Mathematical and Statistical Foundations of Machine Learning

Lectures: Ulrich Terstiege
Tutorials: Tizian Wenzel

April 24th, 2025

Chapter I: Introduction

What is Machine Learning?

Some tasks appear too complicated to directly program for a computer, e.g. recognition of objects in images or autonomous driving.

Machine learning may be defined as computational methods for converting experience into expertise.

Experience: past information, data collections, e.g. human labeled training sets such as images (e.g. labeled with 1 or 0 depending on whether or not the image shows a cat) or emails (spam or not spam).

Expertise: Prediction of future outcomes.
Similar to statistics but with a strong emphasis on efficient algorithms (optimization)

Some applications for learning algorithms

- Text / document classification
- speech regonition
- automatic translation
- image recognition / face recognition
- autonomous driving
- search engines, recommendation systems
- Games: chess, Go
- medical diagnosis
- analysis of social networks
- text generation

Some standard learning tasks

- Classification: Assign a category to each item.
  Usually a small number of categories
  Binary classification: two categories ($\{-1,+1\}$, $\{0,+1\}$,...)

  Example: Predict whether email is spam or not
- Regression: Predict (continuous) value for each item

  Examples:
  - Predict maximal temperature of the next day at some place given some of today's weather parameters
  - Predict share price of a company (that is about to go public) from revenues.
- Ranking: Order items accoring to some criterion

  Example: web search: ranking of webpages

Some standard learning tasks

- Clustering: Partition items into several groups

  Example: identify communities in social networks
- Dimensionality reduction / manifold learning: try to find lower dimensional representation of items (vectors) in a high dimensional space while preserving properties of original representation (e.g. distance)

  Example: Preprocessing of digital images in computer vision tasks.

This course

Two parts:

1. Theoretical foundations of Machine Learning and Statistical learning theory: The PAC-Learning Framework, Rademacher Complexity and VC-Dimension
2. Analysis of some machine learning methods and algorithms (with applications of part 1): Some possible topics
   - Support Vector Machines and Kernel Methods
   - Boosting
   - Logistic Regression
   - Stochastic Gradient Descent
   - Neural Networks (Deep Learning)
   - further topics as time allows, e.g. Decision Trees, Clustering, Reinforcement Learning,...

Prerequisites

- Analysis
- Linear algebra
- Basic probability theory

Main References

- S. Shalev Shwartz, S. Ben-David *Understanding Machine Learning – From Theory to Algorithms*. Cambridge University Press 2014
- M. Mohri, A. Rostamizadeh, A. Talwalkar *Foundations of Machine Learning*. second edition, MIT Press 2018

Basic machine learning setup

Input space X, e.g. $X = \mathbb{R}^n$ or $X = [0,1]^2$ or ...

Output space Y, e.g. $Y = \{0,1\}$ or $Y = \mathbb{R}$ or ...

Training data $(x_i, y_i),\ i = 1, \ldots, m$ (labeled data)
[or $x_i,\ i = 1, \ldots, m$ (unlabeled data)]

Hypothesis class $\mathcal{H}$: A set of functions $h : X \to Y$.

Machine learning scenarios I: Supervised Learning

Learner receives labeled data $(x_i, y_i),\ i = 1, \ldots, m$ and tries to make predictions on new data (unseen), i.e. tries to find a function $h : X \to Y,\ h \in \mathcal{H}$, such that for future data (with unknown label) $h(x_i) \approx y_i,\ i = m+1, m+2, \ldots$

Example: $x_i \in X = ([0, 255] \cap \mathbb{Z})^{n_1 \times n_2}$ represents greyscale image with $n_1 \times n_2$ pixels, $y_1 \in Y = \{1, 0\}$ represents whether or not a cat is in the image. (Alternatively, $X = [0, 255]^{n_1 \times n_2}$ or $[0, 1]^{n_1 \times n_2}$ etc.
For colored pictures one could use $X = ([0, 255] \cap \mathbb{Z})^{3n_1 \times n_2}$ (rgb channels).

Machine learning scenarios II: Unsupervised Learning

Learner receives unlabeled data $x_i, \ i = 1, \ldots, m$ and tries to make predictions on unseen data, for instance learns something about the structure of the data points, e.g. they may be contained in a subspace or submanifold of $X \subset \mathbb{R}^n$ ($\rightarrow$ dimensionality reduction, manifold learning), or they may cluster into a few clusters ($\rightarrow$ clustering)

Example: A retailer might want to cluster its costumers into a few groups in order to adapt its strategy.

Machine learning scenarios III: Semisupervised Learning

Learner receives labeled data $(x_i, y_i),\ i = 1, \ldots, m$ and unlabeled data $x_i,\ i = m+1, \ldots, N$ and predicts labels of unseen unlabeled data $x_{N+1}, \ldots$, i.e. tries to find a function $h : X \to Y,\ h \in \mathcal{H}$, such that for future data (with unknown label) $h(x_i) \approx y_i,\ i = N+1, N+2, \ldots$ (Hope that unlabeled training data $x_i,\ i = m+1, \ldots, N$ help to improve the prediction, e.g. by helping to learn something about the structure of the set of the possible data x.)

Machine learning scenarios IV: Online learning

There are multiple training rounds. At each round, the learner receives an unlabeled training point x_j, makes a prediction $\hat{y}_j \in Y$ of its label y_j, then the true label y_j is received and the learner incurs the loss $\ell(y_j, \hat{y}_j)$.
Goal: Minimize cumulative loss $\sum_{j=1}^{m} \ell(y_j, \hat{y}_j)$.

Example: At each round, receive an email x_j, predict whether it is spam or not, receive information whether it is spam or not (by a human reader of the email), if prediction was wrong, adapt the predictor.

Machine learning scenarios V: Reinforcement learning

Mixed training and testing phase. The learner decides on actions interacting with environment and receives immediate reward / loss for each action.
Task: Maximize total reward over course of actions

→ Exploration vs. exploitation dilemma: decide between unexplored action to gain more information about environment and known action exploiting already collected information.

Examples: Games (e.g. Go, Chess), Advertisement on websites

Machine learning scenarios VI: Active learning

Similar to supervised learning, but learner can decide on data points x_i to query the label y_i.

Examples: Scientific experiments, oil exploration

Hope: Better predictions / less needed samples than in supervised learning.

Chapter II: The PAC-Learning Framework

PAC: Probably Approximately Correct

Consider supervised learning scenario for binary classification:

- X: Input space, e.g. Input space X, e.g. $X = \mathbb{R}^n$ or $X = [0,1]^2$ or $X = ([0,255] \cap \mathbb{Z})^{3n_1 \times n_2}$ or...
- Set of labels Y, for now $Y = \{0,1\}$ (or $Y = \{-1,1\}$)
- Training data $S = ((x_1, y_1), (x_2, y_2), \ldots (x_m, y_m)) \in (X \times Y)^m$ (labeled domain points)

Assumptions

1. The points $x_i, i = 1, \ldots, m$ are drawn independently and identically distributed (i.i.d.) according to some unknown probability distribution $\mathcal{D}$ on X.
2. The labels are given as $y_i = f(x_i), i = 1, \ldots, m$ for some map $f : X \to Y$.

Goal: Find f (at least approximately)

More generally (later): Assume that the labeled examples (x_i, y_i) are drawn i.i.d. according to some unknown probability distribution $\mathcal{D}$ on $X \times Y$.

Learner's output: A prediction rule (predictor, hypothesis, classifier)

$$h\colon X \to Y.$$

Ideally: $h = f$.
Learner selects hypothesis from a hypothesis set

$$\mathcal{H} \subset \{g\colon X \to Y\}$$

and f may or may not be contained in $\mathcal{H}$.

Assumptions on measurable spaces

We will assume (unless otherwise mentioned) that for measurable spaces (typically denoted by $X, Y, Z, \dots$) the following assumptions apply.

- If the space is finite or countably infinite, it is equipped with the σ-algebra consisting of all subsets of the space (applies e.g. for $X = \mathbb{N},\ X = ([0, 255] \cap \mathbb{Z})^{3n_1 \times n_2}, X = \mathbb{Q}^n, Y = \{0, 1\}, \dots)$
- Otherwise it is a metric space which is complete and separable and equipped with the corresponding Borel σ-algebra (applies e.g. for $X = \mathbb{R}^n, [0, 1]^2, \dots)$
- If it is given as a product $X \times Y$, then the product carries the product σ-algebra of X and Y.

(Note that the case of a finite or countably infinite space X can be seen as a special case of a complete separable metric space with corresponding Borel σ-algebra by using e.g. the trivial metric $d(x, y) = \delta_{x,y}$, i.e $d(x, x) = 0$ and $d(x, y) = 1$ for $x \neq y$. Also e.g. the product of two spaces as in the first or as in the second case have again this form, respectively.)

Generalization Error

For the next definition, we assume a fixed labeling function $f\colon X \to Y = \{0,1\}$ and a probability distribution $\mathcal{D}$ on X.

Definition 2.1 (Generalization error)

For $h\colon X \to Y = \{0,1\}$, the generalization error or risk of h is defined as

$$R(h) = \underset{x\sim\mathcal{D}}{\mathbb{P}}(h(x) \neq f(x)) = \mathbb{E}[\mathbb{1}_{\{x|h(x)\neq f(x)\}}].$$

Here:

- $\mathbb{P}$ denotes the probability of an event.
- $\mathbb{E}$ denotes expectation (wrt. $\mathcal{D}$).
- $\mathbb{1}_A$ denotes the indicator function of A, i.e. $\mathbb{1}_A(x) = 1$ for $x \in A$ and $\mathbb{1}_A(x) = 0$ for $x \notin A$.
- f and h are always assumed to be measurable.

Note that the generalization error is not directly accessible since $\mathcal{D}$ and f are unknown.

Empirical Risk

Definition 2.2 (Empirical Risk)

For $h\colon X \to Y = \{0,1\}$, the (true) labeling function $f\colon X \to Y = \{0,1\}$ and a sample $S = (x_1, \ldots, x_m)$ (with $x_i \in X$), the empirical risk of h is defined as

$$\hat{R}(h) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}_{h(x_i) \neq f(x_i)} = \frac{1}{m} \#\{i \in [m]\colon h(x_i) \neq f(x_i)\}.$$

Here $[m] := \{1, \ldots, m\}$ and for a finite set A, we denote by $\#A$ the number of elements in A.

Drawing $x_1, \ldots, x_m$ i.i.d. according to $\mathcal{D}$, i.e. drawing $S = (x_1, \ldots, x_m) \sim D^m$, we obtain a random variable which we also denote by $\hat{R}(h)$.

Lemma 2.3

If $x_1, \ldots, x_m$ are drawn i.i.d. according to $\mathcal{D}$ then for any (measurable) $h \colon X \to \{0,1\}$:

$$\mathbb{E}[\hat{R}(h)] = R(h).$$

Definition 2.4 (Empirical Risk Minimization)

Let $f \colon X \to Y = \{0,1\}$ be the true labeling function and let $\mathcal{H} \subset \{h \colon X \to Y\}$ be a hypothesis set. Given a sample $S = (x_1, \ldots, x_m) \in X^m$ with corresponding labels $y_i = f(x_i)$ for $i \in \{1, \ldots, m\}$, empirical risk minimization consists of selecting a minimizer $h \in \mathcal{H}$ of $\hat{R}$, i.e. selecting a h realizing

$$\min_{h \in \mathcal{H}} \hat{R}(h) = \min_{h \in \mathcal{H}} \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}_{h(x_i) \neq f(x_i)}.$$

PAC Learning

PAC = Probably Approximately Correct

Definition 2.5 (PAC-Learning, consistent case)

A hypothesis class $\mathcal{H} \subset \{h \colon X \to Y = \{0,1\}\}$ is PAC-learnable if there exists a function $m_{\mathcal{H}} \colon (0,1)^2 \to \mathbb{N}$ and a learning algorithm $\mathcal{A}$ with the following property:
For every $\varepsilon, \delta \in (0,1)$, for every probability distribution $\mathcal{D}$ over X, and for every labeling function $f \in \mathcal{H}$, the following holds:
If $m \geq m_{\mathcal{H}}(\varepsilon, \delta)$ and $S = (x_1, \dots, x_m)$ is an i.i.d. sample, $S \sim \mathcal{D}^m$, then given the data $(x_i, y_i) = (x_i, f(x_i)), i = 1, \dots, m$, the algorithm $\mathcal{A}$ returns a hypothesis $h_S \in \mathcal{H}$ such that with probability at least $1 - \delta$ (over $S \sim \mathcal{D}^m$), it holds

$$R(h_S) \leq \varepsilon.$$

Remarks

Analogous definitions of empirical risk, empirical risk minimization and PAC-learning for any Y consisting of two elements.

The sample complexity $m_{\mathcal{H}} \colon (0,1)^2 \to \mathbb{N}$ determines the number of required training data in order to learn $\mathcal{H}$. Depends on accuracy ε and confidence δ and on properties of $\mathcal{H}$. Ideally, $m_{\mathcal{H}}$ is bounded by a polynomial in $\frac{1}{\varepsilon}$ and $\frac{1}{\delta}$. (More precisely, the sample complexity should be defined as the minimal possible $m_{\mathcal{H}}$ satisfying the conditions in the definition.)

The definition does not require the algorithm to be efficient, only the existence of a possibly slow (intractable) algorithm is assumed. If the runtime of the algorithm is polynomial (in $\frac{1}{\varepsilon}, \frac{1}{\delta}$ and the "computational representation of $f \in \mathcal{H}$") then we call $\mathcal{H}$ efficiently PAC-learnable.

ERM is a possible "algorithm", but it may not always be the optimal one. Depending on $\mathcal{H}$ it may be efficient or not.

In some cases, PAC-learnability may be shown directly (example: Learning axis aligned rectangles, see exercises).

Theorem 2.6 (Finite $\mathcal{H}$, consistent case)

Let $\mathcal{H}$ be a finite set of functions $h\colon X \to Y = \{0,1\}$. Assume that the labeling function f belongs to $\mathcal{H}$ and let $\mathcal{A}$ be an algorithm that for each i.i.d. sample $S = (x_1, \dots, x_m)$ and labeled training data $(x_i, y_i) = (x_i, f(x_i)), i = 1, \dots, m$ returns a consistent hypothesis $h_S \in \mathcal{H}$, i.e. $\hat{R}(h_S) = 0$. Then, for $\varepsilon, \delta \in (0,1)$, if

$$m \geq \frac{1}{\varepsilon}(\log |\mathcal{H}| + \log(\frac{1}{\delta}))$$

the inequality $\mathbb{P}[R(h_S) \leq \varepsilon] \geq 1 - \delta$ holds.

In other words, with probability at least $1 - \delta$:

$$R(h_S) \leq \frac{1}{m}(\log |\mathcal{H}| + \log(\frac{1}{\delta})).$$

The theorem shows that under its assumptions $\mathcal{H}$ is PAC-learnable with $m_{\mathcal{H}}(\varepsilon, \delta) = \left\lceil \frac{1}{\varepsilon}(\log |\mathcal{H}| + \log(\frac{1}{\delta})) \right\rceil$.