

Convergence of a Neural Network Classifier

John S. Baras and Anthony LaVigna

Electrical Engineering Department and

Systems Research Center

University of Maryland, College Park, Maryland 20742

Abstract

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 for a larger set of initial conditions. Finally, we show that LVQ is a general histogram classifier and that its risk converges to the Bayesian optimal risk as the appropriate parameters go to infinity with the number of past observations.

1 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,

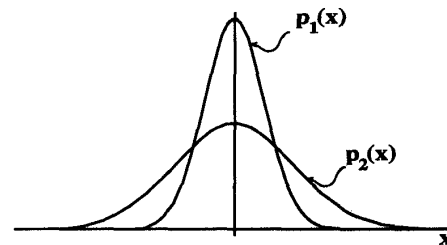


Figure 1: Plot of two pattern densities

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 the next sections, the LVQ algorithm is presented. Using theorems from stochastic approximation, we prove that the update algorithm converges under suitable conditions. We discuss a modification to the algorithm which provides convergence for a larger set of initial conditions. Finally, we prove that the detection error associated with LVQ converges to the lowest possible error as the appropriate parameters go to infinity.

2 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



Figure 2: Plot of decision regions

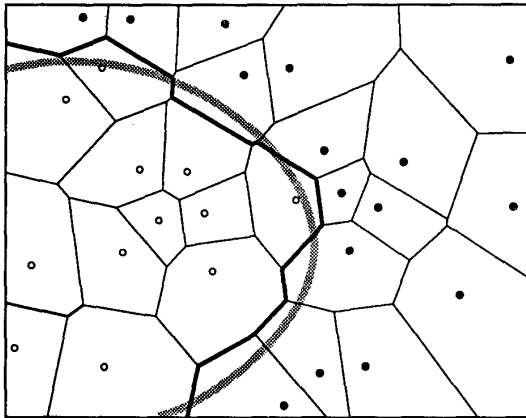


Figure 3: Voronoi vectors and their approximate decision region.

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 nonparametric density estimators and use them to find approximate decision regions.

LVQ is an algorithm which estimates the decision regions directly. Unlike some nonparametric classification schemes, it does not first estimate the densities and then proceed to calculate the 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 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 3. 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 (LaVigna [1989]). 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]).

3 Description of the Algorithm

Now we mathematically describe the LVQ algorithm. To begin with, let the past observations lie in \mathbb{R}^d and let $\Theta = \{\theta_1, \dots, \theta_k\}$ 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

$$V_{\theta_i} = \{x \in \mathbb{R}^d \mid \rho(\theta_i, x) < \rho(\theta_j, x), j \neq i\} \quad i = 1, \dots, k. \quad (1)$$

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} . 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:

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

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

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

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 when the number of past observations becomes arbitrarily large and each observation is presented once. Using similar arguments it is possible to show convergence 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]).

4 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 Θ^* , 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

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

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

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

or, more compactly,

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

where 1_A is the indicator function of the set A . This is a stochastic approximation algorithm with $g_n(\Theta, x) \equiv 0$ (see (Benveniste, Metivier & Priouret [1987])). It has the form

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

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

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

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

Throughout this section we consider the case of two pattern densities; convergence is obtained via the ODE method discussed in (Benveniste, Metivier & Priouret [1987]).

4.1 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

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

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

$$f_i(\Theta, x) = 1_{\{x \in V_{\theta_i}\}} \nabla_{\theta_i} \rho(\theta_i, x) (1_{\{i \leq k_0\}} - 1_{\{i > k_0\}}) \quad (10)$$

then we see from (9) that

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

Assume that the training data $\{z_n\}_{n=1}^N$ consist of pairs of independent, identically distributed observations with the property that if $z_n = (y_n, d_{y_n})$ then 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 denote 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,

$$\begin{aligned} E_z[H_i(\Theta, z)] &= E_1[H_i(\Theta, (y, 1))] \pi_1 + E_2[H_i(\Theta, (y, 2))] \pi_2 \\ &= E_1[\gamma(1, d_{\theta_i}, y, \Theta) \nabla_{\theta_i} \rho(\theta_i, y)] \pi_1 \\ &\quad + E_2[\gamma(2, d_{\theta_i}, y, \Theta) \nabla_{\theta_i} \rho(\theta_i, y)] \pi_2 \\ &= -E_1[f_i(\Theta, y)] \pi_1 + E_2[f_i(\Theta, y)] \pi_2 \\ &= h_i(\Theta). \end{aligned}$$

From the results above it is possible to show that $\xi_i(n)$ is a zero mean process with finite variance.

We assume that $\rho(\theta, x)$ satisfies the following three properties:

- $\rho(\theta, x)$ is a twice continuously differentiable function of θ and x and for every fixed $x \in \mathbb{R}^d$ it is a convex function of θ .
- For any fixed x , if $\theta(k) \rightarrow \infty$ as $k \rightarrow \infty$, then $\rho(\theta(k), x) \rightarrow \infty$.
- For every compact $Q \subset \mathbb{R}^d$, there exist constants C_1 and q_1 such that for all $\theta \in Q$

$$|\nabla_{\theta} \rho(\theta, x)| < C_1(1 + |x|^{q_1}). \quad (12)$$

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 $\lambda \geq 1$. 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 \mathbb{R}^d for all $t \in [0, T]$, then for every $\delta > 0$ and all $X_0 = x$

$$\lim_{\alpha_{10}} P_{x,a} \left\{ \sup_{n \leq m(T)} |\Theta_n - \bar{\Theta}_a(t_n)| > \delta \right\} = 0 \quad (13)$$

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 Θ^* 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

$$\lim_{n \rightarrow \infty} \Theta_n = \bar{\Theta}^* \quad a.s. \quad (14)$$

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

$$\begin{aligned} P\{z_{n+1} \in B | \mathcal{F}_n\} &= \Pi_\Theta(z_n, B) \\ &= \int_B p_2(x) \pi_2 dx + \int_B p_1(x) \pi_1 dx. \end{aligned} \quad (15)$$

Hence hypothesis [H.2] is satisfied.

Note that

$$|H_i(\Theta, z)| = |\nabla_{\theta_i} \rho(\theta_i, z)|. \quad (17)$$

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.

4.2 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.

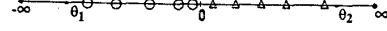


Figure 4: A possible distribution of observations and two Voronoi vectors.

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 \circ represent an observation from pattern 2 and let Δ represent an observation from pattern 1. We assume that the observations are scalar and that $\rho(\theta, x)$ is the Euclidean distance function. Figure 4 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 4. 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.

5 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 iteration¹. Let

$$g_i(\Theta; N) = \begin{cases} 1 & \text{if } \frac{1}{N} \sum_{j=1}^N 1_{\{y_j \in V_{\theta_i}\}} 1_{\{d_{y_j} = 1\}} > \\ & \frac{1}{N} \sum_{j=1}^N 1_{\{y_j \in V_{\theta_i}\}} 1_{\{d_{y_j} = 2\}} \\ 2 & \text{otherwise.} \end{cases} \quad (18)$$

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

$$\theta_i(n+1) = \theta_i(n) + \alpha_n \gamma(d_{y_n}, g_i(\Theta_n; N), y_n, \Theta_n) \nabla_{\theta_i(n)} \rho(\theta_i(n), y_n). \quad (19)$$

This equation has the same form as (4) with the function $\bar{H}(\Theta, z)$ defined from (19) replacing $H(\Theta, z)$.

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

We can show that as the number of observations becomes large that the function in the ODE related to (19) converges with probability one to $\bar{h}(\Theta)$ given by

$$\bar{h}_i(\Theta) = -\text{sign} \left\{ \int_{V_{\theta_i}} q(x) dx \right\} \int_{V_{\theta_i}} \nabla_{\theta_i} \rho(\theta_i, x) q(x) dx \quad (20)$$

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

$$\hat{h}_i(\Theta) = - \int_{V_{\theta_i}} \nabla_{\theta_i} \rho(\theta_i, x) |q(x)| dx. \quad (21)$$

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

$$J(\Theta) = \sum_{i=1}^k \int_{V_{\theta_i}} \rho(\theta_i, x) |q(x)| dx. \quad (22)$$

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

6 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.

6.1 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

$$r^* \leq r \leq 2r^*(1 - r^*) \leq 2r^*. \quad (23)$$

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

6.2 Other Choices for 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 $\Theta_N = \{\theta_1, \dots, \theta_{k_N}\}$ and assume that the Voronoi vectors are initialized so that

$$\text{Vol}(V_{\theta_i}) = O\left(\frac{1}{k_N}\right). \quad (24)$$

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

$$\hat{q}(y; N) = \frac{1}{N} \sum_{j=1}^N Y_j \quad (25)$$

with

$$Y_j = \frac{\mathbf{1}_{\{y_j \in V_{\theta_i}\}} (\mathbf{1}_{\{d_{y_j}=2\}} - \mathbf{1}_{\{d_{y_j}=1\}})}{\text{Vol}(V_{\theta_i})}. \quad (26)$$

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.

7 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. Finally, 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.

8 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.

9 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.
- A. LaVigna [1989], "Nonparametric Classification using Learning Vector Quantization," Ph.D. Dissertation, Department of Electrical Engineering, University of Maryland.