CHAPTER

STATISTICAL MACHINE LEARNING

1

CHAPTER CONTENTS

Types of Learning 3
Examples of Machine Learning Tasks 4
Supervised Learning 4
Unsupervised Learning 5
Further Topics 6
Structure of This Textbook 8

Recent development of computers and the Internet allows us to immediately access a vast amount of information such as texts, sounds, images, and movies. Furthermore, a wide range of personal data such as search logs, purchase records, and diagnosis history are accumulated everyday. Such a huge amount of data is called *big data*, and there is a growing tendency to create new values and business opportunities by extracting useful knowledge from data. This process is often called *data mining*, and *machine learning* is the key technology for extracting useful knowledge. In this chapter, an overview of the field of machine learning is provided.

1.1 TYPES OF LEARNING

Depending on the type of available data, machine learning can be categorized into *supervised learning*, *unsupervised learning*, and *reinforcement learning*.

Supervised learning would be the most fundamental type of machine learning, which considers a student learning from a supervisor through questioning and answering. In the context of machine learning, a student corresponds to a computer and a supervisor corresponds to a user of the computer, and the computer learns a mapping from a question to its answer from paired samples of questions and answers. The objective of supervised learning is to acquire the *generalization ability*, which refers to the capability that an appropriate answer can be guessed for unlearned questions. Thus, the user does not have to teach everything to the computer, but the computer can automatically cope with unknown situations by learning only a fraction of knowledge. Supervised learning has been successfully applied to a wide range of real-world problems, such as hand-written letter recognition, speech recognition,

An Introduction to Statistical Machine Learning. DOI: 10.1016/B978-0-12-802121-7.00012-1

image recognition, spam filtering, information retrieval, online advertisement, recommendation, brain signal analysis, gene analysis, stock price prediction, weather forecasting, and astronomy data analysis. The supervised learning problem is particularly called *regression* if the answer is a real value (such as the temperature), *classification* if the answer is a categorical value (such as "yes" or "no"), and *ranking* if the answer is an ordinal value (such as "good," "normal," or "poor").

Unsupervised learning considers the situation where no supervisor exists and a student learns by himself/herself. In the context of machine learning, the computer autonomously collects data through the Internet and tries to extract useful knowledge without any guidance from the user. Thus, unsupervised learning is more automatic than supervised learning, although its objective is not necessarily specified clearly. Typical tasks of unsupervised learning include *data clustering* and *outlier detection*, and these unsupervised learning techniques have achieved great success in a wide range of real-world problems, such as system diagnosis, security, event detection, and social network analysis. Unsupervised learning is also often used as a preprocessing step of supervised learning.

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 student's questions. Instead, the supervisor *evaluates* the student's behavior and gives feedback about it. The objective of reinforcement learning is, based on the feedback from the supervisor, to let the student improve his/her behavior to maximize the supervisor's evaluation. Reinforcement learning is an important model of the behavior of humans and robots, and it has been applied to various areas such as autonomous robot control, computer games, and marketing strategy optimization. Behind reinforcement learning, supervised and unsupervised learning methods such as regression, classification, and clustering are often utilized.

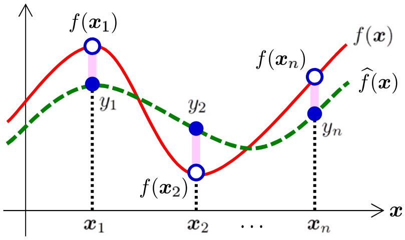
The focus on this textbook is supervised learning and unsupervised learning. For reinforcement learning, see references [99, 105]

1.2 EXAMPLES OF MACHINE LEARNING TASKS

In this section, various supervised and unsupervised learning tasks are introduced in more detail.

1.2.1 SUPERVISED LEARNING

The objective of *regression* is to approximate a real-valued function from its samples (Fig. 1.1). Let us denote the input by d-dimensional real vector $\boldsymbol{x}$, the output by a real scalar y, and the learning target function by $y = f(\boldsymbol{x})$. The learning target function f is assumed to be unknown, but its input-output paired samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$ are observed. In practice, the observed output value y_i may be corrupted by some noise ϵ_i, i.e., $y_i = f(\boldsymbol{x}_i) + \epsilon_i$. In this setup, $\boldsymbol{x}_i$ corresponds to a question that a student asks the supervisor, and y_i corresponds to the answer that the supervisor gives to the student. Noise ϵ_i may correspond to the supervisor's mistake or



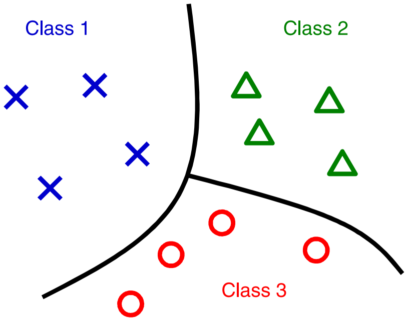


FIGURE 1.1

Regression.

FIGURE 1.2

Classification.

the student's misunderstanding. The learning target function f corresponds to the supervisor's knowledge, which allows him/her to answer any questions. The objective of regression is to let the student learn this function, by which he/she can also answer any questions. The level of generalization can be measured by the closeness between the true function f and its approximation $\widehat{f}$.

On the other hand, *classification* is a *pattern recognition* problem in a supervised manner (Fig. 1.2). Let us denote the input pattern by d-dimensional vector $\boldsymbol{x}$ and its class by a scalar $y \in \{1, \ldots, c\}$, where c denotes the number of classes. For training a classifier, input-output paired samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ are provided in the same way as regression. If the true classification rule is denoted by $y = f(\boldsymbol{x})$, classification can also be regarded as a function approximation problem. However, an essential difference is that there is no notion of closeness in y: $y = 2$ is closer to $y = 1$ than $y = 3$ in the case of regression, but whether y and y' are the same is the only concern in classification.

The problem of *ranking* in supervised learning is to learn the rank y of a sample $\boldsymbol{x}$. Since the rank has the order, such as $1 < 2 < 3$, ranking would be more similar to regression than classification. For this reason, the problem of ranking is also referred to as *ordinal regression*. However, different from regression, exact output value y is not necessary to be predicted, but only its relative value is needed. For example, suppose that "values" of three instances are 1, 2, and 3. Then, since only the ordinal relation $1 < 2 < 3$ is important in the ranking problem, predicting the values as $2 < 4 < 9$ is still a perfect solution.

1.2.2 UNSUPERVISED LEARNING

Clustering is an unsupervised counter part of classification (Fig. 1.3), and its objective is to categorize input samples $\{\boldsymbol{x}_i\}_{i=1}^n$ into clusters $1, 2, \ldots, c$ without any supervision $\{y_i\}_{i=1}^n$. Usually, similar samples are supposed to belong to the same cluster, and

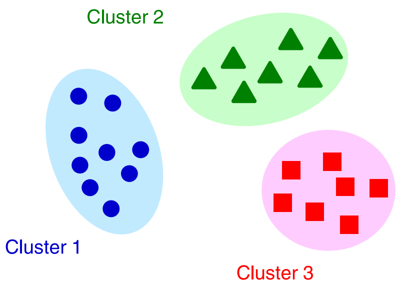


FIGURE 1.3

Clustering.

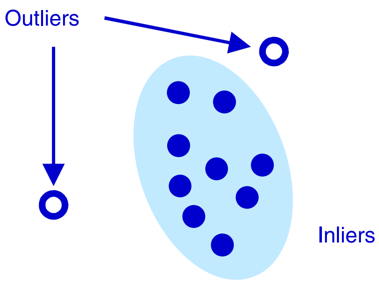


FIGURE 1.4

Outlier detection.

dissimilar samples are supposed to belong to different clusters. Thus, how to measure the *similarity* between samples is the key issue in clustering.

Outlier detection, which is also referred to as *anomaly detection*, is aimed at finding irregular samples in a given data set $\{\boldsymbol{x}_i\}_{i=1}^n$. In the same way as clustering, the definition of similarity between samples plays a central role in outlier detection, because samples that are dissimilar from others are usually regarded as outliers (Fig. 1.4).

The objective of *change detection*, which is also referred to as *novelty detection*, is to judge whether a newly given data set $\{\boldsymbol{x}'_{i'}\}_{i'=1}^{n'}$ has the same property as the original data set $\{\boldsymbol{x}_i\}_{i=1}^n$. Similarity between samples is utilized in outlier detection, while similarity between data sets is needed in change detection. If $n' = 1$, i.e., only a single point is provided for detecting change, the problem of change detection may be reduced to an outlier problem.

1.2.3 FURTHER TOPICS

In addition to supervised and unsupervised learnings, various useful techniques are available in machine learning.

Input-output paired samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ are used for training in supervised learning, while input-only samples $\{\boldsymbol{x}_i\}_{i=1}^n$ are utilized in unsupervised learning. In many supervised learning techniques, collecting input-only samples $\{\boldsymbol{x}_i\}_{i=1}^n$ may be easy, but obtaining output samples $\{y_i\}_{i=1}^n$ for $\{\boldsymbol{x}_i\}_{i=1}^n$ is laborious. In such a case, output samples may be collected only for $m \ll n$ input samples, and the remaining $n - m$ samples are input-only. *Semisupervised learning* is aimed at learning from both input-output paired samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^m$ and input-only samples $\{\boldsymbol{x}_i\}_{i=m+1}^n$. Typically, semisupervised learning methods extract distributional information such as cluster structure from the input-only samples $\{\boldsymbol{x}_i\}_{i=m+1}^n$ and utilize that information for improving supervised learning from input-output paired samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^m$.

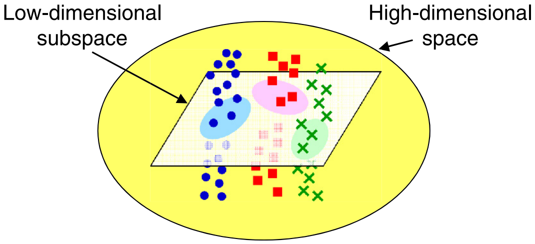


FIGURE 1.5

Dimensionality reduction.

Given weak learning algorithms that perform only slightly better than a random guess, *ensemble learning* is aimed at constructing a strong learning algorithm by combining such weak learning algorithms. One of the most popular approaches is voting by the weak learning algorithms, which may complement the weakness of each other.

Standard learning algorithms consider vectorial data $\boldsymbol{x}$. However, if data have a two-dimensional structure such as an image, directly learning from matrix data would be more promising than vectorizing the matrix data. Studies of *matrix learning* or *tensor learning* are directed to handle such higher-order data.

When data samples are provided sequentially one by one, updating the learning results to incorporate new data would be more natural than re-learning all data from scratch. *Online learning* is aimed at efficiently handling such sequentially given data.

When solving a learning task, transferring knowledge from other similar learning tasks would be helpful. Such a paradigm is called *transfer learning* or *domain adaptation*. If multiple related learning tasks need to be solved, solving them simultaneously would be more promising than solving them individually. This idea is called *multitask learning* and can be regarded as a bidirectional variant of transfer learning.

Learning from high-dimensional data is challenging, which is often referred to as the *curse of dimensionality*. The objective of *dimensionality reduction* is to extract essential information from high-dimensional data samples $\{\boldsymbol{x}_i\}_{i=1}^n$ and obtain their low-dimensional expressions $\{\boldsymbol{z}_i\}_{i=1}^n$ (Fig. 1.5). In linear dimensionality reduction, the low-dimensional expressions $\{\boldsymbol{z}_i\}_{i=1}^n$ are obtained as $\boldsymbol{z}_i = \boldsymbol{T}\boldsymbol{x}_i$ using a fat matrix $\boldsymbol{T}$. Supervised dimensionality reduction tries to find low-dimensional expressions $\{\boldsymbol{z}_i\}_{i=1}^n$ as preprocessing, so that the subsequent supervised learning tasks can be solved easily. On the other hand, unsupervised dimensionality reduction tries to find low-dimensional expressions $\{\boldsymbol{z}_i\}_{i=1}^n$ such that certain structure of original data is maintained, for example, for visualization purposes. *Metric learning* is similar to dimensionality reduction, but it has more emphasis on learning the metric in the original high-dimensional space rather than reducing the dimensionality of data.

1.3 STRUCTURE OF THIS TEXTBOOK

The main contents of this textbook consist of the following four parts.

Part 2 introduces fundamental concepts of statistics and probability, which will be extensively utilized in the subsequent chapters. Those who are familiar with statistics and probability or those who want to study machine learning immediately may skip Part 2.

Based on the concept of statistics and probability, Part 3, Part 4, and Part 5 introduce various machine learning techniques. These parts are rather independent, and therefore readers may start from any part based on their interests.

Part 3 targets the *generative approach* to statistical pattern recognition. The basic idea of generative methods is that the probability distribution of data is modeled to perform pattern recognition. When prior knowledge on data generation mechanisms is available, the generative approach is highly useful. Various classification and clustering techniques based on generative model estimation in the *frequentist* and *Bayesian* frameworks will be introduced.

Part 4 focuses on the *discriminative approach* to statistical machine learning. The basic idea of discriminative methods is to solve target machine learning tasks such as regression and classification directly without modeling data-generating probability distributions. If no prior knowledge is available for data generation mechanisms, the discriminative approach would be more promising than the generative approach. In addition to statistics and probability, knowledge of *optimization theory* also plays an important role in the discriminative approach, which will also be covered.

Part 5 is devoted to introducing various advanced issues in machine learning, including *ensemble learning*, *online learning*, *confidence of prediction*, *semisupervised learning*, *transfer learning*, *multitask learning*, *dimensionality reduction*, *clustering*, *outlier detection*, and *change detection*.

Compact MATLAB codes are provided for the methods introduced in Part 3, Part 4, and Part 5. So readers can immediately test the algorithms and learn their numerical behaviors.