Proceedings of the 38th
Conference on Decision & Control
Phoenix, Arizona USA • December 1999

Adaptive Classification Based on Compressed Data Using Learning Vector Quantization[1]

John S. Baras [2] Subhrakanti Dey [3]

Abstract

Classification problems using compressed data are becoming increasingly important in many applications with large amounts of sensory data and large sets of classes. These applications range from aided target recognition (ATR), to medical diagnosis, to speech recognition, to fault detection and identification in manufacturing systems. In this paper, we develop and analyze a learning vector quantization (LVQ) based algorithm for the combined compression and classification problem. We show convergence of the algorithm using techniques from stochastic approximation, namely, the ODE method.

Index Terms- **Learning vector quantization, classification, stochastic approximation, compression, non-parametric**

1 Introduction

Quite often in applications, we are faced with the problem of classifying signals (or objects) from vast amounts of noisy data. Equally often, the number of different distinct signals (classes) that we have in the problem may be quite large. If we could compress each observation (observed signal) significantly without distorting or annihilating the most significant features used for classification, we can achieve significant advantages in two directions:
(i) We can reduce significantly the memory required for storing both the on-line and class model data;
(ii) We can increase significantly the speed of searching and matching that is essential in any classification problem.

Furthermore, performing classification on compressed data can result in better classification, due to the fact that compression (done correctly) can reduce the noise more than the signal [1]. For all these reasons, it is important to develop methods and algorithms to perform classification of compressed data, or to analyze jointly the problem of compression and classification. In [2] and [3], vector quantization methods have been used for minimizing both the distortion of compressed images and errors in classifying their pixel blocks.

There is yet another significant advantage in investigating the problem of combined compression and classification. If such a framework is developed, we can then analyze progressive classification schemes, which offer significant advantages for both memory savings and for speeding up searching and matching. Progressive classification uses very compressed representations of the signals at first to perform many simple (and therefore fast) matching tests, and then progressively perform fewer but more complex (and therefore slower) matching tests, as needed for classification. In the last four years, we have analyzed such progressive classification schemes on a variety of problems with substantial success. The structure of the algorithms we have developed has remained fairly stable, regardless of the particular application. This structure consists of a multiresolution preprocessor followed by a tree-structured classifier as the postprocessor. Sometimes a nonlinear feature extraction component needs to be placed between these two components. Often the postprocessor incorporates learning.

To date, we have utilized wavelets as the multiresolution preprocessor and Tree-structured-vector-quantization (TSVQ) as the clustering postprocessor. We have applied the resulting WTSVQ algorithm to various ATR problems based on radar [4] [5] [6], ISAR and face recognition problems [7]. We have established similar results on ATR based on FLIR using polygonization of object silhouettes [8] [9] as the multiresolution preprocessor. Incorporation of compression into these algorithms is part of our current research.

As a first step towards developing a progressive classification scheme with compression, we need to develop an algorithm for combined compression and classification at a fixed resolution. As opposed to the algorithm described in [3] that achieves this with *a-posteriori* estimation of the probability models underlying the different classes of signals, our goal is to develop an algorithm that is nonparametric, in the sense that it does

[1]Research supported by ONR contract 01-5-28834 under the MURI Center for Auditory and Acoustics Research, by NSF grant 01-5-23422 and by the Lockheed Martin Chair in Systems Engineering.

[2]Department of Electrical and Computer Engineering and Institute for Systems Research, University of Maryland, College Park, MD 20742, e-mail: baras@isr.umd.edu.
[3]Department of Systems Engr., RSISE, Australian National University, Canberra, ACT 0200 Australia, e-mail: borkar@tifr.res.in.

0-7803-5250-5/99/$10.00 © 1999 IEEE

not use estimates of probability distributions of the underlying sources generating the data. In this paper, we achieve that goal by using a variation of Learning Vector Quantization (LVQ), that cleverly takes into account the distortion present. LVQ as described in [10] [11], although primarily designed to perform classification, achieves some compression as a byproduct since it is inherently a vector quantization algorithm (an observation also made in [2] [3]). However, our algorithm is designed to obtain a systematic trade-off between its compression and classification performances by minimizing a linear combination of the compression error (measured by average distortion) and classification error (measured by Bayes risk) using a variation of LVQ based on a stochastic approximation scheme. The convergence analysis of this algorithm essentially follows similar techniques as presented in [12] and as used in [13]. However, our treatment is considerably simpler since to start with, we recognize that the algorithm is a special class of the Robbins-Monro algorithm.

In Section 2, we describe the LVQ-based algorithm for combined compression and classification. In Sections 2.1 and 2.2, we provide analysis and convergence of the algorithm using stochastic approximation techniques and the so-called ODE method. Section 3 presents some concluding remarks.

2 Classification using compressed data and learning vector quantization

Learning vector quantization (LVQ) introduced in [11] is a nonparametric method of pattern classification. As opposed to parametric methods, this method does not attempt to obtain *a-posteriori* estimates of the underlying probability models of the different patterns that generate the data to be classified. As noted in [14] (p. 266), classification is easier than density estimation. So an algorithm such as ours offers considerable advantages over algorithms that use Bayes rules based on estimated class densities. LVQ simply uses a set of training data for which the classes are known in a supervised learning algorithm to divide the data space into a number of Voronoi cells represented by the corresponding Voronoi vectors and their associated class decisions. Using the training vectors, these Voronoi vectors are updated iteratively until they converge. The algorithm involves three main steps:

1. Find out which Voronoi cell a given training vector belongs to by the nearest-neighbor rule.
2. If the decision of the training vector coincides with that of the Voronoi vector of this particular cell, move the Voronoi vector towards the training vector, else, move it away from the training vector.

   All the other Voronoi vectors are not changed.
3. Obtain the next training vector and perform the first two steps.

This process is usually carried out in multiple passes of the finite set of training vectors. A detailed description of this algorithm with a preliminary analysis of its convergence properties using stochastic approximation techniques of [12] has been given in [13]. A sketch of a proof for the convergence of the classification error achieved by the LVQ algorithm was described in [13]. If we have N training pairs $\{(X_i, d_{X_i}),\ i = 1, \ldots, N\}$, we denote by K_N the number of Voronoi vectors (or the number of sets in the corresponding partitions in $\mathbb{R}^d$). It was noted in [13] that as $K_N \to \infty$, if the Voronoi vectors are initialized according to a uniform partition of $\mathbb{R}^d$, then the LVQ algorithm does not move the vectors from their initial values. As a result, the error associated with initial conditions dominates the overall classification error. By considering the LVQ algorithm for large K_N without learning iterations, it can be shown as sketched in [13] that the classification error in LVQ converges to the optimal Bayes error as long as the volume of the Voronoi cells goes to zero as $K_N \to \infty$, provided we have that $\lim_{N\to\infty} K_N \to \infty$ while $\lim_{N\to\infty} \frac{K_N}{N} \to 0$. More complete results on the weak and strong consistency of the error of classification rules based on partitions (including data dependent clustering partitions) can be found in Theorem 21.2 (p. 368) and Theorem 21.5 (p. 379) of [14]. We will discuss the second theorem in Section 2.1 a little more. These results hold for general distributions for (X, d) (i.e., pairs of data and class labels) with compact support and general functions measuring data proximity, satisfying the typical conditions given here and in [13].

Although its primary goal is to classify the data into different patterns, the LVQ algorithm compresses the data in the process into a codebook of size equal to the number of Voronoi cells, where each Voronoi vector is the codeword representing all the vectors belonging to that cell.

In what follows, we present a simple variation of the LVQ algorithm in [13], that achieves the task of combined compression and classification. We present a convergence analysis of this algorithm much along the lines of [13]. However, we present a simpler analysis by recognizing that the algorithm is a special case of the Robbins Monro algorithm. Also, simulation results show that as a certain parameter is increased, the compression error gradually decreases compared to the error achieved by the standard LVQ (represented by the value zero of this parameter).

In the next subsection, we introduce our notation and describe the algorithm.

Algorithm for combined compression and classification

Consider a complete probability space $(\Omega, \mathcal{F}, P)$. Let $X_l \in \mathbb{R}^d$, $l = 1, 2, \ldots, N$, represent the training vectors defined on this space, generated by either of the two patterns 1 or 2. The *a-priori* probabilities of the two patterns are π_1 and π_2 respectively and the corresponding pattern densities are $p_1(x)$ and $p_2(x)$ respectively such that

$$P(X_l \in B) = \pi_1 \int_B p_1(x)dx + \pi_2 \int_B p_2(x)dx \quad (1)$$

We also assume that X_l is independent of $X_j, j \neq l$.

The Voronoi vectors are represented by $\theta_i \in \mathbb{R}^d$, $i = 1, 2, \ldots, K$ and the corresponding Voronoi cells are represented by V_{θ_i}. Let the decision associated with the training vector X_l be represented by d_{X_l} and that of the cell V_{θ_i} by d_{θ_i}, where $d_{X_l}, d_{\theta_i} \in \{1, 2\}$.

Consider a non-increasing sequence of positive real numbers ϵ_n, $n = 1, 2, \ldots$, such that

Assumption 2.1 $\sum_{n=1}^{\infty} \epsilon_n = \infty$.

Consider also a proximity metric function $\rho(\theta, x)$ which satisfies the following assumptions:

Assumption 2.2 *$\rho(\theta, x)$ 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.*

Assumption 2.3 *For any fixed x, if $|\theta(k)| \to \infty$, as $k \to \infty$, then $\rho(\theta(k), x) \to \infty$.*

Assumption 2.4 *For every compact set $Q \subset \mathbb{R}^d$, there exist constants C_1 and q_1 such that for all $\theta \in Q$,*

$$|\nabla_\theta \rho(\theta, x)| < C_1(1 + |x|^{q_1}) \quad (2)$$

In Assumptions 2.2-2.4, $|.|$ is the Euclidean norm in $\mathbb{R}^d$ (whenever the quantity inside is a vector, and this should be obvious from the context). An example of a proximity function that satisfies the properties above is $\rho(\theta, x) = |\theta - x|^2$.

Define further the following quantities:

Definition 2.1

$$\gamma(d_{X_{n+1}}, d_{\theta_i(n)}, X_{n+1}, \Theta(n)) = -1_{X_{n+1} \in V_{\theta_i(n)}} (1_{d_{X_{n+1}} = d_{\theta_i(n)}} - 1_{d_{X_{n+1}} \neq d_{\theta_i(n)}}) \quad (3)$$

where $\Theta(n) = (\theta_1(n), \ldots, \theta_K(n))'$ and $\theta_i(n)$ is the n-th iterate of θ_i, $n \geq 0$. Also 1_A is the indicator function that takes the value 1 if A is true and 0 otherwise.

Definition 2.2

$$g_i(\Theta(n); N) = \begin{cases} 1 \text{ if } \frac{1}{N} \sum_{j=1}^{N} 1_{X_j \in V_{\theta_i(n)}} 1_{d_{X_j}=1} > \\ \quad \frac{1}{N} \sum_{j=1}^{N} 1_{X_j \in V_{\theta_i(n)}} 1_{d_{X_j}=2} \\ 2 \text{ otherwise} \end{cases} \quad (4)$$

Remark 2.1 Note that $g_i(\Theta(n); N)$ above denotes the decision associated with the i-th Voronoi cell according to the majority vote rule.

With the above definitions and assumptions, we can now write the following multi-pass combined compression and classification algorithm for (scalar) $\lambda \geq 0$,

1. *Initialization*: The algorithm is initialized with $\Theta(0)$ usually found by running a vector quantization algorithm, e.g., LBG [15] algorithm over the set of training vectors.
2. $n = 0$.
3. *Assigning the training vectors to their respective cells*: Find $i_l = \text{argmin}_m |\theta_m(n) - X_l|^2$, $l = 1, 2, \ldots, N$. Then X_l belongs to $V_{\theta_{i_l}(n)}$.
4. *Cell decisions*: Calculate $g_i(\Theta(n); N)$, $i = 1, 2, \ldots, K$.
5. *Updating the Voronoi vectors*: For $i \in \{1, 2, \ldots, K\}$,

$$\theta_i(n+1) = \theta_i(n) + \epsilon_{n+1}(-\lambda 1_{X_{n+1} \in V_{\theta_i(n)}} + \gamma(d_{X_{n+1}}, g_i(\Theta(n); N), X_{n+1}, \Theta(n))) \nabla_\theta \rho(\theta, X_{n+1}) \mid_{\theta=\theta_i(n)} \quad (5)$$

6. $n \leftarrow n + 1$.
7. If $n < N$, repeat Steps 3-6. If $n = N$, repeat Steps 3-4.

The above algorithm can be executed for multiple passes over the same training set (in case the size of the training set is small) by using the values $\Theta(N)$ from the m-th pass to initialize the algorithm for the $(m+1)-th$ pass, until $m = M$ where M is the maximum number of passes.

Remark 2.2 Note that Step 5, i.e., updating of the Voronoi vectors, can be written in the following simplified manner:
If $X_{n+1} \in V_{\theta_i(n)}$, then

$$\theta_i(n+1) = \begin{cases} \theta_i(n) - \epsilon_{n+1}(\lambda+1)\nabla_\theta \rho(\theta, X_{n+1})\mid_{\theta=\theta_i(n)} \\ \qquad if\ d_{X_{n+1}} = g_i(\Theta(n); N) \\ \theta_i(n) - \epsilon_{n+1}(\lambda-1)\nabla_\theta \rho(\theta, X_{n+1})\mid_{\theta=\theta_i(n)} \\ \qquad if\ d_{X_{n+1}} \neq g_i(\Theta(n); N) \end{cases} \quad (6)$$

For $j \neq i$, $\theta_j(n+1) = \theta_j(n)$.

2.1 Analysis of the combined compression and classification algorithm

In this subsection, we present a summary of the analysis of the above algorithm using the "mean ODE" method of [12]. For the complete analysis we refer to [20].

Denote the vectors

$$h(\Theta(n)) = (h_1(\Theta(n)), \ldots, h_K(\Theta(n))'$$

and

$$H(\Theta(n), X_{n+1}) = (H_1(\Theta(n), X_{n+1}), \ldots,$$
$$H_K(\Theta(n), X_{n+1}))'$$

where

$$H_i(\Theta(n), X_{n+1}) = (-\lambda 1_{X_{n+1} \in V_{\theta_i(n)}} + \gamma(d_{X_{n+1}},$$
$$g_i(\Theta(n); N), X_{n+1}, \Theta(n)))\nabla_\theta \rho(\theta, X_{n+1})\mid_{\theta=\theta_i(n)} \quad (7)$$

and $h_i(\Theta(n))$, $i = 1, 2, \ldots, K$ is defined in Definition 2.4. Note that one can write the above algorithm (5) in the following manner:

$$\Theta(n+1) = \Theta(n) + \epsilon_{n+1} H(\Theta(n), X_{n+1}),\ n \geq 0\ . \quad (8)$$

Note that this is a special case of the general stochastic approximation algorithm of [12], quoted in Section 2, [13].

Define

$$\begin{aligned} p(x) &= p_1(x)\pi_1 + p_2(x)\pi_2 \\ q(x) &= p_2(x)\pi_2 - p_1(x)\pi_1\ . \end{aligned} \quad (9)$$

Due to the assumption that $\{X_l\}$, $l = 1, 2, \ldots$, is a sequence of *i.i.d.* random vectors and the fact that they are distributed independently of $\Theta(l)$, the transition probability function $\Pi_{\Theta(n)}(A, X_n) \triangleq P(X_{n+1} \in A \mid \mathcal{F}_n)$ is given by $\mu(A) = \int_A p(x)dx$, where $\mathcal{F}_n \triangleq \sigma\{\Theta(0), X_0, \ldots, \Theta(n), X_n\}$ (the σ-algebra generated by these random variables). This makes the above algorithm a special case of the Robbins-Monro algorithm with the transition probability function being independent of $\Theta(n)$.

Now, we introduce the following definitions:

Definition 2.3

$$\bar{\gamma}_i(\Theta(n); N) = sign\left(\frac{1}{N}\sum_{j=1}^{N} 1_{X_j \in V_{\theta_i(n)}}(1_{d_{X_j}=2} - 1_{d_{X_j}=1})\right) \quad (10)$$

Definition 2.4

$$h_i(\Theta) = -\int_{V_{\theta_i}} [\bar{\gamma}_i(\Theta; N)q(x) + \lambda p(x)]$$
$$\nabla_\theta \rho(\theta, x)_{|\theta=\theta_i} dx,\ i = 1, 2, \ldots, K\ . \quad (11)$$

One can now prove the following Lemma:

Lemma 2.1

$$H_i(\Theta(n), X_{n+1}) = h_i(\Theta(n)) + \xi_i(n),\ i = 1, 2, \ldots, K\ , \quad (12)$$

where $\{\xi_i(n)\}$ is a $\mathcal{F}_n$-adapted martingale difference sequence such that

$$h_i(\Theta(n)) = E_a[H_i(\Theta(n), X_{n+1}) \mid \mathcal{F}_n],\ \forall i\ . \quad (13)$$

Here, E_a denotes expectation under P_a where P_a denotes the probability distribution for $\{X_n, \Theta(n)\}$, $n \geq 0$ where $\Theta(0) = a$. Note that since $\{X_n\}$ is a sequence of *i.i.d.* random vectors, P_a is functionally independent of X_0.

We write the mean ODE associated with (8) as

$$\dot{\Theta} = h(\Theta),\ \Theta(0) = a\ , \quad (14)$$

where

$$h_i(\Theta) = \lim_{n\to\infty} E_a[H_i(\Theta, X_{n+1})|\mathcal{F}_n] = \int H_i(\Theta, x)p(x)dx\ , \quad (15)$$

since in this case $\{X_n\}$ is a sequence of *i.i.d.* random variables where $P(X_{n+1} \in A \mid \mathcal{F}_n)$ is independent of $\Theta(k)$, $k \leq n$.

It is hard to establish a convergence result for general $h(\Theta)$ and often it is assumed that (14) has an attractor Θ^*, whose domain of attraction is given by D^* [12]. If Q is a compact subset of D^* and $\Theta(0) = a \in Q$, one can show that for any $\delta > 0$,

$$P\{\max_n ||\Theta(n) - \Theta(a, t_n)|| > \delta\} < C(\alpha, Q)\sum_n \epsilon_n^\alpha\ , \quad (16)$$

where $t_n = \sum_{i=1}^{n} \epsilon_i$ and $\Theta(a, t_n)$ is the solution to (14) for $t = t_n$, and $C(\alpha, Q)$ is a constant dependent on α and Q (see Theorem 4, page 45, [12]). Here, we have assumed Assumption 2.1.

One could also derive the following corollary (see Corollary 6, page 46, [12]), which states that under the assumptions (16) is true, for the set of trajectories $\{\Theta(n)\}$ that visit Q infinitely often, we have

$$\Theta(n) \to \Theta^*, \ P_a - a.s. \tag{17}$$

$$P\{\limsup_{n\to\infty} ||\Theta(n) - \Theta(a, t_n)|| > \delta\} = 0 \ . \tag{18}$$

Note that for a complete theory, it is essential to prove that the desired points of convergence Θ^* are indeed the stable equilibrium points of (14). One way to do this is to find a potential function $J(\Theta)$, if it exists, such that $h_i(\Theta) = -\nabla_{\theta_i} J(\Theta)$. Then one can apply results from Lyapunov stability to establish results for stable equilibrium by studying the local minima of $J(.)$ and their domains of attraction. Although, we refrain from such pursuits for the time being, we do notice that (see [13]) as $N \to \infty$, $\bar{\gamma}_i(\Theta; N) \to sign(\int_{V_{\theta_i}} q(x)dx)$ and using the mean value theorem when the size of each Voronoi cell is small, one can write that $h_i(\Theta)$ is approximately equal to

$$\bar{h}_i(\Theta) \approx -\int_{V_{\theta_i}} \nabla_\theta \rho(\theta, x)_{|_{\theta=\theta_i}} (|q(x)| + \lambda p(x))dx \ , \tag{19}$$

which is the negative gradient of the cost function

$$\bar{J}(\Theta) = \sum_{i=1}^{K} \int_{V_{\theta_i}} \rho(\theta_i, x)(|q(x)| + \lambda p(x))dx \ . \tag{20}$$

For those readers who are more oriented towards intuitive reasoning, we comment here that this was indeed the inspiration for obtaining the combined compression and classification algorithm given above. The reason for this intuition is that under general conditions, it can be shown following the sketch of the proof given in [13], and the methods and results in chapter 21 of Devroye et al [14], that for the LVQ algorithm the first part of the integrand in (20) converges to the Bayes classification error when the number of Voronoi vectors tends to infinity. Details of this analysis are outside the scope and size of the present paper. The second part of (20) is clearly the average distortion.

The proof sketched in [13] can be used and extended to establish such a convergence as long as $K_N \to \infty$, $N \to \infty$, with $K_N/N \to 0$, as already mentioned in the introduction to section 2. The convergence of the algorithm is concerned with a sequence of partitions of $\mathbb{R}^d$, or of a compact subset of $\mathbb{R}^d$. The strongest convergence results can be obtained for general probability distributions for (X, d) pairs ((data, class label) pairs) which have compact support in $\mathbb{R}^d$. Let D_N denote the sequence of N training pairs of data $\{(X_i, d_i);\ i = 1, \ldots, N\}$. We generate a sequence of partitions $\{\mathcal{P}(K_N)\}$ each partition utilizing K_N Voronoi vectors, and the associated cells using the general proximity function p. We iteratively pass the training data through the algorithm (6) of updating the Voronoi vectors $\Theta(n, K_N)$ where n is the iteration index. The limit of this sequence as $n \to \infty$, $\Theta^*(K_N)$ provides one member of our family of paritions. We then increase the number of Voronoi sectors to $K_N + 1$ and repeat the process, etc. The general convergence problem for our algorithm, refers to limits of (20), and of $\Theta(n, K_N)$ as $n \to \infty$, $K_N \to \infty$, $N \to \infty$. The most appropriate framework to investigate this general convergence with respect to K_N, N, is the convergence of classification error (in our case it would be combined classification and compression errors) based on Voronoi type partitions, using as starting methods those of chapter 21 (Vapnik-Cervonenkis ideas) of Devroye et al [14], see for instance Theorem 21.5 on page 378 of [14]. In the latter Theorem it is shown that for distributions of x with compact support in $\mathbb{R}^d$, and a majority rule classification based on a Voronoi- type partition with K_N cells and Euclidean proximity function, the classification error converges to the Bayes error with probability one, when $K_N \to \infty$ in such a way that $K_N^2 \log N/N \to 0$ as $N \to \infty$.

Similar results can be obtained for our algorithm, but they are beyond the scope (and space) of the present paper and will be pursued elsewhere. There is also a rich set of related problems regarding general proximity metrics, empirical errors, and computational complexity reductions that could be investigated.

Here we concentrate on the convergence of $\Theta(n, K_N)$ as a function of n, for fixed K_N; this being the first step in the general convergence analysis outlined above. This convergence (w.r.t. n) is the subject of the next section.

2.2 Convergence analysis of the combined compression and classification algorithm.

The convergence analysis for a class of learning vector quantization algorithm was presented in [13] following the analysis in [12] (see Part II- Chapter 1). However, as we noted before, since the algorithm under investigation is a special case of the Robbins-Monro algorithm, where the transition probability function is independent of Θ, we can simplify the set of assumptions needed greatly.

Consider again the algorithm:

$$\Theta(n+1) = \Theta(n) + \epsilon_{n+1} H(\Theta(n), X_{n+1}), \ n \geq 0 \tag{21}$$

Suppose Assumption 2.1 holds. Also, let us make the following additional assumptions that will be sufficient

for our analysis:

Assumption 2.5 *For any compact subset Q of D, there exist constants $\bar{C}_1, r_1$ such that*

$$|H(\Theta, x)| \leq \bar{C}_1(1 + |x|^{r_1}) \tag{22}$$

Remark 2.3 Note that for our choice of $H(\Theta, x)$ described in the previous section, (22) is satisfied if Assumption 2.3 is satisfied.

Assumption 2.6 $h(\Theta) \stackrel{\Delta}{=} (h_1(\Theta), \ldots, h_k(\Theta))'$ *where $h_i(\Theta)$ given by (13) is locally Lipschitz.*

Assumption 2.7 *For any $q \geq 1$, $\exists$ a constant $M < \infty$ such that*

$$\sup_n E\{|X_n|^q 1_{n \leq \nu(\epsilon, Q)}\} \leq M \tag{23}$$

Remark 2.4 Since $\{X_n\}$ is a sequence of *i.i.d.* random vectors, one can simply write (23) as

$$\int_{\mathbf{R}^d} |x|^q \mu(dx) \leq M \ . \tag{24}$$

We present next a theorem that gives an upper bound on the L_q norm of the distance between the actual iterate $\Theta(n)$ and $\Theta(a, t_n)$ which is the solution to (14) for $t = t_n$. In other words, this result gives an upper bound on the quality of approximation by the mean trajectory represented by (14). We do not provide the proof due to space limitations (see [20] for the complete proof).

Theorem 2.1 *Consider the update equation (21) and (14). Suppose Assumptions 2.1, 2.5, 2.6, 2.7 hold. Suppose $Q_1 \subset Q_2$ are compact subsets of D, and $q > 2$. Then there exist constants $B_1(q), \bar{L}_2$ ($\bar{L}_2$ is the Lipschitz constant for h in Q_2), such that for all $T > 0$ (that satisfy the condition that for all $a \in Q_1$, all $t \leq T$, $d(\Theta(a,t), Q_2^c) \geq \delta_0 > 0$), all $\delta < \delta_0$, all $a \in Q_1$,*

$$P_a\{\sup_{n \leq m(0,T)} |\Theta(n) - \Theta(a, t_n)|^q \geq \delta\} \leq$$

$$\frac{B_1(q)}{\delta^q}(1+T)^{q-1}\exp(q\bar{L}_2 T) \sum_{i=1}^{m(0,T)} \epsilon_i^{1+\frac{q}{2}} \ . \tag{25}$$

We now present an asymptotic result without proof that states that $\Theta(n)$ asymptotically converges to a compact subset of D, based on the assumption that the mean ODE has a point of asymptotic stability Θ^* in D with domain of attraction D. We need the following additional assumptions:

Assumption 2.8 *There exists α such that $\sum \epsilon_n^\alpha < \infty$.*

Assumption 2.9 *There exists a positive function U of class C^2 on D such that $U(\Theta) \to C \leq \infty$ if $\Theta \to \partial D$ or $|\Theta| \to \infty$ and $U(\Theta) < C$ for $\Theta \in D$ satisfying*

$$\langle U'(\Theta), h(\Theta) \rangle \leq 0, \ \forall \Theta \in D \ . \tag{26}$$

Remark 2.5 Note that if there is such a point Θ^* in D which is a point of asymptotic stability for the mean ODE (14) with domain of attraction D, this means that any solution of (14) for $a \in D$ indefinitely remains in D and converges to Θ^* as $t \to \infty$. It can then be shown that (see [16], Th. 5.3, p.31) there exists a function $U(\Theta)$ which satisfies the conditions mentioned in Assumption 2.9.

We use the following notation:

$$\begin{aligned} K(c) &= \{\Theta;\ U(\Theta) \leq c\} \\ \tau(c) &= \inf(n;\ \Theta(n) \notin K(c)) \\ q_0(\alpha) &= \sup(2, 2(\alpha - 1)) \end{aligned} \tag{27}$$

With these notations and assumptions, we have established the following theorem (with arguments similar to those in [12], pp. 301-304):

Theorem 2.2 *Consider (21). Suppose Assumptions 2.1, 2.5, 2.6, 2.7, 2.8, 2.9 hold and suppose that F is a compact set such that*

$$F = \{\Theta;\ U(\Theta) \leq c_0\} \supset \{\Theta;\ U'(\Theta).h(\Theta) = 0\}$$

for some $c_0 < C$ where C is defined in Assumption 2.9. Then, for any compact subset Q of D, and $q \geq q_0(\alpha)$, there exists a constant $B_2(q)$ such that for all $a \in Q$:

$$P_a(\Theta(n) \text{ converges to } F) \geq 1 - B_2(q) \sum_{i \geq 1} \epsilon_i^{1+\frac{q}{2}} \tag{28}$$

3 Conclusions and future research

We have developed an algorithm based on learning vector quantization (LVQ) for combined compression and classification. We have shown convergence of the algorithm for fixed numer of Voronoi vectors, under reasonable conditions, using the ODE method of stochastic approximation. Examples illustrating the performance of the algorithm can be found in [20]. The sensitivity of the performance of the algorithm with respect to the

weight parameter λ indicates that the compression error decreases with increasing λ whereas the increase in classification error is relatively insignificant.

The immediate future research problem is to establish convergence of the algorithm as N and $K_N \to \infty$, and related performance evaluation problems as described at the end of Section 2.1. Another important future research problem that we are currently working on is the extension of the algorithm when the VQ is replaced by TSVQ. In this extension, we use and extend the methods and analysis of [19]. With this extension, we will be able to treat the performance of the WTSVQ algorithm of [4] [5] [6], [7] analytically including compression of the wavelet coefficients.

References

[1] E. Frantzeskakis, *On Image Coding and Understanding: A Bayesian Formulation for the Problem of Template Matching Based on Coded Image Data.* 1990. M. S. Thesis, ISR Technical Report MS 90-5.

[2] K. O. Perlmutter, S. Perlmutter, R. Gray, R. Olsen, and K. L. Oehler, "Bayes risk weighted vector quantization with posterior estimation for image compression and classification." preprint, 1997.

[3] K. L. Oehler and R. M. Gray, "Combining image compression and classification using vector quantization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, pp. 461–473, May 1995.

[4] J. Baras and S. Wolk, "Model based automatic target recognition from high range resolution radar returns," in *Proceedings of SPIE International Symposium on Intelligent Information Systems*, vol. 2234, (Orlando, FL), pp. 57–66, April 1994.

[5] J. Baras and S. Wolk, "Wavelet based progressive classification of high range resolution radar returns," in *Proceedings of SPIE International Symposium on Intelligent Information Systems*, vol. 2242, (Orlando, FL), pp. 967–977, April 1994.

[6] J. Baras and S. Wolk, "Wavelet based progressive classification with learning: Applications to radar signals," in *Proceedings of the SPIE 1995 International Symposium on Aerospace/ Defense Sensing and Dual-Use Photonics*, vol. 2491, (Orlando, FL), pp. 339–350, April 1995.

[7] J. Baras and S. Wolk, "Wavelet-based hierarchical organization of large image databases: ISAR and face recognition," in *Proceedings of SPIE 12th International Symposium on Aerospace, Defense Sensing, Simulation and Control*, vol. 3391, (Orlando, FL), pp. 546–558, April 1998.

[8] J. Baras and D. MacEnany, "Model-based ATR: Algorithms based on reduced target models, learning and probing," in *Proceedings of the Second ATR Systems and Technology Conference*, vol. 1, pp. 277–300, February 1992.

[9] D. MacEnany and J. Baras, "Scale-space polygonalization of target silhouettes and applications to model-based ATR," in *Proceedings of the Second ATR Systems and Technology Conference*, vol. 2, pp. 223–247, February 1992.

[10] A. LaVigna, *Nonparametric Classification Using Learning Vector Quantization.* PhD thesis, Dept. of Electr. Eng., University of Maryland, College Park, Maryland 20742, 1989. ISR Technical Report PhD 90-1.

[11] T. Kohonen, *Self-Organizing Maps.* Heidelberg, Germany: Springer-Verlag, 1995.

[12] A. Benveniste, M. Metivier, and P. Priouret, *Adaptive Algorithms and Stochastic Approximation.* Berlin and New York: Springer-Verlag, 1990.

[13] J. S. Baras and A. LaVigna, "Convergence of a neural network classifier," in *Proc. of 29th IEEE Conf. on Decision and Control*, pp. 1735–1740, December 1990.

[14] L. Devroye, L. Gyorfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition.* Springer-Verlag, 1996.

[15] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Transactions on Communications*, vol. 28, pp. 84–95, 1980.

[16] Krasovskii, *Stability of Motion.* Stanford University Press, 1963.

[17] H. Gish and M. Schmidt, "Text-independent speaker identification," *IEEE Signal Processing Magazine*, pp. 18–32, October 1994.

[18] D. A. Reynolds, "Speaker identification and verification using gaussian mixture speaker models," *Speech Communication*, vol. 17, pp. 91–108, 1995.

[19] A. B. Nobel and R. A. Olshen, "Termination and continuity of greedy growing for tree-structured vector quantizers," *IEEE Transactions on Information Theory*, vol. 42, January 1996.

[20] J. Baras and S. Dey, "Combined Compression and Classification with Learning Vector Quantization," *IEEE Transactions on Information Theory*, vol. 45, No. 6, pp. 1911-1920, September 1999.