Chapter 7

NONPARAMETRIC CLASSIFICATION AND ERROR ESTIMATION

After studying the nonparametric density estimates in Chapter 6, we are now ready to discuss the problem of how to design *nonparametric classifiers* and estimate their *classification errors*.

A nonparametric classifier does not rely on any assumption concerning the structure of the underlying density function. Therefore, the classifier becomes the *Bayes classifier* if the density estimates converge to the true densities when an infinite number of samples are used. The resulting error is the *Bayes error*, the smallest achievable error given the underlying distributions. As was pointed out in Chapter 1, the Bayes error is a very important parameter in pattern recognition, assessing the classifiability of the data and measuring the discrimination capabilities of the features even before considering what type of classifier should be designed. The selection of features always results in a loss of classifiability. The amount of this loss may be measured by comparing the Bayes error in the feature space with the Bayes error in the original data space. The same is true for a classifier. The performance of the classifier may be compared with the Bayes error in the original data space. However, in practice, we never have an infinite number of samples, and, due to the finite sample size, the density estimates and, subsequently, the estimate of the Bayes error have large biases and variances, particularly in a high-dimensional space.

A similar trend was observed in the parametric cases of Chapter 5, but the trend is more severe with a nonparametric approach. These problems are addressed extensively in this chapter.

Both *Parzen* and *kNN* approaches will be discussed. These two approaches offer similar algorithms for classification and error estimation, and give similar results. Also, the *voting kNN procedure* is included in this chapter, because the procedure is very popular, although this approach is slightly different from the *kNN* density estimation approach.

7.1 General Discussion

Parzen Approach

Classifier: As we discussed in Chapter 3, the *likelihood ratio classifier* is given by $-\ln p_1(X)/p_2(X) \gtrless t$, where the threshold t is determined in various ways depending on the type of classifier to be designed (e.g. Bayes, Neyman-Pearson, minimax, etc.). In this chapter, the true density functions are replaced by their estimates discussed in Chapter 6. When the *Parzen density estimate* with a *kernel function* $\kappa_i(\cdot)$ is used, the likelihood ratio classifier becomes

$$-\ln\frac{\hat{p}_1(X)}{\hat{p}_2(X)} = -\ln\frac{\frac{1}{N_1}\sum_{j=1}^{N_1}\kappa_1(X-\mathbf{X}_j^{(1)})}{\frac{1}{N_2}\sum_{j=1}^{N_2}\kappa_2(X-\mathbf{X}_j^{(2)})} \underset{\omega_2}{\overset{\omega_1}{\gtrless}} t \,, \tag{7.1}$$

where $\mathbf{S} = \{\mathbf{X}_1^{(1)}, \ldots, \mathbf{X}_{N_1}^{(1)}, \mathbf{X}_1^{(2)}, \ldots, \mathbf{X}_{N_2}^{(2)}\}$ is the given data set. Equation (7.1) classifies a test sample X into either ω_1 or ω_2, depending on whether the left-hand side is smaller or larger than a threshold t.

Error estimation: In order to estimate the error of this classifier from the given data set, $\mathbf{S}$, we may use the *resubstitution* (R) and *leave-one-out* (L) methods to obtain the lower and upper bounds for the Bayes error. In the R method, all available samples are used to design the classifier, and the same sample set is tested. Therefore, when a sample $\mathbf{X}_k^{(1)}$ from ω_1 is tested, the following equation is used.

$$-\ln\frac{\frac{1}{N_1}\sum_{j=1}^{N_1}\kappa_1(\mathbf{X}_k^{(1)}-\mathbf{X}_j^{(1)})}{\frac{1}{N_2}\sum_{j=1}^{N_2}\kappa_2(\mathbf{X}_k^{(1)}-\mathbf{X}_j^{(2)})} \underset{\omega_2}{\overset{\omega_1}{\gtrless}} t \qquad (R \text{ method}) . \tag{7.2}$$

If $<$ is satisfied, $\mathbf{X}_k^{(1)}$ is correctly classified, and if $>$ is satisfied, $\mathbf{X}_k^{(1)}$ is misclassified. The R estimate of the ω_1-error, ε_{1R}, is obtained by testing $\mathbf{X}_1^{(1)},\ldots,\mathbf{X}_{N_1}^{(1)}$, counting the number of misclassified samples, and dividing the number by N_1. Similarly, ε_{2R} is estimated by testing $\mathbf{X}_1^{(2)},\ldots,\mathbf{X}_{N_2}^{(2)}$.

On the other hand, when the L method is applied to test $\mathbf{X}_k^{(1)}$, $\mathbf{X}_k^{(1)}$ must be excluded from the design set. Therefore, the numerator of (7.2) must be replaced by

$$\hat{\mathbf{p}}_{1L}(\mathbf{X}_k^{(1)}) = \frac{1}{N_1-1}\left[\sum_{j=1}^{N_1}\kappa_1(\mathbf{X}_k^{(1)}-\mathbf{X}_j^{(1)}) - \kappa_1(\mathbf{X}_k^{(1)}-\mathbf{X}_k^{(1)})\right] . \tag{7.3}$$

Again, $\mathbf{X}_k^{(1)}$ $(k=1,\ldots,N_1)$ are tested and the misclassified samples are counted. Note that the amount subtracted in (7.3), $\kappa_1(0)$, does not depend on k. When an ω_2-sample is tested, the denominator of (7.2) is modified in the same way.

Typical kernel functions, such as (6.3), generally satisfy $\kappa_i(0) \geq \kappa_i(Y)$ (and subsequently $\kappa_i(0) \geq \hat{\mathbf{p}}_i(Y)$). Then,

$$\hat{\mathbf{p}}_{1L}(\mathbf{X}_k^{(1)}) = \hat{\mathbf{p}}_1(\mathbf{X}_k^{(1)}) + \frac{1}{N_1-1}[\hat{\mathbf{p}}_1(\mathbf{X}_k^{(1)}) - \kappa_1(0)] \leq \hat{\mathbf{p}}_1(\mathbf{X}_k^{(1)}) . \tag{7.4}$$

That is, the L density estimate is always smaller than the R density estimate. Therefore, the left-hand side of (7.2) is larger in the L method than in the R method, and consequently $\mathbf{X}_k^{(1)}$ has more of a chance to be misclassified. Also, note that the L density estimate can be obtained from the R density estimate by simple scalar operations - subtracting $\kappa_1(0)$ and dividing by (N_1-1). Therefore, the computation time needed to obtain both the L and R density estimates is almost the same as that needed for the R density estimate alone.

kNN Approach

Classifier: Using the *kNN* density estimate of Chapter 6, the likelihood ratio classifier becomes

$$-\ln\frac{\hat{\mathbf{p}}_1(X)}{\hat{\mathbf{p}}_2(X)} = -\ln\frac{(k_1-1)N_2\mathbf{v}_2(X)}{(k_2-1)N_1\mathbf{v}_1(X)}$$

$$= -n\ \ln\frac{d_2(\mathbf{X}^{(2)}_{k_2NN},X)}{d_1(\mathbf{X}^{(1)}_{k_1NN},X)} - \ln\frac{(k_1-1)N_2|\Sigma_2|^{1/2}}{(k_2-1)N_1|\Sigma_1|^{1/2}} \underset{\omega_2}{\overset{\omega_1}{\gtrless}} t\ , \tag{7.5}$$

where $v_i = \pi^{n/2}\Gamma^{-1}(n/2+1)|\Sigma_i|^{1/2}d_i^n$ from (B.1), and $d_i^2(Y,X) = (Y-X)^T\Sigma_i^{-1}(Y-X)$. In order to classify a test sample X, the k_1*th NN* from ω_1 and the k_2*th NN* from ω_2 are found, the distances from X to these neighbors are measured, and these distances are inserted into (7.5) to test whether the left-hand side is smaller or larger than t. In order to avoid unnecessary complexity, $k_1 = k_2$ is assumed in this chapter.

Error estimation: The classification error based on a given data set S can be estimated by using the L and R methods. When $\mathbf{X}_k^{(1)}$ from ω_1 is tested by the R method, $\mathbf{X}_k^{(1)}$ must be included as a member of the design set. Therefore, when the *kNN*'s of $\mathbf{X}_k^{(1)}$ are found from the ω_1 design set, $\mathbf{X}_k^{(1)}$ itself is included among these *kNN*'s. Figure 7-1 shows how the *kNN*'s are selected and how the distances to the kth *NN*'s are measured for $k = 2$. Note in Fig. 7-1 that the locus of points equidistant from $X_k^{(1)}$ becomes ellipsoidal because the distance is normalized by Σ_i. Also, since $\Sigma_1 \neq \Sigma_2$ in general, two different ellipsoids are used for ω_1 and ω_2. In the R method, $\mathbf{X}_k^{(1)}$ and $\mathbf{X}_{NN}^{(1)}$ are the nearest and second nearest neighbors of $\mathbf{X}_k^{(1)}$ from ω_1, while $\mathbf{X}_{NN}^{(2)}$ and $X_{2NN}^{(2)}$ are the nearest and second nearest neighbors of $\mathbf{X}_k^{(1)}$ from ω_2. Thus,

$$-\ln\frac{\hat{\mathbf{p}}_{1R}(\mathbf{X}_k^{(1)})}{\hat{\mathbf{p}}_2(\mathbf{X}_k^{(1)})} = -n\ \ln\frac{d_2(\mathbf{X}^{(2)}_{2NN},\mathbf{X}_k^{(1)})}{d_1(\mathbf{X}^{(1)}_{NN},\mathbf{X}_k^{(1)})} - \ln\frac{N_2|\Sigma_2|^{1/2}}{N_1|\Sigma_1|^{1/2}} \underset{\omega_2}{\overset{\omega_1}{\gtrless}} t \quad (R \text{ method})\ . \tag{7.6}$$

On the other hand, in the L method, $\mathbf{X}_k^{(1)}$ is no longer considered a member of the design set. Therefore, $\mathbf{X}_{NN}^{(1)}$ and $\mathbf{X}_{2NN}^{(1)}$ are selected as the nearest and second nearest neighbors of $\mathbf{X}_k^{(1)}$ from ω_1. The selection of ω_2 neighbors is the same as before. Thus,

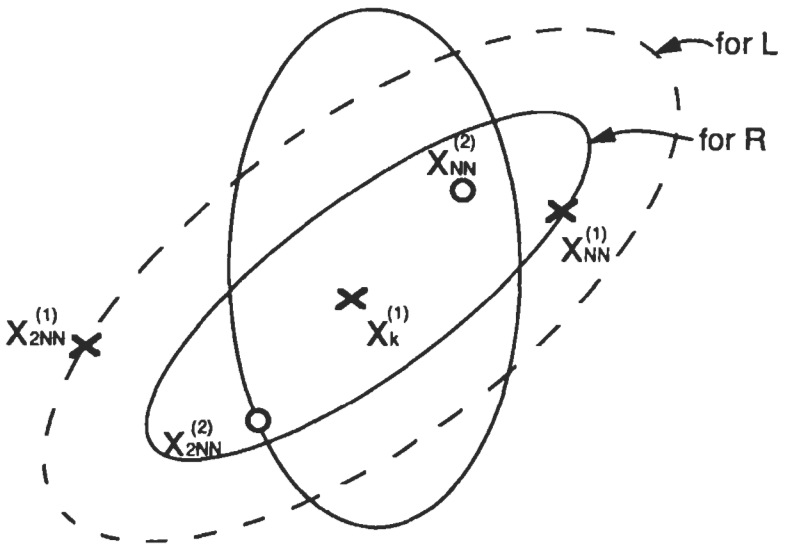

Fig. 7-1 Selection of neighbors.

$$-\ln\frac{\hat{p}_{1L}(\mathbf{X}_k^{(1)})}{\hat{p}_2(\mathbf{X}_k^{(1)})} = -n\ \ln\frac{d_2(\mathbf{X}_{2NN}^{(2)},\mathbf{X}_k^{(1)})}{d_1(\mathbf{X}_{2NN}^{(1)},\mathbf{X}_k^{(1)})} - \ln\frac{N_2|\Sigma_2|^{1/2}}{N_1|\Sigma_1|^{1/2}} \underset{\omega_2}{\overset{\omega_1}{\gtrless}} t \quad (L \text{ method}) . \tag{7.7}$$

Obviously, $d_1(\mathbf{X}_{2NN}^{(1)},\mathbf{X}_k^{(1)}) \geq d_1(\mathbf{X}_{NN}^{(1)},\mathbf{X}_k^{(1)})$, making the left-hand side of (7.7) larger than the left-hand side of (7.6). Thus, $\mathbf{X}_k^{(1)}$ is more likely to be misclassified in the L method than in R method.

Also, note that, in order to find the NN sample, the distances to all samples must be computed and compared. Therefore, when $d_1(\mathbf{X}_{NN}^{(1)},\mathbf{X}_k^{(1)})$ is obtained, $d_1(\mathbf{X}_{2NN}^{(1)},\mathbf{X}_k^{(1)})$ must also be available. This means that the computation time needed to get both the L and R results is practically the same as the time needed for the R method alone.

Voting *kNN* Procedure

The kNN approach mentioned above can be modified as follows. Instead of selecting the kth NN from each class separately and comparing the distances, the kNN's of a test sample are selected from the mixture of classes, and the

number of neighbors from each class among the k selected samples is counted. The test sample is then classified to the class represented by a majority of the *kNN*'s. That is,

$$\mathbf{k}_i = \max\{\mathbf{k}_1, \ldots, \mathbf{k}_L\} \quad \rightarrow \quad X \in \omega_i \tag{7.8}$$

$$\mathbf{k}_1 + \ldots + \mathbf{k}_L = k \ ,$$

where $\mathbf{k}_i$ is the number of neighbors from ω_i $(i = 1, \ldots, L)$ among the *kNN*'s. In order to avoid confusion between these two *kNN* procedures, we will call (7.8) the *voting kNN* procedure and (7.5) the *volumetric kNN* procedure.

For the voting *kNN* procedure, it is common practice to use the same metric to measure the distances to samples from all classes, although each class could use its own metric. Since the k_i's are integers and a ranking procedure is used, it is hard to find a component of (7.8) analogous with the threshold of (7.5).

It can be shown that the volumetric *kNN* and voting $(2k-1)$*NN* procedures give identical classification results for the two-class problem using the same metric for both classes. For example, let k and $(2k-1)$ be 3 and 5 respectively. In the voting 5*NN* procedure, a test sample is classified to ω_1, if 3, 4, or 5 of the 5*NN*'s belong to ω_1. This is equivalent to saying that the 3rd *NN* from ω_1 is closer to the test sample than the 3rd *NN* from ω_2.

7.2 Voting *kNN* Procedure—Asymptotic Analysis

In this section, let us study the expected performance of the voting *kNN* procedure, first for the *asymptotic* case $(N_i = \infty)$ and later for the *finite sample* case.

Two-Class *kNN*

NN: We start our discussion with the simplest case, setting $k = 1$ and $L = 2$ in (7.8). That is, in order to classify a test sample, $\mathbf{X}$, the *NN* sample $\mathbf{X}_{NN}$ is found. Then, $\mathbf{X}$ is classified to either ω_1 or ω_2, depending on the class membership of $\mathbf{X}_{NN}$. An error occurs when $\mathbf{X} \in \omega_1$ but $\mathbf{X}_{NN} \in \omega_2$, or when $\mathbf{X} \in \omega_2$ but $\mathbf{X}_{NN} \in \omega_1$. Therefore, the *conditional risk* given X and X_{NN} is expressed by

$$r_1(X,X_{NN})=Pr\left\{\{\mathbf{X}\varepsilon\omega_1 \text{ and } \mathbf{X}_{NN}\varepsilon\omega_2\} \text{ or } \{\mathbf{X}\varepsilon\omega_2 \text{ and } \mathbf{X}_{NN}\varepsilon\omega_1\} \mid X,X_{NN}\right\}$$
$$=Pr\{\mathbf{X}\varepsilon\omega_1 \text{ and } \mathbf{X}_{NN}\varepsilon\omega_2 \mid X,X_{NN}\}+Pr\{\mathbf{X}\varepsilon\omega_2 \text{ and } \mathbf{X}_{NN}\varepsilon\omega_1 \mid X,X_{NN}\}$$
$$=q_1(X)q_2(X_{NN})+q_2(X)q_1(X_{NN}) \tag{7.9}$$

where

$$q_i(X) = Pr\{X\varepsilon\omega_i \mid X\} : \textit{a posteriori probability} . \tag{7.10}$$

The 2nd line of (7.9) is obtained because the two events in the first line are mutually exclusive. The 3rd line is obtained because $\mathbf{X}$ and $\mathbf{X}_{NN}$ are mutually independent. When an infinite number of samples is available, $\mathbf{X}_{NN}$ is located so close to $\mathbf{X}$ that $q_i(X_{NN})$ can be replaced by $q_i(X)$. Thus, the *asymptotic conditional risk* of the *NN* method is

$$r_1^*(X) = 2q_1(X)q_2(X) = 2\xi(X) \tag{7.11}$$

where

$$\xi(X) = q_1(X)q_2(X) . \tag{7.12}$$

2NN: When k is even, $\mathbf{k}_1 = \mathbf{k}_2$ may occur and a decision cannot be made. In this case, we may set a rule that X be rejected and not counted as an error. In the simplest case of $k = 2$, the rejection occurs when $X_{NN}\varepsilon\omega_1$ and $X_{2NN}\varepsilon\omega_2$, or $X_{NN}\varepsilon\omega_2$ and $X_{2NN}\varepsilon\omega_1$. On the other hand, X is misclassified, when $X\varepsilon\omega_1$ but $X_{NN},X_{2NN}\varepsilon\omega_2$, or $X\varepsilon\omega_2$ but $X_{NN},X_{2NN}\varepsilon\omega_1$. Therefore, the conditional risk is

$$r_2(X,X_{NN},X_{2NN})=q_1(X)q_2(X_{NN})q_2(X_{2NN})+q_2(X)q_1(X_{NN})q_1(X_{2NN}) . \tag{7.13}$$

For the asymptotic case with $q_i(X) = q_i(X_{NN}) = q_i(X_{2NN})$,

$$r_2^*(X) = q_1(X)q_2(X) = \xi(X) \tag{7.14}$$

where $q_1(X) + q_2(X) = 1$ is used.

kNN: Extending the above discussion to larger values of k, the asymptotic conditional risks for odd k and even k are

$$r^*_{2k-1}(X) = \sum_{i=1}^{k} \frac{1}{i}\binom{2i-2}{i-1}\xi^i(X) + \frac{1}{2}\binom{2k}{k}\xi^k(X) , \tag{7.15}$$

$$r^*_{2k}(X) = \sum_{i=1}^{k} \frac{1}{i}\binom{2i-2}{i-1}\xi^i(X) . \tag{7.16}$$

On the other hand, the *conditional Bayes risk* given X is

$$r^*(X) = \min[q_1(X), q_2(X)] = \frac{1}{2} - \frac{1}{2}\sqrt{1-4\xi(X)}$$

$$= \sum_{i=1}^{\infty} \frac{1}{i}\binom{2i-2}{i-1}\xi^i(X) , \tag{7.17}$$

where the 2nd line is the MacLaurin series expansion of the first line. Using (7.15)-(7.17), it is not difficult to prove that these conditional risks satisfy the following inequalities, regardless of ξ [1].

$$\frac{1}{2}r^* \le r^*_2 \le r^*_4 \le \ldots \le r^* \le \ldots \le r^*_3 \le r^*_1 \le 2r^* . \tag{7.18}$$

The proof for $r^* \le r^*_1$ was given in (3.157). Figure 7-2 shows these risks as functions of ξ. The inequalities of (7.18) can also be seen in Fig. 7-2. In addition, $\sqrt{\xi}$ is plotted in Fig. 7-2, because $E\{\sqrt{\xi(\mathbf{X})}\}$ is the Bhattacharyya bound of the Bayes error. Figure 7-2 shows that the *kNN* risks are better bounds than the Bhattacharyya bound. Taking the expectation of these risks with respect to $\mathbf{X}$, the corresponding errors can be obtained. Therefore, these errors also satisfy the inequalities of (7.18). Thus,

$$\frac{1}{2}\varepsilon^* \le \varepsilon^*_{2NN} \le \varepsilon^*_{4NN} \le \ldots \le \varepsilon^* \le \ldots \le \varepsilon^*_{3NN} \le \varepsilon^*_{NN} \le 2\varepsilon^* , \tag{7.19}$$

where

$$\varepsilon^* = E\{r^*(\mathbf{X})\} \quad \text{and} \quad \varepsilon^*_{kNN} = E\{r^*_k(\mathbf{X})\} . \tag{7.20}$$

Equation (7.19) indicates that the error of the voting *NN* procedure is less than twice the Bayes error. This is remarkable, considering that the procedure does not use any information about the underlying distributions and only the class of the single nearest neighbor determines the outcome of the decision.

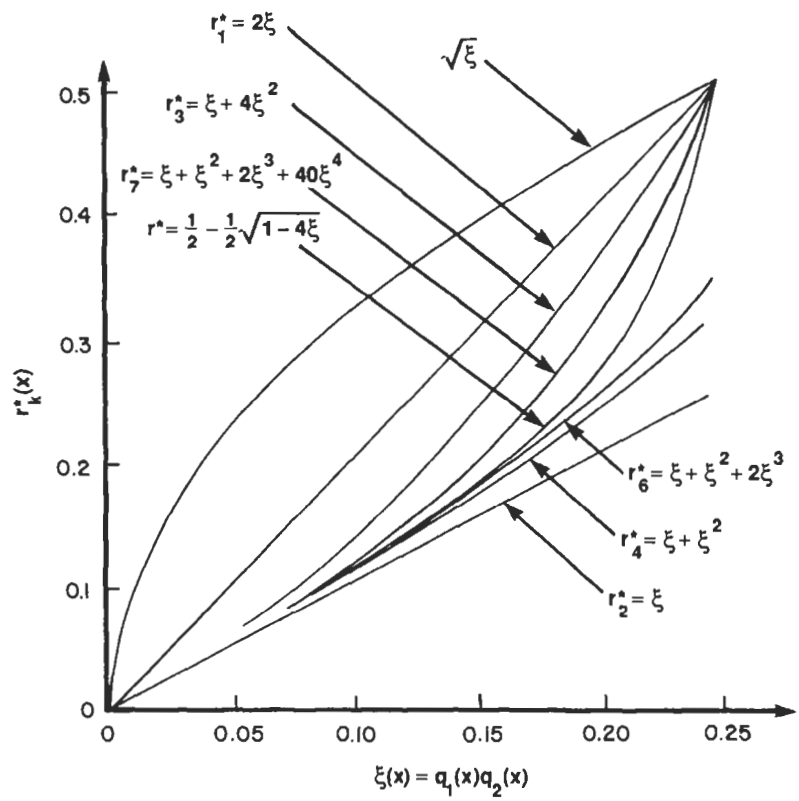

Fig. 7-2 Asymptotic risks vs. ξ.

Example 1: Figure 7-3 gives a simple example to demonstrate how the voting *NN* procedure produces an error between the Bayes error and twice the Bayes error. If the true Bayes classifier is known, samples 5 and 6 from ω_1 and samples 1 and 3 from ω_2 are misclassified. By the voting *NN* procedure, these four samples are indeed misclassified, because their *NN*'s are from the other classes. However, some of these misclassified samples (1 from ω_2 and 5 from ω_1) become the *NN*'s of samples from the other classes (2 from ω_1 and 4 from ω_2), and produce additional errors (2 and 4). This may (for 1 and 5) or may not (for 3 and 6) occur, depending on the distribution of samples. Therefore, roughly speaking, the *NN* error is somewhere between the Bayes error and twice the Bayes error. Also, Fig. 7-3 shows that only 3 samples are misclassified by the voting 2*NN* procedure. For samples 3, 4, and 5, the votes are split and the samples are rejected.

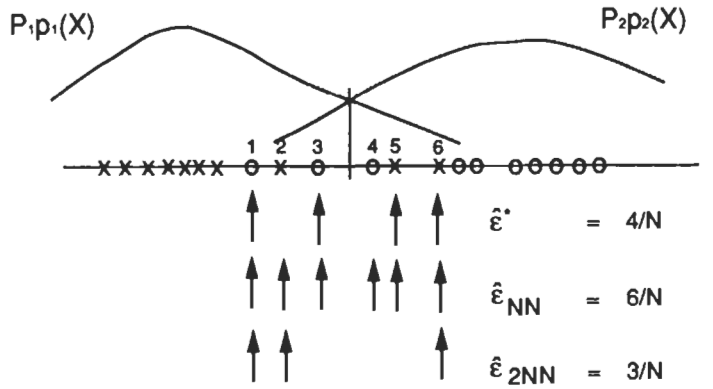

Fig. 7-3 Example of *kNN* classification.

Multiclass *NN*

The voting *NN* procedure can also be applied to general L-class problems, in which a test sample is classified to the class of the *NN* sample. The asymptotic conditional risk is

$$r_1^*(X) = q_1(X)\sum_{\substack{j=1\\ j\neq 1}}^{L} q_j(X) + \ldots + q_L(X)\sum_{\substack{j=1\\ j\neq L}}^{L} q_j(X)$$

$$= \sum_{i=1}^{L} q_i(X)[1-q_i(X)] = 1 - \sum_{i=1}^{L} q_i^2(X) . \tag{7.21}$$

On the other hand, the Bayes conditional risk is

$$r^*(X) = 1 - \max_j\{q_j(X)\} = 1 - q_m(X) . \tag{7.22}$$

Using the Schwartz's inequality,

$$(L-1)\sum_{\substack{j=1\\ j\neq m}}^{L} q_j^2(X) \geq [\sum_{\substack{j=1\\ j\neq m}}^{L} q_j(X)]^2 = [1-q_m(X)]^2 = r^{*2}(X) . \tag{7.23}$$

Adding $(L-1)q_m^2(X)$ to both sides,

$$(L-1)\sum_{j=1}^{L} q_j^2(X) \geq r^{*2}(X) + (L-1)[1-r^*(X)]^2 . \tag{7.24}$$

Substituting (7.24) into (7.21) [1],