Mathematical foundations of machine learning

Hông Vân Lê
Institute of Mathematics, CAS

VIASM, Hà nôi, August 21-24, 2018

Lecture 1: Learning, machine learning and artificial intelligent.

1. What are learning, deductive learning and machine learning.

2. History of machine learning and artificial intelligence.

3. Current tasks and main type of machine learning.

4. Basic questions in mathematical foundation of machine learning.

1. What are learning, deductive learning and machine learning?

(a) Small children learn to speak by observing, repeating and mimicking adults' phrases. Their way of learning is inductive learning.

(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.

(c) Experimental physicists design experiments and observe the outcomes of the experiments to validate or dispute a conjecture on the nature of the observables. This type of learning is inductive learning.

• A learning is a process of gaining new knowledge by examination of empirical data of the observables. A learning is successful if the knowledge can be tested in examination of new data. A machine learning is an automatized process of learning.

Definition (Russell and Norvig - Artificial Intelligence - A modern Approach) An agent (human, robot, machine) is learning if it improves its performance on future tasks after making observations about the world.

Mathematical definition (Vapnik) Learning is a problem of function estimation on the basis of empirical data.

In mathematical language experience is empirical data and knowledge is function estimation.

• A classical example of learning is that of learning a physical law by curve fitting to data. Assuming the law, an unknown function $f : \mathbf{R} \to \mathbf{R}$, has a specific form and that the space of all functions having this form can be parameterized by N real numbers. For instance, if f is assumed to be a polynomial of degree d, then $N = d + 1$ and the parameters are the coefficients $w_0, \cdots, w_d$ of f. In this case, finding the best fit by the least squares method estimates the unknown f from a set of pairs $(x_1, y_1), \cdots, (x_m, y_m)$.

One computes the vector of coefficients w such that the value

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

is minimized where, typically $m > N$. If the measurements generating this set were exact, then $f(x_i)$ would be equal to y_i. But in general one expects the values y_i to be affected by noise. The least square technique, going back to Gauss and Legendre, which is computational efficient and relies on numerical linear algebra.

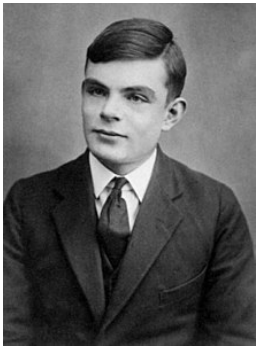
The least-squares method is usually credited to Carl Friedrich Gauss (1809), but it was first published by Adrien-Marie Legendre (1805).

2 History of machine learning and artificial intelligence

• 1945 Vannevar Bush proposed in "As We May Think" published in "The Atlantic", a system which amplifies peoples own knowledge and understanding. Bush's memex was based on what was thought, at the time, to be advanced technology of the future: ultra high resolution microfilm reels, coupled to multiple screen viewers and cameras, by electromechanical controls. Through this machine, Bush hoped to transform an information explosion into a knowledge explosion.

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

• 1950 Alan Turing asked Can machines think? in "Computing Machine and Intelligence" and proposed the famous Turing test. The Turing 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.



Alan Turing (1912-1950)

• 1956 John McCarthy coined the term artificial intelligence.

• 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).

However, an increasing emphasis on the logical, knowledge-based approach caused a rift between AI and machine learning. Probabilistic systems were plagued by theoretical and practical problems of data acquisition and representation, which were unsolvable because of small capacity of hardware memory and slow speed of computers that time. By 1980, expert systems had come to dominate AI, and statistics was out of favor. Expert system uses the idea that "intelligent systems derive their power from the knowledge they possess rather than from the specific formalisms and inference schemes ".

Work on symbolic based learning did continue within AI, leading to inductive logic programming. Neural networks research had been abandoned by AI and computer science around the same time. Their main success came in the mid-1980s with the reinvention of a algorithm in neural network which was able thanks to increasing speed of computers and increasing hardware memory.

Machine learning, reorganized as a separate field, started to flourish in the 1990s.

• AI $\rightsquigarrow$ ML: tackling solvable problems of a practical nature.

• Methods and models borrowed from statistics and probability theory. Laplacian determinism $\rightsquigarrow$ probabilistic modeling of random observables - new paradigm shift in sciences.

• 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 ML.

- 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 problem 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.

3. Current tasks and types of ML.

Main tasks

• Classification task assigns a category to each item. For example, document classification may assign items with categories such as politics, email spam, sports, or weather, image classification may assign items with landscape, portrait, or animal. 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 mathematical example of learning (p. 6) 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. In real life and for machine learning we need to model the noise using probability theory. Therefore machine learning is based on statistical learning theory, which we shall learn in tomorrow lecture.

• Regression task predicts a real value for each item. Examples of regression tasks include prediction of stock values or variations of economic variables. In this problem, the penalty for an incorrect prediction depends on the magnitude of the distance between the true and predicted values, in contrast with the classification problem, where there is typically no notion of closeness between various categories. A regression task is a (construction of a) function, on the set of items that takes value in $\mathbb{R}$, taking into account a "noise" from incorrect measurement.

The term regression was coined by Francis Galton in the 19 century to describe a biological phenomenon 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 to a more general statistical context. Galton method of investigation is non-standard at that time: first he collected the data, then he guessed the relationship model of the events.

• Density estimation task finds the distribution of inputs in some distribution space. Karl Pearson (1857-1936) proposed that all observations come from some probability distribution and the purpose of sciences is to estimate the parameter of these distributions. Density estimation problem has been proposed by Ronald Fisher (1980-1962) as a key element of his simplification of statistical theory, namely he assumed the existence of a density function $p(\xi)$ that defines the randomness (noise) of a problem of interest.

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, 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 $\xi \in A$. The problem is to estimate the parameter ξ using i.i.d. data $X_1, \cdots, X_l$ distributed according to this unknown density $p(x, \xi)$.

Karl Pearson (1857-1936) Ronald Fisher (1890-1962)

• 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. 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 while preserving some properties of the initial representation. A common example involves pre-processing digital images in computer vision tasks. We can regard clustering as dimension reduction too.

Main types of ML

The type of ML is defined by the type of interaction between the learner and the environment: the type of training data, i.e., the data available to the learner before making decision and prediction; and the type of the test data that are used to evaluate and apply the learning algorithm.

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

• 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. In computer sciences language, a known instance is an input and and its feature is the output of a program that predicts the label for unseen instances. Examples of sets of labeled data are emails that are labeled "spam" or "no spam" and medical histories that are labeled with the occurrence or absence of a certain disease.

• Most of classification and regression problems of machine learning belong to supervised learning.

• In unsupervised learning there is no additional label attached to the data and the task is to describe structure of data. Since the examples (the avalable 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.

At the current time, 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.

Basic questions in mathematical foundations of ML

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 "good" prediction on unseen data, which improves when we have more data.

Mathematical foundations of machine learning aim to answer the following questions
How and why do machine learn successfully?

1. What is the mathematical model of learning?

To answer Question 1 we need to specify our definition of learning in a mathematical language which can be used to build instructions for machines.

2. How to quantify the difficulty/complexity of a learning problem?

The difficulty of a problem shall be defined in terms of complexity: time complexity to solve a problem, resource complexity of a problem to have enough data/space/energy to solve a problem. If the complexity of a problem is very large then we cannot not learn it. So Question 2 contains the sub-question why can we learn a problem?

3. How to choose a learning algorithm?

Clearly we want to have the best learning algorithm, once we know a model of a machine learning which contains all possible learning algorithms. To answer Question 3 we need to measure success of a learning algorithm/a learning machine, e.g. we quantify the success in the rate/number of mistakes, which can be linked to the complexity of a learning problem. Thus Question 3 is related to the first and second Question.

4. Is there a mathematical theory underlying intelligence?

I shall discuss the last Question in the last lecture.

Future of machine learning and AI

Nowadays many machine learning systems can automate things that humans do well. Examples include image recognition, speech recognition, and email spam classification which are mostly supervised learning.

We are now surpassing human-level performance on more and more of the tasks where we can get easily labeled training data. Unsupervised learning currently is mostly experimental, since we cannot quantify the notion of success for unsupervised learning, e.g. for clustering. For example, it is not clear what is the "correct" clustering for given data or how to evaluate a proposed clustering. If we can quantify the "success" in an unsupervised learning problem then we can make a mathematical model for this problem.

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 important for building general purpose artificial intelligence, also called AGI, or UAI.

Recommended literature for the first two lectures

• 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.

• Z. Ghahramani, Probabilistic machine learning and artificial intelligence, Nature, 521(2015), 452-459.

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

• V. Vapnik, The nature of statistical learning theory, Springer, 2000.

THANK YOU FOR YOUR ATTENTION!

Mathematical foundations of machine learning

Lecture 1: Learning, machine learning and artificial intelligent

Machine learning is a process of gaining new knowledge by generalization from empirical data, whose performance improves with "experience ".

Lecture 2: Statistical models and frameworks for machine 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, anticipate outcomes of their actions, and update their ability to make predictions in light of new data

The model for machine learning depends on type of statistical learning.

1. Statistical model and framework for supervised learning.

2. Statistical model and framework for unsupervised learning.

3. Statistical model and framework for reinforcement learning.

1. Statistical model and framework for supervised learning The model of supervised learning is based on Vapnik's statistical learning theory and Cucker-Smale's mathematical learning theory.

Definition (Vladimir Vapnik) Learning is a problem of function estimation on the basis of empirical data.

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), his/her computer sciences skill, rated by the mark (from 1 to 20), 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.

This type of machine learning is supervised learning.

Discriminative model of supervised learning

• A domain set $\mathcal{X}$ (also called an input space) whose elements $x \in \mathcal{X}$, called instance/input, is 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 $\mu_{\mathcal{X}}(A)$. According to Kolmogorov's probability axiom, we need to supply $\mathcal{X}$ with a σ-algebra $\Sigma_{\mathcal{X}}$ and we can compute only $\mu(A)$ for $A \in \Sigma_{\mathcal{X}}$.

• An output space $\mathcal{Y}$, also called a label set, whose elements are possible features (also called labels) y for each input $x \in \mathcal{X}$. (We may wish to estimate the probability that y is a feature of x, which is expressed in an unknown conditional measure $P(y \in B|x)$ - the probability that $y \in B \subset \Sigma_{\mathcal{Y}}$ is a feature of $x \in \mathcal{X}$).

We assume that $P(y \in B|x)$ is a regular conditional probability, which is more over dominated, i.e., there is a measure $\mu_{\mathcal{Y}}$ on $\mathcal{Y}$ such that

$$P(y \in B|x = x_0) = \int_B f^{\mu_{\mathcal{Y}}}(y|x = x_0)\, \mu_{\mathcal{Y}}$$

for a density function $f^{\mu_{\mathcal{Y}}}(y|x = x_0)$ of the conditional probability $P(y \in B|x = x_0)$ w.r.t. a probability measure $\mu_{\mathcal{Y}}$ on $\mathcal{Y}$.

• 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 pairs of instances and their label. S is also called a training data, which are given by a "supervisor".

• A subset $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ of possible predictors $h : \mathcal{X} \to \mathcal{Y}$.

• The aim of a learning machine is to find the best prediction rule - a map $h_S \in \mathcal{H}$, given a training data S.

- The learner needs to find an algorithm

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

Remark In a discriminative model of supervised learning we estimate the desired function from $\mathcal{H}$.

(In a generative model of supervised learning we estimate the joint distribution of instances and their feature, or conditional distribution of a feature, given an instance).

Let us consider the toy example of applicants of a ML firm. The domain set $\mathcal{X} = [1, 20] \times [1, 20] \times [1, 5]$. The label set $\mathcal{Y} = [1, 10]$. The set $\mathcal{H}$ of possible hypotheses we want to consider is a subset of the set of all functions from $[1, 20] \times [1, 20] \times [1, 20]$ with values in $[1, 10]$, e.g., $\mathcal{H}$ is the set of all functions with values in 5 or 6. In the discriminant model of learning we estimate potential of applicants as satisfactory (with mark 6) or unsatisfactory (with mark 5), but not the conditional probability of the potential given the marks of applicants (which we learn in a generative model).

Different probabilistic modelings of random observables.

(i) The most general way is to model a random correlation of $x \in \mathcal{X}$ with $y \in \mathcal{Y}$ via a measure $\mu_{\mathcal{X}\times\mathcal{Y}}$, also called the joint distribution, on $\mathcal{X} \times \mathcal{Y}$. The probability that $(x, y) \in D \subset \mathcal{X} \times \mathcal{Y}$ is in correlation is equal to $\mu_{\mathcal{X}\times\mathcal{Y}}(D)$. In most cases (if $\mathcal{Y}$ is a finite set, a separable metrizable topological space endowed with the Borel σ-algebra), the measure on the slice $\{(y, x)|x = x_0\}$ is the conditional probability of y conditioning by x_0.

Example. Let $\mathcal{X}$ be a finite set of n elements and $\mathcal{Y}$ be a finite set of m elements.

1) Assume that the correlation between x and y is defined via the counting measure μ^{count} on $\mathcal{X} \times \mathcal{Y}$. Then the conditional probability measure $P(y \in B | x = x_0)$ is equal to the counting measure $\mu_{\mathcal{Y}}(B)$ on the slice $x = x_0$ which does not depend on x_0, moreover the conditional density $f^{\mu_{\mathcal{Y}}}(y|x) = 1/m$ does not depend on y either.

2) $P^{\mu_{count}}(A|B) = \frac{\mu_{count}(A \cap B)}{\mu_{count}(B)}$.

Recommende book for a short summary and clear explanation of Probability Theory: J. Jacod and P. Protter, Probability Essentials, Springer, 2.edition, 2004.

(ii) Another way to model a random correlation between an instance $x \in \mathcal{X}$ and its possible feature $y \in \mathcal{Y}$ is to regard y as a value of a probabilistic (or random) map K from $\mathcal{X}$ to $\mathcal{Y}$. The probability that $K(x) \in B$ is defined by a conditional probability $P^{\mu_{\mathcal{Y}}}(y \in B|x)$ for some measure $\mu_{\mathcal{Y}}$ on $\mathcal{Y}$. If K is a probabilistic map from $\mathcal{X}$ to $\mathcal{Y}$ then its graph $Gr_K : \mathcal{X} \to \mathcal{X} \times \mathcal{Y}$, $x \mapsto (x, K(x))$, is a probabilistic mapping from $\mathcal{X}$ to the product space $\mathcal{X} \times \mathcal{Y}$ with the probability that $K(x) \in A \times B$ is equal to

$$(*) \quad \int_A P(y \in B|x)\mu_{\mathcal{X}} = \int_A \int_B p^{\mu_{\mathcal{Y}}}(y|x)\mu_{\mathcal{Y}}\mu_{\mathcal{X}}.$$

• Every probabilistic map $K : \mathcal{X} \to \mathcal{Y}$ defines a random correlation on the $\mathcal{X} \times \mathcal{Y}$, i.e., K defines a probabilistic measure on $\mathcal{X} \times \mathcal{Y}$. Given a measure $\mu_{\mathcal{X}}$ on $\mathcal{X}$ and probabilistic map $K : \mathcal{X} \to \mathcal{Y}$ we define the probabilistic measure $(Gr_K)_*(\mu_{\mathcal{X}})$, the push-forwarding of $\mu_{\mathcal{X}}$ via the map Gr_K, on $\mathcal{X} \times \mathcal{Y}$ as follows

$$(Gr_K)_*\mu_{\mathcal{X}}(D) := \int_{\mathcal{X}} P^{\mu_{\mathcal{Y}}}((x,y) \in D|x)\mu_{\mathcal{X}}$$

$$\stackrel{(*)}{=} \int_D p^{\mu_{\mathcal{Y}}}(y|x)\mu_{\mathcal{Y}}\mu_{\mathcal{X}} \text{ for } D \subset \mathcal{X} \times \mathcal{Y}.$$

• Assuming $K_*\mu_{\mathcal{X}} = f(x,y)\,dxdy$ and $\mu_{\mathcal{X}} = f(x)\,dx$, we deduce from the above formula the following relation

$$f(x,y) = f(x)f(y|x).$$

The probability of the joint distribution x and y is the product of the probability of x and the probability of y conditioning on x.

• Any deterministic (usual) $K : \mathcal{X} \to \mathcal{Y}$ can be regarded as a probabilistic mapping with

$$P^{\mu_K}(y \in B|x) = \delta_{K(x)}(B) = 1_B(K(x)).$$

(iii) In the current ML and SL literature, following Fisher suggestion, we consider discriminative models of supervised learning where

$$p(y|x) = f(x) + \varepsilon,$$

and the random error ε (a measurable function on $\mathcal{X}$) has $\mathbb{E}_\mu(\varepsilon) = 0$ and is independent of x.

• A solution (a desired predictor) for the supervised learning problem must have minimal (expected) error/loss/risk $R_{\mu_{\mathcal{X}\times\mathcal{Y}}} : \mathcal{H} \to \mathbf{R}$.

• Let $L : \mathcal{X} \times \mathcal{Y} \times \mathcal{H} \to \mathbf{R}$ be an instantaneous loss function $L(x, y, f) := d(y, f(x))$ where $d : \mathcal{Y} \times \mathcal{Y} \to \mathbf{R}$ is a quasi-distance function on $\mathcal{Y}$, i.e., $d(y, y') \geq 0$ and $d(y, y') = 0$ iff $y = y'$. Then

$$R_{\mu_{\mathcal{X}\times\mathcal{Y}}}(f) := \int_{\mathcal{X}\times\mathcal{Y}} L(x, y, f)\mu_{\mathcal{X}\times\mathcal{Y}}$$

where $\mu_{\mathcal{X}\times\mathcal{Y}}$ is the unknown joint distribution of (x, y).

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

$$R_{\mu_{\mathcal{X}\times\mathcal{Y}}}(f) = \mu_{\mathcal{X}\times\mathcal{Y}}\{(x, y) \in \mathcal{X} \times \mathcal{Y} |\, f(x) \neq y\}$$

$$= 1 - \mu_{\mathcal{X}\times\mathcal{Y}}(\{x, f(x)\}).$$

• We don't know μ and therefore we don't know R_μ. Therefore we have to find a hypothesis h_S, given a sequence S of empirical data generated by the probability distribution μ that defines the probability of the correlation of labeled pairs in $\mathcal{X} \times \mathcal{Y}$, such that the expected error of h_S is smallest possible. This motivates the notion of the empirical risk discussed below.

For a loss function $L : \mathcal{X} \times \mathcal{Y} \times \mathcal{H} \to \mathbf{R}$, $L(x, y, h) = d(y, h(x))$,
$S = \{(x_1, y_1) \cdots, (x_n, y_n)\} \in (\mathcal{X} \times \mathcal{Y})^n$
we define the empirical risk of h as follows

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

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 very small expected risk, which goes to zero as the size of the empirical data increases. This is the empirical risk minimization principle.

We shall show an counter-example of the ERM principle.

• The empirical risk corresponding to 0-1-loss function $L : \mathcal{X} \times \mathcal{Y} \times \mathcal{Y}^{\mathcal{X}} \to \{0, 1\}$ is called the training error. It is defined as follows

$$\widehat{R}_S(h) := \frac{|i \in [n] : \ h(x_i) \neq y_i|}{n}$$

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

Given a training data $S = \{(x_i, y_i = f(x_i))\}$ for a map $f : \mathcal{X} \to \mathcal{Y}$ we shall find a predictor h_S with vanishing training error $\hat{R}_S(h_S)$, nevertheless its expected 0-1 risk is equal to ε for any given $\varepsilon \in (0, 1)$. Let

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

• Clearly $\hat{R}_S(h_S) = 0$, since $h_S(x) = 0$ except finite (at most n) points x in $\mathcal{X}$.

• Let $\mathcal{X}$ be the unit cube I^k in $\mathbf{R}^k$, $k \geq 1$, and $\mathcal{Y} = \mathbf{Z}_2$. Let μ_0 be the Lebesgue measure on I^k. For $\varepsilon \in (0,1)$, we shall find a map $f : \mathcal{X} \to \mathcal{Y}$, such that the expected 0-1 risk of the function h_S defined in (**) is equal

$$R_{(Gr_f)_*\mu_0}(h_S) = \varepsilon.$$

Let $\mathcal{X} = A_1 \dot{\cup} A_2$ s.t. $\mu_0(A_2) = \varepsilon$.
Let $f : \mathcal{X} \to \mathbf{Z}_2$ be the indicator function 1_{A_1}.
Then
(♠)

$$R_{(Gr_f)_*\mu_0}(h_S) = (\{x \in \mathcal{X} | \, h_S(x) \neq 1_{A_1}(x)\}).$$

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

$$R_{(Gr_f)_*\mu_0}(h_S) = \mu_0(A_2) = \varepsilon.$$

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

The idea for the construction of the counter-example of ERM principle is simple: the expected 0-1 risk of a function f is the average of the 0-1 loss of f, which is zero if f is zero everywhere except a null set. On the other hand, any empirical sequence $\{x_1, \cdots, x_n\}$ is a null set in the unit cube $(I^k)^n$ where $k \geq 1$, and the zero set of the indicator function 1_{A_1} is the susbet A_2 of measure ε.

The phenomenon of overfitting suggests the following questions

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.

• There is also another method of minimizing the expected risk, without knowing the probability measure μ corresponding to the joint distribution of labeled pairs (x, y) on $\mathbf{Z} = \mathcal{X} \times \mathcal{Y}$. It is called the stochastic approximation method, suggested by Robbins and Monroe in 1951. This method has been developed into the main methods of machine learning nowadays: the back-propagation method and its version - the Amari natural gradient. The stochastic approximation method works also for density estimation, so we shall consider it in unsupervised learning.

2. Statistical model for unsupervised learning In unsupervised learning we are given output data, a sequence of observables $(x_1, \cdots, x_n)$, without any additional label y_i to the observable x_i. The goal of the learner therefore is to discover "interesting structure" in the data (knowledge discovery). Human learning is mostly unsupervised learning. We live for 10^9 seconds and the brains visual system has 10^{14} neural connections. So its no use learning one bit per second. You need more like 10^5 bits per second.

Currently there is no general mathematical theory for unsupervised learning. The main question is how to formulate the notion of an interesting structure and quantify the degree of "interesting" in mathematical language. Today we shall consider main problem in unsupervised learning: to estimate the probability distribution of a random element x on a measurable space $\mathcal{X}$ as discover the total structure of x, (till now this problem is called density estimation problem), clustering and dimension reduction.

Statistical model and framework for density estimation Density estimation is a central problem of classical statistics. Over a hundred years ago, Karl Pearson proposed that all observations arise from probability distributions, and that the purpose of science is to estimate the parameters of those distributions.

We assume that a sequence of observables $(x_1, \cdots, x_n) \in \Omega^n$ are i.i.d. (identically independently distributed). This is assumption of classical statistics.

The general setting for density estimate is similar to the setting for supervised learning. We consider a family $\mathcal{P}$ of density functions on a measure space (Ω, μ_0). The loss function $L : \Omega \times \mathcal{P} \to \mathbf{R}$ is the minus log-likelihood function

$$L(\omega, p) = -\log p(\omega) d\mu_u$$

where $\mu_u = p_u(\omega) \cdot \mu_0$ is the unknown probability measure to be estimated and $p \in \mathcal{P}$. Hence the expected risk function is

$$R(p) = -\int_\Omega \log p(\omega) p_u d\mu_0.$$

Note that minimizing the risk functional $R(p)$ is the same as minimizing the following modified risk function

$$R^*(p) = R(p) + \int_\Omega \log p_u p_u d\mu_0$$

$$= -\int_\Omega \log \frac{p(\omega)}{p_0(\omega)} p_0(\omega) d\mu_0.$$

The expression on the RHS is the Kullback-Leibler divergence. The Kullback-Leibler divergence can be defined on the space $\mathcal{P}(\Omega)$ of all probability measures on Ω. It is a quasi-distance, i.e.,

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

It is important to find such quasi-instance functions on $\mathcal{P}(\Omega)$ that satisfying certain natural statistical requirement. This problem has been considered in information geometry, which has been initiated by Chentsov under influence of Kolmogorov's lecture at the Poincaré institute Paris in 1955, and later by Amari in Japan motivated by many problems in applied statistics. Recommended books "Information Geometry " (N. Ay, J. Jost, H.V. L, L. Schwachhoefer), "Information Geometry and its applications" (S. Amari).

To minimize a risk functional

$$R : \Omega \times \mathcal{P} \to \mathbf{R},\ R(p) = \int_{\Omega} L(\omega, p)\mu_u(d\omega),$$

using i.i.d. observables $\omega_1, \omega_2, \cdots, \omega_l$, one uses stochastic approximation as follows. One iterates the following procedure for $k = 1, \cdots, l$

$$p(k+1) = p(k) - \gamma_k \nabla_{\mathcal{P}} L(\omega_k, p(k)). \quad (1)$$

If the variable ω_k is fixed, then (1) is the discretization of the gradient flow

$$\dot{p}(t) = \nabla_{\mathcal{P}} L(\omega_k, p(t)).$$

By taking ω randomly, i.e., $\omega = \omega_k$ in (1), we can approximate the expected risk function R^L.

Statistical model for clustering

• Clustering is the process of grouping similar objects together.

• In similarity-based clustering, the input to the algorithm is an $N \times N$ dissimilarity matrix or distance matrix $\mathbf{D}$.

• In feature-based clustering, the input to the algorithm is an $N \times D$ feature matrix or design matrix $\mathbf{X}$.

• Two possible types of output: partitional clustering, where we partition the objects into disjoint sets; and hierarchical clustering, where we create a nested tree of partitions.

• The goal of clustering is to assign points that are similar to the same cluster, and to ensure that points that are dissimilar are in different clusters. There are several ways of measuring these quantities. However, these internal criteria may be of limited use.

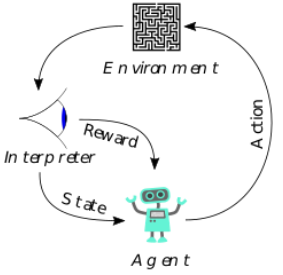
The validation of clustering structures is the most difficult and frustrating part of cluster analysis. Without a strong effort in this direction, cluster analysis will remain a black art accessible only to those true believers who have experience and great courage according to Murphy.

Statistical model and framework for dimension reduction

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. This process is closely related to the concept of (lossy) compression in information theory. That is, if the original data is in $\mathbf{R}^d$ and we want to embed it into $\mathbf{R}^n$, $n < d$, then we would like to find a matrix $W \in Mat_{d,n}$ that induces the mapping $x \mapsto Wx$.

To find a best matrix W we also define certain functional on the space of all possible matrices W and find an optimization problem. A popular method is called Principal Component Analysis (PCA), first proposed by Karl Pearson, where we are also search for a matrix $U \in Mat_{d,n}$ and minimize the distance $\rho(x, UW(x))$.

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 the set of available actions, which is subsequently sent to the environment. The environment moves to a new state s_{t+1} 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 Markov decision process.

Conclusion • Mathematical models for machine learning use the language of statistical decision theory which is the theory of choosing an optimal non-deterministic behavior in incompletely known situations.

• An optimal decision is an algorithm that assign to each sequence S of data a hypothesis h_S that minimizes the expected loss (or expected risk) for supervised or unsupervised learning, respectively maximizes the expected reward/success.

• In the discriminative model for supervised learning we search for a good approximation of a noisy feature y of an instance x by a function h from a wide class $\mathcal{H}$ of possible hypotheses.

• In the generative model for supervised learning we search for underlying distribution of the correlation between a noisy feature y and an instance x.

• The main difficulty for solving optimization problem is that we don't know how to compute the expected loss since the probability measure is unknown. Thus we are lead to use the ERM principle, which may suffer from overfitting, and the stochastic approximation method. To apply ERM method we need to find a sufficient condition to avoid overfitting.

• We also have not discussed the question if our solvable decisions require an acceptable resources for 1) collecting and keeping statistical data 2) computing time of algorithms. In other words we need to quantify the difficulty/complexity of a learning problem which is also related to the last question in yesterday lecture: how to measure a success of a learning machine.

Recommended literature

- M. Mohri, A. Rostamizadeh, A. Talwalkar, Foundations of Machine Learning, MIT Press, 2012.

- K. P. Murphy, Machine Learning: A Probabilistic Perspective, MIT Press, 2012.

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

- M. Sugyiama, Introduction to Statistical Machine Learning, Elsevier, 2016.

THANK YOU FOR YOUR ATTENTION!

Mathematical foundations of machine learning

Lecture 1: Learning, machine learning and artificial intelligent

Lecture 2: Statistical models and frameworks for machine learning

Machine learning is minimization of the unknown expected risk on the basis of empirical data.

Lecture 3: PAC-learning

Question 1 How many random examples does the ERM algorithm need to draw before it has sufficient information to learn unknown hypothesis from the hypothesis class $\mathcal{H}$?

Question 2 How much computation time is required for learning?

The answer to the first question led Vladimir Vapnik to his notion of sample complexity.

The term PAC - probability approximately correct has been proposed by Valiant in 1984.

• The sample complexity and computational complexity quantify the notion of hardness of a learning problem and success of a learning method based on empirical data.

• The term PAC has been proposed by Valiant in 1984. The notion of PAC-learning corresponds to the notion of consistent learning in Vapnik's theory, and the notion of consistent learning stems from the notion of consistency of statistical estimation that has been introduced by Ronald Fisher in 1922.

• There are many versions of consistent learning theory and PAC learning addressing different hypothesis spaces, loss function, algorithm type.

Plan of today lecture:

1. PAC learning.

2. No-Free-Lunch and VC-dimension.

3. Complexity of regression problems.

1. PAC learning

To quantify the success of a learning algorithm A we need to estimate the error (expected risk) of the predictor $A(S) \in \mathcal{H}$, where $S \in \mathcal{Z}^n$ is a random sample. In PAC learning we consider estimations for the error of a predictor with accuracy/correctness ε (approximately)up to confidence $(1-\delta)$ (probably) in probability measure for $S \in \mathcal{Z}^n$.

- For $D \in \mathcal{P}(\mathcal{Z})$, we set

$$P_{S\sim D^m}[f(S)] := D^m(\{S \in \mathcal{Z}^m \mid f(S) \text{ holds } \}).$$

The relation $f(S)$ is usually expressed in terms of inequalities.

$$\mathbb{E}_{S\sim D^m} f := \int_{\Omega^m} f dD^m.$$

- We write $P[f(S)]$, $\mathbb{E}(f)$, if $m = 1$ and D is known (and hence can be omitted).

We say that $\mathcal{H}$ is PAC-learnable w.r.t. the (expected) risk function R^L, if there exist a sample complexity function $m_{\mathcal{H}} : (0,1)^2 \to \mathbf{R}$ and a learning algorithm A with the following property. For every $(\varepsilon, \delta) \in (0,1)^2$, for every distribution D over $\mathcal{Z}$, when running the learning algorithm A on $m \geq m_{\mathcal{H}}(\varepsilon, \delta)$ i.i.d. examples generated by D, the algorithm returns $h_S \in \mathcal{H}$ such that

$$P_{S \sim D^m}[\left(R_D^L(h_S) - \inf_{h' \in \mathcal{H}} R_D^L(h')\right) \leq \varepsilon] \geq 1 - \delta.$$

The sample complexity $m_{\mathcal{H}} : (0,1)^2 \to \mathbf{N}$ of a hypothesis class $\mathcal{H}$ is the function of the accuracy (ε) and the confidence (δ), regarded as variables of the function $m_{\mathcal{H}}$, that appears in the definition of PAC-learnability of $\mathcal{H}$ such that for each (ε, δ) the sample complexity $m_{\mathcal{H}}(\varepsilon, \delta)$ is the minimal number that appears in the Definition of the PAC-learnability.

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

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

Theorem 1 implies that overfitting never happens if we have a finite number of possible hypothesis and arbitrary binary valued loss function.

Explanation of Theorem 1. Given a hypothesis class $\mathcal{H}$ and a training example S the ERM rule selects a minimizer $h_S \in \mathcal{H}$ of the empirical risk $\widehat{R}_S^L(h)$. To make sure that a minimizer h of the empirical risk with respect to S is an expected risk minimizer (or has expected risk close to the minimum) with respect to the true data probability distribution as well, it suffices to ensure that uniformly over all hypotheses in $\mathcal{H}$, the empirical risk will be close to the true (expected) risk.

If $\mathcal{H}$ is finite, then the uniform approximation can be obtained if we can show that for any fixed hypothesis $h \in \mathcal{H}$ the gap between the true and empirical risks, $|\hat{R}_S(h) - L_D(h)|$ is likely to be small. The weak law of large numbers, states that when m goes to infinity, empirical averages converge in probability to their true expectation. However, since the law of large numbers is only an asymptotic result, it provides no information about the gap between the empirically estimated error and its true value for any given, finite, sample size.

Instead, we will use a measure concentration inequality due to Hoeffding, which quantifies the gap between empirical averages and their expected value.

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

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

For more details, see S. Shalev-Schwart and S. Ben-David: Understanding Machine Learning: From theory to Algorithm, p. 57.).

3. No-Free-Lunch and VC-dimension (cf. SSBD2015)

No-Free-Lunch-Theorem Let $\mathcal{X}$ be an infinite domain set. Then the hypothesis class $\mathcal{H} := \{0,1\}^{\mathcal{X}}$ is not PAC-learnable. More precisely, let m be any number smaller than $(1/2)\#\mathcal{X}$, representing a training set size. Then there exist a distribution D over $\mathcal{X}$ and $f \in \{0,1\}^{\mathcal{X}}$ such that

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

$\Longrightarrow$ Without further restriction on $\mathcal{H}$, machines cannot learn, overfitting can happen.

• What are sufficient conditions for learnability of a machine learning problem $(\mathcal{Z}, \mathcal{H}, R^L, A)$?

A partial answer is in terms of VC(Vapnik-Chervonenkis)-dimension.

The VC-dimension is a measure of the capacity (complexity, expressive power, richness, or flexibility) of a space $\mathcal{H}$ of functions that can be learned by a statistical classification algorithm. It is defined as the cardinality of the largest set of points that the algorithm can shatter.

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

Example. A hypothesis class $\mathcal{H}$ shatters a set of one point $x_0 \in \mathcal{X}$ if and only if there are two function $f, g \in \mathcal{H}$ such that $f(x_0) \neq g(x_0)$.

Note that any binary function $h : \mathcal{X} \to \{0, 1\}$ is defined uniquely the subset $h^{-1}(1)$. Thus a hypothesis class $\mathcal{H} \subset \{0, 1\}^{\mathcal{X}}$ can be identified with a collection also denoted by $\mathcal{H}$ of subsets of $\mathcal{X}$. Thus Definition of shattered finite subset can be rewritten as follows.

Definition (M. Steele, Ph.D. Thesis 1975, published in 1978) Given a collection $\mathcal{H}$ of subsets of a set $\mathcal{X}$, we say that the finite subset C of $\mathcal{X}$ is shattered by $\mathcal{H}$ if every subset B contained in C can be written as intersection of C with an element of $\mathcal{H}$.

Definition. The VC-dimension, 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 Let $\mathcal{H}$ be the class of intervals in the real line, namely,

$$\mathcal{H} = \{h_{(a,b)} : a < b \in \mathbf{R}\},$$

where $h_{(a,b)} : \mathbf{R} \to \{0, 1\}$ is the indicate function of $h_{(a,b)}$. Take the set $C = \{1, 2\}$. Then, $\mathcal{H}$ shatters C all the function $\{1, 2\}^{(0,1)}$ 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.

Fundamental theorem of binary classification

Let $\mathcal{H} \subset \{0,1\}^{\mathcal{X}}$ be a hypotheses class with true risk. Then the following are equivalent:

1. Any ERM rule is a successful PAC-learner for $\mathcal{H}$, i.e., $\mathcal{H}$ is PAC-learnable with ERM-rule.

2. $VC\dim(\mathcal{H}) < \infty$.

The main idea of the proof of the fundamental theorem of binary classification consists in establishing that the finiteness of the VC-dimension of $\mathcal{H}$ is sufficient and necessary for the uniform convergence (i.e., independent from $D \in \mathcal{P}(\mathcal{X} \times \mathcal{Y})$ and from $S \in (\mathcal{X} \times \mathcal{Y})^m$) of the estimation error of the empirical minimizer

$$R_D^L(h_S) - \min_{h \in \mathcal{H}} R^L(h)$$

to zero, when m goes to infinity. Once we have the uniform convergence the Hoeffding inequality implies the required PAC-inequality.

4. Complexity of regression problems

In the PAC-learning setting, the VC-dimension is a combinatorial characterization of the hypothesis class $\mathcal{H}$, which carries no topology. The PAC-learning setting is satisfactory for classification problem, where we do not have topology on the label set $\mathcal{Y}$. For the regression problem, where $\mathcal{Y} = \mathbf{R}$, we consider the quadratic loss function and the MSE for a predictor $h \in \mathcal{H}$.

The complexity of regression problems has been considered by Cucker and Smale and their paper "On the mathematical foundations of learning" in 2001. They proved the learnability (or generalization ability) of the regression problem under the compactness assumption of the hypothesis class $\mathcal{H}$. The compactness is related to the topology on the Banach space $C(X)$ of continuous function with the C^0-norm

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

- For a function $f : \mathcal{X} \to \mathcal{Y} = \mathbf{R}$ let

$$f_Y : \mathcal{X} \times \mathcal{Y} \to \mathcal{Y},\ (x, y) \mapsto f(x) - y.$$

Then $MSE(f) = \mathbb{E}_\rho(f_Y^2)$.

- For $g : \mathcal{X} \times \mathcal{Y} \to \mathcal{Y}$ its variance $V_\rho(g)$ is

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

- For a compact $\mathcal{H} \subset C(X)$ and $f \in \mathcal{H}$

$$MSE_{\mathcal{H}}(f) := MSE(f) - MSE(f_{\mathcal{H}}),$$

where $f_{\mathcal{H}} = arg\min_{f \in \mathcal{H}} MSE(f)$.

• $MSE_{\mathcal{H}}(f)$ is (called) the estimation error of f, which is also called the sample error of f.

• For $S = (z_1, \cdots, z_m) \in \mathbf{Z}^m = (\mathcal{X} \times \mathcal{Y})^m$ denote by f_S the minimizer of the empirical MSE

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

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

Theorem 2 (CS2001) Let $\mathcal{H}$ be a compact subset of $C(\mathcal{X})$. Assume that for all $f \in \mathcal{H}$ we have $|f(x) - y| \leq M$ ρ- a. e., and let

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

For all $\varepsilon > 0$, $P_{S \sim \rho^m}[MSE_{\mathcal{H}}(f_S) \leq \varepsilon] \geq$

$$1 - \mathcal{N}(\mathcal{H}, \frac{\varepsilon}{16M}) 2e^{-\frac{m\varepsilon^2}{8(4V_\rho(\mathcal{H}) + \frac{1}{3}M^2\varepsilon)}},$$

where for $s \in \mathbf{R}$ the covering number $\mathcal{N}(\mathcal{H}, s)$ is the minimal l such that there exists l disks in $\mathcal{H}$ with radius s covering $\mathcal{H}$.

Outline of the proof of the Cucker-Smale theorem. The proof of the Cucker-Smale theorem is somewhat similar to the proof of the PAC-learnability of a finite hypothesis class. First we establish a PAC-inequality for one function. Then using the compactness of $\mathcal{H}$ we establish the PAC-inequality for $\mathcal{H}$. Here we need the covering number of $\mathcal{H}$ which could be though as an analogue of the VC-dimension in the case of binary classification.

Theorem 2 implies an upper bound for the sample complexity for learning a regression problem with a compact set $\mathcal{H} \subset C(\mathcal{X})$. Namely for given $(\varepsilon, \delta) \in (0,1)$ to ensure that the expected risk of the empirical risk minimizes f_S is less than ε for almost all $S \in \mathbf{Z}^m$ with confidence $1-\delta$ it is sufficient that m satisfies

$$m \geq \frac{8(4V_\rho(\mathcal{H}) + \frac{1}{3}M^2\varepsilon)}{\varepsilon^2} \times$$

$$\left(\ln\left(2\mathcal{N}(\mathcal{H}, \frac{\varepsilon}{16M})\right) + \ln(\frac{1}{\delta})\right).$$

• If the hypothesis class $\mathcal{H}$ in Theorem 2 is a convex subset in $\mathcal{H}$ then Cucker-Smale got an improved estimation of the confidence $1-\delta$ for a given accuracy ε (CS2001).

Conclusion In this lecture we learn that in presence of uncertainty, the complexity of a learning problem, specified via an expected loss R^L on a hypothesis class $\mathcal{H}$, as well as the success of a learning algorithm must be measured up to probability component. The PAC-learnability of a hypothesis class $\mathcal{H}$ satisfies all these requirements. In the binary classification problem a hypothesis class is PAC-learnable if and only if the VC-dimension $VC\dim(\mathcal{H})$ is finite.

The PAC-learning theory also implies that there is No-Free-Lunch, that is, there is no PAC-learning algorithm for the universal hypothesis class of all hypotheses. In the regression problem, since a loss function is defined in terms of a distance between the true and predicted value, the topology of a hypothesis class is important. Instead of VC-dimension, which is a combinatorial sample complexity, the covering number of a hypothesis class quantifies its complexity.

Final Remarks We note that the PAC-estimation in Cucker-Smale theorem depends on the unknown distribution underlying a labeled data set, which is specified in the number $V_\rho(\mathcal{H})$. On the other hand and the PAC-learnability concept in Vapnik-Chervonenkis theory is distribution free, i.e., it does not depends on the unknown distribution underlying a labeled data set. Nowadays there are many notions of sample complexity that depends on the underlying distribution of a labeled data set, the most notable of them is the Rademacher complexity.

In general to estimate a sample complexity of a hypothesis class is very hard but we should never use experimental fact to jump to conclusion.

As an example I like to mention experiments with the Riemann Hypothesis ($\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s}$ has its zeros only at the negative even integers and complex numbers with real part 1/2). There are several ways how to test the Riemann Hypothesis. An equivalent version is to use the Mobius function $\mu(i)$ ($\mu(n) = 0$ if n is divisible by a square of a prime, $\mu(n) = -1$ if n is an odd number of distinct primes, $\mu(n) = 1$ if n is a product of an even number of distinct primes) and prove/disprove that for all $n > n_0$ $|\sum_{i=1}^{n} \mu(i)| \leq n^{1/2+\varepsilon}$.

For $n > 200$ the value $|\sum \mu(i)| < 1/2\sqrt{n}$ but suddenly for $n = 7,725,038,629$ it exceeds $1/2\sqrt{n}$. In 1985, a counter-example for the Mertens hypothesis $|\sum \mu(i)| < \sqrt{n}$ was found with the present best bound is $n < e^{1,5910^{40}}$. Thus the Mertens conjecture is false, in spite of all empirical evidence that we have so far.

Recommended literature.

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

2. P. Pudlak, Logical Foundations of Mathematics and Computational Complexity, Springer 2013.

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

Mathematical foundations of machine learning

Lecture 1: Learning, machine learning and artificial intelligent

Lecture 2: Statistical models and frameworks for machine learning

Lecture 3: PAC-learning

Lecture 4: Deep learning, pattern theory and algebra of human thoughts

Can machines learn to think like human?

• Machine learning groups are on race to build new models and techniques for the coming age of AGI (B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman in "Building machines that learn and think like people. Behavioral and Brain Sciences" (2016), Ghahramani "Probabilistic machine learning and artificial intelligence" (2015)).

• The current main techniques are deep learning via neural networks and probabilistic (Bayesian) machine learning.

• Deep learning is related to the pattern theory founded by Grenander and developed by Mumford and many others.

• Grenander also analyzed the importance of the pattern theory for AGI.

Plan of today lecture

1. Neural network.

2. Deep learning.

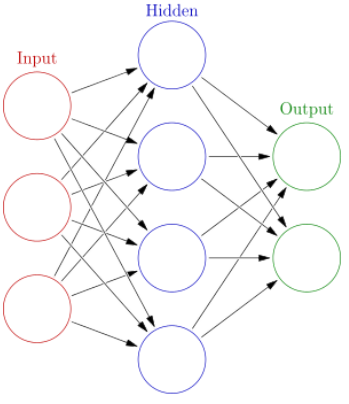
3. Pattern theory and algebra of human thoughts.

4. Summary of our course.

1. Neural network.

• Today most powerful learning machines are artificial neural networks (ANN or NN).

• Neural networks encode a certain class of hypotheses/predictors which can present most of functions machine could learn.



- A NN: $=$ (V, E, σ, w) where V - vertexes (nodes), E - directed edges of the network.

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

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

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

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

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

• The input $I(\mathfrak{n})$ of a neuron $\mathfrak{n}$ is equal to the weighted sum of the outputs of all the neuron connected to it: $I(\mathfrak{n}) = \sum w(\mathfrak{n}'\mathfrak{n})O(\mathfrak{n}')$, where $\mathfrak{n}'\mathfrak{n} \in E$ is a directed edge, $w(\mathfrak{n}'\mathfrak{n}) \in \mathbf{R}$, 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. For the input space $\mathbf{R}^n$ we have $n+1$ input-nodes, one of them is the "constant" neuron, whose output is 1.

• (E, V, w, σ) represents a function $h_{V,E,\sigma,w}$.

• $\mathcal{H}_{V,E,\sigma} = \{h_{V,E,\sigma,w} : \ w \in \mathbf{R}^E\}$ the underlying hypothesis class of functions from the input space to the output space of the network.

Remark Neural networks are abstraction of biological neural networks, and the activation function is usually an abstraction representing the rate of action potential firing in the cell. In its simplest form, this function is binary, that is, either the neuron is firing or not. We can consider activation function as a filter of relevant information.

Neural networks are classified by type of their underlying graph.

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

- A layered feedforward neural network FN has vertices arranged in a disjoint union of layers $V = \cup_{l=0}^{n} V_i$ such that every edge in E connects nodes in neighboring layers V_i, V_{i+1}.

The depth of the network FN is n. V_0 is called the input layer, V_n is called the output layer, the other layer is called hidden.

Example A perceptron $f_{(w,b)}$, where $w \in \mathbf{R}^n$, is a neuron network with $(n+1)$ inputs and with a single neuron and of depth 1. The activation function is $x \mapsto sign\,(x)$. The perceptron, invented in 1957 by Frank Rosenblatt, is an algorithm for supervised learning of binary classifiers: $f_{(w,b)} : \mathbf{R}^n \to \{0, 1\}$.

Here $w \in \mathbf{R}^n$ and $b \in \mathbf{R}$ are defined as follows

$$f_{(w,b)}(x) = \begin{cases} 1 & \text{if } \langle w, x \rangle + b > 0, \\ 0 & \text{otherwise.} \end{cases}$$

Given a sequence of labeled pair $(x_1, y_1), \cdots, (x_n, y_n)$, where $y_i \in \{0, 1\}$, a perceptron has to define the best possible function $f_{(w,b)} \in \mathcal{H}$ such that

$$f_{(w,b)}(x_i) = y_i$$

for all i.

Training perceptron in an ideal condition

$$\tilde{w} := (w, b) \in \mathbf{R}^{n+1},\ \tilde{x} := (x, 1) \in \mathbf{R}^{n+1}.$$

With this new notations our goal is to have $\tilde{w}$ s.t.

(P) $\qquad y_i \langle \tilde{w}, \tilde{x}_i \rangle > 0$ for all i

$$\Longleftrightarrow f_{(w,b)}(x_i) = y_i.$$

Step 1: set $\tilde{w}(1) := (0, \cdots, 0) \in \mathbf{R}^n$.
Step t for $t \geq 2$:
if $\exists i = i(t)$ s.t. $y_i \langle \tilde{w}(t), \tilde{x}_i \rangle \leq 0$ then we set $\tilde{w}(t+1) = \tilde{w}(t) + y_i \tilde{x}_i$,
otherwise $\tilde{w}(t+1) = \tilde{w}(t)$.

• This training will stop after a finite number of steps, if the equation (P) has a solution $\tilde{w}$ and we don't need to define the loss function and minimize it.

• NN are popular since firstly they encode almost all functions we need to compute in a convenient way and secondly we can train them effectively.

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

(2) Every Boolean function $f : \{0,1\}^d \to \{0,1\}$ can be represented exactly by a feed-forward neural network with a single hidden layer containing at most 2^d neurons, if $\sigma(x) = (sign(x) + 1)/2$ is used as activation function.

How to train a neural network?

We shall consider only the case where the input space and the output space of the networks are euclidean spaces $\mathbf{R}^n$ and $\mathbf{R}^m$ respectively.

The risk function we use train our network is the MSE with the loss function

$$L(h_w(x), y) = \frac{1}{2}||h_w(x) - y||^2.$$

Hence for $D \in \mathcal{P}(\mathbf{R}^n \times \mathbf{R}^m)$

$$R_D^L(h_w) = \mathbb{E}_D\Big(L(h_w(x), y\Big).$$

Since D is unknown the ERM principle would minimizes R_S^L and using sample complexity to prove that if h_S minimizes R_S^L then it also minimizes R_D^L with ε-accuracy and δ-confidence if $\#(S) \geq m(\varepsilon, \delta)$.

Current training methods of neural networks uses stochastic gradient we considered in lecture 2, p. 79.

As in lecture 2 we set

$$w_{(k+1)} = w_k - \eta \nabla_w L(h(w_k), z_k)$$

replacing $p(k) \in \mathcal{P}$ by w_k - the parameter of the hypothesis class $\mathcal{H}_{V,E,\sigma}$ and replacing $\omega_k \in \Omega$ by $z_k = (x_k, y_k) \in \mathbf{R}^n \times \mathbf{R}^m$. It can be proved that under certain condition the stochastic gradient converges.

2. Deep learning The success of machine learning algorithms generally depends on data representation, also called feature learning ("Representation Learning: A review and Perspective, IEEE Transactions on Pattern Analysis and Machine Intelligence (2013) by Yoshua Bengio, Aaron Coyrville and Pascal Vincent, "Deep Learning" by Ian Goodfellow Yoshua Bengio and Aaron Courville, MIT Press, 2016).

In machine learning, representation learning is a set of techniques that allows a system to automatically discover the representations needed for feature detection or classification from raw data. This replaces manual feature engineering and allows a machine to both learn the features and use them to perform a specific task.

Feature learning can be either supervised or unsupervised. The question is: how do we automatically find compact representations of data? In probabilistic modeling, we view this as a problem of finding latent variables, which provide a simpler and often lower-dimensional representation of our high-dimensional data.

In statistics, latent variables (from Latin: present participle of lateo (“lie hidden), as opposed to observable variables), are variables that are not directly observed but are rather inferred (through a mathematical model) from other variables that are observed (directly measured). Mathematical models that aim to explain observed variables in terms of latent variables are called latent variable models.

3 Pattern theory and algebra of human thought The methods in representation learning I just mentioned above is more or less empirical and heuristic. A mathematical theory of representation theory can be developed from the general pattern theory by Grenander.

Pattern theory

1. In the real world, signals are mostly stochastic. Signal processing makes use of stochastic properties to find the hidden structure we want to know about.

1a. The set of variables, observed and hidden, typically forms the vertices of a graph, as in Gibbs models, and one must formulate prior probability distributions for the hidden variables as well as models for the observed variables.

1b. When all the stochastic factors affecting any given observation are suitably identified, they show a large amount of conditional independence. We need techniques e.g. PCA (principal component analysis), ICA (independent component analysis), to decompose signals into independent components.

2. The various objects, processes, and rules of the world produce patterns that can be described as precise pure patterns distorted and transformed by a limited family of deformations, similar across all modalities.

2a. One can list the different types of deformations patterns are subject to, thus creating the basic classes of stochastic models that can be applied.

2b. Signals decompose into elementary components which combine and transform via stochastic rules into more complicated signals.

Algebra of human thought U. Grenander, A Calculus of Ideas, A mathematical Study of Human Thought, World Scientific, 2012.

In order to build AGI we need to answer the following question

Can we formalize human thinking?

We can formalize human thinking.

- Aristotle (384322 BC) invented syllogism (a process of logic in which two general statements lead to a more particular statement) as foundation for reasoning and thinking.

• David Hume (1711-1776) "Though our thought seems to possess this unbounded liberty, we shall find, upon a nearer examination, that it is really confined within very narrow limits, and that all this creative power of the mind amounts to no more than the faculty of compounding, transposing, augmenting, or diminishing the materials afforded us by the senses and experience."

• Immanuel Kant (1724 -1804) argued that human thought is essentially architectonic: starting with simple sensory inputs the thinker combines them into abstractions, then combines these into higher level abstractions, and so on.

• Siegmund Freud (1856-1939) analyzed emotional thinking terms of elements: id, ego, superego, censor, libido, castration fear, child sexuality, transfer, repression, Oedipus complex.... They are combined to form the nucleus of the mind of the patient, or at least the subconscious part of it, and are supposed to be discovered by the analyst through examination of dreams, slips, free associations and other expressions of the subconscious.

In Grenander's theory the model of the mind is built in pattern theoretic terms. Starting from simple, atomic, mental entities (the generators of pattern theory) we shall combine them into regular structures, thoughts, (configurations) later on to be controlled by probabilistic rules of connections. In this way patterns of thought will be built as hierarchies of more and more complex structures for which we shall introduce a calculus of ideas.

Main rules of thinking in Grenander's theory

1. Thoughts are made up of discrete entities: ideas.

2. Ideas are connected via bonds; defines semantic.

3. The number of connections to an idea, the arity, is huge.

4. Ideas constituting a thought are bound tightly together: a p-clique (a fully connected graph with p nodes).

5. Thought processes form a metric mind space.

6. Thinking is realized physically by a connected network.

7. The mind equation attributes strengths to ideas and connections between ideas.

8. The network structure implies that thinking is organized in terms of graphs.

9. Language, a small subset of thinking, must also be organized by graphs.

10. Thoughts are created probabilistically by the mind equation.

11. Thoughts are conditioned by boundary conditions.

12. Thoughts are concentrated to neighborhoods N (thought) in mind space.

13. The energy function E has a large number of local minima concentrating around thoughts.

14. The high level study of thinking should take place in mind space, not physical space.

Conclusion The most powerful technique of ML today is deep learning that is performed with neural networks. There are several motivations for applying neural networks. Neural networks have excellent expressive power and good performance using SGD. They are capable of tackling with high dimensional data.

• Many machine learning problems become exceedingly difficult when the number of dimensions in the data is high. This phenomenon is known as the curse of dimensionality.

• Representation learning is a solution to curse of dimensionality.

• Pattern Theory could serve a mathematical foundation for learning representation.

Recommended literature

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

2. Y. Bengio, A. Courville, and P. Vincent, Representation Learning: A Review and New Perspectives, arXic:1206.553.

3. I. Goodfellow, Y. Bengio and A, Courville, Deep Learning, MIT, 2016.

4. D. Mumford and A. Desolneuz, The Stochastic Analysis of Real-World Signals, A K Peters, Ltd., 2010.

5. U. Grenander, A calculus of ideas, A Mathematical Study of Human Thought, World Scientific, 2012.

4. Summary of our course

1. Machine learning is inductive learning based on empirical data, whose performance improves with the increase of size of data.

2. To model noise/incomplete information/random observables we use probability language with different complexity.

3. The sample complexity of a machine learning problem and the success of a learning algorithm is expressed in language of probability theory.

4. To reach AGI we need representation learning, whose mathematical theory might be the Grenander pattern theory.