CHAPTER

11 PATTERN RECOGNITION VIA GENERATIVE MODEL ESTIMATION

CHAPTER CONTENTS

In this chapter, the framework of pattern recognition based on generative model estimation is first explained. Then, criteria for quantitatively evaluating the goodness of a classification algorithm are discussed.

11.1 FORMULATION OF PATTERN RECOGNITION

In this section, the problem of statistical pattern recognition is mathematically formulated.

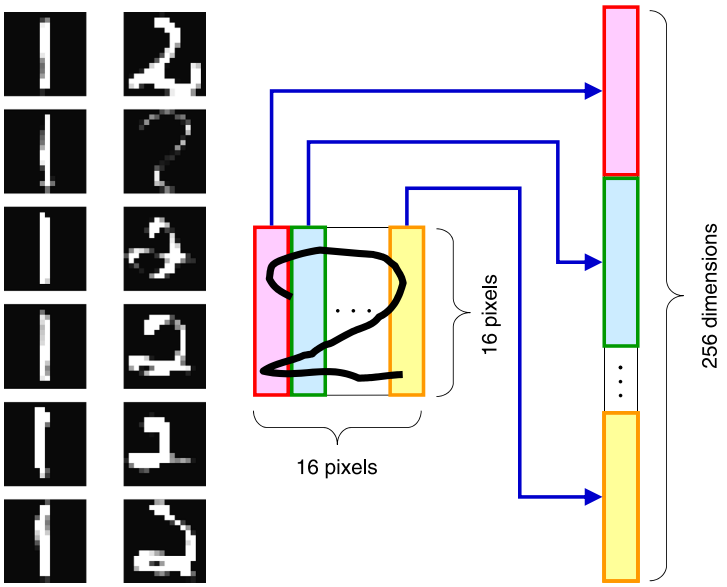
Let $\boldsymbol{x}$ be a *pattern* (which is also called a *feature vector*, an *input variable*, an *independent variable*, an *explanatory variable*, an *exogenous variable*, a *predictor variable*, a *regressor*, and a *covariate*), which is a member of a subset $\mathcal{X}$ of the d-dimensional Euclidean space $\mathbb{R}^d$:

$$\boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^d,$$

$\mathcal{X}$ is called the *pattern space*. Let y be a *class* (which is also called a *category*, an *output variable*, a *target variable*, and a *dependent variable*) to which a pattern $\boldsymbol{x}$ belongs. Let c be the number of classes, i.e.,

$$y \in \mathcal{Y} = \{1, \ldots, c\}.$$

An Introduction to Statistical Machine Learning. DOI: 10.1016/B978-0-12-802121-7.00022-4



(a) Hand-written digit image

(b) Vectorization of two-dimensional image

FIGURE 11.1

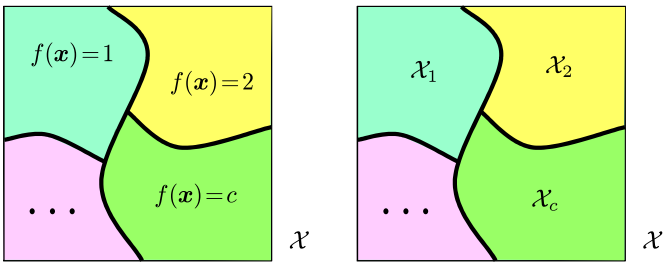
Hand-written digit image and its vectorization.

In hand-written digit recognition, a scanned digit image is a pattern. If the scanned image consists of 16×16 pixels, pattern $\boldsymbol{x}$ is a 256-dimensional real vector which vertically stacks the pixel values as illustrated in Fig. 11.1. Rigorously speaking, pixel values are integers (e.g., 0–255), but they are regarded as real numbers here. When the pixel values are normalized to be in $[0, 1]$, the pattern space is given by $\mathcal{X} = [0, 1]^{256}$. Classes are the numbers "0," "1," ..., "9," and thus the number of classes is $c = 10$.

A classifier is a mapping from a pattern $\boldsymbol{x}$ to a class y. Such a mapping is called a *discrimination function* (see Fig. 11.2(a)) and is denoted by $f(\boldsymbol{x})$. A region to which patterns in class y belong is called a *decision region* (see Fig. 11.2(b)) and is denoted by $\mathcal{X}_y$. A boundary between decision regions is called a *decision boundary*. Thus, pattern recognition is equivalent to dividing the pattern space $\mathcal{X}$ into decision regions $\{\mathcal{X}_y\}_{y=1}^{c}$.

In practice, the discrimination function (or decision regions or decision boundaries) is unknown. Here, pattern $\boldsymbol{x}$ and class y are treated as *random variables* and learn the optimal discrimination function based on their statistical properties. Such an approach is called *statistical pattern recognition*.

Let us illustrate how hard directly constructing a discrimination function (or decision regions and decision boundaries) in hand-written digit recognition. Let the



(a) Discrimination function

(b) Decision region and decision boundary

FIGURE 11.2

Constructing a classifier is equivalent to determine a discrimination function, decision regions, and decision boundaries.

number of pixels be 100 (=10 × 10) and each pixel takes an integer from 0 to 255. Then the number of possible images is

$$256^{100} = (2^8)^{100} = (2^{10})^{80} \approx (10^3)^{80} = 10^{240},$$

which is an astronomical number having 240 zeros after the first one. Therefore, even for a toy hand-written digit recognition example from tiny images with 10×10 pixels, just enumerating all possible images is not realistic. In practice, instead of just memorizing classes of all possible patterns, the class of unlearned patterns may be predicted from some learned patterns. The capability that unlearned patterns can be classified correctly is called the *generalization ability*. The objective of pattern recognition is to let a classifier being equipped with the generalization ability.

11.2 STATISTICAL PATTERN RECOGNITION

In this section, a statistical approach to pattern recognition is explained.

Suppose that pairs of patterns and their classes, called *training samples*, are available:

$$\{(\boldsymbol{x}_i, y_i) \mid \boldsymbol{x}_i \in \mathcal{X}, y_i \in \mathcal{Y}\}_{i=1}^{n},$$

where n denotes the number of training samples. Among the n training samples, the number of samples which belong to class y is denoted by n_y. Below, the training samples are assumed to be generated for $i = 1, \ldots, n$ as the following:

1. Class y_i is selected according to the *class-prior probability* $p(y)$.
2. For chosen class y_i, pattern $\boldsymbol{x}_i$ is generated according to the *class-conditional probability density* $p(\boldsymbol{x}|y = y_i)$.

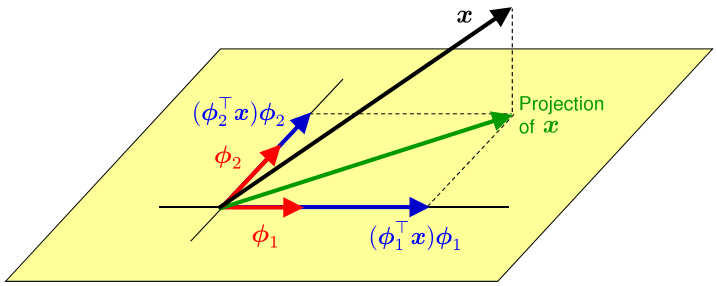


FIGURE 11.3

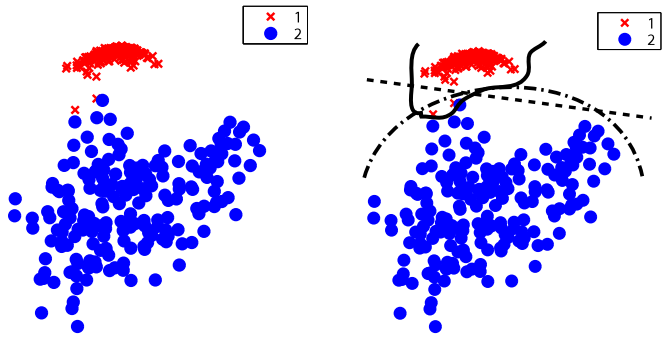
Dimensionality reduction onto a two-dimensional subspace by principal component analysis (see Section 35.2.1).

Then training samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ independently follow the joint probability density $p(\boldsymbol{x}, y)$, i.e., $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ are i.i.d. with $p(\boldsymbol{x}, y)$ (see Section 7.3). This i.i.d. assumption is one of the most fundamental presuppositions in statistical pattern recognition. Machine learning techniques when this i.i.d. assumption is violated are discussed in Chapter 33; see also [81, 101].

Given training samples, what is the best way to learn the discrimination function? Let us illustrate the distribution of patterns for the hand-written digit data shown in Fig. 11.1. Since the hand-written digit samples are 256-dimensional, their distribution cannot be directly visualized. Here, a dimensionality reduction method called *principal component analysis* (PCA) is used to reduce the dimensionality from 256 to 2. More specifically, the variance-covariance matrix of training samples $\{\boldsymbol{x}_i\}_{i=1}^n$ is eigendecomposed (see Fig. 6.2), and eigenvalues $\lambda_1 \geq \cdots \geq \lambda_d$ and corresponding eigenvectors $\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_d$ are obtained. Then each training sample $\boldsymbol{x}_i$ is transformed as $(\boldsymbol{\phi}_1^\top \boldsymbol{x}_i, \boldsymbol{\phi}_2^\top \boldsymbol{x}_i)^\top$, where the eigenvectors $\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_d$ are assumed to be normalized to have unit norm. Since $\boldsymbol{\phi}_j^\top \boldsymbol{x}_i$ corresponds to the length of projection of $\boldsymbol{x}_i$ along $\boldsymbol{\phi}_j$, the above transformation is the projection of $\boldsymbol{x}_i$ onto the subspace spanned by $\boldsymbol{\phi}_1$ and $\boldsymbol{\phi}_2$ (Fig. 11.3). As detailed in Section 35.2.1, PCA gives the best approximation to original data in a lower-dimensional subspace, and therefore it is often used for *data visualization*.

The PCA projection of the hand-written digit data shown in Fig. 11.1 is plotted in Fig. 11.4(a), showing that digit "2" is distributed more broadly than digit "1." This well agrees with the intuition that the shape of "2" may have more individuality than the shape of "1."

What is the best decision boundary for the samples plotted in Fig. 11.4(a)? Examples of decision boundaries are illustrated in Fig. 11.4(b). The decision boundary shown by the solid line is a complicated curve, but it can perfectly separate "1" and "2." On the other hand, the decision boundaries shown by the dashed line and dashed-dotted line are much simpler, but some training samples are classified



(a) Distribution of hand-written digits (b) Examples of decision boundary

FIGURE 11.4

Illustration of hand-written digit samples in the pattern space.

incorrectly. For the purpose of classifying training samples, the decision boundary shown by the solid line is better than those shown by the dashed line and dashed-dotted line. However, the true objective of pattern recognition is not only to classify training samples correctly but also to classify unlearned test samples given in the future, i.e., to acquire the generalization ability, as mentioned in Section 11.1.

11.3 CRITERIA FOR CLASSIFIER TRAINING

In order to equip a classifier with the generalization ability, it is important to define a criterion that quantitatively evaluate the goodness of a discrimination function (or decision regions or decision boundaries). In this section, three examples of such criteria are introduced and their relation is discussed.

11.3.1 MAP RULE

When deciding which class a given pattern belongs to, it would be natural to choose the one with the highest probability. This corresponds to choosing the class that maximizes the *class-posterior probability* $p(y|\boldsymbol{x})$, i.e., pattern $\boldsymbol{x}$ is classified into class $\widehat{y}$, where

$$\widehat{y} = \operatorname*{argmax}_{y} p(y|\boldsymbol{x}).$$

Here, "argmax" indicates *the argument of the maximum*, i.e., the maximizer of an objective function. Such a decision rule is called the MAP rule. The MAP rule is equivalent to setting the decision regions as follows (Fig. 11.5):

$$\mathcal{X}_y = \{\boldsymbol{x} \mid p(y|\boldsymbol{x}) \geq p(y'|\boldsymbol{x}) \text{ for all } y' \neq y\}. \tag{11.1}$$

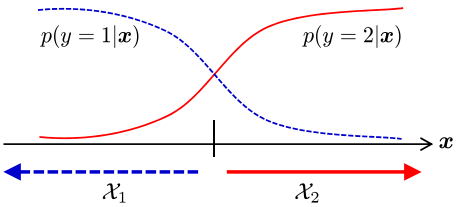


FIGURE 11.5

MAP rule.

11.3.2 MINIMUM MISCLASSIFICATION RATE RULE

Another natural idea is to choose the class with the lowest misclassification error, which is called the *minimum misclassification rate rule*.

Let $p_e(y \to y')$ be the probability that a pattern in class y is misclassified into class y'. Since $p_e(y \to y')$ is equivalent to the probability that pattern $\boldsymbol{x}$ in class y falls into decision region $\mathcal{X}_{y'}$ (see Fig. 11.6), it is given by

$$p_e(y \to y') = \int_{\boldsymbol{x} \in \mathcal{X}_{y'}} p(\boldsymbol{x}|y) \mathrm{d}\boldsymbol{x}.$$

Then the probability that a pattern in class y is classified into an incorrect class, denoted by $p_e(y)$, is given as

$$\begin{aligned} p_e(y) &= \sum_{y' \neq y} p_e(y \to y') = \sum_{y' \neq y} \int_{\boldsymbol{x} \in \mathcal{X}_{y'}} p(\boldsymbol{x}|y) \mathrm{d}\boldsymbol{x} \\ &= \sum_{y' \neq y} \int_{\boldsymbol{x} \in \mathcal{X}_{y'}} p(\boldsymbol{x}|y) \mathrm{d}\boldsymbol{x} + \int_{\boldsymbol{x} \in \mathcal{X}_y} p(\boldsymbol{x}|y) \mathrm{d}\boldsymbol{x} - \int_{\boldsymbol{x} \in \mathcal{X}_y} p(\boldsymbol{x}|y) \mathrm{d}\boldsymbol{x} \\ &= 1 - \int_{\boldsymbol{x} \in \mathcal{X}_y} p(\boldsymbol{x}|y) \mathrm{d}\boldsymbol{x}. \end{aligned}$$

The second term in the last equation,

$$\int_{\boldsymbol{x} \in \mathcal{X}_y} p(\boldsymbol{x}|y) \mathrm{d}\boldsymbol{x},$$

denotes the probability that a pattern in class y is classified into class y, i.e., the correct classification rate. Thus, the above equation shows the common fact that the misclassification rate is given by one minus the correct classification rate.

Finally, the overall misclassification rate, denoted by p_e, is given by the expectation of $p_e(y)$ over all classes:

$$p_e = \sum_{y=1}^{c} p_e(y) p(y).$$

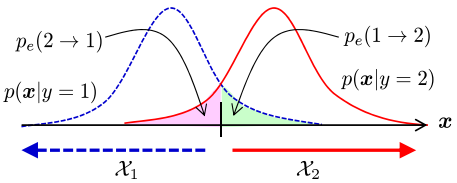


FIGURE 11.6

Minimum misclassification rate rule.

The minimum misclassification rate rule finds the classifier that minimizes the above p_e.

The overall misclassification rate p_e can be expressed as

$$\begin{aligned} p_e &= \sum_{y=1}^{c} \left(1 - \int_{\boldsymbol{x}\in\mathcal{X}_y} p(\boldsymbol{x}|y)\mathrm{d}\boldsymbol{x}\right) p(y) \\ &= \sum_{y=1}^{c} p(y) - \sum_{y=1}^{c} \int_{\boldsymbol{x}\in\mathcal{X}_y} p(\boldsymbol{x}|y)p(y)\mathrm{d}\boldsymbol{x} \\ &= 1 - \sum_{y=1}^{c} \int_{\boldsymbol{x}\in\mathcal{X}_y} p(y,\boldsymbol{x})\mathrm{d}\boldsymbol{x} = 1 - \sum_{y=1}^{c} \int_{\boldsymbol{x}\in\mathcal{X}_y} p(y|\boldsymbol{x})p(\boldsymbol{x})\mathrm{d}\boldsymbol{x}. \end{aligned}$$

Thus, minimizing p_e is equivalent to determining the decision regions $\{\mathcal{X}_y\}_{y=1}^{c}$ so that the second term, $\sum_{y=1}^{c} \int_{\boldsymbol{x}\in\mathcal{X}_y} p(y|\boldsymbol{x})p(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$, is maximized. This can be achieved by setting $\mathcal{X}_y$ to be the set of all $\boldsymbol{x}$ such that

$$p(y|\boldsymbol{x}) \geq p(y'|\boldsymbol{x}) \quad \text{for all } y' \neq y.$$

This is actually equivalent to Eq. (11.1), and therefore minimizing the misclassification error is actually equivalent to maximizing the class-posterior probability.

11.3.3 BAYES DECISION RULE

According to the MAP rule (equivalently the minimum misclassification rate rule), when the probability of precipitation is 40%, the anticipated weather is no rain. If there will be no rain, then there is no need to carry an umbrella. However, perhaps many people will bring an umbrella with them if the probability of precipitation is 40%. This is because, the loss of no rain when an umbrella is carried (i.e., need to carry a slightly heavier bag) is much smaller than the loss of having rain when no umbrella is carried (i.e., getting wet with rain and catching a cold) in reality (see Table 11.1). In this way, choosing the class that has the smallest loss is called the *Bayes decision rule*.

Table 11.1 Example Of Asymmetric Loss

	Carry An Umbrella	Leave An Umbrella At Home
Rain	Avoid get wet	Get wet and catch cold
No rain	Bag is heavy	Bag is light

Let $\ell_{y,y'}$ be the *loss* that a pattern in class y is misclassified into class y'. Since the probability that pattern $\boldsymbol{x}$ belongs to class y is given by the class-posterior probability $p(y|\boldsymbol{x})$, the expected loss for classifying pattern $\boldsymbol{x}$ into class y' is given by

$$R(y'|\boldsymbol{x}) = \sum_{y=1}^{c} \ell_{y,y'} p(y|\boldsymbol{x}).$$

This is called the *conditional risk* for pattern $\boldsymbol{x}$.

In the Bayes decision rule, pattern $\boldsymbol{x}$ is classified into the class that incurs the minimum conditional risk. More specifically, pattern $\boldsymbol{x}$ is classified into class $\widehat{y}$, where

$$\widehat{y} = \operatorname*{argmin}_{y} R(y|\boldsymbol{x}).$$

This is equivalent to determining the decision regions $\{\mathcal{X}_y\}_{y=1}^{c}$ as

$$\mathcal{X}_y = \{\boldsymbol{x} \mid R(y|\boldsymbol{x}) \leq R(y'|\boldsymbol{x}) \text{ for all } y' \neq y\}.$$

The expectation of the conditional risk for all $\boldsymbol{x}$ is called the *total risk*:

$$R = \int_{\mathcal{X}} R(\widehat{y}|\boldsymbol{x}) p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x},$$

where $\widehat{y}$ is an output of a classifier. The value of the total risk for the Bayes decision rule is called the *Bayes risk*, and this is the lowest possible risk for the target classification problem. Note that the Bayes risk is not zero in general, meaning that the risk cannot be zero even with the optimally trained classifier.

Suppose the loss for correct classification is set at $zero$ and the loss for incorrect classification is set at a positive constant ℓ:

$$\ell_{y,y'} = \begin{cases} 0 & (y = y'), \\ \ell & (y \neq y'). \end{cases} \tag{11.2}$$

Then the conditional risk is expressed as

$$R(y|\boldsymbol{x}) = \ell \sum_{y' \neq y} p(y'|\boldsymbol{x}) = \ell \left(\sum_{y'=1}^{c} p(y'|\boldsymbol{x}) - p(y|\boldsymbol{x}) \right) = \ell \left(1 - p(y|\boldsymbol{x})\right). \tag{11.3}$$

Since ℓ is just a proportional constant, minimization of Eq. (11.3) is equivalent to maximization of class-posterior probability $p(y|\boldsymbol{x})$. Thus, when loss $\ell_{y,y'}$ is given by Eq. (11.2), the Bayes decision rule is reduced to the MAP rule (and therefore the minimum misclassification rate rule, too).

11.3.4 DISCUSSION

Among the MAP rule, the minimum misclassification rate rule, and the Bayes decision rule, the Bayes decision rule seems to be natural and the most powerful. However, in practice, it is often difficult to precisely determine the loss $\ell_{y,y'}$, which makes the use of the Bayes decision rule not straightforward. For example, in the rain-umbrella example described in Table 11.1, it would be clear that the loss of no rain when an umbrella is carried is much smaller than the loss of having rain when no umbrella is carried. However, it is not immediately clear how small the loss of no rain when an umbrella is carried should be.

For this reason, in the following sections, we focus on the MAP rule (and therefore the minimum misclassification rate rule, too).

11.4 GENERATIVE AND DISCRIMINATIVE APPROACHES

Learning a classifier based on the MAP rule requires to find a maximizer of the class-posterior probability $p(y|\boldsymbol{x})$. Pattern recognition through estimation of the class-posterior probability $p(y|\boldsymbol{x})$ is called the *discriminative approach* and will be covered in Part 4.

Another approach is to use the Bayes' theorem explained in Section 5.4 to express the class-posterior probability $p(y|\boldsymbol{x})$ as

$$p(y|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|y)p(y)}{p(\boldsymbol{x})}.$$

Since the denominator in the right-hand side, $p(\boldsymbol{x})$, is independent of class y, it can be ignored when the class-posterior probability $p(y|\boldsymbol{x})$ is maximized with respect to y:

$$p(y|\boldsymbol{x}) \propto p(\boldsymbol{x}|y)p(y),$$

where "$\propto$" means "proportional to." Statistical pattern recognition through estimation of the *class-conditional probability density* $p(\boldsymbol{x}|y)$ and the *class-prior probability* $p(y)$ is called the *generative approach* since

$$p(\boldsymbol{x}|y)p(y) = p(\boldsymbol{x}, y),$$

which is the data-generating probability distribution.

In the following chapters, the generative approach to statistical pattern recognition is explored. The class-prior probability $p(y)$ may simply be estimated by the ratio of

training samples in class y, i.e.,

$$\widehat{p}(y) = \frac{n_y}{n}, \tag{11.4}$$

where n_y denotes the number of training samples in class y and n denotes the number of all training samples. On the other hand, the class-conditional probability $p(\boldsymbol{x}|y)$ is generally a high-dimensional probability density function, and therefore its estimation is not straightforward. In the following chapters, various approaches to estimating the class-conditional probability $p(\boldsymbol{x}|y)$ will be discussed.

For the sake of simplicity, the problem of estimating an unconditional probability density $p(\boldsymbol{x})$ from its i.i.d. training samples $\{\boldsymbol{x}_i\}_{i=1}^n$ is considered in the following chapters, because this allows us to estimate a conditional probability density $p(\boldsymbol{x}|y)$ by only using n_y samples in class y, $\{\boldsymbol{x}_i\}_{i:y_i=y}$, for density estimation.

Methods of probability density functions are categorized into *parametric* and *nonparametric* methods. The parametric methods seek the best approximation to the true probability density function from a parameterized family of probability density functions, called a parametric model. For example, the Gaussian model contains the mean vector and the variance-covariance matrix as parameters and they are estimated from training samples. Once a parametric model is considered, the problem of estimating a probability density function is reduced to the problem of learning a parameter in the model. On the other hand, methods that do not use parametric models are called nonparametric.

In the following chapters, various parametric and nonparametric methods will be introduced.