MATHEMATICAL FOUNDATIONS OF MACHINE LEARNING

(NMAG 469, FALL TERM 2018-2019)

HÔNG VÂN LÊ *

Contents

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

Date: October 22, 2018.

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

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

1. Learning, machine learning and artificial intelligence

sec:intr

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.

subs:mathlearning

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.

def:learn1

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".

ex:interpolate

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, then f belongs to a N-dimensional linear space, where $N = d + 1$, and to estimate f is the same as to estimate the unknown coefficients $w_0, \cdots, w_d$ of 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* in terms of the value

eq:curvef

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

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.

subs:his

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 scientist.

• 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).

In the early days of AI, statistical and probabilistic methods were employed. Perceptrons which are simple models used in statistics were used

[1]The least-squares method is usually credited to Carl Friedrich Gauss (1809), but it was first published by Adrien-Marie Legendre (1805)

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].

subs:taks

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

[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

subs:tasks

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 to each item. In mathematical language, a category in 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. [3]

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, [4] proposed that all observations come from some probability distribution and the purpose of sciences is to estimate the parameter of these distributions. A particular case of parameter estimation is density estimation problem. Density estimation problem has been proposed

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

[4] 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.

by Ronald Fisher (1980-1962), the father of modern statistics and experiment designs, [5] 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 4.1, 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 $p(x, \xi)$, where ξ is a parameter that belongs to the statistical model A. The problem is to estimate the parameter ξ using i.i.d. data $X_1, \cdots, X_l$ distributed according to this unknown density $p(x, \xi)$.

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.

Dimensionality reduction or manifold learning transforms an initial representation of items in high dimensional space into a space of lower dimension

[5] 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.

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.

subs:types

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.

subs:foundation

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.

prob:q1 **Question 1.3.** *What is the mathematical model of learning?*

To answer Question prob:q1 1.3 we need to specify our definition of learning in a mathematical language which can be used to build instructions for machines.

prob:q2 **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 prob:q2 1.4 contains the sub-question " why can we learn a problem?"

prob:q3 **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 def:learn1 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.

prob:q4 **Question 1.6.** *Is there a mathematical theory underlying intelligence?*

I shall discuss this speculative question in the last lecture.

subs:conclusion

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.

sec:supervised

2. Statistical models and frameworks for supervised learning

Last week we discussed the concept of learning and examine 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.

def:learnvapnik

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.

subs:discr

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.

ex:ML

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 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)* [6] 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 exists, 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 i. i.d. (independently identically distributed) observed labeled pairs. Thus 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".

[6] classically, elements of $\mathcal{X}$ are called *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

• A *hypothesis space* $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ of possible predictors $h : \mathcal{X} \to \mathcal{Y}$.
(In Example ex:ML 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 ex:interpolate 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*

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

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 $(x, y(x))$. Thus a risk function R must depend on μ, so we denote it by R_μ. It is accepted that the risk function R_μ is defined as follows.

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

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

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

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 $L(x, y, f) \in L^1(\mathcal{X} \times \mathcal{Y}, \mu)$ for all $f \in \mathcal{H}$.

• The main question of learning theory is to find necessary and sufficient conditions for the existence of a prediction rule A in (eq:classifierN 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.

rem:super1
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 $(\mathcal{X}\times\mathcal{Y})^n$. 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 $L(x,y,f)\in L^1(\mathcal{X}\times\mathcal{Y},\mu)$ for all $f\in\mathcal{H}$. 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

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

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

$$R_{\mu^n}^{L(d^n)}(f)=\int_{(\mathcal{X}\times\mathcal{Y})^n} d^n([y_1,\cdots,y_n],[f(x_1),\cdots,f(x_n)])d\mu^n$$

eq:neloss
$$=n\int_{\mathcal{X}\times\mathcal{Y}} L(x,y,f)d\mu. \tag{2.5}$$

Thus it suffices to consider only $R_\mu(f)$, 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 $(\mathcal{X}\times\mathcal{Y})^n$ [7]. Now let $d(y,y')=|y-y'|^2$, it is not hard to see that

eq:dirac1
$$R_{\mu_S}^{L(d^n)}(f)=\sum_{i=1}^n |f(x_i)-y_i|^2 \tag{2.6}$$

coincides with the error of estimation in (1.1).

ex:01 **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: $L(x,y,f):=d(y,f(x))=1-\delta^y_{f(x)}$. The corresponding expected 0-1 loss determines the probability of the answer $f(x)$ that does not correlate with x:

eq:error3
$$R^{(0-1)}_{\mu_{\mathcal{X}\times\mathcal{Y}}}(f)=\mu_{\mathcal{X}\times\mathcal{Y}}\{(x,y)\in\mathcal{X}\times\mathcal{Y}|\, f(x)\neq y\}=1-\mu_{\mathcal{X}\times\mathcal{Y}}(\{x,f(x)\}). \tag{2.7}$$

ex:super1 **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

[7] 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

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

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

eq:super1
$$R^{(0-1)}_{\mu_h}(f) = 1 - \mu_{\mathcal{X}}(x|f(x) = h(x)). \tag{2.9}$$

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

subs:gener

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* $\mathcal{H}$

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

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:

eq:cprob1
$$P(A|B) := \frac{P(A \cap B)}{P(B)}. \tag{2.11}$$

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, see e.g. [JP2003, Theorem 3.2, p. 16] for a proof. 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 $\mathbb{E}_{P(\cdot|B)}$, see e.g. [JP2003, Definition 23.1, p. 197]. We can think that 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 $B = h^{-1}(x_0)$, 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 4.2 in Appendix. What is important for our applications in many case is the notion of conditional distribution $P(A|h(x) = x_0)$ which is regular w.r.t. to a measure $\mu_{\mathcal{X}}$, i.e., there exists a density function $f(x|x_0)$ on $\mathcal{X}$ such that

$$P(A|h(x) = x_0) = \int_A f(x|x_0)\mu_{\mathcal{X}}.$$

We may also wish to estimate the joint distribution $\mu := \mu_{\mathcal{X}\times\mathcal{Y}}$ of i.i.d. labeled pairs (x, y). 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.

ex:bayesop **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: $X : \mathcal{X} \times \mathbb{R} \to \mathcal{X}$ and $\mathcal{X} \times \mathcal{Y} \to \mathcal{Y}$.) For a given probability measure μ on $\mathcal{X} \times \mathbb{R}$ we set

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

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

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

eq:qloss (2.12) $$L(x, y, h) := |y - h(x)|^2,$$

eq:reg1 (2.13) $$R^L_\mu(h) = \mathbb{E}_\mu\big(|Y - h(x)|^2\big) = |i_2(Y) - i_1(h)|^2_{L^2(\mathcal{X}\times\mathbb{R},\mu)}.$$

The expected risk R^L_μ 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.

def:gen **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*.

rem:fishergen **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.

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

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.

subs:overf 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 $R^L_\mu(h_S)$ tends to the optimal performance error $R^L_{\mu,\mathcal{H}}$ 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

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

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.

rem:erm **Remark 2.10.** We note that

eq:erm1
$$\hat{R}_S^{L(d)}(h) = \frac{1}{n} R_{\mu_S}^{L(d^n)}(h) \tag{2.16}$$

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 $R_{\mu,\mathcal{H}}^L$.

ex:trerror **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

eq:erisk1
$$\hat{R}_S^{0-1}(h) := \frac{|i \in [n] : \ h(x_i) \neq y_i|}{n} \tag{2.17}$$

for a training data $S = \{(x_1, y_1), \cdots, (x_n, y_n)\}$ and a function $h : \mathcal{X} \to \mathcal{Y}$.

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

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

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

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

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.

ex:ERM **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 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^L_μ 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 $R^L_\mu(h_S)$ converges to the optimal performance error $R^L_{\mu,\mathcal{H}}$. 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

subs:unsuper

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.

subs:density

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

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. [8] 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

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

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

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

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

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

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.

rem:KL
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

[8] 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.

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 $R_u(\theta)$ 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 μ_u. The Kullback-Leibler divergence, unlike the expected log-likelihood function, can be defined on the space $\mathcal{P}(\mathcal{X})$ of all probability measures on $\mathcal{X}$. 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 an 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 peoples considers 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 $\mathcal{X} = \mathbb{R}^n$ 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

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

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*

eq:Rosenblatt

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

which can be rewritten in the following form

eq:Rosenblatt2

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

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

eq:PR

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

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 *bandwith* 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 $S_n \in \mathbb{R}^n$:

eq:MSEPR

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

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].

rem:compareunsuper

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.

subs:clustering

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.

ex:kmean

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 mu_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

eq:kmean

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

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].

SSBD2014

rem:pcluster

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.

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

$$W(S_n) := \{W(x_i)\} \subset \mathbb{R}^m.$$

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

ex:pca **Example 3.5.** A popular linear method for dimension reduction is called *Principal Component Analysis* (PCA). 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 $R_{S_n} : Hom(\mathbb{R}^d, \mathbb{R}^n) \times Hom(\mathbb{R}^n, \mathbb{R}^d) \to \mathbb{R}$

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

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

exe:pca **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 is an orthogonal embedding and $W \circ U = Id_{|\mathbb{R}^n}$.

Let $Hom_g(\mathbb{R}^d, \mathbb{R}^n)$ denotes the set of all orthogonal 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 $\mathcal{F}$.

ex:pca3 **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* $\psi : \mathbb{R}^d \to \mathbb{R}^k$, and a *decoder function* $\varphi : \mathbb{R}^k \to \mathbb{R}^d$. The goal of the learning process is to P find a pair of functions (ψ, φ) such that *the reconstruction error*

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

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.

rem:paec

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 [TP1999].

SSBD2014
GBC2016
TP1999

subs:reinforce

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].

MRT2012

subs:conclusion2

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. Appendix 1: Some basic notions in probability theory

sec:app

Basis objects in probability theory (and mathematical statistics) are measurable spaces (X, Σ), where Σ is a σ-algebra of subsets of a space X. A countably additive measure μ on Σ is called *a probability measure* if $\mu \geq 0$ and $\mu(X) = 1$.

For this Appendix I use [Bogachev2007] as my main reference on measure theory, the book [JP2003] for a clear and short exposition of probability theory, and [Kallenberg2002] for a modern treatment of probability theory. I also use [Borovkov1998] for its clear exposition of probability theory and applications in statistics.

subs:RN

4.1. **Dominating measures and the Radon-Nikodym theorem.** Let μ and ν be countably additive measures on a measurable space (X, Σ)
(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: $\nu \perp \mu$.

thm:radonnikodym

Theorem 4.1. *(cf.* [Bogachev2007, Theorem 3.2.2, p. 178]*) Let μ and ν be two finite measures on a measurable space (S, Σ). The measure ν is dominated by the measure μ precisely when there exists a μ-integrable function f such that ν is given by*

eq:rn

$$\nu(A) = \int_A f \, d\mu \tag{4.1}$$

for each $A \in \Sigma$.

We denote ν by $f\mu$ for μ, ν, f satisfying the equation (4.1). 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 $d\nu/d\mu$.

subs:conditional

4.2. **Conditional expectation, conditional probability (measure) and joint distribution.** The important notion of conditional probability in the case of uncountable measurable space $(\mathcal{X}, \Sigma_{\mathcal{X}})$ is defined via the notion of *conditional expectation*. There are two reasons for this approach. Firstly conditional probability can be obtained from conditional expectation: $P(A|B) = \mathbb{E}_{P(\cdot|B)} 1_A$. Secondly, in general case, we can treat conditional expectation as a function satisfying certain axioms ([Bogachev2007, Chapter 10, p. 339].

4.2.1. *Conditional expectation.*

def:conde

Definition 4.2. Let $f \in L^1(\mu)$. A *conditional expectation* of f with respect to the σ-algebra $\mathcal{B}$ and the measure μ is a *$\mathcal{B}$-measurable μ-integrable* function

$\mathbb{E}^{\mathcal{B}}_{\mu} f$ such that

eq:cond1 (4.2) $$\int_{\Omega} gf\, d\mu = \int_{\Omega} g\mathbb{E}^{\mathcal{B}}_{\mu} f\, d\mu$$

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

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

rem:condt **Remark 4.3.** 1) The defining equality (4.2) is equivalent to the following relationship obtained by the substitution $g = 1_{\mathcal{B}}$:

eq:condt2 (4.3) $$\int_{\Omega} f\, d\mu = \int_{B} \mathbb{E}^{\mathcal{B}}_{\mu} f\, d\mu \, \forall B \in \mathcal{B}.$$

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

2) In the case where only one measure μ is given, for simplification of notation and terminology one uses:

$$\mathbb{E}^{\mathcal{B}} f := \mathbb{E}^{\mathcal{B}}_{\mu} f.$$

In the probabilistic literature one uses the notation

$$\mathbb{E}(f|\mathcal{B}) := \mathbb{E}^{\mathcal{B}}_{\mu} f.$$

3) If Y is an integrable function on a probability space (Ω, μ) and $\mathcal{B} = \eta^{-1}(\Sigma')$ is generated by a measurable function (or mapping) $\eta : (\Omega, \Sigma) \to (\Omega', \Sigma')$ [9] then one uses the notation

$$\mathbb{E}(Y|\eta) := \mathbb{E}^{\sigma(\eta)} Y,$$

where $\sigma(\eta)$ is the sub-σ-algebra generated by η.

thm:exicondt **Theorem 4.4.** ([Bogachev2007, Theorem 10.1.5, p. 341]. *Suppose that μ is a probability measure. To every function $f \in L^1(\mu)$, one can associate a $\mathcal{B}$-measurable function $\mathbb{E}^{\mathcal{B}} f$ such that*
(1) $\mathbb{E}^{\mathcal{B}} f$ is a conditional expectation of f with respect to $\mathcal{B}$;
(2) $\mathbb{E}^{\mathcal{B}} f = f$ μ-a.e. for every $\mathcal{B}$-measurable μ-integrable function f ;
(3) $\mathbb{E}^{\mathcal{B}} f \geq 0$ μ-a.e. if $f \geq 0$ μ-a.e.;
(4) if a sequence of μ-integrable functions f_n converges monotonically decreasing or increasing to a μ-integrable function f, then $\mathbb{E}^{\mathcal{B}} f_n \to \mathbb{E}^{\mathcal{B}} f$ μ-a.e.;
(5) For every $p \in [1, +\infty]$, the mapping $\mathbb{E}^{\mathcal{B}}$ defines a continuous linear operator with norm 1 on the space $L^p(\mu)$. In addition, $\mathbb{E}^{\mathcal{B}}$ is the orthogonal projection of $L^2(\mu)$ to the closed linear subspace generated by $\mathcal{B}$-measurable functions.

rem:conde **Remark 4.5.** In [JP2003, Definition 23.5, p. 200] the author choose the last part of Property (5) of Theorem 4.4, namely the orthogonal projection to $L^2(\mu)$ as the definition of conditional expectation, using some extension arguments [JP2003, Lemma 23.1, Theorem 23.4, p. 201, 202].

[9] One can take η to be the identity mapping, and $\Sigma' = \mathcal{B}$.

ex:cmeasure **Example 4.6.** In the case $\mathcal{B}$ is generated by a mapping η to $\mathbb{R}^n$ the conditional expectation of an integrable random variable ξ can be computed as follows (see e.g. [Bogachev2007, p. 345])

$$\mathbb{E}(\xi|\eta) : \mathbb{E}^{\sigma(\eta)}(\xi) = f(\eta)$$

where f is a function on $\mathbb{R}^n$ that is defined by

$$f(x) = \lim_{r\to 0} \frac{1}{P(\eta \in B(x,r))} \int_{\{\eta\in B(x,r)\}} \xi dP.$$

4.2.2. *Conditional measure and conditional probability. The conditional measure* (or *conditional probability* in the case of probability measure) *of* $A \in \Sigma$ *w.r.t.* $\mathcal{B}$, is defined as follows

eq:cprob
$$\mu(A|\mathcal{B}) := \mathbb{E}_\mu(1_A|\mathcal{B}). \tag{4.4}$$

In probabilistic literature one omits μ in (4.4) and writes instead

$$P(A|\mathcal{B}) := \mu(A|\mathcal{B}).$$

If $\mathcal{B} = \xi^{-1}(\Sigma')$ where $\xi : (\Omega, \Sigma) \to (\Omega', \Sigma')$ is a measurable map, one uses the notation

$$P(A|\xi) := \mu(A|\xi) := \mu(A|\mathcal{B}).$$

rem:conditioning **Remark 4.7.** (1) In general one can say that for every $A \in \Sigma$ there exists a function $\zeta_A : \Omega' \to \mathbb{R}$ such that

$$\mu(A|\xi)(x) = \zeta_A(\xi(x))$$

for all $x \in \Omega$. Then one sets

eq:conditioning
$$\mu^{\mathcal{B}}(A|\xi = x) := \zeta_A(x). \tag{4.5}$$

The RHS of (4.5) is called *the measure of* A *under conditioning* $\xi = x$. (2) Let $\pi : (X, \Sigma) \to (Y, \Sigma')$ be a measurable map. For any $A \in \Sigma$ and $B \in \mathcal{B} = \pi^{-1}(\Sigma')$ we have

eq:conditioning2
$$\mu(A \cap B) = \int_B \mu^{\mathcal{B}}(A|x)\mu(dx). \tag{4.6}$$

Since $\mu^{\mathcal{B}}(A|x)$ is a $\mathcal{B}$-measurable function on X for each A, by [Bogachev2007, Theorem 2.12.3, p. 144] on $\mathcal{B}$-measurable functions, we have for any $E \in \Sigma'$

eq:conditioning3
$$\mu(A \cap \pi^{-1}(E)) = \int_E \mu^y(A)\, \pi_*(\mu)(dy) \tag{4.7}$$

where $\mu^y(A) := \mu^{\mathcal{B}}(A, x)$ is a Σ'-measurable function.

4.2.3. *Regular conditional measure.*

def:regcp **Definition 4.8.** ([Bogachev2007, Definition 10.4.1, p. 357]) Suppose we are given a sub-σ-algebra $\mathcal{B} \subset \Sigma$. A function

$$\mu^{\mathcal{B}}(\cdot, \cdot) : \Sigma \times \Omega \to \mathbb{R}$$

is called *a regular conditional measure* on Σ w.r.t. $\mathcal{B}$ if
(1) for every $x \in \Omega$ the function $A \mapsto \mu^{\mathcal{B}}(A, x)$ is a *measure on* Σ,
(2) for every $A \in \Sigma$ the function $x \mapsto \mu^{\mathcal{B}}(A, x)$ is $\mathcal{B}$-*measurable and* μ-*integrable*,
(3) For all $A \in \Sigma, B \in \mathcal{B}$ the following formula for joint probability holds, cf (4.6)

eq:jointpr
$$\mu(A \cap B) = \int_B \mu^{\mathcal{B}}(A, x)|\mu|(dx). \tag{4.8}$$

rem:regcond **Remark 4.9.** (cf. [Bogachev2007, Definition 10.4.2, p. 358]) In the case $\mathcal{B}$ is generated by a measurable map $\pi : (X, \Sigma) \to (Y, \Sigma')$ as in Remark 4.7(2), it is more convenient to parameterize regular conditional measures by points of the space Y. We say that a system of regular conditional measures μ^y, $y \in Y$, is a function on $\Sigma \times Y$ such that for every fixed y it is a measure on Σ, for every fixed $A \in \Sigma$ is measurable w.r.t. Σ' and $|\mu| \circ \pi^{-1}$-integrable, and for all $A \in \Sigma$ and $E \in \Sigma'$ we have

eq:conditioning4
$$\mu(A \cap \pi^{-1}(E)) = \int_E \mu^y(A)\, \pi_*(|\mu|)(dy). \tag{4.9}$$

4.2.4. *Regular conditional probability and joint distribution.*

def:condd **Definition 4.10.** ([Borovkov1998, Definition 2, §20, p. 107]) Assume that a (regular) [10] conditional probability $P(x \in B|y)$ for each $y \in Y$ is absolutely continuous w.r.t. some measure μ in X, i.e.,

eq:condd
$$P(B|\eta = y) = \int_B f(x|y)d\mu(x). \tag{4.10}$$

Then the density $f(x|y)$ is called *the conditional density of* ξ *subject to the condition that* $\eta = y$.

thm:condd **Theorem 4.11.** *([Borovkov1998, Theorem 2, §20, p. 108]) If the joint distribution of* ξ *and* η *in* $X \times Y$ *has density function* $f(x, y)$ *w.r.t. the product of measures* $\mu \in \mathcal{P}(X)$ *and* $\lambda \in \mathcal{P}(Y)$ *then the function*

$$f(x|y) := \frac{f(x, y)}{q(y)} \text{ where } q(y) = \int f(x, y)\mu(dx)$$

is the conditional density of ξ *given* $\eta = y$ *and the function* $q(y)$ *is the density of* η *w.r.t. the measure* λ.

[10] Borovkov did not use the terminology "regular conditional probability", instead he use the terminology "conditional distribution: and "conditional density ".

References

Amari2016 [Amari2016] S. Amari, Information Geometry, Springer, 2016.

AJLS2017 [AJLS2017] N. Ay, J. Jost, H. V. Lê and L. Schwachhöfer, Information Geometry, Springer, 2017.

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

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

CS2001 [CS2001] F. Cucker and S. Smale, On mathematical foundations of learning, Bulletin of AMS, 39(2001), 1-49.

GBC2016 [GBC2016] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT, 2016.

Ghahramani2013 [Ghahramani2013] Z. Ghahramani, Bayesian nonparametrics and the probabilistic approach to modelling. Philosophical Transactions of the Royal Society A 371 (2013), 20110553.

JLS2017 [JLS2017] J. Jost, H. V. Lê and L. Schwachhöfer, The Cramér-Rao inequality on singular statistical models, arXiv:1703.09403.

JP2003 [JP2003] J. Jacod and P. Protter, Probability Essentials, Springer, 2. edition, 2004.

Kallenberg2002 [Kallenberg2002] O. Kallenberg, Foundations of modern Probability, Springer, 2002.

Kullback1959 [Kullback1959] S. Kullback, Information theory and statistics, John Wiley and Sons, 1959.

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

Murphy2012 [Murphy2012] K. P. Murphy, Machine Learning: A Probabilistic Perspective, MIT Press, 2012.

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

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

Sugiyama2016 [Sugiyama2016] M. Sugiyama, Introduction to Statistical Machine Learning, Elsevier, 2016.

TP1999 [TP1999] M. E. Tipping, C. Bishop, Probabilistic Principal Component Analysis, Journal of the Royal Statistical Society, Series B, 21/3(1999), 611-622.

Tsybakov2009 [Tsybakov2009] A. B. Tsybakov, Introduction to Nonparametric Estimation, Springer, 2009.

Vapnik1998 [Vapnik1998] V. Vapnik, Statistical learning theory, John Willey and Sons, 1998.

Vapnik2000 [Vapnik2000] V. Vapnik, The nature of statistical learning theory, Springer, 2000.

Vapnik2006 [Vapnik2006] V. Vapnik, Estimation of Dependences Based on Empirical Data, Springer, 2006.