Combined Compression and Classification with Learning Vector Quantization

John S. Baras, *Fellow, IEEE*, and Subhrakanti Dey, *Member, IEEE*

***Abstract*— Combined compression and classification problems are becoming increasingly important in many applications with large amounts of sensory data and large sets of classes. These applications range from automatic target recognition (ATR) to medical diagnosis, speech recognition, and fault detection and identification in manufacturing systems. In this paper, we develop and analyze a learning vector quantization (LVQ) based algorithm for combined compression and classification. We show convergence of the algorithm using the ODE method from stochastic approximation. We illustrate the performance of our algorithm with some examples.**

***Index Terms*— Classification, compression, learning vector quantization, nonparametric, stochastic approximation.**

I. Introduction

QUITE often in applications, we are faced with the problem of classifying signals (or objects) from vast amounts of noisy data. The number of different distinct signals (classes) may be quite large. Compressing each observation (observed signal) while retaining significant signal features presents two significant advantages.

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. This area has attracted recently more interest due to the increased number of applications requiring such algorithms. 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.

Manuscript received July 10, 1998; revised January 19, 1999. This work was supported by ONR under Contract 01-5-28834 under the MURI Center for Auditory and Acoustics Research, by NSF under Grant 01-5-23422, and by the Lockheed Martin Chair in Systems Engineering.

J. S. Baras is with the Department of Electrical and Computer Engineering and the Institute for Systems Research, University of Maryland, College Park, MD 20742 USA.

S. Dey was with the Institute for Systems Research, University of Maryland, College Park, MD 20742 USA. He is now with the Department of Systems Engineering, RSISE, Australian National University, Canberra, ACT 0200, Australia.

Communicated by P. Moulin, Associate Editor for Nonparametric Estimation, Classification, and Neural Networks.

Publisher Item Identifier S 0018-9448(99)06068-X.

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 signal representations at first to perform many simple (and, therefore, fast) matching tests, and then progressively performs fewer but more complex (and, therefore, slower) matching tests, as needed for classification. Thus compression becomes an indispensable component in such schemes, and in particular variable-rate (and, therefore, resolution) compression. 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]–[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 toward 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 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] and [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] and [3]). However, our algorithm is designed to obtain a systematic tradeoff between compression and classification performance 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 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 II, we describe the LVQ-based algorithm for combined compression and classification. In Sections II-A and II-B, we provide analysis and convergence of the algorithm using stochastic approximation techniques and the so-called ODE method. In Section III, we provide simulation results of the performance of the algorithm for some typical problems. Section IV presents some concluding remarks.

II. Combined Compression and Classification with 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 toward 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 $\{(X_i, d_{X_i}), i = 1, \cdots, N\}$, 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 $\lim_{N\to\infty} (K_N/N) \to 0$. 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 [14, Theorem 21.2, p. 368] and [14, Theorem 21.5, p. 379]. We will discuss the second theorem in Section II-A 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 $X_l \in \mathbb{R}^d$, $l = 1, 2, \cdots, N$, 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

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

for any $B \subset \mathcal{F}$. We also assume that X_l is independent of $X_j, j \neq l$.

The Voronoi vectors are represented by $\theta_i \in \mathbb{R}^d$, $i = 1, 2, \cdots, K$ 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 nonincreasing sequence of positive real numbers ϵ_n, $n = 1, 2, \cdots$, 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 is convex in θ for every fixed $x \in \mathbb{R}^d$.

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 $\theta \in Q$

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

In Assumptions 2.2–2.4, $|\cdot|$ 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 $\rho(\theta, x) = |\theta - x|^2$. In our implementation of the algorithm, we use this function although for the sake of generality in the analysis, we would refer to it in its general form $\rho(\theta, x)$.

Define further the following quantities.

Definition 2.1:

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

where $\Theta(n) = (\theta_1(n), \cdots, \theta_K(n))'$ and $\theta_i(n)$ is the nth 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:

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

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

With the above definitions and assumptions, we can now write the following multipass 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., the LBG [15] algorithm over the set of training vectors.
2. $n = 0$.
3. *Assigning the training vectors to their respective cells:* Find $i_l = \operatorname{argmin}_m |\theta_m(n) - X_l|^2$, $l = 1, 2, \cdots, N$, then X_l belongs to $V_{\theta_{i_l}(n)}$.
4. *Cell decisions:* Calculate $g_i(\Theta(n); N)$, $i = 1, 2, \cdots, K$.
5. *Updating the Voronoi vectors:* For $i \in \{1, 2, \cdots, K\}$

$$\begin{aligned}\theta_i(n+1) = \theta_i(n) + \epsilon_{n+1}(&-\lambda 1_{X_{n+1} \in V_{\theta_i(n)}} \\ &+ \gamma(d_{X_{n+1}}, g_i(\Theta(n); N), X_{n+1}, \Theta(n))) \\ &\cdot \nabla_\theta \rho(\theta, X_{n+1})|_{\theta = \theta_i(n)}.\end{aligned} \tag{5}$$

6. $n \leftarrow n + 1$.
7. If $n < N$, repeat Steps 3–6. If $n = N$, repeat Steps 3 and 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 mth 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

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

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

Remark 2.3: Note that for $\lambda = 0$, the above algorithm becomes the modified LVQ algorithm resulting in better convergence properties as reported in [13].

A. Analysis of the Combined Compression and Classification Algorithm

In this subsection, we present the analysis of the above algorithm using the "mean ODE" method of [12].

Denote the vectors

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

and

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

where

$$\begin{aligned}H_i(\Theta(n), X_{n+1}) = \big(&-\lambda 1_{X_{n+1} \in V_{\theta_i(n)}} \\ &+ \gamma(d_{X_{n+1}}, g_i(\Theta(n); N), X_{n+1}, \Theta(n))\big) \\ &\cdot \nabla_\theta \rho(\theta, X_{n+1})|_{\theta=\theta_i(n)}\end{aligned} \tag{7}$$

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

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

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

Define

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

Due to the assumption that $\{X_l\}$, $l = 1, 2, \cdots$, is a sequence of independent and identically distributed (i.i.d.)

random vectors and that are distributed independently of $\Theta(l)$, the transition probability function

$$\Pi_{\Theta(n)}(A, X_n) \triangleq P(X_{n+1} \in A | \mathcal{F}_n)$$

is given by

$$\mu(A) = \int_A p(x)\, dx,$$

where

$$\mathcal{F}_n \triangleq \sigma\{\Theta(0), X_0, \cdots, \Theta(n), X_n\}$$

(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:

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

Remark 2.4: Note that $\gamma_i(\Theta(n); N) = 1$ if $g_i(\Theta(n); N) = 2$, and -1 otherwise.

Definition 2.4:

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

One can now prove the following Lemma.

Lemma 2.1:

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

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

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

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

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

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 $P(X_{n+1} \in A | \mathcal{F}_n)$ is independent of $\Theta(k)$, $k \leq n$.

It is hard to establish a convergence result for general $h(\Theta)$. 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 [12, Theorem 4, p. 45]). Here, we have used Assumption 2.1.

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

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

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

However, there is no general theory which gives conditions under which $P(\Theta(n) \in Q \text{ infinitely often}) = 1$ is satisfied [12].

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 $J(\cdot)$ and their domains of attraction. Although we refrain from such pursuits for the time being, we do notice that (see [13]) as $N \to \infty$, $\overline{\gamma}_i(\Theta; N) \to \operatorname{sign}(\int_{V_{\theta_i}} q(x)\, dx)$. Using the mean value theorem when the size of each Voronoi cell is small, one can write

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

which is the negative gradient of the cost function

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

For those readers who are more oriented toward 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 Devroye *et al*. [14, Ch. 21], that for the LVQ algorithm the first part of (20) converges to the Bayes classification error when the number of Voronoi vectors tends to infinity. Details of this analysis are outside the scope 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 II. 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 $\{(X_i, d_i); i = 1, \cdots, N\}$. 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) which updates 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 partitions. 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 Devroye *et al*. [14, Ch. 21] (Vapnik–Chervonenkis ideas); see, for example, [14, Theorem 21.5, p. 378]. 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 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 with respect to n is the subject of the next subsection.

B. 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 Pt. II—Ch. 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 greatly simplify the set of assumptions needed. In particular, the assumptions described as A.4 in [12, p. 216], become trivial and follow from the single assumption that $h(\Theta)$ is locally Lipschitz. In this subsection, we obtain an upper bound on the L_q estimate of a "fluctuation" term to be introduced shortly, for $q > 2$. We will provide a simpler local bound later on for $q = 2$.

Consider again the algorithm

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

Before we introduce the set of assumptions needed for the analysis of our algorithm, let us introduce the following notation:

Notation 2.1:

1) D is an open subset of $\mathbb{R}^d$. Q is a compact subset of D.
2) ϕ is a C^2 function from $\mathbb{R}^d$ to $\mathbb{R}$ with bounded second derivatives, where

$$\begin{aligned} M_0(Q) &= \sup_{\Theta \in Q} |\phi(\Theta)| \\ M_1(Q) &= \sup_{\Theta \in Q} |\phi'(\Theta)| \\ M_2(Q) &= \sup_{\Theta \in Q} |\phi''(\Theta)| \\ M_2 &= \sup_{\Theta \in \mathbb{R}^d} |\phi''.(\Theta)|. \end{aligned} \quad (22)$$

3) There exists $R(\phi, \Theta, \Theta')$ such that

$$\begin{aligned} R(\phi, \Theta, \Theta') &= \phi(\Theta') - \phi(\Theta) - \langle (\Theta' - \Theta), \phi'(\Theta) \rangle \\ |R(\phi, \Theta, \Theta')| &\leq M_2 |\Theta' - \Theta|^2, \qquad \forall \Theta, \Theta' \in \mathbb{R}^d. \end{aligned} \quad (23)$$

4)

$$e_n(\phi) = \phi(\Theta(n+1)) - \phi(\Theta(n)) - \epsilon_{n+1} \langle \phi'(\Theta(n)), h(\Theta(n)) \rangle. \quad (24)$$

5) For $\varepsilon > 0$

$$\begin{aligned} \tau(Q) &= \inf (n; \Theta(n) \notin Q) \\ \sigma(\varepsilon) &= \inf (n \geq 1; |\Theta(n) - \Theta(n-1)| > \epsilon) \\ \nu(\varepsilon, Q) &= \inf (\tau(Q), \sigma(\epsilon)). \end{aligned} \quad (25)$$

6) With $t_0 = 0, t_n = \sum_{i=1}^{n} \epsilon_i$, we define

$$m(n, T) \triangleq \inf \left\{ k \colon k \geq n, \sum_{i=n}^{k} \epsilon_{i+1} \geq T \right\}.$$

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 $\overline{C}_1, r_1$ such that

$$|H(\Theta, x)| \leq \overline{C}_1 (1 + |x|^{r_1}). \quad (26)$$

Remark 2.5: Note that for our choice of $H(\Theta, x)$ described in the previous section, (26) is satisfied if Assumption 2.3 is satisfied.

Assumption 2.6: $h(\Theta) \triangleq (h_1(\Theta), \cdots, h_k(\Theta))'$ where $h_i(\Theta)$ given by (13) is locally Lipschitz.

Remark 2.6: Note that this assumption itself is enough for our analysis and we do not need the assumptions made in [13] following [12] ([12, Assumption A.4, p. 216]) since they trivially follow from Assumption 2.6.

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

$$\sup_n E\{|X_n|^q 1_{n \leq \nu(\varepsilon, Q)}\} \leq M. \quad (27)$$

Remark 2.7: Since $\{X_n\}$ is a sequence of i.i.d. random vectors, one can simply write (27) as

$$\int_{\mathbb{R}^d} |x|^q \mu(dx) \leq M. \quad (28)$$

Remark 2.8: One can, in fact, deduce from Assumptions 2.5 and 2.7 that under certain other restrictions on the distribution function $\mu(dx)$, that Assumption 2.6 holds, since in this case $\mu(dx)$ is independent of Θ (see [12, Sec. II-B.6, pp. 264–265]).

We can now present the following theorem, whose proof is given in the Appendix.

Theorem 2.1: Consider the update equation (21). Consider also (24) and (25). Suppose Assumptions 2.1, 2.5–2.7 hold. Then, for any regular function ϕ with bounded second derivatives satisfying (22), any compact subset Q of D, and for all $q>2$ there exist constants $B(q)$, M_4, $\varepsilon_0>0$ such that for all $\varepsilon<\varepsilon_0$, $T>0$, $a\in D$, we have

$$E_a\left\{\sup_{n<k\le m(n,T)} 1_{k\le\nu(\varepsilon,Q)}\left|\sum_{i=n}^{k-1} c_i(\phi)\right|^q\right\} \le B(q)M_1(Q)T^{(q/2)-1}\sum_{i=n+1}^{m(n,T)}\epsilon_{i+1}^{1+(q/2)} + M_4(q)T^{q-1}\sum_{i=n+1}^{m(n,T)}\epsilon_{i+1}^{1+q}. \tag{29}$$

Next, we present 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 since the result holds under the same set of assumptions as the previous theorem, and the proof can be found in [12, p. 301].

Theorem 2.2: Consider the update equation (21) and (14). Suppose Assumptions 2.1, 2.5–2.7 hold. Suppose $Q_1\subset Q_2$ are compact subsets of D, and $q>2$. Then there exist constants $B_1(q)$, $\overline{L}_2$ ($\overline{L}_2$ 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 $t\le T$, $d(\Theta(a,t),Q_2^c)\ge\delta_0>0$), all $\delta<\delta_0$, all $a\in Q_1$

$$P_a\left\{\sup_{n\le m(0,T)}|\Theta(n)-\Theta(a,t_n)|^q\ge\delta\right\} \le \frac{B_1(q)}{\delta^q}(1+T)^{q-1}\exp(q\overline{L}_2T)\sum_{i=1}^{m(0,T)}\epsilon_i^{1+(q/2)}. \tag{30}$$

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 make more precise statements later. First, we introduce the following additional assumptions and notations.

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 $U(\Theta)\to C\le\infty$ if $\Theta\to\partial D$ or $|\Theta|\to\infty$ and $U(\Theta)<C$ for $\Theta\in D$ satisfying

$$\langle U'(\Theta),h(\Theta)\rangle\le 0,\qquad\forall\Theta\in D. \tag{31}$$

Remark 2.9: 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, Theorem 5.3, p. 31]) there exists a function $U(\Theta)$ which satisfies the conditions mentioned in Assumption 2.9.

Notation 2.2:

$$\begin{aligned} K(c)&=\{\Theta;U(\Theta)\le c\}\\ \tau(c)&=\inf(n;\Theta(n)\notin K(c))\\ q_0(\alpha)&=\sup(2,2(\alpha-1)). \end{aligned} \tag{32}$$

With these notations and assumptions, we can present the following theorem (for a proof see [12, pp. 301–304]).

Theorem 2.3: Consider (21). Suppose Assumptions 2.1, 2.5–2.9 hold and suppose that F is a compact set such that

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

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

$$P_a(\Theta(n)\text{ converges to }F)\ge 1-B_2(q)\sum_{i\ge1}\epsilon_i^{1+(q/2)}. \tag{33}$$

In the next subsection, we provide a simpler local bound for $q=2$, following the analysis given in [12, Pt. II, Sec, V-A].

C. A Simpler Local Bound for $q=2$

Consider again the algorithm

$$\Theta(n+1)=\Theta(n)+\epsilon_{n+1}H(\Theta(n),X_{n+1}),\qquad n\ge0. \tag{34}$$

Since X_n, $n\ge0$ are distributed independently of $\Theta(n)$ and also $\{X_n\}$, $n\ge0$ is a sequence of i.i.d. random variables, we have the main or so-called Robbins–Monro assumption satisfied, namely,

$$E[g(\Theta(n),X_{n+1})|\mathcal{F}_n]=\int_{\mathbb{R}^d}g(\Theta(n),x)p(x)\,dx. \tag{35}$$

Note that we have already observed before in Lemma 2.1 that

$$\begin{aligned} h(\Theta(n))&=E_a[H(\Theta(n),X_{n+1})|\mathcal{F}_n]\\ &=\int_{\mathbb{R}^d}H(\Theta(n),x)p(x)\,dx. \end{aligned} \tag{36}$$

Next, we introduce the two main assumptions of this section.

Assumption 2.10: For all $\Theta(0)=a\in\mathbb{R}^d$

$$E_a[|H(\Theta(n),X_{n+1})|^2|\mathcal{F}_n]\le\tilde{C}_1(1+|\Theta(n)|^2) \tag{37}$$

for some suitable constant $\tilde{C}_1$.

Remark 2.10: Note that this assumption guarantees the existence of $h(\Theta(n))$.

Assumption 2.11: $\exists\Theta^*$ (which is a point of asymptotic stability of (14)) such that for all Θ, $\exists$ a constant $\delta>0$ such that

$$(\Theta-\Theta^*)'h(\Theta)\le-\delta|\Theta-\Theta^*|^2 \tag{38}$$

with, for some $\beta\le 1$,

$$\liminf_{n\to\infty} 2\delta\frac{\epsilon_n^\beta}{\epsilon_{n+1}}+\frac{\epsilon_{n+1}^\beta-\epsilon_n^\beta}{\epsilon_{n+1}^2}>0. \tag{39}$$

Remark 2.11: Note that if $\epsilon_n=(A_1/n^\alpha+A_2), 0\le\alpha\le 1$, then (39) holds for all $\beta<1$. It is also true for $\beta=1$ if $2\delta>(\alpha/A_1)$.

We can now present the following theorem which gives a simple local bound for the expected distance between $\Theta(n)$ and Θ^*.

Theorem 2.4: Consider (34). Suppose Assumptions 2.10 and 2.11 hold. Then

$$E_a(|\Theta(n)-\Theta^*|^2)\le B_{\bar{5}}(a)\epsilon_n^\beta \tag{40}$$

for some suitable constant $B_{\bar{5}}(a)$.

Proof: It is sufficient to show that for some suitable n_0, there exists a $B_{\bar{5}}(a,n_0)$ such that for all $n\ge n_0$

$$E_a(|\Theta(n)-\Theta^*|^2)\le B_{\bar{5}}(a,n_0)\epsilon_n^\beta. \tag{41}$$

Writing $J_n\triangleq\Theta(n)-\Theta^*$, we have

$$\begin{aligned}E_a(|J_{n+1}|^2|\mathcal{F}_n)=&|J_n|^2+2\epsilon_{n+1}\langle J_n,h(\Theta(n)\rangle\\&+\epsilon_{n+1}^2E_a[|H(\Theta(n),X_{n+1})|^2|\mathcal{F}_n].\end{aligned} \tag{42}$$

Suppose that n is sufficiently large such that $1\ge 2\epsilon_{n+1}\delta$. Then, by taking expectations, we have

$$E_a|J_{n+1}|^2\le(1-2\epsilon_{n+1}\delta+\hat{C}_1\epsilon_{n+1}^2)E_a|J_n|^2+\hat{C}_1\epsilon_{n+1}^2 \tag{43}$$

where $\hat{C}_1$ is a constant such that

$$\tilde{C}_1(1+|\Theta|^2)\le\hat{C}_1(1+|\Theta-\Theta^*|^2). \tag{44}$$

Now, one can use the following result which can be proved directly from (39). There exist B^0 and n_0 such that for all $B_{\bar{5}}\ge B^0$ and $n\ge n_0$, the sequence $u_n=B_{\bar{5}}\epsilon_n^\beta$ satisfies

$$u_{n+1}\ge(1-2\epsilon_{n+1}\delta+\hat{C}_1\epsilon_{n+1}^2)u_n+\hat{C}_1\epsilon_{n+1}^2. \tag{45}$$

Choose $B_{\bar{5}}(a,n_0)\ge B^0$ such that

$$E_a|J_{n_0}|^2\le B_{\bar{5}}(a,n_0)\epsilon_{n_0}^\beta.$$

It follows immediately by induction on n that the sequence $u_n=B_{\bar{5}}(a,n_0)\epsilon_n^\beta$, $n\ge n_0$ satisfies $E_a|J_n|^2\le u_n$ from which (40) follows. □

III. Simulation Studies

In this section, we present some simulation results illustrating the compression performance of our algorithm while a tradeoff is obtained with respect to its classification performance. We consider two examples, one with computer-simulated data distributed according to either of two bimodal Gaussian densities and the other with "mel-cepstral" coefficients of two female speakers obtained from their speech.

A. Bimodal Gaussian Data

This part of the simulation study is carried out with computer-generated random numbers distributed according to either of two two-dimensional bimodal Gaussian mixture distributions. The first pattern is generated from the bimodal Gaussian mixture density

$$0.5N([1.0\ 1.0]',I)+0.5N([-1.0\ -1.0]',I)$$

where $N([m_1\ m_2]',\Sigma)$ is the two-dimensional normal distribution function with the mean vector $[m_1\ m_2]'$ and covariance matrix Σ. The second pattern is generated from the density

$$0.4N([0.0\ 0.0]',4I)+0.6N([0.5\ 0.5]',4I).$$

The training set was formed by 500 vectors from each pattern (meaning $\pi_1=\pi_2=0.5$). This set was used to train the Voronoi vectors in multiple passes, the total number of passes being 20. The number of Voronoi vectors that would result in a good classification performance was found by increasing the number of Voronoi vectors by one until the classification performance (for a given size of test data set) reached a floor. Thus 16 Voronoi cells were chosen and their centroids initialized by the output of an LBG algorithm processing the training data. Each test data set had a size of 1000 each containing vectors from patterns 1 and 2 such that the *a priori* probabilities were satisfied. The learning rate ϵ_n was kept fixed over one pass such that $\epsilon_p=(\epsilon_1/\sqrt{p})$ where p denotes the number of the pass, and $\epsilon_1=0.01$. The compression performance averaged over ten test data sets for a range of $\lambda\in[0.0,5.0]$ is given in Fig. 1. The compression error was measured by the minimum mean-square error, that is, the average of the squared distances between the test vectors and their representative Voronoi vectors and normalized with respect to the compression error achieved by the pure LVQ algorithm ($\lambda=0.0$). More explicitly, if $(\theta_1^*(\lambda),\cdots,\theta_K^*(\lambda))$ are the centroids after the combined algorithm has converged for a specific value of λ, then the unnormalized compression error is expressed as

$$E_\lambda\triangleq\frac{1}{N_T}\sum_{i=1}^{N_T}\min_{1\le j\le K}||X_i-\theta_j^*(\lambda)||^2$$

whereas the normalized compression error is given by E_λ/E_0. Obviously, E_0 is the unnormalized compression error for the pure LVQ algorithm ($\lambda=0$). Here, N_T is the number of test vectors.

It is seen that as λ increases up to 5.0, there is a reduction of approximately 3.5% in the normalized compression error. The classification performance measured by the percentage of misclassified data did not change very much with increasing value of λ and tended to hover around 30% in the range of λ as mentioned above. Hence we do not include a separate plot for the classification performance.

B. Mel-Cepstral Coefficients of Two Speakers

This example is based on "mel-cepstrum" coefficients of two female speakers. "Mel-cepstrum" features based on the nonlinear human perception of the frequency of sounds have been

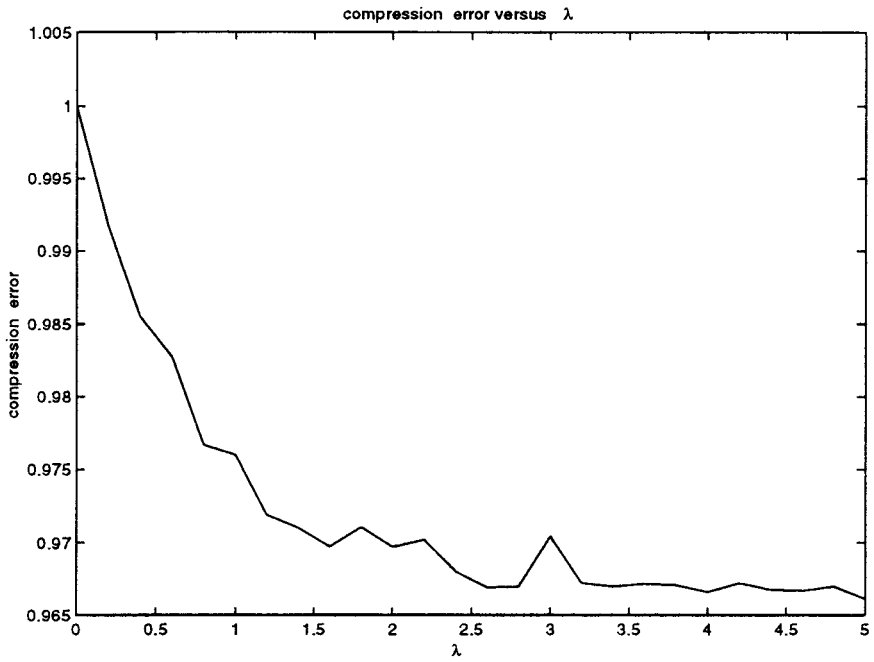

Fig. 1. Compression error perfomance of the combined LVQ algorithm for bimodal Gaussian patterns.

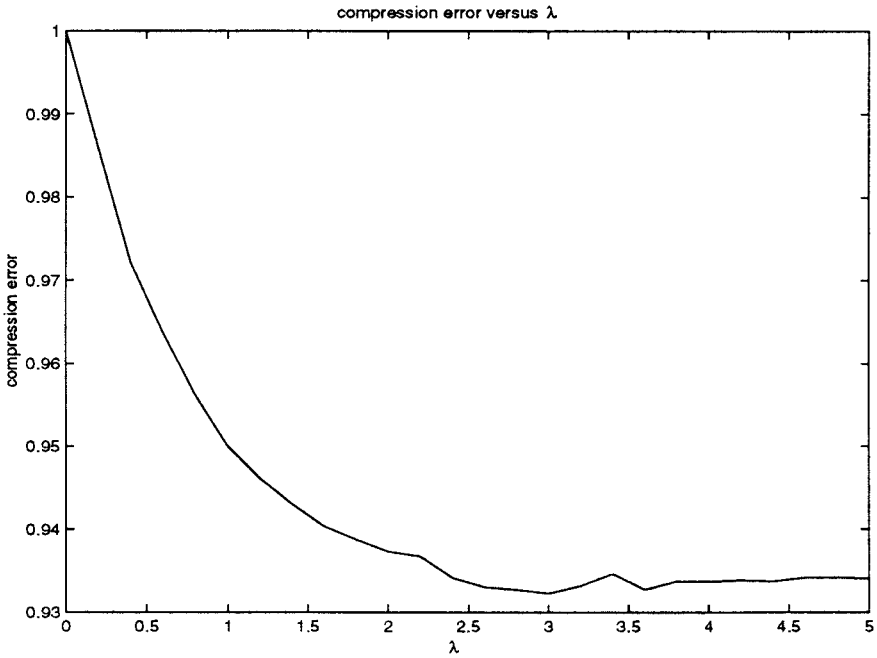

Fig. 2. Compression error performance of the combined LVQ algorithm for "mel-cepstrum" features of female speakers.

well studied and successfully applied to speaker identification problems. These studies have shown that the mel-cepstrum can effectively extract the vocal tract shape information of the speakers and yield good distinguishing performance [17], [18]. In our example, the labeled phonetic speech data of the two female speakers are extracted from the TIMIT database for dialect region 2. The speech waveform is segmented into 16-ms frames overlapped by 8 ms and parameterized to 14-dimensional mel-cepstrum vectors to establish the feature space.

Since the performance of an LVQ-type algorithm depends critically on the number of Voronoi vectors and the number of training vectors per Voronoi cell, to achieve a tradeoff with the computational time required, the following parameters were chosen. The training set was randomly chosen to have 500 data vectors from each speaker. The number of Voronoi cells was chosen to be 20. The training set was used to update the Voronoi vectors in multiple passes, the total number of passes being 30. The learning rate ϵ_n was taken to be constant over one pass where $\epsilon_p = (\epsilon_1/\sqrt{p})$ where p denotes the number of passes with $\epsilon_1 = 0.04$. The Voronoi vectors were initialized by passing the training set through an LBG algorithm. Once the training was completed, five sets of test data, each containing 250 vectors taken randomly from

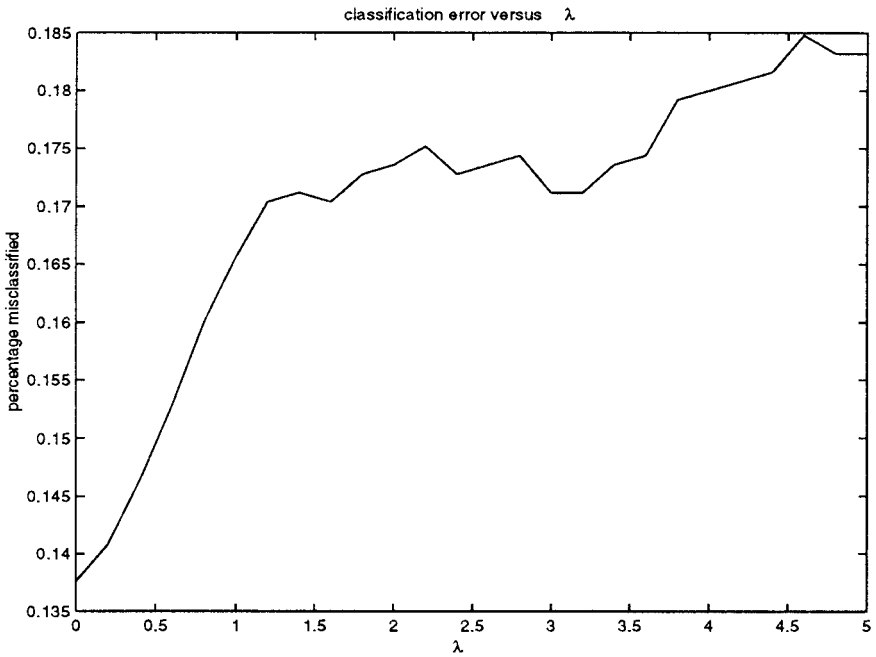

Fig. 3. Classification error performance of the combined LVQ algorithm for "mel-cepstrum" features of female speakers.

the database for both speakers, were used to obtain the compression and classification performances of our algorithm. Figs. 2 and 3 illustrate the results averaged over five test data sets, for a range of $\lambda \in [0.0, 5.0]$. As expected, the compression error (measured by the mean-square distance between the data and its representative Voronoi vector), which was normalized with respect to the error obtained by the pure LVQ algorithm ($\lambda = 0.0$), decreases by approximately 7%, whereas the classification error goes up by 4.5%. We would like to comment here that the classification error can be further reduced by choosing a larger number of Voronoi cells which would obviously require a larger number of training vectors.

IV. 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 a fixed numer of Voronoi vectors, under reasonable conditions, using the ODE method of stochastic approximation. We have also illustrated the performance of the algorithm with some examples. 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 II-A. 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]. This will allow us to evaluate the performance of the WTSVQ algorithm of [4]–[7] analytically, including compression of the wavelet coefficients.

Appendix

Proof of Theorem 2.1: In this proof, $C_1(q)$, $C_2(q)$, $C_3(q)$, $C_4(q)$, $B(q)$, and $M_4(q)$ denote constants dependent only on q. From (24), (23), and (21), one can write

$$\begin{aligned} e_k(\phi) &= \epsilon_{k+1}\langle \phi'(\Theta(k)), (H(\Theta(k), X_{k+1}) - h(\Theta(k)))\rangle \\ &\quad + R(\phi, \Theta(k), \Theta(k+1)) \\ &= e_k^{(1)} + e_k^{(2)} \end{aligned} \tag{46}$$

where

$$\begin{aligned} e_k^{(1)} &= \epsilon_{k+1}\langle \phi'(\Theta(k)), (H(\Theta(k), X_{k+1}) - h(\Theta(k)))\rangle \\ e_k^{(2)} &= R(\phi, \Theta(k), \Theta(k+1)). \end{aligned}$$

Note that we have

$$\begin{aligned} \left|\sum_{i=n}^{k-1} e_i(\phi)\right|^q &= \left|\sum_{i=n}^{k-1} e_i^{(1)} + \sum_{i=n}^{k-1} e_i^{(2)}\right|^q \\ &\le \left[\left|\sum_{i=n}^{k-1} e_i^{(1)}\right| + \left|\sum_{i=n}^{k-1} e_i^{(2)}\right|\right]^q \\ &\le 2^{q-1}\left[\left|\sum_{i=n}^{k-1} e_i^{(1)}\right|^q + \left|\sum_{i=n}^{k-1} e_i^{(2)}\right|^q\right]. \end{aligned} \tag{47}$$

From now on, we write m for $m(n,T)$ and ν for $\nu(\varepsilon, Q)$ for notational simplicity. We write

$$\begin{aligned} S_1 &= E\left\{\sup_{n<k\le m} 1_{k\le\nu}\left|\sum_{i=n}^{k-1} e_i^{(1)}\right|^q\right\} \\ &= E\left\{\sup_{n<k\le m}\left|\sum_{i=n}^{k-1} U_i\right|^q\right\} \end{aligned}$$

where

$$U_i = \epsilon_{i+1}\langle \phi'(\Theta(i)), (H(\Theta(i), X_{i+1}) - h(\Theta(i)))\rangle 1_{i+1\le\nu}.$$

Denoting

$$V_i = \langle \phi'(\Theta(i)), (H(\Theta(i), X_{i+1}) - h(\Theta(i))) \rangle 1_{i+1 \le \nu}$$

we have $U_i = \epsilon_{i+1} V_i$.

We notice that from (13), $E(U_{i+1}|\mathcal{G}_i) = 0$.

We also observe that from (22)

$$\begin{aligned} E|V_i|^q &\le M_1(Q) C_1(q) [E|H(\Theta(i), X_{i+1})|^q \\ &\quad + E|h(\Theta(i))|^q] \\ &\le M_1(Q) C_2(q) E|H(\Theta(i), X_{i+1})|^q. \end{aligned} \tag{48}$$

The last inequality follows from Jensen's inequality and (13).

One can now use Assumptions 2.5 and 2.7 to obtain the following upper bound:

$$E|V_i|^q \le M_1(Q) C_3(q). \tag{49}$$

One can now apply Burkholder's inequality (see [12, Lemma 6, p. 294]) to obtain

$$S_1 \le C_4(q) E \left(\sum_{i=n}^{m-1} \epsilon_{i+1}^2 V_i^2 \right)^{q/2}. \tag{50}$$

For $q > 2$, one can further apply a result based on Holder's inequality (see [12, Lemma 7, p. 294]) to obtain

$$\begin{aligned} S_1 &\le C_4(q) \left(\sum_{i=n}^{m-1} \epsilon_{i+1} \right)^{(q/2)-1} \sum_{i=n}^{m-1} \epsilon_{i+1}^{1+(q/2)} E|V_i|^q \\ &\le B(q) M_1(Q) T^{(q/2)-1} \sum_{i=n}^{m-1} \epsilon_{i+1}^{1+(q/2)}. \end{aligned} \tag{51}$$

We prove the following bound on

$$S_2 = E \left\{ \sum_{i=n}^{m-1} \left| e_i^{(2)} \right|^q 1_{i+1 \le \nu} \right\}^q$$

using (23), (26), and Assumption 2.7

$$S_2 \le M_4(q) T^{q-1} \sum_{i=n}^{m-1} \epsilon_{i+1}^{1+q}. \tag{52}$$

Combining (51) and (52), we obtain (29) from (47).

ACKNOWLEDGMENT

The authors wish to thank an annonymous referee for the careful and constructive suggestions made and for suggesting [14] as a useful reference for appropriate methodology for convergence analysis.

REFERENCES

[1] E. Frantzekakis, "On Image Coding and Understanding: A Bayesian formulation for the problem of template matching based on coded image data," M.S. thesis, ISR Tech. Rep. MS 90-5, 1990.

[2] K. O. Perlmutter, S. M. Perlmutter, R. M. Gray, R. A. Olshen, and K. L. Oehler, "Bayes risk weighted vector quantization with posterior estimation for image compression and classification," *IEEE Trans. Image Processing*, vol. 5, pp. 347–360, Feb. 1996.

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

[4] J. Baras and S. Wolk, "Model based automatic target recognition from high rate resolution radar returns," in *Proc. SPIE Int. Symp. Intelligent Information Systems*, vol. 2234 (Orlando, FL, Apr. 1994), pp. 57–66.

[5] ———, "Wavelet based progressive classification of high range resolution radar returns," in *Proc. SPIE Int. Symp. Intelligent Information Systems*, vol. 2242 (Orland, FL, Apr. 1994), , pp. 967–977.

[6] ———, "Wavelet based progressive classification with learning: Applications to radar signals," in *Proc. SPIE 1995 Int. Symp. Aerospace/Defense Sensing and Dual-Use Photonics*, vol. 2491 (Orlando, FL, Apr. 1995), pp. 339–350.

[7] ———, "Wavelet-based hierarchical organization of large image databases: ISAR and face recognition," in *Proc. SPIE 12th Int. Symp. Aerospace, Defense Sensing, Simulation and Control* vol. 3391 (Orlando, FL, Apr. 1998), pp. 546–558.

[8] J. Baras and D. MacEnany, "Model-based ATR: Algorithms based on reduced target models, learning, and probing," in *Proc. 2nd ATR Systems and Technology Conf.*, vol. 1, Feb. 1992, pp. 277–300.

[9] D. MacEnany and J. Baras, "Scale-space polygonalization of target silhouettes and applications to model-based ATR," in *Proc. 2nd ATR Systems and Technology Conf.*, vol. 2, Feb. 1992, pp. 223–247.

[10] A. LaVigna, "Nonparametric classification using learning vector quantization," Ph.D. dissertation, Dep. Elec. Eng., Univ. Maryland, College Park, MD, 1989. ISR Tech. Rep. PhD 90, 1.

[11] T. Kohonen, *Self-Organizing Map*. Heidelberg, Germay: 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. 29th IEEE Conf. Decision and Control*, Dec. 1990, pp. 1735–1740.

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

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

[16] N. N. Krasovskii, *Stability of Motion*. Stanford, CA: Stanford Univ. Press, 1963.

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

[18] D. A. Reynolds, "Speaker identification and verification using Gaussian mixture speaker models," *Speech Commun.*, 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 Trans. Inform. Theory*, vol. 42, Jan. 1996.