A class of Order Statistics Learning Vector Quantizers

I. Pitas C. Kotropoulos N. Nikolaidis
Department of Electrical & Computer Engineering
University of Thessaloniki
GR-54006 Thessaloniki
Greece
(+30) 31-996-305
pitas@vergina.eng.auth.gr

R. Yang M. Gabbouj
Signal Processing Laboratory
Tampere University of Technology
SF-33101 Tampere
Finland
(+358) 31-161-967
moncef@cs.tut.fi

ABSTRACT

A novel class of Learning Vector Quantizers (LVQs) based on multivariate order statistics is proposed in order to overcome the drawback that the estimators for obtaining the reference vectors in LVQ do not have robustness either against erroneous choices for the winner vector or against the outliers that may exist in vector-valued observations. The performance of the proposed variants of LVQ is demonstrated by experiments. In the case of marginal median LVQ, its asymptotic properties are derived as well.

INTRODUCTION

Neural networks (NN) is a rapidly expanding research field which attracted the attention of scientists and engineers in the last decade. A large variety of artificial neural networks has been developed based on a multitude of learning techniques and having different topologies [2]. One prominent example of neural networks is the Learning Vector Quantizer (LVQ). It is an autoassociative nearest-neighbor classifier which classifies arbitrary patterns into classes using an error correction encoding procedure related to competitive learning [1]. In order to make a distinction between the (standard) LVQ algorithm and the proposed variants that are based on multivariate order statistics, the LVQ algorithm will be called linear LVQ algorithm hereafter.

Let us assume a sequence of vector-valued observations $\mathbf{x}(t) \in I\!R^p$ and a set of variable reference vectors $\{\mathbf{w}_i(t); \mathbf{w}_i \in I\!R^p, i = 1, 2, \ldots, K\}$. Let $\mathbf{w}_i(0)$ be randomly initialized. Competitive learning tries to find the best-matching reference vector $\mathbf{w}_c(t)$ to $\mathbf{x}(t)$ (i.e., the winner) where $c = \arg\min_i \| \mathbf{x} - \mathbf{w}_i \|$ with $\| \cdot \|$ denoting the Euclidean distance between any two vectors. This vector is updated and the process is repeated. After a large number of iterations, the different reference vectors $\mathbf{w}_i$ tend to be placed into the input space $I\!R^p$ in such a way that they approximate the probability density function (pdf) $f(\mathbf{x})$ in the sense of some minimal residual error $\varepsilon = \int_{\mathcal{X}} \| \mathbf{x} - \mathbf{w}_c \|^2 \ f(\mathbf{x})\, d\mathbf{x}$ where $\mathcal{X}$ is the domain of the input vector-valued observations and $d\mathbf{x}$ is the volume differential in the space $I\!R^p$. If the stochastic-gradient-descent algorithm [3] is applied to the minimization of ε in the $\mathbf{w}_c$ space and the weight vectors are updated as blocks concentrated around the winner, the following recursive relations result:

$$\begin{aligned} \mathbf{w}_i(t+1) &= \mathbf{w}_i(t) + \alpha(t)[\mathbf{x}(t) - \mathbf{w}_i(t)] \ \ \forall i \in \mathcal{N}_c(t) \\ \mathbf{w}_i(t+1) &= \mathbf{w}_i(t) \ \ \forall i \notin \mathcal{N}_c(t) \end{aligned} \quad (1)$$

where $\alpha(t)$ is the adaptation step and $\mathcal{N}_c(t)$ denotes a neighborhood around the winner. In the following, we use the notation n instead of t to denote discrete events. It can easily be seen that the reference vector for each class $i = 1, \ldots, K$ at time $n+1$ is a linear combination of the input vectors $\mathbf{x}(j)$ $j = 0, \ldots, n$ that have been assigned to class i. Moreover, it can be shown that in the special case of only one class and the adaptation step sequence $\alpha(n) = 1/(n+1)$, the winner vector is the arithmetic mean of the observations that have been assigned to the class (i.e., the maximum likelihood estimator of location). Neither in the case of multiple classes that are normally distributed nor in the case of non-Gaussian multivariate data distributions the linear LVQ is the optimal estimator of the cluster means. In general, linear LVQ and its variations suffer from the following drawbacks:

1. They do not use optimal estimators for obtaining the reference vectors $\mathbf{w}_i$, $i = 1, \ldots, K$ that match the pdf $f_i(\mathbf{x})$ of each class $i = 1, \ldots, K$.
2. They do not have robustness against erroneous choices for the winner vector, since it is well known that linear estimators have poor robustness properties [4].
3. They do not have robustness against the outliers that may exist in the vector observations.

In order to overcome these problems, we propose a variant of Learning Vector Quantizer that is based on multivariate order statistics [6, 5]. The performance of the proposed variants of LVQ is demonstrated by experiments. In the case of marginal median LVQ, its asymptotic properties are derived as well.

LEARNING VECTOR QUANTIZERS BASED ON MULTIVARIATE DATA ORDERING

There is no unambiguous, universally agreeable total ordering of N p-variate samples $\mathbf{x}_1, \ldots, \mathbf{x}_N$ where $\mathbf{x}_i = (x_{1i}, x_{2i}, \ldots, x_{pi})^T$, $i = 1, \ldots, N$. The following so-called sub-ordering principles are discussed in [6]: *marginal ordering, reduced (aggregate) ordering, partial ordering*, and *conditional (sequential) ordering*. In marginal ordering, the multivariate samples are ordered along each one of the p-dimensions:

$$x_{i(1)} \leq x_{i(2)} \leq \ldots \leq x_{i(N)} \quad i = 1, \ldots, p \qquad (2)$$

i.e., the sorting is performed in each channel of the multichannel signal independently. The i-th marginal order statistic is the vector $\mathbf{x}_{(i)} = \left(x_{1(i)}, x_{2(i)}, \ldots, x_{p(i)}\right)^T$. The marginal median has the following definition:

$$\mathbf{x}_{med} = \begin{cases} \left(x_{1(\nu+1)}, \ldots, x_{p(\nu+1)}\right)^T \\ \qquad \text{for } N = 2\nu + 1 \\ \left(\frac{x_{1(\nu)}+x_{1(\nu+1)}}{2}, \ldots, \frac{x_{p(\nu)}+x_{p(\nu+1)}}{2}\right)^T \\ \qquad \text{for } N = 2\nu. \end{cases} \qquad (3)$$

It can be used in the following way in order to define the *marginal median LVQ*. Let us denote by $\mathbf{X}_i(n)$ the set of the vector observations that have been assigned to each class i, $i = 1, \ldots, K$ until time $n-1$. We find at time n the winner vector $\mathbf{w}_c(n)$ that minimizes $\| \mathbf{x}(n) - \mathbf{w}_i(n) \|$, $i = 1, \ldots, K$. The marginal median LVQ (MMLVQ) updates the winner reference vector as follows:

$$\mathbf{w}_c(n+1) = \text{median}\,\{\mathbf{x}(n) \cup \mathbf{X}_c(n)\} \qquad (4)$$

The median operation is given by (3). Thus, all past class assignment sets $\mathbf{X}_i(n)$, $i = 1, \ldots, K$ are needed for MMLVQ. MMLVQ needs the calculation of the median of data sets of ever increasing size, as can be seen from (4). This may pose severe computational problems for relatively large n. However, for integer-valued data, a modification of the *running median algorithm* proposed by Huang et *al.* [7] can be devised to facilitate greatly median calculations by exploiting the fact that the marginal median of the already assigned samples $\mathbf{X}_i(n)$ is known.

Another definition of the multichannel median (based on R-ordering principles) is the so-called *vector median* proposed in [9]. The vector median is the observation that has the minimum distance from all the remaining observations, i.e.:

$$\sum_{i=1}^{N} |\mathbf{x}_i - \mathbf{x}_{med}| \leq \sum_{i=1}^{N} |\mathbf{x}_i - \mathbf{x}_j| \quad j = 1, \ldots, N. \qquad (5)$$

The *vector median* LVQ (VMLVQ) uses the following formula to update the winner vector $\mathbf{w}_c(n)$ at step n:

$$\mathbf{w}_c(n+1) = \text{vector median}\,\{\mathbf{x}(n) \cup \mathbf{X}_c(n)\} \qquad (6)$$

where $\mathbf{X}_i(n)$ is again the set of vector-valued observations that have been assigned to class i, $i = 1, \ldots, K$ so far and $\mathbf{x}(n)$ is the current observation. The vector median operator in the previous expression is the one defined in (5). Vector median LVQ keeps tract of all its history and therefore all data samples have equal contribution to the reference vector update procedure. In the case of non-stationary data, we can evaluate the vector median using a moving window to discard the older samples as new observations become available.

Since the reference vector update is restricted to belong to $\{\mathbf{x}(n) \cup \mathbf{X}_c(n)\}$, the VMLVQ trajectory is more smooth, in general, than the MMLVQ or the linear LVQ weight trajectory. The evaluation of the vector median of a data set is a rather computationally intensive operation (5) since it requires the evaluation of n sums, each containing $(n-1)$ terms of the form $|\mathbf{x}_j - \mathbf{x}_i|$ and also the evaluation of the minimum of n values. In the case of the vector median LVQ, for each time instant n, we have to calculate the vector median (6). The fact that the vector median of the data in $\mathbf{X}_c(n)$ has already been evaluated can be exploited in order to speed up the computations.

The *marginal weighted median LVQ* (MWMLVQ) can be defined as follows. Let us denote by

$$\mathbf{w}_i(n) = (w_{i1}(n), w_{i2}(n), \cdots, w_{ip}(n))^T \qquad (7)$$

the winner vector, i.e., $c = i$. In MWMLVQ, the elements of the winner vector are updated as follows:

$$w_{ij}(n+1) = \text{median}\,\{C_{i0} \diamond x_j(n), \ldots, C_{in} \diamond x_j(0)\} \qquad (8)$$

where $(C_{i0}, C_{i1}, \ldots, C_{in})^T$ is the vector of the duplication coefficients for the i-th class. The duplication coefficients can be chosen in such a way so that they weigh heavily the desired section of the observation data (i.e., the new observations or the old ones). If a weight C_{il} is zero, this means that the corresponding sample $\mathbf{x}(n-l)$ has not been assigned to the i-th class.

ASYMPTOTIC PROPERTIES OF MARGINAL MEDIAN LEARNING VECTOR QUANTIZER

In this section, the asymptotic properties of MMLVQ are studied. Due to lack of space, we shall sketch only the steps of the mathematical analysis and we shall present the basic conclusions. First, the expected stationary state of the MMLVQ is derived and is compared to the expected stationary state of the linear LVQ. Since both the linear LVQ

and the MMLVQ operate on each dimension independently, an 1-d contaminated Gaussian model

$$f(x) = \epsilon N(m_1, \sigma) + (1-\epsilon)N(m_2, \sigma) \qquad (9)$$

is considered. To this end, the thresholds determined by the linear LVQ and the MMLVQ at the equilibrium for discriminating the two input data classes must be known. A very simple algorithm for solving the set of equations that define implicitly the stationary state of the linear LVQ and the MMLVQ is developed [10]. In addition, the thresholds determined by the linear LVQ and the MMLVQ have been compared to the threshold predicted by the statistical detection theory, i.e., the threshold that minimizes the probability of false classification [8]. The bias introduced by the linear LVQ and the MMLVQ in estimating the unconditional mean for the dominating cluster in the contaminated Gaussian model (9) is depicted in Figure 1 for $\sigma = 3$. We have also included the conditional mean that corresponds to the decision region for the dominating cluster that is predicted by the statistical detection theory. In other words, we have plotted the following quantities:

$$\begin{array}{ll} \mid \overline{w}_2 - m_2 \mid & \text{for } \epsilon \leq 0.5, \text{ and} \\ \mid \overline{w}_1 - m_1 \mid & \text{for } \epsilon > 0.5 \end{array} \qquad (10)$$

versus ϵ. It is seen that the MMLVQ outperforms the linear LVQ with respect to the bias.

From Figure 1, it is evident that both the linear LVQ as well as the MMLVQ are not unbiased estimators of the data cluster means. Accordingly, the asymptotic variance $V(T, F)$, T = LVQ, MMLVQ defined by:

$$V(T, F) = \int \mathrm{IF}(\mathbf{x};\, T,\, F)^2 f(\mathbf{x}) d\mathbf{x} \qquad (11)$$

where $\mathrm{IF}(\mathbf{x};\, T,\, F)$ is the influence function of T at F [4, 5] does not take into account the bias introduced by each estimator, since it is simply the variance of the random variable $\sqrt{n}(T_n - T(F))$ that is normally distributed as $n \to \infty$. Observe that the asymptotic variance of the estimator T at model F is essentially the upper bound of its variance, i.e., $V(T, F) = \max_n \mathrm{E}\left[(T_n - T(F))^2\right] = \mathrm{E}\left[(T_n - T(F))^2\right] \mid_{n=1}$. Therefore, the asymptotic relative efficiency (ARE) of LVQ and MMLVQ defined by:

$$\mathrm{ARE}(\mathrm{MMLVQ},\, \mathrm{LVQ}) = \frac{V(\mathrm{LVQ}, F)}{V(\mathrm{MMLVQ}, F)} \qquad (12)$$

is not appropriate for comparing the performance of the two estimators. We propose the following modified ARE:

$$\widetilde{\mathrm{ARE}}(\mathrm{MMLVQ},\, \mathrm{LVQ}) = \frac{\max_n \mathrm{E}\left[(\mathrm{LVQ}_n - \mathbf{M})^2\right]}{\max_n \mathrm{E}\left[(\mathrm{MMLVQ}_n - \mathbf{M})^2\right]} \qquad (13)$$

where $\mathbf{M} = (\mathbf{m}_1 \mid \ldots \mid \mathbf{m}_K)^T$ is the vector of the unconditional means to be estimated. The modified ARE (13) has been evaluated for the distribution model (9) under study. In Figure 2, the modified ARE is plotted for several $\epsilon \in [0.2,\, 0.8]$ and σ. It can be seen that the performance of the MMLVQ is improved as σ increases. However, even in this case, linear LVQ is better than the MMLVQ with respect to the mean-squared estimation error. In the case of a contaminated Laplacian distribution model, it can be shown that MMLVQ outperforms the linear LVQ not only with respect to the bias, but also with respect to the mean-squared estimation error.

EXPERIMENTAL RESULTS

The performance of the proposed order statistics LVQs has been tested on a two-dimensional sample set that is described by the probability density function of the form

$$\begin{aligned} f(x_1, x_2) &= p\, U([-5, 20], [-5, 20]) + (1-p) \cdot \\ &\cdot \quad [\epsilon\, N(5, 5;\, 1,\, 1;\, 0) + \\ &+ \quad (1-\epsilon) N(10, 10;\, 1,\, 1;\, 0)] \end{aligned} \qquad (14)$$

where $U([-5, 20], [-5, 20])$ denotes the pdf of uniformly distributed outliers in the domain $[-5, 20] \times [-5, 20]$ and $N(m_{i1}, m_{i2};\, \sigma_{i1}, \sigma_{i2};\, r)$ denotes a two-dimensional Gaussian distribution with mean m_{ij} and standard deviation σ_{ij} along each dimension j $(j = 1, 2)$ and correlation coefficient r. Such a data set having $p = 0.2$ and $\epsilon = 0.5$ is shown in Figure 3a, together with the trajectories of the weights determined by the marginal median LVQ algorithm. It must be stressed that this data set is heavily corrupted. It is clear that the MMLVQ converges close the correct solution. The VMLVQ with the same initial weights has also converged close to the cluster means on this data set as can be seen in Figure 3b. On the contrary, the linear LVQ does not converge to the correct solution in this case, as can be seen in Figure 3c.

REFERENCES

[1] T.K. Kohonen, *Self-Organization and Associative Memory*, 3rd ed. Berlin, Heidelberg, Germany: Springer-Verlag, 1989.

[2] Special issue on neural networks, I: Theory & modeling; II: Analysis, techniques, & applications, *Proceedings of the IEEE*, 78, 9-10, pp. 1409–1680, September-October 1990.

[3] J.G. Proakis, and D.G. Manolakis, *Introduction to Digital Signal Processing*. New York: Macmillan, 1988.

[4] P.J. Huber, *Robust Statistics*. New York: J. Wiley, 1981.

[5] I. Pitas, and A. N. Venetsanopoulos, *Nonlinear Digital Filters: Principles and Applications*. Hingham, MA: Kluwer Academic Publishers, 1990.

[6] V. Barnett, "The ordering of multivariate data," *J. R. Statist. Soc. A*, 139, 3, pp. 318–354, 1976.

[7] T.S. Huang, G.J. Yang, and G.Y. Tang, "A fast two-dimensional median filtering algorithm," *IEEE Trans. on Acoustics, Speech and Signal Processing*, 27, 1, pp. 13–18, 1979.

[8] H.L. Van Trees, *Detection, Estimation and Modulation Theory*. New York: J. Wiley, 1968.

[9] J. Astola, P. Haavisto, and Y. Neuvo, "Vector median filters," *Proceedings of the IEEE*, 78, 4, pp. 678–689, April 1990.

[10] C. Kotropoulos, I. Pitas and M. Gabbouj, "Marginal median Learning Vector Quantizer," *European Signal Processing Conference '94* Edinburg 1994, U.K., to be presented.

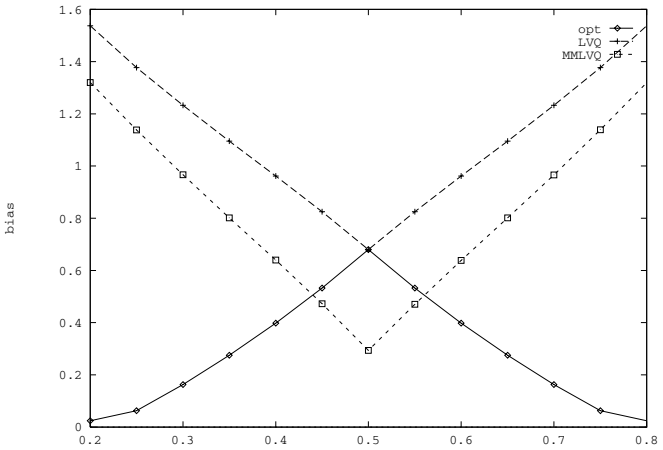

Figure 1

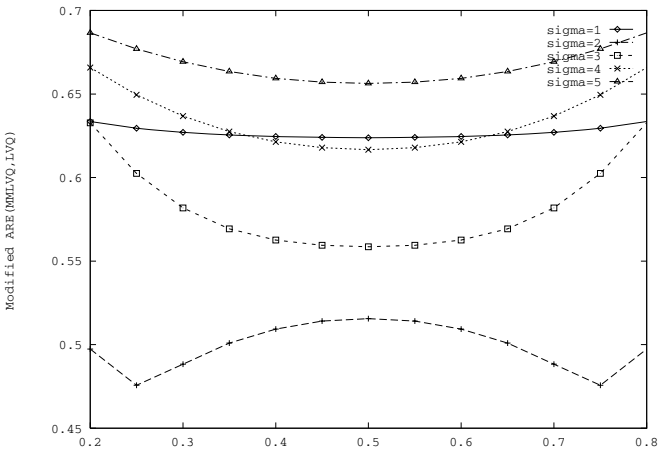

Figure 2

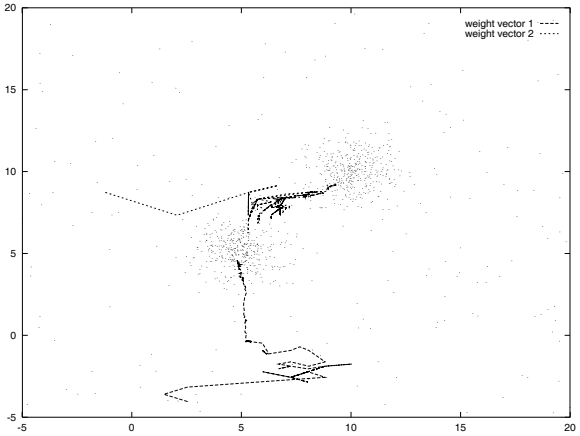

Figure 3a

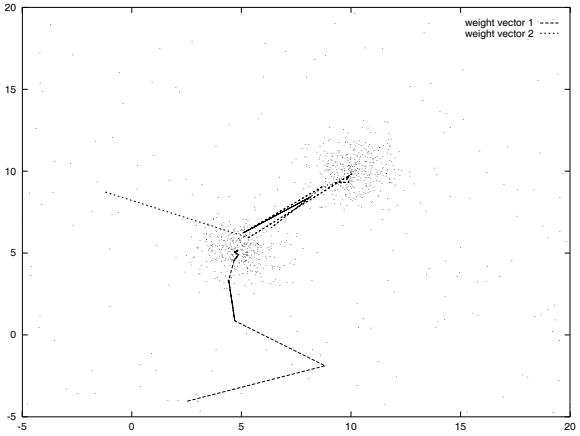

Figure 3b

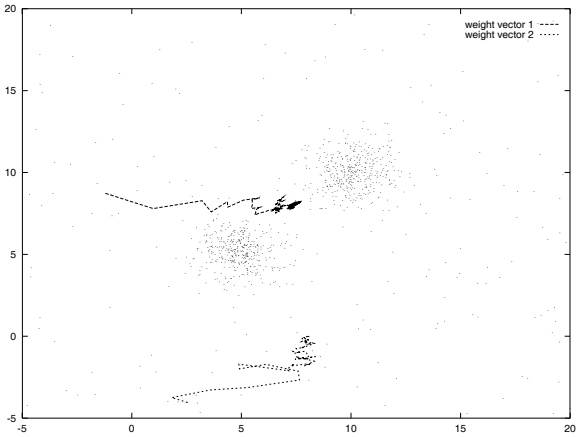

Figure 3c