorem 26.13, p. 384], we have the following estimate of the sample complexity of the learning machine $(V \times \mathbb{Z}_2, \mathcal{H}_{lin}, L^{(0-1)}, P_{(\gamma,\rho)}(V \times \mathbb{Z}_2), A_{hs})$.

Theorem 7.18. *(*[SSBD2014, Theorem 15.4, p. 206]*) Let $\mu \in \mathcal{P}_{(\gamma,\rho)}(V \times \mathbb{Z}_2)$. Then we have*

$$\mu^m\{S \in (V \times \mathbb{Z}_2)^m |\, R_\mu^{(0-1)}(A_{hs}(S)) \leq \sqrt{\frac{4(\rho/\gamma)^2}{m}} + \sqrt{\frac{2\log(2/\delta)}{m}}\} \geq 1 - \delta.$$

Denote by $\mathcal{P}_\rho(V \times \mathbb{Z}_2)$ the set of of probability measures on $V \times \mathbb{Z}_2$ whose support lies in $B(0, \rho) \times \mathbb{Z}_2$ where $B(0, \rho)$ is the ball of radius ρ centered at the origin of V. Now we shall examine the sample complexity of the soft SVM learning machine $(V \times \mathbb{Z}_2, \mathcal{H}_{lin}, L^{hinge}, P_\rho(V \times \mathbb{Z}_2), A_{ss})$. The following theorem is a a consequence of Lemma 7.11 and a general result on the sample complexity of RLM under certain conditions, see [SSBD2014, Corollary 13.8, p. 179].

Theorem 7.19. *(*[SSBD2014, Corollary 15.7, p. 208]*) Let $\mu \in \mathcal{P}_\rho(V \times \mathbb{Z}_2)$. Then for every $r > 0$ we have*

$$\begin{aligned} \mathbb{E}_{\mu^m}\Big(R_\mu^{(0-1)}\big(A_{ss}(S)\big)\Big) &\leq \mathbb{E}_{\mu^m}\Big(R_\mu^{hinge}\big(A_{ss}(S)\big)\Big) \\ &\leq \min_{w \in B(0,r)} R_\mu^{hinge}(h_w) + \sqrt{\frac{8\rho^2 r^2}{m}}. \end{aligned} \tag{7.14}$$

Exercise 7.20. Using the Markov inequality, derive from Theorem 7.19 an upper bound for the sample complexity of the soft SVM.

Theorem 7.19 and Exercise 7.20 imply that we can control the sample complexity of a soft SVM algorithm as a function of the norm of the underlying Hilbert space V, independently of the dimension of V. This becomes highly significant when we learn classifiers $h : V \to \mathbb{Z}_2$ via embeddings into high dimensional feature spaces.

7.4. **Conclusion.** In this section we consider two classes of learning machines for binary classification. The first class consists of learning models $(V\times\mathbb{Z}_2, \mathcal{H}_{lin}, L^{(0-1)}, \mathcal{P}(V\times\mathbb{Z}_2))$ with finite VC-dimension iff and only if V is finite dimensional. If $\mu\in\mathcal{P}(V\times\mathbb{Z}_2)$ is separable with (γ,ρ)-margin then we can upper bound the sample complexity of the hard SVM algorithm A_{hs} for $(V\times\mathbb{Z}_2, \mathcal{H}_{lin}, L^{(0-1)}, \mu)$ using the ration ρ/γ and Rademacher's complexity. The second class consists of learning machines $(V\times\mathbb{Z}_2, \mathcal{H}_{lin}, L^{hinge}, \mathcal{P}_\rho(V\times\mathbb{Z}_2), A_{ss})$. The soft SVM algorithm A_{ss} is a solution of a regularized ERM and therefore we can apply here general results on sample complexity of regularized ERM algorithms.

8. Kernel based SVMs

In the previous lecture we considered the hypothesis class $\mathcal{H}_{lin}$ of linear classifiers. A linear classifier $sign\, f_{(w,b)}$ correctly classifies a training sample $S\subset(V\times\{\pm1\})^m$ iff the zero set $H_{(w,b)}$ of $f_{(w,b)}$ separates the subsets $[Pr(S_-)]$ and $[Pr(S_+)]$. 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. Now we observe that any (polynomial) function f on $\mathbb{R}^d$ can be regarded as the restriction of a new coordinate function y on $\mathbb{R}^d\times\mathbb{R}(y)$ to the image of the graph $\Gamma_f(\mathbb{R}^d)\subset\mathbb{R}^d\times\mathbb{R}$ of f, i.e., $f(x)=y(\Gamma_f(x))$. However, the computational complexity of SVM with learning by polynomial embedding $\{(x,f(x))|\, x\in\mathbb{R}^d\}$ 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. Namely we are interested in classifiers of the form

$$sign\,\tilde{h}:\mathcal{X}\to\mathbb{Z}_2,\ \tilde{h}(x):=\langle h,\psi(x)\rangle,$$

where h is an element in a Hilbert space W, $\psi:\mathcal{X}\to W$ is a "feature map" and the kernel function $K_\varphi:\mathcal{X}\times\mathcal{X}\to\mathbb{R}$ is defined

$$K_\psi(x,y):=\langle\psi(x),\psi(y)\rangle. \tag{8.1}$$

We shall see that to solve the hard SVM optimization problem (7.7) for $h\in W$ it suffices to learn K. This kernel trick requires less computational complexity than the one for learning $\psi:\mathcal{X}\to W$.

8.1. **Kernel trick.** It is known that a solution of a hard SVM can be expressed as a linear combination of support vectors (Exercise 7.9). 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:

$$w_0 = arg\min_{w\in W}\Big(f\big(\langle w,\psi(x_1)\rangle,\cdots,\langle w,\psi(x_m)\rangle\big) + R(\|w\|)\Big) \tag{8.2}$$

where $x_i \in \mathcal{X}$, 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 $\psi : \mathcal{X} \to W$ is often called *the feature map*, and W is called *the feature space*.

The following examples show that the optimization problem for the hard (resp. soft) SVM algorithm is an instance of the optimization problem (8.2).

Example 8.1. Let $S = \{(x_1, y_1), \cdots, (x_m, y_m) \in (V \times \mathbb{Z}_2)^m$.

(1) Plugging in Equation (8.2)

$$R(a) := a^2,$$

$$f(a_1,\cdots,a_m) := \begin{cases} 0 & \text{if } y_i(a_i) \geq 1 \text{ for all } i \\ \infty & \text{otherwise} \end{cases}$$

we obtain Equation (7.7) of hard SVM for homogeneous vectors $(w,0)$, replacing a_i by $\langle w, x_i\rangle$. The general case of non-homogeneous solutions (w, b) is reduced to the homogeneous case by Remark 7.17.

(2) Plugging in Equation (8.2)

$$R(a) := \lambda a^2,$$

$$f(a_1,\cdots,a_m) := \frac{1}{m}\sum_{i=1}^{m}\max\{0, 1 - y_i a_i\}$$

we obtain Equation (7.13) of soft SVM for a homogeneous solution $A_{ss}(S)$, identifying $A_{ss}(S)$ with its parameter $(w, 0)$, S with $\{(x_1, y_1)\cdots(x_m, y_m)\}$ and replacing a_i with $\langle w, x_i\rangle$.

Theorem 8.2 (Representer Theorem)**.** *Let $\psi : \mathcal{X} \to W$ be a feature mapping from an instance space $\mathcal{X}$ to a Hilbert space W and w_0 a solution of (8.2). 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 (8.2).*

Proof. Assume that w_0 is a solution of (8.2). 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

$$\|\bar{w}_0\| \leq \|w_0\| \tag{8.3}$$

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

$$f\big(\langle\bar{w}_0,\psi(x_1)\rangle,\cdots,\langle\bar{w}_0,\psi(x_m)\rangle\big) = f\big(\langle w_0,\psi(x_1)\rangle,\cdots,\langle w_0,\psi(x_m)\rangle\big). \tag{8.4}$$

From (8.3), (8.4) and taking into account the monotonicity of R, we conclude that $\bar{w}_0$ is also a solution of (8.2). This completes the proof of Theorem 8.2. □

The Representer Theorem implies that it suffices to find a solution of Equation (8.2) in a finite dimensional subspace $W_1 \subset W$. In what follows we shall describe a method to solve the minimization problem of (8.2) on W_1, 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*,
- $G = (G_{ij}) := K(x_i, x_j)$ - *a Gram matrix*,
- $w_0 = \sum_{i=1}^m \alpha_j \psi(x_j) \in W_1$ - a solution of Equation (8.2).

Then $\alpha = (\alpha_1, \cdots, \alpha_m)$ is a solution of the following minimization problem

$$arg \min_{\alpha \in \mathbb{R}^m} f\big(\sum_{j=1}^m \alpha_j G_{j1}, \cdots, \sum_{j=1}^m \alpha_j G_{jm}\big) + R\Big(\sqrt{\sum_{i,j=1}^m \alpha_i \alpha_j G_{ji}}\Big). \tag{8.5}$$

Recall that the solution $w_0 = \sum_{i=1}^m \alpha_j \psi(x_j)$ of the hard (resp. soft) SVM optimization problem, where $(\alpha_1, \cdots, \alpha_m)$ is a solution of (8.5), produces a "nonlinear" classifier $\hat{w}_0 : \mathcal{X} \to \mathbb{Z}_2$ associated to as follows

$$\hat{w}_0(x) := sign\, w_0(x)$$

where

$$w_0(x) := \langle w_0, \psi(x) \rangle = \sum_{i=1}^m \alpha_j \langle \psi(x_j), \psi(x) \rangle = \sum_{j=1}^m \alpha_j K(x_j, x). \tag{8.6}$$

To compute (8.6) 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 8.3. *Find a sufficient and necessary condition for a kernel 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 feature mapping $\psi : \mathcal{X} \to W$, where W is a real Hilbert space.*

Definition 8.4. If K satisfies the condition in Problem 8.3 we shall say that K *is generated by a (feature) mapping* ψ. The target Hilbert space is also called *a feature space*.

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

8.2.1. *Positive semi-definite kernel.*

Definition 8.5. 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.

Theorem 8.6. *A kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is induced by a feature map to a Hilbert space if and only if it is positive semi-definite.*

Proof. 1) First let us prove the "only if" assertion of Theorem 8.6. 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 8.6

2) Now let us prove the "if" part. Assume that $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is positive semi-definite. 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

$$W := \{ f \in \mathbb{R}^{\mathcal{X}} \mid f = \sum_{i=1}^{N(f)} a_i K_{x_i},\ a_i \in \mathbb{R} \text{ and } N(f) < \infty \}.$$

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 $x \in \mathcal{X}$

$$\langle f, K_x \rangle^2 \leq \langle f, f \rangle \langle K_x, K_x \rangle. \tag{8.7}$$

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

$$f(x) = \langle f, K_x \rangle. \tag{8.8}$$

Using (8.8), we obtain from (8.7) 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 8.6. □

Example 8.7. (1) (Polynomial kernels). Assume that P is a polynomial in one variable with non-negative coefficients. Then the polynomial kernel of the form $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, $(x,y) \mapsto P(\langle x, y \rangle)$ is a PSD kernel. This follows from the observations that if $K_i : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, $i = 1, 2$, are PDS kernel then $(K_1 + K_2)(x,y) := K_1(x,y) + K_2(x,y)$ is a PSD kernel, and $(K_1 \cdot K_2)(x,y) := K_1(x,y) \cdot K_2(x,y)$ is a PSD kernel. In particular, $K(x,y) := (1 + \langle x, y \rangle)^2$ is a PSD kernel.

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

Exercise 8.8. (1) Show that the Gaussian kernel $K(x,y) := \exp\big(-\frac{\gamma}{2}||x-y||^2\big)$ is a PSD 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 PSD kernel for any $p \in \mathbb{N}^+$.

8.2.2. *Reproducing kernel Hilbert space.* Given a PSD kernel $K : \mathcal{X}\times\mathcal{X} \to \mathbb{R}$ there exist many feature maps $\varphi : \mathcal{X} \to W$ such that K is generated by a feature map $\varphi : \mathcal{X} \to W$. Indeed, if $K(x,x) = \langle \varphi(x), \varphi(x)\rangle$ then $K(x,x) = \langle e\circ\varphi(x), e\circ\varphi(x)\rangle$ for any isometric embedding $e : W \to W'$. However, there is a canonical choice for the feature space, a so-called reproducing kernel Hilbert space.

Definition 8.9 (Reproducing kernel Hilbert space)**.** Let $\mathcal{X}$ be an instance 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. [17] 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 8.10. 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* $\mathcal{H}$.

Thus every RKHS $\mathcal{H}$ on $\mathcal{X}$ produces a PSD kernel $K : \mathcal{X}\times\mathcal{X} \to \mathbb{R}$. Conversely, Theorem 8.11 below asserts that every PSD kernel reproduces a RKHS $\mathcal{H}$.

Theorem 8.11. *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 8.6, given a PSD kernel $K : \mathcal{X}\times\mathcal{X} \to \mathbb{R}$, there exists a RKHS

$$\mathcal{H} := \{K_x |\, K_x(y) = K(x,y) \text{ for all } x,y \in \mathcal{X}\}$$

such that

$$\forall x,y \in \mathcal{X} \text{ we have } K(x,y) = \langle K_x, K_y\rangle. \tag{8.9}$$

[17] In other words, the vector structure on $\mathcal{H}$ is induced from the vector structure on $\mathbb{R}$ via the evaluation map.

From (8.9) we conclude that the evaluation map $ev_x : \mathcal{H} \to \mathbb{R}$, $ev_x(K_y) = K(y,x)$ is a linear bounded map for all x, y, since

$$||ev_x|| = \max_{||K_y||=1} ev_x(K_y) = \max_{||K_y||=1} K(y,x) \leq \sqrt{K(x,x)}.$$

Hence $\mathcal{H}$ is a RKHS.

To show the uniqueness of a RKHS $\mathcal{H}$ such that K is the reproducing kernel of $\mathcal{H}$ we assume that there exists another RKHS $\mathcal{H}'$ such that for all $x, y \in \mathcal{X}$ there exist $k_x, k_y \in \mathcal{H}'$ with the following properties

$$K(x,y) = \langle k_x, k_y \rangle \text{ and } f(x) = \langle f, k_x \rangle \text{ for all } f \in \mathcal{H}.$$

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 8.11. □

8.3. **Kernel based SVMs and their generalization ability.**

8.3.1. *Kernel based SVMs.* Let $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a PSD kernel. Denote by $\mathcal{H}(K)$ the RHKS of functions on $\mathcal{X}$ that produces K. Each function $h \in \mathcal{H}(K)$ defines a binary classifier

$$sign\, h : \mathcal{X} \to \mathbb{Z}_2.$$

Denote by K_{lin} the set of all binary classifiers $sign\, h$ where $h \in \mathcal{H}(K)$. Using the Representer Theorem 8.2 and Example 8.1 (1), we replace the algorithm A_{hs} of a hard SVM by a kernel based algorithm.

Definition 8.12. *A kernel based hard SVM* is a learning machine $(\mathcal{X} \times \mathbb{Z}_2, K_{lin}, L^{(0-1)}, \mathcal{P}(\mathcal{X} \times \mathbb{Z}_2), A_{hk})$, where for $S = \{(x_1, y_1), \cdots, (x_m, y_m)\} \in (\mathcal{X} \times \mathbb{Z}_2)^m$ and any $x \in \mathcal{X}$ we have

$$A_{hk}(S)(x) = sign \sum_{i=1}^{m} \alpha_j K(x_j, x) \in \mathbb{Z}_2,$$

and $\alpha := (\alpha_1, \cdots, \alpha_m)$ is the solution of the following optimization problem (8.10)

$$\alpha = arg\min\Big(f\big(\sum_{j=1}^{m} \alpha_j K(x_j, x_1), \cdots, \sum_{j=1}^{m} \alpha_j K(x_j, x_m)\big) + R\big(\sqrt{\sum_{i,j=1}^{m} \alpha_i \alpha_j K(x_i, x_j)}\big)\Big),$$

where R and f are defined in Example 8.1(1).

Using the Representer Theorem 8.2 and Example 8.1 (2) , we replace the algorithm A_{ss} of a soft SVM by a kernel based algorithm.

Definition 8.13. *A kernel based soft SVM* is a learning machine $(\mathcal{X}\times\mathbb{Z}_2, K_{lin}, L^{hinge}, \mathcal{P}(\mathcal{X}\times\mathbb{Z}_2), A_{sk})$, where for $S = \{(x_1, y_1), \cdots, (x_m, y_m)\} \in (\mathcal{X} \times \mathbb{Z}_2)^m$ and any $x \in \mathcal{X}$ we have

$$A_{sk}(S)(x) = sign \sum_{i=1}^{m} \alpha_j K(x_j, x) \in \mathbb{Z}_2,$$

and $\alpha := (\alpha_1, \cdots, \alpha_m)$ is the solution of the following optimization problem (8.11)

$$\alpha = arg\min\Big(f(\sum_{j=1}^{m} \alpha_j K(x_j, x_1), \cdots, \sum_{j=1}^{m} \alpha_j K(x_j, x_m)) + R(\sqrt{\sum_{i,j=1}^{m} \alpha_i \alpha_j K(x_i, x_j)})\Big),$$

where R and f are defined in Example 8.1(2).

8.3.2. *Generalization ability of kernel based SVMs.*

• The advantage of working with kernels rather than directly optimizing in the feature space is that in some situations the dimension of the feature space is extremely large while implementing the kernel function is very simple and in many case the computational time complexity of solving (8.10) is a polynomial on the variable of the size of x_i, $i \in [1, m]$, see [SSBD2014, p. 221-223].

• The upper bound for the sample complexity of hard SVM in Theorem 7.18 is also valid for the sample complexity of the kernel based hard SVM [SSBD2014, Theorem 26.3, p. 384] after adapting the condition of separability with (γ, ρ)-margin of a measure $\mu \in \mathcal{P}(\mathcal{X} \times \mathbb{Z}_2)$ in terms of the kernel function.

• The upper bound for the sample complexity of soft SVM in Theorem 7.19 is also valid for the sample complexity of the kernel based soft SVM, after adapting the support condition a measure $\mu \in \mathcal{P}(\mathcal{X} \times \mathbb{Z}_2)$ in terms of the kernel function.

8.4. **Conclusion.** In this section we learn the kernel trick, which simplifies the algorithm of solving hard SVM and soft SVM optimization problem, using embedding of patterns into a Hilbert space. The kernel trick is based on the theory of RKHS and has many applications, e.g., for defining a feature map $\varphi : \mathcal{P}(\mathcal{X}) \to V$, where V is a RHKS, see e.g. [MFSS2016]. The main difficulty of the kernel method is that we still have no general method of selecting a suitable kernel for a concrete problem. Another open problem is to improve the upper bound for sample complexity of SVM algorithm, i.e., to find new conditions on $\mu \in P$ such that the sample complexity of A_{hk}, A_{sk} which is computed w.r.t. μ is bounded.

9. Neural networks

In the last lecture we examined kernel based SVMs which are generalizations of linear classifiers, which are also called *perceptrons*.

Today we shall examine other generalizations of linear classifiers which are artificial neural networks, shortened as neural networks (otherwise, non-artificial neural networks are (called) biological neural networks). The idea behind neural networks is that many neurons can be joined together by communication links to carry out complex computations. Neural networks achieve outstanding performance on many important problems in computer vision, speech recognition, and natural language processing.

Note that under "a neural network" one may think of a computing device, a learning model, or a hypothesis class of a learning model, a class of (sequences of) multivariable functions.

In today lecture we shall investigate several types of neural networks, their expressive power, i.e., the class of functions that can be realized as elements in a hypothesis class of a neural network. In the next lecture we shall discuss the current learning algorithm -stochastic gradient descend - on neural networks.

9.1. **Neural networks as computing devices.** A neural network has a graphical representation for multivariate functions of multi-variables $h_{V,E,\sigma,w} : \mathbb{R}^n \to \mathbb{R}^m$ or a sequence of multivariate functions of multi-variables $\{h^i_{V,E,\sigma,w} : \mathbb{R}^n \to \mathbb{R}^m | \, i \in \mathbb{N}\}$.

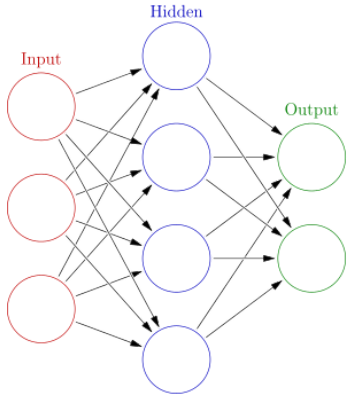
• The quadruple (V, E, σ, w) consists of

+ *the network graph* (V, E), also called *the underlying graph of the network*, where V is the set of nodes $\mathfrak{n}$, also called *neurons*, and E is the set of directed edges connecting nodes of

+ σ - a family of functions $\sigma_{\mathfrak{n}} : \mathbb{R} \to \mathbb{R}$, also called *the activation function of neuron* $\mathfrak{n}$. Usually $\sigma_{\mathfrak{n}} = \sigma$ is independent of $\mathfrak{n}$. 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;

+ $w : E \to \mathbb{R}$ - *the weight function of the network.*

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



• A weight $w : E \to \mathbb{R}$ endows each neuron $\mathfrak{n}$ with a computing instruction of type " input-output". *The input* $I(\mathfrak{n})$ *of a neuron* $\mathfrak{n}$ is equal to the weighted sum of the outputs of all the neurons connected to it: $I(\mathfrak{n}) = \sum w(\mathfrak{n}'\mathfrak{n})O(\mathfrak{n}')$, where $\mathfrak{n}'\mathfrak{n} \in E$ 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.

• There is a neuron in the hidden layer that has no incoming edges. This neuron will output the constant $\sigma(0)$.

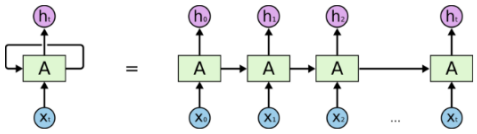
• *A feedforward neural network* (FNN) has underlying acyclic directed graph. Each FNN (E, V, w, σ) represents a multivariate multivariable function $h_{V,E,\sigma,w} : \mathbb{R}^n \to \mathbb{R}^m$ which is obtained by composing the computing instruction of each neuron on directed paths from input neurons to output neurons. For each architecture (V, E, σ) of a FNN 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.

• *A recurrent network* (RNN) has underlying directed graph with a cycle. By unrolling cycles in a RNN in discrete time $n \in \mathbb{N}$, a RNN defines a map $r : \mathbb{N}^+ :\to \{FNN\}$ such that $[r(n)] \subset [r(n+1)]$, where $[r(n)]$ is the underlying

graph of $r(n)$, see [GBC2016, §10.2, p. 368] and [Graves2012, §3.2, p. 22]. Thus a RNN can be regarded as a sequence of multivariate multivariable functions which serves as in a discriminative model for supervised sequence labelling.



An unrolled recurrent neural network.

Digression The goal of supervised sequence labelling is to assign sequences of labels, drawn from a label space, to sequences of input data. For example, one might wish to transcribe a sequence of acoustic features with spoken words (speech recognition), or a sequence of video frames with hand gestures (gesture recognition). Although such tasks commonly arise when analysing time series, they are also found in domains with non-temporal sequences, such as protein secondary structure prediction.

If the sequences are assumed to be independent and identically distributed, we recover the basic framework of pattern classification, only with sequences in place of patterns (of course the data-points within each sequence are not assumed to be independent). In practice this assumption may not be the case.

Example 9.1. *A multilayer perceptron (MLP)* is a type of FNN that 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 MLP is m. V_0 is called *the input layer*, V_n is called *the output layer*, the other layer is called *hidden*.

Remark 9.2. Neural networks are abstraction of biological neural networks, the connection weights w represent the strength of the synapses between the neurons and the activation function σ 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.

In the remaining of today lecture we consider only FNNs. In particular under NN's we mean FNNs.

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

Question 9.3. *What type of functions can be implemented using neural networks.*

First we consider representation of Boolian functions by neural networks.

Proposition 9.4 (Representation of Boolean functions)**.** *(* [SSBD2014, Claim 20.1, p. 271]*) Every Boolean function $f : \mathbb{Z}_2^d \to \mathbb{Z}_2$ can be represented exactly by a feedforward neural network $\mathcal{H}_{V,E,sign}$ with a single hidden layer containing at most $2^d + 1$ neurons and with the activation function $\sigma(x) = sign(x)$.*

Proof. Let (V, E) be a two-layer FNN with $\#V_0 = d + 1$, $\#V_1 = 2^d + 1$, $\#V_2 = 1$ and E consist of all possible edges between adjacent layers. As before $\mathbb{Z}_2 = \{\pm 1\}$. Now let $f : \mathbb{Z}_2^d \to \mathbb{Z}_2$. Let $u_i \in f^{-1}(1) \subset \mathbb{Z}_2^d$ and $k := \#f^{-1}(1)$. Set

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

Then $\{g_i | \, i \in [1, k]\}$ are linear classifiers and therefore can be implemented by the neurons in V_1. Now set

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

which is also a linear classifier. This completes the proof of Proposition 9.4 □

In general case we have the following Universal Approximation Theorem, see e.g. [Haykin2008, p. 167].

Theorem 9.5. *Let φ be a nonconstant, bounded and monotone increasing continuous function. For any $m \in \mathbb{N}$, $\varepsilon > 0$ and any function $F \in C_0([0, 1]^m)$ there exists an integer $m_1 \in \mathbb{N}$ and constants a_i, b_j, w_{ij} where $i \in [1, m_1]$, $j \in [1, m]$ such that*

$$f(x_1, \cdots, x_m) := \sum_{i=1}^{m_1} \alpha_i \varphi(\sum_{j=1}^{m} w_{ij} x_j + b_i)$$

for all $(x_1, \cdots, x_m) \in [0,1]^m$ *we have*

$$|F(x_1, \cdots, x_m) - f(x_1, \cdots, x_m)| < \varepsilon.$$

9.3. **Sample complexities of neural networks.** A learning model $(\mathcal{X} \times \mathcal{Y}, \mathcal{H}_{V,E,\sigma}, L, P)$ is called *a neural network* if $\mathcal{H}_{V,E,\sigma}$ is a neural network and $\mathcal{H}_{V,E,\sigma} \subset \mathcal{Y}^{\mathcal{X}}$.

9.3.1. *Neural networks for binary classification problem.* In neural networks with hypothesis class $\mathcal{H}_{V,E,\sigma}$ for binary classification problems one often choose the activation function σ to be the sign function, and the loss function L to be $L^{(0-1)}$.

Proposition 9.6. *(*[SSBD2014, Theorem 20.6, p. 274]*) Let* $\mathcal{H}_{V,E,sign}$ *be a MLP. The VC-dimension of* $\mathcal{H}_{V,E,sign}$ *is* $O(|E| \log |E|)$.

Outline of the proof If $\mathcal{H} := \mathcal{H}_{V,E,sign}$ consists of exactly one perceptron, then Proposition 9.6 is valid since in this case $VC\dim \mathcal{H} = |E^{in}| = O(|E| \log |E|)$, where E^{in} denotes the set of directed edges coming into the perceptron.

We want to reduce the proof for the general case of a neural network to the case of a single perceptron, using the known VC-dimension of a single perceptron. Let $m := VC\dim(\mathcal{H})$. Using

$$\Gamma_{\mathcal{H}}(m) = 2^m, \tag{9.1}$$

to prove Proposition 9.6, it suffices to show that

$$\Gamma_{\mathcal{H}_{V,E,\sigma}}(m) \leq (em)^{|E|}, \tag{9.2}$$

since $\log_2(em) < 4\log(E)$ by (9.2) and ((9.1)).

Let $V_0, \cdots, V_T$ be the layers of (E, V). For $t \in [1, T]$ denote by $\mathcal{H}_t$ the neural network $\mathcal{H}_{W_t,E_t,sign}$ where W_t consists of inputs neurons in V_{t-1} and output neurons in V_t and E_t consists of edges of $\mathcal{H}$ that connect V_{t-1} with V_t. Now we decompose

$$\mathcal{H} = \mathcal{H}_T \circ \cdots \circ \mathcal{H}_1. \tag{9.3}$$

Lemma 9.7 (Exercises)**.** *(1) (*[SSBD2014, Exercise 4, p. 282]*) Let* $\mathcal{F}_1 \subset \mathcal{Z}^{\mathcal{X}}$ *and* $\mathcal{F}_2 \subset \mathcal{Y}^{\mathcal{Z}}$. *Set* $\mathcal{H} := \mathcal{F}_2 \circ \mathcal{F}_1$. *Then* $\Gamma_{\mathcal{H}}(n) \leq \Gamma_{\mathcal{F}_2}(n)\Gamma_{\mathcal{F}_1}(n)$.

(2) ([SSBD2014, Exercise 3, p. 282]*) Let* $\mathcal{F}_i$ *be a set of function from* $\mathcal{X}$ *to* $\mathcal{Y}_i$ *for* $i = 1, 2$. *Then* $\Gamma_{\mathcal{F}_1 \times \mathcal{F}_2}(n) \leq \Gamma_{\mathcal{F}_1}(n)\Gamma_{\mathcal{F}_2}(n)$.

By Lemma 9.7 (1) we have

$$\Gamma_{\mathcal{H}}(m) \leq \Pi_{t=1}^{T} \Gamma_{\mathcal{H}_t}(m).$$

Next we observe that

$$\mathcal{H}_t = \mathcal{H}_{t,1} \times \cdots \times H_{t,|V_t|}. \tag{9.4}$$

Each neuron $\mathfrak{n}_i$ on V_t has $d_{t,i}$ heading edges presenting the number of the inputs for the linear classifier $\mathfrak{n}_i$. Hence $VC\dim \mathcal{H}_{t,i} = d_{t,i} < m-1$. By Lemma 9.7 and by Vapnik-Chervonenski-Sauer-Lemma we have

$$\Gamma_{\mathcal{H}}(m) \leq \Pi_{t=1}^{T}\Pi_{i=1}^{|V_t|}(\frac{em}{d_{t,i}})^{d_{t,i}} < (em)^{|E|},$$

which completes the proof of Proposition 9.6.

It follows from Proposition 9.6 that the sample complexity of the ERM algorithm for $(V\times\mathbb{Z}_2, \mathcal{H}_{V,E,sign}, L^{(0-1)}, \mathcal{P}(V\times\mathbb{Z}_2))$ is finite. But the running time for ERM algorithm in a neural network $\mathcal{H}_{V,E,sign}$ is non-polynomial and therefore it is impractical to use it [SSBD2014, Theorem 20.7, p. 276]. The solution is to use the stochastic gradient descend, which we shall learn in the next lecture.

9.3.2. *Neural networks for regression problem.* In neural networks with hypothesis class $\mathcal{H}_{V,E,\sigma}$ for regression problems one often chooses the activation function σ to be the sigmoid function $\sigma(a) := (1+e^{-a})^{-1}$, and the loss function L to be L_2, i.e., $L(x,y,h_w) := \frac{1}{2}||h_w(x)-y||^2$ for $h_w \in \mathcal{H}_{V,E,\sigma}$ and $x\in\mathcal{X}=\mathbb{R}^n$, $y\in\mathcal{Y}=\mathbb{R}^m$.

9.3.3. *Neural networks for generative models in supervised learning.* In generative models of supervised learning we need to estimate the conditional distribution $p(t|x)$. In many regression problems p is chosen as follows [Bishop2006, (5.12), p. 232]

$$p(t|x) = \mathcal{N}(t|y(x,w),\beta^{-1}) = \frac{\beta}{\sqrt{2\pi}}\exp\frac{-\beta}{2}(t-y(x,w)), \tag{9.5}$$

where β is unknown parameter and $y(x,w)$ is the expected value of t. Thus the learning model is of the form $(\mathcal{X},\mathcal{H},L,P)$ where $\mathcal{H} := \{y(t,x,w)\}$ parameterized by a parameter w,β, and a statistical model P is a subset of $\mathcal{P}(\mathcal{X})$, since the joint distribution $\mu(x,y)$ is completely defined by $\mu_{\mathcal{X}}$ and the conditional distribution $\mu(y|x)$.

Now assume that $X=(x_1,\cdots,x_n)$ are i.i.d. by $\mu\in P$ along with labels $(t_1,\cdots,t_n)$. Then (9.5) implies

$$-\log p(t|X,w,\beta) = -\frac{n}{2}\log\beta + \frac{n}{2}\log(2\pi) + \frac{\beta}{2}\sum_{i=1}^{n}|t_n - y(x_n,w)|^2. \tag{9.6}$$

As in the density estimation problem we want to minimize the LHS of (9.5). Leaving $\beta = const$ we minimize first the β -independent component of the loss function

$$L_S(w) = \frac{1}{2}\sum_{i-1}^{n}|y(x_n,w)^2 - t_n|^2. \tag{9.7}$$

Once we know a solution w_{ML} of the equation minimizing $L_S(w)$, the value of β can be found by the following formula

$$\frac{1}{\beta_{ML}} = \frac{1}{n}\sum_{i=1}^{n} |y(x_i, w_{ML}) - t_i|^2. \tag{9.8}$$

9.4. **Conclusion.** In this lecture we considered learning machines whose hypothesis class consisted of functions or sequence of functions that can be graphical represented by neural networks. Neural networks have good expressive power and finite VC-dimension in binary classification problems but the ERM algorithm in these networks has very high computational complexity and therefore they are unpractical.

10. Training neural networks

Training a neural network is a popular name for running a learning algorithm in a neural network learning model. We consider in this lecture only the case where the input space and the output space of a network are Euclidean spaces $\mathbb{R}^n$ and $\mathbb{R}^m$ respectively. Our learning model is of the form $(\mathbb{R}^n \times \mathbb{R}^m, \mathcal{H}_{V,E,\sigma}, L, P)$ and the learning algorithm is stochastic gradient descend (SGD), which aims to find a minimizer of the expected risk function $R_\mu^L : \mathcal{H}_{V,E,\sigma} \to \mathbb{R}$. Since $\mathcal{H}_{V,E,\sigma}$ is parameterized by the weight function $w \in \mathbb{R}^E \cong \mathbb{R}^{|E|}$, we regard R_μ^L as a function of variable w on $\mathbb{R}^N$, where $N = |E|$. We begin with classical (deterministic) gradient and subgradient descend of a function on $\mathbb{R}^N$ and then analyze the SGD if the loss function L is convex. In this case we get an upper bound for the sample complexity of SGD. Finally we discuss SGD in general FNNs.

10.1. **Gradient and subgradient descend.** For any differentiable function f on a $\mathbb{R}^N$ denote by $\nabla_g f$ the gradient of f w.r.t. a Riemannian metric g on $\mathbb{R}^N$, i.e., for any $x \in \mathbb{R}^N$ and any $V \in \mathbb{R}^N$ we have

$$df(V) = \langle \nabla_g f, X \rangle. \tag{10.1}$$

If g is fixed, for instance g is the standard Euclidean metric on $\mathbb{R}^N$, we just write ∇f instead of $\nabla_g f$.

The negative gradient flow of f on $\mathbb{R}^N$ is a dynamic system on $\mathbb{R}^N$ defined by the following ODE with initial value $w_0 \in \mathbb{R}^N$

$$w(0) = w_0 \in \mathbb{R}^N \text{ and } \dot{w}(t) = -\nabla f(w(t)). \tag{10.2}$$

If $w(t)$ is a solution of (10.2) then $f(w(t)) < f(w(t'))$ for any $t' > t$ unless $\nabla f(w(t)) = 0$, i.e., $w(t)$ is a critical point of f.

If f is not differentiable we modify the notion of the gradient of f as follows.

Definition 10.1. 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

$$\forall u \in S,\ f(u) \geq f(w) + \langle u - w, v \rangle. \tag{10.3}$$

The set of subgradients of f at w is called *the differential set* and denoted by $\partial f(w)$.

Exercise 10.2. (1) Show that if f is differentiable at w then $\partial f(w)$ contains a single element $\nabla f(w)$.

(2) Find a subgradient of the generalized hinge loss function $f_{a,b,c}(w) = \max\{a, 1 - b\langle w, c\rangle\}$ where $a, b \in \mathbb{R}$ and $w, c \in \mathbb{R}^N$ and $\langle \cdot, \cdot \rangle$ a scalar product.

Remark 10.3. 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 descend algorithm* discretizes the solution of the gradient flow equation (10.2). We begin with an arbitrary initial point $x_0 \in \mathbb{R}^N$. We set

$$w_{n+1} = w_n - \gamma_n \nabla f(w_n), \tag{10.4}$$

where $\gamma_n \in \mathbb{R}_+$ is a constant, called a "learning rate" in machine learning, to be optimized. This algorithm can be slightly modified. For example, after T iterations we set the output point $\bar{w}_T$ to be

$$\bar{w}_T := \frac{1}{T} \sum_{i=1}^{T} w_i, \tag{10.5}$$

or

$$\bar{w}_T := \arg \min_{i \in [1,T]} f(w_i). \tag{10.6}$$

If a function f on $\mathbb{R}^N$ has a critical point which is not the minimizer of f, then the gradient flow (10.2) and its discrete version (10.4) may not yield the required minimizer of f. If f is convex, then f has only a unique critical point w_0 which is also the minimizer of f. In fact we have the following stronger assertion.

$$f(w) - f(u) \leq \langle w - u, \nabla f(w) \rangle \text{ for any } w, u \in \mathbb{R}^N. \tag{10.7}$$

It also follows from (10.7) that there exists a unique minimizer of f, and hence the gradient flow (10.2) works. Its discrete version (10.4) also works, as stated in the following.

Proposition 10.4. *(*[SSBD2014, Corollary 14.2, p. 188]*) Let f be a convex ρ-Lipschitz function on $\mathbb{R}^N$,*[18] *and let $w^* \in \arg\min_{w \in B(0,r) \subset \mathbb{R}^N} f(w)$. If we run the GD algorithm (10.4) on f for T steps with $\gamma_t = \eta = \frac{r}{\rho\sqrt{T}}$ for $t \in [1, T]$, then the output $\bar{w}_T$ defined by (10.5) satisfies*

$$f(\bar{w}_T) - f(w^*) \leq \frac{r\rho}{\sqrt{T}}.$$

[18] i.e., $|f(w) - f(u)| \leq \rho\,|w - u|$

Under the conditions in Proposition 10.4, for every $\varepsilon > 0$, to achieve $f(\bar{w}_T) - f(w^*) \leq \varepsilon$, it suffices to run the GD algorithm for a number of iterations that satisfies

$$T \geq \frac{r^2\rho^2}{\varepsilon^2}.$$

Lemma 10.5. *([SSBD2014, Lemma 14.1, p. 187]) Let $w^*, v_1, \cdots, v_T \in \mathbb{R}^N$. Any algorithm with an initialization $w_1 = 0$ and*

$$w_{t+1} = w_t - \eta v_t \tag{10.8}$$

satisfies

$$\sum_{i=1}^{T} \langle w_t - w^*, v_t \rangle \leq \frac{||w^*||^2}{2\eta} + \frac{\eta}{2}\sum_{t=1}^{T} ||v_t||^2. \tag{10.9}$$

In particular, for every $r, \rho > 0$, if for all $t \in [1, T]$ we have $||v_t|| \leq \rho$ and if we set $\eta = (r/\rho)T^{-1/2}$ then if $||w^*|| \leq r$ we have

$$\frac{1}{T}\sum_{t=1}^{T} \langle w_t - w^*, v_t \rangle \leq \frac{r\rho}{\sqrt{T}}. \tag{10.10}$$

To apply Lemma 10.5 to Proposition 10.4 we set $v_t := \nabla f(w_t)$ and note that $||\nabla f(w_t)|| \leq \rho$ since f is ρ-convex, moreover

$$\begin{aligned} f(\bar{w}_T) - f(w^*) &= f(\frac{1}{T}\sum_{t=1}^{T} w_t) - f(w^*) \\ \overset{\text{since } f \text{ is convex}}{\leq} & \frac{1}{T}\sum_{t=1}^{T} f(w_t) - f(w^*) = \frac{1}{T}\sum_{i=1}^{T}(f(w_t) - f(w^*)) \\ \overset{by(10.7)}{\leq} & \frac{1}{T}\sum_{i=1}^{T}\langle w_t - w^*, \nabla f(w_t)\rangle. \end{aligned}$$

• *Subgradient descend algorithm.* Comparing (10.3) with (10.7), taking into account the technical Lemma 10.5, we conclude that the gradient descend algorithm can be applied the case of non-differentiable function f that has subgradient at every point.

10.2. **Stochastic gradient descend (SGD).** Let $(\mathcal{Z}, \mathcal{H}, L, P)$ be a learning model. Given a sample $S := (z_1, \cdots, z_n) \in \mathcal{Z}^n$ consisting of observables z_i that are i.i.d. by $\mu \in P$, a SGD searches for an approximate minimizer $h_S \in \mathcal{H}$ of the function $R_\mu^L : \mathcal{H} \to \mathbb{R}$ using the following formula of "differentiation under integration"

$$\nabla R_\mu^L(h) = \int_{\mathcal{Z}} \nabla_h L(h, z)\, d\mu(z) \tag{10.11}$$

if L is differentiable. Thus $\nabla R^L_\mu(h)$ can be computed in two steps. First we compute $\nabla_h L(h, z_i)$ for $z_i \in S$. Then we approximate the RHS of (10.11) by the empirical gradient $\frac{1}{n}\sum_{z_i\in S}\nabla_h L(h, z_i)$ which is equal to the gradient of the empirical risk function.

$$\nabla \hat{R}^L_S(h) = \frac{1}{n}\sum_{z_i\in S}\nabla_h L(h, z_i).$$

Next we apply the algorithm for gradient flow described above to $\nabla \hat{R}^L_S(h)$. The weak law of large numbers ensures the convergence in probability of $\nabla \hat{R}^L_S(h)$ to RHS of (10.11), and heuristically the convergence of the empirical gradient descend algorithm to the gradient descend of the expected risk function R^L_μ.

There are several versions of SGD with minor modifications.

For simplicity and applications in NN we assume $\mathcal{Z} = \mathbb{R}^n \times \mathbb{R}^m$, $\mathcal{H} := \mathcal{H}_{E,V,\sigma}$ is parameterized by $w \in \mathbb{R}^N$ and L is differentiable in w. A version of SGD works as follows.

1) Choose a parameter $\eta > 0$ and $T > 0$.
2) Assume that $S = (z_1, z_2, \cdots, z_n) \in \mathcal{Z}^n$. Take arbitrary $z \in S$.
3) Set $w_1 = 0 \in \mathbb{R}^N$.
4) $w_{t+1} := w_t - \eta\nabla_w L(w_t, z)$.
5) Set the output $\bar{w}_T(z) := \frac{1}{n}\sum_{t=1}^n w_t$.

Proposition 10.6. *(*[SSBD2014, Corollary 14.12, p. 197]*) Assume L is a convex function in variable w and $\mu \in P$ governs the probability distribution of i.i.d. $z_i \in S \in (\mathbb{R}^n \times \mathbb{R}^m)$. Assume that $r, \rho \in \mathbb{R}_+$ are given with the following properties.*

1) $w^* \in \arg\min_{w\in B(0,r)} R^L_\mu\big(w\big)$.

2) The SGD is run for T iterations with $\eta = \sqrt{\frac{r^2}{\rho^2 T}}$.

3) For all $t \in [1, T]$ we have $\mathbb{E}_\mu\big(||\nabla_w L(w_t, z)||\big) \leq \rho$ (e.g., $||\nabla_w L(w_t, z)|| \leq \rho$ for all z).

4) Assume that $T \geq \frac{r^2\rho^2}{\varepsilon^2}$.

Then

$$\mathbb{E}_\mu\Big(R^L_\mu\big(\bar{w}_T(z)\big)\Big) \leq R^L_\mu\big(h(w^*)\big) + \varepsilon. \tag{10.12}$$

Exercise 10.7. Find an upper bound for the sample complexity of the SGD in Proposition 10.6.

Example 10.8. Let us consider layered FNN with $\mathcal{H} = H_{E,V,\sigma}$ where $V = V_0 \cup V_1 \cup \cdots \cup V_T$. For the loss function

$$L(x, y, w) := \frac{1}{2}||h_w(x) - y||^2$$

and a vector $v \in \mathbb{R}^N$ on $\mathbb{R}^n$ we compute the gradient of L w.r.t. the Euclidean metric on $\mathbb{R}^N$, regarding x, y as parameters:

$$\langle \nabla L(x, y, w), v\rangle = \langle h_w(x) - y, \nabla_v h_w(x)\rangle.$$

To compute $\nabla_v h_w(x) = dh(v)$ we decompose $h_w = h_T \circ \cdots \circ h_1$ as in (9.3) and using the chain rule

$$d(h_T \circ \cdots \circ h_1)(v) = dh_T \circ \cdots dh_1(v).$$

To compute dh_i we use the decomposition (9.4)

$$d(h_{t,1} \times \cdots \times h_{t,|V_t|}) = dh_{t,1} \times \cdots \times dh_{t,|V_t|}.$$

Finally for $h_{t,j} = \sigma(\sum a_j x_j)$ we have

$$dh_{t,j} = d\sigma \circ (\sum a_j dx_j).$$

The algorithm for computing the gradient ∇L w.r.t. w efficiently is called *backpropogation*. [19]

Remark 10.9. (1) In a general FNN the loss function L is not convex therefore we cannot apply Proposition 10.6. Training FNN is therefore subject to experimental tuning.

(2) Training a RNN is reduced to training of sequence of FNN given a sequence of labelled data, see [Haykin2008, §15.6, p. 806] for more details.

10.3. **Online gradient descend and online learnability.** For training neural networks one also use Online Gradient Descend (OGD), which works as an alternative method of SGD [SSBD2014, p. 300]. Let $L : \mathbb{R}^N \times \mathcal{Z} \to \mathbb{R}$ be a loss function. A version of OGD works almost like SGD
1) We choose a parameter $\eta > 0$ and $T > 0$.
2) A sample $S = (z_1, \cdots, \cdots z_T) \in \mathcal{Z}^T$ is given.
3) Set $w_1 = 0$.
4) For t$\in [1, T]$ set $v_t := \nabla_w f(w_t, z_t)$.
5) Set $w_t := w_t - \eta v_t$.

Despite on their similarity, at the moment there is no sample complexity analysis of OGD. Instead, ML community develops a concept of online learnability for understanding OGD.

10.3.1. *Setting of online-learning.* Let $(\mathcal{X}\times\mathcal{Y}, \mathcal{H}, L, P)$ be a supervised learning model. The general on-line learning setting involves T rounds. At the t-th round, $1 \le t \le T$, 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 find a predictor $A(x_t)$ that minimizes the *cumulative loss*, which is an analogue of the notion of empirical risk in our unified learning model $R_A(T) := \sum_{i=1}^T L(A(x_t), y_t)$ over T rounds [MRT2012, p. 148].

[19] According to [Bishop2006, p. 241] the term "backpropogation" is used in the neural computing literature to mean a variety of different things.

In the case of 0-1 loss function $L^{(0-1)}$ the value $R_A(T)$ is called the number of mistakes that A makes after T rounds.

Definition 10.10 (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 10.11. 1) Similarly we also have the notion of a successful online learner in regression problems [SSBD2014, p. 300] and within this concept online gradient descent is a successful online learner whenever the loss function is convex and Lipschitz.

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 expected risk. So there is an open question if we can make explain it using statistical learning theory.

10.4. **Conclusion.** In this section we study stochastic gradient descend as a learning algorithm which works if the loss function is convex. To apply stochastic gradient flow as a learning algorithm in FNN where the loss function is not convex one needs experimentally modify the algorithm so it does not stay in a critical point which is not the minimizer of the empirical risk function. One also trains NN with online gradient descends for which we need a new concept of online learnability which has not yet interpreted using probability framework.

11. Bayesian machine learning

Under "Bayesian learning" one means application of Bayesian statistics to statistical learning theory. Bayesian inference is an approach to statistics in which all forms of uncertainty are expressed in terms of probability. Ultimately in Bayesian statistics we regard all unknown quantities as random variables and we consider a joint probability distribution for all of them, which contains the most complete information about the correlation between the unknown quantities.

11.1. **Bayesian concept of learning.** A Bayesian approach to a problem starts with the formulation of a model that we hope is adequate to describe the situation of interest. We then formulated a prior distribution over the unknown parameters of the model, which is meant to capture our beliefs about the situation before seeing the data. After observing some data, we apply Bayes' Theorem A.6, to obtain a posterior distribution for these unknowns, which takes account of both the prior and the data. From this posterior distribution we can compute predictive distributions for future observations using (11.1).

To predict the value of an unknown quantity z^{n+1}, given a sample $(z^1, \cdots, z^n)$, a prior distribution $d\theta \in \mathcal{P}(\Theta)$, one uses the following formula

$$P(z^{n+1}|\, z^1, \cdots, z^n) = \int P(z^{n+1}|\theta) P(\theta|z^1, \cdots, z^n) d\theta \tag{11.1}$$

which is a consequence of formula (A.8). The conditional distribution $P(z^{n+1}|\theta)$ is called *sampling distribution of data* z^{n+1} which is assumed to known, the conditional probability $P(\theta|z^1, \cdots, z^n)$ is called *posterior distribution of* θ *after observing* $(z^1, \cdots, z^n)$.

Example 11.1 (Bayesian neural networks)**.** ([Neal1996, §1.1.2, p. 5]) In Bayesian neural network the aim of a learner is to find a conditional probability $P(y|x^{n+1}, (x^1, y^1), \cdots, (x^n, y^n))$, where y is a label, x^{n+1} is a new input and $\{(x^i, y^i)|\, i = 1, n\}$ is training data. Let θ be a parameter of the neural network. Then we have

(11.2)

$$P(y|\, x^{n+1}, (x^1, y^1), \cdots, (x^n, y^n)) = \int P(y|\, x^{n+1}, \theta) P(\theta|\, (x^1, y^1), \cdots, (x^n, y^n))\, d\theta.$$

The conditional sampling probability $P(y|x^{n+1}, \theta)$ is assumed to known. Hence we can compute the LHS of (11.2), which is called *predictive distribution of* y.

11.2. **Estimating decisions using posterior distributions.** In statistical decision theory we consider the problem of making optimal decisions, that is, decisions or actions that minimize our expected loss. For example, in our unified learning model $(\mathcal{Z}, \mathcal{H}, L, P)$ our learning algorithm A should minimize the expected loss $R^L_\mu : \mathcal{H} \to \mathbb{R}$, which is the average of the instantaneous loss function $L : \mathcal{H} \times \mathcal{Z} \to \mathbb{R}$ over the sample space $\mathcal{Z}$ using the unknown probability measure μ that governs the distribution of observable $z \in \mathcal{Z}$. Since we shall not consider asymptotic theory in Bayesian decision theory and in Bayesian machine learning, see Remark 11.7 below, our discussion of Bayesian optimal decisions should be compared with our discussion on efficient estimators in Subsection 4.4.

In Bayesian decision theory we consider a decision model (or learning model) $(\mathcal{X}, \Theta, \mathcal{D}, L, \pi \in \mathcal{P}(\Theta))$ where

- $\mathcal{X}$ is a sample space, also called observation space,
- a family of conditional sampling distributions $\{P_\theta \in \mathcal{P}(\mathcal{X}), |\, \theta \in \Theta\}$ is given
- Θ is the parameter space with given a prior distribution $\pi(\theta)$,
- $\mathcal{D}$ is a decision space (e.g., $\mathcal{D} = h(\Theta)$ is a "feature space" of Θ) we are interested to learn, observing $x \in \mathcal{X}$,
- $L : \Theta \times \mathcal{D} \to \mathbb{R}$ is an instantaneous loss that measures the discrepancy between $\theta \in \Theta$ and $d \in \mathcal{D}$,
- *the posterior expected loss* is defined as follows

$$\rho^L(\pi, d|x) := \int_\Theta L(\theta, d) d\pi(\theta|x) \tag{11.3}$$

• *the integrated risk* is defined as follows

$$r^L(\pi, \delta) = \int_\Theta \int_{\mathcal{X}} L(\theta, \delta(x))\, d\mu(x|\theta) d\pi(\theta), \tag{11.4}$$

where $\delta : \mathcal{X} \to \mathcal{D}$ is an estimator.

Theorem 11.2. *(*[Robert2007, Theorme 2.3.2]*) An estimator minimizing the integrated risk $r^L(\pi, \delta)$ can be obtained by selecting for every $x \in \mathcal{X}$ the value $\delta(x)$ which minimizes the posterior expected risk $\rho^L(\pi, \delta|x)$.*

Theorem 11.2 says that the posterior loss function and the integrated risk function lead to the same decision and the Bayesian approach agrees with the classical approach. It also leads to the following definition.

Definition 11.3. A Bayes estimator associated with a prior distribution π and a loss function L is any estimator δ^π which minimizes $r(\pi, \delta)$. For every $x \in \mathcal{X}$ it is given by $\delta^\pi := \arg\min \rho(\pi, d|x)$. The value $r(\pi) := r(\pi, \delta^\pi)$ is called *the Bayes risk*.

Example 11.4. The most common risk function used for Bayesian estimation is the mean square error (MSE), which is defined by $L(\theta, d) = |\theta - d|^2$. In this case the Bayes estimator is defined the mean of the posterior distribution (cf. Exercise 2.7)

$$\delta^\pi(x) = \mathbb{E}_\pi[\theta|x]. \tag{11.5}$$

Exercise 11.5. (cf. [Robert2007, Proposition 2.5.7, p. 81]) Prove that the Bayes optimal predictor in exercise 2.6 is a Bayes estimator.

Remark 11.6. Following Bayesian ultimately probabilistic approach we also define the notion of *randomized estimators*, also called *statistical decision rule* [Chentsov1972, Definition 5.1, p. 65], which is a stochastic/probabilistic map from a sample space $\mathcal{X}$ taking value in the space $\mathcal{D}$. Fortunately we don't need to consider randomized estimators since there is a known theorem that the Bayes risk on the set of randomized estimators is the same as the Bayes risk on the set of nonrandomized estimators [Robert2007, Theorem 2.4.2, p. 66].

Remark 11.7 (Asymptotic properties of Bayes learning algorithm)**.** ([Robert2007, p. 48]) In Bayesian decision theory one did not consider asymptotic theory. Firstly, the Bayesian point of view is intrinsically conditional. When conditioning on the observation $S \in \mathcal{X}^n$, there is no reason to wonder what might happen if n goes to infinity since n is fixed by the sample size. Theorizing on future values of the observations thus leads to a frequentist analysis, opposite to the imperatives of the Bayesian perspective. Secondly, even though it does not integrate asymptotic requirements, Bayesian procedures perform well in a vast majority of cases under asymptotic criteria. In a general context, Ibragimov and Hasminskii show that Bayes estimators are consistent [IH1981, chapter 1].

11.3. **Bayesian model selection.** In Example 11.1 we gives an example of using Bayesian methods to learning using neural network. Another application of Bayesian methods is model selection. First we enumerate all reasonable models of the data and assigning a prior belief μ_i to each of these models M_i. Then, upon observing the data x you evaluate how probable the data was under each of these models to compute $P(x|\mu_i)$. To compare two models M_i with M_j, we need to compute their relative probability given the data: $\mu_i P(x|M_i)/\mu_j P(x|M_j)$.

11.4. **Conclusion.** In our lecture we considered main ideas and some applications of Bayesian methods in machine learning. Bayesian machine learning is an emerging promising trend in machine learning that is well suitable for solving complex problems on one hand and consistent with most basic techniques of non-Bayesian machine learning. There are several problems in implementing Bayesian approach, for instance to translating our subjective prior beliefs into a mathematically formulated model and prior. There may also computational difficulties with the Bayesian approach.

Appendix A. Some basic notions in probability theory

Basis objects in probability theory (and mathematical statistics) are measurable spaces $(\mathcal{X}, \Sigma)$, where Σ is a σ-algebra of subsets of a space $\mathcal{X}$. A countably additive measure μ on Σ is called *a probability measure* if $\mu \geq 0$ and $\mu(\mathcal{X}) = 1$.

For this Appendix I use [Bogachev2007] as my main reference on measure theory and [Schervish1997] for theoretical statistics, see also the book [JP2003] for a clear and short exposition of probability theory and [AJLS2015, AJLS2017, AJLS2018] for geometric approach in statistics.

A.1. **Dominating measures and the Radon-Nikodym theorem.** Let μ and ν be countably additive measures on a measurable space $(\mathcal{X}, \Sigma)$
(i) The measure ν is called *absolutely continuous with respect to* μ (or *dominated by* μ) if $|\nu|(A) = 0$ for every set A with $|\mu|(A) = 0$. Notation: $\nu << \mu$.
(ii) The measure ν is called *singular with respect to* μ, if there exists a set $\Omega \in \Sigma$ such that

$$|\mu|(\Omega) = 0 \text{ and } |\nu|(X \setminus \Omega) = 0.$$

Notation: $\nu \perp \mu$.

Theorem A.1. *(cf.* [Bogachev2007, Theorem 3.2.2, vol 1, p. 178]*) Let μ and ν be two finite measures on a measurable space $(\mathcal{X}, \Sigma)$. The measure ν is dominated by the measure μ precisely when there exists a μ-integrable function f such that ν is given by*

$$\nu(A) = \int_A f \, d\mu \tag{A.1}$$

for each $A \in \Sigma$.

We denote ν by $f\mu$ for μ, ν, f satisfying the equation (A.1). The function f is called the (Radon-Nikodym) density (or the Radon-Nikodym derivative) of ν w.r.t. μ. The function f is denoted by $d\nu/d\mu$.

A.2. **Conditional expectation and regular conditional measure.**

A.2.1. *Conditional expectation.* In this section we define the notion of conditional expectation using the Radon-Nykodym theorem. We note that any sub-σ-algebra $\mathcal{B} \subset \Sigma$ can be written as $\mathcal{B} = Id^{-1}(\mathcal{B})$ where $Id : (\mathcal{X}, \Sigma) \to (\mathcal{X}, \mathcal{B})$ is the identity mapping. In what follow we let $\mathcal{B} := \sigma(\eta)$ where $\eta : (\mathcal{X}, \Sigma) \to (\mathcal{Y}, \Sigma')$ is a measurable map.

$$\begin{array}{ccc} (\mathcal{X}, \Sigma) & \xrightarrow{\eta} & (\mathcal{Y}, \Sigma') \\ \downarrow{\scriptstyle Id_\eta} & \nearrow{\scriptstyle \eta} & \\ (\mathcal{X}, \eta^{-1}(\Sigma')) & & \end{array}$$

Definition A.2. Let $\mu \in \mathcal{P}(\mathcal{X})$ and $f \in L^1(\mathcal{X}, \mu)$. The conditional expectation $\mathbb{E}^{\sigma(\eta)} f$ is defined as follows

$$\mathbb{E}_\mu^{\sigma(\eta)} f := (Id_\eta)_*^\mu f := \frac{d(Id_\eta)_*(f\mu)}{d(Id_\eta)_*(\mu)} \in L^1(\mathcal{X}, \eta^{-1}(\Sigma'), (Id_\eta)_*\mu). \tag{A.2}$$

It follows immediately from (A.2) that for all $x \in \mathcal{X}$ we have

$$\mathbb{E}^{\sigma(\eta)} f(x) = g(\eta(x)) \text{ where } g = \eta_*^\mu f := \frac{d\eta_*(f\mu)}{d\eta_*(\mu)} \in L^1(\mathcal{Y}, \Sigma', \eta_*\mu). \tag{A.3}$$

In the probabilistic literature for $\mathcal{B} = \sigma(\eta)$ one uses the notation

$$\mathbb{E}(f|\mathcal{B}) := \mathbb{E}_\mu^{\mathcal{B}} f.$$

Remark A.3. There are many approaches to conditional expectations. Formula (A.2), defined in [AJLS2015], and Formula (A.7) defined in [Halmos1950, p. 207] using the Radon-Nykodym theorem, are the simplest.

A.2.2. *Conditional measure and conditional probability.* Given a measure μ on $(\mathcal{X}, \Sigma)$ *the conditional measure* (or *conditional probability* in the case of probability measure μ) *of* $A \in \Sigma$ *w.r.t.* $\mathcal{B}$, is defined as follows

$$\mu(A|\mathcal{B}) := \mathbb{E}_\mu(1_A|\mathcal{B}). \tag{A.4}$$

In probabilistic literature one omits μ in (A.4) and writes instead

$$P(A|\mathcal{B}) := \mu(A|\mathcal{B}).$$

If $\mathcal{B} = \eta^{-1}(\Sigma')$ where $\eta : (\mathcal{X}, \Sigma) \to (\mathcal{Y}, \Sigma')$ is a measurable map, one uses the notation

$$P(A|\eta) := \mu(A|\eta) := \mu(A|\mathcal{B}).$$

For any $A \in \Sigma$ and $B \in \mathcal{B} = \eta^{-1}(\Sigma')$ formulas (A.2) and (A.1) imply

$$\mu(A \cap B) = \int_B \mu(A|\mathcal{B}) d\mu. \tag{A.5}$$

By (A.3) the measurable function $\zeta_A := \eta^{\mu}_*(1_A) : \mathcal{Y} \to \mathbb{R}$ satisfies the following relation

$$\mu(A|\eta)(x) = \zeta_A(\eta(x))$$

for all $x \in \mathcal{X}$. Then one sets for $y \in \mathcal{Y}$

$$\mu^{\mathcal{B}}(A|y) := \mu^{\mathcal{B}}(A|\eta(x) = y) := \zeta_A(y). \tag{A.6}$$

The RHS of (A.6) is called *the measure of A under conditioning $\eta = y$.*

We also rewrite formula (A.6) as follows for $E \in \Sigma'$

$$\mu(A \cap \eta^{-1}(E)) = \int_E \mu^y(A)\, d\eta_*(\mu)(y) \tag{A.7}$$

where $\mu^y(A) := \mu^{\mathcal{B}}(A|y)$ is a Σ'-measurable function.

Note that it is not always the case that for $\eta_*(\mu)$-almost all $y \in \mathcal{Y}$ the set function $\mu^y(A)$ is countably addictive measure, see Example 10.4.9 in [Bogachev2007, p. 367, v.2]. Nevertheless this becomes possible under some additional conditions on set-theoretic or topological character, see [Bogachev2007, Theorems 10.4.5, 10.4.8, v. 2].

A.2.3. *Regular conditional measure.*

Definition A.4. [Bogachev2007, Definition 10.4.1, p. 357]) Suppose we are given a sub-σ-algebra $\mathcal{B} \subset \Sigma$. A function

$$\mu^{\mathcal{B}}(\cdot, \cdot) : \Sigma \times \mathcal{X} \to \mathbb{R}$$

is called *a regular conditional measure* on Σ w.r.t. $\mathcal{B}$ if
(1) for every $x \in \mathcal{X}$ 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 *$\mathcal{B}$-measurable and μ-integrable*,
(3) For all $A \in \Sigma$, $B \in \mathcal{B}$ the following formula holds, cf (A.5)

$$\mu(A \cap B) = \int_B \mu^{\mathcal{B}}(A, x) d\mu(x). \tag{A.8}$$

Remark A.5. Assume that $\mu^{\mathcal{B}}(A, x)$ is a regular conditional measure. Formulas (A.8) and (A.5) imply that $\mu^{\mathcal{B}}(A, x) : \mathcal{X} \to \mathbb{R}$ is a representative of the conditional measure $\mu(A|\mathcal{B}) \in L^1(\mathcal{X}, \mathcal{B}, (Id_\eta)_*(\mu))$. Thus one also uses the notation $\mu(A|x)$ instead of $\mu(A, x)$.

A.3. **Joint distribution and Bayes' theorem.** Till now we define conditional probability measure $\mu(A|\mathcal{B})$ on a measurable space $(\mathcal{X}, \Sigma, \mu)$ where $\mathcal{B}$ is a sub σ-algebra of Σ. We can also define conditional probability measure $\mu_{\mathcal{X}\times\mathcal{Y}}(A|\mathcal{B})$ where A is a subset of the σ-algebra $\Sigma_{\mathcal{X}}$ of a measurable space $\mathcal{X}$ and $\mathcal{B}$ is a sub-σ algebra of the σ-algebra $\Sigma_{\mathcal{Y}}$ of a measurable space $\mathcal{Y}$, if a joint probability measure $\mu_{\mathcal{X}\times\mathcal{Y}}$ on $(\mathcal{X} \times \mathcal{Y}, \Sigma_{\mathcal{X}} \times \Sigma_{\mathcal{Y}})$ is given. This can be done by push forwarding the measure $\mu_{\mathcal{X}\times\mathcal{Y}}$ to $\mathcal{X}$ and $\mathcal{Y}$ and lifting as in Exercise 2.7. There are several sufficient conditions on $\Sigma_{\mathcal{X}}$ for the existence of regular conditional measure $\mu(A|y)$ for any $A \in \Sigma_{\mathcal{X}}$ and $y \in \mathcal{Y}$,

see Corollaries A.9, A.10 below and Theorem 10.4.14 in [Bogachev2007, p. 364, v. 2].

The Bayes theorem stated below assumes the existence of regular conditional measure $\mu_{\mathcal{X}|\Theta}(A|\theta)$, [20] where μ is a joint distribution of random elements $x \in \mathcal{X}$ and $\theta \in \Theta$. Furthermore we also assume the condition that there exists a measure $\nu \in \mathcal{P}(\mathcal{X})$ such that ν dominates $\mu_\theta := \mu(\cdot|\theta)$ for all $\theta \in \Theta$.

Theorem A.6. *[Bayes' theorem]([Schervish1997, Theorem 1.31, p. 16]) Suppose that $\mathcal{X}$ has a parametric family $\{P_\theta | \, \theta \in \Theta\}$ such that $P_\theta << \nu$ for some $\nu \in \mathcal{P}(\mathcal{X})$ for all $\theta \in \Theta$. Let $f_{\mathcal{X}|\Theta}(x|\theta)$ denotes the conditional density of P_θ w.r.t. ν. Let μ_Θ be the prior distribution of Θ and let $\mu_{\Theta|\mathcal{X}}(\cdot|x)$ the conditional distribution of Θ given x. Then $\mu_{\Theta|\mathcal{X}} << \mu_\Theta$ ν- a.s. and the Radon-Nykodim derivative is*

$$\frac{d\mu_{\Theta|\mathcal{X}}}{d\mu_\Theta}(\theta|x) = \frac{f_{\mathcal{X}|\Theta}(x|\theta)}{\int_\Theta f_{\mathcal{X}|\Theta}(x|t)\, d\mu_\Theta(t)}$$

for those x s.t. the dominator is neither 0 or infinite. The prior predictive probability of the set of x values s.t. the dominator is 0 or infinite is 0, hence the posterior can be defined arbitrary for such x values.

A.4. **Transition measure, Markov kernel, and parameterized statistical model.** Regular conditional measures in Definition A.4 are examples of transitions measures for which we shall have a generalized version of Fubini theorem (Theorem A.8)

Definition A.7. ([Bogachev2007, Definition 10.7.1, vol. 2, p. 384]) Let $(X_1, \mathcal{B}_1)$ and $(\mathcal{X}, \mathcal{B}_2)$ be a pair of measurable spaces. *A transition measure for this pair* is a function $P(\cdot|\cdot) : \mathcal{X}_1 \times \mathcal{B}_2 \to \mathbb{R}$ with the following properties:

(i) for every fixed $x \in \mathcal{X}_1$ the function $B \mapsto P(x|B)$ is a measure on $\mathcal{B}_2$;

(ii) for every fixed $B \in \mathcal{B}_2$ the function $x \mapsto P(x|B)$ is measurable w.r.t. $\mathcal{B}_1$.

In the case where transition measures are probabilities in the second argument, they are called *transition probabilities*. In probabilistic literature transition probability is also called *Markov kernel*, or *(probability) kernel* [Kallenberg2002, p. 20].

Theorem A.8. *(*[Bogachev2007, Theorem 10.7.2, p. 384, vol. 2]*) Let $P(\cdot|\cdot)$ be a transition probability for spaces $(\mathcal{X}_1, \mathcal{B}_1)$ and $(\mathcal{X}_2, \mathcal{B}_2)$ and let ν be a probability measure on $\mathcal{B}_1$. Then there exists a unique probability measure μ on $(\mathcal{X}_1 \times \mathcal{X}_2, \mathcal{B}_1 \otimes \mathcal{B}_2)$ with*

$$\mu(B_1 \times B_2) = \int_{B_1} P(x|B_2) d\nu(x) \text{ for all } B_1 \in \mathcal{B}_1,\ B_2 \in \mathcal{B}_2. \tag{A.9}$$

[20] Schervish considered only parametric family of conditional distributions [Schervish1997, p.13]

In addition, given any $f \in L^1(\mu)$ for ν- a.e. $x_1 \in \mathcal{X}_1$ the function $x_2 \mapsto f(x_1, x_2)$ on $\mathcal{X}_2$ is measurable w.r.t. the completed σ-algebra $(\mathcal{B}_2)_{P(x_1|\cdot)}$ and $P(x_1|\cdot)$-integrable, the function

$$x_1 \mapsto \int_{\mathcal{X}_2} f(x_1, x_2) dP(x_1|x_2)$$

is measurable w.r.t. $(\mathcal{B}_1)_\nu$, and ν-integrable, and one has

$$\int_{\mathcal{X}_1 \times \mathcal{X}_2} f(x_1, x_2) d\mu(x_1, x_2) = \int_{\mathcal{X}_1} \int_{\mathcal{X}_2} f(x_1, x_2) dP(x_1|x_2) d\nu(x_1). \tag{A.10}$$

Corollary A.9. *If a parametrization $(\Theta, \Sigma_\Theta) \to \mathcal{P}(\mathcal{X})$, $\theta \mapsto \mathbf{p}_\theta$, defines a transition measure then $\mathbf{p}_\theta$ can be regarded as a regular conditional probability measure $\mu(\cdot|\theta)$ for μ defined by (A.9).*

Recall that $\mathcal{P}(\mathcal{X})$ as a subset of the Banach space $\mathcal{S}(\mathcal{X})$ has a natural topology which is called *strong topology*.

Corollary A.10. *Assume that Θ is a topological space and Σ_Θ is a Borel σ-algebra. If the parametrization mapping $\Theta \to \mathcal{P}(\mathcal{X})$, $\theta \mapsto \mathbf{p}_\theta$, is continuous w.r.t. the strong topology, then $\mathbf{p}_\theta$ can be regarded as a regular conditional probability measure.*

Proof. Since the parametrization is continuous, for any $A \in \Sigma_\mathcal{X}$ the function $\theta \mapsto \mathbf{p}_\theta(A)$ is continuous and bounded, and hence measurable. Hence the parametrization $\Theta \to \mathcal{P}(\mathcal{X})$ defines a transition probability measure and applying Theorem A.8 we obtain Corollary A.10. □

It can be shown that for any measurable space $(\mathcal{X}, \Sigma_\mathcal{X})$ the space $\mathcal{P}(\mathcal{X})$ has a unique smallest σ-algebra $\Sigma_{\mathcal{P}(\mathcal{X})}$ such that the pairing $(\mathcal{P}(\mathcal{X}), \Sigma_\mathcal{X}) \to \mathbb{R}$, $(\mu, A) \mapsto \mu(A)$, defines a probabilistic mapping $Ev : (\mathcal{P}(\mathcal{X}), \Sigma_{\mathcal{P}(\mathcal{X})}) \to (\mathcal{X}, \Sigma_\mathcal{X})$ and hence any statistical model $P \subset (\mathcal{P}(\mathcal{X}), \Sigma_{\mathcal{P}(\mathcal{X})})$ can be regarded as a family of regular conditional probability measures on $(\mathcal{X}, \Sigma_\mathcal{X})$.

Appendix B. Concentration-of-measure inequalities

In probability theory, the concentration of measure is a property of a large number of variables, such as in laws of large numbers. Concentration-of-measure inequalities provide bounds on the probability that a random variable X deviates from its mean, median or other typical value by a given amount. Very roughly speaking, the concentration of measure phenomenon can be stated in the following simple way: "A random variable that depends in a smooth way on many independent random variables (but not too much on any of them) is essentially constant". For the proofs of concentration-of measure inequalities in this Appendix we refer to [Lugosi2009]. I also recommend [Ledoux2001, Shioya2016] for more advanced treatment the concentration-of-measure theory.

B.1. **Markov's inequality.** For any nonnegative random variable $X : (\mathcal{X}, \mu) \to (\mathbb{R}_+, dt)$, and any $t > 0$ we have

$$\mu\{x \in \mathcal{X} |\, X(x) > t\} \leq \frac{\mathbb{E}_\mu X}{t}. \tag{B.1}$$

B.2. **Hoeffding's inequality.** ([Hoeffding1963]) Let $\theta = (\theta_1, \cdots, \theta_n)$ be a sequence of i.i.d. $\mathbb{R}$-valued random variables on $\mathcal{Z}$ and $\mu \in \mathcal{P}(\mathcal{Z})$. Assume that $\mathbb{E}_\mu(\theta_i(z)) = \bar{\theta}$ for all i and $\mu\{z \in \mathcal{Z} |\, [a_i \leq \theta_i(z) \leq b]\} = 1$. Then for any $\varepsilon > 0$ we have

$$\mu^m\{\mathbf{z} \in \mathcal{Z}^m :\, |\frac{1}{m}\sum_{i=1}^m \theta_i(z_i) - \bar{\theta}| > \varepsilon\} \leq 2\exp\big(\frac{-2m\varepsilon^2}{(b-a)^2}\big), \tag{B.2}$$

where $\mathbf{z} = (z_1, \cdots, z_m)$.

B.3. **Bernstein's inequality.** Let θ be a $\mathbb{R}$-valued random variable on a probability space $(\mathcal{Z}, \mu)$ with the mean $\mathbb{E}_\mu(\theta) = \bar{\theta}$ and variance $\sigma^2 = V_\mu(\theta)$. If $|\xi - \mathbb{E}_\mu(\xi)| \leq M$ then for all $\varepsilon > 0$ we have

$$\mu^m\{\mathbf{z} \in \mathcal{Z}^m :\, |\frac{1}{m}\sum_{i=1}^m \theta_i(z_i) - \bar{\theta}| > \varepsilon\} \leq 2\exp\big(\frac{-m\varepsilon^2}{2(\sigma^2 + \frac{1}{3}M\varepsilon)}\big), \tag{B.3}$$

where $\mathbf{z} = (z_1, \cdots, z_m)$.

B.4. **McDiarmid's inequality.** (or Bounded Differences or Hoeffding/Azuma Inequality). Let $X_1, \cdots, X_m \in \mathcal{X}$ are i.i.d. by a probability measure μ. Assume that $f : \mathcal{X}^m \to \mathbb{R}$ satisfies the following property form some $c > 0$. For all $i \in [1, m]$ and for all $x_1, \cdots, x_m, x_i' \in \mathcal{X}$ we have

$$|f(x_1, \cdots, x_m) - f(x_1, \cdots, x_{i-1}, x_i', x_{i+1}, \cdots, x_m)| \leq c.$$

Then we have for all $\delta \in (0, 1)$

$$\mu^m\{S \in \mathcal{X}^m |\, |f(S) - \mathbb{E}_{\mu^m} f(S)| \leq c\sqrt{\frac{\ln(2/\delta)}{m}}\} \geq 1 - \delta. \tag{B.4}$$

References

[Amari2016] S. Amari, Information Geometry and its applications, Springer, 2016.

[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.

[AJLS2018] N. Ay, J. Jost, H. V. Lê and L. Schwachhöfer, Parametrized measure models, Bernoulli vol. 24 Nr 3 (2018), 1692-1725, arXiv:1510.07305.

[Billingsley1999] P. Billingsley, Convergence of Probability measures, 2nd edition, John Wiley ans Sohns, 1999.

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

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

[Bogachev2010] V. I. Bogachev, Differentiable Measures and the Malliavin Calculus, AMS, 2010.

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

[Chentsov1972] N. N. Chentsov, Statistical Decision Rules and Optimal Inference, Translation of mathematical monographs, AMS, Providence, Rhode Island, 1982, translation from Russian original, Nauka, Moscow, 1972.

[CS2001] F. Cucker and S. Smale, On mathematical foundations of learning, Bulletin of AMS, 39(2001), 1-49.

[Fisher1925] R. A. Fisher, Theory of statistical estimation, Proceedings of the Cambridge Philosophical Society, 22(1925), 700-725.

[GBC2016] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT, 2016.

[Ghahramani2013] Z. Ghahramani, Bayesian nonparametrics and the probabilistic approach to modelling. Philosophical Transactions of the Royal Society A 371 (2013), 20110553.

[Ghahramani2015] Z. Ghahramani, Probabilistic machine learning and artificial intelligence, Nature, 521(2015), 452-459.

[Graves2012] A. Graves, Supervised Sequence Labelling with Recurrent Neural Networks, Springer, 2012.

[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.

[Haykin2008] S. Haykin, Neural Networks and Learning Machines, 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. Janssen, A nonparametric Cramér-Rao inequality, Statistics & Probability letters, 64(2003), 347-358.

[Jaynes2003] E. T. Jaynes, Probability Theory The Logic of Sciences, Cambridge University Press, 2003.

[Jost2005] J. Jost, Postmodern Analysis, Springer, 2005.

[JP2003] J. Jacod and P. Protter, Probability Essentials, Springer, 2. edition, 2004.

[Kallenberg2002] O. Kallenberg, Foundations of modern Probability, Springer, 2002.

[Kallenberg2017] O. Kallenberg, Random measures, Springer, 2017.

[Kullback1959] S. Kullback, Information theory and statistics, John Wiley and Sons, 1959.

[Ledoux2001] M. Ledoux, The concentration of measure phenomenon, AMS, 2001.

[Lugosi2009] G. Lugosi, Concentration of measure inequalities - lecture notes, 2009, available at http://www.econ.upf.edu/∼lugosi/anu.pdf.

[LC1998] E. L. Lehmann and G. Casella, Theory of Point Estimation, Springer, 1998.

[MFSS2016] K. Muandet, K. Fukumizu, B. Sriperumbudur and B. Schölkopf, Kernel Mean Embedding of Distributions: A Review and Beyonds, arXiv:1605.09522.

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

[Murphy2012] K. P. Murphy, Machine Learning: A Probabilistic Perspective, MIT Press, 2012.

[Neal1996] R. M. Neal, Bayesian Learning for Neural Networks, Springer, 1996.

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

[Robert2007] C. P. Robert, The Bayesian Choice, From Decision-Theoretic Foundations to Computational Implementation, Springer, 2007.

[Shioya2016] T. Shioya, Metric Measure Geometry Gromov's Theory of Convergence and Concentration of Metrics and Measures, EMS, 2016.

[SSBD2014] S. Shalev-Shwartz and S. Ben-David, Understanding Machine Learning: From Theory to Algorithms, Cambridge University Press, 2014.
[Schervish1997] M. J. Schervish, Theory of Statistics, Springer, 2d corrected printing, 1997.
[Sugiyama2016] M. Sugiyama, Introduction to Statistical Machine Learning, Elsevier, 2016.
[Tsybakov2009] A. B. Tsybakov, Introduction to Nonparametric Estimation, Springer, 2009.
[Valiant1984] L. Valiant, A theory of the learnable, Communications of the ACM, 27, 1984.
[Vapnik1998] V. Vapnik, Statistical learning theory, John Willey and Sons, 1998.
[Vapnik2000] V. Vapnik, The nature of statistical learning theory, Springer, 2000.
[Vapnik2006] V. Vapnik, Estimation of Dependences Based on Empirical Data, Springer, 2006.