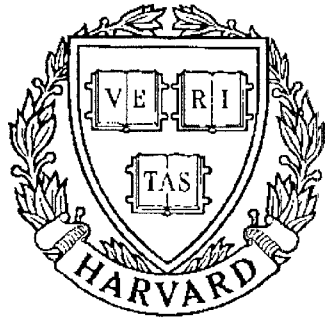
TECHNICAL RESEARCH REPORT

SYSTEMS RESEARCH CENTER



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

Convergence of a Neural Network Classifier

by J.S. Baras and A. LaVigna

TR 90-12

Convergence of a Neural Network Classifier

John S. Baras and Anthony LaVigna
Systems Research Center, University of Maryland
College Park, Maryland 20742

Abstract

Kohonen's Learning Vector Quantization (LVQ) is a neural network architecture that performs nonparametric classification. It classifies observations by comparing them to k templates called Voronoi vectors. The locations of these vectors are determined from past labeled data through a learning algorithm. When learning is complete, the class of a new observation is the same as the class of the closest Voronoi vector. Hence LVQ is similar to nearest neighbors, except that instead of all of the past observations being searched only the k Voronoi vectors are searched.

In this paper, we show that the LVQ learning algorithm converges to locally asymptotic stable equilibria of an ordinary differential equation. We show that the learning algorithm performs stochastic approximation. Convergence of the Voronoi vectors is guaranteed under the appropriate conditions on the underlying statistics of the classification problem. We also present a modification to the learning algorithm which we argue results in convergence of the LVQ error to the Bayesian optimal error as the appropriate parameters become large.

2 Introduction

A common problem in signal processing is the problem of signal classification. An instance of this problem in radar signal processing, is the determination of the presence or absence of a target in the reflected signal. In adaptive control, it is manifested as the problem of determining the operating environment in order to use the appropriate gain in a gain scheduling algorithm. More generally in feedback control, when a precise system model is not known, pattern classifiers play an increasingly important role; see for example recent applications in expert controllers. In all 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, from past observations, in order to build a classifier which identifies each new observation with the lowest possible error. 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 paper 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 a larger set of problems.
- We show by means of simulation results that LVQ has a better overall performance

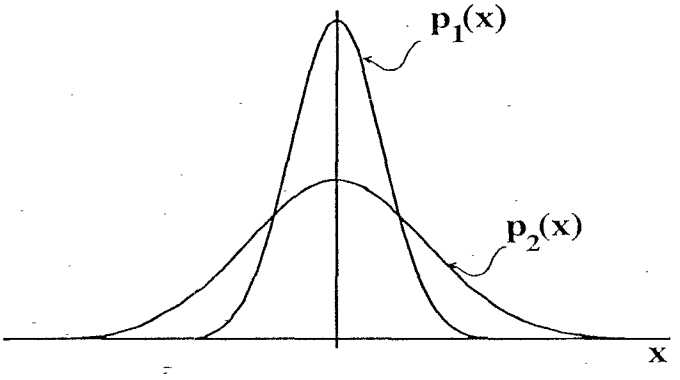


Figure 1: *Plot of two pattern densities*

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 to study LVQ.

In the next sections, the LVQ algorithm is presented. Using theorems from stochastic approximation, 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.

3 Learning Vector Quantization

From the theory of statistical pattern recognition, it is known that the optimal decision regions for a classifier 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 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 is equally likely. These regions are displayed in Figure 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. Given these observations it is possible to construct

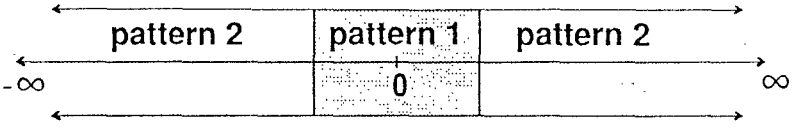


Figure 2: *Plot of decision regions*

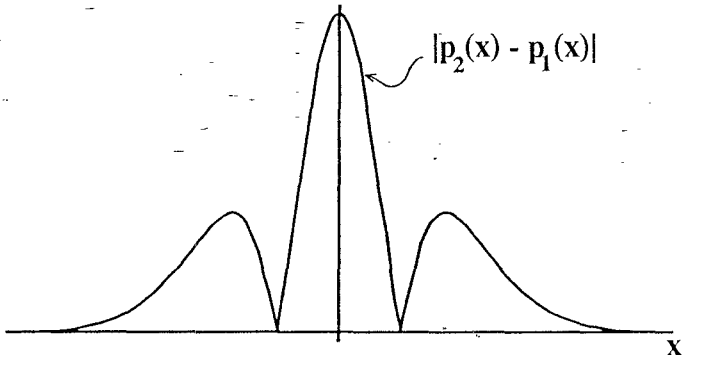


Figure 3: *Absolute value of the difference of the pattern densities*

a consistent nonparametric density estimator and to use these estimators to find approximate decision regions.

For 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. Hence, estimates of both densities will try to minimize the error near zero since that is where most of the observations are located. However, we are only calculating the densities in order to calculate the optimal decision regions therefore, 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. This function is directly related to the optimal decision regions.

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

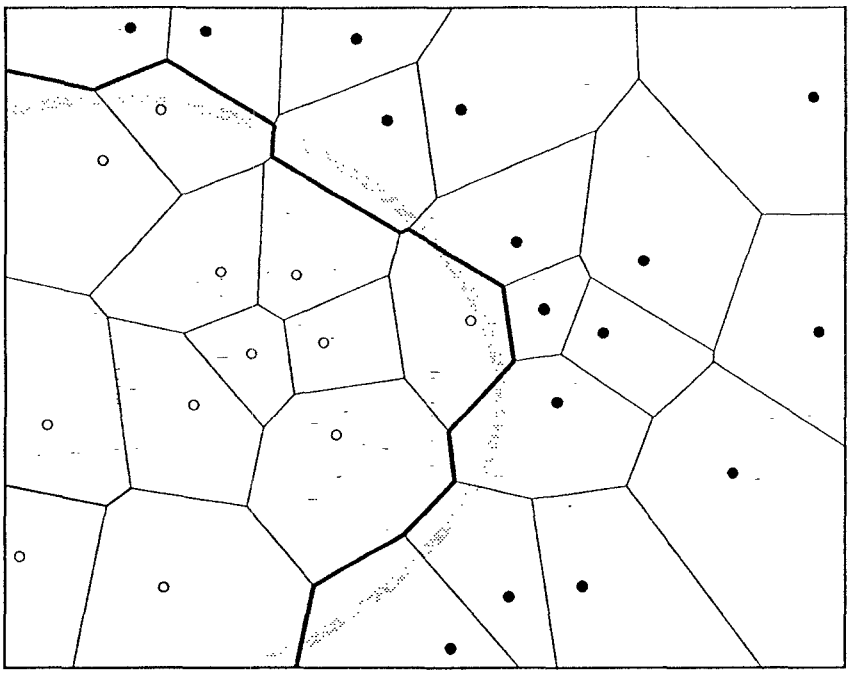


Figure 4: *Voronoi vectors and their approximate decision region.*

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 a two class problem in which some of the Voronoi vectors are of class 1 and others are of class 2 is shown in Figure 4. The shaded region represents the optimal decision boundary and the bold line represents the LVQ approximation to it. 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 detail in Section 9. The vectors are then adjusted by a gradient search type algorithm. Specifically, an 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 (LaVigna [1989]).

4 Description of the Algorithm

Now we describe 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 some reasonable conditions (LaVigna [1989]). Voronoi cells are characterized mathematically by

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

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

The vectors θ_i are adjusted as follows. 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:

$$\theta_c(n+1) = \theta_c(n) - \alpha_n\ \nabla_\theta \rho(\theta_c(n), y_n) \tag{2}$$

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

$$\theta_c(n+1) = \theta_c(n) + \alpha_n\ \nabla_\theta \rho(\theta_c(n), y_n) \tag{3}$$

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 a function $h(\Theta)$ in an associated ODE and studying its properties in order to apply the convergence theorems (Benveniste, Metivier & Priouret [1987]).

5 Convergence to Stationary Points

The stochastic approximation theorems of (Benveniste, Metivier & Priouret [1987]) 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 (8). Given an iterative scheme of the form (2) and (3), 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.

The LVQ algorithm has the general form

$$\theta_i(n+1) = \theta_i(n) + \alpha_n \, \gamma(d_{y_n}, d_{\theta_i(n)}, y_n, \Theta_n) \, \nabla_\theta \rho(\theta_i(n), y_n) \tag{4}$$

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

$$\gamma(d_{y_n}, d_{\theta_i(n)}, y_n, \Theta_n) = \begin{cases} -1_{\{y_n \in V_{\theta_i}\}} & \text{if } d_{y_n} = d_{\theta_i(n)} \\ 1_{\{y_n \in V_{\theta_i}\}} & \text{if } d_{y_n} \neq d_{\theta_i(n)} \end{cases} \tag{5}$$

or, more compactly,

$$\gamma(d_{y_n}, d_{\theta_i(n)}, y_n, \Theta_n) = -1_{\{y_n \in V_{\theta_i}\}}(1_{\{d_{y_n} = d_{\theta_i}\}} - 1_{\{d_{y_n} \neq d_{\theta_i}\}}). \tag{6}$$

This is a stochastic approximation algorithm with $\varrho_n(\Theta, x) \equiv 0$ (see (Benveniste, Metivier & Priouret [1987])). It has the form

$$\Theta_{n+1} = \Theta_n + \alpha_n \; H(\Theta_n, z_n) \tag{7}$$

where Θ is the vector with components θ_i; $H(\Theta, z)$ is the vector with components defined in the obvious manner in (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

$$\frac{d}{dt}\bar{\Theta}(t) = h(\bar{\Theta}(t)) \tag{8}$$

for the appropriate choice of $h(\Theta)$.

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 (Benveniste, Metivier & Priouret [1987]).

5.1 Convergence for an Infinite Number of Observations

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 that $h(\Theta)$ of the associated ODE takes the form

$$h(\Theta) = \begin{pmatrix} h_1(\Theta) \\ \vdots \\ h_{k_0}(\Theta) \\ h_{k_0+1}(\Theta) \\ \vdots \\ h_k(\Theta) \end{pmatrix} = \begin{pmatrix} \int_{V_{\theta_1}} q(x)\ \nabla_{\theta_1}\rho(\theta_1, x)\, dx \\ \vdots \\ \int_{V_{\theta_{k_0}}} q(x)\ \nabla_{\theta_{k_0}}\rho(\theta_{k_0}, x)\, dx \\ -\int_{V_{\theta_{k_0+1}}} q(x)\ \nabla_{\theta_{k_0+1}}\rho(\theta_{k_0+1}, x)\, dx \\ \vdots \\ -\int_{V_{\theta_k}} q(x)\ \nabla_{\theta_k}\rho(\theta_k, x)\, dx \end{pmatrix} \tag{9}$$

with $q(x) = p_2(x)\,\pi_2 - p_1(x)\,\pi_1$. To this end, let

$$f_i(\Theta, x) = 1_{\{x \in V_{\theta_i}\}}\ \nabla_{\theta_i}\rho(\theta_i, x)\ \left(1_{\{i \le k_0\}} - 1_{\{i > k_0\}}\right) \tag{10}$$

then we see from (9) that

$$h_i(\Theta) = \int_\Omega f_i(\Theta, x)\, q(x)\, dx. \tag{11}$$

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

$$\pi_2 = P(\, d_{y_n} = 2\,) \quad \text{and} \quad \pi_1 = P(\, d_{y_n} = 1\,), \tag{12}$$

$\pi_1 + \pi_2 = 1$. 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 it is shown 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,

$$\begin{aligned}
E_z[\,H_i(\Theta,z)\,] &= E_z\left[1_{\{d=1\}}H_i(\Theta,(y,1))\right] + E_z\left[1_{\{d=2\}}H_i(\Theta,(y,2))\right] && (13)\\
&= E_1\left[\,H_i(\Theta,(y,1))\right]\pi_1 + E_2\left[\,H_i(\Theta,(y,2))\right]\pi_2 && (14)\\
&= E_1\left[\,\gamma(1,d_{\theta_i},y,\Theta)\;\nabla_{\theta_i}\rho(\theta_i,y)\right]\pi_1 \\
&\qquad +E_2\left[\,\gamma(2,d_{\theta_i},y,\Theta)\;\nabla_{\theta_i}\rho(\theta_i,y)\right]\pi_2 && (15)\\
&= E_1\left[1_{y\in V_{\theta_i}}\left(-1_{\{i\le k_0\}}+1_{\{i>k_0\}}\right)\nabla_{\theta_i}\rho(\theta_i,y)\right]\pi_1 \\
&\qquad +E_2\left[1_{y\in V_{\theta_i}}\left(1_{\{i\le k_0\}}-1_{\{i>k_0\}}\right)\nabla_{\theta_i}\rho(\theta_i,y)\right]\pi_2 && (16)\\
&= -E_1\left[f_i(\Theta,y)\right]\pi_1 + E_2\left[f_i(\Theta,y)\right]\pi_2 && (17)\\
&= h_i(\Theta). && (18)
\end{aligned}$$

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

$$E_z\left[\;\|H_i(\Theta,z)-h_i(\Theta)\|^2\;\right] = E_z\left[\;\|H_i(\Theta,z)\|^2\right] - \|h_i(\Theta)\|^2 \qquad (19)$$

where

$$\begin{aligned}
E_z\left[\;\|H_i(\Theta,z)\|^2\;\right] &= E_z\left[\,\|\nabla_{\theta_i}\rho(\theta_i,y)\|^2\right] && (20)\\
&= E_1\left[\|\nabla_{\theta_i}\rho(\Theta,y)\|^2\right]\pi_1 + E_2\left[\|\nabla_{\theta_i}\rho(\Theta,y)\|^2\right]\pi_2 && (21)\\
&= \sum_{i=1}^{k}\int_{V_{\theta_i}}\|\,\nabla_{\theta_i}\rho(\theta_i,x)\|^2\,\left(p_1(x)\,\pi_1 + p_2(x)\,\pi_2\right)dx && (22)
\end{aligned}$$

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 $\theta\in Q$

$$|\nabla_\theta\rho(\theta,x)| < C_1(1+|x|^{q_1}). \qquad (23)$$

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

We further assume that the sequence α_n satisfies $\sum\alpha_n=\infty$ and $\sum\alpha_n^\lambda<\infty$ for some $\lambda\ge 1$. We now state the two convergence theorems alluded to.

Theorem 1 *Let $\{z_n\}$ be the sequence of independent, identically distributed random vectors given above. Suppose $\{\alpha_n\}$ and $\rho(\theta,x)$ satisfy the properties above. Assume that the pattern densities $p_1(x)$ and $p_2(x)$ are continuous 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 $X_0 = x$

$$\lim_{\alpha_1 \downarrow 0} P_{x,a}\{ \sup_{n \le m(T)} |\Theta_n - \bar{\Theta}_a(t_n)| > \delta\} = 0 \tag{24}$$

where Θ_n satisfies (7) and $\bar{\Theta}_a(t)$ satisfies (8) with $h(\Theta)$ defined in (9). Here $t_n = \sum_{i=1}^{n} \alpha_i$.

Theorem 2 *In addition to the conditions of Theorem 1, assume $\bar{\Theta}^*$ is a locally asymptotically stable equilibrium of (8) with domain of attraction D^*. Let Q be a compact subset of D^*. If $\Theta_n \in Q$ for infinitely many n then*

$$\lim_{n \to \infty} \Theta_n = \bar{\Theta}^* \quad a.s. \tag{25}$$

Proof of Theorem 1:

We need only verify that [H.1]–[H'.5] in (Benveniste, Metivier & Priouret [1987,Chapter 4]) are satisfied then apply their results. 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

$$\begin{aligned} P\{z_{n+1} \in B \,|\, \mathcal{F}_n\} &= \Pi_\Theta(z_n, B) && (26) \\ &= \int_B p_2(x)\pi_2 \, dx + \int_B p_1(x)\pi_1 \, dx. && (27) \end{aligned}$$

Hence hypothesis [H.2] is satisfied.

Note that

$$|H_i(\Theta, z)| = |\nabla_{\theta_i} \rho(\theta_i, z)|. \tag{28}$$

Therefore, in view of (c) above [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 property (c).

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

The proof of Theorem 2 is similar that of Theorem 1.

5.2 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

$$\hat{p}_1(x;N) = \frac{1}{N_1}\sum_{j=1}^{N}\delta(x=y_j)1_{\{d_{y_j}=1\}} \tag{29}$$

$$\hat{p}_2(x;N) = \frac{1}{N_2}\sum_{j=1}^{N}\delta(x=y_j)1_{\{d_{y_j}=2\}}, \tag{30}$$

and estimates of the priors are given by

$$\hat{\pi}_1 = \frac{N_1}{N} \qquad \text{and} \qquad \hat{\pi}_2 = \frac{N_2}{N} \tag{31}$$

where $\delta(x)$ is the delta function. Let $H(\Theta, z)$ be the vector of components defined in (4). We see that

$$h_i(\Theta;N) = \hat{E}_z[H_i(\Theta,z)] \tag{32}$$

$$= \hat{E}_1[H_i(\Theta,(y,1))]\hat{\pi}_1 + \hat{E}_2[H_i(\Theta,(y,2))]\hat{\pi}_2 \tag{33}$$

$$= -\frac{1}{N}\sum_{j=1}^{N}\nabla_{\theta_i}\rho(y_j,\theta_i)\,1_{\{y_j\in V_{\theta_i}\}}(1_{\{d_{y_j}=d_{\theta_i}\}} - 1_{\{d_{y_j}\neq d_{\theta_i}\}}). \tag{34}$$

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 *Let $\{z_{n_j}\}_{j=1}^\infty$ be the independent sequence of random vectors picked from Z as described above. Suppose $\{\alpha_n\}$ and $\rho(\theta,x)$ satisfy the same hypotheses as before.*

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 $X_0 = x$

$$\lim_{\alpha_1\downarrow 0} P_{x,a}\{\sup_{n\le m(T)}|\Theta_n - \bar{\Theta}_a(t_n;N)| > \delta\} = 0 \tag{35}$$

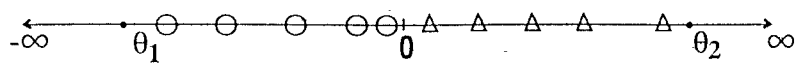


Figure 5: *A possible distribution of observations and two Voronoi vectors.*

where Θ_n *satisfies (7) and* $\bar{\Theta}_a(t;N)$ *satisfies (8) with* $h(\Theta;N)$ *defined by (34). Here* $t_n = \sum_{i=1}^{n} \alpha_i$.

Theorem 4 *In addition to the conditions of Theorem 3, assume* $\bar{\Theta}^*$ *is a locally asymptotically stable solution of (8) with* $h(\Theta;N)$ *defined by (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*

$$\lim_{n\to\infty} \Theta_n = \bar{\Theta}^* \qquad a.s. \tag{36}$$

The proofs of these theorems follow directly from the proofs of Theorem 1 and Theorem 2 with $h(\Theta) = h(\Theta;N)$ and $P(z = z_i) = 1/N$. We note that by the Strong Law of Large Numbers (SLLN) as N_1 and N_2 go to infinity, $h(\Theta;N)$ converges with probability one to the function $h(\Theta)$ given by (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.

5.3 Remarks on Convergence

The convergence results above require that the initial conditions are close to the stable points of (8), i.e., within the domain of attraction of a stable equilibrium, in order for the algorithm to converge. Next a modification to the LVQ algorithm is presented which increases the number of stable equilibrium for equation (8) and hence increases the chances of convergence. In the remainder of this section a simple example is presented 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 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 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.

6 The Modified LVQ Algorithm

Recall that during the update procedure in (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. In order to analyze this procedure mathematically, we insist that the correction be done at each iteration[1]. Let

$$g_i(\Theta; N) = \begin{cases} 1 & \text{if } \frac{1}{N}\sum_{j=1}^{N} 1_{\{y_j \in V_{\theta_i}\}} 1_{\{d_{y_j}=1\}} > \frac{1}{N}\sum_{j=1}^{N} 1_{\{y_j \in V_{\theta_i}\}} 1_{\{d_{y_j}=2\}} \\ 2 & \text{otherwise.} \end{cases} \tag{37}$$

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

$$\theta_i(n+1) = \theta_i(n) + \alpha_n \, \gamma(d_{y_n}, g_i(\Theta_n; N), y_n, \Theta_n) \, \nabla_{\theta_i(n)} \rho(\theta_i(n), y_n). \tag{38}$$

This equation has the same form as (4) with the function $\overline{H}(\Theta, z)$ defined from (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

$$\begin{aligned} \bar{h}_i(\Theta; N) &= E_z[\overline{H}_i(\Theta, z)] & (39) \\ &= -\bar{\gamma}_i(\Theta; N) \frac{1}{N} \sum_{j=1}^{N} \nabla_{\theta_i} \rho(\theta_i, y_j) 1_{\{y_j \in V_{\theta_i}\}} (1_{\{d_{y_j}=2\}} - 1_{\{d_{y_j}=1\}}) & (40) \\ &= \bar{\gamma}_i(\Theta; N) \, (1_{\{d_{\theta_i}=2\}} - 1_{\{d_{\theta_i}=1\}}) \, h_i(\Theta; N) & (41) \end{aligned}$$

where

$$\bar{\gamma}_i(\Theta; N) = \text{sign} \left\{ \frac{1}{N} \sum_{j=1}^{N} 1_{\{y_j \in V_{\theta_i}\}} (1_{\{d_{y_j}=2\}} - 1_{\{d_{y_j}=1\}}) \right\} \tag{42}$$

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

and $h_i(\Theta; N)$ is as defined in (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

$$\bar{h}_i(\Theta) = -\text{sign}\left\{\int_{V_{\theta_i}} q(x)\,dx\right\} \int_{V_{\theta_i}} \nabla_{\theta_i}\rho(\theta_i, x)\, q(x)\,dx \tag{43}$$

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

$$\hat{h}_i(\Theta) = -\int_{V_{\theta_i}} \nabla_{\theta_i}\rho(\theta_i, x)\,|q(x)|\,dx. \tag{44}$$

The right-hand side of the last equation is minus the (ith component of) gradient of the cost function

$$J(\Theta) = \sum_{i=1}^{k} \int_{V_{\theta_i}} \rho(\theta_i, x)\,|q(x)|\,dx. \tag{45}$$

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

7 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

$$q(x, \theta_i) = p_{d_{\theta_i}}(x)\pi_{d_{\theta_i}} - \sum_{\substack{j=1 \\ j\neq d_{\theta_i}}}^{\ell} p_j(x)\,\pi_j \tag{46}$$

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 (11) are given by

$$h_i(\Theta) = -\int_{V_{\theta_i}} \nabla_{\theta_i}\rho(\theta_i, x)\, q(x, \theta_i)\,dx \quad i = 1, \ldots, k. \tag{47}$$

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

$$\int_{V_{\theta_i}} q(x)\,dx < 0 \quad \text{for } i \leq k_0 \tag{48}$$

$$\int_{V_{\theta_i}} q(x)\,dx > 0 \quad \text{for } i > k_0 \tag{49}$$

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

$$d_{\theta_i} = \arg \max_{j=1,\ldots,\ell} \{ \int_{V_{\theta_i}} p_j(x)\, \pi_j \, dx \} \tag{50}$$

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

$$\int_{V_{\theta_i}} q(x, \theta_i)\, dx > 0 \qquad i = 1, \ldots, k. \tag{51}$$

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

$$P_c(V_{\theta_i}) = \int_{V_{\theta_i}} p_{d_{\theta_i}}(x)\, \pi_{d_{\theta_i}} \, dx \tag{52}$$

and the probability of error is equal to

$$P_e(V_{\theta_i}) = \int_{V_{\theta_i}} \sum_{\substack{j=1 \\ j \neq d_{\theta_i}}}^{\ell} p_j(x)\, \pi_j \, dx. \tag{53}$$

Hence this requirement (expressed by equation (51)) is nothing more than the requirement that the probability of correct decision be greater than the probability of error for each region.

8 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 convergent estimator of the Bayes risk. 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.

8.1 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 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

$$r^* \leq r \leq 2r^*(1 - r^*) \leq 2r^*. \tag{54}$$

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

8.2 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 (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

$$\text{Vol}(V_{0_i}) = O(\frac{1}{k_N}). \tag{55}$$

Here we assume that the pattern densities have compact support. Let $y \in V_{0_i}$ and suppose that

$$\hat{q}(y; N) = \frac{1}{N}\sum_{j=1}^{N} Y_j \tag{56}$$

with

$$Y_j = \frac{1_{\{y_j \in V_{0_i}\}}(1_{\{d_{y_j}=2\}} - 1_{\{d_{y_j}=1\}})}{\text{Vol}(V_{0_i})}. \tag{57}$$

Then an argument using the weak law of large numbers shows that $\hat{q}(y;N)$ converges in probability to $q(y)$. 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.

9 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.

9.1 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 paper. 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]).

9.2 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.

10 Discussion

In this paper, 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 and 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. We are currently using these techniques to further the design of efficient LVQ schemes.

11 Acknowledgements

This work was supported by the National Science Foundation through grant NSF CDR-8803012, Texas Instruments through a TI/SRC Fellowship and the Office of Naval Research through an ONR Fellowship.

12 References

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

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.

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.

A. LaVigna [1989], "Nonparametric Classification using Learning Vector Quantization," Ph.D. Dissertation, Department of Electrical Engineering, University of Maryland.

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