Chapter 2
Classification Technique for HSI

Classification is one of the most basic and most important research contents of the hyperspectral data processing (Richards and Jia 2006). Classification is an analytical technique of describing the land object target or class, with the main task of a process of giving a class mark to each pixel point of the data volume to generate the thematic map. It is one of the important ways for people to extract the useful information from the remote-sensing image. The thematic map upon classification can clearly reflect the spatial distribution of the land objects, so that people can know and discover the rules and the hyperspectral remote-sensing image possesses the real use value and is effectively put into the practical application. After introducing several typical classification methods and evaluation criteria, this chapter focuses on the burgeoning SVM (Vapnik 2000)-based classification method.

2.1 Typical Classification Methods

1. **Spectral angle match**

Spectral Angle Match (SAM) (Sohn and Rebello 2002) is an angle-based hyperspectral image classification method. It automatically compares the image spectrum with various spectrums or spectral libraries. According to the physical basis of remote sensing, the reflection spectrum of the land objects can determine the land object types to a great extent, and accordingly lead out SAM classification. Thereby finishing the transformation from the measurement space to the eigenspace through mapping the measurement spectral vector into a series of angle values on behalf of the similarity of this vector and the reference spectral vector. Calculating the spectral angle between the two spectrums can determine the level of their similarity. The dimension of the spectral vector is the number of bands. The similarity α between the unknown spectrum $\boldsymbol{t}$ and reference spectrum $\boldsymbol{r}$ shall be determined by the following formula.

$$\alpha = \cos^{-1} \frac{\langle \boldsymbol{t}, \boldsymbol{r} \rangle}{\|\boldsymbol{t}\| \cdot \|\boldsymbol{r}\|} \tag{2.1}$$

L. Wang and C. Zhao, *Hyperspectral Image Processing*,
DOI 10.1007/978-3-662-47456-3_2

With the standard spectrum measured in the laboratory or the average spectrum of the known point directly extracted from the image for reference, take the generalized included angle α for each pixel vector and reference the spectral vector in the image. The smaller α is, the greater the level of their similarity is. In the general application, usually select the known type of the area from the image, classify with the sample center of the average spectrum, take the included angle between each pixel of the image and the center of each class, and then put the pixel into the class of the corresponding minimum included angle.

2. **Maximal likelihood classification**

Maximal likelihood (ML), also known as Bayes code (Chen and Tu 1996; Jia and Richards 1994), discrimination function is the parametric method of the statistical pattern recognition. This method should employ various a priori probabilities $p(\omega_i)$ and conditional probability density functions $p(\boldsymbol{X}/\omega_i)$. The priori probabilities $p(\omega_i)$ are generally given in accordance with all kinds of a priori knowledge (practical situation of the specific problems, and information accumulated in history etc.) or are supposed to be equal. Moreover, $p(\boldsymbol{X}/\omega_i)$ firstly determines the distribution form, and then estimates the parameters used in this form by the training field. The estimation of the distribution form has multiple methods such as the maximum entropy method and the polynominal method. In the remote-sensing problem, the assumption of the normal distribution is reasonable, i.e., some non-normal problems can be converted into the normal problems for handling in the mathematical method.

Given $p(\boldsymbol{X}/\omega_i)$ is the $i(i = 1, 2, \ldots, N)$ class of the probability density function in d dimension characteristic data space, and $p(\omega_i)$ is the i class of the occurrence probability in the data set, the decision $\boldsymbol{X}$ is ω_i class, rather than ω_j class, equivalent to

$$p(\boldsymbol{X}/\omega_i)p(\omega_i) \geq p(\boldsymbol{X}/\omega_j)p(\omega_j) \tag{2.2}$$

In the practical application, the probability density function is often assumed to be the normal or Gaussian distribution, and then the class probability density function is expressed as

$$p(\boldsymbol{X}/\omega_i) = \frac{1}{(2\pi)^{N/2}|\boldsymbol{\Sigma}_i|^{1/2}} \exp\left[-\frac{1}{2}(\boldsymbol{X} - \boldsymbol{\mu}_i)^{\mathrm{T}}\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{X} - \boldsymbol{\mu}_i)\right] \tag{2.3}$$

where, $\boldsymbol{\mu}_i$ is the class mean value vector, and $\boldsymbol{\Sigma}_i$ is the covariance matrix. In this case, the class mean value vector and the class covariance matrix can be estimated only if selecting the appropriate samples.

If the assumption of Gaussian distribution is correct, the decision function can be simplified further. Then, for all $j = 1, 2, \ldots, N$, if the formula (2.2) is correct, then

$$\ln[p(\boldsymbol{X}/\omega_i)p(\omega_i)] \geq \ln\left[p(\boldsymbol{X}/\omega_j)p(\omega_j)\right] \tag{2.4}$$

It is also correct, so the decision function can be expressed as

$$g(\boldsymbol{X}) = \ln[p(\omega_i)] - \frac{1}{2}\ln|\boldsymbol{\Sigma}_i| - \frac{1}{2}(\boldsymbol{X}-\boldsymbol{\mu}_i)^{\mathrm{T}}\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{X}-\boldsymbol{\mu}_i) \tag{2.5}$$

We make the classification and recognition mainly in accordance with the decision criterion in the above formula.

3. **Fisher discriminant analysis**

Fisher discriminant analysis is a supervised classification method, and the main idea is to conduct the linear combination for the multivariate observed value to set up the new discriminant amount, and maximize the specific value of the inter-class variance and intra-class variance of the new discriminant amount.

Given unmixing the Nc classes, and each class has Ntr_i training samples and the number of bands for each training sample are ND, so the training sample of each class constitute a matrix of ND $\times$ Ntr_i. Meanwhile, given the training samples are expressed by $x_1, x_2, x_3, \ldots, x_{\mathrm{Ntr}_i}$.

Each sample mean $\overline{\boldsymbol{m}_i}$ shows the mean value of each sample.

$$\overline{\boldsymbol{m}_i} = \frac{\sum_{p=1}^{\mathrm{Ntr}_i}\boldsymbol{x_p}}{\mathrm{Ntr}_i} \quad i = 1, 2, 3, \ldots, \mathrm{Nc} \tag{2.6}$$

The inter-class sample mean $\bar{\boldsymbol{m}}$ shows the total mean value of all samples.

$$\bar{\boldsymbol{m}} = \frac{\sum_{q=1}^{\mathrm{Nc}}\sum_{p=1}^{\mathrm{Ntr}_i}\boldsymbol{y_{qp}}}{\sum \mathrm{Ntr}_i} \quad i = 1, 2, 3, \ldots, \mathrm{Nc} \tag{2.7}$$

In formula (2.7), $\boldsymbol{y_{qp}}$ shows the p training sample in the q class.

For the sample intra-class dispersion matrix $\boldsymbol{S_i}$ and total intra-class dispersion matrix $\boldsymbol{S}_w$, $\boldsymbol{S_i}$ stands for the internal differences of the i class training, while $\boldsymbol{S}_w$ stands for the total internal differences of all training samples.

$$\boldsymbol{S_i} = \sum_{p=1}^{\mathrm{Ntr}_i}(\boldsymbol{y_{ip}} - \overline{\boldsymbol{m}_i})(\boldsymbol{y_{ip}} - \overline{\boldsymbol{m}_i})^{\mathrm{T}} \quad i = 1, 2, 3, \ldots, \mathrm{Nc} \tag{2.8}$$

$$\boldsymbol{S}_w = \sum \boldsymbol{S_i} \quad i = 1, 2, 3, \ldots, \mathrm{Nc} \tag{2.9}$$

The inter-class dispersion matrix $\boldsymbol{S_b}$ stands for the total inter-class dispersion, which can stand for the differences among all classes.

$$\boldsymbol{S_b} = \sum_{\mathrm{i}=1}^{\mathrm{Nc}} (\overline{\boldsymbol{m}_i} - \bar{\boldsymbol{m}})(\overline{\boldsymbol{m}_i} - \bar{\boldsymbol{m}})^{\mathrm{T}} \tag{2.10}$$

Considering the linear combination

$$y = \boldsymbol{U}\boldsymbol{x} \tag{2.11}$$

where, $\boldsymbol{U}$ is the matrix of $1 \times \mathrm{ND}$, indicating some linear combination operation on the original spectrum. Then the degree of separation upon the matrix $\boldsymbol{U}$ transformation is:

$$J = \frac{\boldsymbol{U}\boldsymbol{S_b}\boldsymbol{U}^{\mathrm{T}}}{\boldsymbol{U}\boldsymbol{S_w}\boldsymbol{U}^{\mathrm{T}}} \tag{2.12}$$

In the formula (2.12), J is the dispersion. When $\boldsymbol{U}$ maximizes J, $\boldsymbol{U}$ will minimize the intra-class distance of the sample and maximize the inter-class distance. Thus, such $\boldsymbol{U}$ is the linear transformation which we seek. Such $\boldsymbol{U}$ can be obtained by calculating the characteristic vector of $\boldsymbol{S}_w^{-1}\boldsymbol{S_b}$. The characteristic vector enabling J to gain the maximum vector is called the first discriminant vector, and the characteristic vector enabling J to gain the second is called the second characteristic vector. Similarly, we can gain multiple discriminant vectors, and distinguish Nc classes to find the $\mathrm{Nc} - 1$ discriminant vector.

2.2 Typical Assessment Criterions

The pixel level-based accuracy assessment of the hyperspectral image classification results is obtained on the basis of the classification confusion matrix. The form of the classification confusion matrix is as follows:

$$\boldsymbol{M} = \begin{bmatrix} m_{11} & m_{12} & \cdots & m_{1\mathrm{Nc}} \\ m_{21} & m_{22} & \cdots & m_{2\mathrm{Nc}} \\ & \cdots & \cdots & \\ m_{\mathrm{Nc}1} & m_{\mathrm{Nc}2} & \cdots & m_{\mathrm{NcNc}} \end{bmatrix} \tag{2.13}$$

where, m_{ij} shows the number of pixels that should belong to the i class sample in the experimental area and are assigned to the j class, and Nc is the number of classification classes. In the confusion matrix, the larger the element numerical value of the diagonal is, the higher the reliability of the classification results is. On the contrary, it shows the more serious phenomena of the mis-classification.

According to the classification confusion matrix, it can calculate the overall accuracy OA, user's accuracy CA_i_user, and producer's accuracy CA_i_producer:

$$\begin{aligned} \mathrm{OA} &= \frac{1}{\mathrm{Nte}} \sum_{i=1}^{\mathrm{Nc}} m_{ii}, \\ \mathrm{CA}_i\_\mathrm{user} &= \frac{m_{ii}}{\mathrm{Nte}_i}, \\ \mathrm{CA}_i\_\mathrm{producer} &= \frac{m_{ii}}{\mathrm{Nte}_i^*}, \quad i = 1, 2, \ldots, \mathrm{Nc} \end{aligned} \tag{2.14}$$

In the formula, Nte is the total test samples, Nte_i is the total test samples of the i class, and Nte_i^* is the total pixels assigned to the i class. Thus, m_{ii} is the number of samples of the i class correct classification. When the number of various samples is equal, the overall accuracy is equivalent to the average user's accuracy.

The other accuracy analytical method is to make the quantification assessment for the overall effective performance of the classifier on the basis of the classification confusion matrix, and the most common one is Kappa coefficient.

$$\mathrm{Kappa} = \frac{\mathrm{Nte} \sum_{i=1}^{\mathrm{Nc}} m_{ii} - \sum_{i=1}^{\mathrm{Nc}} m_{i+} m_{+i}}{\mathrm{Nte}^2 - \sum_{i=1}^{\mathrm{Nc}} m_{i+} m_{+i}} \tag{2.15}$$

where, + shows the summation of the line or row, and Nte is the total test samples. Thus, this calculation uses each element in the classification matrix. The significance of Kappa coefficient can be interpreted as, if Kappa value of the classification result is 0.8, the classification method should be superior to 80 % of the method which randomly gives each point to a class. In the practical application, it often adopts the following form:

$$\mathrm{Kappa} = \frac{\theta_1 - \theta_2}{1 - \theta_2} \tag{2.16}$$

where,

$$\theta_1 = \frac{\sum_{i=1}^{\mathrm{Nc}} m_{ii}}{\mathrm{Nte}}, \theta_2 = \frac{\sum_{i=1}^{\mathrm{Nc}} m_{i+} m_{+i}}{\mathrm{Nte}^2}.$$

This book mainly adopts the overall accuracy to express the classification results, and if necessary, makes assessment by the user's accuracy of each class (referred to as classification accuracy of each class) and the mean value (referred to as the mean classification accuracy).

2.3 SVM-Based Classification Method

SVM is the new generation machine learning theory developed on the basis of the statistical learning theory, and seeks the best compromise between the model complexity and learning ability in accordance with the limited sample information, in the hope of gaining the best generalization ability.

2.3.1 Theory Foundation

SVM is the epitome of several standard techniques in the field of the machine learning, which integrates multiple techniques such as maximum margin hyperplane, Mercer kernel, convex quadratic programming, sparse solution, and slack variable. Here, we will introduce the important theoretical basis of SVM—VC dimension theory, and the footstone of SVM algorithm—structural risk minimization principle.

1. **VC dimension** (Karpinski and Werther 1989)

The statistical learning theory defines a series of the performance indexes relevant to the function set learning, and the most important one is the VC dimension. VC dimension of an indication function set $Q(z, \boldsymbol{\alpha}), \boldsymbol{\alpha} \in \Lambda$ means the maximum number h that can be divided into two classes of the vectors $z_1, z_2, \ldots, z_h$ in all possible 2^h modes by the centralized function of the function, i.e., the maximum number of the vector that can be scattered by the function set. If the set of an n vector for any natural number n can be scattered by the function set $Q(z, \boldsymbol{\alpha}), \boldsymbol{\alpha} \in \Lambda$, VC dimension of the function set is infinity. Illustrate VC dimension below.

VC dimension of the linear indication function set $Q(z, \boldsymbol{\alpha}) = \sum_{i=1}^{d} \alpha_i z_i + \alpha_0$, $\alpha_0, \ldots, \alpha_d \in (-\infty, \infty)$ in d dimension coordinate space $Z = \{z_1, z_2, \ldots, z_d\}$ is $d + 1$, because the function in this set can scatter $d + 1$ vectors to the maximum. VC dimension reflects the learning ability of the function set. The larger the VC dimension, the more complex the learning machine is. As shown in Fig. 2.1, the linear function set of two-dimensional space can make eight possible second-class classifications for three data points, and VC dimension is 3.

2. **Structural risk minimization principle**

The statistical learning theory systematically researches the relationship between the empirical risk and actual risk for various kinds of the function sets, i.e., the promotional boundary. In regard to two classification problems, for all functions in the indication function set, the empirical risk $R_{\text{emp}}(\alpha)$ and actual risk $R(\alpha)$ meet the following relations by at least $1 - \eta$ probability.

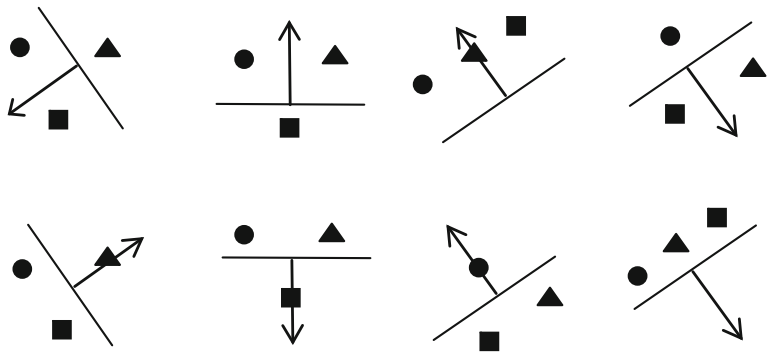

Fig. 2.1 2 VC dimension of dimensional space linear function set

$$R(\alpha) \leq R_{emp}(\alpha) + \sqrt{\frac{h[\ln(2n/h) + 1 - \ln(\eta/4)]}{n}} \tag{2.17}$$

where, h is the VC dimension of the function set, and n is the number of samples.

This conclusion in theory indicates that the actual risk of the learning machine is composed of two sections, involving the empirical risk (training error) and confidence risk (VC confidence), and reflects the generalization ability of the learning machine gained in accordance with the empirical risk minimization principle, so-called as the promotional boundary. It shows that, under the limited training sample, the higher VC dimension of the learning machine is, the greater the confidence risk is. Accordingly, it results in the larger differences between the actual risk and the empirical risk. It is the reason that it has the overfitting phenomenon.

In the traditional method, the process of selecting the learning model and algorithm is the process of adjusting the confident range. If the model is relatively applicable to the existing training sample, it can gain the preferable effect. However, because of lack of theoretical guidance, such selection can only depend on the priori knowledge and experience, and result in the overdependence of the neural network and other methods on the user's skills. When the training sample is applicable to the existing model, the expected risk is close to the value of the empirical risk. In this case, the smaller empirical risk can guarantee the smaller expected risk. For the sample with the number of n, if the specific value n/h is smaller (generally subject to 20 times), we deem that the number of the samples is small, i.e., we deem that the sample set is the small sample. If the training sample is the small sample, a small empirical risk value cannot guarantee the small expected risk value. In this case, to minimize the actual risk value, the learning ability (VC dimension) of the function set must become a controllable variable. The statistical learning theory puts forward a new strategy, i.e., constructing the function set into a function subset sequence, making each subset arrange in accordance with the size of VC dimension, seeking the minimum empirical risk, in each subset, compromising the consideration of the

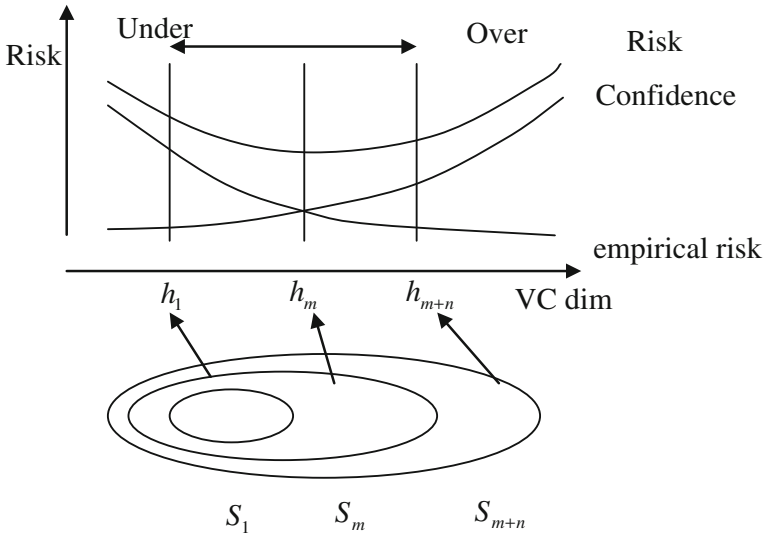

Fig. 2.2 Schematic diagram of structural risk minimization

empirical risk and the confidence risk among the subsets, and obtaining the minimum of the actual risk. This idea is called the structural risk minimization (SRM), as shown in Fig. 2.2.

2.3.2 Classification Principle

The original SVM theory is used for handling two types of classification problems. The classification principle can be summarized as seeking a classification hyperplane, enabling two types of sample points in the training sample to be separated, and having the distance from the plane as far as possible. While for the problems of the linear inseparability, we map the data of the low-dimensional input space into the high-dimensional space through the kernel function, and accordingly transform the linear inseparability problems of the original low-dimensional space into the linear separability problem of the high-dimensional space. Before the specific introduction, we firstly provide the basic definition and theorem in several optimization theories.

Define the original problem in the domain $\Omega \subseteq R^n$:

$$\begin{aligned} \min \quad & f(\boldsymbol{w}) \boldsymbol{w} \in \Omega \\ \text{s.t.} \quad & g_i(\boldsymbol{w}) \leq 0 \quad i = 1, \ldots, k_1 \\ & h_i(\boldsymbol{w}) = 0 \quad i = 1, \ldots, k_2 \end{aligned} \tag{2.18}$$

Then the generalized Lagrange form of the original problem (2.18):

$$\begin{aligned} L(\boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\beta}) &= f(\boldsymbol{w}) + \sum_{i=1}^{k_1} \alpha_i g_i(\boldsymbol{w}) + \sum_{i=1}^{k_2} \beta_i h_i(\boldsymbol{w}) \\ &= f(\boldsymbol{w}) + \langle \boldsymbol{\alpha}, g(\boldsymbol{w}) \rangle + \langle \boldsymbol{\beta}, h(\boldsymbol{w}) \rangle \end{aligned} \tag{2.19}$$

Furthermore, the Lagrange dual problem of the original problem (2.18) can be expressed as:

$$\begin{aligned} &\max \quad \theta(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \inf_{\boldsymbol{w} \in \Omega} L(\boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\beta}) \\ &\text{s.t.} \quad \boldsymbol{\alpha} \geq 0 \end{aligned} \tag{2.20}$$

Kuhn–Tucker theorem (Cristianini and Shawe-Taylor 2004): Given an optimization problem (2.18) defining in the convex domain $\Omega \subseteq R^n$, where f is the continuous convex function, and g_i and h_i are affine functions. Generally, the necessary and sufficient condition that a point $\boldsymbol{w}^*$ is the optimal point will have $\boldsymbol{\alpha}^*$ and $\boldsymbol{\beta}^*$, meeting:

$$\begin{aligned} \frac{\partial L(\boldsymbol{w}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)}{\partial \boldsymbol{\alpha}} = 0 \\ \frac{\partial L(\boldsymbol{w}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)}{\partial \boldsymbol{\beta}} = 0 \end{aligned} \tag{2.21}$$

$$\begin{aligned} &\alpha_i^* g_i(\boldsymbol{w}^*) = 0 \quad i = 1, \ldots, k_1 \\ &g_i(\boldsymbol{w}^*) \leq 0 \quad i = 1, \ldots, k_1 \\ &\alpha_i^* \geq 0 \quad i = 1, \ldots, k_1 \end{aligned} \tag{2.22}$$

where, the relational expression $g_i(\boldsymbol{w}^*) \leq 0 \quad i = 1, \ldots, k_1$ is called the KKT complementary condition.

1. **Optimal classification hyperplane**

For two types of classification problems of the linear separability, one of the key techniques is to seek the optimal classification hyperplane, i.e., determining the optimal linear discriminant function. Given $\boldsymbol{x}_i \in R^d$ is the sample data, and $y_i \in \{+1, -1\}$ is the corresponding class mark, $i = 1, \ldots,$ Ntr. The general form of the linear discriminant function is $g(\boldsymbol{x}) = \langle \boldsymbol{w}, \boldsymbol{x} \rangle + b$, and the corresponding classification plane is $\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b = 0$. In the formula, $\boldsymbol{x}$ is the d dimension characteristic vector, and $\boldsymbol{w}$ is called the weight vector. It can be expressed as $\boldsymbol{w} = [w_1, w_2, \ldots, w_d]^{\mathrm{T}}$. b is the constant, called as threshold weight. For the linear classifier with two types of problems, it can adopt the following decision rules:

$$\begin{cases} g(\boldsymbol{x}) > 0 \Rightarrow \boldsymbol{x} \in \omega_1 \\ g(\boldsymbol{x}) < 0 \Rightarrow \boldsymbol{x} \in \omega_2 \\ g(\boldsymbol{x}) = 0 \Rightarrow \boldsymbol{x} \in \omega_1 \text{ or } \omega_2 \end{cases} \tag{2.23}$$

The equation $g(\boldsymbol{x}) = 0$ defines a decision surface, and divides the points of belonging to different classes. The decision hyperplane is recorded as H.

$g(\boldsymbol{x})$ can be regarded as an algebra measurement of the distance from some point $\boldsymbol{x}$ to the hyperplane H in the eigenspace. If $\boldsymbol{x}$ is expressed as

$$\boldsymbol{x} = \boldsymbol{x}_p + r\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|} \tag{2.24}$$

In the formula, $\boldsymbol{x}_p$ is the projection vector of $\boldsymbol{x}$ in H, r is the vertical distance from $\boldsymbol{x}$ to H, $\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}$ is the unit vector in the $\boldsymbol{w}$ direction. In combination with the two formulas of (2.23) and (2.24), it can obtain

$$g(\boldsymbol{x}) = \left\langle \boldsymbol{w}, \left(\boldsymbol{x}_p + r\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|} \right) \right\rangle + b = \langle \boldsymbol{w}, \boldsymbol{x}_p \rangle + b + r\frac{\langle \boldsymbol{w}, \boldsymbol{w} \rangle}{\|\boldsymbol{w}\|} = r\|\boldsymbol{w}\| \tag{2.25}$$

or indicating it as

$$r = \frac{g(\boldsymbol{x})}{\|\boldsymbol{w}\|} \tag{2.26}$$

If $\boldsymbol{x}$ is the original point, then

$$g(\boldsymbol{x}) = b \tag{2.27}$$

In combination with the two formulas of (2.26) and (2.27), the distance from the original point to the hyperplane is obtained:

$$r_0 = \frac{b}{\|\boldsymbol{w}\|} \tag{2.28}$$

Thus, in order to separate the samples to be separated as well as possible, we demand the maximum geometric margin (i.e., the projection of two classes of the minimum distance segment in the direction perpendicular to the classification hyperplane), equivalent to minimizing $\|\boldsymbol{w}\|$. Figure 2.3 provides the graphical representation of the SVM maximization margin attribute in two cases.

Maximizing the geometric margin is equivalent to minimizing $\|\boldsymbol{w}\|$. Therefore, seeking the optimal classification plane is transformed into the following optimal problem:

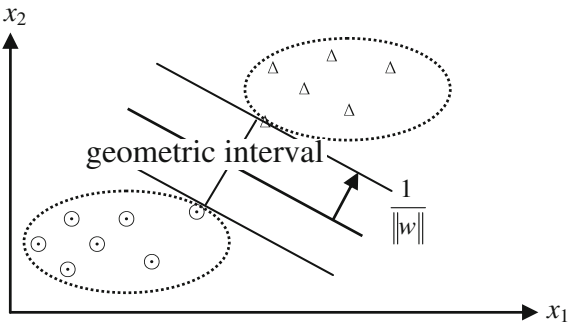

Fig. 2.3 Classification hyperplane of maximization geometric margin

$$\begin{aligned} &\min \quad \frac{1}{2}\|\boldsymbol{w}\|^2 \\ &\text{s.t.} \quad y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) - 1 \geq 0, \quad i = 1, 2, \ldots, \text{Ntr} \end{aligned} \tag{2.29}$$

Construct Lagrange function by the expression (2.29):

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{\text{Ntr}} \alpha_i [y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) - 1] \tag{2.30}$$

Here, Lagrange multiplier (support value) is $\alpha_i \geq 0$. By taking the derivative for the corresponding $\boldsymbol{w}$ and b, the following relational expression can be obtained.

$$\begin{aligned} \frac{\partial L(\boldsymbol{w}, b, \boldsymbol{\alpha})}{\partial \boldsymbol{w}} &= \boldsymbol{w} - \sum_{i=1}^{\text{Ntr}} \alpha_i y_i \boldsymbol{x}_i = 0 \\ \frac{\partial L(\boldsymbol{w}, b, \boldsymbol{\alpha})}{\partial b} &= \sum_{i=1}^{\text{Ntr}} \alpha_i y_i = 0 \end{aligned} \tag{2.31}$$

i.e.,

$$\begin{aligned} \boldsymbol{w} &= \sum_{i=1}^{\text{Ntr}} \alpha_i y_i \boldsymbol{x}_i \\ \sum_{i=1}^{\text{Ntr}} \alpha_i y_i &= 0 \end{aligned} \tag{2.32}$$

Substitute this formula into the Lagrange function (2.30), and obtain the dual problem of the original problem, i.e., maximizing the following objective functions:

$$\begin{aligned} L(\boldsymbol{w}, b, \boldsymbol{\alpha}) &= \sum_{i=1}^{\mathrm{Ntr}} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\mathrm{Ntr}} \alpha_i \alpha_j y_i y_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle \\ \text{s.t.} \quad & \sum_{i=1}^{\mathrm{Ntr}} \alpha_i y_i = 0, \quad \alpha_i \geq 0, \quad i = 1, 2, \ldots, \mathrm{Ntr} \end{aligned} \tag{2.33}$$

This dual problem is generally easier to handle than the original problem. According to Kuhn–Tucher theorem, the optimal solution meets:

$$\alpha_i [y_i (\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) - 1] = 0, \quad i = 1, 2, \ldots, \mathrm{Ntr} \tag{2.34}$$

Given $(\boldsymbol{\alpha}^*, b^*)$ is the optimal solution of the maximization (2.33), the corresponding discrimination functional expression is:

$$f(x) = \mathrm{sgn}\{\langle \boldsymbol{w}^*, \boldsymbol{x}_i \rangle + b^*\} = \mathrm{sgn}\left\{ \sum_{i=1}^{\mathrm{Ntr}} \alpha_i^* y_i K(\boldsymbol{x}_i, \boldsymbol{x}) + b^* \right\} \tag{2.35}$$

where, the vector is $\boldsymbol{w}^* = \sum_{i=1}^{\mathrm{Ntr}} \alpha_i^* y_i \boldsymbol{x}_i$. It is noted that the value of b does not occur in the dual problem, and the optimal value b^* can be inferred (the form is not sole) as by Kuhn–Tucher theorem:

$$b^* = -\frac{\max_{y_i=-1}(\langle \boldsymbol{w}^*, \boldsymbol{x}_i \rangle) + \min_{y_i=+1}(\langle \boldsymbol{w}^*, \boldsymbol{x}_i \rangle)}{2} \tag{2.36}$$

2. **Generalized optimal classification hyperplane**

When we are confronted with an inversion problem, i.e., needing inferring the unknown reason from the known result, it is necessary to consider the theoretical exposition of the ill-posed problem. The ill-posed problem is not only the mathematical phenomenon, but also exists extensively in the practical problems. The regularization theory is just proposed in allusion to this problem. The important content is that the minimization functional cannot obtain a very good solution in solving the problem of the operator equation of defining the ill-posed problem. On the contrary, we should adopt the unobvious solution, i.e., minimizing "deteriorated" (regularization) functional, to solve. Constructing the SVM with the generalized optimal classification hyperplane is just the embodiment of this idea.

While handling the linear inseparability problem, introduce the slack variable e_i, $i = 1, 2, \ldots, \mathrm{Ntr}$, the constraint condition in formula (2.29) becomes:

$$y_i [\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b] \geq 1 - e_i, \quad i = 1, 2, \ldots, \mathrm{Ntr} \tag{2.37}$$

Meanwhile, introduce the penalty factor γ to make the condition control for the misclassification sample. The corresponding objective function becomes:

$$J(\boldsymbol{w},\boldsymbol{e}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{\gamma}{2}\sum_{i=1}^{\mathrm{Ntr}} e_i \tag{2.38}$$

Moreover, the constraint condition corresponding to $\alpha_i \geq 0, \quad i = 1, 2, \ldots, \mathrm{Ntr}$ becomes $\gamma \geq \alpha_i \geq 0, \quad i = 1, 2, \ldots, \mathrm{Ntr}$. When the class division has the error, the corresponding slack variable is more than 0. Therefore, the sum of the slack variable is the upper bound of the classification error in the training set.

3. **Nonlinear problems**

While handling the nonlinear problems, SVM maps the data point without the linear separation by the original space into the linear separable point in the transformation space through introducing the nonlinear mapping ϕ, as shown in Fig. 2.4. Under this condition, $\boldsymbol{x}_i$ in the optimal expression should be correspondingly replaced with $\phi(\boldsymbol{x}_i)$, while the inner product $\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle$ is replaced with

$$\boldsymbol{K}(i,j) = K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \rangle \tag{2.39}$$

Here, K is called as the kernel function operator, which is an inner product algorithm in the transformation space. $\boldsymbol{K}$ is the kernel function matrix of the sample set under the effect of the kernel function operator. In case that it is not confused, both in this book are often referred to as kernel function.

The nonlinear mapping ϕ is generally difficult to construct, and usually the dimension of the corresponding transformation space is very high and even infinity, resulting in great difficulties for the analysis. It is noted that above-discussed optimal and generalized linear classification functions, and the final classification discrimination function only includes the inner product of the support vector in the samples to be classified and the training sample, i.e., kernel function operation. Meanwhile, the solution process also involves the kernel function operation among the training samples. Thus, if solving the optimal linear classification problems in an eigenspace, it is necessary to only know the inner product operation in this space. If the inner product in the transformation space can be directly calculated by

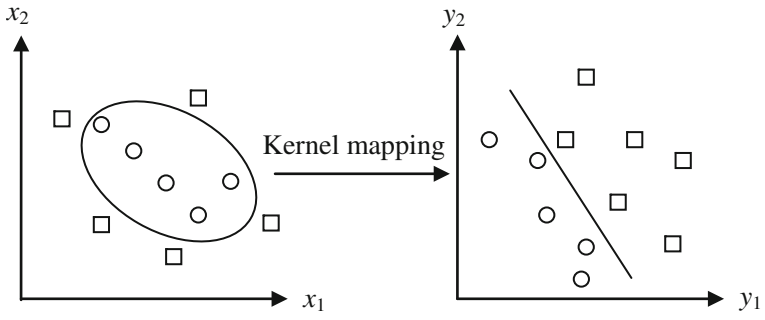

Fig. 2.4 Kernel mapping transforming nonlinear problem into linear problem

the variable in the original space through the kernel function, the calculation complexity of solving the problem of the optimal classification plane will not be increased much even if the dimension of the transformation space is increased much. In this way, the quotation of the kernel function skillfully solves the problem of constructing and disposing the nonlinear mapping.

The statistical learning theory indicates that, according to Hilbert–Schmidt principle, only if an operation meets the Mercer conditions, it can be used as the kernel function here. Mercer conditions state that, for any symmetric function $K(\boldsymbol{x},\boldsymbol{x}')$, the necessary and sufficient condition as the inner product operation in the eigenspace is, for any $\phi(\boldsymbol{x}) \neq 0$ and $\int \phi^2(\boldsymbol{x})\mathrm{d}\boldsymbol{x} < \infty$, then

$$\iint K(\boldsymbol{x},\boldsymbol{x}')\phi(\boldsymbol{x})\phi(\boldsymbol{x}')\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{x}' > 0 \tag{2.40}$$

Such a symmetric function $K(\boldsymbol{x},\boldsymbol{x}')$ can be regarded as the kernel function.

The kernel function is closely related to the performance of SVM. How to construct the kernel function relevant to the practical problem has always been the main content of SVM research. The selection of the kernel function does not have some theoretical guidance, and the parameter selection still adopts the empirical mode. The several common kernel functions now are shown in Table 2.1. From a large number of experiment results, the classification result of the Gaussian radial basis kernel function is relatively good.

Below we replace the dot product in the optimal classification plane by the kernel function (inner product) $K(\boldsymbol{x},\boldsymbol{x}')$, equivalent to transforming the original eigenspace into a certain new eigenspace. Then, seek the optimization problem as shown in (2.41) transformed from the optimal classification plane:

$$\begin{aligned} \min \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 \\ \text{s.t.} \quad & y_i[\langle \boldsymbol{w}, \phi(\boldsymbol{x}_i)\rangle + b] - 1 \geq 0 \\ & i = 1, 2, \ldots, \mathrm{Ntr} \end{aligned} \tag{2.41}$$

Then, Lagrange function is as follows:

Table 2.1 Several common Kernel functions

Kernel function name	Kernel function expression
Linear kernel function	$K(\boldsymbol{x},\boldsymbol{y}) = \langle \boldsymbol{x},\boldsymbol{y}\rangle$
Polynominal kernel function	$K(\boldsymbol{x},\boldsymbol{y}) = [\langle \boldsymbol{x},\boldsymbol{y}\rangle + 1]^d$
Gaussian radial basis kernel function	$K(\boldsymbol{x},\boldsymbol{y}) = \exp[-\|\boldsymbol{x}-\boldsymbol{y}\|^2/2\sigma^2]$
Index radial basis kernel function	$K(\boldsymbol{x},\boldsymbol{y}) = \exp[-\|\boldsymbol{x}-\boldsymbol{y}\|/2\sigma^2]$
Sigmoid kernel function	$K(\boldsymbol{x},\boldsymbol{y}) = \tanh(k\langle \boldsymbol{x},\boldsymbol{y}\rangle - \delta)$

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{\text{Ntr}} \alpha_i [\langle \boldsymbol{w}, \phi(\boldsymbol{x}_i)\rangle + b - y_i] \tag{2.42}$$

(2.41) can be transformed into the objective function under the maximization in the same mode as the linear problem:

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \sum_{i=1}^{\text{Ntr}} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{\text{Ntr}} \alpha_i \alpha_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{2.43}$$

Then, the corresponding discrimination function is:

$$f(x) = \text{sgn}\{\langle \boldsymbol{w}^*, \phi(\boldsymbol{x}_i)\rangle + b^*\} = \text{sgn}\left\{\sum_{i=1}^{\text{Ntr}} \alpha_i^* y_i K(\boldsymbol{x}_i, \boldsymbol{x}) + b^*\right\} \tag{2.44}$$

After introducing the concept of the high-dimensional space inner product (i.e., kernel function), the basic idea of SVM can be simply summarized as firstly transforming the input space into a high-dimensional space by the nonlinear transformation, and then gaining the optimal linear classification plane in this new space, while the nonlinear transformation is realized by defining the proper inner product function.

4. **Expression form of basic theory**

Maximizing the classification interval is actually controlling the generalization ability, which is one of the core ideas of SVM. The statistical learning theory indicates that, in d dimension space, given the sample is distributed in the hypersphere range with the radius of R, VC dimension of the indication function set $f(\boldsymbol{x}, \boldsymbol{w}, b)$ composed by the regular hyperplane in conformity with the condition of $\|\boldsymbol{w}\| \leq k$ will meet the following boundary:

$$h \leq \min(R^2 k^2, d) + 1 \tag{2.45}$$

Therefore, minimizing $\|\boldsymbol{w}\|^2$ is the minimum classification, i.e., minimizing the upper bound of VC dimension, and accordingly realizing the selection of the dual function complexity in SRM code. The optimal classification plane and generalized optimal classification plane are actually dividing the classification function set $S = \{(\langle \boldsymbol{w}, \boldsymbol{x}\rangle + b)\}$ into some normalized subsets in accordance with the module of the weight (classification interval in the case of the linear separability). Each subset is as follows:

$$S = \left\{(\langle \boldsymbol{w}, \boldsymbol{x}\rangle + b) : \|\boldsymbol{w}\|^2 \leq c_k\right\} \tag{2.46}$$

For the linear separability, the optimal classification plane is to seek the normalized subset with the minimum bound of the expected risk on the premise of the fixed empirical risk of 0. Moreover, in the case of the linear inseparability, the generalized optimal classification plane is to seek the minimum bound of the expected risk under the condition of controlling the misclassification sample. Therefore, they are optimal in the sense of the bound of the expected risk, and are the specific embodiment of the structural risk minimization principle. For the linear function in d dimension space, VC dimension is $d + 1$. However, from the above discussion, under the constraint condition of $\|\boldsymbol{w}\| \leq k$, VC dimension may be reduced greatly. The smaller VC dimension function set can also be gained even in very high-dimensional space, to guarantee the relatively good promotion. At the same time, we can see that, the complexity of the calculation depends on the number of samples, especially the support vector number in the sample, rather than the spatial dimension, through transforming the original problem to the dual problem. These characteristics make it possible for SVM to effectively handle the high- dimensional problems.

2.3.3 *Construction of Multi-class Classifier with the Simplest Structure*

At present, there are two most common multi-class classifiers, respectively, 1-a-r (1-against-rest) multi-class classifier and 1-a-1 (1-against-1) multi-class classifier. Figure 2.5 provides the structure of these two traditional multi-class classifiers. These two classifiers have the relatively complex structural design and very large calculated amount. Take N class problem as an example. The algorithm 1-a-r is to construct N two-class target subclassifiers. The k subclassifier regards the training sample in the k class as the positive training sample, and the others are the negative training samples. For some input sample, the classification result is the subclassifier output value as the maximum corresponding class. The algorithm 1-a-1 is proposed by Knerr, which is to construct each two classes in N class into a subclassifier, and

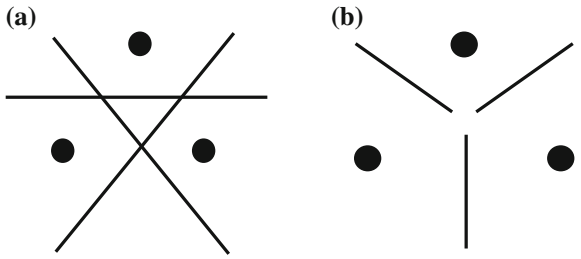

Fig. 2.5 Structure of two classic multi-class classifier. **a** 1-a-r structure. **b** 1-a-1 structure

accordingly needs constructing $N(N-1)/2$ subclassifiers, combining these subclassifiers, and determining the classification result by the voting method.

The common weakness of these two methods is unbounded promotion error, and a large number of classifiers, resulting in the low speed of decision. On account of the problems of the complex classifier structure due to the excessive subclassifiers, this section puts forward a method of simplifying the structure of the multi-class target classifier.

1. **Construction of multi-class classifier with the simplest structure**

For the convenience of the description, it can firstly describe the construction of the classifier structure with eight classes of problems as an example. Given all sample sets as P, the construction process of the classifier is as follows:

1. Firstly, P is equally divided into two sample sets by category, respectively, noted as P_1 and P_2. Also it is noted as $A_1 = P_1$ and $B_1 = P_2$, and the corresponding class marks of A_1 and B_1 are reset as +1 and −1. Then train A_1 and B_1 as two classes of targets, and gain the first two-class target sub-classifier C_1.
2. Firstly, P_1 is equally divided into two sample sets by category, respectively, noted as P_{11} and P_{21}. Then P_2 is equally divided into two sample sets by category, respectively, noted as P_{12} and P_{22}. Also it is noted as $A_2 = P_{11} \cup P_{12}$ ($\cup$ is the set union operator), $B_2 = P_{21} \cup P_{22}$, and the corresponding class marks of A_2 and B_2 are reset as +1 and −1. Finally, train A_2 and B_2 as two classes of targets, and gain the second two-class target sub-classifier C_2.
3. Firstly, P_{11} is equally divided into two sample sets by category, respectively, noted as P_{111} and P_{211}. Then P_{21} is equally divided into two sample sets by category, respectively noted as P_{121} and P_{221}. P_{12} is equally divided into two sample sets by category, respectively noted as P_{112} and P_{212}. P_{22} is equally divided into two sample sets by category, respectively noted as P_{122} and P_{222}. Also it is noted as $A_3 = P_{111} \cup P_{121} \cup P_{112} \cup P_{122}$, $B_3 = P_{211} \cup P_{221} \cup P_{212} \cup P_{222}$, and the corresponding class marks of A_3 and B_3 are reset as +1 and −1. Finally, train A_2 and B_2 as two classes of targets, and gain the third two-class target sub-classifier C_3.
4. Three subclassifiers of C_1, C_2 and C_3 are combined into a multi-class target classifier C.

Thus, the sample to be decided can be decided as the only one class through the decision intersection of three subclassifiers. Figure 2.6 provides the construction

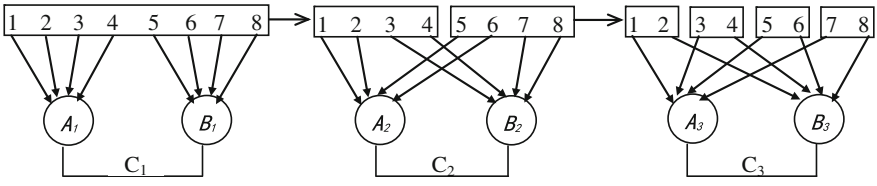

Fig. 2.6 Construction of three subclassifiers in eight classes

schematic diagram of three subclassifiers in this example. The rectangle box shows the original class set and the division situation. The circular box shows the original class mark set assigned into a class in some step. It is noted that the class dividing mode of each step is arbitrary in the above construction process of the classifier.

Generally, for 2^N type of problem, it can be finished by the following descriptive process.

1. 2^{N-1} class sample in the original sample is combined into a class sample set, and the remaining is the other class sample set. Thus, train and gain the first two-class target sub-classifier. The original sample is divided into two sets by category.
2. Take 2^{N-2} (totaling to 2^{N-1}) class samples respectively in the two divided sets from the above step, to combine into a class sample set, and the remaining is the other class sample set. Thus, train and gain the second two-class target sub-classifier. The original sample is divided into four sets by category.
3. In the k step, take 2^{N-k} (totaling to 2^{N-1}) class samples respectively in the 2^{k-1} divided class set from the $k-1$ step, to combine into a class sample set, and the remaining is the other class sample set. Thus, train and gain the k two-class target sub-classifier. The original sample is divided into 2^k sets by category.
4. Continue, gain N different subclassifiers, and combine them into a multi-class target classifier. Each sample can be decided as the only one class through the output value of N subclassifiers.

If the number of classes is between 2^{N-1} and 2^N, the number of classes can be transformed into "2^N" in the method of adding the virtual classes, and then the virtual class sample is removed from N subclassifiers constructed finally. In fact, the added virtual class is the formal participation without the practical participation. For example, when the number of classes is 6, it can add the 7th and 8th classes. So, we will gain the same construction form of the classifier as Fig. 2.6. The difference is that the design result removes the 7th and 8th classes of samples in the final A_1, B_1, A_2, B_2, A_3 and B_3.

It is easy to analyze that the number of subclassifiers from the new method is far less than two typical methods of 1-a-1 and 1-a-r. Table 2.2 is the comparison of the required number of subclassifiers by three methods under different classes (including $K = 2^N$).

If failing to consider the complexity of the design, the time for classification can be used for measuring the complexity of the classifier. For training, the complexity

Table 2.2 Comparison of required number of subclassifiers by structure of three classifiers

Number of subclassifiers		Structure of classifier		
		1-a-1	1-a-r	New method
Number of classes	4	6	4	2
	16	120	16	4
	K	$K(K-1)/2$	K	N

Table 2.3 Relative frequency of Kernel function calculation by three classifier structures in test

Frequency of Kernel function calculation		Structure of classifier		
		1-a-1	1-a-r	New method
Number of classes	4	4	3	2
	16	16	15	4
	K	K	K–1	N

of the subclassifiers under different structures may be different, and various subclassifiers in the same structure have the specific relations, so the number of the subclassifiers cannot only be used for measuring the complexity of the classifier. By contrast, the test process of the classification is not restricted by the above factors, and the time consuming is mainly used for the kernel function operation. Thus, the frequency of the kernel function operation as required for processing can be used for the measurement index of the complexity. Under the same conditions, for the same $K = 2^N$ class of the training sample, it can calculate the relative value for the frequency of the kernel function operation as required for processing by the classifier with different structures in the test (see Table 2.3).

For the essence, relative to the two traditional methods, the new method abandons the massive redundant information among various subclassifiers, and gains the simplification of the classifier structure, and the greater improvement of the classification speed.

2.3.4 *Least Squares SVM and Its SMO Optimization Algorithm*

In recent years, there have been many developed and transformed SVM types. During these development types, the least squares SVM has been widely applied due to the efficient classification and regression functions. What is more, the mathematical model of the least squares SVM is only an optimization problem of the error cost function sum of squares with the equality constraint, and the solution can be made in the linear system. This book focuses on adopting such type of SVM.

1. **Least Squares SVM** (Suykens et al. 2002)

The optimization problem expression of the least squares SVM is as follows:

$$\begin{aligned} \min_{\boldsymbol{w},b,\boldsymbol{e}} \quad & J(\boldsymbol{w},\boldsymbol{e}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{\gamma}{2}\sum_{i=1}^{\mathrm{Ntr}} e_i^2 \\ \text{s.t.} \quad & y_i = \langle \boldsymbol{w}, \phi(\boldsymbol{x}_i)\rangle + b + e_i, \\ & i = 1, 2, \ldots, \mathrm{Ntr},\ \gamma > 0. \end{aligned} \tag{2.47}$$

where, $\boldsymbol{x}_i \in R^d$ is the sample data, and $y_i \in \{+1, -1\}$ and e_i are, respectively, the class mark and discrimination error. $i = 1, \ldots, \mathrm{Ntr}$. The corresponding dual problem is:

$$\begin{aligned} \min_{\boldsymbol{w},b,\boldsymbol{e},\alpha} \quad & L(\boldsymbol{w}, b, \boldsymbol{e}, \boldsymbol{\alpha}) \\ & = J(\boldsymbol{w}, \boldsymbol{e}) - \sum_{i=1}^{\mathrm{Ntr}} \alpha_i \{\langle \boldsymbol{w}, \phi(\boldsymbol{x}_i)\rangle + b + e_i - y_i\} \end{aligned} \tag{2.48}$$

The optimal KKT condition is as follows:

$$\begin{cases} \dfrac{\partial L}{\partial \boldsymbol{w}} = 0 \rightarrow \boldsymbol{w} = \displaystyle\sum_{i=1}^{\mathrm{Ntr}} \alpha_i \phi(\boldsymbol{x}_i) \\ \dfrac{\partial L}{\partial b} = 0 \rightarrow \displaystyle\sum_{i=1}^{\mathrm{Ntr}} \alpha_i = 0 \\ \dfrac{\partial L}{\partial e_i} = 0 \rightarrow \alpha_i = \gamma e_i, \quad i = 1, 2, \ldots, \mathrm{Ntr} \\ \dfrac{\partial L}{\partial \alpha_i} = 0 \rightarrow \langle \boldsymbol{w}, \phi(\boldsymbol{x}_i)\rangle + b + e_k - y_i = 0, \quad i = 1, 2, \ldots, \mathrm{Ntr} \end{cases} \tag{2.49}$$

Upon eliminating $\boldsymbol{w}$ and e by the elimination method, the above formula can be further expressed as:

$$\begin{bmatrix} 0 & 1_v^T \\ 1_v & \boldsymbol{K} + \boldsymbol{I}/\gamma \end{bmatrix} \begin{bmatrix} b \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ \boldsymbol{y} \end{bmatrix} \tag{2.50}$$

where, $\boldsymbol{y} = [y_1, y_2, \ldots, y_{\mathrm{Ntr}}]^{\mathrm{T}}$, $1_v = [1, 1, \ldots, 1]^{\mathrm{T}}$, $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_{\mathrm{Ntr}}]^{\mathrm{T}}$.

2. **SMO optimization algorithm of least squares SVM**

We know that the least squares SVM can be directly and conveniently solved into the linear system. However, when the number of the training samples is too large, the direct solution becomes very difficult. Therefore, it is necessary to promote the efficient SMO algorithm (Shevade et al. 2000) to a such type of SVM solution, as the effective replacement of the linear solution.

The dual form of the formula (2.48) is as follows:

$$L = \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{\gamma}{2}\sum_{i=1}^{\mathrm{Ntr}} e_i^2 + \sum_{i=1}^{\mathrm{Ntr}} \alpha_i [y_i - \langle \boldsymbol{w}, \phi(\boldsymbol{x}_i)\rangle - b - e_i] \tag{2.51}$$

Upon applying the Wolfe duality theory, we can obtain the following form of the optimization problem:

$$\begin{aligned} \max \quad & f(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{i=1}^{\mathrm{Ntr}}\sum_{j=1}^{\mathrm{Ntr}}\alpha_i\alpha_j\tilde{K}(\boldsymbol{x}_i,\boldsymbol{x}_j) + \sum_{i=1}^{\mathrm{Ntr}}\alpha_i y_i \\ s.t. \quad & \sum_{i=1}^{\mathrm{Ntr}}\alpha_i = 0 \end{aligned} \tag{2.52}$$

where,

$$\tilde{K}(\boldsymbol{x}_i,\boldsymbol{x}_j) = K(\boldsymbol{x}_i,\boldsymbol{x}_j) + \frac{1}{\gamma}\delta_{i,j}, \quad \delta_{i,j} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \tag{2.53}$$

The Lagrange form of the formula (2.52) is as follows:

$$\bar{L} = -\frac{1}{2}\sum_{i=1}^{\mathrm{Ntr}}\sum_{j=1}^{\mathrm{Ntr}}\alpha_i\alpha_j\tilde{K}(\boldsymbol{x}_i,\boldsymbol{x}_j) + \sum_{i=1}^{\mathrm{Ntr}}\alpha_i y_i + \beta\sum_{i=1}^{\mathrm{Ntr}}\alpha_i \tag{2.54}$$

Defining

$$F_i = -\frac{\partial f}{\partial \alpha_i} = \sum_{i=1}^{\mathrm{Ntr}}\alpha_i\tilde{K}(\boldsymbol{x}_i,\boldsymbol{x}_j) - y_i, \quad i = 1,2,\ldots,\mathrm{Ntr}. \tag{2.55}$$

From KKT conditions of the formula (2.54), we can get

$$\frac{\partial \bar{L}}{\partial \alpha_i} = \beta - F_i = 0 \Rightarrow F_i = \beta, \quad i = 1,2,\ldots,\mathrm{Ntr}. \tag{2.56}$$

This formula explains that the necessary and sufficient condition of the support value vector $\boldsymbol{\alpha} = [\alpha_1,\alpha_2,\ldots,\alpha_{\mathrm{Ntr}}]^{\mathrm{T}}$ as the optimal solution is:

$$\max_i\{F_i\} = \min_i\{F_i\} \tag{2.57}$$

Thus, we can provide the iterative method of the solution optimal $\boldsymbol{\alpha} = [\alpha_1,\alpha_2,\ldots,\alpha_{\mathrm{Ntr}}]^{\mathrm{T}}$. Note

$$\begin{aligned} i_{\max} &= \arg\max_i\{F_i\} \\ i_{\min} &= \arg\min_i\{F_i\} \end{aligned} \tag{2.58}$$

$$\begin{aligned} \tilde{\boldsymbol{\alpha}} &= [\tilde{\alpha}_1,\tilde{\alpha}_2,\ldots,\tilde{\alpha}_{\mathrm{Ntr}}]^{\mathrm{T}} \\ \tilde{\alpha}_i &= \begin{cases} \alpha_i - t, & i = i_{\max} \\ \alpha_i + t, & i = i_{\max} \\ \alpha_i, & \text{other } i \end{cases} \end{aligned} \tag{2.59}$$

For the given $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_{\mathrm{Ntr}}]^{\mathrm{T}}$, if it fails to meet the optimum condition (2.57), $\alpha_{i_{\max}}$ and $\alpha_{i_{\min}}$ are, respectively, replaced by $\alpha_{i_{\max}} - t$ and $\alpha_{i_{\min}} + t$, i.e., replacing $\boldsymbol{\alpha}$ by $\tilde{\boldsymbol{\alpha}}$. The selection of the parameter t needs maximizing $f(\tilde{\boldsymbol{\alpha}})$, and the optimal value is provided by the following formula:

$$\begin{aligned} &\frac{\partial f(\tilde{\boldsymbol{\alpha}}(t))}{\partial t} = 0 \Rightarrow t = t^* = (F_{i_{\min}} - F_{i_{\max}})/\eta \\ &\eta = \{2 * \tilde{K}(\boldsymbol{x}_{i_{\max}}, \boldsymbol{x}_{i_{\min}}) - \tilde{K}(\boldsymbol{x}_{i_{\max}}, \boldsymbol{x}_{i_{\max}}) - \tilde{K}(\boldsymbol{x}_{i_{\min}}, \boldsymbol{x}_{i_{\min}})\}. \end{aligned} \tag{2.60}$$

After getting $\tilde{\boldsymbol{\alpha}}$, the new iterative process starts from here. Thus, the least squares support vector machine theory is promoted completely.

2.3.5 *Triply Weighted Classification Method*

SVM shows good performance in the hyperspectral image classification, but how to further improve the classification performance is still a researchable content. During the process of the hyperspectral image classification, the generalization performance of SVM is sensitive to the outliers point and noise interference pixel (collectively referred to as anomaly pixel) during the training process, while they inevitably exist extensively in the hyperspectral data, and influence the correctness of the model. The modeling method of SVM excessively depends on the training samples, and is very sensitive to the existence of the anomaly pixel. Generally, the introduction of a few anomaly pixels may fully destroy the generalization performance of the model.

Suykens et al. (2002) came up with LSSVM weighted method, to make the pixel and outliers point seriously suffering from the noise interference in the hyperspectral image control effectively, and accordingly gain better Robust feature and generalization ability. Such a weighted method includes the complete preliminary training. Moreover, the calculated amount required by the training is generally larger, especially when the training samples are more, this method will be time consuming, Due to this reason, the method is not popularized effectively.

The existing hyperspectral image classification weighted method is generally implemented on account of the training samples, while little literature considers the following two situations. Firstly, the influence of different characteristics (or band and spectral section) of the hyperspectral image on the class separability is different, i.e., their effect on the classification is different. Thus, they should not be treated equally in the classifier design. Secondly, in the practical application, the remote-sensing data classes are numerous, while the significance of different classes on the hyperspectral data analysis is often different, or the researchers have different degree of interests on them. Thus, it is necessary to consider in the classifier design. For this purpose, this section introduces a triply weighted method in the LSSVM

theory-based hyperspectral image classification problem, so as to further enhance the classification analysis effect.

1. **Pixel weighting in the hyperspectral image classification**

The optimization problem expression of LSSVM is shown in (2.47). In order to make the samples with different anomaly degrees embodying in the classification model, their corresponding classification error should be distributed with different weights in the cost function, i.e., gaining the weighting training model of LSSVM. Given e_i corresponds to the weight v_i, this formula becomes:

$$\begin{aligned} \min_{\boldsymbol{w},b,\boldsymbol{e}} \quad & J(\boldsymbol{w},\boldsymbol{e}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{\gamma}{2}\sum_{i=1}^{\mathrm{Ntr}} (v_i e_i)^2 \\ \text{s.t.} \quad & y_i = \langle \boldsymbol{w}, \phi(\boldsymbol{x}_i)\rangle + b + e_i \\ & i = 1, 2, \ldots, \mathrm{Ntr}, \quad \gamma > 0. \end{aligned} \tag{2.61}$$

Thus, how to reasonably determine the weight v_i becomes the key problem in the sample weighting. Due to longer relative distance between the anomaly sample and the corresponding class center in the training sample, the anomaly degree can be measured by the distance scale (Song et al. 2002). In this way, the smaller weight can be distributed for the sample with the larger anomaly degree to weaken the adverse effects. On the other hand, because of the differences of the intra-class spectrum, the pure sample cannot concentrate on the corresponding class center, but has a relatively small deviation. In view of this, while calculating the distance, we can subtract a correction constant from the distance obtained previously. For this purpose, we can firstly determine the class center as the center of the circle, including the minimum radius of the specified proportional sample point in the class. Further this radius is given as the above correction constant.

Given the class center corresponding to the sample $\boldsymbol{x}_i$ is $\boldsymbol{x}_0$, while the circle with $\boldsymbol{x}_0$ as the center and the radius of r includes the minimum circle of the specified proportional sample in the class. The uncorrected distance from the sample $\boldsymbol{x}_i$ to $\boldsymbol{x}_0$ is expressed by $\hat{D}(\boldsymbol{x}_i, \boldsymbol{x}_0)$, and then the calculation formula of $\hat{D}(\boldsymbol{x}_i, \boldsymbol{x}_0)$ is as follows:

$$\begin{aligned} \hat{D}(\boldsymbol{x}_i, \boldsymbol{x}_0) &= \|\phi(\boldsymbol{x}_i) - \phi(\boldsymbol{x}_0)\| \\ &= \left(K(\boldsymbol{x}_i, \boldsymbol{x}_i) + K(\boldsymbol{x}_0, \boldsymbol{x}_0) - 2K(\boldsymbol{x}_i, \boldsymbol{x}_0)\right)^{1/2} \end{aligned} \tag{2.62}$$

Accordingly, the correction distance $D(\boldsymbol{x}_i, \boldsymbol{x}_0)$ from $\boldsymbol{x}_i$ to the class center $\boldsymbol{x}_0$ can be stipulated as:

$$D(\boldsymbol{x}_i, \boldsymbol{x}_0) = \hat{D}(\boldsymbol{x}_i, \boldsymbol{x}_0) - r, \quad i = 1, 2, \ldots, \mathrm{Ntr}. \tag{2.63}$$

Noting

$$\begin{aligned} D_{\max} &= \max_i(D(\boldsymbol{x}_i, \boldsymbol{x}_0)) \\ D_{\min} &= \min_i(D(\boldsymbol{x}_i, \boldsymbol{x}_0)) \end{aligned} \tag{2.64}$$

And $\mathrm{Nor}D(\boldsymbol{x}_i, \boldsymbol{x}_{y_i})$ is used for expressing the normalization form of $D(\boldsymbol{x}_i, \boldsymbol{x}_{y_i})$, i.e.,

$$\mathrm{Nor}D(\boldsymbol{x}_i, \boldsymbol{x}_{y_i}) = D(\boldsymbol{x}_i, \boldsymbol{x}_{y_i})/D_{\max}, \quad i = 1, 2, \ldots, \mathrm{Ntr}. \tag{2.65}$$

The weight factor can be obtained by the following formula:

$$v_i = 1 - \mathrm{Nor}D(\boldsymbol{x}_i, \boldsymbol{x}_{y_i})^2 + (D_{\min}/D_{\max})^2 \quad i = 1, 2, \ldots, \mathrm{Ntr}. \tag{2.66}$$

It is easy to verify $0 < v_i \leq 1$. The original error term $\{e_i\}_{i=1}^{\mathrm{Ntr}}$ is replaced by the weighted form $\{v_i e_i\}_{i=1}^{\mathrm{Ntr}}$, and the sample weighting type LSSVM as shown in the formula (2.61) can be obtained.

2. **Feature weighting in the hyperspectral image classification**

The key to the feature weighting is to find out an appropriate weighting matrix. This matrix can enhance the effective feature, and weaken the feature with the poorer class separability. Fisher linear discriminant analysis is a widely used classification technique, and has been extensively applied in the pattern recognition. The inverse matrix of the intra-class divergence matrix can reflect different contributions of different features for the classification effect well (Ji et al. 2004), and the effect has been verified in the spectral separation (Chang and Ji 2006). Thus, it can be applied to the feature weighting of the hyperspectral image classification, and the specific methods are as follows.

Given Ntr training sample vectors are used for the classification, $\boldsymbol{\mu}_j$ is the mean value of the j class of the sample ($j = 1, 2, \ldots, \mathrm{Ntr}$), i.e.,

$$\boldsymbol{\mu}_j = \frac{1}{n_j} \sum_{\boldsymbol{x}_i \in C_j} \boldsymbol{x}_i \tag{2.67}$$

C_j and n_j, respectively, stand for the j class of the sample set and the number of samples, and accordingly the intra-class divergence matrix $\boldsymbol{S}_W$ can be defined as follows:

$$\boldsymbol{S}_w = \sum_{j=1}^{\mathrm{Nc}} \boldsymbol{S}_j \tag{2.68}$$

Here

$$\boldsymbol{S}_j = \sum_{\boldsymbol{x} \in C_j} (\boldsymbol{x} - \boldsymbol{\mu}_j)(\boldsymbol{x} - \boldsymbol{\mu}_j)^{\mathrm{T}} \tag{2.69}$$

$\boldsymbol{S}_w$ is the real symmetric matrix, consequently, the orthogonal matrix $\boldsymbol{U}$ transforms the opposite angle into the matrix $\boldsymbol{B}$:

$$\boldsymbol{U}^{\mathrm{T}} \boldsymbol{S}_w \boldsymbol{U} = \boldsymbol{B} \tag{2.70}$$

Further, it can be inferred:

$$\boldsymbol{S}_w{}^{-1} = (\boldsymbol{U}\boldsymbol{B}\boldsymbol{U}^{\mathrm{T}})^{-1} = (\boldsymbol{U}\boldsymbol{B}^{-1/2})(\boldsymbol{U}\boldsymbol{B}^{-1/2})^{\mathrm{T}} \tag{2.71}$$

Noting $\boldsymbol{G} = (\boldsymbol{U}\boldsymbol{B}^{-1/2})^{\mathrm{T}}$, then $\boldsymbol{G}$ can be used for the feature weighting matrix in the classification problem.

3. **Class weighting in the hyperspectral image classification**

The matrix equation of LSSVM is rewritten as follows:

$$\begin{bmatrix} 0 & 1_v^T \\ 1_v & \boldsymbol{K} + \boldsymbol{I}/\gamma \end{bmatrix} \begin{bmatrix} b \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ \boldsymbol{y} \end{bmatrix} \tag{2.72}$$

$\boldsymbol{I}$ is the unit matrix of Ntr × Ntr. When $\boldsymbol{I}$ is the unit matrix, it shows that the training process equally considers each training sample. According to the thought of Suykens et al. (2002), if different classes of the training samples are treated differently in the optimization model, it will directly reflect as the different diagonal element assignments of $\boldsymbol{I}$ in the corresponding linear Eq. (2.72). In other words, the diagonal element assignment of $\boldsymbol{I}$ can embody the emphasis on each training sample. The larger some weight of $\boldsymbol{I}$ is, the less indifference the training process has on the corresponding sample, and vice versa. The class weighting means the diagonal element value through resetting the corresponding position of some class sample in $\boldsymbol{I}$, rather than the original equivalent setting, for changing the emphasis on each class, accordingly protecting the class of interest, and restraining the non-essential class. Thus, reducing the weight corresponding to the training sample in the class of interest properly and increasing the weight corresponding to the training sample in the class of non-interest properly can effectively enhance the classification accuracy of the class of interest.

The above three weighting methods can be used independently, and also can be used in combination by any check mode. Figure 2.7 is the relationship diagram for mapping the class center distance of some real hyperspectral data sample into the weighting value. Figure 2.8 provides the schematic diagram for the operation interface of the check weighting classification of the hyperspectral image.

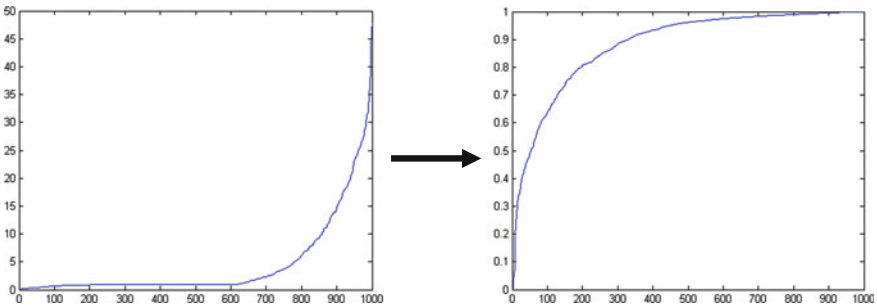

Fig. 2.7 Mapping from uncorrected distance to weight (X-coordinate sample is reciprocal correspondence relation)

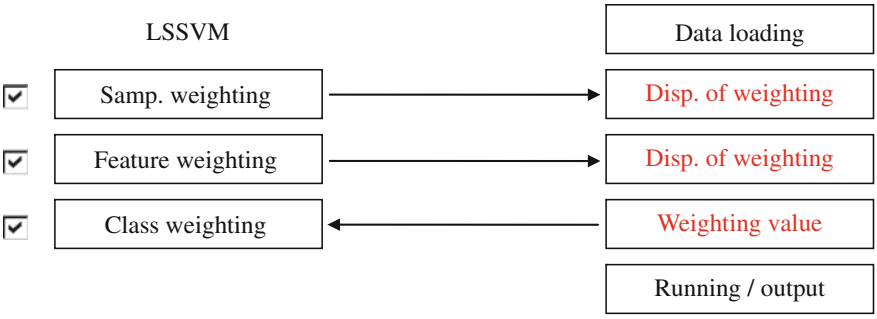

Fig. 2.8 Operation interface of least squares SVM check weighting

2.4 Performance Assessment for SVM-Based Classification

One of the hyperspectral remote-sensing images mainly used in this book was from one section of the Indian agriculture and forestry hyperspectral remote-sensing experimental area in the northwest of Indiana of America shot in June 1992. After removing some bands with larger noise effect, select 200 bands as the research object from the original 220 bands. The images are supervised, and the land object classes represented by the supervision class mark 0–17 are successively Background, Alfalfa, Corn-notill, Corn-min, Corn, Grass/ Pasture, Grass/ Trees, Grass/ pasture-mowed, Hay-windrowed, Oats, Soybeans-notill, Soybeans-min, Soybean-clean, Wheat, Woods, Bldg-Grass-, ree-Drives, and Stone-steel towers. Figure 2.9 provides the false color composite image by the bands of 50, 27 and 17. The number of various land object pixels and the image data characteristics are, respectively, shown as Fig. 2.10 and Table 2.4.

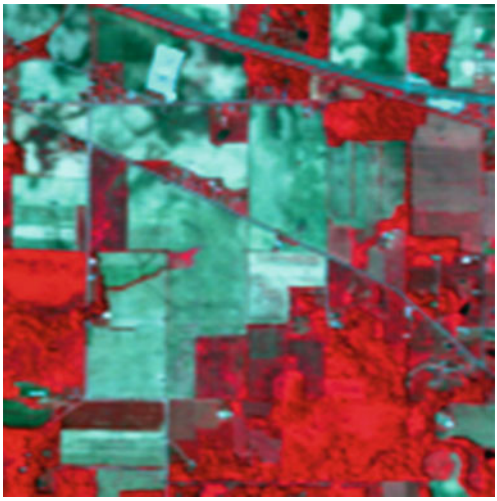

Fig. 2.9 False color composite image by Bands of 50, 27 and 17 as RGB channel

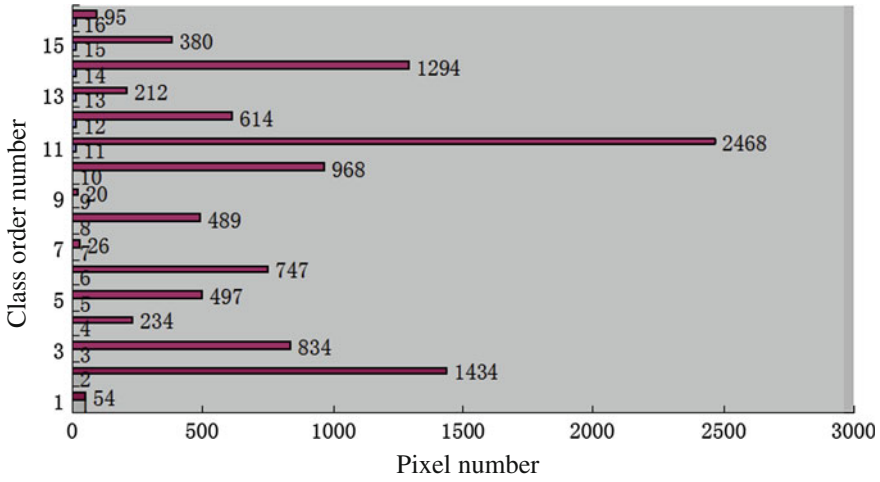

Fig. 2.10 Quantity statistics of pixels included in each class of image

Table 2.4 Data characteristics of experimental image

Number of bands	220 units
Wave length range	400–2500 nm
Spectral resolution	≈10 nm
Spatial resolution	20 × 20 m
Image size	144 × 144
Flight height	20 km (NASA ER-2 airplane)

This chapter focuses on comparing the SVM classification method and two common methods i.e., spectral angle matching method and maximum likelihood method by this image.

2.4.1 *Performance Assessment for Original SVM-Based Classification*

The experiment will make the detailed comparison on the classification performance of various methods by transforming the training numbers and sample dimension. In Experiment 1, the selected number of classes is six classes, involving corn, soybean, grass, forest land, hay, and wheat. The total number of the training samples is 1031, and the number of the inspection samples is 5144. SAM classification method and ML classification method are used for comparing with the SVM classification method with different kernel functions. Table 2.5 and Fig. 2.11 provide the result of SVM (Gaussian kernel)-based classification method and the thematic mapping-based image representation in details. The experimental result comparison of various methods is shown in Table 2.6. The result shows that the SVM classification method has the best classification effect, and the SAM method has the worst effect. In the SVM classification method, the Gaussian kernel SVM effect is the best, and the linear kernel SVM effect is relatively low.

Experiment 2 selects 400 training samples (100 samples of corn, meadow, soybean and forest land, respectively) and 320 test samples (80 samples of corn, meadow, soybean and forest land, respectively) from four land object classes, and

Table 2.5 Classification accuracy of each class (%)

Classes	Corn	Soybean	Grass	Forest land	Hay	Wheat
Classification accuracy	97.7	99.0	96.2	99.54	99.59	99.06

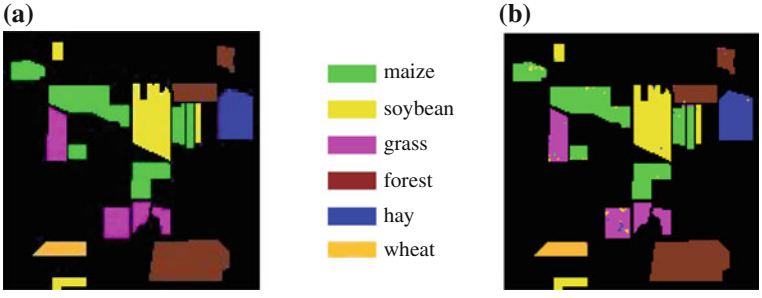

Fig. 2.11 Real land object chart and classification chart. **a** Real class. **b** SVM (Gauss)-based classification result

Table 2.6 Comparison of classification accuracy

Classification accuracy (%)		Classification method				
		SAM	ML	Linear kernel SVM	Polynomial kernel SVM	Gaussian kernel SVM
Number of samples	1031, 5144	79.35	95.63	96.64	97.65	98.46
	400, 320	71.88	82.81	88.13	93.44	96.56
	50, 320	71.25	–	82.19	83.44	85.31

then reduce the dimension to 50 in the wavelet fusion method. The experiment gains the similar result to Experiment 1 (see Table 2.6). In Experiment 3, the training sample in Experiment 2 is reduced to 50. In this case, the classification effect of SVM method still keeps the best, while ML method cannot be implemented due to less training samples.

The experiment shows that the classification accuracy gained from the maximum likelihood classification method is generally higher than SAM method, but the number of the training samples cannot be insufficient (theoretically, it should be more than the spectral dimension, while actually it is requested to be more). The classification accuracy gained from the SVM-based classification method is generally higher than the maximum likelihood classification method. In SVM, the efficiency of Gaussian kernel function is generally the maximum, while the efficiency of the linear kernel function is relatively poor. The experiment result shows clearly the excellent performance of SVM.

2.4.2 *Performance Assessment for Multi-class Classifier with the Simplest Structure*

It still adopts the agriculture and forestry remote-sensing area in Indiana of America. Four types of the land objects in the real chart are selected for the classification experiment. There are 400 pairs of training samples, and 320 pairs of test samples. The experiment adopts the least squares support vector machine of Gaussian kernel function and the efficient SMO algorithm. The iterative operation does not store the kernel function. At the same time, it adopts the 1-a-r method and 1-a-1 method for future reference. Table 2.7 provides the training time and test time used under the same iteration termination standard and different methods, with the time unit of second. The classification accuracy gained from the proposed method is 93.75 %, and the classification accuracy of two reference methods is, respectively,

Table 2.7 Comparison of training time and test time under structure of three classifiers

	1-a-r	1-a-1	Proposed method
Training time	81.7500	59.7970	37.5780
Test time	14.4702	11.21870	7.4288

94.69 and 94.37 %. The experiment result shows that the time as required during the training and testing by the algorithm constructed on the basis of the classifier structure proposed in this book is less than that of the two traditional methods, while the classification accuracy is reduced for less than 1 %. The experiment result fully verifies the previous theoretical analysis.

2.4.3 *Performance Assessment for Triply Weighted Classification*

The first group of the experimental samples is combined by the land object data of three classes of 3, 8, and 11 (the number of pixels is successively 834, 489, and 2468) in the agriculture and forestry remote-sensing image in Indiana. Extract the spectral characteristics of some pixels as the training sample, and the entire class data as the test sample. Adopt the unweighted, sample weighting, feature weighting, class weighting mode, and triply weighting mode successively for the effect test. The classification result is shown successively in Fig. 2.12a–e. In the experiment,

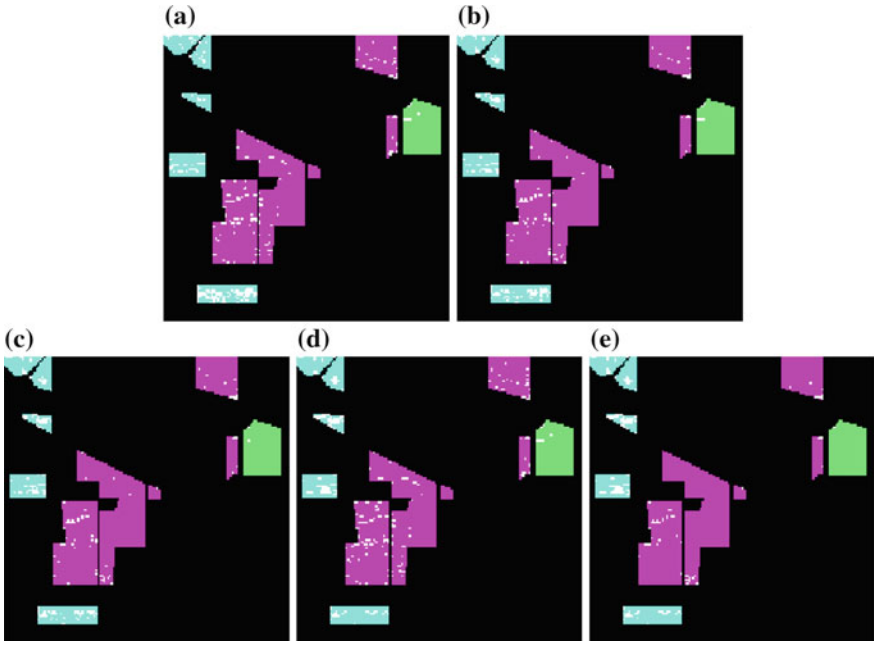

Fig. 2.12 Classification result chart under different weighting conditions in the first group of classification experiment. **a** Unweighted classification result chart. **b** Classification result chart of sample weighting. **c** Classification result chart of feature weighting. **d** Classification result chart of class weighting. **e** Classification result chart of triply weighting

SVM adopts the Gaussian kernel function, and the training samples are taken from the front 100 pixels of each class. In the class weighting experiment, the weights of three classes are successively set as 1, 5, and 10, i.e., focusing on considering the classification effect of Class 3. In the classification result, the above three classes are successively marked by different colors. The pixel of the classification error in the image is displayed by the white dot. The experiment result indicates that, using the sample weighting and feature weighting methods can enhance the entire classification accuracy to varying degrees, while the class weighting method can improve the analysis effect of the class corresponding to the relatively small weight (meanwhile reduce the analysis effect of the class corresponding to the relatively large weight). Although the triply weighting classification analysis effect of Class 3 is not as good as the effect of independently applying the class weighting, using three weighting methods at the same time can reach better analysis effect in general.

The second group of the experiment selects the land objects in three classes of 2, 6, and 10, and the number of pixels is successively 1434, 747, and 968. The experimental mode is ditto, and the classification result is shown in Fig. 2.13. The objective evaluation indexes of the above two groups of the experiments are respectively shown in Tables 2.8 and 2.9. The soft classification error means the absolute error mean statistics between the SVM decision result without the

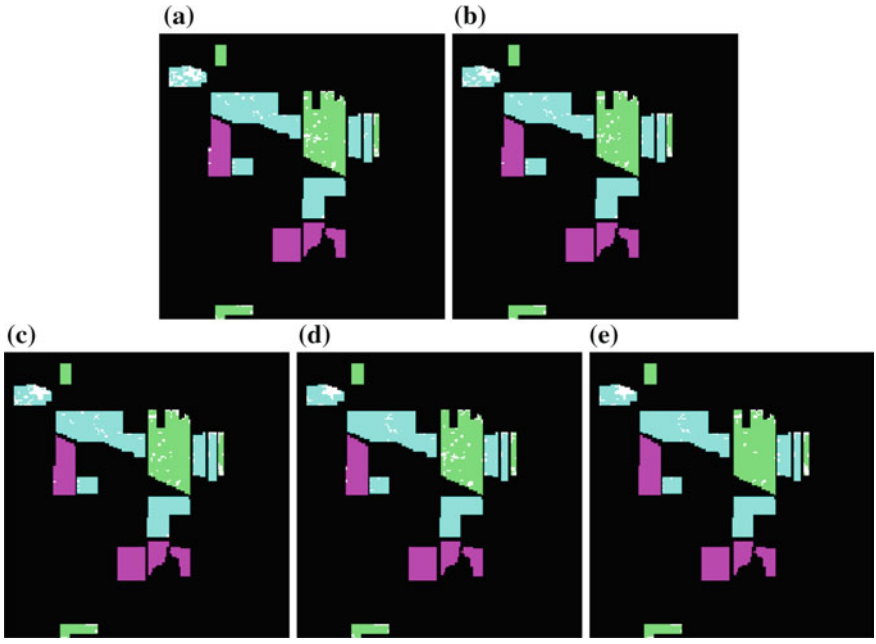

Fig. 2.13 Classification result chart under different weighting conditions in the second group of classification experiment. **a** Unweighted classification result chart. **b** Classification result chart of sample weighting. **c** Classification result chart of feature weighting. **d** Classification result chart of class weighting. **e** Classification result chart of triply weighting

Table 2.8 Number of misclassification pixels in the first group of classification experiment

Land object class	Unweighted	Sample weighting	Feature weighting	Class weighting	Triply weighting
Class 3	165/0.2248	145/0.1992	138/0.1914	130/0.1880	133/0.1897
Class 8	7/0.2248	6/0.2248	2/0.2248	7/0.2248	1/0.2248
Class 11	136/0.2248	87/0.2248	70/0.2248	147/0.2248	54/0.2248

Table 2.9 Number of misclassification pixels in the second group of classification experiment

Land object class	Unweighted	Sample weighting	Feature weighting	Class weighting	Triply weighting
Class 2	114/0.125	107/0.125	104/0.125	83/0.125	82/0.125
Class 6	2/0.125	1/0.125	1/0.125	3/0.125	0/0.125
Class 10	89/0.125	82/0.125	79/0.125	90/0.125	82/0.125

two-value quantization and the supervised classification result. This mode is more accurate than the traditional hard classification accuracy statistics. The experimental result further affirms that the weighting method has the effect.

2.5 Chapter Conclusions

One of the main contents in this chapter is to put forward a classifier with the simplest structure, which can greatly reduce the complexity of the classifier. There are many advantages, such as reducing the training time, reducing the test time, lowering the complexity of programming, and reducing the number of the sub-classifiers, to make the independent parameter adjustment in each decision function possible. It is noted that the advantage from the proposed method is at the cost of sacrificing the smaller classification accuracy. In the target classification problem, classification accuracy and classification speed are usually a pair of contradictory indexes. During solving the practical problem, which classifier is adopted should be determined as per the requirements of the user. For the problem with higher requirements on the classification speed such as the real-time application of SVM, the method proposed in this book is very effective. In the case of comprehensively considering the classification accuracy and classification speed, the traditional classifier and the classifier proposed in this book can be combined into the mixed classifier to coordinate the demand contradiction between the two.

The other main content in this chapter is the proposed multiple weighting classification method, to map the nonlinear distance into the corresponding weight to finish the sample weighting in accordance with the relationship between the sample anomaly degree and the distance of the sample deviating from the class center. According to the weighting characteristics of the intra-class divergence matrix on the linear spectral separation problem, it is promoted to LSSVM

classification problem to finish the feature weighting. According to the special meaning of the unit matrix diagonal element in the system of LSSVM linear equation, it is set as different numerical values of reflecting the importance of the class, to finish the class weighting. In three weighting methods, the sample weighting is the way specially implemented on the training sample, and the feature weighting is the operation on all data, while the class weighting is the resetting of the matrix diagonal element during the training process. Three weighting methods can be used independently, and also can be used in combination by any check mode. In the practical application, it can be selected as per the specific demand.

In addition, for the mass data problem, the dominant solution method of LSSVM is perplexed by the aspects of the data storage. SMO optimization solution algorithm proposed in this chapter, as the substitute way, is conducive to solving this problem.

References

Chang C-I, Ji B (2006) Weighted abundance constrained linear spectral mixture analysis. IEEE Trans Geosci Remote Sens 44:378–388

Chen C-H, Tu T-M (1996) Computation reduction of the maximum likelihood classifier using the winograd identity. Pattern Recognit 29(7):1213–1220

Cristianini N, Shawe-Taylor J (2004) Support vector machine introduction. In: Guozheng L, Meng W, Huajun Z (eds). Electronic Industry Press, Beijing

Emami H Introducing correctness coefficient as an accuracy measure for sub pixel classification results. http://www.ncc.org.ir/articles/poster83/H.Emami.pdf

Ji B, Chang Chein-I, Jensen JO, Jensen JL (2004) Unsupervised constrained linear Fisher's discriminant analysis for hyperspectral image classification. In: 49th annual meeting, SPIE international symposium optical science and technology, vol 49. Imaging spectrometry IX (AM105), Denver, CO, pp 2–4

Jia XP, Richards JA (1994) Efficient maximum likelihood classification for imaging spectrometer data sets. IEEE Trans Geosci Remote Sens 32(2):274–281

Karpinski M, Werther T (1989) VC dimension and uniform learnability of sparse polynomials and rational functions. SIAM J Computing. Preprint 8537-CS, Bonn University

Richards JA, Jia XP (2006) Remote sensing digital image analysis, 3rd edn. Springer, Berlin

Shevade SK, Keerthi SS, Bhattacharyya C et al (2000) Improvements to the SMO algorithm for SVM regression. IEEE Trans on Neural Networks 11(5):1188–1193

Sohn Y, Rebello NS (2002) Supervised and unsupervised spectral angle classifiers. Photogram Eng Remote Sens 68(12):1271–1280

Song Q, Hu WJ, Xie WF (2002) Robust support vector machine with bullet hole image classification. IEEE Trans Systems Man and Cybern Part C 32:440–448

Suykens JAK, Brabanter JD, Lukas L, Vandewalle J (2002) Weighted least squares support vector machines: robustness and sparse approximation. Neurocomputing 48(1–4):85–105

Vapnik VN (2000) The nature of statistical learning theory. In: Zhang X (ed). Tsinghua University press, Beijing

http://www.springer.com/978-3-662-47455-6

Hyperspectral Image Processing
Wang, L.; Zhao, C.
2016, XVII, 315 p. 121 illus., 15 illus. in color.,
Hardcover
ISBN: 978-3-662-47455-6