1

Introduction

The goal of mathematical analysis of machine learning algorithms is to study the statistical and computational behaviors of methods that are commonly used in machine learning, and to understand their theoretical properties, such as the statistical rate of convergence (usually deriving upper bounds for specific algorithms), the optimality of a statistical method (whether the derived statistical upper bound matches the information theoretical lower bound), and the computational efficiency for various learning models under different assumptions.

This book mainly focuses on the analysis of two common learning models: supervised learning and sequential decision-making problems.

In supervised learning, we train a machine learning model using training data, and then evaluate the model's prediction performance on unseen test data. In this case, we want to investigate the performance of this model on test data.

A mathematical theory for supervised learning answers the following basic questions, where we take the linear model as an example.

- Suppose that we learn a d-dimensional linear classifier with n training data by minimizing the training error. Assume that the training error is 10%. What is the classifier's test error on the (unseen) test data? The test error in this setting is also referred to as the *generalization error*, because it is not observed.
- Can we find a linear classifier that has a test error nearly as small as the optimal linear classifier?
- Can we find a computationally efficient procedure to find a linear classifier with a small test error?

The online learning model is an example of sequential decision-making problems. In online learning, we are interested in the sequential prediction problem, where we train a statistical model using historic data, and then test it on the data in the next time step. We then observe the true outcome after prediction. This process is repeated in a sequential manner. The problem itself is motivated from time series analysis and forecasting problems. We want to know the ability of a learning algorithm to predict future events based on historic observations.

A mathematical theory for online learning needs to answer the following basic questions, where we, again, take the linear model as an example.

https://doi.org/10.1017/9781009093057.002 Published online by Cambridge University Press

- In the online sequential prediction setting. Given a time step t, can we construct an online learning algorithm that predicts nearly as well as the optimal linear classifier up to time step t?

This course develops the mathematical tools to answer these questions.

1.1 Standard Model for Supervised Learning

In supervised learning, we observe an input random variable (feature vector) $X \in \mathbb{R}^d$ that represents the known information, and an output variable (label) Y that represents the unknown information we want to predict. The goal is to predict Y based on X.

As an example, we may want to predict whether an image (represented as input vector X) contains a cat or a dog (label Y).

In practice, the set of prediction rules are derived by parametrized functions $f(w, \cdot)\colon \mathbb{R}^d \to \mathbb{R}^k$, where $w \in \Omega$ is the model parameter that can be learned on the training data. As an example, for the k-class classification problem, where $Y \in \{1, \ldots, k\}$, we predict Y using the following prediction rule given function $f(w, x) = [f_1(w, x), \ldots, f_k(w, x)] \in \mathbb{R}^k$:

$$q(x) = \arg \max_{\ell \in \{1,\ldots,k\}} f_\ell(w, x).$$

The prediction quality is measured by a loss function $L(f(x), y)$: the smaller the loss, the better the prediction accuracy.

The supervised learning approach is to estimate $\hat{w} \in \Omega$ based on observed (labeled) historical data $\mathcal{S}_n = [(X_1, Y_1), \ldots, (X_n, Y_n)]$.

A supervised learning algorithm $\mathcal{A}$ takes a set of training data $\mathcal{S}_n$ as input, and outputs a function $f(\hat{w}, \cdot)$, where $\hat{w} = \mathcal{A}(\mathcal{S}_n) \in \Omega$. The most common algorithm, which we will focus on in this course, is *empirical risk minimization* (ERM):

$$\hat{w} = \arg \min_{w \in \Omega} \sum_{i=1}^{n} L(f(w, X_i), Y_i). \tag{1.1}$$

In the standard theoretical model for analyzing supervised learning problems, we assume that the training data $\{(X_i, Y_i) : i = 1, \ldots, n\}$ are iid (independent and identically distributed), according to an unknown underlying distribution $\mathcal{D}$. The loss of a classifier $\hat{f}(x) = f(\hat{w}, x)$ on the training data is the training error:

$$\text{training-loss}(\hat{w}) = \frac{1}{n} \sum_{i=1}^{n} L(f(\hat{w}, X_i), Y_i).$$

Moreover, we assume that the test data (X, Y) (future unseen data) are also taken from the same distribution $\mathcal{D}$, and we are interested in knowing the generalization error of $\hat{f}$ on the test data, defined as:

$$\text{test-loss}(\hat{w}) = \mathbb{E}_{(X,Y)\sim\mathcal{D}} L(f(\hat{w}, X), Y).$$

https://doi.org/10.1017/9781009093057.002 Published online by Cambridge University Press

Since we only observe the training error of $\hat{f} = f(\hat{w}, \cdot)$, a major goal is to estimate the test error (i.e. generalization error) of $\hat{f}$ based on its training error, referred to as *generalization bound*, which is of the following form. Given $\epsilon \geq 0$, we want to determine $\delta_n(\epsilon)$ so that

$$\Pr\left(\mathbb{E}_{(X,Y)\sim\mathcal{D}} L(f(\hat{w}, X), Y) \geq \frac{1}{n}\sum_{i=1}^{n} L(f(\hat{w}, X_i), Y_i) + \epsilon\right) \leq \delta_n(\epsilon),$$

where the probability is with respect to the randomness over the training data $\mathcal{S}_n$. In general, $\delta_n(\epsilon) \to 0$ as $n \to \infty$.

In the literature, this result is often stated in the following alternative form, where we want to determine a function $\epsilon_n(\delta)$ of δ, so that with probability at least $1 - \delta$ (over the random sampling of the training data $\mathcal{S}_n$):

$$\mathbb{E}_{(X,Y)\sim\mathcal{D}} L(f(\hat{w}, X), Y) \leq \frac{1}{n}\sum_{i=1}^{n} L(f(\hat{w}, X_i), Y_i) + \epsilon_n(\delta). \tag{1.2}$$

We want to show that $\epsilon_n(\delta) \to 0$ as $n \to \infty$.

Another style of theoretical result, often referred to as *oracle inequalities*, is to show that with probability at least $1 - \delta$ (over the random sampling of training data $\mathcal{S}_n$):

$$\mathbb{E}_{(X,Y)\sim\mathcal{D}} L(f(\hat{w}, X), Y) \leq \inf_{w\in\Omega} \mathbb{E}_{(X,Y)\sim\mathcal{D}} L(f(w, X), Y) + \epsilon_n(\delta). \tag{1.3}$$

This shows that the test error achieved by the learning algorithm is nearly as small as that of the optimal test error achieved by $f(w, x)$ with $w \in \Omega$. We say the learning algorithm is consistent if $\epsilon_n(\delta) \to 0$ as $n \to 0$. Moreover, the rate of convergence refers to the rate of $\epsilon_n(\delta)$ converging to zero when $n \to \infty$.

Chapter 2 and Chapter 3 establish the basic mathematical tools in empirical processes for analyzing supervised learning. Chapter 4, Chapter 5, and Chapter 6 further develop the techniques. Chapter 7 considers a different analysis that directly controls the complexity of a learning algorithm using stability. This analysis is gaining popularity due to its ability to work directly with algorithmic procedures such as SGD. Chapter 8 introduces some standard techniques for model selection in the supervised learning setting. Chapter 9 analyzes the kernel methods. Chapter 10 analyzes additive models with a focus on sparsity and boosting. Chapter 11 investigates the analysis of neural networks. Chapter 12 discusses some common techniques and results for establishing statistical lower bounds.

1.2 Online Learning and Sequential Decision-Making

In online learning, we consider observing (X_t, Y_t) one by one in a time sequence from $t = 1, 2, \ldots$. An online algorithm $\mathcal{A}$ learns a model parameter $\hat{w}_t$ at time t based on previously observed data $(X_1, Y_1), \ldots, (X_t, Y_t)$:

$$\hat{w}_t = \mathcal{A}(\{(X_1, Y_1), \ldots, (X_t, Y_t)\}).$$

https://doi.org/10.1017/9781009093057.002 Published online by Cambridge University Press

We then observe the next input vector X_{t+1}, and make prediction $f(\hat{w}_t, X_{t+1})$. After the prediction, we observe Y_{t+1}, and then compute the loss $L(f(\hat{w}_t, X_{t+1}), Y_{t+1})$. The goal of online learning is to minimize the aggregated loss:

$$\sum_{t=0}^{T-1} L(f(\hat{w}_t, X_{t+1}), Y_{t+1}).$$

In the mathematical analysis of online learning algorithms, we are interested in the following inequality, referred to as *regret bound*, where the aggregated loss of an online algorithm is compared to the optimal aggregated loss:

$$\sum_{t=0}^{T-1} L(f(\hat{w}_t, X_{t+1}), Y_{t+1}) \leq \inf_{w\in\Omega} \sum_{t=0}^{T-1} L(f(w, X_{t+1}), Y_{t+1}) + \epsilon_T. \tag{1.4}$$

The regret ϵ_T, is the extra loss suffered by the learning algorithm, compared to that of the optimal model at time T in retrospect.

As an example, we consider the stock price prediction problem, where the opening price of a certain stock at each trading day is $p_1, p_2, \ldots$. At the beginning of each day t, we observe $p_1, \ldots, p_t$, and want to predict p_{t+1} on day $t+1$, so we use this prediction to trade the stock.

The input X_{t+1} is a d-dimensional real-valued vector in R^d that represents the observed historical information of the stock on day t. The output $Y_{t+1} = \ln(p_{t+1}/p_t)$ will be observed on day $t+1$. We consider a linear model with $f(w, x) = w^\top x$, with $\Omega = \mathbb{R}^d$. The quality is measured by the least squares error:

$$L(f(w, X_{t+1}), Y_{t+1}) = (f(w, X_{t+1}) - Y_{t+1})^2.$$

The learning algorithm can be empirical risk minimization, where

$$\hat{w}_t = \arg\min_{w\in\mathbb{R}^d} \frac{1}{t} \sum_{i=1}^{t} (w^\top X_i - Y_i)^2.$$

In regret analysis, we compare the prediction error

$$\sum_{t=0}^{T-1} (\hat{w}_t^\top X_{t+1} - Y_{t+1})^2$$

to the optimal prediction

$$\inf_{w\in\mathbb{R}^d} \sum_{t=0}^{T-1} (w^\top X_{t+1} - Y_{t+1})^2.$$

Martingale inequalities used in the analysis of sequential decision problems will be introduced in Chapter 13. The online learning model will be studied in Chapter 14 and Chapter 15. The related bandit problem will be investigated in Chapter 16 and Chapter 17. In the bandit problem, we investigate online problems with incomplete information, where Y_t is only partially revealed based on actions of the learning algorithm. The goal is to take an optimal sequence of actions to maximize rewards (or minimize loss). Finally, in Chapter 18, we will introduce some

https://doi.org/10.1017/9781009093057.002 Published online by Cambridge University Press

basic techniques to analyze reinforcement learning. The reinforcement learning model can be considered as a generalization of the bandit model, where, at each time step (epoch), multiple actions are taken to interact with the environment. This is still an actively developing field, with major theoretical advances appearing in recent years. We will only cover some basic results that are most closely related to the analysis of bandit problems.

1.3 Computational Consideration

In the ERM method, the model parameter $\hat{w}$ is the solution of an optimization problem. If the optimization problem is convex, then the solution can be efficiently computed. If the optimization problem is nonconvex, then its solution may not be obtained easily.

Theoretically, we separately consider two different types of complexity. One is statistical complexity, where we may ignore the complexity of computation, and try to derive bounds (1.3) and (1.4) even though the computational complexity of the underlying learning algorithm (such as ERM) may be high.

However, in practice, an important consideration is computational complexity, where we are interested in computationally efficient algorithms with good generalization performance or regret bounds. For nonconvex models, this kind of analysis can be rather complex, and usually requires problem specific analyses that are not generally applicable.

A generally studied approach to a nonconvex problem is to use convex approximation (also referred to convex relaxation) to solve the nonconvex problem approximately. The related theoretical question is under what circumstances the solution has statistical generalization performance comparable to that of the nonconvex methods. An example is the sparse learning problem, where the convex formulation with L_1 regularization is used as a proxy to the nonconvex L_0 regularization. In this case, we are interested in establishing the condition under which one can obtain a solution from L_1 regularization that is close to the true sparse model.

The combined analysis of computational and statistical complexity is a major research direction in theoretical machine learning. This book mainly covers the statistical analysis aspect. Nevertheless, the computational complexity will also be considered when practical algorithms are investigated.

1.4 Basic Concepts in Generalization Analysis

The goal of machine learning is to find a function $f(\hat{w}, x)$ that predicts well on unseen data (test data). However, we only observe the prediction accuracy of $f(\hat{w}, x)$ on the training data. In order to achieve high prediction accuracy, we need to balance the following two aspects of learning:

- The prediction function should fit the training data well; that is, achieve a small training error. This requires a more expressive model, with a larger parameter space Ω.

https://doi.org/10.1017/9781009093057.002 Published online by Cambridge University Press

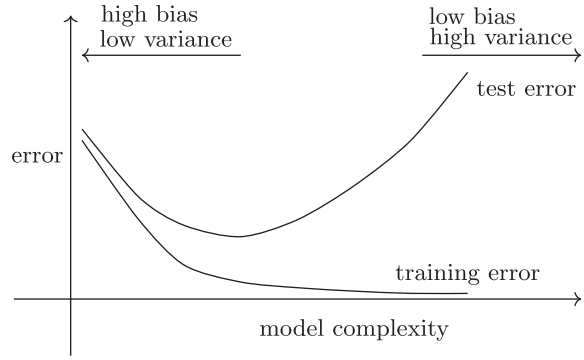

Figure 1.1 Training and test errors versus model complexity

- Performance of prediction function on the test data should match that on the training data. The difference is smaller for a less expressive model with a smaller parameter space Ω.

The gap between training error and test error depends on model complexity, which characterizes how large the model parameter space Ω is. When Ω is too large, the training error becomes smaller, but the difference between training error and test error increases. Therefore, in practice, there is a trade-off in machine learning, and the best prediction performance is achieved with the right balance, often via a tuning parameter in the learning algorithm that characterizes model complexity. The phenomenon is described in Figure 1.1. Such a tuning process is often referred to as hyperparameter optimization.

When the class of prediction functions is too large (or complex), then the difference between training error and test error increases. This leads to the so-called *overfitting* phenomenon. A simple example of overfitting can be described as follows. Let X be a one-dimensional feature uniformly distributed in $[-1, 1]$, with class label $Y = 1$ when $X \geq 0$ and $Y = -1$ when $X < 0$. The optimal classifier can achieve a test error of 0.

Given training data (X_i, Y_i) $(i = 1, \ldots, n)$, and assuming X_i are all different, if we consider a prediction function class that contains all possible functions, then the empirical risk minimization method with the following solution can fit the data perfectly:

$$\hat{f}(X) = \begin{cases} Y_i & \text{if } X = X_i \text{ for some } i, \\ 1 & \text{otherwise.} \end{cases}$$

This model class has high model complexity measured by its *covering number*, which we will study in the book. However, the resulting ERM prediction rule does not make any meaningful prediction when X is not in the training data. This is because, although the training error of 0 is small, it is significantly different from the test error of 0.5.

https://doi.org/10.1017/9781009093057.002 Published online by Cambridge University Press

In contrast, if we let the prediction model contain only one function $\{f(x)\colon f(x) \equiv 0\}$, then, using the tail inequality of independent random variables of Chapter 2, we know that the difference between the training error and the test error will be small when n is large. However, since the training error of ≈ 0.5 is large, the test error is also large.

Let $\mathbb{1}(x \in A)$ be the set indicator function that takes value 1 if $x \in A$, and 0 if $x \notin A$. Assume that we pick the model function class $\{f(w,x)\colon f(w,x) = 2\mathbb{1}(x \geq w) - 1\}$ parametrized by a parameter $w \in R$. Assume also that we find a classifier $f(\hat{w}, x)$ that minimizes the training error. Using techniques in Chapter 3, it can be shown that both training error and test error of this classifier converge to zero when $n \to \infty$. This model class balances the training error and generalization performance. In summary, a key technique of the mathematical theory for machine learning is to estimate the generalization performance (prediction accuracy on unseen data) of learning algorithms, and quantify the degree of overfitting.

Finally it is worth pointing out that the mathematical theory developed for limiting model size and preventing overfitting is the key classical technique to obtain good generalization results in machine learning. However, in recent years, this classical view point has evolved due to the empirical observation in modern neural network models that large models nearly always perform better. For such models, one observes the so-called *benign overfitting* phenomenon, where learning algorithms with appropriate *implicit bias* can still achieve good test performance even if the resulting model completely overfits the noise. This is an active research area that is still developing rapidly. Consequently, the related theoretical results are less mature. We will thus only discuss some theoretical intuitions behind this phenomenon in Section 11.7, but dedicate the main parts of the book to the classical learning theory.

1.5 Historical and Bibliographical Remarks

Machine learning is now considered the key technology for artificial intelligence (AI), which aims to create computing machines that can mimic the problem-solving skills of a human (McCarthy et al., 2006). In recent years, machine learning has become an important scientific research field on its own, and has many applications that have made significant impact in our modern society. The term "machine learning" has often been attributed to Samuel (1959), who defined it as the "field of study that gives computers the ability to learn without being explicitly programmed."

There are two approaches to machine learning (AI): one is to use statistical methods to learn functions from data and past experience, in order to predict future events. This is the approach considered in this book. An alternative approach to AI is symbolic reasoning, which creates a knowledge base, and then uses logic to create rules that can perform inference (Haugeland, 1989). The latter approach explicitly incorporates human knowledge into computer programs, without the need for direct learning from past experiences. Although the symbolic approach

https://doi.org/10.1017/9781009093057.002 Published online by Cambridge University Press

showed some promise in the early decades of AI research (Studer et al., 1998), it has major limitations in dealing with uncertainty in real-world applications. For complex problems, the symbolic rules needed to handle difficult situations are often too complex to build and maintain. For this reason, the modern applications of machine learning heavily relied on the statistical approach, although the hybrid of statistic-based machine learning and symbolic AI is still an active research direction.

The mathematical foundation of machine learning has its origin in probability and theoretical statistics. In particular, the theory of empirical processes has been used to analyze the generalization performance of machine learning algorithms. The first part of the book will describe the basic tools of empirical processes that are commonly used in machine learning. Learning in the sequential decision setting is a different paradigm for theoretical analysis, and the key quantity of interests, regret bound, has its origin in theoretical computer science. The techniques used in the analysis are also closely related to stochastic optimization and stochastic processes. Both computational and statistical aspects are considered in some of the procedures while only the statistical aspects are considered for others. The second part of the book will describe the mathematical tools for analyzing learning problems in the sequential decision setting.

https://doi.org/10.1017/9781009093057.002 Published online by Cambridge University Press