THESIS REPORT
Ph.D.

SYSTEMS
RESEARCH
CENTER

Supported by the National Science Foundation Engineering Research Center Program (NSFD CD 8803012), Industry and the University

Nonparametric Classification Using Learning Vector Quantization

by A. LaVigna
Advisor: J.S. Baras

Ph.D. 90-1
Formerly TR 90-8

Nonparametric Classification using Learning Vector Quantization

by

Anthony LaVigna

Dissertation submitted to the Faculty of the Graduate School of
the University of Maryland in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy
1989

Advisory Committee:

Professor John S. Baras, Chairman/Advisor
Professor Carlos A. Berenstein
Associate Professor Nariman Farvardin
Associate Professor Evaggelos Geraniotis
Associate Professor Shihab Shamma

Dedication

To my wife Ann,
for her love and support

Acknowledgements

I would like to first express my sincere gratitude to my advisor Professor John S. Baras. He has provided me with guidance, support and encouragement. He has created more opportunities for me than I could have ever imagined. It has truly been an exciting time being one of his students.

I would like to thank Clif Penn for many stimulating discussions and Richard Wiggins for providing me the opportunity to do research at Texas Instruments.

I wish to thank my good friend Lahcen Saydy for his careful reading of my dissertation, his moral support, and his help in keeping my eye on the goal. He is truly a great friend. I also wish to thank Ouassima Akhrif for her moral support and encouragement.

My family deserves many thanks for their understanding and moral support, particularly my parents who encouraged me to pursue all my goals.

I gratefully acknowledge the financial support for this research from the Office of Naval Research through an ONR Fellowship, Texas Instruments through a TI/SRC Fellowship, and from Westinghouse and Naval Research Laboratory through research contracts. In addition, partial support was provided from NSF funds through the Systems Research Center.

Finally, I would like to thank a special person in my life, my wife Ann. She has endured the hard condition of being married to a graduate student. I cannot thank her enough for her understanding and love. She has truly been an inspiration.

Abstract

Title of Dissertation: *Nonparametric Classification using Learning Vector Quantization*

Anthony LaVigna, *Doctor of Philosophy, 1989*

Dissertation directed by: *John S. Baras*
Professor
Electrical Engineering Department

In this thesis we study several properties of Learning Vector Quantization. LVQ is a nonparametric detection scheme proposed in the neural network community by Kohonen. We examine it in detail, both theoretically and experimentally, to determine its properties as a nonparametric classifier. In particular, we study the convergence of the parameter adjustment rule in LVQ, we present a modification to LVQ which results in improving the convergence of the algorithm, we show that LVQ performs as well as other classifiers on two sets of simulations, and we show that the classification error associated with LVQ can be made arbitrarily small.

Contents

0 Introduction 1

1 Nonparametric Detection 4

1.1 Statistical Pattern Recognition 4

1.1.1 Bayes' Decision Rule for Minimum Error 5

1.1.2 Bayes' Decision Rule for Minimum Risk 5

1.1.3 Neyman–Pearson Test 7

1.2 Bayes' Risk Consistent Density Estimators 8

1.2.1 Definitions of Consistency 8

1.2.2 Convergence for Bayes Risk 9

1.3 Nonparametric Density Estimation 11

1.3.1 Histogram Method of Density Estimation 12

1.3.2 Nearest Neighbor Density Estimation 14

1.3.3 Kernel Density Estimation 14

1.3.4 Comparisons of Density Estimators 16

1.4 Vector Quantization 17

1.4.1 Convergence of the Estimated Cost 20

1.4.2 Relation to the Global Optimal Quantizer 22

1.5 Remarks 23

2 Review of Stochastic Approximation 24

2.1 The Heuristic Idea behind the ODE Method 24

2.2 Detailed Description of the Algorithm 27

2.3 Convergence in Probability of the Paths 30

2.4 Ljung–Type Convergence 35

2.5 Proof of Theorem 2.4.1 38

3 Learning Vector Quantization 43

3.1 Description of the Algorithm 46

3.2 Convergence to Stationary Points 48
3.2.1 Convergence for an Infinite Number of Observations 49
3.2.2 Convergence for a Finite Number of Observations 52
3.2.3 Remarks on Convergence 54
3.3 The Modified LVQ Algorithm 55
3.4 Generalization to Several Patterns 56
3.5 Decision Error 58
3.5.1 Nearest Neighbor 58
3.5.2 Other Choices for the Number of Voronoi Vectors 58
3.6 Initialization 60
3.6.1 The Number of Vectors 60
3.6.2 The Initial Locations 60
3.7 Application to Other Risks 61
3.8 Remarks 62

4 Simulations **63**
4.1 Simulation Setup 64
4.2 The Gaussian Examples 66
4.3 Rayleigh vs. Lognormal Examples 66
4.4 Analysis of the Results 68
4.4.1 Number of Voronoi Vectors 68
4.4.2 Number of Iterations 70
4.4.3 Size of the Learning Rate 71
4.4.4 Overall Performance 71
4.4.5 Sensitivity to Initial Conditions 72
4.5 Remarks 73
4.6 Results of Simulations 74

5 Discussion **88**
5.1 Implementation 89
5.2 Ergodic Input 91
5.3 Time Series Data 91
5.4 Further Issues 91

References **92**

List of Figures

3.1 *Plot of two pattern densities* 43
3.2 *Plot of decision regions* 44
3.3 *Absolute value of the difference of the pattern densities* 45
3.4 *Voronoi vectors and their Voronoi cells* 46
3.5 *A possible distribution of observations and two Voronoi vectors.* . . 54

5.1 *Architecture for implementing LVQ* 89

List of Tables

4.1 *Specifications of the Gaussian simulation set* 67
4.2 *Rayleigh and lognormal densities and their properties* 68
4.3 *Specifications of the non-Gaussian simulation set* 69
4.4 *Number of parameters in the adaptive histogram method* 70
4.5 *Performance of LVQ vs adaptive histogram* 71
4.6 *Performance of LVQ vs adaptive histogram and second order parametric* 72
4.7 *Gaussian simulation set with 20 observations* 74
4.8 *Gaussian simulation set with 50 observations* 75
4.9 *Gaussian simulation set with 100 observations* 76
4.10 *Gaussian simulation set with 200 observations* 77
4.11 *Gaussian simulation set with 10 iterations of LVQ* 78
4.12 *Gaussian simulation set with 20 iterations of LVQ* 79
4.13 *Gaussian simulation set with 40 iterations of LVQ* 80
4.14 *Non-Gaussian simulation set with 20 observations* 81
4.15 *Non-Gaussian simulation set with 50 observations* 82
4.16 *Non-Gaussian simulation set with 100 observations* 83
4.17 *Non-Gaussian simulation set with 200 observations* 84
4.18 *Non-Gaussian simulation set with 10 iterations of LVQ* 85
4.19 *Non-Gaussian simulation set with 20 iterations of LVQ* 86
4.20 *Non-Gaussian simulation set with 40 iterations of LVQ* 87

Chapter 0

Introduction

A common problem in signal processing is the problem of signal classification. In radar signal processing, it is the problem of determining the presence or absence of a target in the reflected signal. In adaptive control, it is the problem of determining the operating environment in order to use the appropriate gain in a gain scheduling algorithm. In both cases, a signal processor must be designed which correctly classifies a new observation based on past observations.

Loosely speaking, the general problem consists in extracting the necessary information in order to build a classifier which identifies each new observation with the lowest possible error, given past observations. As such, a classifier is nothing more than a partition of the observation space into disjoint regions; observations falling in the same region are declared to originate from the same pattern.

There are basically two approaches for solving this problem. The first one, referred to as the parametric approach, consists in using the past data to build a model and then using it in the classification scheme. The second approach, referred to as the nonparametric approach, consists in using the past data directly in the classification scheme. In the first approach, a statistical model is postulated *a priori* and its parameters are determined by minimizing a cost function which depends on the observation data and the assumed model. The success of the resulting classifier depends crucially on the nature of the assumed model, the characteristics of the cost function, and the accuracy of the parameters of the optimal model. Usually, simplifying assumptions are made on the model and the cost (e.g. Gaussian model and quadratic cost) in order to find an optimal solution.

Hence, a compromise exists between model accuracy and problem solvability.

In the second approach, a scheme is devised that uses past data directly in the classification scheme. New observations are classified by computing a suitable quantity which depends on the observation and comparing that quantity to similar ones computed from past observations. These tests are computed directly, without the intermediate step of identifying a statistical model. Among these tests are the nearest neighbor scheme, the kernel method, the histogram method, and the Learning Vector Quantization (LVQ) method. These tests do not assume any model form for the underlying problem. Consequently, they are not subject to the kinds of errors associated with assuming an incorrect model.

In this dissertation we prove several properties of the nonparametric classification scheme known as the LVQ method. The LVQ method, subsequently referred to as LVQ, originated in the neural network community and was introduced by Kohonen (Kohonen [1986]). Despite the considerable interest it has generated in the research community, most of the work related to LVQ is confined to pure simulations. Although this is a natural and important first step in the development of LVQ, we feel that an investigation of the theoretical underpinnings of the method is warranted. Our goal is to examine LVQ, both theoretically and experimentally, and determine its performance as a nonparametric classifier. More specifically, the following contributions are made:

- We prove the convergence of the parameter adjustment rule in LVQ under reasonable assumptions.
- We introduce a modification to LVQ which results in convergence in more cases.
- We show by means of simulation results that LVQ has a better overall performance than other classifiers.
- We show that the classification error associated with LVQ can be made asymptotically optimal in a sense to be specified later.

The main tools used to carry out this program originated from stochastic approximation. A judicious casting of LVQ as a stochastic approximation algorithm,

provides the general framework used throughout this dissertation to study LVQ.

In Chapter 1, we present a review of statistical classification schemes, nonparametric detection and vector quantization. In addition, a new result related to the convergence of a density estimate constructed from a vector quantizer is presented.

In Chapter 2, we review some stochastic approximation results that are pertinent to the present work.

In Chapter 3, the LVQ algorithm is presented. Using theorems from Chapter 2, we prove that the update algorithm converges under suitable conditions. We prove that the detection error associated with LVQ converges to the lowest possible error as the appropriate parameters go to infinity. We also discuss a modification to the algorithm which provides convergence for a larger set of initial conditions. Finally, we discuss how this method can be used with the various risks commonly found in classification.

In Chapter 4, we present several simulation results. Three types of classifiers are constructed and their classification performances compared against LVQ for two distinct sets of simulations. The first set involves the discrimination between two Gaussian distributed patterns and the second involves the discrimination between Rayleigh versus lognormal distributed patterns. Throughout the simulations, the LVQ algorithm is computed for several different values for its parameters.

In Chapter 5, we conclude with a discussion of implementation issues for LVQ and future directions for this work. In addition, we discuss how this method could be used in connection with other types of observation data.

Chapter 1

Nonparametric Detection

In this chapter we review classification theory, nonparametric density estimation and vector quantization. We present several definitions and results which will be used throughout this dissertation.

1.1 Statistical Pattern Recognition

The material presented in this section is covered in standard texts on statistical pattern recognition (e.g. (Fukunaga [1972])). It is reviewed here to set the notation and to show how the underlying statistical models strongly effect the optimal classifier.

In order to simplify the notation and better illustrate the notions behind statistical pattern recognition, we consider the case of two patterns. In this case, we are given two probability density functions $p_1(x)$ and $p_2(x)$ with observations from the first pattern distributed according to the density $p_1(x)$ and those from the second pattern distributed according to the density $p_2(x)$. If the prior probabilities of occurrence of the patterns are known, then a classifier can be designed using the Bayesian approach. Otherwise, the classifier can be designed using the Neyman–Pearson method.

A classifier takes an observation as input and determines which pattern was observed. Thus, the classifier can be represented by two disjoint sets $\{S_1, S_2\}$. The observations that fall in set S_1 are declared to be from pattern 1, those which fall in set S_2 are declared to be from pattern 2.

In general, the classifier can make two types of errors in performing its task. It can declare pattern 2 when in fact, pattern 1 was observed or it can declare pattern 1 when pattern 2 was observed. Classifiers typically make errors when the pattern probabilities overlap, i.e., when there is a positive probability of finding either pattern in a particular region. The goal of optimal classification is to minimize the errors of misclassification. In order to control these errors, different cost functions may be used. We will discuss three methods for designing classifiers: (1) Bayes' decision rule for minimum error, (2) Bayes' decision rule for minimum risk, and (3) the Neyman–Pearson test.

1.1.1 Bayes' Decision Rule for Minimum Error

As its name suggests, this rule is used when a classifier having the smallest possible probability of error is sought. To be precise, let π_1 (resp. π_2) denote the prior probability that pattern 1 (resp. 2) is observed. Given a classifier $S = \{S_1, S_2\}$, the probability of error is

$$r_1(S) = \int_{S_1} p_2(x)\,\pi_2\,dx + \int_{S_2} p_1(x)\,\pi_1\,dx \tag{1.1}$$

$$= \pi_1 + \int_{S_1} \left(p_2(x)\,\pi_2 - p_1(x)\,\pi_1\right)\,dx. \tag{1.2}$$

This cost is clearly minimized when all points for which the integrand is positive are declared to be members of the region S_2. The resulting optimal decision regions are thus defined by[1]

$$S_2 = \{x \,|\, p_2(x)\,\pi_2 - p_1(x)\,\pi_1 > 0\} \tag{1.3}$$

$$S_1 = \Re^d \setminus S_2. \tag{1.4}$$

were $\Re^d$ denotes d-dimensional Euclidean space.

1.1.2 Bayes' Decision Rule for Minimum Risk

Suppose that with each decision there is an associated cost $C(\delta, H)$, for deciding δ when pattern H is true. Let the cost be given by

$$C(\delta, H) = C_{ij} \quad \text{if} \quad d = i,\ H = j, \qquad i, j \in \{1, 2\}. \tag{1.5}$$

[1]Note that we can arbitrarily assigned points on the boundary to the region S_1.

Here, we assume that it costs less to make a correct decision than it does to make an incorrect one, i.e., we assume that $C_{ii} < C_{ij},\ j \neq i$. The Bayesian optimal minimum risk rule seeks to minimize the average cost or the expected risk

$$\begin{aligned} r_2(S) &= E(C(\delta, H)) && (1.6)\\ &= C_{11}\, P(\delta = 1, H = 1) + C_{21}\, P(\delta = 2, H = 1) && (1.7)\\ &\quad + C_{12}\, P(\delta = 1, H = 2) + C_{22}\, P(\delta = 2, H = 2). && (1.8) \end{aligned}$$

An application of Bayes rule yields

$$\begin{aligned} r_2(S) &= C_{11}\, P(\delta_1 \,|\, H_1)\, \pi_1 + C_{21}\, P(\delta_2 \,|\, H_1)\, \pi_1 && (1.9)\\ &\quad + C_{12}\, P(\delta_1 \,|\, H_2)\, \pi_2 + C_{22}\, P(\delta_2 \,|\, H_2)\, \pi_2. && (1.10) \end{aligned}$$

Suppose $\{S_1, S_2\}$ are given. Then

$$P(\delta_i \,|\, H_j) = \int_{S_i} p_j(x)\, dx. \tag{1.11}$$

Since $\Omega = S_2 \bigcup S_1$ and $S_2 \bigcap S_1 = \emptyset$, we have

$$\int_{S_2} p_i(x)\, dx = 1 - \int_{S_1} p_i(x)\, dx, \qquad i = 1, 2. \tag{1.12}$$

Therefore,

$$\begin{aligned} r_2(S) &= C_{21}\pi_1 + C_{22}\pi_2 && (1.13)\\ &\quad + \int_{S_1} \{\pi_2\, (C_{12} - C_{22})\, p_2(x) - \pi_1\, (C_{21} - C_{11})\, p_1(x)\}\, dx. && (1.14) \end{aligned}$$

Again, the decision regions are chosen so as to minimize the integral. This is accomplished by choosing $\{S_1, S_2\}$ as

$$\begin{aligned} S_2 &= \{x \,|\, p_2(x)\, \pi_2 - \gamma\, p_1(x)\, \pi_1 > 0\} && (1.15)\\ S_1 &= \Re^d \setminus S_2 && (1.16) \end{aligned}$$

where $\gamma := (C_{21} - C_{11})/(C_{12} - C_{22})$. Observe that it is without loss in generality to assume that $C_{ii} = 0$ in the search for S^*.

1.1.3 Neyman–Pearson Test

In the Neyman–Pearson test, the observations are assigned to regions which depend on the pattern probabilities explicitly. There are two types of errors made in deciding which pattern is true. The first error, ϵ_1, occurs in deciding pattern 2 when pattern 1 is true. The second error, ϵ_2, occurs in deciding pattern 1 when pattern 2 is true. If pattern 2 is interpreted as "target", and pattern 1 as "no target", then the first error is known as a false alarm and the second error is known as a miss. These errors can be explicitly calculated from

$$\epsilon_1 = \int_{S_2} p_1(x)\,dx \tag{1.17}$$

$$\epsilon_2 = \int_{S_1} p_2(x)\,dx. \tag{1.18}$$

The Neyman-Pearson test seeks to minimize ϵ_2 subject to ϵ_1 being equal to some constant, say β. This is a constrained optimization problem so the decision rule is found by minimizing

$$r_3(S) = \epsilon_2 + \mu(\epsilon_1 - \beta) \tag{1.19}$$

where μ is the Lagrange multiplier. Using (1.17)–(1.18) yields

$$r_3(S) = \int_{S_1} p_2(x)\,dx + \mu\Big(\int_{S_2} p_1(x)\,dx - \beta\Big) \tag{1.20}$$

$$= \mu(1-\beta) + \int_{S_1} (p_2(x) - \mu\, p_1(x))\,dx. \tag{1.21}$$

Proceeding as before, we see that the optimal decision regions are given by

$$S_2 = \{x \,|\, p_2(x) - \mu\, p_1(x) > 0\} \tag{1.22}$$

$$S_1 = \Re^d \setminus S_2. \tag{1.23}$$

We note that these three *different* decision strategies lead to similar definitions of the decision regions. Indeed, in all three cases

$$S_2 = \{x \,|\, p_2(x) - t\, p_1(x) > 0\} \tag{1.24}$$

for some appropriately chosen t. In the case of the minimum probability of error, t is chosen as π_1/π_2 when $\pi_2 \neq 0$; in the case of minimum Bayes risk, t is chosen

as $\gamma\pi_1/\pi_2$; in the case of Neyman–Pearson test, t is chosen so that the probability of false alarm equals β.

Throughout the remaining sections, the term Bayes' risk will refer to one of the costs above, either $r_1(S)$, $r_2(S)$, or $r_3(S)$. The precise cost will be specified when needed.

If the underlying densities are unknown then the previous methods for statistical pattern recognition are obviously not applicable. However, estimates of the pattern densities can be formed based on the past observations. Therefore, in the next section we discuss the effect on the Bayes' risk in using estimated densities as if they were the actual densities. It will be shown that if the estimated densities converge in the appropriate sense to the true densities, then the resulting estimated risk converges to the true optimal risk.

1.2 Bayes' Risk Consistent Density Estimators

In this section we discuss Bayes' risk consistency of density estimators. Consistency is the property of convergence of an estimated value to the true value as some parameter goes to infinity. We present several definitions of consistency which are used throughout this dissertation. We then present a fundamental theorem about Bayes risk consistency from (Glick [1972]). This theorem shows that if an appropriate density estimator is used in any of the classification schemes above then the resulting estimated optimal risk converges to the true optimal risk.

1.2.1 Definitions of Consistency

Let $x_1, \ldots, x_N$ be independent observations distributed according to $p(x)$. By $\hat{p}(x; N)$ we denote a density estimate of $p(x)$ which is based on the N observations. Let E_p denote the expectation with respect to the density p. The following definitions will be used throughout this dissertation.

The *mean square error* and *mean integrated squared error* of $\hat{p}(x; N)$ at x under the density p are respectively

$$E_p[\hat{p}(x; N) - p(x)]^2 \tag{1.25}$$

and

$$E_p \int_{-\infty}^{\infty} [\hat{p}(x;N) - p(x)]^2 \, dx = \int_{-\infty}^{\infty} E_p[\hat{p}(x;N) - p(x)]^2 \, dx. \tag{1.26}$$

A sequence of density estimates is *consistent in quadratic mean* if for every x

$$\lim_{N\to\infty} E_p[\hat{p}(x;N) - p(x)]^2 = 0. \tag{1.27}$$

Likewise, a sequence of density estimates is *integratedly consistent in quadratic mean* if for every x

$$\lim_{N\to\infty} \int_{-\infty}^{\infty} E_p[\hat{p}(x;N) - p(x)]^2 dx = 0. \tag{1.28}$$

A sequence of density estimates is *weakly consistent* if, for every x

$$\lim_{N\to\infty} \hat{p}(x;N) = p(x) \qquad \text{in probability.} \tag{1.29}$$

Finally, a sequence of density estimators is *strongly consistent* if

$$\lim_{N\to\infty} \hat{p}(x;N) = p(x) \qquad \text{P–a.s.} \tag{1.30}$$

Notice that in all of the above definitions the estimate $\hat{p}(x;N)$ may not itself be a density function, i.e., $\int_{-\infty}^{\infty} \hat{p}(x;N)\, dx$ need not equal 1 for any finite N. In fact, the integral may not exist at all. The lack of this property can result in density estimates which are not Bayes' risk consistent.

1.2.2 Convergence for Bayes Risk

We now consider the error associated in using the estimated densities as if they were the true densities. In this case, we consider the Bayes risk for minimum probability of error. The results hold equally well for all the risks discussed previously. In this problem we are given independent observations. For the case of two patterns, the four quantities to be estimated are π_1, π_2, $p_1(x)$, and $p_2(x)$. The observed data consists of the set $Z_N = \{z_j\}_{j=1}^N$ where z_j is the random vector $z_j = (x_j, d_{x_j})$, such that for $d_{x_j} = i$, x_j is an independent vector distributed according to $p_i(x)$, $i = 1, 2$.

The Bayes risk for minimum probability of error is given by

$$r_1(S) = \int_{S_2} p_1(x)\, \pi_1 \, dx + \int_{S_1} p_2(x)\, \pi_2 \, dx. \tag{1.31}$$

with $S^* = \{S_1^*, S_2^*\}$ given by

$$S_2^* = \{x \mid p_2(x)\,\pi_2 - p_1(x)\,\pi_1 > 0\} \tag{1.32}$$

$$S_1^* = \Re^d \setminus S_2^*. \tag{1.33}$$

Given a new observation x_{N+1} the goal is to infer the value of d_{N+1} based on the past observations Z. Let N_i denote the number of observations in Z for which $d_{x_j} = i$. Suppose that the past observations are used to estimate both the *a priori* probabilities and the conditional probabilities. The *a priori* probabilities can be estimated by $\hat{\pi}_i(N) = N_i/N$ and the conditional densities can be estimated by one of the methods to be discussed in Section 1.3. Let $\hat{p}_i(x;N)$, $i = 1, 2$, denote the estimated conditional probabilities. These estimates can be used to construct an estimate of the Bayes risk given by

$$\hat{r}_1(S;N) = \int_{S_2} \hat{p}_1(x;N)\hat{\pi}_1(N)\,dx + \int_{S_1} \hat{p}_2(x;N)\hat{\pi}_2(N)\,dx. \tag{1.34}$$

Here we assume that the integrals exist. As before, this integral is minimized by $\hat{S}^*(N)$ where

$$\hat{S}_2^*(N) = \{x \mid \hat{p}_2(x;N)\hat{\pi}_2(N) - \hat{p}_1(x;N)\hat{\pi}_1(N) > 0\} \tag{1.35}$$

$$\hat{S}_1^*(N) = \Re^d \setminus \hat{S}_2^*. \tag{1.36}$$

The main result from (Glick [1972, Theorem B]) is that if the estimates of the densities and the estimates of the priors converge and if all of the estimates are consistent then the associated estimated Bayes risk is also consistent. In other words, the risk associated with the estimated densities approaches the optimal risk. We have the following result.

Theorem 1.2.1 *(Glick [1972]) Let $\hat{p}_2(x;N)$ and $\hat{p}_1(x;N)$ be strongly consistent density estimates. If they are also densities for each N, then the sample-based risk $\hat{r}_1(S;N)$ converges a.s. to the true risk $r_1(S)$, uniformly over the domain of all classification rules. Suppose further that*

$$\int_\Omega \{\hat{p}_2(x;N)\hat{\pi}_2(N) + \hat{p}_1(x;N)\hat{\pi}_1(N)\}\,dx \to 1, \qquad a.s. \tag{1.37}$$

then

$$\sup_S |\hat{r}_1(S;N) - r_1(S)| \to 0, \qquad a.s. \tag{1.38}$$

Moreover, the theorem remains valid if all of the above convergences are replaced by convergence in probability.

Proof: For any classification rule S,

$$\begin{aligned} 0 &\leq |\hat{r}_1(S) - r_1(S)| && (1.39)\\ &\leq \int_{S_2} |\hat{p}_1(x)\hat{\pi}_1 - p_1(x)\,\pi_1|\,dx + \int_{S_1} |\hat{p}_2(x)\hat{\pi}_2 - p_2(x)\,\pi_2|\,dx && (1.40)\\ &\leq \int_{\Omega} |\hat{p}_1(x)\hat{\pi}_1 - p_1(x)\,\pi_1|\,dx + \int_{\Omega} |\hat{p}_2(x)\hat{\pi}_2 - p_2(x)\,\pi_2|\,dx && (1.41) \end{aligned}$$

Next, we show that the integrals

$$\int_{\Omega} |\hat{p}_i(x)\hat{\pi}_i - p_i(x)\,\pi_i|\,dx \to 0, \quad i = 1,2, \qquad \text{a.s.} \tag{1.42}$$

The assumptions

$$\hat{\pi}_i \to \pi_i \quad \text{and} \quad \hat{p}_i(x) \to p_i(x), \quad i = 1,2, \quad \text{a.s.} \tag{1.43}$$

imply that

$$\hat{p}_i(x)\,\hat{\pi}_i \to p_i(x)\,\pi_i \quad \text{a.s.} \tag{1.44}$$

Since

$$0 \leq \hat{p}_i(x)\,\hat{\pi}_i \leq \hat{p}_1(x)\,\hat{\pi}_1 + \hat{p}_2(x)\,\hat{\pi}_2 \tag{1.45}$$

and

$$\int_{\Omega} \{\hat{p}_2(x)\hat{\pi}_2 + \hat{p}_1(x)\hat{\pi}_1\}\,dx \to 1, \quad \text{a.s.} \tag{1.46}$$

the desired convergence follows from a variation of the Lebesgue bounded convergence theorem (Pratt [1960]). ■

This theorem shows that for large N, a detector can be built using the estimated densities instead of the actual densities. In addition, the estimated risk is close to the risk of the optimal detector. In the next section we show several techniques for generating consistent density estimators from past observations.

1.3 Nonparametric Density Estimation

We have shown that if a suitable approximation to the underlying densities is known, then a classifier that performs well compared to the optimal classifier can be

constructed. In Section 1.2 we showed that if these estimates converge to the true densities, then the corresponding risk converges to the true risk. In this section, three methods for density estimation are presented along with a discussion of their strengths and weaknesses. They are histogram estimation, nearest neighbor estimation, and kernel estimation. Throughout the discussion, we assume that the training data consist of N independent, identically distributed observations $x_1, \ldots, x_N$ from density $p(x)$. The book (Silverman [1986]) provides an excellent introduction to this material.

1.3.1 Histogram Method of Density Estimation

The histogram method is perhaps the oldest and most basic approach to density estimation. The simplest histogram estimator, referred to as a simple histogram, is characterized by an origin y_0 and a bin width h. Its regions are the intervals $[y_0 + mh, y_0 + (m+1)h)$ with $m = 1, \ldots, M$. The density estimate is given by

$$\hat{p}(x; N) = \frac{1}{N\,h} \ \{ \text{ Number of } x_i \text{ in same bin as } x \ \} \ . \tag{1.47}$$

This is a special case of a more general form of a histogram density estimator. In general, any density estimator which is constant on connected regions is a histogram density estimator. More complex histograms have bins which have variable shape. Simple histograms play an important role in the analysis of univariate data. However, they are of little value for multidimensional data since the number of bins increases exponentially as the dimension increases.

Simple histogram estimators are sensitive to the location of the origin y_0, i.e., shifting y_0 can result in very different looking histograms. This sensitivity to origin location has led to the development of other density estimation techniques. However, in the context of classification, histograms are still valuable.

One way to get around the problem of origin placement is to construct variable width histograms. In general, the histogram density estimate is given by

$$\hat{p}(x; N) = \frac{1}{N} \times \frac{\{ \text{ Number of } x_i \text{ in same bin as } x \ \}}{\{ \text{ Width of bin containing } x \ \}}. \tag{1.48}$$

In order to better account for the data, it is possible to let the bins depend on the observations. This results in random partition histograms which have bins that

are constructed directly from the data. Specifically, let $Y_1, \ldots, Y_N$ be the order statistics of $x_1, \ldots, x_N$, i.e. Y_1 is the smallest x_i, Y_2 is the next smallest, etc... Let $Y_0 = -\infty$ and $Y_{N+1} = \infty$. Suppose k_N is a sequence of positive integers satisfying

$$\lim_{N\to\infty} \frac{\log N}{k_N} = 0 \quad \text{and} \quad \lim_{N\to\infty} \frac{k_N}{N} = 0. \tag{1.49}$$

For example, $k_N = \lfloor \sqrt{N} \rfloor$ satisfies (1.49). Define

$$J_N = \{0, 1, k_N + 1, 2k_N + 1, 3k_N + 1, \ldots\} \tag{1.50}$$

and

$$A_N(x) = \max\{\alpha \,|\, \alpha \in J_N,\ Y_\alpha < x\} \tag{1.51}$$

$$B_N(x) = \min\{N, A_N(x) + k_N\}. \tag{1.52}$$

Let

$$\hat{p}(x; N) = \frac{K(A, B)}{N(Y_B - Y_A)} \tag{1.53}$$

be a density estimator where $[\, Y_A, Y_B \,)$ is the semi-open interval containing x and $K(A, B)$ represents the number of observations in that interval (usually $K(A, B)$ equals k_N). The symbols A and B are abbreviations for $A_N(x)$ and $B_N(x)$, respectively. This estimator has been studied in (Van Ryzin [1973]) where it was shown that under appropriate conditions, it is consistent. This result is presented next.

Lemma 1.3.1 *(Van Ryzin [1973, Corollary 2]): If the sequence $\{k_N\}$ satisfies (1.49) and if $x \in C(p)$, where $C(p)$ is the set of points where $p(x)$ is continuous, then $\hat{p}(x; N)$ is a strongly consistent estimator for $p(x)$.*

More general histograms are used when the observation dimension is large because of exponential growth problems associated with the simple histogram method. There are two conflicting goals. The first is to have enough bins to obtain some detailed information for the density estimate and the second is to have the required number of observations be low. Since it is generally believed that the number of estimated parameters should be much smaller than the number of observations (Duda & Hart [1973]), it is easy to see that a simple histogram would require a very large amount of data in order the achieve reasonable accuracy for observations in several dimensions.

In order to alleviate this problem, general histograms have regions which are adapted to the data. This allows the use of connected regions instead of simple bins. Adjusting the regions helps get better accuracy and keeps the number of regions down thus requiring a small number of observations.

1.3.2 Nearest Neighbor Density Estimation

The k-nearest neighbors method is a nonparametric detection scheme in its own right. We show that this approach can be used to form a nonparametric density estimator. This notion of density estimator is dual to the histogram method. Indeed, the idea is to find the smallest hypersphere centered on x which contains k points instead of finding the number of points in a fixed region.

To describe this density estimation scheme in detail, let $\rho(x, y)$ denote a metric measuring the distance between x and y. The k-nearest neighbors of x are the k closest points to x in the metric ρ. Let ρ_k represent the distance between x and the kth closest point and define $\mathcal{N}_k(x)$ to be the k-nearest neighborhood of x, i.e.,

$$\mathcal{N}_k(x) = \{\, y \,|\, \rho(x, y) \leq \rho_k \}. \tag{1.54}$$

The k-nearest neighbor density estimate of $p(x)$ which is usually credited to (Loftsgaarden & Quesenberry [1965]), but was first proposed in (Fix & Hodges [1951]) and is given by

$$\hat{p}(x; N) = \frac{k}{N \operatorname{Vol}(\mathcal{N}_k(x))}. \tag{1.55}$$

If we let k depend on the sample size N then a strongly consistent density estimate can be formed. Thus we have

Lemma 1.3.2 *(Rao [1983, Theorem 3.2.2]) Let $p(x)$ be continuous at x and let $\{k_N\}$ be a sequence of integers satisfying $\lim(k_N/N) = 0$, and $\lim(k_N/\log\log(N)) = \infty$. Then $\hat{p}(x; N)$ is strongly consistent.*

1.3.3 Kernel Density Estimation

Another method for density estimation is the kernel density estimator. The idea behind kernel density estimation is that each observation point x_i is replaced by

a function which depends on x_i. The density estimate is obtained by summing up the values of these functions. This technique is widely used for low dimensional problems since it has many desirable features. We now describe how a kernel estimator can be built up from a "naive estimator".

To begin with, consider that one method for estimating the density would be to form the so–called "naive estimator". This estimate is formed by first calculating the empirical distribution function $\hat{P}(x;N)$ and then estimating the density by the central difference operation

$$\hat{p}(x;N) = \frac{\hat{P}(x+h_N;N) - \hat{P}(x-h_N;N)}{2h_N} \tag{1.56}$$

where h_N tends to zero as N goes to infinity.

The estimate can be written in a more general form. Define the weight function

$$\bar{w}(x) = \begin{cases} \frac{1}{2} & \text{if } |x| < 1 \\ 0 & \text{otherwise.} \end{cases} \tag{1.57}$$

The naive estimator is given by

$$\hat{p}(x;N) = \frac{1}{N\,h_N} \sum_{i=1}^{N} \bar{w}\left(\frac{x-x_i}{h_N}\right). \tag{1.58}$$

The estimate is constructed by centering boxes of height $(2Nh_N)^{-1}$ and width $2h_N$ around each observation and summing them up. We note that $\hat{p}(x;N)$ is itself a density function since $\int \bar{w}(x)\,dx = 1$. The regularity of this estimate is controlled by h_N. If we consider the behavior of this estimate as h_N goes from zero to some small number, we first see that the estimate consists of delta functions located at the observation points which then expand to include neighbors of the observations. Eventually, the boxes of neighboring observations overlap completely and the density estimate loses all detail.

This adjustable characteristic exists in all of the density estimation schemes mentioned so far. In the k-nearest neighbors estimates, the adjustable parameter was k, the number of neighbors of x. In the simple histogram, it was the bin width. Finally, in this method, it is the parameter h_N.

The function $\bar{w}$ is just a specific example of a class of functions which can be used to create consistent density estimators. Let $w(\cdot)$ be a function satisfying the conditions below

a) $w(\cdot)$ is a density on $\Re^d$,

b) $\lim_{\|x\|\to\infty} \|x\|^d w(x) = 0$, and

c) $\sup_{x\in\Re^d} w(x) < \infty$.

The function $w(\cdot)$ is called a kernel function and

$$\hat{p}(x;N) = \frac{1}{N h_N^d} \sum_{i=1}^{N} w\left(\frac{x - x_i}{h_N}\right) \tag{1.59}$$

is the resulting kernel density estimate.

Under the appropriate conditions, $\hat{p}(x;N)$ is a strongly consistent density estimator (Cacoullos [1966]). Specifically, we have the following:

Lemma 1.3.3 *(Rao [1983, Theorem 3.1.5]) Suppose that $w(\cdot)$ satisfies conditions (a)-(c), $\lim h_N = 0$ and $\lim(N h_N^d) = \infty$. In addition suppose that for all $\alpha > 0$*

$$\sum_{N=1}^{\infty} \exp(-\alpha N h_N) < \infty. \tag{1.60}$$

Then $\hat{p}(x;N)$ is strongly consistent. Note that (1.60) is true if $\lim(N h_N / \log N) = \infty$.

1.3.4 Comparisons of Density Estimators

We have seen that the density plays a critical role in the construction of the optimal classifier, and that if a consistent density estimator were found which was itself a density, then the associated estimated Bayes risk converged to the optimal Bayes risk. Here we discuss the advantages and disadvantages of the various density estimation schemes and the feasibility of using these schemes to construct a nonparametric classifier.

Note that in the construction of the actual nonparametric classifier it is not necessary to explicitly calculate the density estimator. The amount of computation required could be prohibitive, even in the scalar case. The abstraction of the consistent density estimator was merely a device employed to conveniently prove the appropriate behavior of the resulting classification schemes.

Kernel density estimates are themselves densities and require the storage of all the observations and N evaluations of the kernel function for each estimate, therefore they can be computationally expensive.

Nearest neighbor density estimates are not themselves densities because the estimates of the tails do not decay fast enough, therefore they are not Bayes' risk consistent. This means that it is not possible for the nearest neighbor classifier to reach the Bayes' optimal risk. As with Kernel estimates, they require the storage of all of the observations.

Histogram density estimates do not require the storage of all of the observations. They only require the storage of the description of the bins. For simple histograms, the number of bins grows exponentially with the dimension of the observation space. In higher dimensions, connected regions should be used instead of uniform bins because of the high number of simple bins required and hence the high amount of observation data required for each bin.

From the implementation point of view, some of these estimates have the drawback that they require the storage of a large number of parameters. We seek a method for reducing the amount of data stored while controlling the associated error. This can be accomplished by a data reduction scheme such as Vector Quantization.

1.4 Vector Quantization

In this section we briefly discuss vector quantization and show how it can be used for density approximation. Vector quantization is commonly used for data compression. It consists of taking a continuous random vector and replacing it by a discrete approximation. The approximation will necessarily result in an error and the goal is to pick the approximation so that the expected error is minimized.

More specifically, let X be a d-dimensional random vector described by the probability density function $p(x)$. Let $D \subset \Re^d$ be such that $P(X \in D) = 1$. A k-level quantizer $Q = \{\Theta, V\}$ consists of: (i) a reproduction alphabet $\Theta = \{\theta_1, \ldots, \theta_k\}$; (ii) a partition $V = \{V_1, \ldots, V_k\}$ of D; and (iii) a mapping $\mathcal{Q} : D \rightarrow \Theta$ defined by $\mathcal{Q}(x) = \theta_i$ if $x \in V_i$.

An error or cost metric, $\rho(\theta, x)$, is incurred for reproducing x as θ. The cost $\rho(\theta, x)$ satisfies the following two conditions:

a) $\rho(\theta, x)$ is a twice continuously differentiable function of θ and x and for every fixed $x \in \Re^d$ it is a convex function of θ.

b) For any fixed x, if $\theta \to \infty$, then $\rho(\theta, x) \to \infty$.

The following are examples of cost functions satisfying these requirements which are commonly used in vector quantization:

(i) Let $\|\cdot\|$ be a norm and g be a nonconstant convex function on $[0, \infty)$ with $g(0) = 0$,

$$\rho(\theta, x) = g(\|x - \theta\|). \tag{1.61}$$

(ii) Let $R(x)$ be a positive definite matrix *depending on* x,

$$\rho(\theta, x) = (x - \theta)^T R(x)(x - \theta). \tag{1.62}$$

This cost function is known as the Itakura–Saito distortion measure.

Let $\rho(\theta, x)$ satisfy (a)–(b), then the average error associated with the quantizer $\{\Theta, V\}$ is given by

$$\begin{aligned} J(\Theta, V) &= E(\rho(Q(x), x)) && (1.63) \\ &= \sum_{i=1}^{k} \int_{V_i} \rho(\theta_i, x)\, p(x)\, dx. && (1.64) \end{aligned}$$

A quantizer $\{\Theta^*, V^*\}$ is said to be optimal for $J(\Theta, V)$, with respect to the density $p(x)$, if

$$J(\Theta^*, V^*) \leq J(\Theta, V) \tag{1.65}$$

for all other quantizers $\{\Theta, V\}$.

There are two standard results relating the reproduction alphabet Θ to the partition V. Let $V_{\theta_i} = \{x \in D \,|\, \rho(\theta_i, x) < \rho(\theta_j, x),\ j \neq i\}$ with equidistant points being assigned to the region with the lowest index. V_{θ_i} is called a *Voronoi cell* and the collection $\{V_{\theta_i}\}$ is called a Dirichlet partition of D (Gray [1984]).

Property 1 *Given a reproduction alphabet* $\Theta = \{\theta_1 \ldots, \theta_k\}$, *the partition* $V_\Theta = \{V_{\theta_1}, \ldots, V_{\theta_k}\}$ *has an error which is less than or equal to that of any other partition* V.

Property 2 *Let cent*(V_i) *be the generalized centroid of* V_i. *It is defined by*

$$\text{cent}(V_i) = \arg\min_{\theta \in D} \int_{V_i} \rho(\theta, x)\, p(x)\, dx. \tag{1.66}$$

Given a partition $V = \{V_1, \ldots V_k\}$ *the reproduction alphabet* $\hat{\Theta} = \{\text{cent}(V_i)\}_{i=1}^k$ *has an error which is less than or equal to that of any other reproduction alphabet* Θ.

The above properties can be used to construct an algorithm which finds a sequence of partitions which successively lower the error (1.64). The algorithm alternates between finding a partition, $V(n+1)$, which is optimal for the current reproduction alphabet, $\Theta(n)$, and then finding a reproduction alphabet, $\Theta(n+2)$, which is optimal for the current partition, $V(n+1)$. Here n is the iteration number for the algorithm. It has been shown (Linde, Buzo & Gray [1980]) that at each step the error is decreased and that in the limit as n goes to infinity the algorithm converges to a local optimum of $J(\Theta, V)$. This algorithm is known as the Linde-Buzo-Gray (LBG) algorithm.

In view of the Properties 1-2, the function $J(\Theta, V)$ can be considered a function of Θ only. Hence we can write $J(\Theta) = J(\Theta, V_\Theta)$ with $V_\Theta = \{V_{\theta_1}, \ldots, V_{\theta_k}\}$. In addition, we represent the optimal vector quantizer as Θ^* with the understanding that the corresponding optimal partition is given by V_{Θ^*}.

Unfortunately, the density is not usually available; instead one has independent samples $x_1, \ldots, x_N$ distributed according to $p(x)$ from which to estimate the cost in (1.64). This leads to considering an approximate average error given by

$$\tilde{J}(\Theta; N) = \frac{1}{N} \sum_{j=1}^{N} \sum_{i=1}^{k} \rho(\theta_i, x_j)\, 1_{\{x_j \in V_{\theta_i}\}} \tag{1.67}$$

where $1_{\{A\}}$ denotes the indicator function of the set A. A local minimum to (1.67) can also be found with the LBG algorithm by using sample averages instead of expectations.

It is possible to construct a density estimate from the Θ_N^* which minimizes $\tilde{J}(\Theta; N)$. This density estimate is a general histogram estimator with convex, random connected regions. Let Θ_N^* be fixed and suppose $x \in V_{\theta_i^*(N)}$ for some i. Then

$$\hat{p}(x;N) = \frac{1}{N \operatorname{Vol}(V_{\theta_i^*(N)})} \{ \text{ Number of } x_i \text{ in region } V_{\theta_i^*(N)} \} \tag{1.68}$$

$$= \frac{\sum_{j=1}^{N} 1_{\{x_j \in V_{\theta_i^*(N)}\}}}{N \operatorname{Vol}(V_{\theta_i^*(N)})} \tag{1.69}$$

is a density estimate for $p(x)$. In the sections below, we show that this estimate is weakly consistent.

1.4.1 Convergence of the Estimated Cost

Let $\Theta = \{\theta_1, \ldots, \theta_k\}$ and consider

$$J(\Theta) = \sum_{i=1}^{k} \int_{V_{\theta_i}} \rho(\theta_i, x)\, p(x)\, dx. \tag{1.70}$$

We want to find Θ^* which minimizes $J(\Theta)$. It can be shown that there exists Θ^*, a countable set with $k = \infty$, such that $J(\Theta^*) = 0$, the lowest possible cost.

We recall the strong law of large numbers (SLLN) and the weak law of large numbers (WLLN), respectively below.

Theorem 1.4.1 *(Billingsley [1979, p250]) Suppose $\{Y_j\}$ is a sequence of independent, zero mean random variables and suppose $\sum_j Var[Y_j]/j^2 < \infty$ then $\frac{1}{n}\sum_{j=1}^{n} Y_j \to 0$ almost surely.*

Theorem 1.4.2 *(Billingsley [1979, p252]) Suppose that for each n, $(\Omega_n, \mathcal{F}_n, P_n)$ is a probability space and that $Y_1(n), \ldots, Y_{r_n}(n)$ are independent random variables. Let $S(n) = \sum_{j=1}^{r_n} Y_j(n)$ and let $E[Y_j(n)] = m_j(n)$, $Var[Y_j(n)] = \sigma_j^2(n)$. Define*

$$E[S(n)] = m(n) = \sum_{j=1}^{r_n} m_j(n) \qquad Var[S_n] = \sigma^2(n) = \sum_{j=1}^{r_n} \sigma_j^2(n). \tag{1.71}$$

If for each n, $v(n) > 0$ is such that $\sigma(n)/v(n) \to 0$, then

$$P_n \left[\left| \frac{S(n) - m(n)}{v(n)} \right| \geq \epsilon \right] \to 0 \tag{1.72}$$

for all positive ϵ.

Next we show that for k fixed, $\tilde{J}(\Theta; N) \to J(\Theta)$ with probability one. Let

$$Y_j = \sum_{i=1}^{k} \rho(\theta_i, x_j)\, 1_{\{x_j \in V_{\theta_i}\}} - \sum_{i=1}^{k} \int_{V_{\theta_i}} \rho(\theta_i, x)\, p(x)\, dx. \tag{1.73}$$

Then $E[Y_j] = 0$ and

$$\begin{aligned}
\mathrm{Var}[Y_j] &= E[Y_j^2] && (1.74)\\
&\leq E[(\sum_{i=1}^{k} \rho(\theta_i, x_j)\, 1_{\{x_j \in V_{\theta_i}\}})^2] && (1.75)\\
&= E[\sum_{i=1}^{k} \sum_{ii=1}^{k} \rho(\theta_i, x_j)\, \rho(\theta_{ii}, x_j)\, 1_{\{x_j \in V_{\theta_i}\}} 1_{\{x_j \in V_{\theta_{ii}}\}}] && (1.76)\\
&= E[\sum_{i=1}^{k} \rho^2(\theta_i, x_j)\, 1_{\{x_j \in V_{\theta_i}\}}] && (1.77)\\
&= \sum_{i=1}^{k} \int_{V_{\theta_i}} \rho^2(\theta_i, x)\, p(x)\, dx < \infty. && (1.78)
\end{aligned}$$

Hence $\tilde{J}(\Theta; N) \to J(\Theta)$ follows from (SLLN).

Now, we are interested in exploring the value of the optimal cost when k is also allowed to go to infinity. First we consider a simple case. Let $k_N = N$ then if $\theta_i = x_i$, we see that $\tilde{J}(\Theta; N) = 0$ for all N and the optimal cost is reached.

Next, we consider another choice for k_N and $\Theta(N)$ which results in an asymptotically optimal cost, i.e., $\tilde{J}(\Theta_N; N) \to 0$ in probability as $N \to \infty$. To this end, we assume that $p(x)$ is continuous, with compact support D.

Let k_N satisfy (1) $\lim_N (k_N/N) = 0$ and (2) $\lim_N k_N = \infty$. Suppose that Θ_N is chosen so that the Voronoi cells form a "roughly" uniform partition of the domain D, i.e.,

$$\mathrm{Vol}(V_{\theta_{i(N)}}) = O(\frac{1}{k_N}) \qquad \text{with} \quad D = \bigcup_{i=1}^{k_N} V_{\theta_{i(N)}}. \tag{1.79}$$

Letting $Y_j(N)$ be defined by (1.73) , we have

$$\begin{aligned}
\mathrm{Var}[Y_j(N)] &= \sum_{i=1}^{k_N} \int_{V_{\theta_{i(N)}}} \rho^2(\theta_i(N), x)\, p(x)\, dx && (1.80)\\
&= \sum_{i=1}^{k_N} \mu(V_{\theta_{i(N)}})\, \rho^2(\theta_i(N), c_i)\, p(c_i) && (1.81)\\
&\qquad \text{for } c_i \in V_{\theta_{i(N)}} \text{ by MVT}
\end{aligned}$$

$$\leq k_N \frac{L}{k_N} \max_{i=1,\ldots,k_N} \rho^2(\theta_i(N), c_i)\, p(c_i) \tag{1.82}$$

$$\leq L \max_{i=1,\ldots,k_N} \rho^2(\theta_i(N), c_i)\, p(c_i) \tag{1.83}$$

$$\leq \bar{L}. \tag{1.84}$$

where $\bar{L}$ is some constant. This follows since $p(x)$ and $\rho(\theta, x)$ are continuous and D is compact. Therefore, $\sum Y_j(N) \leq N\bar{L}$. Let $v(N) = N$ and apply (WLLN) to get

$$\tilde{J}(\Theta_N; N) \to 0 \qquad \text{in probability as } N \to \infty \tag{1.85}$$

1.4.2 Relation to the Global Optimal Quantizer

For each N, let Θ_N^* be a global minimum of $\tilde{J}(\Theta; N)$. From (1.85) and the property of the global minimum, we know that

$$0 \leq \tilde{J}(\Theta_N^*; N) \leq \tilde{J}(\Theta_N; N) \to 0 \tag{1.86}$$

in probability, therefore $\tilde{J}(\Theta_N^*; N) \to 0$. It follows that

$$\text{Vol}(V_{\theta_i^*(N)}) \to 0. \tag{1.87}$$

Suppose not. Let $\theta_i^*(N)$ be such that $\text{Vol}(V_{\bar{\theta}(N)}) \to C > 0$ then

$$E[\tilde{J}(\Theta_N^*; N) \geq E[\frac{1}{N} \sum_{j=1}^{N} 1_{\{x_j \in V_{\theta_i^*(N)}\}}] \tag{1.88}$$

$$= \int_{V_{\theta_i^*(N)}} \rho(\theta_i^*(N), x)\, p(x)\, dx \not\to 0. \tag{1.89}$$

We have the following.

Theorem 1.4.3 *Let Θ_N^* be an optimal vector quantizer for the problem (1.67). Let $\hat{p}(x; N)$ be the generalized histogram estimator constructed from V_{Θ^*} defined in (1.69). Then for each $x \in V_{\theta_i^*(N)}$, $\hat{p}(x; N)$ is a weakly consistent density estimator of $p(x)$.*

Proof: Apply (WLLN) with $Y_j = 1_{\{x_j \in V_{\theta_i^*}\}}/\text{Vol}(V_{\theta_i^*})$ and $v(N) = N$. ∎

1.5 Remarks

In this chapter we discussed the classification problem, nonparametric detection and vector quantization. We demonstrated that a consistent nonparametric detector can be built from a consistent nonparametric density estimator. We presented vector quantization and showed that it can be used to construct a consistent density estimate. In the next chapter, we review several results from stochastic approximation.

Chapter 2

Review of Stochastic Approximation

In this chapter we present a review of some stochastic approximation techniques together with results on their convergence. Stochastic approximation has a long history beginning with the work of (Robbins & Monro [1951]). In the sections below, we closely follow the presentation in (Benveniste, Metivier & Priouret [1987]). This presentation is particularly clear. The results on convergence of stochastic approximation will be used in subsequent chapters to show convergence of the LVQ algorithm.

2.1 The Heuristic Idea behind the ODE Method

Stochastic approximation consists of an iterative scheme for determining the critical points of a function by using random observations of that function. It is common to many recursive adaptive estimation schemes. The convergence of the parameters can be obtained by examining the stable equilibria of an ODE which is related to the update equation. In this section we give an informal presentation of stochastic approximation and indicate the method of proof for the theorems to follow.

The equation

$$\Theta_{n+1} = \Theta_n + \alpha_{n+1} H(\Theta_n, X_{n+1}) \tag{2.1}$$

is a stochastic approximation algorithm. The term stochastic refers to the fact

that for each n, X_{n+1} is an instance of a random variable. We assume that there exists a family, (μ_Θ), of probability distributions where $\mu_\Theta(dx)$ is the conditional probability density of X_{n+1} given Θ. That is, we assume that conditioned on Θ_n, X_{n+1} is independent of $\{X_k,\ k \leq n\}$. Let $\{\alpha_n\}_{n\geq 0}$ be a sequence of nonincreasing positive numbers and let

$$h(\Theta) := \int H(\Theta, x)\mu_\Theta(dx). \tag{2.2}$$

The study of the convergence of (2.1) is accomplished through relating Θ_n to the solution $\bar{\Theta}(t)$ of the equation

$$\frac{d\bar{\Theta}(t)}{dt} = h(\bar{\Theta}(t)). \tag{2.3}$$

This equation is called the ordinary differential equation associated to (2.1). Let $\bar{\Theta}_a(t)$ denote the solution of (2.3) (if it exists) with initial condition $\bar{\Theta}_a(0) = a$.

The algorithm (2.1) can be viewed as a random perturbation of (2.3). To this end, set

$$\xi(\Theta, x) := H(\Theta, x) - h(\Theta). \tag{2.4}$$

Let $\mathcal{F}_n$ denote the sigma algebra generated by $\{\Theta_0, \ldots, \Theta_n, X_0, \ldots, X_n\}$. The process defined by

$$M_n := \sum_{k\leq n} \alpha_k \xi(\Theta_{k-1}, X_k) \tag{2.5}$$

$$M_0 := 0 \tag{2.6}$$

is a martingale. This follows from the fact that

$$E[M_n - M_{n-1}|\mathcal{F}_{n-1}] = \alpha_n E[\xi(\Theta_{n-1}, X_n)|\mathcal{F}_{n-1}] = 0. \tag{2.7}$$

Equation (2.1) can therefore be written as

$$\Theta_{n+1} = \Theta_n + \alpha_{n+1}h(\Theta_n) + \Delta_n M \tag{2.8}$$

where

$$\Delta_n M := M_n - M_{n-1}. \tag{2.9}$$

If we introduce the following definitions:

$$t_n := \sum_{i=1}^{n} \alpha_i, \quad t_0 := 0 \tag{2.10}$$

$$\Theta(t) = \Theta_n \qquad \text{if } t_n \le t < t_{n+1} \tag{2.11}$$

$$m(t) := \max\{n : t_n \le t\} \tag{2.12}$$

$$a(t) := \sum_{k=1}^{m(t)} \alpha_k = t_{m(t)}. \tag{2.13}$$

Then, (2.8) becomes

$$\Theta(t) = \Theta_0 + \sum_{0<k\le m(t)} \int_{t_{k-1}}^{t_k} h(\Theta(s))ds + \sum_{0<k\le m(t)} \Delta_k M, \tag{2.14}$$

$$= \Theta_0 + \int_0^t h(\Theta(s))ds + R(t) + M(t) \tag{2.15}$$

where

$$R(t) := -\int_{a(t)}^{t} h(\Theta(s))ds \tag{2.16}$$

and

$$M(t) := \sum_{0<k\le m(t)} \Delta_k M. \tag{2.17}$$

Hence we see that (2.1) can be viewed as a random perturbation of (2.3) with $R + M$ being the perturbation.

The study of the convergence of equation (2.1) consists in comparing the behavior of Θ_n to the behavior of $\bar{\Theta}(t)$ when both start from the same initial condition. Two convergence results will be presented in this chapter.

The first result is that if $\bar{\Theta}^*$ is a locally stable point of (2.3) then, with high probability, Θ_n will come to visit a neighborhood of $\bar{\Theta}^*$ and will stay there an interval of time which is related to the size of α. More precisely, assume that $\Theta_0 = a$ and $\bar{\Theta}_a(0) = a$. Then for every finite T and $\eta > 0$

$$\lim_{\alpha_1 \downarrow 0} P\left(\sup_{t_n<T} |\Theta_n - \bar{\Theta}_a(t_n)| > \eta\right) = 0. \tag{2.18}$$

This result is proved in Section 2.3.

The second result involves the convergence of Θ_n to an asymptotically stable equilibrium of (2.3). If $\bar{\Theta}^*$ is an asymptotically stable equilibrium of (2.3), with

domain of attraction $D(\bar{\Theta}^*)$, and if Θ_n visits a compact subset of $D(\bar{\Theta}^*)$ infinitely often, then Θ_n converges to $\bar{\Theta}^*$ with probability one. This result is proved in Section 2.4, and is referred to as a Ljung–type result.

In the next section we give a detailed description of the stochastic approximation algorithm considered in this chapter.

2.2 Detailed Description of the Algorithm

Let $\{\Theta_n, X_n\}_{n\geq 0}$ be a sequence of random variables defined on a probability space (Ω, A, P), with values in $D \subset \Re^d$ and $S \subset \Re^k$, respectively. It is assumed that the conditional probability of X_{n+1} given $\mathcal{F}_n = \sigma(X_0, \ldots, X_n, \Theta_0, \ldots, \Theta_n)$ is expressed by $\Pi_{\Theta_n}(X_n; dx_{n+1})$ where for each $\Theta \in D$, $\Pi_\Theta(x, dx')$ is a transition probability matrix from S into S.

The general stochastic approximation model to be considered can be written as

$$\Theta_{n+1} = \Theta_n + \alpha_{n+1} H(\Theta_n, X_{n+1}) + \alpha_{n+1}^2 \varrho_{n+1}(\Theta_n, X_{n+1}) \tag{2.19}$$

where $H(\Theta, x)$ is a given "adaptation function" mapping $D \times S$ into D and ϱ_n is a given function mapping $D \times S$ into D.

The following hypotheses are assumed:

[H.1] $\{\alpha_k\}$ is a nonincreasing sequence of positive reals such that $\sum_n \alpha_n = \infty$.

[H.2] There exists a family $\{\Pi_\Theta : \Theta \in \Re^d\}$ of transition probabilities from $\Re^k \times \Re^d$ into $\Re^k$ such that, for every Borel subset A of $\Re^d$

$$P(X_{n+1} \in A | \mathcal{F}_n) = \Pi_{\Theta_n}(A, X_n). \tag{2.20}$$

Observe that this implies

$$E[g(\Theta_n, X_{n+1}) | \mathcal{F}_n] = \int g(\Theta_n, x) \Pi_{\Theta_n}(dx, X_n) \tag{2.21}$$

for every Borel measurable, positive $g(\Theta, x)$ such that $E|g(\Theta_n, X_{n+1})| < \infty$. The equation (2.21) implies that the random variable $\int g(\Theta_n(\omega), x) \Pi_{\Theta(n)}(dx, X_n(\omega))$ is a version of the conditional expectation of $g(\Theta_n, X_{n+1})$ given $\mathcal{F}_n$, that is to say, given the values taken by the variables Θ_k and X_k for $k \leq n$.

From hypothesis [H.2] we see that $\{\Theta_n, X_n\}_{n\geq 0}$ is a Markov process. Its transition probability depends on n through α_n and ϱ_n. If $\alpha_n = \alpha$ constant and $\varrho_n = \varrho$ then it is independent of n.

The following notation will be used throughout this chapter.

a) $P_{x,a}$ denotes the probability distribution of $\{\Theta_n, X_n\}_{n\geq 0}$ for the initial conditions $X_0 = x, \Theta_0 = a$, and $E_{x,a}$ denotes the expectation with respect to $P_{x,a}$.

b) Let $\Theta(t)$ be defined by

$$\Theta(t) = \sum_{k\geq 0} 1_{\{t_k \leq t < t_{k+1}\}}\, \Theta_k \tag{2.22}$$

where $1_{\{A\}}$ denotes the indicator function of the set A. We call $\Theta(t)$ the continuous process associated with the sequence $\{\Theta_n\}$.

The study of the behavior of $\Theta(t)$ between t_n and $t_n + T$ reduces to the study of Θ_k for $n \leq k \leq m(n,T)$ where

$$m(n,T) := \inf\{k : k \geq n, \alpha_{n+1} + \ldots + \alpha_{k+1} \geq T\}. \tag{2.23}$$

In the case where $t_n = 0$, the notation is simplified to

$$m(T) := m(0,T). \tag{2.24}$$

c) For every function $f(\Theta, x)$ on $\Re^d \times \Re^k$, f_Θ denotes the application $x \to f(\Theta, x)$. In particular, let $\Pi_\Theta f_\Theta$ denote the function

$$x \to \int f(\Theta, y)\, \Pi_\Theta(dy, x). \tag{2.25}$$

d) For every compact $Q \subset D$ and every $\epsilon > 0$ we set

$$\tau(Q) = \inf(n; \Theta_n \notin Q) \tag{2.26}$$

$$\sigma(\epsilon) = \inf(n : |\Theta_n - \Theta_{n-1}| > \epsilon) \tag{2.27}$$

$$\nu(\epsilon, Q) = \min(\tau(Q), \sigma(\epsilon)) \tag{2.28}$$

In what follows, D is an open subset of $\Re^d$. The functions H and ϱ_n are assumed to satisfy the following additional hypotheses:

[H.3] For every compact $Q \subset D$, there exist constants C_1, C_2, q_1, q_2 (depending on Q) such that for every $\Theta \in Q$ and all n

(i) $|H(\Theta, x)| \leq C_1(1 + |x|^{q_1})$

(ii) $|\varrho_n(\Theta, x)| \leq C_2(1 + |x|^{q_2})$.

[H.4] There exist a function h on D, and for each $\Theta \in D$, a function $v_\Theta(\cdot)$ on $\Re^k$ such that

(i) h is locally Lipschitz on D

(ii) $(I - \Pi_\Theta)\, v_\Theta = H_\Theta - h(\Theta)$ for every $\Theta \in D$. In the vector case, this means that for every coordinate $i = 1, \ldots, d$,

$$(I - \Pi_\Theta)\, v_{i\Theta} = H_{i\Theta} - h_i(\Theta) \tag{2.29}$$

(iii) For every compact $Q \subset D$, there exist constants C_3, C_4, q_3, q_4, $\kappa \in [1/2, 1]$ such that for every Θ, $\hat{\Theta} \in Q$

$$|v_\Theta(x)| \leq C_3(1 + |x|^{q_3}) \tag{2.30}$$

$$|\Pi_\Theta\, v_\Theta(x) - \Pi_{\hat{\Theta}}\, v_{\hat{\Theta}}(x)| \leq C_4|\Theta - \hat{\Theta}|^\kappa\, (1 + |x|^{q_4}). \tag{2.31}$$

The ODE associated with (2.19) is (2.3) with the function $h(\cdot)$ defined in [H.4]. For example, if we assume that for each fixed Θ, the transition matrix Π_Θ is positive recurrent (see for example (Revuz [1975])) with invariant probability Γ_Θ and if

$$h(\Theta) = \int H_\Theta(y)\, \Gamma_\Theta(dy), \tag{2.32}$$

then the function $H_\Theta(\cdot) - h(\Theta)$ is zero mean with respect to Γ_Θ and the solution v_Θ of the equation (H.4.i.i), called *Poisson's equation*, is expressed as

$$v_\Theta(x) = \sum_{k \geq 0} \Pi_\Theta^k (H_\Theta - h(\Theta))\,(x) \tag{2.33}$$

provided the series is convergent. In applications, Γ_Θ is usually expressed in the form

$$\int g(x)\, \Gamma_\Theta(dx) = \lim_{n \to \infty} \Pi_\Theta^n\, g(x) \tag{2.34}$$

for a set of functions g which is dense in the space of continuous functions.

One of the following two hypotheses on moments of $P_{x,a}$ can be verified in most applications.

[H.5] For every compact $Q \subset D$ and all $q > 0$, there exist a finite constant $M(q,Q) < \infty$ such that for every $n \in \Re^k, a \in D$

$$E_{x,a}[1_{\{\Theta_k \in Q, k \leq n\}} (1 + |X_{n+1}|^q)] \leq M(q,Q)(1 + |x|^q) \tag{2.35}$$

Condition [H.5] is however, too strong to be true for a general linear dynamical system. Instead the following hypothesis will hold in that case.

[H'.5] For every compact Q in D and $q > 1$ there exist positive constants ε_0 and M such that for all $\varepsilon \leq \varepsilon_0$, $a \in Q$ and for all x

$$\sup_n E_{x,a}[|X_n|^q 1_{\{n \leq \nu(\varepsilon,Q)\}}] \leq M(1 + |x|^q). \tag{2.36}$$

For example, [H'.5] is satisfied when $\{X_n\}$ is a sequence of independent observations distributed according to a probability density function $p(x)$ which is continuous.

2.3 Convergence in Probability of the Paths

In this section we prove convergence in probability of the paths of Θ_n to $\bar{\Theta}(t)$. We have

Theorem 2.3.1 *Assume that [H.1]-[H'.5] hold and that $\alpha_1 \leq 1$. Let Q be a compact subset of D, $T > 0, a \in Q$, such that $\bar{\Theta}_a(t) \in Q$ for all $t \in [0,T]$. Then for every $\delta > 0$ and all x*

$$\lim_{\alpha \downarrow 0} P_{x,a} \left(\sup_{n \leq m(T)} |\Theta_n - \bar{\Theta}_a(t_n)| > \delta \right) = 0. \tag{2.37}$$

Furthermore, let $Q_2 \supset Q_1$ be two compact subsets of D. Let $T > 0$, such that for all $a \in Q_1$, all $t \leq T$,

$$d(\bar{\Theta}_a(t), Q_2^c) \geq \delta_0 > 0. \tag{2.38}$$

Then there exist constants B_1, L_2, s_1, such that for all $\delta < \delta_0$, $a \in Q_1$, $q > q_0(\lambda)$ and all x

$$P_{x,a}\left(\sup_{n\le m(T)} |\Theta_n - \bar{\Theta}_a(t_n)| \ge \delta\right) \tag{2.39}$$

$$\le \frac{B_1}{\delta^q}(1+|x|^{s_1})(1+T)^{q-1}\exp(qL_2T)\sum_{k=1}^{m(T)} \alpha_k^{1+q/2} \tag{2.40}$$

Proof:

Let

$$\Theta_{n+1} = \Theta_n + \alpha_{n+1}\, h(\Theta_n) + \varepsilon_n \tag{2.41}$$

$$= \Theta_n + \alpha_{n+1}\, H(\Theta_n, X_{n+1}) + \alpha_{n+1}\, \varrho_{n+1}(\Theta_n, X_{n+1}) \tag{2.42}$$

hence

$$\varepsilon_n = \alpha_{n+1}[H(\Theta_n, X_{n+1}) - h(\Theta_n) + \alpha_{n+1}\, \varrho_{n+1}\,(\Theta_n, X_{n+1})]. \tag{2.43}$$

A basic ingredient in the proof of Theorem 2.3.1 is the inequality in Proposition 1 below. For a function Φ mapping $\Re^d$ into $\Re$, with bounded continuous second order derivatives, set

$$\varepsilon_n(\Phi) = \Phi(\Theta_{n+1}) - \Phi(\Theta_n) - \alpha_{n+1}\, \Phi'(\Theta_n) h(\Theta_n) \tag{2.44}$$

Proposition 1 *Under the hypotheses of Theorem 2.3.1, there exist constants B and s such that for all $\varepsilon \le \varepsilon_0$, $T > 0$, $X_0 = x$, $a \in Q$*

$$E_{x,a}\left[\sup_{n<k\le m(n,T)} 1_{\{k\le \nu(\varepsilon,Q)\}} |\sum_{i=n}^{k-1} \varepsilon_i(\Phi)|^q\right] \tag{2.45}$$

$$\le B(1+T)^{q-1}(1+|x|^s)\sum_{i=n+1}^{m(n,T)} \alpha_i^{1+q/2} \tag{2.46}$$

Proof: (see (Benveniste, Metivier & Priouret [1987]))

Proof of Theorem 2.3.1 (continued):

Let us now consider Q_1, Q_2, T and δ_0 as in (2.38). This condition implies that for every $\delta < \delta_0$ the "tube of diameter δ" around the solution $\bar{\Theta}_a(t)$, for $t \le T$ is

included in Q_2. As a consequence of [H.4] there exist constants $L_1 = L_1(Q_2)$, $L_2 = L_2(Q_2)$ such that

$$|h(\Theta)| \le L_1 |h(\Theta) - h(\Theta')| \le L_2 |\Theta - \Theta'|, \qquad \text{for all } \Theta, \Theta' \in Q_2. \tag{2.47}$$

Then for all t_n with $t_{n+1} < T$

$$\bar{\Theta}_a(t_{n+1}) - \bar{\Theta}_a(t_n) = \int_{t_n}^{t_{n+1}} h(\bar{\Theta}_a(s))\, ds = \alpha_{n+1}\, h(\bar{\Theta}_a(t_n)) + \gamma_n \tag{2.48}$$

with

$$|\gamma_n| \le \alpha_{n+1}^2 L_2. \tag{2.49}$$

Applying (2.46) to the coordinate functions $\Phi_i(\Theta) = \Theta_i$, for some constants B and s, we have

$$E_{x,a}\left[\sup_{n \le m(T)} 1_{\{n \le \nu(\varepsilon, Q)\}} \left|\sum_{k=0}^{n-1} \varepsilon_k\right|\right]^q \tag{2.50}$$

$$\le B(1 + |x|^s)\,(1+T)^{q-1} \sum_{k=1}^{m(T)} \alpha_k^{1+q/2}. \tag{2.51}$$

Then

$$\Theta_r - \bar{\Theta}_a(t_r) = \Theta_{r-1} - \bar{\Theta}_a(t_{r-1}) + \alpha_r(h(\Theta_{r-1}) - h(\bar{\Theta}_a(t_{r-1})) + \varepsilon_{r-1} + \gamma_{r-1}. \tag{2.52}$$

Therefore

$$\Theta_r - \bar{\Theta}_a(t_r) = \sum_{k=0}^{r-1} \alpha_{k+1}(h(\Theta_k) - h(\bar{\Theta}_a(t_k)) + \sum_{k=0}^{r-1} \varepsilon_k + \sum_{k=0}^{r-1} \gamma_k, \tag{2.53}$$

$$|\Theta_r - \bar{\Theta}_a(t_r)| \le L_2 \sum_{k=0}^{r-1} \alpha_{k+1} |\Theta_k - \bar{\Theta}_a(t_k)| + \left|\sum_{k=0}^{r-1} \varepsilon_k\right| + L_2 \sum_{k=0}^{r-1} \alpha_{k+1}^2. \tag{2.54}$$

For all ω in the set $\{\omega : n \le \nu(\omega) \wedge m(T)\}$ and for $r = 0, 1, \ldots, n$

$$|\Theta_r - \bar{\Theta}_a(t_r)| \le L_2 \sum_{k=0}^{r-1} \alpha_{k+1} |\Theta_k - \bar{\Theta}_a(t_k)| \tag{2.55}$$

$$+ \sup_{m \le m(T)} \left\{ 1_{\{m \le \nu\}} \left| \sum_{k=0}^{m-1} \varepsilon_k \right| \right\} + L_2 \sum_{k=1}^{m(T)} \alpha_k^2 \tag{2.56}$$

$$= L_2 \sum_{k=0}^{r-1} \alpha_{k+1} |\Theta_k - \bar{\Theta}_a(t_k)| + U_1 + U_2. \tag{2.57}$$

An elementary computation shows that if

$$v_r \leq r_1 \sum_{i=1}^{r} \alpha_i \, v_{i-1} + r_2 \tag{2.58}$$

for $r_1, r_2, \alpha_i \geq 0$ and $r = 0, \ldots, n$, then

$$v_n \leq r_2 \exp(r_1 \sum_{i=1}^{n} \alpha_i). \tag{2.59}$$

Using this for ω in the set $\{\omega : n \leq \nu(\omega)\}$ implies

$$\sup_{n \leq \nu} |\Theta_n - \bar{\Theta}_a(t_n)|^q \leq 2^q \exp(q\, L_2\, T)(U_1^q + U_2^q). \tag{2.60}$$

According to Proposition 1

$$E[U_1^q] \leq B(1+T)^{q-1}\,(1+|x|^s) \sum_{k=1}^{m(T)} \alpha_k^{1+q/2}. \tag{2.61}$$

Hölder's inequality yields that for any $a_i \geq 0, b_i \in \Re, u > 1, 0 < \delta < 1$

$$|\sum_{i=n}^{m} a_i b_i|^u \leq \left(\sum_{i=n}^{m} a_i^{\delta u/(u-1)} \right)^{u-1} \sum_{i=n}^{m} a_i^{(1-\delta)u} |b_i|^u. \tag{2.62}$$

Applying this we obtain

$$U_2^q \leq L_2^q \left\{ \sum_{k=1}^{m(T)} \alpha_k^2 \right\}^q \leq L_2^q\, T^{q-1} \sum_{k=1}^{m(T)} \alpha_k^{1+q}, \tag{2.63}$$

and therefore

$$E_{x,a}\left[\sup_{n \leq \nu \wedge m(T)} |\Theta_n - \bar{\Theta}_a(t_n)|^q \right] \tag{2.64}$$

$$\leq A_2(1+|x|^{s_2})\,(1+T)^{q-1} \exp(q\, L_2\, T) \sum_{k=1}^{m(T)} \alpha_k^{1+q/2}. \tag{2.65}$$

Now for $\delta < \delta_0$ set

$$\Omega(\delta) = \left\{ \sup_{n \leq m(T)} |\Theta_n - \bar{\Theta}_a(t_n)| \geq \delta \right\} \tag{2.66}$$

and write

$$\Omega(\delta) \subset \left\{ \sup_{n \leq m(T)} |\Theta_n - \bar{\Theta}_a(t_n)| \geq \delta; m(T) \leq \nu \right\} \cup \{\nu < m(T)\}. \tag{2.67}$$

At time $n = \tau(Q_2)$, $|\Theta_n - \bar{\Theta}_a(t_n)| \geq \delta$, hence we have

$$\{\tau \leq \sigma, \tau < m(T)\} \subset \left\{ \sup_{n \leq \nu \wedge m(T)} |\Theta_n - \bar{\Theta}_a(t_n)| \geq \delta \right\} \tag{2.68}$$

and therefore

$$\Omega(\delta) \subset \left\{ \sup_{n \leq \nu \wedge m(T)} |\Theta_n - \bar{\Theta}_a(t_n)| \geq \delta \right\} \cup \{\sigma \leq \tau, \sigma < m(T)\}. \tag{2.69}$$

The theorem follows from (2.65) and Lemma 2.3.1 below. ∎

Lemma 2.3.1 *For every compact $Q \subset D$ and $q \geq 2$, there exist constants M and s_1 such that, for every $T > 0$, all $\varepsilon < \varepsilon_0$, all $a \in Q$ and all x*

$$P_{x,a}\{\sigma(\varepsilon) \leq \tau(Q), \sigma(\varepsilon) \leq m(T)\} \leq M(1 + |x|^{s_1}) \sum_{k=1}^{m(T)} \alpha_k^q. \tag{2.70}$$

Proof:

We have

$$\begin{aligned}
P(\sigma \leq \tau, \sigma \leq m) &= \sum_{k=1}^{m} P(\{\sigma = k\} \leq \tau) && (2.71)\\
&= \sum_{k=1}^{m} P(\{\sigma = k\} \leq \tau, |\Theta_k - \Theta_{k-1}| > \varepsilon) && (2.72)\\
&\leq \sum_{k=1}^{m} P(k \leq \{\sigma \wedge \tau\},\ C_1\, \alpha_k(1 + |X_k|^{q_1}) \\
&\qquad + C_2\, \alpha_k^2(1 + |X_k|^{q_2}) > \varepsilon) && (2.73)
\end{aligned}$$

Using H.3 we obtain

$$\begin{aligned}
&\leq \sum_{k=1}^{m} P(k \leq \sigma \wedge \tau, C'\, \alpha_k(1 + |X_k|^{s}) > \varepsilon) && (2.74)\\
&\leq \sum_{k=1}^{m} \left(\frac{C'}{\varepsilon}\right)^q \alpha_k^q\, E[(1 + |X_k|^s)^q\, 1_{\{k \leq \sigma \wedge \tau\}}] && (2.75)\\
&\leq M(1 + |x|^{sq}) \sum_{k=1}^{m} \alpha_k^q && (2.76)
\end{aligned}$$

where the last inequality follows from H.5. ∎

2.4 Ljung–Type Convergence

In this section we are interested in the asymptotic behavior of $\{\Theta_n\}$ when $n \to \infty$. For each N, consider the "tail-algorithm"

$$\Theta_{n+1}^N = \Theta_n^N + \alpha_{N+n+1} H(\Theta_n^N, X_{n+1}^N) + \alpha_{n+1}^2 \varrho_n(\Theta_n^N, X_{n+1}^N) \tag{2.77}$$

with initial conditions

$$\Theta_0^N = x, \; X_0^N = a. \tag{2.78}$$

Two observations can immediately be made:

1) The law $P_{x,a}^N$ of $\{\Theta_n^N\}_{n \geq N+1}$ for the initial condition $\Theta_0^N = x$, $X_0^N = a$ is the conditional law of $\{\Theta_k\}_{k \geq N}$ given $X_N = x$, $\Theta_N = a$.

2) Let the continuous process associated with $\{\Theta_n^N\}_{n \geq 0}$ be

$$\Theta^N(t) = \Theta_n^N \quad \text{for } t, \text{ such that } \sum_{i=N+1}^{N+n} \alpha_i \leq t < \sum_{i=N+1}^{N+n+1} \alpha_i \tag{2.79}$$

It follows from Section 2.3 that if $\alpha_N \downarrow 0$ when $N \to \infty$, for each T and $\delta > 0$

$$\lim_{N \to \infty} P_{x,a}^N \left(\sup_{t \leq T} |\Theta^N(t) - \bar{\Theta}_a(t)| \geq \delta \right) = 0 \tag{2.80}$$

where $\bar{\Theta}_a(t)$ is the solution of equation (2.3) with the initial condition $\bar{\Theta}_a(0) = a$.

This asymptotic approximation of the tail of the algorithm by the deterministic function $\bar{\Theta}_a(t)$ and the estimates of the previous section can be used to derive the asymptotic properties of the sequence $\{\Theta_n\}$ when equation (2.3) is assumed to have locally asymptotic stable equilibrium Θ^* (or stable equilibria).

When several locally asymptotic stable equilibria exist, the situation is more complicated. This leads to the statement that, given $(\Theta_N, X_N,) = (a, x)$, the probability of convergence of the algorithm to the attractor of (a, x) tends to 1 when $N \to \infty$. A set of special conditions must be imposed in order to obtain boundedness and convergence of the algorithm.

Recall that if $\bar{\Theta}^* \in D$ is an asymptotically stable point for the equation (2.3) with domain of attraction D, then the solution of (2.3) with initial condition

$a \in D$ stays in D and converges to $\bar{\Theta}^*$ when $t \to \infty$. It is then possible to show (see (Krasovskii [1963, Th.5.3. p.31]) the existence of a C^2 function U on D (a Lyapunov function) such that

(i) $U(\bar{\Theta}^*) = 0$; $U(\bar{\Theta}) > 0$ for all $\bar{\Theta} \in D$ $\bar{\Theta} \neq \bar{\Theta}^*$

(ii) $U'(\bar{\Theta})h(\bar{\Theta}) < 0$ for all $\bar{\Theta} \in D$ $\bar{\Theta} \neq \bar{\Theta}^*$

(iii) $U(\bar{\Theta}) \to \infty$ if $\bar{\Theta} \to \partial D$ or $|\bar{\Theta}| \to \infty$.

We shall consider a slightly more general situation where the domain of attraction can be a compact set $F \subset D$ and therefore introduce the following hypotheses

[H.6] Assume that α_n as in [H.1] and

$$\sum_{n \geq 0} \alpha_n^{\lambda} < \infty, \qquad \lambda > 1 \tag{2.81}$$

[H.7] There exists a positive function U on D, which is twice continuously differentiable, such that $U(\Theta) \to C \leq \infty$ if $\Theta \to \partial D$ or $|\Theta| \to +\infty$ and $U(\Theta) < C$ if $\Theta \in D$. Moreover,

$$U'(\Theta)h(\Theta) \leq 0 \text{ for all } \Theta \in D. \tag{2.82}$$

Let

$$K(c) = \{\Theta : U(\Theta) \leq c\} \tag{2.83}$$

$$\bar{\tau}(c) = \tau(K(c)) = \inf(n : \Theta_n \notin K(c)) \tag{2.84}$$

$$\nu(c) = \inf(n : \Theta_n \in K(c)) \tag{2.85}$$

$$q_0(\lambda) = \sup(2, 2(\lambda - 1)) \tag{2.86}$$

and consider a compact set $F \subset D$ such that

$$F = \{\Theta : U(\Theta) \leq c_0\}. \tag{2.87}$$

Hypothesis [H.7] is true if $F = \{\bar{\Theta}^*\}$, $c_0 = 0$ and D is the domain of attraction of $\bar{\Theta}^*$.

Theorem 2.4.1 *Assume [H.1]-[H.7] hold, and assume F is a compact set satisfying (2.87). Then for every compact $Q \subset D$ and $q \geq q_0(\lambda)$ there exist constants B and s such that for all $N \geq 0, a \in Q$ and all x*

$$P^N_{x,a}\{(\Theta^N_n)_{n\geq 0} \text{ tends to } F\} \geq 1 - B(1+|x|^s) \sum_{k=N+1}^{\infty} \alpha_k^{1+q/2}. \tag{2.88}$$

Proof: (see Section 2.5)

The following classical form of the "convergence theorem" for stochastic algorithms can be deduced from this theorem. This type of theorem has been popularized by the classical works (Kushner & Clark [1978]) and (Ljung [1977]).

Theorem 2.4.2 *Assume [H.1]-[H.7] hold, and assume Θ^* is a locally asymptotically stable equilibrium of ODE with domain of attraction D. Let Q be a compact subset of D and Y a positive finite R.V. Define*

$$\Omega(Q,Y) = \{\omega : \text{ for infinitely many } n,\ \Theta_n(\omega) \in Q \text{ and } |\Theta_n(\omega)| \leq Y(\omega)\} \tag{2.89}$$

Then $\Theta_n(\omega)$ converges to Θ^* a.s. for $\omega \in \Omega(Q,Y)$.

Proof:

Let

$$A := \{\omega : \Theta_n(\omega) \text{ converges to } \Theta^*\} \tag{2.90}$$

and

$$\Omega_m = \{\omega : \Theta_n(\omega) \in Q \text{ infinitely often and } |\Theta_n(\omega)| \leq m\}. \tag{2.91}$$

Clearly Ω_m increases to $\Omega(Q,Y)$ when $m \to \infty$. We define $t_k := \inf\{n > t_{k-1} : \Theta_n \in Q \text{ and } |\Theta_n| \leq m\}$ with $t_0 := 0$. By construction, the sequence $\{t_k\}$ is strictly increasing and the t_k are finite on Ω_m. Moreover for ω in $\{\omega : t_k(\omega) < \infty\}$, the set A is invariant by the time translation $t_k(\omega)$. The Markov property of (Θ_n, X_n) implies

$$\begin{aligned}
P(A^c \cap \Omega_m) &\leq \lim_{k\to\infty} P(A^c \cap (t_k < \infty)) && (2.92)\\
&\leq \lim_{k\to\infty} E[\,1_{\{t_k<\infty\}} P^{t_k}_{X_{t_k},\Theta_{t_k}}(A^c)\,] && (2.93)\\
&\leq \lim_{k\to\infty} B_4(1+|m|^{s_4}) \sum_{i=k+1}^{\infty} \alpha_i^{1+q/2} && (2.94)\\
&= 0. && (2.95)
\end{aligned}$$

Therefore,

$$0 \leq P(A^c \cap \Omega) \leq \sum_{m=1}^{\infty} P(A^c \cap \Omega_m) = 0. \tag{2.96}$$

■

2.5 Proof of Theorem 2.4.1

The proof of Theorem 2.4.1 relies on the following four lemmas. From the definition of $\varepsilon_n(\Phi)$ in equation (2.44), we see that

Lemma 2.5.1 *Under the hypotheses of Proposition 1, the following inequalities hold for some constants B and $S \geq 0$:*

$$(i) \quad E_{x,a}\left[\sup_n \sup_{n<k\leq m(n,T)} 1_{\{k\leq \nu(\varepsilon,Q)\}} |\sum_{i=n}^{k-1} \varepsilon_i(\Phi)|^q\right] \tag{2.97}$$

$$\leq 3^q B(1+T)^{q-1}(1+|x|^s)\sum_{i\geq 1} \alpha_i^{1+q/2} \tag{2.98}$$

$$(ii) \quad \text{If } \textstyle\sum_{i\geq 1} \alpha_i^{1+q/2} < \infty, \text{ then on } \{\nu(\varepsilon,Q) = +\infty\} \tag{2.99}$$

$$\lim_{n\to\infty} \sup_{n<k\leq m(n,T)} |\sum_{i=n}^{k-1} \varepsilon_i(\Phi)| = 0 \quad P_{x,a} - \ a.s. \tag{2.100}$$

Proof:

We see that

$$\sup_{n<k\leq m(n,T)} 1_{\{k\leq\nu\}} \sum_{i=n}^{k-1} \varepsilon_i(\Phi) = \sup_{n<k\leq \nu\wedge m} \sum_{i=n}^{k-1} \varepsilon_i(\Phi) \tag{2.101}$$

$$\leq \sup_{n<k\leq m} \sum_{i=n}^{k-1} I(i+1\leq\nu)\varepsilon_i(\Phi). \tag{2.102}$$

Set

$$Z_i = 1_{\{i+1\leq\nu\}}\varepsilon_i(\Phi) \tag{2.103}$$

and define recursively $n_r := m(n_{r-1}, T)$ with $n_0 := 0$. For $n \in [n_r, n_{r+1}]$

$$\sum_{i=n}^{k-1} Z_i = \sum_{i=n_r}^{k-1} Z_i - \sum_{i=n_r}^{n-1} Z_i \qquad \text{if } k \in [n_r, n_{r+1}] \tag{2.104}$$

and

$$\sum_{i=n}^{k-1} Z_i = \sum_{i=n_r}^{n_{r+1}-1} Z_i + \sum_{i=n_{r+1}}^{k-1} Z_i - \sum_{i=n_r}^{n-1} Z_i \qquad \text{if } k \notin [n_r, n_{r+1}] \tag{2.105}$$

From this we derive

$$\sup_{n \geq n_p} \left(\sup_{n<k\leq m(n,T)} 1_{\{k\leq n\}} \Big| \sum_{i=n}^{k-1} \varepsilon_i(\Phi) \Big|^q \right) \tag{2.106}$$

$$\leq 3^q \sup_{r\geq p} \sup_{n_2<k\leq n_{r+1}} 1_{\{k\leq n\}} \Big| \sum_{i=n_r}^{k-1} \varepsilon_i(\Phi) \Big|^q. \tag{2.107}$$

Statement (i) follows from this inequality and from Proposition 1 since

$$E\left[\sup_{r\geq 0} \sup_{n_r<k\leq n_{r+1}} 1_{\{k\leq r\}} \Big| \sum_{i=n_r}^{k-1} \varepsilon_i(\Phi) \Big|^q \right] \tag{2.108}$$

$$\leq \sum_{r\geq 0} E\left[\sup_{n_r<k\leq n_{r+1}} 1_{\{k\leq n\}} \Big| \sum_{i=n_r}^{k-1} \varepsilon_i(\Phi) \Big|^q \right] \tag{2.109}$$

$$\leq B(1+T)^{q-1}(1+|x|^s) \sum_{r\geq 0} \sum_{i=n_r}^{n_{r+1}-1} \alpha_i^{1+q/2}. \tag{2.110}$$

The property (ii) follows equally since the inequality

$$E\left[\sum_{r\geq 0} \sup_{n_r<k\leq n_{r+1}} 1_{\{k\leq n\}} \Big| \sum_{i=n_r}^{k-1} \varepsilon_i(\Phi) \Big|^q \right] < \infty \tag{2.111}$$

implies

$$\lim_{r\to\infty} \sup_{n_r<k\leq n_{r+1}} \Big| \sum_{i=n_r}^{k-1} \varepsilon_i(\Phi) \Big|^q = 0 \quad \text{a.s. on } \{\nu = +\infty\}. \tag{2.112}$$

■

Lemma 2.5.2 *Given c_1, c_2, q such that $c_0 < c_1 < c_2 < C$ and $q \geq q_0(\lambda)$ (λ in [H.6]) , there exist ε_0, B_2, s_2 such that for $\varepsilon \leq \varepsilon_0, a \in K(c_1)$ and x:*

$$P_{x,a}(\tau(c_2) < \infty, \sigma(\varepsilon) > \tau(c_2)) \leq B_2(1+|x|^{s_2}) \sum_{k\geq 1} \alpha_k^{1+q/2}. \tag{2.113}$$

(For the definition of $\tau(c)$ see (2.84) and of $\sigma(\varepsilon)$ see (2.27))

Proof:

Let ε_0 be given from hypothesis [H'.5]. According to [H.2] and (2.87) there exist $\eta > 0$ such that, for every $\Theta \in K(c_2) - K(c_1)$

$$U'(\Theta)h(\Theta) \leq -\eta < 0. \tag{2.114}$$

Let T satisfy $(T-1)\eta \geq c_2 - c_1$ and let Φ be a C^2-function on $\Re^d$, with bounded second order derivatives, equal to U on $K(c_2)$ and greater than or equal to c_2 on the complement of $K(c_2)$. We define the following integer-valued random variables:

$$\sigma := \sup\{n : n \leq \tau(c_2),\ \Theta_n \in K(c_1)\} \tag{2.115}$$

$$\mu := \inf\{n : n > \sigma, \alpha_{\sigma+1} + \cdots + \alpha_{n+1} \geq T\}. \tag{2.116}$$

(According to definition (2.23): $\mu = m(\sigma, T)$). Set

$$\Omega_1 := \{\omega : \tau(c_2) < \infty,\ \sigma(\varepsilon) > \tau(c_2)\} \tag{2.117}$$

$$\overline{\mu} := \mu \wedge \tau(c_2). \tag{2.118}$$

For ω in Ω_1,

$$\bar{\mu} \leq \tau(c_2) \wedge \sigma(\varepsilon) = \nu(\varepsilon, K(c_2)) \tag{2.119}$$

Formula (2.44) gives

$$\Phi(\Theta_{\overline{\mu}}) - \Phi(\Theta_\sigma) - \sum_{i=\sigma}^{\overline{\mu}-1} \alpha_{i+1}\Phi'(\Theta_i) \cdot h(\Theta_i) = \sum_{i=\sigma}^{\overline{\mu}-1} \varepsilon_i(\Phi). \tag{2.120}$$

For ω in Ω_1, the left hand of (2.120) is greater than $c_2 - c_1$. If $\overline{\mu} = \tau(c_2)$ and if $\bar{\mu} = \mu < \tau(c_2)$ then from (2.114) it is greater than $\eta(T-1) \geq c_2 - c_1$. Therefore

$$(c_2 - c_1)^q P_{x,a}(\Omega_1) \leq E_{x,a}\left[1_{\{\Omega_1\}} |\sum_{i=\sigma}^{\bar{\mu}-1} \varepsilon_i(\Phi)|^q\right] \tag{2.121}$$

$$\leq E_{x,a}\left[\sup_n \sup_{n<k\leq m(n,T)} |\sum_{i=n}^{k-1} \varepsilon_i(\Phi)|^q\right]. \tag{2.122}$$

An application of Lemma 2.5.1 gives Lemma 2.5.2. ∎

Next, we have

Lemma 2.5.3 *Let c_1, c_2, q be such that $c_0 < c_1 < c_2 < C$ and $q \geq q_0(\lambda)$, then there exist ε_0, B_3, s_3 such that for all $a \in K(c_1)$ and all x*

$$P_{x,a}(\sigma(\varepsilon_0) = +\infty, \tau(c_2) = +\infty) \geq 1 - B_3(1 + |x|^{s_3}) \sum_{k\geq 1} \alpha_k^{1+q/2}. \tag{2.123}$$

Proof:

The complement of the set $\{\sigma(\varepsilon_0) = +\infty,\ \tau(c_2) = +\infty\}$ is $\{\tau(c_2) < \infty,\ \sigma(\varepsilon_0) \geq \tau(c_2)\} \cup \{\sigma(\varepsilon_0) < +\infty, \tau(c_2) \geq \sigma(\varepsilon_0)\}$ where ε_0 is the constant in hypothesis [H'.5]. Applying Lemma 2.5.2 to

$$P_{x,a}(\tau(c_2) < \infty, \sigma(\varepsilon_0) > \tau(c_2)) \tag{2.124}$$

and applying Lemma 2.3.1 we see that

$$P_{x,a}(\sigma(\varepsilon_0) \leq \tau(c_2), \sigma(\varepsilon_0) < \infty) \leq M(1 + |x|^{s_1}) \sum_{k \geq 1} \alpha_k^q. \tag{2.125}$$

This gives Lemma 2.5.3. ∎

Finally, we have

Lemma 2.5.4 *Let c and ε satisfy $c_0 < c$, $\varepsilon \leq \varepsilon_0$. Then for every x and every a in the interior of $K(c)$, the sequence $\{\Theta_n\}$ converges a.s. to F for ω in $\{\omega : \tau(c) = +\infty, \sigma(\varepsilon) = +\infty\}$*

Proof:

Let c_1 be any number between c_0 and c. Set

$$\Omega_2 := \{\tau(c) = +\infty, \sigma(\varepsilon) = +\infty\} \cap \{\limsup U(\Theta_n) > c_1\}. \tag{2.126}$$

The lemma follows if we show that

$$P_{x,a}(\Omega_2) = 0. \tag{2.127}$$

Let c' satisfy $c_0 < c' < c_1 < c$. In view of [H.7] and (1.1.4) there exist $\eta > 0$ such that, for $\Theta \in K(c) \setminus K(c')$

$$U'(\Theta)h(\Theta) \leq -\eta. \tag{2.128}$$

Choose T big enough for

$$(T-1)\eta - c \geq c_1 - c'. \tag{2.129}$$

A sequence (V_r, W_r) of integer-valued random variables can be constructed such that on Ω_2

$$r < V_r < W_r \leq m(V_r, T) \tag{2.130}$$

and

$$U(\Theta_{W_r}) - U(\Theta_{V_r}) - \sum_{i=V_r}^{W_{r-1}} \alpha_{i+1} U'(\Theta_i).h(\Theta_i) \geq c_1 - c'. \tag{2.131}$$

If Φ is a regular extension of U outside $K(c)$, then on Ω_2

$$0 < c_1 - c' \leq \sum_{i=V_r}^{W_{r-1}} \varepsilon_i(\Phi) \leq \sup_{n>r} \sup_{n<k\leq m(n,T)} |\sum_{i=n}^{k-1} \varepsilon_i(\Phi)|. \tag{2.132}$$

According to Lemma 2.5.1(ii) this quantity tends to zero a.s. on Ω_2 which implies that $P_{x,a}(\Omega_2) = 0$ and hence the lemma follows from the construction of the sequence (V_r, W_r) satisfying (2.130) and (2.131) hold.

Next we show how to construct this sequence. Let N be given and set $\sigma := \inf\{n \geq N, \Theta_n \in K(c')\}$

1st Case : $\sigma = +\infty$.

Set $V = N, W = m(N,T)$. The property (2.131) then holds for (V,W) since on $[V,W)$, $\Theta_n \in K(c) \setminus K(c')$, and $U(\Theta_W) - U(\Theta_V) \geq -c$.

2nd Case: $\sigma < \infty$

Set $\mu := \inf\{n > \sigma, \Theta_n \notin K(c_1)\}$ and observe that for ω in Ω_2, μ is less than infinity. Define

$$\bar{\varrho} := \sup\{n \geq \sigma, n \leq \mu, \Theta_n \in K(c')\} \tag{2.133}$$

$$\bar{\mu} := \inf\{n > \bar{\sigma}, \alpha_{\bar{\sigma}+1} + \cdots + \alpha_{n+1} \geq T\}. \tag{2.134}$$

Let $V = \bar{\sigma}$, $W = \mu \wedge \bar{\mu}$.

(1) If $\mu \leq \bar{\mu}$ then $\Theta_W \notin K(c_1)$ and $\Theta_V \in K(c')$. Therefore $U(\Theta_W) - U(\Theta_V) \geq c_1 - c'$.

(2) If $\mu \geq \bar{\mu}$ then $\Theta_W \notin K(c')$ and $\Theta_V \in K(c')$ for every i such that $V \leq i \leq W$. Hence one has $\Theta_V \in K(c) - K(c')$, which implies

$$-\sum_{i=V}^{W-1} \alpha_{i+1} U'(\Theta_i).h(\Theta_i) \geq \eta T \geq c_1 - c'. \tag{2.135}$$

In both case, V and W have been chosen such that $N < V < W$ and (3.4.4) holds. This procedure can be applied for $N = 1$ and then recursively to obtain the sequence (V_r, W_r). This completes the proof of Lemma 2.5.4. ∎

Lemma 2.5.3 and Lemma 2.5.4 together give Theorem 2.4.1.

Chapter 3

Learning Vector Quantization

In this chapter we discuss Learning Vector Quantization (LVQ), a method for non-parametric classification proposed in (Kohonen [1986]). We present a modification to the algorithm yielding classification regions for a larger set of initial conditions. We prove that the algorithm converges to asymptotically stable points of an ordinary differential equation. Finally, we demonstrate that as a certain parameter becomes large, it is possible to closely approximate the optimal Bayes risk function.

In Chapter 1, we showed that the optimal decision regions can be calculated directly from the pattern densities. To illustrate, suppose there are two patterns and that each pattern density is Gaussian with zero mean. Figure 3.1 shows a plot of two such pattern densities. Here pattern 1 has a variance equal to 1, and pattern 2 has a variance equal to 4. The decision regions are easy to calculate if we follow the Bayes decision rule for minimum error and assume that each pattern

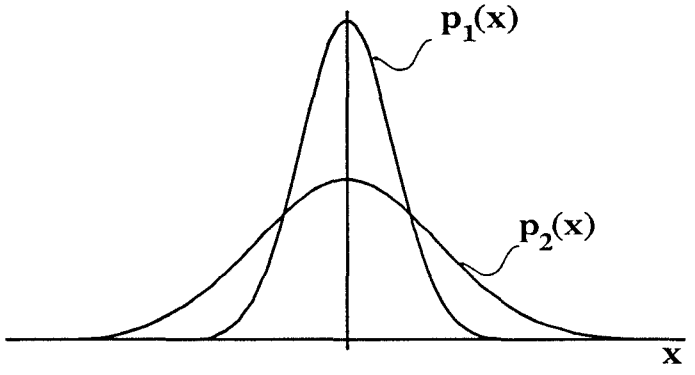


Figure 3.1: *Plot of two pattern densities*

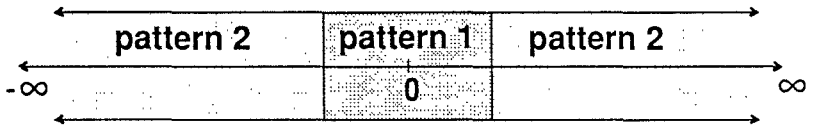


Figure 3.2: *Plot of decision regions*

is equally likely. These regions are calculated using (1.2) and are displayed in Figure 3.2.

The decision regions are computed using the individual pattern densities. However, the pattern densities are usually not available, instead, the only knowledge available is a set of independent observations of each pattern. In Chapter 1 we showed that if we use consistent nonparametric density estimators and if the density estimators are legitimate densities, then the approximate risk approaches the optimal risk as the number of observations approaches infinity.

Continuing with the example above, we see that for both densities a majority of the observations occur near zero. Nonparametric density estimation schemes try to minimize the expected error. In this example, estimates of both densities will try to minimize the error near zero since that is where most of the observations are located. Since we are only calculating the densities in order to calculate the optimal decision regions, we need to be concerned with the fact that the errors in the density estimates contribute to errors in the resulting classifier. In general, it is hard to predict how this two step approach will behave. LVQ is an algorithm which attempts to alleviate this problem by estimating the decision regions directly. Unlike some other nonparametric classification schemes, it does not first estimate the densities and then proceed to calculate the decision regions.

The idea behind LVQ is to perform vector quantization using the absolute value of the difference of the two pattern densities. In this example, this is the function displayed in Figure 3.3. This function can be used as a density function for the vector quantization algorithm. As was shown in Chapter 1, given a vector quantizer it is possible to construct a consistent density estimate. Applying those results here, we see that the vectors in vector quantization can be employed to construct a consistent estimate of the optimal decision regions. The resulting

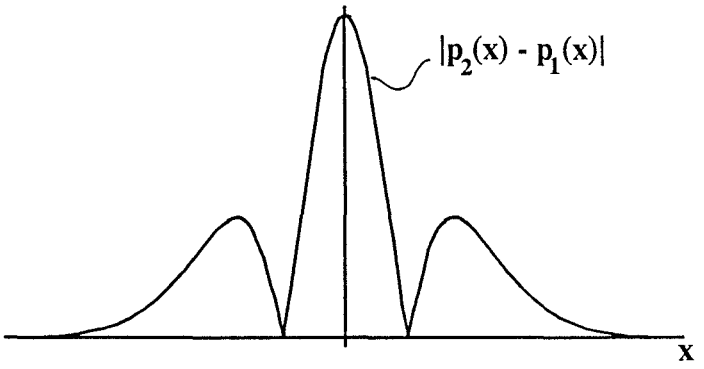


Figure 3.3: *Absolute value of the difference of the pattern densities*

quantization vectors can be used to define decision regions via a majority vote of the observations that fall in their Voronoi cells.

In LVQ, vectors representing averages of past observations are calculated. These vectors are called Voronoi vectors. Each vector defines a region in the observation space and hence characterizes an associated decision class. In the classification phase, a new observation is compared to all of the Voronoi vectors. The closest Voronoi vector is found and the observation is classified according to the class of that closest Voronoi vector. Hence, around each Voronoi vector is a region, called the Voronoi cell, which defines an equivalence class of points all belonging to the decision class of that vector. An example of eight Voronoi vectors in $\Re^2$ and their associated Voronoi cells are shown in Figure 3.4. LVQ is similar to nearest neighbor classification except that only the nearest Voronoi vector is found instead of finding the nearest past observation.

In the design or learning phase, a set of training data consisting of already classified past observations is used to adjust the locations and the decisions of the Voronoi vectors. The vectors are initialized by setting both the initial locations and the initial decisions. Once the initial locations are fixed, the initial decisions are found by a simple majority vote of all the past observations falling in each Voronoi cell. This initialization process is discussed in full detail in Section 3.6. The vectors are then adjusted by a gradient search type algorithm. Specifically, an

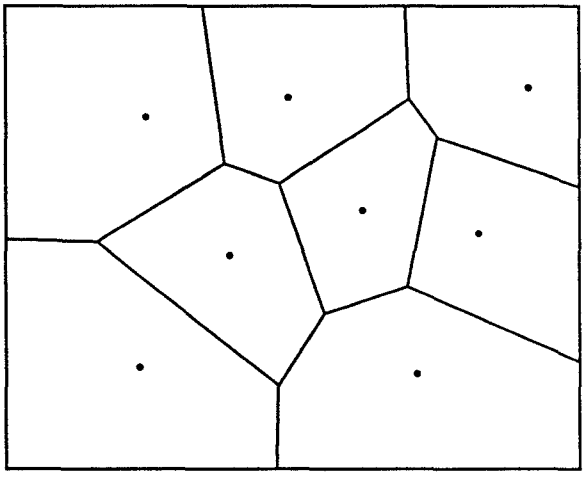


Figure 3.4: *Voronoi vectors and their Voronoi cells*

observation is picked at random from the past observations; if the decision of the closest Voronoi vector and the decision associated with the new observation agree, then the Voronoi vector is moved in the direction of the observation, if however the decisions disagree then the Voronoi vector is moved away from that observation. This process is continued for several iterations through the past observations until all the Voronoi vectors' locations converge.

The heuristic idea behind this adjustment rule is that if the decision of the new observation and the decision of the closest vector agree then the Voronoi cell is probably close to the correct position and the Voronoi vector should be moved closer to that observation, conversely, if the decisions disagree then the Voronoi vector should move away from that observation. On the average, the vectors will converge to positions which approximate the optimal decision regions. We will make this more precise in the sections to follow. The amazing feature of this algorithm is that it only takes a small number of vectors to get satisfactory classification results as will be seen from the simulation results presented in Chapter 4.

3.1 Description of the Algorithm

The LVQ algorithm was originally presented in (Kohonen [1986]). In what follows, we descrbe the LVQ algorithm. To begin with, let the past observations lie in

$\Re^d$ and let $\Theta = \{\theta_1, \ldots, \theta_k\}$ be the Voronoi vectors. The observation space is partitioned into Voronoi cells. Each Voronoi cell has a defining vector θ_i and an associated decision class d_{θ_i}. The cell consists of all points in the observation space which are closer to that vector than to any other Voronoi vector. An observation x is classified as type d_{θ_i} if it falls within the Voronoi cell defined by θ_i. Let $\rho(\theta, x)$ be a cost function satisfying the conditions described in Section 1.5. Voronoi cells are characterized mathematically by

$$V_{\theta_i} = \{\, x \in \Re^d \,|\, \rho(\theta_i, x) < \rho(\theta_j, x),\ j \neq i\} \qquad i = 1, \ldots, k. \tag{3.1}$$

By convention, we assign equidistant points to that Voronoi cell with the lowest index.

The algorithm for adjusting the vectors θ_i is now described. Let $\{(y_n, d_{y_n})\}_{n=1}^N$ be the past observations set. This means that y_n is observed and has as its pattern class d_{y_n}. In order for this problem to be well-posed, we assume that there are many more observations than Voronoi vectors (see (Duda & Hart [1973])), i.e., N is much greater than k.

Once the Voronoi vectors are initialized, training proceeds by taking a sample (y_n, d_{y_n}) from the past observation data set, finding the ρ–closest Voronoi vector, say θ_c, and then adjusting θ_c as follows:

$$\theta_c(n+1) = \theta_c(n) - \alpha_n\ \nabla_\theta \rho(\theta_c(n), y_n) \tag{3.2}$$

if $d_{\theta_c} = d_{y_n}$ and

$$\theta_c(n+1) = \theta_c(n) + \alpha_n\ \nabla_\theta \rho(\theta_c(n), y_n) \tag{3.3}$$

if $d_{\theta_c} \neq d_{y_n}$. Here n is the iteration number. In words, if y_n and $\theta_c(n)$ have the same decision then $\theta_c(n)$ is moved closer to y_n, however, if they have different decisions then $\theta_c(n)$ is moved away from y_n. The constants $\{\alpha_n\}$ are positive and nonincreasing. Notice that only the Voronoi vector which is closest to the observation is adjusted by the algorithm. The other vectors remain unchanged.

In the next section, we show convergence of the algorithm in two cases: (1) when the number of past observations becomes arbitrarily large and each observation is presented once and (2) when the number of past observations is fixed and the number of presentations of each observation becomes arbitrarily large. In both

cases, convergence is shown by finding the function $h(\Theta)$ in the associated ODE and studying its properties in order to apply the convergence theorems of Chapter 2.

3.2 Convergence to Stationary Points

The convergence theorems of Chapter 2 show that as the number of iterations goes to infinity, the estimate Θ_n converges to $\bar{\Theta}^*$, an asymptotic stable equilibrium of the associated ODE (2.3). Given an iterative scheme of the form (2.19), one only needs to find the function $h(\Theta)$ in order to study the convergence properties of that scheme. In this section, we find $h(\Theta)$ for the case of an infinite number of observations and the case of a finite number of observations. First, we present the LVQ algorithm precisely.

The LVQ algorithm has the general form

$$\theta_i(n+1) = \theta_i(n) + \alpha_n \, \gamma(d_{y_n}, d_{\theta_i(n)}, y_n, \Theta_n) \, \nabla_\theta \rho(\theta_i(n), y_n) \tag{3.4}$$

where the function γ determines whether there is an update and what its sign should be. It is given by

$$\gamma(d_{y_n}, d_{\theta_i(n)}, y_n, \Theta_n) = \begin{cases} -1_{\{y_n \in V_{\theta_i}\}} & \text{if } d_{y_n} = d_{\theta_i(n)} \\ 1_{\{y_n \in V_{\theta_i}\}} & \text{if } d_{y_n} \neq d_{\theta_i(n)} \end{cases} \tag{3.5}$$

or, more compactly,

$$\gamma(d_{y_n}, d_{\theta_i(n)}, y_n, \Theta_n) = -1_{\{y_n \in V_{\theta_i}\}}(1_{\{d_{y_n} = d_{\theta_i}\}} - 1_{\{d_{y_n} \neq d_{\theta_i}\}}). \tag{3.6}$$

This is a stochastic approximation algorithm with $\varrho_n(\Theta, x) \equiv 0$ (see (2.19)). It has the form

$$\Theta_{n+1} = \Theta_n + \alpha_n \, H(\Theta_n, z_n) \tag{3.7}$$

where Θ is the vector with components θ_i; $H(\Theta, z)$ is the vector with components defined in the obvious manner in (3.4) and z_n is the random pair consisting of the observation and the associated *true* pattern number. If the appropriate conditions are satisfied by α_n, H, and z_n, then Θ_n approaches the solution of

$$\frac{d}{dt}\bar{\Theta}(t) = h(\bar{\Theta}(t)) \tag{3.8}$$

for the appropriate choice of $h(\Theta)$ (Theorems 2.3.1, 2.4.2).

Throughout this section we consider the case of two pattern densities. In the subsections below we treat convergence separately for the cases of infinite past observations presented consecutively and finite past observations presented infinitely many times. In both cases we obtain convergence via the ODE method discussed in Chapter 2.

3.2.1 Convergence for an Infinite Number of Observations

In this section, we discuss convergence for the LVQ algorithm as the number of observations becomes arbitrarily large. Throughout this section we assume that the Voronoi vectors are ordered so that the first k_0 vectors have decision class equal to pattern 1 and the remaining have decision class equal to pattern 2.

It is shown next that the function $h(\Theta)$ of the associated ODE takes the form

$$h(\Theta) = \begin{pmatrix} h_1(\Theta) \\ \vdots \\ h_{k_0}(\Theta) \\ \\ h_{k_0+1}(\Theta) \\ \vdots \\ h_k(\Theta) \end{pmatrix} = \begin{pmatrix} \int_{V_{\theta_1}} q(x)\ \nabla_{\theta_1}\rho(\theta_1, x)\, dx \\ \vdots \\ \int_{V_{\theta_{k_0}}} q(x)\ \nabla_{\theta_{k_0}}\rho(\theta_{k_0}, x)\, dx \\ \\ -\int_{V_{\theta_{k_0+1}}} q(x)\ \nabla_{\theta_{k_0+1}}\rho(\theta_{k_0+1}, x)\, dx \\ \vdots \\ -\int_{V_{\theta_k}} q(x)\ \nabla_{\theta_k}\rho(\theta_k, x)\, dx \end{pmatrix} \tag{3.9}$$

with $q(x) = p_2(x)\,\pi_2 - p_1(x)\,\pi_1$. If we let

$$f_i(\Theta, x) = 1_{\{x \in V_{\theta_i}\}}\ \nabla_{\theta_i}\rho(\theta_i, x)\ \left(1_{\{i \le k_0\}} - 1_{\{i > k_0\}}\right) \tag{3.10}$$

then we see from (3.9) that

$$h_i(\Theta) = \int_\Omega f_i(\Theta, x)\, q(x)\, dx. \tag{3.11}$$

We assume that the training data $\{z_n\}_{n=1}^N$ consist of pairs of independent, identically distributed observations. The second component of the pair represents

the pattern that was *true* when the first component was observed. For example, a generic pair in the training data can be represented as $z_n = (y_n, d_{y_n})$ with

$$\pi_2 = P(\, d_{y_n} = 2\,) \quad \text{and} \quad \pi_1 = P(\, d_{y_n} = 1\,), \tag{3.12}$$

$\pi_1 + \pi_2 = 1$. For each n, y_n is distributed according to the probability density function $p_2(y)$ when $d_{y_n} = 2$ and according to $p_1(y)$ when $d_{y_n} = 1$.

Next we show that $H_i(\Theta_n, z_n) = h_i(\Theta_n) + \xi_i(n)$ where $\xi_i(n)$ is a noise sequence. Let E_z denotes the expectation with respect to the random variable z_n where we have dropped the subscript n for ease of notation and let E_1 (resp. E_2) denote the expectation with respect to $p_1(y)$ (resp. $p_2(y)$). To begin the analysis,

$$\begin{aligned}
E_z[\, H_i(\Theta, z)\,] &= E_z\left[1_{\{d=1\}} H_i(\Theta, (y,1))\right] + E_z\left[1_{\{d=2\}} H_i(\Theta, (y,2))\right] && (3.13)\\
&= E_1\left[\, H_i(\Theta, (y,1))\right] \pi_1 + E_2\left[\, H_i(\Theta, (y,2))\right] \pi_2 && (3.14)\\
&= E_1\left[\, \gamma(1, d_{\theta_i}, y, \Theta)\ \nabla_{\theta_i} \rho(\theta_i, y)\right] \pi_1 \\
&\qquad + E_2\left[\, \gamma(2, d_{\theta_i}, y, \Theta)\ \nabla_{\theta_i} \rho(\theta_i, y)\right] \pi_2 && (3.15)\\
&= E_1\left[1_{y \in V_{\theta_i}} \left(-1_{\{i \le k_0\}} + 1_{\{i > k_0\}}\right) \nabla_{\theta_i} \rho(\theta_i, y)\right] \pi_1 \\
&\qquad + E_2\left[1_{y \in V_{\theta_i}} \left(1_{\{i \le k_0\}} - 1_{\{i > k_0\}}\right) \nabla_{\theta_i} \rho(\theta_i, y)\right] \pi_2 && (3.16)\\
&= -E_1\left[\, f_i(\Theta, y)\right] \pi_1 + E_2\left[\, f_i(\Theta, y)\right] \pi_2 && (3.17)\\
&= h_i(\Theta). && (3.18)
\end{aligned}$$

From the results above we see that $\xi_i(n)$ is a zero mean process with variance given by

$$E_z\left[\ \|H_i(\Theta, z) - h_i(\Theta)\|^2\ \right] = E_z\left[\ \|H_i(\Theta, z)\|^2\right] - \|h_i(\Theta)\|^2 \tag{3.19}$$

where

$$\begin{aligned}
E_z\left[\ \|H_i(\Theta, z)\|^2\ \right] &= E_z\left[\, \|\nabla_{\theta_i} \rho(\theta_i, y)\|^2\right] && (3.20)\\
&= E_1\left[\|\nabla_{\theta_i} \rho(\Theta, y)\|^2\right] \pi_1 + E_2\left[\|\nabla_{\theta_i} \rho(\Theta, y)\|^2\right] \pi_2 && (3.21)\\
&= \sum_{i=1}^{k} \int_{V_{\theta_i}} \|\, \nabla_{\theta_i} \rho(\theta_i, x)\|^2\ (p_1(x)\, \pi_1 + p_2(x)\, \pi_2)\, dx && (3.22)
\end{aligned}$$

For the remainder of this chapter we assume that $\rho(\theta, x)$ satisfies the following three properties:

(a) $\rho(\theta, x)$ is a twice continuously differentiable function of θ and x and for every fixed $x \in \Re^d$ it is a convex function of θ.

(b) For any fixed x, if $\theta(k) \to \infty$ as $k \to \infty$, then $\rho(\theta(k), x) \to \infty$.

(c) For every compact $Q \subset \Re^d$, there exist constants C_1 and q_1 such that for all $\theta \in Q$

$$|\nabla_\theta \rho(\theta, x)| < C_1(1 + |x|^{q_1}). \tag{3.23}$$

An example of a function which satisfies the properties above is $\rho(\theta, x) = \|\theta_i - x\|^2$.

We now state the two convergence theorems alluded to in Section 3.1.

Theorem 3.2.1 *Let $\{z_n\}$ be the sequence of independent, identically distributed random vectors given above. Suppose $\{\alpha_n\}$ satisfies [H.1],[H.6] and that $\rho(\theta, x)$ satisfies the properties* **(a)–(c)** *above. Assume that the pattern densities $p_1(x)$ and $p_2(x)$ satisfy [H'.5] and $h(\Theta)$ is locally Lipschitz.*

If $\bar{\Theta}_a(t)$ remains in a compact subset of $\Re^d$ for all $t \in [0, T]$, then for every $\delta > 0$ and all $X_0 = x$

$$\lim_{\alpha_1 \downarrow 0} P_{x,a}\{ \sup_{n \le m(T)} |\Theta_n - \bar{\Theta}_a(t_n)| > \delta\} = 0 \tag{3.24}$$

where Θ_n satisfies (3.7) and $\bar{\Theta}_a(t)$ satisfies (3.8) with $h(\Theta)$ defined in (3.9). Here $t_n = \sum_{i=1}^{n} \alpha_i$.

Theorem 3.2.2 *In addition to the conditions of Theorem 3.2.1, assume $\bar{\Theta}^*$ is a locally asymptotically stable equilibrium of (3.8) with domain of attraction D^*. Let Q be a compact subset of D^*. If $\Theta_n \in Q$ for infinitely many n then*

$$\lim_{n \to \infty} \Theta_n = \bar{\Theta}^* \qquad a.s. \tag{3.25}$$

Proof of Theorem 3.2.1:

In view of Theorem 2.3.1, we need only verify that [H.1]–[H'.5] are satisfied.

The observations z_n are independent, identically distributed and are independent of the values of Θ and $\{z_i\}_{i<n}$ therefore $\{\Theta_n, z_n\}$ forms a trivial Markov chain.

If we let $\Pi_\Theta(z, B)$ denote its transition probability then

$$P\{z_{n+1} \in B \,|\, \mathcal{F}_n\} = \Pi_\Theta(z_n, B) \tag{3.26}$$

$$= \int_B p_2(x)\pi_2\, dx + \int_B p_1(x)\pi_1\, dx. \tag{3.27}$$

Hence hypothesis [H.2] is satisfied.

Note that

$$|H_i(\Theta, z)| = |\nabla_{\theta_i}\rho(\theta_i, z)|. \tag{3.28}$$

Therefore, [H.3] is satisfied.

The transition probability function is independent of Θ therefore if we let $\nu(\Theta, z) = H(\Theta, z)$ then

i) $h(\Theta) = \Pi_\theta \nu_\theta$, and therefore [H.4 ii] is satisfied;

ii) $|\nu_{i\Theta}(z)| = |H_i(\Theta, z)| = |\nabla_{\theta_i}\rho(\theta_i, z)|$, and therefore [H.4 iii] is satisfied using by property (**c**).

Therefore, [H.1]–[H'.5] are satisfied, which proves Theorem 3.2.1. ∎

The proof of Theorem 3.2.2 is similar that of Theorem 3.2.1.

3.2.2 Convergence for a Finite Number of Observations

The convergence above applies when the number of observations goes to infinity. Unfortunately, it is usually the case that only a fixed set of data is available. The update in this case consists in picking a point uniformly at random from the observation set and presenting it to the LVQ update. Several iterations are necessary in order to achieve convergence. This method is known as the bootstrap learning method. Next, we explore the convergence properties of the algorithm using a fixed data set of size N.

Let $Z = \{z_n\}_{n=1}^N$ represent the set of past observations and let N_1 represent the number of observations from pattern 1 and N_2 represent the number of observations from pattern 2 in Z. For each update, a point z_{n_j} is picked at random from Z; an update of the LVQ algorithm is performed; the point is returned to Z and the process starts over again. Here $\{z_{n_j}\}_{j=1}^\infty$ represents the sequence of updates. We assume that the points are picked independently with probability $1/N$.

Once Z is given, the randomness in this algorithm enters only through the process of picking the points to be used in the update of the Voronoi vectors. Estimates of the pattern densities based on Z are given by

$$\hat{p}_1(x;N) = \frac{1}{N_1}\sum_{j=1}^{N}\delta(x=y_j)1_{\{d_{y_j}=1\}} \tag{3.29}$$

$$\hat{p}_2(x;N) = \frac{1}{N_2}\sum_{j=1}^{N}\delta(x=y_j)1_{\{d_{y_j}=2\}}, \tag{3.30}$$

and estimates of the priors are given by

$$\hat{\pi}_1 = \frac{N_1}{N} \qquad \text{and} \qquad \hat{\pi}_2 = \frac{N_2}{N} \tag{3.31}$$

where $\delta(x)$ is the delta function. Let $H(\Theta,z)$ be the vector of components defined in (3.4). We see that

$$h_i(\Theta;N) = \hat{E}_z[H_i(\Theta,z)] \tag{3.32}$$

$$= \hat{E}_1[H_i(\Theta,(y,1))]\hat{\pi}_1 + \hat{E}_2[H_i(\Theta,(y,2))]\hat{\pi}_2 \tag{3.33}$$

$$= -\frac{1}{N}\sum_{j=1}^{N}\nabla_{\theta_i}\rho(y_j,\theta_i)\,1_{\{y_j\in V_{\theta_i}\}}\,(1_{\{d_{y_j}=d_{\theta_i}\}} - 1_{\{d_{y_j}\neq d_{\theta_i}\}}). \tag{3.34}$$

where $h(\Theta;N)$ denotes the function based on the N observations. We are now ready to state convergence theorems analogous to those obtained in the case of an infinite number of observations.

Theorem 3.2.3 *Let $\{z_{n_j}\}_{j=1}^{\infty}$ be the independent sequence of random vectors picked from Z as described above. Suppose $\{\alpha_n\}$ satisfies [H.1],[H.6] and $\rho(\theta,x)$ satisfies the properties* (a)–(c).

If $\bar{\Theta}_a(t;N)$ remains in a compact subset of $\Re^d$ for all $t\in[0,T]$, then for every $\delta>0$ and all $X_0=x$

$$\lim_{\alpha_1\downarrow 0} P_{x,a}\{\sup_{n\leq m(T)}|\Theta_n - \bar{\Theta}_a(t_n;N)|>\delta\} = 0 \tag{3.35}$$

where Θ_n satisfies (3.7) and $\bar{\Theta}_a(t;N)$ satisfies (3.8) with $h(\Theta;N)$ defined by (3.34). Here $t_n=\sum_{i=1}^{n}\alpha_i$.

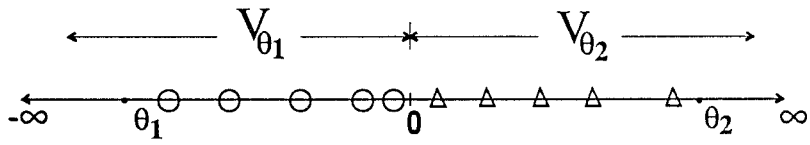


Figure 3.5: *A possible distribution of observations and two Voronoi vectors.*

Theorem 3.2.4 *In addition to the conditions of Theorem 3.2.3, assume* $\bar{\Theta}^*$ *is a locally asymptotically stable solution of (3.8) with* $h(\Theta; N)$ *defined by (3.34) and with domain of attraction* D^*. *Let* Q *be a compact subset of* D^*. *If* $\Theta_n \in Q$ *for infinitely many* n *then*

$$\lim_{n\to\infty} \Theta_n = \bar{\Theta}^* \qquad a.s. \tag{3.36}$$

The proofs of these theorems follow directly from the proofs of Theorem 3.2.1 and Theorem 3.2.2 with $h(\Theta) = h(\Theta; N)$ and $P(z = z_i) = 1/N$. We note that by (SLLN) as N_1 and N_2 go to infinity, $h(\Theta; N)$ converges with probability one to the function $h(\Theta)$ given by (3.9). This follows since by (SLLN) we have that $\hat{p}_1(x; N)$, $\hat{p}_2(x; N)$, $\hat{\pi}_1$ and $\hat{\pi}_2$ converge with probability one to their true values.

3.2.3 Remarks on Convergence

The convergence results above require that the initial conditions are close to the stable points of (3.8), i.e., within the domain of attraction of a stable equilibrium, in order for the algorithm to converge. In the next section we present a modification to the LVQ algorithm which increases the number of stable equilibrium for equation (3.8) and hence increases the chances of convergence. In the remainder of this section we present a simple example which emphasizes a defect of LVQ and suggests an appropriate modification to the algorithm.

Let ○ represent an observation from pattern 2 and let △ represent an observation from pattern 1. We assume that the observations are scalar and that $\rho(\theta, x)$ is the Euclidean distance function. Figure 3.5 shows a possible distribution of observations. Suppose there are two Voronoi vectors θ_1 and θ_2 with decisions 1 and 2, respectively, initialized as shown in Figure 3.5. At each update of the LVQ algorithm, a point is picked at random from the observation set and the Voronoi

vector corresponding to the Voronoi cell within which the point falls is modified. We see that during this update, $\theta_2(n)$ is pushed towards ∞ and $\theta_1(n)$ is pushed towards $-\infty$, hence the Voronoi vectors do not converge.

This divergence happens because the decisions of the Voronoi vectors do not agree with the majority vote of the observations falling in their Voronoi cells. As a result, the Voronoi vectors are pushed away from the origin. This phenomena occurs even though the observation data is bounded. The point here is that if the decision associated with a Voronoi vector does not agree with the majority vote of the observations contained in its Voronoi cell then it is possible for the vector to diverge. A simple solution to this problem is to correct the decisions of all the Voronoi vectors after every adjustment so that their decisions correspond to the majority vote. This is pursued further in the next section.

3.3 The Modified LVQ Algorithm

In this section we investigate how the majority vote correction affects the LVQ algorithm. Recall that during the update procedure in (3.4), the Voronoi cells are changed by changing the location of one Voronoi vector. After an update, the majority vote of the observations in each new Voronoi cell may not agree with the decision previously assigned to that cell. In addition, after the majority vote correction, the number of pattern 1 Voronoi vectors can change. This results in a change in the number k_0 since during the correction a Voronoi vector's associated decision class can be changed from pattern 1 to pattern 2. For this procedure to be mathematically sound, we insist that the correction be done at each iteration[1]. Let

$$g_i(\Theta; N) = \begin{cases} 1 & \text{if } \frac{1}{N}\sum_{j=1}^{N} 1_{\{y_j \in V_{\theta_i}\}} 1_{\{d_{y_j}=1\}} > \frac{1}{N}\sum_{j=1}^{N} 1_{\{y_j \in V_{\theta_i}\}} 1_{\{d_{y_j}=2\}} \\ 2 & \text{otherwise.} \end{cases} \tag{3.37}$$

Clearly, g_i represents the decision of the majority vote of the observations falling in V_{θ_i}. The update equation for θ_i becomes

$$\theta_i(n+1) = \theta_i(n) + \alpha_n \, \gamma(d_{y_n}, g_i(\Theta_n; N), y_n, \Theta_n) \, \nabla_{\theta_i(n)} \rho(\theta_i(n), y_n). \tag{3.38}$$

[1]In practice, the frequency of re-calculation would be determined by the problem and would probably not be done at every step.

This equation has the same form as (3.4) with the function $\overline{H}(\Theta, z)$ defined from (3.38) replacing $H(\Theta, z)$. Let $\bar{h}(\Theta; N)$ be the function for the associated ODE. In the case of a finite number of observations, it follows that

$$\begin{aligned}
\bar{h}_i(\Theta; N) &= E_z[\overline{H}_i(\Theta, z)] && (3.39)\\
&= -\bar{\gamma}_i(\Theta; N)\frac{1}{N}\sum_{j=1}^{N} \nabla_{\theta_i}\rho(\theta_i, y_j) 1_{\{y_j \in V_{\theta_i}\}}(1_{\{d_{y_j}=2\}} - 1_{\{d_{y_j}=1\}}) && (3.40)\\
&= \bar{\gamma}_i(\Theta; N)\,(1_{\{d_{\theta_i}=2\}} - 1_{\{d_{\theta_i}=1\}})\, h_i(\Theta; N) && (3.41)
\end{aligned}$$

where

$$\bar{\gamma}_i(\Theta; N) = \text{sign}\left\{\frac{1}{N}\sum_{j=1}^{N} 1_{\{y_j \in V_{\theta_i}\}}(1_{\{d_{y_j}=2\}} - 1_{\{d_{y_j}=1\}})\right\} \qquad (3.42)$$

and $h_i(\Theta; N)$ is as defined in (3.34). Therefore we see that the equilibrium points of $h_i(\Theta; N)$ are the same as the equilibrium points of $\bar{h}(\Theta; N)$. Showing that the majority vote modification results in a larger number of stable equilibrium points is a hard problem and more work needs to be done to support this claim.

In the case of an infinite number of observations, we can give a heuristic argument that supports this claim. Notice that from (SLLN) as the number of observations goes to infinity, $\bar{h}(\Theta; N)$ converges with probability one to $\bar{h}(\Theta)$ given by

$$\bar{h}_i(\Theta) = -\text{sign}\left\{\int_{V_{\theta_i}} q(x)\,dx\right\} \int_{V_{\theta_i}} \nabla_{\theta_i}\rho(\theta_i, x)\, q(x)\,dx \qquad (3.43)$$

with $q(x) = p_2(x)\pi_2 - p_1(x)\pi_1$. If the size of each Voronoi cell is small then by the mean value theorem $\bar{h}_i(\Theta)$ is approximately equal to

$$\hat{h}_i(\Theta) = -\int_{V_{\theta_i}} \nabla_{\theta_i}\rho(\theta_i, x)\,|q(x)|\,dx. \qquad (3.44)$$

The right-hand side of the last equation is minus the (i^{th} component of) gradient of the cost function

$$J(\Theta) = \sum_{i=1}^{k} \int_{V_{\theta_i}} \rho(\theta_i, x)\,|q(x)|\,dx. \qquad (3.45)$$

Therefore, from Lyapunov stability it follows that all of the equilibria are stable.

3.4 Generalization to Several Patterns

The convergence results above are true in the case of several pattern densities with the appropriate modification to the notation and some additional assumptions.

Suppose there are ℓ patterns then

$$q(x,\theta_i) = p_{d_{\theta_i}}(x)\pi_{d_{\theta_i}} - \sum_{\substack{j=1 \\ j\neq d_{\theta_i}}}^{\ell} p_j(x)\,\pi_j \tag{3.46}$$

where $p_{d_{\theta_i}}(x)$ is the pattern density associated with the decision of θ_i and $\pi_{d_{\theta_i}}$ its prior probability of occurrence. The functions $h_i(\Theta)$ resulting from equation (3.11) are given by

$$h_i(\Theta) = -\int_{V_{\theta_i}} \nabla_{\theta_i}\rho(\theta_i,x)\,q(x,\theta_i)\,dx \quad i=1,\ldots,k. \tag{3.47}$$

In order for the decision regions to make sense their decisions must agree with the majority vote of the observations falling in their Voronoi cells. For the binary case discussed above, this was enforced via the requirement that

$$\int_{V_{\theta_i}} q(x)\,dx < 0 \quad \text{for } i \leq k_0 \tag{3.48}$$

$$\int_{V_{\theta_i}} q(x)\,dx > 0 \quad \text{for } i > k_0 \tag{3.49}$$

Two requirements are necessary for the decision regions in the case of several patterns. The first requirement is that the decision of each cell must be the majority vote of the observations falling in that cell. More precisely,

$$d_{\theta_i} = \arg\max_{j=1,\ldots,\ell}\{\int_{V_{\theta_i}} p_j(x)\,\pi_j\,dx\} \tag{3.50}$$

where $p_j(x)$ is the pattern density for pattern j and π_j its prior probability of occurrence. The second requirement is that for each Voronoi cell

$$\int_{V_{\theta_i}} q(x,\theta_i)\,dx > 0 \qquad i=1,\ldots,k. \tag{3.51}$$

This requirement can be explained by noting that for region V_{θ_i} the probability of a correct decision is equal to

$$P_c(V_{\theta_i}) = \int_{V_{\theta_i}} p_{d_{\theta_i}}(x)\,\pi_{d_{\theta_i}}\,dx \tag{3.52}$$

and the probability of error is equal to

$$P_e(V_{\theta_i}) = \int_{V_{\theta_i}} \sum_{\substack{j=1 \\ j\neq d_{\theta_i}}}^{\ell} p_j(x)\,\pi_j\,dx. \tag{3.53}$$

Hence this requirement (expressed by equation (3.51)) is nothing more than the requirement that the probability of correct decision be greater than the probability of error for each region.

3.5 Decision Error

In this section we discuss the error associated with the modified LVQ algorithm. Here two results are shown. The first is the simple comparison between LVQ and the nearest neighbor algorithm. The second result shows that if the number of Voronoi vectors is allowed to go to infinity at an appropriate rate as the number of observations goes to infinity, then it is possible to construct a consistent estimator for every risk discussed in Chapter 1. That is, the error associated with LVQ can be made to approach the optimal error. As before, we concentrate on the binary pattern case for ease of notation. The multiple pattern case can be handled with the modifications discussed above.

3.5.1 Nearest Neighbor

If a Voronoi vector is assigned to each observation then the LVQ algorithm reduces to the nearest neighbor algorithm. For that algorithm, it was shown (Cover & Hart [1967]) that its Bayes minimum probability of error is less than twice the that of the optimal classifier. More specifically, let r^* be the Bayes optimal risk and let r be the nearest neighbor risk. It was shown that

$$r^* \leq r \leq 2r^*(1 - r^*) \leq 2r^*. \tag{3.54}$$

Hence in the case of no iteration, the Bayes' risk associated with LVQ is given from the nearest neighbor algorithm.

3.5.2 Other Choices for the Number of Voronoi Vectors

We saw above that if the number of Voronoi vectors equals the number of observations then LVQ coincides with the nearest neighbor algorithm. Let k_N represent the number of Voronoi vectors for an observation sample size of N. We are interested in determining the probability of error for LVQ when k_N satisfies (1) $\lim k_N = \infty$

and (2) $\lim(k_N/N) = 0$. In this case, there are more observations than vectors and hence the Voronoi vectors represent averages of the observations.

Letting the number of Voronoi vectors go to infinity with the number of observations presents a problem of interpretation for the LVQ algorithm. To see what we mean, suppose that $k_N = \lfloor \sqrt{N} \rfloor$, then every time N is a perfect square, k is incremented by one. When k is incremented the iteration (3.7) stops, a new Voronoi vector is added, and the decisions associated with all of the Voronoi vectors are recalculated. Unfortunately, it is not clear how to choose the location of the added Voronoi vector. Furthermore, if the number of Voronoi vectors is large and if the Voronoi vectors are initialized according to a uniform partition of the observation space, then the LVQ algorithm does not move the vectors far from their initial values. As a result, the error associated with initial conditions starts to dominate the overall classification error. In view of these facts, we now consider the effects of the initial conditions on the classification error and examine the algorithm without learning iterations for large k_N.

Let $\Theta_N = \{\theta_1, \ldots, \theta_{k_N}\}$ and assume that the Voronoi vectors are initialized so that

$$\text{Vol}(V_{\theta_i}) = O(\frac{1}{k_N}). \tag{3.55}$$

Here we assume that the pattern densities have compact support. Let $y \in V_{\theta_i}$ and suppose that

$$\hat{q}(y; N) = \frac{1}{N} \sum_{j=1}^{N} Y_j \tag{3.56}$$

with

$$Y_j = \frac{1_{\{y_j \in V_{\theta_i}\}}(1_{\{d_{y_j}=2\}} - 1_{\{d_{y_j}=1\}})}{\text{Vol}(V_{\theta_i})}. \tag{3.57}$$

Then an argument similar to that in Theorem 1.4.3 shows that $\hat{q}(y; N)$ is a weakly consistent estimator of $q(y)$. Therefore the decision associated with θ_i converges in probability to the optimal decision, i.e., if $q(\theta_i) \geq 0$ then θ_i is assigned decision class 2 and otherwise θ_i is assigned decision class 1.

3.6 Initialization

As with many locally converging adaptive schemes, the initialization of the parameters in LVQ is crucial to the ultimate success of the detector. The initialization for this algorithm involves picking the number of Voronoi vectors and their locations. The decisions for the Voronoi vectors are given by the majority vote algorithm.

3.6.1 The Number of Vectors

In the original presentation of LVQ, Kohonen postulated that in order to preserve the underlying probabilistic structure, the relative number of Voronoi vectors for each pattern should be related to the prior probabilities of occurrence. While this conjecture seems plausible, it need not be true. Consider the example presented at the beginning of this chapter. In that example both patterns were equally likely, however, twice as many Voronoi vectors were needed for pattern 2 as were needed for pattern 1. It seems that the number of Voronoi vectors for each pattern should be chosen as a function of each pattern variance. This observation was also made in (Kangas et al. [1989]).

More work needs to be done to state exactly how the number of Voronoi vectors should relate to the pattern densities, but we note that if the total number of Voronoi vectors is large and if the initial decisions are chosen by majority vote ,then the relative number of Voronoi vectors assigned to each pattern is related to the pattern variances and the priors. Therefore, at least indirectly, the modified algorithm already accounts for pattern variance.

At present, picking the number of Voronoi vectors is somewhat arbitrary. A good rule of thumb is to pick about $\sqrt{N}$ vectors where N is the number of past observations used in training. This number is in keeping with other nonparametric methods (Rao [1983]).

3.6.2 The Initial Locations

There are several methods for initializing the locations of the Voronoi vectors. We will discuss (1) selecting the locations uniformly in the pattern space; (2) choosing

the locations from the past observations; and (3) calculating the locations using vector quantization on the past observations.

Selecting the locations uniformly in the pattern space is desirable when the number of Voronoi vectors is large, or equivalently, when the resulting Voronoi cells are small. In this initialization method, the majority vote algorithm closely approximates the optimal decision regions due to the fact that the integral over the Voronoi cell is estimated by the integrand using the Mean Value Theorem.

Choosing the locations based upon the past observations was first proposed in (Kohonen [1986]). This method has a drawback in that the observations chosen as initial conditions may not be representative of their patterns. In addition, since the locations of observations are probabilistic, it is possible that large regions in the pattern space could be represented by one Voronoi vector. Therefore, this method should only be used when the observations used as initial locations for the Voronoi vectors are representative of the whole observation set.

Calculating the locations using vector quantization is the best method to use when the number of Voronoi vectors is small in comparison to the number of observations and/or the dimension of the observations. This method was proposed in (Kangas et al. [1989]). Let $z_n = (y_n, d_{y_n})$ be an observation. This method involves performing vector quantization on the data set $Y = \{y_n\}$. Once the optimal quantization vectors are found, they are used as Voronoi vectors with their decisions determined by the majority vote of the observations contained in their Voronoi cells. This method results in initial vectors whose locations are representative of the whole observation set.

3.7 Application to Other Risks

In this section we show how to modify LVQ in order to be able to handle the risks discussed in Chapter 1. To account for other risks, one modifies the number of observations in each pattern so that the risk corresponds to the Bayes risk for minimum probability of error of the modified problem. To see this, note that each risk in Chapter 1 had its regions defined by $S_2 = \{x : p_1(x) - tp_2(x) > 0\}$ for the appropriate choice of t. Therefore, we find $\tilde{\pi}_1$ and $\tilde{\pi}_2$ such that $t = \tilde{\pi}_2/\tilde{\pi}_1$ and

then adjust the number of observations so that N_1/N is close to $\tilde{\pi}_1$, N_2/N is close to $\tilde{\pi}_2$ and both converge as the number of observations go to infinity. Here, N_1 (resp. N_2) are the number of observations from pattern 1 (resp. 2).

3.8 Remarks

In this chapter, it was shown that the adaptation rule of LVQ is a stochastic approximation algorithm and under appropriate conditions on the adaptation parameter, the pattern densities, and the initial conditions, that the Voronoi vectors converge to the stable equilibria of an associated ODE. We presented a modification to the Kohonen algorithm argued that it results in convergence for a wider class of initial conditions. We showed that LVQ is a general histogram classifier and that its risk converges to the optimal risk as the appropriate parameters went to infinity with the number of past observations. Finally, we discussed several methods for initializing the Voronoi vectors.

Chapter 4

Simulations

In this chapter we use computer simulations to demonstrate several of the properties of LVQ. The simulations are used to compared LVQ to two other classification techniques, namely, adaptive histogram and second order parametric classification. Two sets of twelve examples were simulated and the results tabulated in the sections to follow. The first set of examples was concerned with the detection between two different Gaussian patterns. The simulation set was taken from (Chi & Van Ryzin [1977]) where it was used to compare the adaptive histogram method to the second order parametric one. The second set of simulations dealt with the discrimination between Rayleigh distributed and lognormal distributed patterns. This simulation set demonstrated the superiority of LVQ over second order parametric classification. In all the simulations, the performance of the optimal detector was displayed in order to compare the adaptive methods to the best possible performance. The optimal detector always performed better than the adaptive classifiers because the optimal classifier has complete statistical information whereas the other classifiers have to estimate their statistical knowledge from the observation set.

Within each set of simulations, the parameters of the LVQ algorithm were varied in order to determine their effects on the overall classification performance. In particular, the size of the observation set, the number of Voronoi vectors, the number of iterations through the LVQ algorithm, and the adaptation rate α_1, were all varied. The initial locations of the Voronoi vectors were fixed in all examples.

The second set of simulations was carried out in order to compare LVQ against

second order parametric classification when both patterns were not Gaussian. We felt that it was important to see how LVQ performed against second order parametric when the pattern models were non-Gaussian. In particular, we wanted to see how the second order parametric classifier would perform when the pattern means or the pattern variances were the same.

As was mentioned before, the power of nonparametric detectors lies in their independence of an assumed model. This is particularly important since assuming a model and identifying its parameters will result in suboptimal performance when the data comes from another model. This was demonstrated clearly by the second simulation set.

This chapter is organized as follows: In Section 4.1, we describe the overall simulations performed. In Sections 4.2–4.3, we describe precisely each example in the simulation sets. In Section 4.4, we analyze the results of the simulation and in Section 4.5 we present some concluding remarks.

4.1 Simulation Setup

In this section we describe how the simulations were carried out. We are concerned with comparing LVQ to the adaptive histogram method of VanRyzin and to second order parametric classification. Second order parametric classification consists in assuming a unimodal Gaussian model for both pattern densities and then calculating the sample means and variances from the observation set. The detection regions are then found by using the corresponding Gaussian densities in the Bayes minimum probability of risk.

The adaptive histogram method of (Chi & Van Ryzin [1977]) was discussed previously in Section 1.3.1. Recall that the adaptive histogram classification consists in ordering the unlabeled observation data and then constructing bins which contained a fixed number of observations. After the bin locations are determined, their decisions are calculated by a majority vote of the observations falling in each bin. The number of observations in each bin is approximately equal to $\lfloor N^{0.55} \rfloor$ where N is the number of observations (Chi & Van Ryzin [1977]).

For each of the twenty-four cases, 100 independent simulations were run. In

each simulation, the same observation data was used as input to each of the classification methods. After the simulations were complete, the averages and the standard deviations for each case were calculated and recorded in the tables. Hence, each table entry corresponds to the average of 100 independent Monte Carlo simulations with the standard deviation calculated in order to help determine the significance of the differences in the mean entries. The distance function, $\rho(\theta, x)$, used for the LVQ method was Euclidean distance.

For each simulation set, two different types of tables were generated. The first table type has the number of LVQ iterations fixed at 10 and the adaptation rate fixed at 0.1, i.e., the past observation set was presented to the adaptation algorithm 10 times and $\alpha_1 = 0.1$. We let $\alpha_n = \alpha_1/\sqrt{n}$ where n represented the number of passes through the entire observation data set. Entries in the table correspond to varying the number of Voronoi Vectors. The number of Voronoi vectors was $\{3, 5, 7\}$ and they were initialized to $[-2, 0, 2]$, $[-2, -1, 0, 1, 2]$ or $[-3, -2, -1, 0, 1, 2, 3]$, respectively[1]. Four tables were generated with total observation data sizes of 20, 50, 100 and 200[2].

In the second table type, the size of the observation set is fixed at 100 and the number of Voronoi Vectors is fixed at 5. Entries in the tables correspond to three different values for α_1 chosen from $\{0.05, 0.10, 0.25\}$ with α_n defined above. Three tables were generated with 10, 20, and 40 complete presentations of the observation data.

Thus, for each simulation set seven tables were generated. For the Gaussian simulation set, the best classifier among the **nonparametric** classifiers is highlighted. For the non-Gaussian simulation set the best classifier among **all** classifiers is highlighted. These tables are given in Section 4.6. In the next two sections, we describe the parameters of each simulation set and discuss the results of varying the parameters of LVQ.

[1]Ideally, a separate calculation would be performed to determine the initial conditions.

[2]Note that when the total observation data size is 20 that means that if the *a priori* probability of pattern 1 is 0.25 then five independent samples of pattern 1 and fifteen independent samples of pattern 2 are used to train the classifiers.

4.2 The Gaussian Examples

The first simulation set uses Gaussian distributed patterns. The first eleven are unimodal densities and the twelfth is bimodal (see Table 4.1). The examples along with a graph of the pattern densities are displayed in Table 4.1. We use this simulation set for comparison against the results presented in (Chi & Van Ryzin [1977]). Since the first eleven cases are unimodal Gaussian densities, we expect that the second order parametric classifier will outperform the other nonparametric detectors. However, the twelfth case is bimodal therefore we expect that the second order parametric classifier will fail, and in this case, fail miserably. These conjectures are borne out in the simulations presented in Tables 4.7–4.13. The twelfth case serves to illustrate the power of nonparametric classification as opposed to second order parametric classification since this case is not handled by a simple mean or variance type test.

4.3 Rayleigh vs. Lognormal Examples

In the second simulation set, the patterns were Rayleigh and lognormal distributed. The expressions for these densities, along with those of their means and variances, are displayed in Table 4.2.

This simulation set was constructed to compare LVQ to the second order parametric classifier when both patterns were non-Gaussian. The examples were constructed in a manner that, we felt, would "confuse" the parametric classifier, e.g., cases 3–8 have the same mean and cases 9–10 have the same variance.

The twelve cases in the non-Gaussian simulation set are given in Table 4.3. This table gives the parameters of both pattern models and a small graph of both densities. These graphs can be used to determine the optimal decision regions using the techniques discussed in Chapter 1.

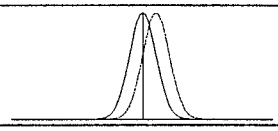
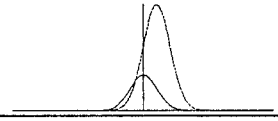
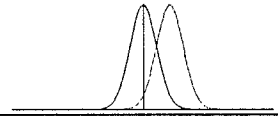
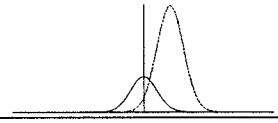
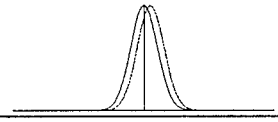
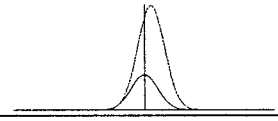
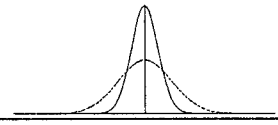
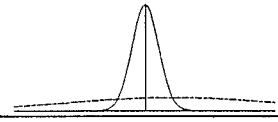
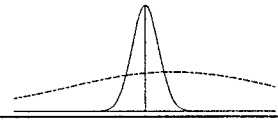
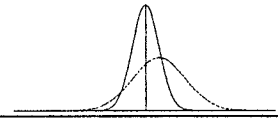
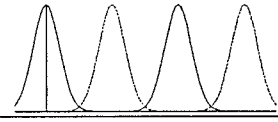
Example Number	Gaussian Pattern 1	Gaussian Pattern 2	Priors		Plot of Both Densities
			π_1	π_2	
1	N(0,1)	N(1,1)	$\frac{1}{2}$	$\frac{1}{2}$	
2	N(0,1)	N(1,1)	$\frac{1}{4}$	$\frac{3}{4}$	
3	N(0,1)	N(2,1)	$\frac{1}{2}$	$\frac{1}{2}$	
4	N(0,1)	N(2,1)	$\frac{1}{4}$	$\frac{3}{4}$	
5	N(0,1)	N(.5,1)	$\frac{1}{2}$	$\frac{1}{2}$	
6	N(0,1)	N(.5,1)	$\frac{1}{4}$	$\frac{3}{4}$	
7	N(0,1)	N(0,4)	$\frac{1}{2}$	$\frac{1}{2}$	
8	N(0,1)	N(0,4)	$\frac{1}{4}$	$\frac{3}{4}$	
9	N(0,1)	N(2,64)	$\frac{1}{2}$	$\frac{1}{2}$	
10	N(0,1)	N(2,64)	$\frac{1}{4}$	$\frac{3}{4}$	
11	N(0,1)	N(1,4)	$\frac{1}{2}$	$\frac{1}{2}$	
12	0.5N(0,1) +0.5N(10,1)	0.5N(5,1) +0.5N(15,1)	$\frac{1}{2}$	$\frac{1}{2}$	

Table 4.1: *Specifications of the Gaussian simulation set*

Density	Mean	Variance
Rayleigh $p_R(x) = \frac{x}{\sigma_R^2} e^{-\frac{x^2}{2\sigma_R^2}}$	$\sigma_R\sqrt{\frac{\pi}{2}}$	$\sigma_R^2(2-\pi/2)$
Lognormal $p_L(x) = \frac{1}{x\sqrt{2\pi}\sigma_L} e^{-\frac{(\log(y))^2}{2\sigma_L}}$	$e^{\frac{\sigma_L}{2}}$	$e^{2\sigma_L} - e^{\sigma_L}$

Table 4.2: *Rayleigh and lognormal densities and their properties*

4.4 Analysis of the Results

In this section we analyze the results of the simulations. We discuss the effects of varying the number of Voronoi vectors, the iteration number, the learning rate, or the observation size. In addition, we discuss the effects of initial conditions on convergence and the overall performance of LVQ in relation to the other methods.

4.4.1 Number of Voronoi Vectors

From the simulations we see that increasing the number of Voronoi vectors does not always result in better detection. This can be seen in case 9 of Table 4.14 where the probability of detection is 0.8559 when three Voronoi vectors are used while for the same problem is 0.8396 when seven Voronoi vectors are used. This phenomena occurs because of the relationship between the number of Voronoi vectors and the number of observations. In Table 4.14, when there are 7 Voronoi vectors and only 20 observations, there is not enough data per Voronoi vector. Even in the ideal situation there can only be 3 observations per vector, as a result there is poor averaging. However, as the number of observations is increased to 200 we see that LVQ with 7 Voronoi vectors does better in most cases than LVQ with 3 or 5. This time, comparing the results of case 9 in Table 4.17 we see that the probability of detection for three Voronoi vectors is 0.8598 while for the same problem is 0.8663 for seven vectors.

It is interesting to compare the number of parameters calculated for each of the nonparametric classifiers. The adaptive histogram classifier has about $\sqrt{N}$ bins

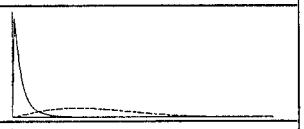
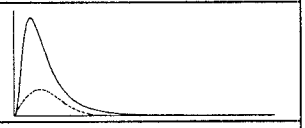
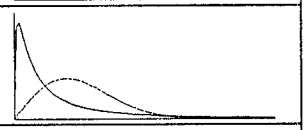
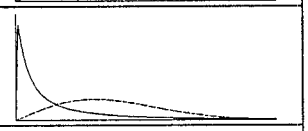
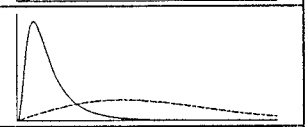
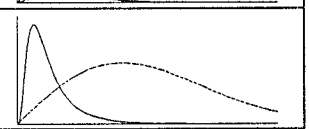
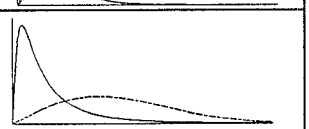
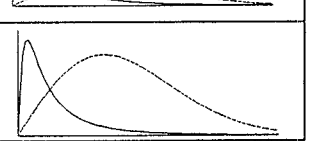
Example Number	Rayleigh σ_R	Lognormal σ_L	Priors π_R	Priors π_L	Plot of Both Densities
1	10	0.25	$\frac{1}{2}$	$\frac{1}{2}$	
2	10	0.25	$\frac{1}{4}$	$\frac{3}{4}$	
3	1	0.4516	$\frac{1}{2}$	$\frac{1}{2}$	
4	1	0.4516	$\frac{1}{4}$	$\frac{3}{4}$	
5	2	1.8379	$\frac{1}{2}$	$\frac{1}{2}$	
6	2	1.8379	$\frac{1}{4}$	$\frac{3}{4}$	
7	3	2.6488	$\frac{1}{2}$	$\frac{1}{2}$	
8	3	2.6488	$\frac{1}{4}$	$\frac{3}{4}$	
9	4	0.4	$\frac{1}{2}$	$\frac{1}{2}$	
10	4	0.4516	$\frac{3}{4}$	$\frac{1}{4}$	
11	3.2989	1	$\frac{1}{2}$	$\frac{1}{2}$	
12	3.2989	1	$\frac{3}{4}$	$\frac{1}{4}$	

Table 4.3: *Specifications of the non-Gaussian simulation set*

Number of Observations	Number of Bins
20	4
50	6
100	8
200	11

Table 4.4: *Number of parameters in the adaptive histogram method*

where N is the number of samples (see Table 4.4) while LVQ has 3, 5 or 7. In many simulations LVQ performed well with only 3 Voronoi vectors.

When the variance of one or both of the patterns is large then more Voronoi vectors are needed to represent the pattern classes. This can be clearly seen in cases 9 and 10 of the Gaussian simulation set. In these cases, pattern 2 had a variance of 64. More specifically, in case 9 of Table 4.7 the probability of detection was 0.7895 for three Voronoi vectors and was 0.8212 for the same problem using seven Voronoi vectors. This improvement arises because of the variance of pattern 2. The larger observation values contained in the observation set pull the Voronoi cells away from the optimal decision regions; increasing the number of Voronoi vectors alleviates this problem. Note that if one of the pattern variances is high, then it is unlikely that an LVQ classifier would be used since a simple threshold classifier would perform quite well.

4.4.2 Number of Iterations

From the simulations, we see that increasing the number of iterations does not always result in better classification. For example, a general comparison between Table 4.18 and Table 4.20 shows no significant difference between classification errors. In particular, the differences between case 6 in Table 4.18 and Table 4.20 are within experimental error. This is consistent with our experience in generating the simulations. However, it was somewhat unexpected. Before conducting the simulations, we were expecting the best classification to occur when $\alpha_1 = .25$ and the number of iterations was 40, because in the beginning of the LVQ iterations the high adaptation rate would tend to move the Voronoi vectors to the correct

Gaussian Simulation set		
Table number	**LVQ wins**	**AH wins**
4.7	12	0
4.8	11	1
4.9	12	0
4.10	10	2
4.11	9	3
4.12	9	3
4.13	9	3
Total	**72**	**12**

Table 4.5: *Performance of LVQ vs adaptive histogram*

regions and as α_n became smaller their locations would be fine tuned. After all, when n=20 we see that $\alpha_{20} = .05$ which is one of the entries in the tables.

Our only explanation for this behavior comes by noting the high standard deviation of the simulation results. This is most likely due to a high noise content in the generated observation data set. As a result, we feel that the data contained enough noise to overcome the initial conditions and hence a small number of iterations performed well.

4.4.3 Size of the Learning Rate

The arguments above apply equally well to the learning rate. Our simulations show no significant difference between the classification errors when the learning rate was varied from $\{0.05, 0.1, 0.25\}$. However, we did find that there is a certain threshold which the learning rate must exceed in order to have a meaningful classifier.

4.4.4 Overall Performance

In Tables 4.5–4.6 we have tabulated the overall classification results which are presented in Tables 4.7–4.20. While in most cases there was a clear winner, it should be noted that some of the decisions were very close. In fact, most of the decisions were within experimental error of each other. These simulations

Non-Gaussian Simulation set			
Table number	LVQ wins	AH wins	SOP wins
4.14	10	2	0
4.15	10	0	2
4.16	9	0	3
4.17	8	1	3
4.18	7	2	3
4.19	6	3	3
4.20	7	2	3
Total	**57**	**10**	**17**

Table 4.6: *Performance of LVQ vs adaptive histogram and second order parametric*

show that LVQ offers a competitive alternative to adaptive histogram classification and for non-Gaussian patterns, a superior alternative to second order parametric classification.

In the Gaussian simulation set, the second order parametric generally outperformed both nonparametric classification techniques. This was expected since the data was Gaussian and hence characterized by its mean and variance. However, case 12 illustrated the problem with second order parametric classification when the data is bimodal. In that case, both nonparametric classifiers did significantly better than the second order parametric classifier.

4.4.5 Sensitivity to Initial Conditions

During the simulations we ran into a problem with sensitivity to initial conditions. Recall from Chapter 3 that convergence in the LVQ algorithm is a local property. Therefore, it is always possible for the vectors to settle in on a local minimum. This phenomena occurred in case 12 of the Gaussian simulation set. Originally all of the Voronoi vectors in the Gaussian simulation set were initialized the same. However, we noticed that the performance for case 12 was not as good as expected. Upon further investigation, we discovered that the LVQ algorithm had settled on a local minimum. To understand this phenomena, let's consider case 12 with 7 Voronoi

vectors initialized at [-3 -2 -1 0 1 2 3]. The pattern densities in case 12 were two bimodal Gaussian densities. The first pattern was distributed $\frac{1}{2}N(0,1)+\frac{1}{2}N(10,1)$ and the second was distributed $\frac{1}{2}N(5,1)+\frac{1}{2}N(15,1)$. Since none of the initial conditions were greater than 5, the Voronoi vectors could not account for the two probability masses at 10 and 15. This happened because the vector near 5 was given decision 2 and hence was repelled by the mass located at 10. Likewise, the mass at 15 was too weak to pull that Voronoi vector over the mass at 10. Therefore, the whole interval $[4.5, \infty)$ was represented by one Voronoi vector. This resulted in an unacceptably high error rate. To prevent this from happening, we adjusted the initial conditions for case 12. The vectors were initialized to [0, 2, 4, 6, 8, 10, 12, 14].

In an application of LVQ to real data, the Voronoi vectors would be initialized after analyzing the observation data, hence this problem would be avoided. It is interesting to note that in the scalar case, a simple uniform partition of the observation space performs well. It is certainly the case that for vector observations, the initialization process must be carefully done using one of the methods discussed in Chapter 3.

4.5 Remarks

These simulations have demonstrated that (1) LVQ provides good detection performance when the size of the observation set is small; (2) only a small number of iterations through the LVQ algorithm is needed in order to obtain convergence; (3) increasing the number of Voronoi vectors leads to better performance; (4) that LVQ is relatively insensitive to the value of the adaptation parameter α and (5) it compares favorably with other parametric and nonparametric classification schemes.

The number of Voronoi vectors and their locations can be determined experimentally. This can be accomplished by picking an initial value for the number of Voronoi vectors k, completing the LVQ training, evaluating the resulting classifier with new observation data, incrementing k and then repeating the whole process. When the estimated error reaches a minimum, the resulting value of k can be used.

In Chapter 3, we showed convergence to the optimal Bayesian cost as the number of observations goes to infinity. The simulations in this chapter show that even with a small number of observations the resulting detector performs quite well. More analytical work needs to be done to investigate these phenomena.

4.6 Results of Simulations

Gaussian Example 20 observation points		Bayesian Optimal	Parametric Model	Adaptive Histogram	LVQ 3 points	LVQ 5 points	LVQ 7 points
Case 1	**mean**	**0.6915**	**0.6625**	**0.6365**	**0.6571**	**0.6383**	**0.6386**
	std		0.0568	0.0691	0.0564	0.0662	0.0674
Case 2	**mean**	**0.7775**	**0.7574**	**0.7217**	**0.7640**	**0.7340**	**0.7293**
	std		0.0244	0.0600	0.0346	0.0607	0.0634
Case 3	**mean**	**0.8413**	**0.8319**	**0.8213**	**0.8286**	**0.8045**	**0.7934**
	std		0.0130	0.0374	0.0247	0.0354	0.0344
Case 4	**mean**	**0.8730**	**0.8566**	**0.8443**	**0.8352**	**0.8558**	**0.8479**
	std		0.0334	0.0391	0.0261	0.0390	0.0199
Case 5	**mean**	**0.5987**	**0.5557**	**0.5352**	**0.5376**	**0.5405**	**0.5389**
	std		0.0548	0.0524	0.0574	0.0516	0.0515
Case 6	**mean**	**0.7510**	**0.7333**	**0.7005**	**0.7326**	**0.6934**	**0.6946**
	std		0.0295	0.0660	0.0325	0.0572	0.0569
Case 7	**mean**	**0.6613**	**0.6345**	**0.5640**	**0.6020**	**0.5817**	**0.5637**
	std		0.0365	0.0603	0.0518	0.0528	0.0544
Case 8	**mean**	**0.7500**	**0.7290**	**0.6954**	**0.7102**	**0.6619**	**0.6355**
	std		0.0342	0.0651	0.0722	0.0555	0.0487
Case 9	**mean**	**0.8818**	**0.8675**	**0.7507**	**0.7895**	**0.7880**	**0.8212**
	std		0.0191	0.0797	0.0657	0.0524	0.0613
Case 10	**mean**	**0.8587**	**0.8419**	**0.7720**	**0.7284**	**0.7643**	**0.7852**
	std		0.0161	0.0837	0.0491	0.0625	0.0642
Case 11	**mean**	**0.6950**	**0.6722**	**0.6023**	**0.6478**	**0.6260**	**0.6068**
	std		0.0199	0.0695	0.0495	0.0540	0.0533
Case 12	**mean**	**0.9907**	**0.5099**	**0.8727**	**0.7043**	**0.9766**	**0.9573**
	std		0.0094	0.0673	0.0387	0.0099	0.0282

Table 4.7: *Gaussian simulation set with 20 observations*

Gaussian Example 50 observation points		Bayesian Optimal	Parametric Model	Adaptive Histogram	LVQ 3 points	LVQ 5 points	LVQ 7 points
Case 1	**mean**	**0.6915**	**0.6862**	**0.6527**	**0.6771**	**0.6709**	**0.6692**
	std		0.0108	0.0390	0.0172	0.0350	0.0382
Case 2	**mean**	**0.7775**	**0.7700**	**0.7387**	**0.7713**	**0.7631**	**0.7595**
	std		0.0090	0.0405	0.0089	0.0256	0.0277
Case 3	**mean**	**0.8413**	**0.8390**	**0.8196**	**0.8333**	**0.8236**	**0.8169**
	std		0.0033	0.0336	0.0097	0.0271	0.0261
Case 4	**mean**	**0.8730**	**0.8690**	**0.8475**	**0.8451**	**0.8552**	**0.8497**
	std		0.0059	0.0298	0.0234	0.0171	0.0234
Case 5	**mean**	**0.5987**	**0.5813**	**0.5511**	**0.5637**	**0.5622**	**0.5631**
	std		0.0318	0.0360	0.0377	0.0405	0.0388
Case 6	**mean**	**0.7510**	**0.7450**	**0.7172**	**0.7410**	**0.7348**	**0.7363**
	std		0.0102	0.0346	0.0117	0.0281	0.0271
Case 7	**mean**	**0.6613**	**0.6512**	**0.6031**	**0.6307**	**0.6122**	**0.6020**
	std		0.0103	0.0481	0.0348	0.0388	0.0394
Case 8	**mean**	**0.7500**	**0.7456**	**0.6983**	**0.7480**	**0.7263**	**0.7099**
	std		0.0133	0.0403	0.0200	0.0410	0.0409
Case 9	**mean**	**0.8818**	**0.8761**	**0.8316**	**0.7859**	**0.7892**	**0.8611**
	std		0.0063	0.0350	0.0362	0.0286	0.0315
Case 10	**mean**	**0.8587**	**0.8536**	**0.8197**	**0.7366**	**0.7978**	**0.8110**
	std		0.0047	0.0307	0.0293	0.0450	0.0266
Case 11	**mean**	**0.6950**	**0.6855**	**0.6481**	**0.6617**	**0.6523**	**0.6465**
	std		0.0104	0.0380	0.0204	0.0425	0.0427
Case 12	**mean**	**0.9907**	**0.5053**	**0.7858**	**0.6746**	**0.9820**	**0.9808**
	std		0.0032	0.0471	0.0357	0.0100	0.0086

Table 4.8: *Gaussian simulation set with 50 observations*

Gaussian Example 100 observation points		Bayesian Optimal	Parametric Model	Adaptive Histogram	LVQ		
					3 points	5 points	7 points
Case 1	**mean**	**0.6915**	**0.6888**	**0.6608**	**0.6808**	**0.6805**	**0.6787**
	std		0.0062	0.0291	0.0155	0.0208	0.0254
Case 2	**mean**	**0.7775**	**0.7727**	**0.7584**	**0.7719**	**0.7680**	**0.7655**
	std		0.0068	0.0222	0.0078	0.0139	0.0155
Case 3	**mean**	**0.8413**	**0.8399**	**0.8257**	**0.8342**	**0.8334**	**0.8293**
	std		0.0019	0.0218	0.0083	0.0172	0.0185
Case 4	**mean**	**0.8730**	**0.8714**	**0.8597**	**0.8571**	**0.8613**	**0.8609**
	std		0.0021	0.0195	0.0130	0.0122	0.0129
Case 5	**mean**	**0.5987**	**0.5892**	**0.5575**	**0.5727**	**0.5693**	**0.5698**
	std		0.0254	0.0301	0.0330	0.0362	0.0351
Case 6	**mean**	**0.7510**	**0.7482**	**0.7250**	**0.7419**	**0.7434**	**0.7441**
	std		0.0046	0.0280	0.0145	0.0168	0.0154
Case 7	**mean**	**0.6613**	**0.6562**	**0.6364**	**0.6485**	**0.6356**	**0.6323**
	std		0.0057	0.0257	0.0230	0.0157	0.0219
Case 8	**mean**	**0.7500**	**0.7494**	**0.7157**	**0.7500**	**0.7446**	**0.7385**
	std		0.0049	0.0249	0.0000	0.0199	0.0207
Case 9	**mean**	**0.8818**	**0.8792**	**0.8508**	**0.7716**	**0.7775**	**0.8508**
	std		0.0034	0.0230	0.0283	0.0261	0.0203
Case 10	**mean**	**0.8587**	**0.8561**	**0.8367**	**0.7436**	**0.7996**	**0.8195**
	std		0.0028	0.0203	0.0203	0.0289	0.0254
Case 11	**mean**	**0.6950**	**0.6914**	**0.6592**	**0.6676**	**0.6682**	**0.6704**
	std		0.0037	0.0238	0.0167	0.0246	0.0194
Case 12	**mean**	**0.9907**	**0.5036**	**0.9407**	**0.6648**	**0.9862**	**0.9856**
	std		0.0007	0.0206	0.0322	0.0075	0.0047

Table 4.9: *Gaussian simulation set with 100 observations*

Gaussian Example 200 observation points		Bayesian Optimal	Parametric Model	Adaptive Histogram	LVQ 3 points	LVQ 5 points	LVQ 7 points
Case 1	**mean**	**0.6915**	**0.6906**	**0.6762**	**0.6870**	**0.6889**	**0.6879**
	std		0.0012	0.0159	0.0052	0.0057	0.0027
Case 2	**mean**	**0.7775**	**0.7752**	**0.7651**	**0.7720**	**0.7705**	**0.7692**
	std		0.0039	0.0178	0.0074	0.0068	0.0071
Case 3	**mean**	**0.8413**	**0.8408**	**0.8325**	**0.8341**	**0.8393**	**0.8387**
	std		0.0007	0.0136	0.0060	0.0029	0.0039
Case 4	**mean**	**0.8730**	**0.8722**	**0.8658**	**0.8581**	**0.8654**	**0.8665**
	std		0.0010	0.0076	0.0107	0.0072	0.0089
Case 5	**mean**	**0.5987**	**0.5957**	**0.5684**	**0.5858**	**0.5828**	**0.5824**
	std		0.0062	0.0219	0.0166	0.0189	0.0190
Case 6	**mean**	**0.7510**	**0.7495**	**0.7355**	**0.7454**	**0.7484**	**0.7472**
	std		0.0023	0.0171	0.0079	0.0069	0.0095
Case 7	**mean**	**0.6613**	**0.6589**	**0.6390**	**0.6554**	**0.6411**	**0.6441**
	std		0.0030	0.0182	0.0094	0.0146	0.0139
Case 8	**mean**	**0.7500**	**0.7500**	**0.7286**	**0.7500**	**0.7500**	**0.7478**
	std		0.0000	0.0195	0.0000	0.0000	0.0108
Case 9	**mean**	**0.8818**	**0.8808**	**0.8659**	**0.7737**	**0.7749**	**0.8424**
	std		0.0008	0.0114	0.0181	0.0209	0.0148
Case 10	**mean**	**0.8587**	**0.8575**	**0.8452**	**0.7492**	**0.7871**	**0.8194**
	std		0.0016	0.0119	0.0076	0.0231	0.0171
Case 11	**mean**	**0.6950**	**0.5000**	**0.6696**	**0.6679**	**0.6789**	**0.6799**
	std		0.0000	0.0185	0.0145	0.0147	0.0107
Case 12	**mean**	**0.9907**	**0.5035**	**0.9024**	**0.6594**	**0.9889**	**0.9870**
	std		0.0006	0.0178	0.0314	0.0024	0.0021

Table 4.10: *Gaussian simulation set with 200 observations*

Gaussian Example		Bayesian	Parametric	Adaptive	LVQ		
10 Iterations of LVQ		Optimal	Model	Histogram	0.05	0.10	0.25
Case 1	**mean**	**0.6915**	**0.6895**	**0.6653**	**0.6768**	**0.6780**	**0.6780**
	std		0.0028	0.0281	0.0357	0.0306	0.0295
Case 2	**mean**	**0.7775**	**0.7722**	**0.7557**	**0.7705**	**0.7691**	**0.7681**
	std		0.0079	0.0267	0.0083	0.0128	0.0120
Case 3	**mean**	**0.8413**	**0.8404**	**0.8298**	**0.8355**	**0.8361**	**0.8357**
	std		0.0013	0.0167	0.0130	0.0123	0.0134
Case 4	**mean**	**0.8730**	**0.8716**	**0.8615**	**0.8607**	**0.8625**	**0.8631**
	std		0.0017	0.0195	0.0106	0.0090	0.0102
Case 5	**mean**	**0.5987**	**0.5881**	**0.5602**	**0.5700**	**0.5701**	**0.5742**
	std		0.0269	0.0255	0.0285	0.0314	0.0312
Case 6	**mean**	**0.7510**	**0.7483**	**0.7294**	**0.7445**	**0.7459**	**0.7454**
	std		0.0053	0.0239	0.0174	0.0121	0.0120
Case 7	**mean**	**0.6613**	**0.6568**	**0.6327**	**0.6261**	**0.6292**	**0.6282**
	std		0.0060	0.0299	0.0204	0.0193	0.0241
Case 8	**mean**	**0.7500**	**0.7497**	**0.7151**	**0.7441**	**0.7465**	**0.7463**
	std		0.0027	0.0285	0.0219	0.0177	0.0186
Case 9	**mean**	**0.8818**	**0.8798**	**0.8490**	**0.7834**	**0.7826**	**0.7947**
	std		0.0020	0.0211	0.0205	0.0227	0.0288
Case 10	**mean**	**0.8587**	**0.8563**	**0.8383**	**0.8077**	**0.7908**	**0.7779**
	std		0.0027	0.0179	0.0405	0.0354	0.0278
Case 11	**mean**	**0.6950**	**0.6908**	**0.6586**	**0.6689**	**0.6733**	**0.6745**
	std		0.0040	0.0259	0.0214	0.0190	0.0193
Case 12	**mean**	**0.9907**	**0.5035**	**0.9374**	**0.9871**	**0.9870**	**0.9867**
	std		0.0005	0.0179	0.0017	0.0017	0.0025

Table 4.11: *Gaussian simulation set with 10 iterations of LVQ*

Gaussian Example 20 Iterations of LVQ		Bayesian Optimal	Parametric Model	Adaptive Histogram	LVQ		
					0.05	0.10	0.25
Case 1	**mean**	**0.6915**	**0.6894**	**0.6669**	**0.6832**	**0.6826**	**0.6810**
	std		0.0048	0.0255	0.0159	0.0141	0.0191
Case 2	**mean**	**0.7775**	**0.7726**	**0.7587**	**0.7675**	**0.7666**	**0.7668**
	std		0.0074	0.0227	0.0160	0.0173	0.0149
Case 3	**mean**	**0.8413**	**0.8401**	**0.8283**	**0.8361**	**0.8367**	**0.8358**
	std		0.0021	0.0160	0.0113	0.0097	0.0099
Case 4	**mean**	**0.8730**	**0.8716**	**0.8638**	**0.8605**	**0.8619**	**0.8627**
	std		0.0020	0.0152	0.0137	0.0124	0.0116
Case 5	**mean**	**0.5987**	**0.5909**	**0.5587**	**0.5721**	**0.5723**	**0.5722**
	std		0.0164	0.0297	0.0279	0.0294	0.0329
Case 6	**mean**	**0.7510**	**0.7481**	**0.7231**	**0.7422**	**0.7444**	**0.7423**
	std		0.0057	0.0276	0.0181	0.0142	0.0151
Case 7	**mean**	**0.6613**	**0.6564**	**0.6339**	**0.6303**	**0.6320**	**0.6348**
	std		0.0101	0.0278	0.0194	0.0205	0.0217
Case 8	**mean**	**0.7500**	**0.7489**	**0.7169**	**0.7428**	**0.7447**	**0.7455**
	std		0.0061	0.0253	0.0237	0.0196	0.0199
Case 9	**mean**	**0.8818**	**0.8796**	**0.8526**	**0.7802**	**0.7792**	**0.7883**
	std		0.0020	0.0262	0.0223	0.0252	0.0318
Case 10	**mean**	**0.8587**	**0.8557**	**0.8342**	**0.8014**	**0.7928**	**0.7767**
	std		0.0029	0.0243	0.0338	0.0318	0.0269
Case 11	**mean**	**0.6950**	**0.6912**	**0.6592**	**0.6706**	**0.6735**	**0.6741**
	std		0.0046	0.0250	0.0230	0.0217	0.0206
Case 12	**mean**	**0.9907**	**0.5035**	**0.9384**	**0.9872**	**0.9869**	**0.9867**
	std		0.0006	0.0153	0.0017	0.0018	0.0017

Table 4.12: *Gaussian simulation set with 20 iterations of LVQ*

Gaussian Example 40 Iterations of LVQ		Bayesian Optimal	Parametric Model	Adaptive Histogram	LVQ		
					0.05	0.10	0.25
Case 1	**mean**	**0.6915**	**0.6895**	**0.6659**	**0.6823**	**0.6821**	**0.6800**
	std		0.0041	0.0304	0.0212	0.0222	0.0224
Case 2	**mean**	**0.7775**	**0.7719**	**0.7588**	**0.7679**	**0.7670**	**0.7671**
	std		0.0078	0.0232	0.0144	0.0155	0.0152
Case 3	**mean**	**0.8413**	**0.8398**	**0.8257**	**0.8322**	**0.8332**	**0.8331**
	std		0.0022	0.0225	0.0187	0.0173	0.0167
Case 4	**mean**	**0.8730**	**0.8713**	**0.8585**	**0.8618**	**0.8614**	**0.8631**
	std		0.0021	0.0188	0.0109	0.0123	0.0104
Case 5	**mean**	**0.5987**	**0.5913**	**0.5618**	**0.5768**	**0.5766**	**0.5781**
	std		0.0143	0.0266	0.0236	0.0239	0.0247
Case 6	**mean**	**0.7510**	**0.7474**	**0.7179**	**0.7409**	**0.7424**	**0.7404**
	std		0.0056	0.0325	0.0198	0.0180	0.0183
Case 7	**mean**	**0.6613**	**0.6573**	**0.6346**	**0.6312**	**0.6308**	**0.6314**
	std		0.0051	0.0271	0.0190	0.0239	0.0244
Case 8	**mean**	**0.7500**	**0.7494**	**0.7170**	**0.7417**	**0.7441**	**0.7460**
	std		0.0045	0.0249	0.0272	0.0240	0.0201
Case 9	**mean**	**0.8818**	**0.8790**	**0.8491**	**0.7785**	**0.7798**	**0.7906**
	std		0.0038	0.0237	0.0224	0.0233	0.0280
Case 10	**mean**	**0.8587**	**0.7500**	**0.8382**	**0.7937**	**0.7838**	**0.7722**
	std		0.0000	0.0163	0.0304	0.0312	0.0301
Case 11	**mean**	**0.6950**	**0.6908**	**0.6605**	**0.6684**	**0.6705**	**0.6719**
	std		0.0051	0.0250	0.0291	0.0247	0.0237
Case 12	**mean**	**0.9907**	**0.5035**	**0.9346**	**0.9870**	**0.9869**	**0.9870**
	std		0.0006	0.0163	0.0016	0.0016	0.0015

Table 4.13: *Gaussian simulation set with 40 iterations of LVQ*

Rayleigh / Lognormal 20 observation points		Bayesian Optimal	Parametric Model	Adaptive Histogram	LVQ 3 points	LVQ 5 points	LVQ 7 points
Case 1	**mean**	**0.9711**	**0.9608**	**0.9374**	**0.9667**	**0.9548**	**0.9559**
	std		0.0162	0.0357	0.0110	0.0281	0.0280
Case 2	**mean**	**0.9806**	**0.9692**	**0.9509**	**0.9734**	**0.9614**	**0.9629**
	std		0.0168	0.0340	0.0185	0.0319	0.0298
Case 3	**mean**	**0.5810**	**0.5141**	**0.5229**	**0.5050**	**0.5142**	**0.5130**
	std		0.0320	0.0329	0.0136	0.0437	0.0452
Case 4	**mean**	**0.7512**	**0.7158**	**0.6894**	**0.7341**	**0.7017**	**0.6984**
	std		0.0563	0.0587	0.0174	0.0405	0.0424
Case 5	**mean**	**0.6944**	**0.6470**	**0.6636**	**0.6134**	**0.6826**	**0.6760**
	std		0.0747	0.0630	0.0764	0.0723	0.0642
Case 6	**mean**	**0.7585**	**0.7039**	**0.7142**	**0.7221**	**0.7057**	**0.6877**
	std		0.0865	0.0304	0.0266	0.0357	0.0281
Case 7	**mean**	**0.7793**	**0.6756**	**0.7089**	**0.7162**	**0.7176**	**0.7251**
	std		0.0848	0.0516	0.0525	0.0535	0.0550
Case 8	**mean**	**0.7848**	**0.7183**	**0.7280**	**0.7526**	**0.7178**	**0.7158**
	std		0.0845	0.0376	0.0309	0.0463	0.0450
Case 9	**mean**	**0.8701**	**0.6648**	**0.8488**	**0.8559**	**0.8370**	**0.8396**
	std		0.1648	0.0414	0.0100	0.0593	0.0571
Case 10	**mean**	**0.8820**	**0.7670**	**0.8553**	**0.7707**	**0.8561**	**0.8523**
	std		0.0417	0.0433	0.0302	0.0351	0.0411
Case 11	**mean**	**0.7940**	**0.6690**	**0.7577**	**0.7671**	**0.7578**	**0.7584**
	std		0.1289	0.0538	0.0331	0.0491	0.0480
Case 12	**mean**	**0.8451**	**0.7532**	**0.8026**	**0.7452**	**0.8081**	**0.8085**
	std		0.0609	0.0454	0.0213	0.0546	0.0545

Table 4.14: *Non-Gaussian simulation set with 20 observations*

Rayleigh / Lognormal 50 observation points		Bayesian Optimal	Parametric Model	Adaptive Histogram	LVQ 3 points	LVQ 5 points	LVQ 7 points
Case 1	mean	0.9711	0.9669	0.9582	0.9554	0.9633	0.9642
	std		0.0064	0.0141	0.0075	0.0115	0.0038
Case 2	mean	0.9806	0.9742	0.9412	0.9792	0.9694	0.9707
	std		0.0087	0.0240	0.0058	0.0077	0.0099
Case 3	mean	0.5810	0.5231	0.5280	0.5073	0.5336	0.5329
	std		0.0289	0.0293	0.0129	0.0325	0.0317
Case 4	mean	0.7512	0.7412	0.7116	0.7416	0.7283	0.7280
	std		0.0206	0.0390	0.0140	0.0264	0.0285
Case 5	mean	0.6944	0.6511	0.6926	0.6697	0.7106	0.7039
	std		0.0684	0.0373	0.0678	0.0277	0.0275
Case 6	mean	0.7585	0.7123	0.7231	0.7392	0.7345	0.7187
	std		0.0932	0.0303	0.0158	0.0252	0.0244
Case 7	mean	0.7793	0.6779	0.7420	0.7314	0.7211	0.7462
	std		0.0936	0.0368	0.0401	0.0557	0.0319
Case 8	mean	0.7848	0.7212	0.7465	0.7562	0.7509	0.7584
	std		0.0781	0.0258	0.0181	0.0276	0.0265
Case 9	mean	0.8701	0.6981	0.8529	0.8625	0.8570	0.8590
	std		0.1601	0.0333	0.0061	0.0247	0.0225
Case 10	mean	0.8820	0.7533	0.8591	0.7937	0.8707	0.8704
	std		0.0173	0.0216	0.0458	0.0135	0.0133
Case 11	mean	0.7940	0.7208	0.7713	0.7854	0.7743	0.7722
	std		0.1049	0.0297	0.0114	0.0211	0.0209
Case 12	mean	0.8451	0.7650	0.8271	0.7510	0.8371	0.8364
	std		0.0629	0.0252	0.0135	0.0257	0.0230

Table 4.15: *Non-Gaussian simulation set with 50 observations*

Rayleigh / Lognormal 100 observation points		Bayesian Optimal	Parametric Model	Adaptive Histogram	LVQ 3 points	LVQ 5 points	LVQ 7 points
Case 1	**mean**	**0.9711**	**0.9686**	**0.9586**	**0.9386**	**0.9684**	**0.9646**
	std		0.0033	0.0118	0.0084	0.0105	0.0038
Case 2	**mean**	**0.9806**	**0.9768**	**0.9559**	**0.9763**	**0.9663**	**0.9746**
	std		0.0033	0.0159	0.0029	0.0062	0.0075
Case 3	**mean**	**0.5810**	**0.5259**	**0.5323**	**0.5075**	**0.5417**	**0.5419**
	std		0.0287	0.0264	0.0133	0.0325	0.0323
Case 4	**mean**	**0.7512**	**0.7464**	**0.7208**	**0.7456**	**0.7399**	**0.7380**
	std		0.0138	0.0326	0.0104	0.0173	0.0195
Case 5	**mean**	**0.6944**	**0.6377**	**0.7099**	**0.6984**	**0.7201**	**0.7210**
	std		0.0735	0.0215	0.0442	0.0164	0.0161
Case 6	**mean**	**0.7585**	**0.7255**	**0.7357**	**0.7427**	**0.7440**	**0.7378**
	std		0.0628	0.0239	0.0155	0.0186	0.0175
Case 7	**mean**	**0.7793**	**0.6590**	**0.7567**	**0.7311**	**0.7100**	**0.7626**
	std		0.0950	0.0207	0.0650	0.0825	0.0140
Case 8	**mean**	**0.7848**	**0.7154**	**0.7566**	**0.7528**	**0.7530**	**0.7623**
	std		0.0953	0.0245	0.0187	0.0259	0.0226
Case 9	**mean**	**0.8701**	**0.6986**	**0.8601**	**0.8605**	**0.8653**	**0.8656**
	std		0.1560	0.0144	0.0074	0.0077	0.0100
Case 10	**mean**	**0.8820**	**0.7530**	**0.8740**	**0.7998**	**0.8745**	**0.8749**
	std		0.0163	0.0185	0.0489	0.0143	0.0117
Case 11	**mean**	**0.7940**	**0.7505**	**0.7781**	**0.7874**	**0.7788**	**0.7794**
	std		0.0782	0.0231	0.0060	0.0158	0.0107
Case 12	**mean**	**0.8451**	**0.7765**	**0.8334**	**0.7500**	**0.8433**	**0.8431**
	std		0.0397	0.0160	0.0000	0.0024	0.0025

Table 4.16: *Non-Gaussian simulation set with 100 observations*

Rayleigh / Lognormal		Bayesian	Parametric	Adaptive	LVQ		
200 observation points		Optimal	Model	Histogram	3 points	5 points	7 points
Case 1	**mean**	**0.9711**	**0.9699**	**0.9513**	**0.9295**	**0.9587**	**0.9650**
	std		0.0015	0.0127	0.0095	0.0098	0.0052
Case 2	**mean**	**0.9806**	**0.9771**	**0.9572**	**0.9676**	**0.9656**	**0.9768**
	std		0.0028	0.0107	0.0055	0.0082	0.0061
Case 3	**mean**	**0.5810**	**0.5249**	**0.5370**	**0.5095**	**0.5538**	**0.5546**
	std		0.0236	0.0183	0.0070	0.0230	0.0200
Case 4	**mean**	**0.7512**	**0.7500**	**0.7379**	**0.7476**	**0.7469**	**0.7465**
	std		0.0000	0.0179	0.0037	0.0057	0.0055
Case 5	**mean**	**0.6944**	**0.6237**	**0.7105**	**0.6536**	**0.7180**	**0.7300**
	std		0.0653	0.0174	0.0869	0.0108	0.0087
Case 6	**mean**	**0.7585**	**0.7227**	**0.7460**	**0.7456**	**0.7478**	**0.7462**
	std		0.0483	0.0111	0.0125	0.0130	0.0136
Case 7	**mean**	**0.7793**	**0.6553**	**0.7531**	**0.7478**	**0.7029**	**0.7677**
	std		0.0879	0.0146	0.0277	0.0925	0.0123
Case 8	**mean**	**0.7848**	**0.7125**	**0.7715**	**0.7481**	**0.7566**	**0.7692**
	std		0.1048	0.0122	0.0121	0.0162	0.0110
Case 9	**mean**	**0.8701**	**0.6988**	**0.8632**	**0.8598**	**0.8639**	**0.8663**
	std		0.1487	0.0108	0.0067	0.0089	0.0091
Case 10	**mean**	**0.8820**	**0.7503**	**0.8756**	**0.7947**	**0.8782**	**0.8780**
	std		0.0027	0.0088	0.0499	0.0087	0.0085
Case 11	**mean**	**0.7940**	**0.7599**	**0.7859**	**0.7891**	**0.7839**	**0.7839**
	std		0.0643	0.0133	0.0043	0.0134	0.0136
Case 12	**mean**	**0.8451**	**0.7830**	**0.8396**	**0.7500**	**0.8421**	**0.8422**
	std		0.0386	0.0100	0.0000	0.0032	0.0033

Table 4.17: *Non-Gaussian simulation set with 200 observations*

Rayleigh / Lognormal 10 Iterations of LVQ		Bayesian Optimal	Parametric Model	Adaptive Histogram	LVQ		
					0.05	0.10	0.25
Case 1	**mean**	**0.9711**	**0.9683**	**0.9559**	**0.9668**	**0.9599**	**0.9470**
	std		0.0036	0.0140	0.0134	0.0104	0.0173
Case 2	**mean**	**0.9806**	**0.9759**	**0.9567**	**0.9666**	**0.9664**	**0.9672**
	std		0.0044	0.0183	0.0048	0.0092	0.0104
Case 3	**mean**	**0.5810**	**0.5261**	**0.5362**	**0.5455**	**0.5425**	**0.5434**
	std		0.0287	0.0218	0.0301	0.0306	0.0292
Case 4	**mean**	**0.7512**	**0.7449**	**0.7248**	**0.7388**	**0.7393**	**0.7379**
	std		0.0255	0.0310	0.0155	0.0144	0.0161
Case 5	**mean**	**0.6944**	**0.6453**	**0.7056**	**0.7154**	**0.7131**	**0.7014**
	std		0.0640	0.0271	0.0278	0.0307	0.0513
Case 6	**mean**	**0.7585**	**0.7231**	**0.7338**	**0.7413**	**0.7404**	**0.7388**
	std		0.0601	0.0259	0.0213	0.0196	0.0184
Case 7	**mean**	**0.7793**	**0.6624**	**0.7555**	**0.7161**	**0.7048**	**0.6909**
	std		0.0887	0.0212	0.0837	0.0950	0.1015
Case 8	**mean**	**0.7848**	**0.7182**	**0.7566**	**0.7537**	**0.7465**	**0.7528**
	std		0.0925	0.0226	0.0212	0.0279	0.0154
Case 9	**mean**	**0.8701**	**0.6876**	**0.8570**	**0.8636**	**0.8623**	**0.8609**
	std		0.1532	0.0185	0.0128	0.0104	0.0107
Case 10	**mean**	**0.8820**	**0.7535**	**0.8751**	**0.8760**	**0.8770**	**0.8754**
	std		0.0180	0.0123	0.0121	0.0109	0.0101
Case 11	**mean**	**0.7940**	**0.7346**	**0.7795**	**0.7783**	**0.7794**	**0.7800**
	std		0.0937	0.0203	0.0169	0.0177	0.0319
Case 12	**mean**	**0.8451**	**0.7703**	**0.8324**	**0.8411**	**0.8387**	**0.8378**
	std		0.0530	0.0165	0.0134	0.0165	0.0142

Table 4.18: *Non-Gaussian simulation set with 10 iterations of LVQ*

Rayleigh / Lognormal 20 Iterations of LVQ		Bayesian Optimal	Parametric Model	Adaptive Histogram	LVQ		
					0.05	0.10	0.25
Case 1	**mean**	**0.9711**	**0.9685**	**0.9561**	**0.9658**	**0.9543**	**0.9446**
	std		0.0034	0.0139	0.0088	0.0113	0.0164
Case 2	**mean**	**0.9806**	**0.9759**	**0.9567**	**0.9655**	**0.9683**	**0.9670**
	std		0.0044	0.0183	0.0065	0.0106	0.0102
Case 3	**mean**	**0.5810**	**0.5261**	**0.5362**	**0.5452**	**0.5434**	**0.5427**
	std		0.0287	0.0218	0.0277	0.0303	0.0299
Case 4	**mean**	**0.7512**	**0.7449**	**0.7248**	**0.7387**	**0.7386**	**0.7384**
	std		0.0255	0.0310	0.0157	0.0158	0.0155
Case 5	**mean**	**0.6944**	**0.6570**	**0.7064**	**0.7197**	**0.7167**	**0.7088**
	std		0.0620	0.0231	0.0152	0.0152	0.0352
Case 6	**mean**	**0.7585**	**0.7266**	**0.7402**	**0.7418**	**0.7381**	**0.7404**
	std		0.0574	0.0195	0.0201	0.0213	0.0175
Case 7	**mean**	**0.7793**	**0.6659**	**0.7543**	**0.6970**	**0.6920**	**0.6774**
	std		0.0908	0.0201	0.0970	0.1010	0.1092
Case 8	**mean**	**0.7848**	**0.7244**	**0.7566**	**0.7511**	**0.7494**	**0.7499**
	std		0.0972	0.0219	0.0240	0.0214	0.0189
Case 9	**mean**	**0.8701**	**0.7026**	**0.8572**	**0.8638**	**0.8624**	**0.8601**
	std		0.1529	0.0156	0.0100	0.0102	0.0128
Case 10	**mean**	**0.8820**	**0.7563**	**0.8756**	**0.8733**	**0.8726**	**0.8722**
	std		0.0256	0.0119	0.0156	0.0157	0.0131
Case 11	**mean**	**0.7940**	**0.7430**	**0.7779**	**0.7793**	**0.7815**	**0.7826**
	std		0.0867	0.0226	0.0130	0.0122	0.0113
Case 12	**mean**	**0.8451**	**0.7831**	**0.8304**	**0.8389**	**0.8374**	**0.8370**
	std		0.0407	0.0171	0.0238	0.0241	0.0171

Table 4.19: *Non-Gaussian simulation set with 20 iterations of LVQ*

Rayleigh / Lognormal 40 Iterations of LVQ		Bayesian Optimal	Parametric Model	Adaptive Histogram	LVQ		
					0.05	0.10	0.25
Case 1	**mean**	**0.9711**	**0.9686**	**0.9569**	**0.9594**	**0.9531**	**0.9479**
	std		0.0031	0.0134	0.0087	0.0104	0.0150
Case 2	**mean**	**0.9806**	**0.9768**	**0.9567**	**0.9680**	**0.9673**	**0.9649**
	std		0.0036	0.0177	0.0086	0.0108	0.0126
Case 3	**mean**	**0.5810**	**0.5276**	**0.5329**	**0.5412**	**0.5413**	**0.5405**
	std		0.0274	0.0206	0.0321	0.0312	0.0293
Case 4	**mean**	**0.7512**	**0.7479**	**0.7223**	**0.7396**	**0.7402**	**0.7406**
	std		0.0103	0.0318	0.0183	0.0154	0.0168
Case 5	**mean**	**0.6944**	**0.6559**	**0.7067**	**0.7195**	**0.7144**	**0.7069**
	std		0.0628	0.0221	0.0135	0.0185	0.0404
Case 6	**mean**	**0.7585**	**0.7224**	**0.7368**	**0.7381**	**0.7392**	**0.7418**
	std		0.0681	0.0204	0.0226	0.0207	0.0197
Case 7	**mean**	**0.7793**	**0.6666**	**0.7586**	**0.7088**	**0.6998**	**0.6867**
	std		0.0874	0.0201	0.0835	0.0902	0.1036
Case 8	**mean**	**0.7848**	**0.7314**	**0.7568**	**0.7515**	**0.7517**	**0.7530**
	std		0.0735	0.0234	0.0217	0.0201	0.0185
Case 9	**mean**	**0.8701**	**0.6504**	**0.8607**	**0.8636**	**0.8625**	**0.8600**
	std		0.1603	0.0116	0.0125	0.0112	0.0126
Case 10	**mean**	**0.8820**	**0.7517**	**0.8723**	**0.8721**	**0.8749**	**0.8742**
	std		0.0103	0.0180	0.0165	0.0126	0.0108
Case 11	**mean**	**0.7940**	**0.7407**	**0.7771**	**0.7804**	**0.7817**	**0.7835**
	std		0.0902	0.0219	0.0102	0.0135	0.0126
Case 12	**mean**	**0.8451**	**0.7767**	**0.8341**	**0.8426**	**0.8412**	**0.8399**
	std		0.0419	0.0120	0.0080	0.0075	0.0065

Table 4.20: *Non-Gaussian simulation set with 40 iterations of LVQ*

Chapter 5

Discussion

In this dissertation we studied the properties of Kohonen's LVQ.

We have shown that the adaptation rule of LVQ was a stochastic approximation algorithm and that under the appropriate conditions on the adaptation parameter, the pattern densities and the initial conditions, that the Voronoi vectors converged to stable equilibria of an associated ODE. We presented a modification to the algorithm, which we argued results in convergence for a wider class of initial conditions. We showed that LVQ was a general histogram classifier and that its risk converged to the optimal risk as the appropriate parameters went to infinity with the number of past observations. In addition, we presented several methods for initializing the Voronoi vectors.

Next, we demonstrated through simulations that LVQ performed well compared to parametric and nonparametric classifiers. We showed how the classification error was affected by changing the values of the adaptation rate, the number of Voronoi vectors, the size of the past observation data set, and the number of iterations.

In this chapter we discuss future directions of this work. In Section 5.1 we discuss some preliminary ideas relating to the implementation of LVQ using neural network technology. In Section 5.2, we discuss the use of ergodic observations of the patterns as input to the LVQ algorithm. In Section 5.3, we discuss the use of LVQ to classify two different time series. Finally, in Section 5.4 we discuss some additional issues associated with LVQ which require further investigation.

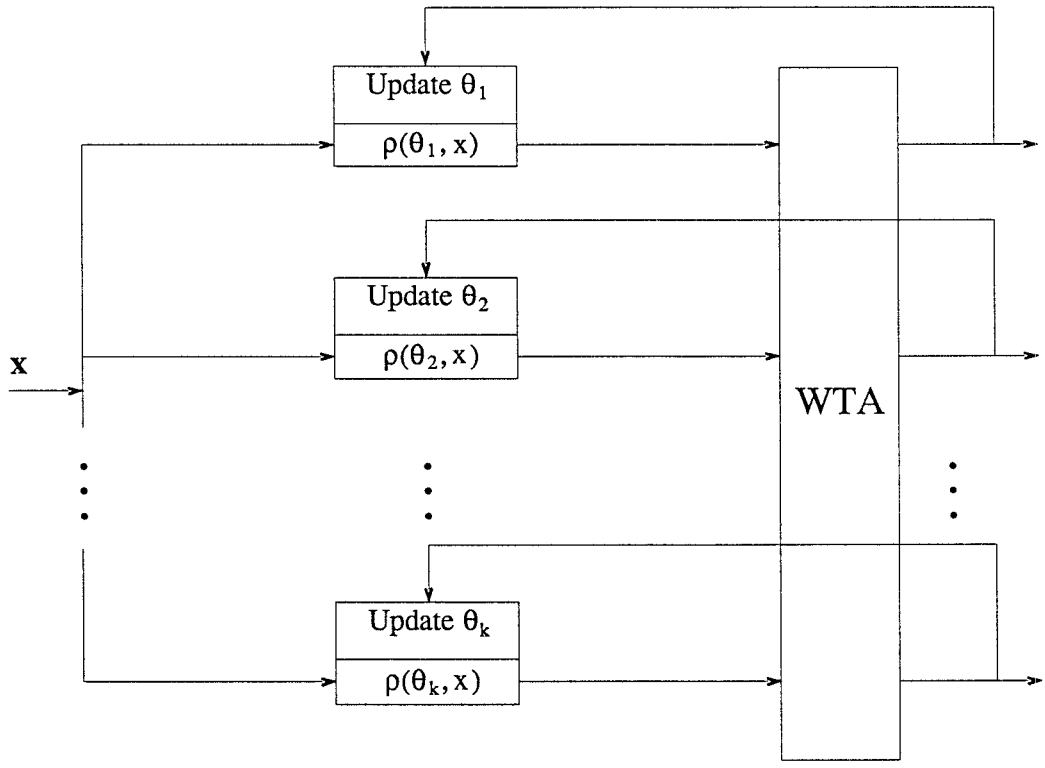


Figure 5.1: *Architecture for implementing LVQ*

5.1 Implementation

The LVQ algorithm originated in Kohonen's work on self-organizing systems so it is only appropriate that it can be implemented using neural network technology. The algorithm consists of a learning and a classification phase. In the learning phase, the Voronoi vectors are adjusted using the past observations. In the classification phase a new observation is classified using the Voronoi vectors. It is the learning phase which is the most computationally intensive since it involves repeatedly taking observations from the observation data set, finding the closest Voronoi vector and updating that vector according to the update rule (see Section 3.1).

Recent work of Carver Meade on analog VLSI has led to the development of an order k winner-take-all network (Lazzaro et al. [1989]). This network computes the maximum among its k inputs. A k winner-take-all network is characterized by the fact that the only nonzero output is the one corresponding to the maximum input. A key feature of this network is its analog implementation and hence com-

putes almost instantaneously. Working chips with 170 input networks have been fabricated (Lazzaro et al. [1989]).

These winner-take-all (WTA) chips can be used in LVQ's learning phase to find the closest Voronoi vector. The input to the network is minus the distance between the observation and the Voronoi vectors plus a bias. The output of the network is used directly in the update rule for LVQ since it indicates whether the observation falls in a particular Voronoi cell.

The implementation of the distance function calculation and the update of the Voronoi vectors need more investigation; however one of the benefits of LVQ is that it only needs local connections and local feedback. This greatly simplifies its implementation as compared to classical neural networks which are massively connected, and suggests that a simple design for the LVQ processor is possible. A block diagram for one such design is depicted in Figure 5.1. During the learning phase, the value of the current observation and its decision are broadcast to all of the processors. Each processor computes the distance between the observation and its Voronoi vector and output minus this value to the winner-take-all network. The output of the winner-take-all network is then fed back to each processor for use in the update equation for the Voronoi vectors. Since the output of the winner-take-all network contains only one nonzero entry, only one of the Voronoi vectors is modified. The learning is continued in this way for several passes through the observation data until the Voronoi vectors converge.

During the classification phase, an observation is broadcast to all of the processors and the closest Voronoi vector is found. The output is the decision of the closest Voronoi vector.

Several questions arise: How should the Voronoi vectors' decisions and locations be stored? Should the update calculations be performed using digital or analog technology? If digital, how many Voronoi vectors should be assigned to each processor? What type of arithmetic should be used? If analog, how should the discrete nature of the update be handled? How can the modified LVQ algorithm be implemented? What is the best way to implement the majority vote correction?

5.2 Ergodic Input

In Chapter 2, we presented the results of Benveniste et. al. (1987) on convergence of the stochastic approximation algorithm. These results provide general convergence theorems which allowed observations from stationary ergodic Markov processes. Therefore, the results in Chapter 3 carry through for these more general observations provided the hypotheses of the convergence theorems are satisfied. In the case of stationary ergodic Markov processes, the invariant measures of the Markov processes play the role of $p_1(x)$ and $p_2(x)$.

5.3 Time Series Data

LVQ can be used to discriminate between two different time series. Suppose that several independent observations of two time series are available. Let $\{x_1(t,n)\}_{n=1}^N$ and $\{x_2(t,n)\}_{n=1}^N$ represent the sets of signals from pattern 1 and pattern 2, respectively. Suppose that the signals are sampled at times $t_0,\ldots,t_m$ and let $X_1(n) = [x_1(t_0,n),\ldots,x_1(t_m,n)]$ and $X_2(n) = [x_2(t_0,n),\ldots,x_2(t_m,n)]$. By training LVQ using $\{X_1(n)\}$ and $\{X_2(n)\}$, the resulting network can perform classification on the new signal $X = [x(t_0),\ldots,x(t_m)]$. This technique allows LVQ to classify time series.

5.4 Further Issues

There are several issues that were raised in Chapter 3 that need further investigation. More work needs to be done to (1) show that the majority vote algorithm does indeed improve the convergence; (2) demonstrate the effects of choosing other distance functions $\rho(\theta,x)$ and investigate whether an optimal one exists; (3) give analytical results which predict the behavior of LVQ when the sample size is small and the number of Voronoi vectors is small; (4) determine the optimal number of Voronoi vectors given an observation set; and (5) determine how the number of Voronoi vectors relates to the pattern variances.

References

A. Benveniste, M. Metivier & P. Priouret [1987], *Algorithmes Adaptatifs et Approximations Stochastiques*, Mason, Paris.

P. Billingsley [1979], *Probability and Measure*, John Wiley & Sons, New York, NY.

T. Cacoullos [1966], "Estimation of a Multivariate Density," *Ann. Inst. Statist. Math* 18, 179–189.

P. Y. Chi & J. Van Ryzin [1977], "A Simple Histogram Method for Nonparametric Classification," in *Classification and Clustering*, J. Van Ryzin, ed., Academic Press, New York, NY, 395–421.

T. M. Cover & P. E. Hart [1967], "Nearest Neighbor Pattern Classification," *IEEE Transactions on Information Theory* IT-13, 21–27.

R. O. Duda & P. E. Hart [1973], *Pattern Classification and Scene Analysis*, John Wiley & Sons, New York, NY.

E. Fix & J. L. Hodges [1951], "Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties," Rep. 4, Project number 21-49-004, USAF School of Aviation Medicine, Randolph Field, Texas.

K. Fukunaga [1972], *Introduction to Statistical Pattern Recognition*, Academic Press, New York, NY.

N. Glick [1972], "Sample-Based Classification Procedures Drevied from Density Estimation," *Journal of the American Statistical Association* 67, 116–122.

R. M. Gray [1984], "Vector Quantization," *IEEE ASSP Magazine* 1, 4–29.

J. Kangas, T. Kohonen, J. Laaksonen, O. Simula & O. Ventä [1989], "Variants of Self-Organizing Maps," *IJCNN International Joint Conference on Neural Networks* II, 517–522.

T. Kohonen [1986], "Learning Vector Quantization for Pattern Recognition," Technical Report TKK-F-A601, Helsinki University of Technology.

N.N. Krasovskii [1963], *Stability of Motion: Applications of Lyapunov's Second Method to Differential Systems and Equations with Delay* (English Translation), Stanford Press, Stanford Calif..

H. J. Kushner & D. S. Clark [1978], *Stochastic Approximation Methods for Constrained and Unconstrained Systems* , Springer-Verlag, New York–Heidelberg–Berlin.

J. Lazzaro, S. Ryckebusch, M. A. Mahowald & C. A. Mead [1989], "Winner-Take-All Network of O(N) Complexity," in *Advances in Neural Information Processing Systems I*, D. S. Touretzky, ed., Morgan Kaufmann Publishers, Inc., San Mateo.

Y. Linde, A. Buzo & R. M. Gray [1980], "An Algorithm for Vector Quantization," *IEEE Transactions on Communication* COM-28, 84–95.

L. Ljung [1977], "Analysis of Recursive Stochastic Algorithms," *IEEE Transactions on Automatic Control* AC-22, 551–575.

D. O. Loftsgaarden & C. P. Quesenberry [1965], "A Nonparametric Estimate of a Multivariate Density Function," *Annals of Mathematical Statistics* 36, 1049–1051.

J.W. Pratt [1960], "On Interchanging Limits and Integrals," *Annals of Mathematical Statistics* 31, 74–77.

B. L. S. Prakasa Rao [1983], *Nonparametric Functional Estimation*, Academic Press, New York, NY.

D. Revuz [1975], *Markov Chains*, North Holland Pub. Co., Amsterdam–New York.

H. Robbins & S. Monro [1951], "A Stochastic Approximation Method," *Annals of Mathematical Statistics* 22, 400–407.

B. W. Silverman [1986], *Density Estimation for Statistics and Data Analysis*, Chapman and Hall Ltd, New York, NY.

J. Van Ryzin [1973], "A Histogram Method of Density Estimation," *Comm. Statist.* 2, 493–506.