A Methodology for Constructing Fuzzy Algorithms for Learning Vector Quantization

Nicolaos B. Karayiannis, *Member, IEEE*

***Abstract*—This paper presents a general methodology for the development of fuzzy algorithms for learning vector quantization (FALVQ). The design of specific FALVQ algorithms according to existing approaches reduces to the selection of the membership function assigned to the weight vectors of an LVQ competitive neural network, which represent the prototypes. According to the methodology proposed in this paper, the development of a broad variety of FALVQ algorithms can be accomplished by selecting the form of the interference function that determines the effect of the nonwinning prototypes on the attraction between the winning prototype and the input of the network. The proposed methodology provides the basis for extending the existing FALVQ 1, FALVQ 2, and FALVQ 3 families of algorithms. This paper also introduces two quantitative measures which establish a relationship between the formulation that led to FALVQ algorithms and the competition between the prototypes during the learning process. The proposed algorithms and competition measures are tested and evaluated using the IRIS data set. The significance of the proposed competition measures in practical applications is illustrated by using various FALVQ algorithms to perform segmentation of magnetic resonance images of the brain.**

***Index Terms*— Competition measure, competitive learning, competitive learning vector quantization (LVQ) network, construction methodology, interference function, membership function, vector quantization, update equation.**

I. Introduction

The objective of *vector quantization* (VQ) is the representation of a set of feature vectors $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^n$ by a set of prototypes $\mathcal{V} = \{\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_c\} \subset \mathbb{R}^n$. Thus, vector quantization can also be seen as a mapping from an n-dimensional Euclidean space into the finite set $\mathcal{V} \in \mathbb{R}^n$, also referred to as the codebook.

Codebook design can be performed by clustering algorithms, which are typically developed by solving a constrained minimization problem using alternating optimization. These clustering techniques include the crisp c-means [4], fuzzy c-means [2], and generalized fuzzy c-means algorithms [8], [9].

Recent developments in neural network architectures resulted in *learning vector quantization* (LVQ) algorithms. LVQ is the name used for unsupervised learning algorithms associated with the competitive network shown in Fig. 1. The network consists of an input layer and an output layer. Each node in the input layer is connected directly to the cells, or nodes, in the output layer. A prototype vector is associated with each cell in the output layer as shown in Fig. 1.

Kohonen [18] proposed an unsupervised learning scheme, known as the (unlabeled data) LVQ. This algorithm can be used to generate crisp c-partitions of unlabeled data vectors. Pal *et al.* [20] identified a close relationship between this algorithm and a clustering procedure proposed earlier by MacQueen, known as the sequential hard c-means algorithm. It must be emphasized here that the LVQ 1, LVQ 2, and LVQ 3 algorithms proposed by Kohonen [17], [19] for fine tuning the *self-organizing feature map* (SOFM) are supervised in the sense that their implementation requires labeled feature vectors, that is, feature vectors whose classification is already known.

Huntsberger and Ajjimarangsee [7] attempted to establish a connection between feature maps and fuzzy clustering by modifying the learning rule proposed by Kohonen for the SOFM. However, the resulting hybrid learning scheme lacked theoretical foundations, formal derivations and clear objectives. Bezdek *et al.* [3], [21] proposed a *batch* learning scheme, known as *fuzzy learning vector quantization* (FLVQ). Karayiannis *et al.* [10], [16] presented a formal derivation of batch FLVQ algorithms, which were originally introduced on the basis of intuitive arguments. This derivation was based on the minimization of a functional defined as the average generalized distance between the feature vectors and the prototypes. This minimization problem is actually a reformulation of the problem of determining fuzzy c-partitions that was solved by fuzzy c-means algorithms [2], [6].

Pal *et al.* [20] suggested that LVQ can be performed through an unsupervised learning process using a competitive neural network whose weight vectors represent the prototypes. According to their formulation, LVQ can be achieved by minimizing a loss function which measures the locally weighted error of the input vector with respect to the winning prototype, that is, the prototype that is closest to the input vector in the Euclidean distance sense. This formulation resulted in the *generalized learning vector quantization* (GLVQ) algorithm [20] and the GLVQ-F algorithms [11].

Karayiannis and Pai [12]–[15] proposed a framework for the development of *fuzzy algorithms for learning vector quantization* (FALVQ). The development of FALVQ algorithms was based on the minimization of the weighted sum of the squared Euclidean distances between an input vector, which represents a feature vector, and the weight vectors of the

Manuscript received June 10, 1995; revised January 8, 1996 and May 18, 1996.

The author is with the Department of Electrical and Computer Engineering, University of Houston, TX 77204-4793 USA.

Publisher Item Identifier S 1045-9227(97)00466-9.

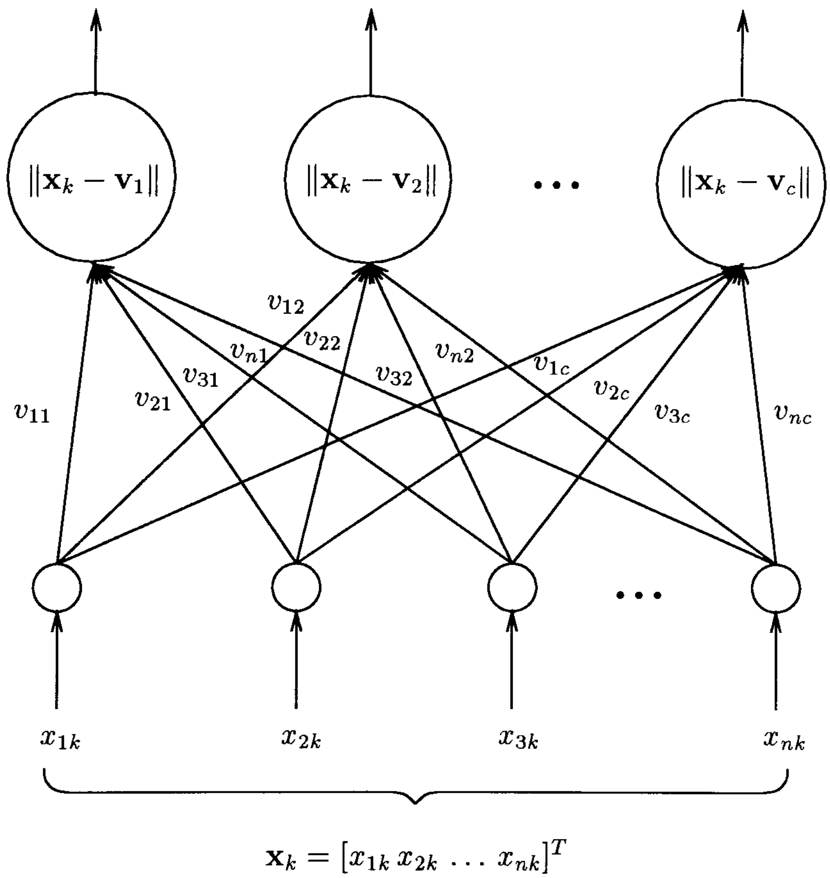


Fig. 1. The LVQ competitive network.

LVQ network, which represent the prototypes. The distances between each input vector and the prototypes are weighted by a set of membership functions, which regulate the competition between various prototypes for each input and, thus, determine the strength of attraction between each input and the prototypes during the learning process. The design of specific FALVQ algorithms reduces to the selection of membership functions that satisfy certain properties [13], [15].

This paper is organized as follows: Section II presents a review of the formulation that led to the development of a broad variety of FALVQ algorithms. Section III proposes a new methodology for constructing FALVQ algorithms. Section IV presents the application of the proposed methodology in the development of the extended FALVQ 1, FALVQ 2, and FALVQ 3 families of algorithms, respectively. Section V introduces two competition measures that can be used to control the competition between the winning and nonwinning prototypes during the learning process. Section VI presents an experimental evaluation of the extended FALVQ algorithms and the proposed competition measures. Section VII contains concluding remarks.

II. Fuzzy Algorithms for Learning Vector Quantization

Consider the set $\mathcal{X}$ of samples from an n-dimensional Euclidean space and let $f(\mathbf{x})$ be the probability distribution function of $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^n$. Learning vector quantization is frequently based on the minimization of the functional [20]

$$L(\mathbf{v}_r, r = 1, 2, \cdots, c) = \int\int \cdots \int_{\mathbb{R}^n} \sum_{r=1}^{c} u_r \|\mathbf{x} - \mathbf{v}_r\|^2 f(\mathbf{x})\, d\mathbf{x} \tag{1}$$

which represents the expectation of the loss function $L_{\mathbf{x}} = L_{\mathbf{x}}(\mathbf{v}_r, r = 1, 2, \cdots, c)$, defined as

$$L_{\mathbf{x}} = L_{\mathbf{x}}(\mathbf{v}_r, r = 1, 2, \cdots, c) = \sum_{r=1}^{c} u_r \|\mathbf{x} - \mathbf{v}_r\|^2. \tag{2}$$

In the above definitions, $u_r = u_r(\mathbf{x}), r = 1, 2, \cdots, c$, is a set of membership functions which regulate the competition between the prototypes $\mathbf{v}_r, r = 1, 2, \cdots, c$, for the input $\mathbf{x}$. The specific form of the membership functions determines the strength of attraction between each input and the prototypes during the learning process [12]–[15]. The loss function is often defined with respect to the winning prototype. Assuming that $\mathbf{v}_i$ is the winning prototype corresponding to the input vector $\mathbf{x}$, that is, the closest prototype to $\mathbf{x}$ in the Euclidean distance sense, the membership $u_r, r = 1, 2, \cdots, c$, can be of the form

$$u_r = u_{ir} = \begin{cases} 1 & \text{if } r = i \\ u\left(\frac{\|\mathbf{x} - \mathbf{v}_i\|^2}{\|\mathbf{x} - \mathbf{v}_r\|^2}\right) & \text{if } r \neq i\,. \end{cases} \tag{3}$$

In such a case, the loss function measures the locally weighted error of each input vector with respect to the winning prototype [20].

The minimization of (1) using gradient descent is a difficult task, since the winning prototype involved in the definition of the loss function $L_{\mathbf{x}}$ is determined with respect to the corresponding input vector $\mathbf{x} \in \mathcal{X}$. Following Tsypkin [22], Pal *et al.* suggested the use of the gradient of the instantaneous loss function (2) when the probability distribution function $f(\cdot)$ is not known [20]. This approach implies the *sequential* update of the prototypes with respect to the input vectors $\mathbf{x} \in \mathcal{X}$ and is frequently used in the development of learning algorithms [17], [22].

The development of fuzzy algorithms for learning vector quantization requires the selection of the membership functions assigned to the prototypes [12]–[15]. A fair competition among the prototypes is guaranteed if the membership function assigned to each prototype: 1) is invariant under uniform scaling of the entire data set; 2) is equal to one if the prototype is the winner; 3) takes values between one and zero if the prototype is not a winner; and 4) approaches zero if the prototype is not a winner and its distance from the input vector approaches infinity.

The relationship between the form of the membership function and the competition between the prototypes during the learning process can be quantified by focusing on the relative contribution of the nonwinning prototypes to the loss function (2). If the membership is given in (3), the loss function (2) can be written as

$$L_{\mathbf{x}} = \sum_{r=1}^{c} u_{ir} \|\mathbf{x}-\mathbf{v}_r\|^2 = \|\mathbf{x}-\mathbf{v}_i\|^2 + \sum_{r \neq i}^{c} u_{ir} \|\mathbf{x}-\mathbf{v}_r\|^2. \quad (4)$$

Assuming that $\mathbf{v}_i$ is the winning prototype, each nonwinning prototype $\mathbf{v}_r \neq \mathbf{v}_i$ contributes to the loss function $L_{\mathbf{x}}$ through the term $u_{ir}\|\mathbf{x} - \mathbf{v}_r\|^2$. Thus, the *relative contribution* of the nonwinning prototype $\mathbf{v}_r$ with respect to the winning prototype $\mathbf{v}_i$ can be measured by the ratio $u_{ir}\|\mathbf{x} - \mathbf{v}_r\|^2/\|\mathbf{x} - \mathbf{v}_i\|^2$.

The search for admissible membership functions can be facilitated by requiring that $u_{ir}\|\mathbf{x} - \mathbf{v}_r\|^2/\|\mathbf{x} - \mathbf{v}_i\|^2$ is a function of the ratio $\|\mathbf{x} - \mathbf{v}_i\|^2/\|\mathbf{x} - \mathbf{v}_r\|^2$, that is,

$$u_{ir} \frac{\|\mathbf{x} - \mathbf{v}_r\|^2}{\|\mathbf{x} - \mathbf{v}_i\|^2} = p\left(\frac{\|\mathbf{x} - \mathbf{v}_i\|^2}{\|\mathbf{x} - \mathbf{v}_r\|^2}\right). \quad (5)$$

The obvious advantage of this choice is that the properties of $p(\cdot)$ relate directly to the relative contribution of the prototype $\mathbf{v}_r$ to the loss function $L_{\mathbf{x}}$. Since $u_{ir} = u(\|\mathbf{x} - \mathbf{v}_i\|^2/\|\mathbf{x} - \mathbf{v}_r\|^2)$, $\forall r \neq i$, the corresponding function $u(\cdot)$ is of the form $u(z) = z\,p(z)$. In the trivial case where $p(z) = 0, \forall z \in (0,1)$, the membership function (3) corresponds to the nearest prototype condition, which results in Kohonen's (unlabeled data) LVQ algorithm. In this case, the nonwinning prototypes are not attracted by the input $\mathbf{x}$ and have no effect on the attraction of the winning prototype by the input $\mathbf{x}$.

If $u(z) = z\,p(z)$, the consistency of the corresponding membership function with the admissibility conditions presented above can be guaranteed by selecting functions $p(\cdot)$ satisfying certain properties [13], [15]. Among the functions $p(z)$ that satisfy the admissibility conditions, the development of FALVQ algorithms is based in this paper on differentiable functions $p(\cdot)$ which satisfy the following conditions: 1) $0 < p(z) < 1$, $\forall z \in (0,1)$; 2) $p(z)$ approaches one as z approaches zero; 3) $p(z)$ is a monotonically decreasing function in the interval $(0,1)$; and 4) $p(z)$ attains its minimum value at $z = 1$.

A variety of fuzzy algorithms for learning vector quantization can be derived by minimizing the loss function (4) using gradient descent. If $\mathbf{x}$ is the input vector, the winning prototype $\mathbf{v}_i$ can be updated by [13], [15]

$$\Delta\mathbf{v}_i = \eta\,(\mathbf{x} - \mathbf{v}_i)\left(1 + \sum_{r \neq i}^{c} w_{ir}\right) \quad (6)$$

where $w_{ir} = w(\|\mathbf{x} - \mathbf{v}_i\|^2/\|\mathbf{x} - \mathbf{v}_r\|^2)$, with

$$w(z) = p(z) + z\,p'(z) = u'(z). \quad (7)$$

Each nonwinning prototype $\mathbf{v}_j \neq \mathbf{v}_i$ can be updated by [13], [15]

$$\Delta\mathbf{v}_j = \eta\,(\mathbf{x} - \mathbf{v}_j)\,n_{ij} \quad (8)$$

where $n_{ij} = n(\|\mathbf{x} - \mathbf{v}_i\|^2/\|\mathbf{x} - \mathbf{v}_j\|^2)$, with

$$n(z) = -z^2\,p'(z) = u(z) - z\,u'(z). \quad (9)$$

The update of the prototypes during the learning process depends on the learning rate $\eta \in [0,1]$, which is a monotonically decreasing function of the number of iterations ν. The learning rate can be a linear function of ν defined as $\eta = \eta(\nu) = \eta_0(1 - \nu/N)$, where η_0 is its initial value and N the total number of iterations predetermined for the learning process.

According to (6), the update of the winning prototype $\mathbf{v}_i$ is affected by all the nonwinning prototypes $\mathbf{v}_r \neq \mathbf{v}_i$, while w_{ir} represents the *interference* from the nonwinning prototype $\mathbf{v}_r$ to the update of the winning prototype $\mathbf{v}_i$. In fact, the term $\sum_{r \neq i}^{c} w_{ir}$ represents the cumulative effect of the nonwinning prototypes on the attraction of the winning prototype by the input vector $\mathbf{x}$. In contrast, (8) indicates that the update of each nonwinning prototype $\mathbf{v}_j \neq \mathbf{v}_i$ is affected only by the winning prototype $\mathbf{v}_i$. In this case, n_{ij} represents the *interference* from the winning prototype $\mathbf{v}_i$ to the update of the nonwinning prototype $\mathbf{v}_j$.

The selection of specific membership functions can be facilitated by examining the relationship between the form of $p(\cdot)$ and the competition among the prototypes in the extreme cases where $p(\cdot)$ equals its lower and upper bounds specified by the inequality $0 < p(z) < 1$, $\forall z \in (0,1)$. If $p(z) = 0$, $\forall z \in (0,1)$, then $w_{ir} = 0$, $\forall r \neq i$ and $n_{ij} = 0$, $\forall j \neq i$. In this case, the winning prototype $\mathbf{v}_i$ is updated according to

$$\Delta\mathbf{v}_i = \eta\,(\mathbf{x} - \mathbf{v}_i) \quad (10)$$

while the nonwinning prototypes $\mathbf{v}_j \neq \mathbf{v}_i$ remain unchanged. If $p(z) = 1$, $\forall z \in (0,1)$, then $u(z) = z$, $\forall z \in (0,1)$. Since $p'(z) = 0$, $\forall z \in (0,1)$, (7) indicates that $w_{ir} = 1$, $\forall r \neq i$ and $\sum_{r \neq i}^{c} w_{ir} = c - 1$. According to (6), the winning prototype $\mathbf{v}_i$ is updated by

$$\Delta\mathbf{v}_i = \eta\,c\,(\mathbf{x} - \mathbf{v}_i). \quad (11)$$

TABLE I
MEMBERSHIP FUNCTIONS AND INTERFERENCE FUNCTIONS FOR THE FALVQ 1, FALVQ 2, AND FALVQ 3 FAMILIES OF ALGORITHMS

FALVQ Family	$u(x)$	$w(x)$	$n(x)$
FALVQ 1 $(0 < \alpha < \infty)$	$x\,(1+\alpha\,x)^{-1}$	$(1+\alpha\,x)^{-2}$	$\alpha\,x^2\,(1+\alpha\,x)^{-2}$
FALVQ 2 $(0 < \beta < \infty)$	$x\exp(-\beta\,x)$	$(1-\beta\,x)\exp(-\beta\,x)$	$\beta\,x^2\exp(-\beta\,x)$
FALVQ 3 $(0 < \gamma < 1)$	$x(1-\gamma\,x)$	$1-2\gamma\,x$	$\gamma\,x^2$

Under the same assumption, $n_{ij} = 0, \forall j \neq i$. Thus, the nonwinning prototypes $\mathbf{v}_j \neq \mathbf{v}_i$ are not updated with respect to the input $\mathbf{x}$ regardless of their distance from the winning prototype $\mathbf{v}_i$.

The above formulation provided the basis for the development of the FALVQ 1, FALVQ 2, and FALVQ 3 families of algorithms [13], [15]. Table I shows the membership functions that generated these families of algorithms and the corresponding interference functions. If $\mathbf{x}$ is the input vector, then the winning prototype is updated by (6) with w_{ir} evaluated in terms of the interference function $w(\cdot)$ shown in Table I as $w_{ir} = w(\|\mathbf{x}-\mathbf{v}_i\|^2/\|\mathbf{x}-\mathbf{v}_r\|^2)$. The nonwinning prototypes $\mathbf{v}_j \neq \mathbf{v}_i$ can be updated by (8) with n_{ij} evaluated in terms of the interference function $n(\cdot)$ shown in Table I as $n_{ij} = n(\|\mathbf{x}-\mathbf{v}_i\|^2/\|\mathbf{x}-\mathbf{v}_j\|^2)$.

The algorithms described above can be summarized as follows.

1) Select c; fix η_0, N; set $\nu = 0$; randomly generate an initial codebook $\mathcal{V}_0 = \{\mathbf{v}_{1,0}, \mathbf{v}_{2,0} \cdots, \mathbf{v}_{c,0}\}$.
2) Calculate $\eta = \eta_0(1-\nu/N)$.
3) Set $\nu = \nu + 1$.
4) For each input vector $\mathbf{x}$:
   - find i such that $\|\mathbf{x}-\mathbf{v}_{i,\nu-1}\|^2 < \|\mathbf{x}-\mathbf{v}_{j,\nu-1}\|^2, \forall j \neq i$.
   - calculate $u_{ir,\nu} = u(\|\mathbf{x}-\mathbf{v}_{i,\nu-1}\|^2/\|\mathbf{x}-\mathbf{v}_{r,\nu-1}\|^2), \forall r \neq i$.
   - calculate $w_{ir,\nu} = u'(\|\mathbf{x}-\mathbf{v}_{i,\nu-1}\|^2/\|\mathbf{x}-\mathbf{v}_{r,\nu-1}\|^2), \forall r \neq i$.
   - calculate $n_{ir,\nu} = u_{ir,\nu} - (\|\mathbf{x}-\mathbf{v}_{i,\nu-1}\|^2/\|\mathbf{x}-\mathbf{v}_{r,\nu-1}\|^2)\, w_{ir,\nu}, \forall r \neq i$.
   - update $\mathbf{v}_i$ by $\mathbf{v}_{i,\nu} = \mathbf{v}_{i,\nu-1} + \eta\,(\mathbf{x}-\mathbf{v}_{i,\nu-1})\,(1+\sum_{r\neq i} w_{ir,\nu})$.
   - update $\mathbf{v}_j \neq \mathbf{v}_i$ by $\mathbf{v}_{j,\nu} = \mathbf{v}_{j,\nu-1} + \eta\,(\mathbf{x}-\mathbf{v}_{j,\nu-1})\, n_{ij,\nu}$.
5) If $\nu < N$, then go to Step 2).

III. CONSTRUCTING FALVQ ALGORITHMS BASED ON THE INTERFERENCE FUNCTION

The development of FALVQ algorithms was based on the selection of membership functions that satisfy certain properties [12]–[15]. Given a membership function, the competition of the prototypes during the learning process is determined by the form of the corresponding interference functions $w(\cdot)$ and $n(\cdot)$. Thus, the development of FALVQ algorithms with desired behavior can be facilitated by directly selecting the interference functions instead of the membership function. This can be accomplished easily in this case, since $w(x) = u'(x)$. If $u(\cdot)$ is of the form $u(x) = x\,p(x)$, then

$$w(x) = u'(x) = p(x) + x\,p'(x). \tag{12}$$

According to the properties of $p(\cdot)$,

$$\lim_{x\to 0} w(x) = \lim_{x\to 0} p(x) = 1. \tag{13}$$

Since $p(\cdot)$ is a monotonically decreasing function in the interval $(0,1)$, $p'(x) < 0, \forall x \in (0,1)$ and also $x\,p'(x) < 0, \forall x \in (0,1)$. Since $p(x) \leq 1, \forall x \in (0,1)$

$$w(x) = p(x) + x\,p'(x) < 1, \forall x \in (0,1). \tag{14}$$

Since $w(x) = u'(x)$, the last admissibility condition for the interference function $w(\cdot)$ can be established by observing that

$$\int_0^1 w(x)\,dx = u(1) - u(0). \tag{15}$$

An admissible membership function $u(\cdot)$ must satisfy the conditions $u(0) = 0$ and $u(1) = p(1) \geq 0$. Thus,

$$\int_0^1 w(x)\,dx \geq 0. \tag{16}$$

In summary, an integrable function $w(\cdot)$ is an admissible interference function if it satisfies the following conditions: 1) $w(x)$ approaches one as x approaches zero; 2) $w(x) < 1, \forall x \in (0,1)$; and 3) $\int_0^1 w(x)\,dx \geq 0$. It must be emphasized here that $w(\cdot)$ is not necessarily a monotonically decreasing function in the interval $(0,1)$.

Given an interference function $w(\cdot)$, the corresponding membership function $u(\cdot)$ can be calculated as

$$u(x) = \int w(x)\,dx + C. \tag{17}$$

The constant C can be determined by requiring that $u(x)$ approaches zero as x approaches zero, that is,

$$\lim_{x\to 0} u(x) = 0. \tag{18}$$

The interference function $n = n(x)$ can be obtained in terms of $w(x)$ and $u(x)$ as $n(x) = u(x) - x\,w(x)$.

IV. NEW FALVQ ALGORITHMS

The construction methodology presented above allows the designer to have a more direct impact on the competition between the prototypes during the learning process. This methodology is used in this section for extending the existing FALVQ 1, FALVQ 2, and FALVQ 3 families of algorithms, which are summarized in Table I [13], [15].

A. *Extending the FALVQ 1 Family of Algorithms*

The FALVQ 1 family of algorithms can be extended by selecting an interference function of the form

$$w(x) = \frac{1}{(1+\alpha x)^{n+1}} \tag{19}$$

where $\alpha > 0$ and $n \geq 1$. The interference function $w(\cdot)$ defined in (19) decreases monotonically from its maximum value $w(0) = 1$ to $w(1) = (1+\alpha)^{-(n+1)}$. For a fixed n, $w(1)$ approaches one as α decreases to zero. Moreover, $w(1)$ decreases and approaches zero as α increases and approaches infinity. If the value of α is fixed, the value of $w(1)$ decreases as n increases.

According to the proposed algorithm construction method, the membership function corresponding to $w(\cdot)$ can be obtained as

$$u(x) = \int w(x)dx = \int \frac{dx}{(1+\alpha x)^{n+1}} = -\frac{1}{n\alpha(1+\alpha x)^n} + C. \quad (20)$$

The constant C can be evaluated using the condition (18), which results in $C = -\frac{1}{n\alpha}$. Thus, the membership function corresponding to (19) is

$$u(x) = \frac{1}{n\alpha}\left(1 - \frac{1}{(1+\alpha x)^n}\right). \quad (21)$$

According to the binomial identity,

$$(1+\alpha x)^n = 1 + n(\alpha x) + \cdots + (\alpha x)^n. \quad (22)$$

Thus, it can be verified that the membership function defined in (21) is of the form $u(x) = x\,p(x)$. For $w(\cdot)$ defined in (19), the interference function $n(\cdot)$ for the nonwinning prototypes is

$$n(x) = u(x) - x\,w(x) = \frac{1}{n\alpha}\left(1 - \frac{1}{(1+\alpha x)^n}\right) - \frac{x}{(1+\alpha x)^{n+1}}. \quad (23)$$

The original FALVQ 1 family of algorithms can be obtained from the interference function $w(\cdot)$ defined in (19) with $n = 1$.

B. Extending the FALVQ 2 Family of Algorithms

The proposed algorithm construction method can be used for extending the FALVQ 2 family of algorithms. This can be accomplished by using the integral

$$\frac{1}{n!}\int (n! - (\beta x)^n)e^{-\beta x}dx = \frac{e^{-\beta x}}{\beta}\sum_{k=1}^{n}\frac{(\beta x)^k}{k!} \quad (24)$$

where $\beta > 0$ and $n \geq 1$. This latter identity indicates that by selecting an interference function of the form

$$w(x) = \frac{1}{n!}(n! - (\beta x)^n)e^{-\beta x} \quad (25)$$

the corresponding membership function becomes

$$u(x) = \frac{e^{-\beta x}}{\beta}\sum_{k=1}^{n}\frac{(\beta x)^k}{k!}. \quad (26)$$

The resulting membership function is of the form $u(x) = x\,p(x)$. Moreover, $u(x)$ attains the value of zero as x approaches zero. The interference function $n(\cdot)$ can be obtained by combining (25) and (26) as

$$n(x) = u(x) - xw(x). \quad (27)$$

If $n = 1$, the interference function $w(\cdot)$ defined in (25) leads to the original FALVQ 2 family of algorithms.

The interference function (25) decreases from its maximum value $w(0) = 1$ to $w(1) = (1 - \frac{\beta^n}{n!})e^{-\beta}$. Nevertheless, $w(\cdot)$ is not necessarily a monotonically decreasing function in the interval $(0, 1)$, as indicated by the proposition which follows.

Proposition 1: The interference function $w(\cdot)$ defined in (25) is monotonically decreasing in the interval $(0, 1)$ if

$$\beta < n. \quad (28)$$

The corresponding membership function (26) is monotonically increasing in the interval $(0, 1)$ if $\beta < (n!)^{\frac{1}{n}}$.

Proof: The proof of this proposition is presented in Appendix A.

C. Extending the FALVQ 3 Family of Algorithms

The FALVQ 3 family of algorithms can be extended using the proposed construction method and some well-known integrals. For $n \geq 1$ and $\gamma \in (0, 1)$,

$$\int [1 - (n+1)\gamma x](1-\gamma x)^{n-1}dx = x(1-\gamma x)^n. \quad (29)$$

According to (29), the selection of an interference function of the form

$$w(x) = [1 - (n+1)\gamma x](1-\gamma x)^{n-1} \quad (30)$$

results in membership functions of the form

$$u(x) = x(1-\gamma x)^n. \quad (31)$$

The corresponding interference function $n(x)$ is given by

$$n(x) = u(x) - x\,w(x) = n\gamma x^2(1-\gamma x)^{n-1}. \quad (32)$$

The family of FALVQ 3 algorithms can be interpreted as the special case of the above formulation which corresponds to $n = 1$. For $n = 1$, $w(x)$ is a linear and monotonically decreasing function of x over the interval $(0, 1)$. If $n > 1$, $w(x)$ is a nonlinear function of x. Moreover, $w(x)$ is not guaranteed to be a monotonically decreasing function of x over the interval $(0, 1)$. The proposition which follows determines the combinations of γ and n which lead to interference functions $w(x)$ that are monotonically decreasing over the interval $(0, 1)$.

Proposition 2: The interference function $w(\cdot)$ defined in (30) is monotonically decreasing in the interval $(0, 1)$ if

$$\gamma < \frac{2}{n+1}. \quad (33)$$

The corresponding membership function (31) is monotonically increasing in the interval $(0, 1)$ if $\gamma < \frac{1}{n+1}$.

Proof: The proof of this proposition is presented in Appendix B.

The extended FALVQ 1, FALVQ 2, and FALVQ 3 families of algorithms can easily be implemented according to the scheme presented in Section II. If $\mathbf{x}$ is the input vector, the winning prototype $\mathbf{v}_i$ can be updated by (6) with w_{ir} evaluated in terms of the interference function $w(\cdot)$ as $w_{ir} = w(\|\mathbf{x}-\mathbf{v}_i\|^2/\|\mathbf{x}-\mathbf{v}_r\|^2)$. The nonwinning prototypes $\mathbf{v}_j \neq \mathbf{v}_i$ can be updated by (8) with n_{ij} evaluated in terms of the interference function $n(\cdot)$ as $n_{ij} = n(\|\mathbf{x}-\mathbf{v}_i\|^2/\|\mathbf{x}-\mathbf{v}_j\|^2)$.

V. Competition Measures

This section establishes a direct relationship between the properties of the membership functions used and the performance of the resulting FALVQ algorithms. This is accomplished by introducing two competition measures, which relate the form of the membership functions with the competition between the winning and nonwinning prototypes during the learning process.

According to Section II, the nonwinning prototypes are not updated to match the input vector if $u(x) = x$ or $u(x) = 0, \forall x \in (0, 1)$. It can be observed that

$$\int_0^1 u(x)\,dx = \begin{cases} \frac{1}{2} & \text{if } u(x) = x \\ 0 & \text{if } u(x) = 0\,. \end{cases} \tag{34}$$

For any other membership function selected according to the conditions presented in Section II

$$0 < \int_0^1 u(x)\,dx < \frac{1}{2}. \tag{35}$$

Thus, the area $A_u = \int_0^1 u(x)\,dx$ can be used as a measure of the competition between the winning and nonwinning prototypes. The development of competitive LVQ algorithms requires that $A_u \in (0, \frac{1}{2})$. Moreover, the nonwinning prototypes become less competitive as A_u approaches 0 or $\frac{1}{2}$. This measure can be used to evaluate the membership functions that resulted in the extended FALVQ 1, FALVQ 2, and FALVQ 3 families of algorithms by investigating the effect of the parameters involved in their definition on the competition between the winning and nonwinning prototypes during the learning process.

The extended FALVQ 1 family of algorithms is generated by membership functions of the form (21). If $n = 1$, then

$$A_u(1, \alpha) = \int_0^1 \frac{x dx}{1 + \alpha x} = \frac{1}{\alpha^2}(\alpha - \ln(1 + \alpha)). \tag{36}$$

It can be verified that $A_u(1, \alpha)$ approaches $\frac{1}{2}$ as α approaches zero. If α approaches infinity, then $A_u(1, \alpha)$ approaches zero. This is a clear indication that the competition between the winning and nonwinning prototypes during the learning process diminishes as α approaches zero or infinity. If $n \neq 1$, then

$$\begin{aligned} A_u(n, \alpha) &= \frac{1}{n\alpha} \int_0^1 \left(1 - \frac{1}{(1 + \alpha x)^n}\right) dx \\ &= \frac{1}{n\alpha}\left[1 - \frac{1}{\alpha(n-1)}\left(1 - \frac{1}{(1 + \alpha)^{n-1}}\right)\right]. \end{aligned} \tag{37}$$

Fig. 2(a) plots the measure $A_u(n, \alpha)$ as a function of α for different values of n. According to Fig. 2(a), $A_u(n, \alpha)$ attains values very close to $\frac{1}{2}$ for small values of α regardless of the value of n. In this case, the nonwinning prototypes are not updated to match the input vectors. As α increases, the value of $A_u(n, \alpha)$ decreases very slowly to zero, the other extreme value of this competition measure which indicates that the nonwinning prototypes are not updated to match the input vector.

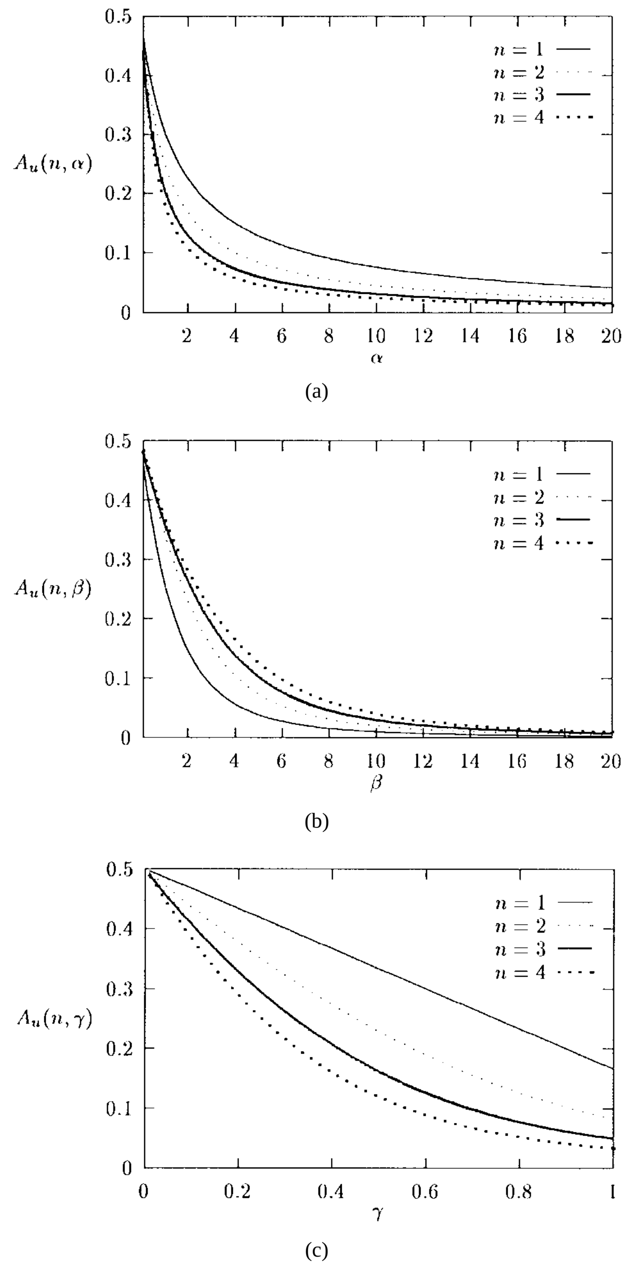


Fig. 2. (a) $A_u(n, \alpha)$ as a function of α for different values of n; (b) $A_u(n, \beta)$ as a function of β for different values of n; (c) $A_u(n, \gamma)$ as a function of γ for different values of n.

The extended FALVQ 2 family of algorithms is generated by membership functions of the form (26). In this case

$$\begin{aligned} A_u(n, \beta) &= \frac{1}{\beta} \int_0^1 \sum_{k=1}^{n} \frac{(\beta x)^k}{k!} e^{-\beta x} dx \\ &= \frac{1}{\beta^2} \sum_{k=1}^{n} \left(1 - e^{-\beta} \sum_{\ell=0}^{k} \frac{\beta^\ell}{\ell!}\right). \end{aligned} \tag{38}$$

If $n = 1$, (38) gives

$$A_u(1, \beta) = \frac{1}{\beta^2}\left(1 - (1 + \beta)e^{-\beta}\right). \tag{39}$$

It can be verified that $A_u(1, \beta)$ approaches $\frac{1}{2}$ as β approaches zero. If β approaches infinity, then $A_u(1, \beta)$ approaches zero.

The FALVQ 2 algorithms corresponding to $n = 1$ become increasingly competitive as β moves away from the extremes zero and infinity. Fig. 2(b) plots $A_u(n, \beta)$ as a function of β for different values of n. According to Fig. 2(b), $A_u(n, \beta)$ decreases quickly to values close to zero for all values of n as the value of β increases. Thus, the competition between the winning and nonwinning prototypes diminishes quickly as the value of β exceeds a certain threshold.

The extended FALVQ 3 family of algorithms is generated by membership functions of the form (31). In this case,

$$\begin{aligned} A_u(n, \gamma) &= \int_0^1 x(1-\gamma x)^n dx \\ &= \frac{1}{(n+1)(n+2)\gamma^2}\left[1-(1-\gamma)^{n+1}(1+\gamma+n\gamma)\right]. \end{aligned} \quad (40)$$

If $n = 1$, then (40) gives

$$A_u(1, \gamma) = \frac{1}{6}(3 - 2\gamma). \quad (41)$$

Clearly, $A_u(1, \gamma)$ attains its maximum value $\frac{1}{2}$ for $\gamma = 0$, which corresponds to no competition, and decreases linearly from $\frac{1}{2}$ to $\frac{1}{6}$ as γ spans the interval $(0, 1)$. Fig. 2(c) plots $A_u(n, \gamma)$ as a function of $\gamma \in (0, 1]$ for different values of n. For $n = 1$, $A_u(n, \gamma)$ decreases linearly from $\frac{1}{2}$ to $\frac{1}{6}$. However, $A_u(n, \gamma)$ decreases much faster for higher values of n. Thus, even for values of γ close to zero, higher values of n allow the nonwinning prototypes to be updated to match the input vector.

The area A_u alone may not be sufficient to establish a relationship between the form of the membership function and the competition between the winning and nonwinning prototypes during the learning process. This can be accomplished by considering the area A_u in conjunction with the "centroid" or "center of gravity" of the membership function $u(\cdot)$. Assuming that $A_u = \int_0^1 u(x)\,dx \neq 0$, the centroid of $u(x)$ over the interval $x \in (0, 1)$ is defined as

$$C_u = \frac{\int_0^1 x\,u(x)\,dx}{\int_0^1 u(x)\,dx}. \quad (42)$$

The centroid (42) is a useful source of information regarding the shape of $u(\cdot)$ and, thus, the bias of the resulting FALVQ algorithm toward the winning prototype. In the extreme case where $u(x) = x$, $C_u = \frac{2}{3}$. If $u(\cdot)$ is an admissible membership function, then $C_u < \frac{2}{3}$. Since the selection of $u(x) = x$ implies that the nonwinning prototypes are not updated to match the input vector, the development of competitive FALVQ algorithms requires a membership function that corresponds to a centroid value lower than $\frac{2}{3}$. Nevertheless, the nonwinning prototypes become increasingly competitive as the centroid C_u decreases below $\frac{1}{2}$. If the value of C_u is sufficiently close to zero, the competition between the winning and nonwinning prototypes diminishes.

The centroid of the membership function that resulted in the FALVQ 1 family of algorithms can be obtained from (42) with $u(x) = x\,(1 + \alpha x)^{-1}$ as

$$C_u(1, \alpha) = \frac{1}{2}\left(\frac{\alpha}{\alpha - \ln(1+\alpha)} - \frac{2}{\alpha}\right). \quad (43)$$

If α approaches zero, then $C_u(1, \alpha)$ approaches its maximum value, i.e., $\lim_{\alpha \to 0} C_u(1, \alpha) = \frac{2}{3}$. This is consistent with the fact that if α approaches zero, then $u(x) = x\,(1 + \alpha x)^{-1}$ approaches x. It can also be verified that $\lim_{\alpha \to \infty} C_u(1, \alpha) = \frac{1}{2}$. Fig. 3(a) plots $C_u = C_u(n, \alpha)$ as a function of α for different values of n. As the value of α increases from zero to infinity, $C_u(n, \alpha)$ decreases asymptotically from its maximum value of $\frac{2}{3}$ to $\frac{1}{2}$, its lower bound. According to Fig. 3(a), $C_u(n, \alpha)$ remains almost constant as α increases from zero to infinity and is practically not affected by the value of n. Thus, with the exemption of values of α sufficiently close to zero, the area $A_u = A_u(n, \alpha)$ is a more reliable competition measure for the extended FALVQ 1 family of algorithms.

The centroid of the membership function that resulted in the FALVQ 2 family of algorithms can be obtained from (42) with $u(x) = x \exp(-\beta x)$ as

$$C_u(1, \beta) = \frac{1}{\beta}\frac{2 - (\beta^2 + 2\beta + 2)e^{-\beta}}{1 - (\beta + 1)e^{-\beta}}. \quad (44)$$

If β approaches zero, then $C_u(1, \beta)$ approaches its maximum value, i.e., $\lim_{\beta \to 0} C_u(1, \beta) = \frac{2}{3}$. It can also be verified that $\lim_{\beta \to \infty} C_u(1, \beta) = 0$. Fig. 3(b) plots $C_u = C_u(n, \beta)$ as a function of β for different values of n. Clearly, the centroid $C_u(n, \beta)$ can take positive values significantly lower than $\frac{1}{2}$ for large values of β. Such values of $C_u(n, \beta)$ indicate that there is practically no competition between the prototypes during the learning process. In conjunction with the area $A_u = A_u(n, \beta)$, $C_u(n, \beta)$ can be used to select the range of values of β that guarantee the competition between the winning and nonwinning prototypes.

The centroid of the membership function that resulted in the FALVQ 3 family of algorithms can be obtained from (42) with $u(x) = x\,(1 - \gamma x)$ as

$$C_u(1, \gamma) = \frac{1}{2}\frac{4 - 3\gamma}{3 - 2\gamma}. \quad (45)$$

Clearly, $C_u(1, \gamma)$ decreases from $\frac{2}{3}$ to $\frac{1}{2}$ as the value of γ increases from zero to one. Fig. 3(c) plots $C_u = C_u(n, \gamma)$ as a function of $\gamma \in (0, 1]$ for different values of n. In this case, the competition measure $C_u(n, \gamma)$ decreases slowly to values considerably higher than zero. Since $C_u(n, \gamma)$ takes values in a neighborhood of $\frac{1}{2}$ as γ spans the interval $(0, 1)$, $C_u(n, \gamma)$ is not a particularly informative competition measure in this case. Thus, the area $A_u = A_u(n, \gamma)$ can be used for selecting the values of γ that result in competitive FALVQ 3 algorithms.

VI. Experimental Results

A. Clustering the IRIS Data

The proposed algorithms were tested using Anderson's IRIS data set [1], which has extensively been used for evaluating the performance of pattern clustering algorithms. This data set contains 150 feature vectors of dimension 4 which belong to

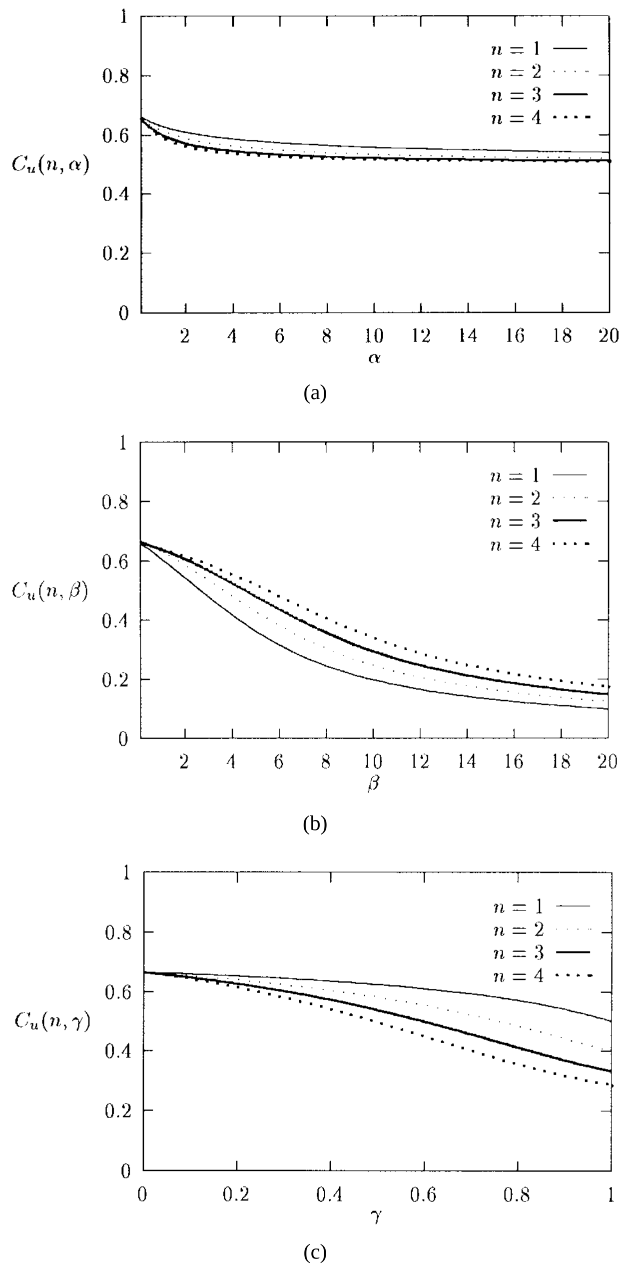


Fig. 3. (a) $C_u(n,\alpha)$ as a function of α for different values of n; (b) $C_u(n,\beta)$ as a function of β for different values of n; (c) $C_u(n,\gamma)$ as a function of γ for different values of n.

three physical classes representing different IRIS subspecies. Each class contains 50 feature vectors. One of the three classes is well separated from the other two, which are not easily separable due to the overlapping of their vectors. The performance of the algorithms is evaluated by counting the number of crisp clustering errors, i.e., the number of feature vectors that are assigned to a wrong physical cluster by terminal nearest prototype partitions of the data. Unsupervised clustering of the IRIS data typically results in 12–17 clustering errors [20].

Tables II–IV show the number of feature vectors from the IRIS data set assigned to a wrong cluster by algorithms from the extended FALVQ 1, FALVQ 2, and FALVQ 3 families, respectively. In all these experiments the total number of iterations was $N = 100$ while the initial value of the learning rate was $\eta_0 = 0.5$. The prototypes were initialized with all zero values. According to Table II, the extended FALVQ 1 algorithms resulted in a large number of clustering errors for very small values of α. Note that when 100 or 50 clustering errors are observed, the algorithm assigns all feature vectors to one or two clusters, respectively. The number of clustering errors increased slightly for values of α above ten. This experimental outcome is consistent with the fact that the nonwinning prototypes are not updated to match the input vector for very small or very large values of α. Nevertheless, there is a very broad range of values of α for which the extended FALVQ 1 algorithms resulted in an acceptable number of clustering errors. The performance of the extended FALVQ 1 algorithms tested on this data set was not significantly affected by the value of n. This experimental outcome is consistent with the behavior of the competition measures $A_u(n,\alpha)$ and $C_u(n,\alpha)$ for different values of α and n. According to Table III, the algorithms from the extended FALVQ 2 family resulted in a large number of clustering errors for very small values of β. Moreover, the performance of these algorithms deteriorated as the value of β increased above two. In this case, n had a rather significant effect on the performance of the algorithms, especially for high values of β. The performance of the algorithms from the extended

TABLE II
NUMBER OF FEATURE VECTORS FROM THE IRIS DATA SET ASSIGNED TO A WRONG CLUSTER BY ALGORITHMS FROM THE EXTENDED FALVQ 1 FAMILY WITH $N = 100$ AND $\eta_0 = 0.5$

α	.001	0.01	0.1	0.3	0.5	0.7	1.0	2.0	5.0	10.0	100.0
$n = 1$	100	99	16	16	16	16	16	16	16	16	17
$n = 2$	100	50	16	16	16	16	16	16	16	16	17
$n = 3$	100	50	16	16	16	16	16	16	16	16	17
$n = 4$	100	50	16	16	16	16	16	16	16	17	17

TABLE III
NUMBER OF FEATURE VECTORS FROM THE IRIS DATA SET ASSIGNED TO A WRONG CLUSTER BY ALGORITHMS FROM THE EXTENDED FALVQ 2 FAMILY WITH $N = 100$ AND $\eta_0 = 0.5$

β	.001	0.01	0.1	0.3	0.5	0.7	1.0	2.0	3.0	5.0	10.0
$n = 1$	100	99	16	16	16	16	16	16	16	16	16
$n = 2$	100	99	16	16	16	16	16	16	38	39	27
$n = 3$	100	99	16	16	16	16	16	16	15	55	37
$n = 4$	100	99	16	16	16	16	17	17	100	100	35

TABLE IV
NUMBER OF FEATURE VECTORS FROM THE IRIS DATA SET ASSIGNED TO A WRONG CLUSTER BY ALGORITHMS FROM THE EXTENDED FALVQ 3 FAMILY WITH $N = 100$ AND $\eta_0 = 0.5$

γ	.001	0.01	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
$n = 1$	100	99	16	16	16	16	16	16	16	16	16	16
$n = 2$	100	50	16	16	16	16	100	16	60	46	16	16
$n = 3$	100	16	16	16	16	16	16	16	16	16	16	16
$n = 4$	100	16	16	16	16	16	16	16	16	16	16	16

FALVQ 2 family is consistent with the behavior of the corresponding competition measures $A_u(n, \beta)$ and $C_u(n, \beta)$. Table IV indicates that the performance of the algorithms from the extended FALVQ 3 family deteriorated for values of γ sufficiently close to zero. Nevertheless, there was no significant change in the number of feature vectors assigned to a wrong cluster as γ approached one. The performance of these algorithms was affected by the value of n only for small values of γ. For example, in the case where $\gamma = 0.01$ the algorithm resulted in an acceptable number of clustering errors for $n = 2$ and $n = 3$. This is consistent with the behavior of the competition measure $A_u(n, \gamma)$ for different values of γ and n.

B. Segmentation of Magnetic Resonance Images

The clinical utility of *magnetic resonance* (MR) imaging rests on the contrasting image intensities obtained for different tissue types, both normal and abnormal. For a given MR image pulse sequence, image intensities will depend on local values of the following relaxation parameters: the spin-lattice relaxation time (T1), the spin-spin relaxation time (T2), and the spin density (SD). Conventional diagnosis based on MR imaging requires the simultaneous visual inspection of up to three or more different weighted MR images. Given the redundancy present in MR images, their interpretation is based on intelligent abstraction. In this context, abstraction means the ability to concentrate on some key details of the image such as unusually high intensity levels that may correspond to abnormalities.

In the context of MR imaging, segmentation usually implies the creation of a single image with much fewer intensity levels than the original images. The resulting segmented image is frequently artificially colored in order to facilitate the diagnostic process. In some cases, the objective of the segmentation process is the characterization of brain tissue reflected in different positions of the MR images. The existence of reliable computer-based MR image segmentation techniques can enhance the ability of radiologists to detect, diagnose, and monitor diseased pathology. MR image segmentation techniques are often evaluated in terms of their ability to 1) differentiate between cerebro-spinal fluid (CSF), white matter, and gray matter, and 2) differentiate between normal tissues and abnormalities. Another important criterion for evaluating MR image segmentation techniques is their ability to quantitatively measure changes in brain tissue volumes caused by degenerative brain diseases.

The use of fuzzy clustering procedures in MR image segmentation is justified by the fact that there are no hard boundaries in MR images of the brain due to tissue mixing [5]. Hall *et al.* [5] compared MR image segmentation techniques based on supervised multilayered neural networks [17], the fuzzy c-means algorithm [2], and approximations of the fuzzy c-means algorithm that were developed to reduce its computational requirements. Although the supervised training of multilayered neural networks was computationally demanding, supervised and unsupervised segmentation techniques provided broadly similar results. Inconsistency of rating among experts was observed in a complex segmentation problem with tumor/edema or CSF boundary, where tissues have similar MR relaxation behavior [5].

The segmentation of MR images is conventionally formulated as the problem of clustering a set of feature vectors. Each feature vector contains as elements the T1, T2, and SD parameters. A clustering procedure is used to assign the feature vectors to a relatively small number of clusters, each represented by a prototype. Following the clustering process, the segmented image is obtained by representing each feature vector by the corresponding prototype. As a result, the segmented image contains a number of intensity levels equal to the number of clusters, which is smaller than the number of intensity levels in the original image. The utility of segmented MR images in the medical diagnostic process depends on the combination of two often conflicting requirements, that is, the elimination of the redundant information present in the original MR images and the preservation of the important details in the resulting segmented images. The discrimination between redundant and useful information is based on the number of intensity levels present in the segmented images, or, equivalently, the number of clusters created during the clustering process. The selection of a small number of clusters can result in the loss of detail necessary for the diagnostic process, while the selection of a large number of intensity levels can undermine the effectiveness of the segmentation process by producing segmented images with a large volume of redundant information.

Fig. 4(a)–(c) shows the T1-weighted, T2-weighted, and spin density MR images of an individual with meningioma. Meningiomas are the most common form of intracranial tumors. In this case, the tumor was located in the right frontal lobe (upper-left quarter of the MR images) and appears bright on the T2-weighted image and dark on the T1-weighted image. The tumor appears very bright and isolated from surrounding tissue in Fig. 4(d), which shows the T1-weighted MR image recorded after the patient was given Gadolinium. There is also a large amount of edema surrounding the tumor, which appears very bright on the T2-weighted image shown in Fig. 4(b).

The MR image shown in Fig. 4 was segmented using the (unlabeled data) LVQ algorithm and algorithms from the FALVQ 1, FALVQ 2, and FALVQ 3 families. In these experiments, the feature vectors were formed using the pixel values of the T1-weighted, T2-weighted, and spin density images shown in Fig. 4(a)–(c), respectively. Fig. 4(d), which shows the T1-weighted image with Gadolinium, was used to evaluate the segmented images since the tumor appears very bright and is well separated from surrounding tissue. In all these experiments, $c = 8$, that is, the segmented images contained 8 different intensity levels which were artificially colored. Fig. 5(a) and (b) shows the segmented images produced by the LVQ algorithm applied with $N = 100$ and initial values of the learning rate $\eta_0 = 0.1$ and $\eta_0 = 0.9$, respectively. It was found that the algorithm achieves its best performance for initial values of the learning rate in this range. Clearly, the LVQ algorithm succeeds in identifying the edema but fails to separate the tumor from surrounding tissue. Fig. 6(a) and (b) shows the segmented images produced by the algorithms from the extended FALVQ 1 family for $n = 1$

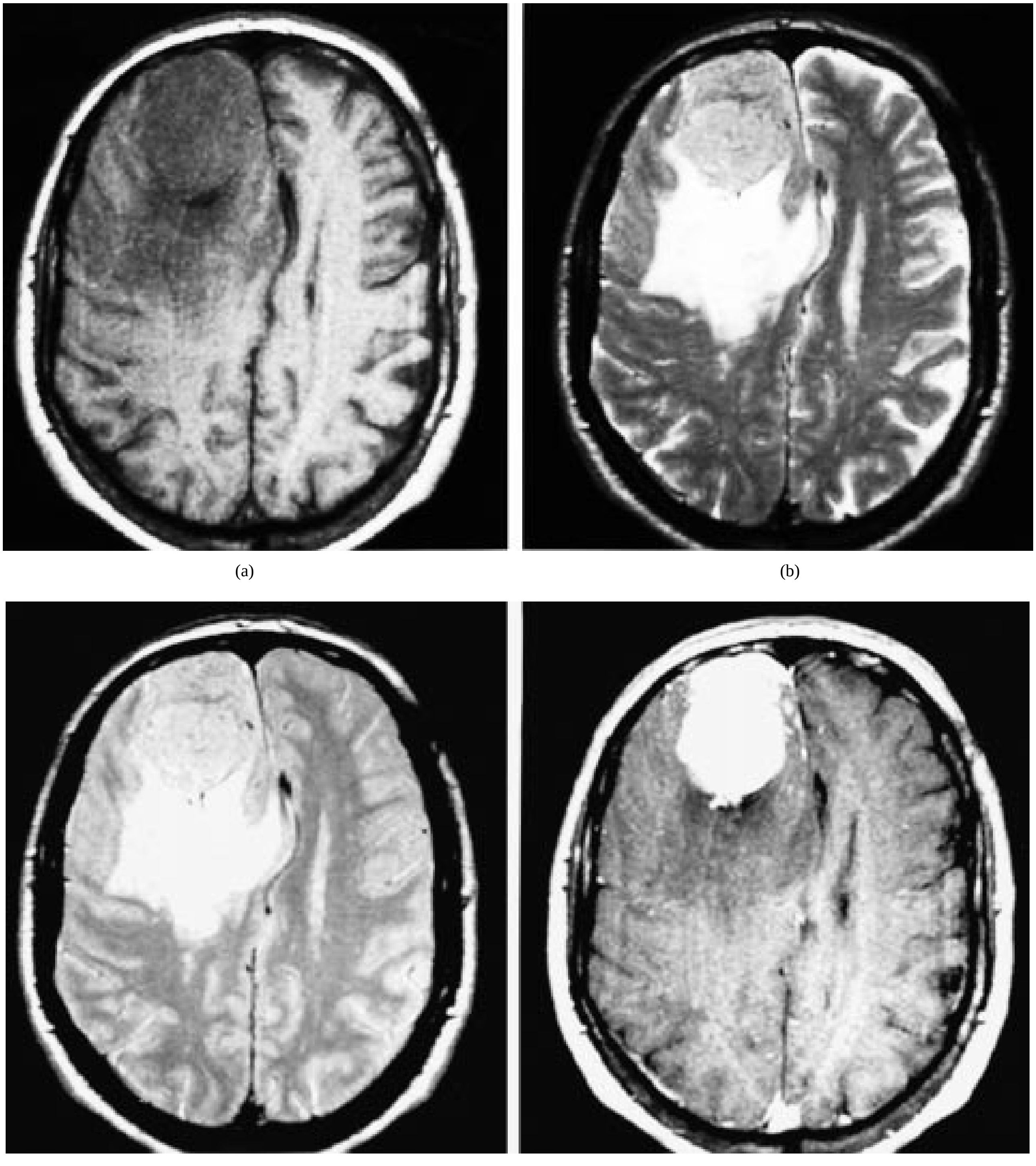
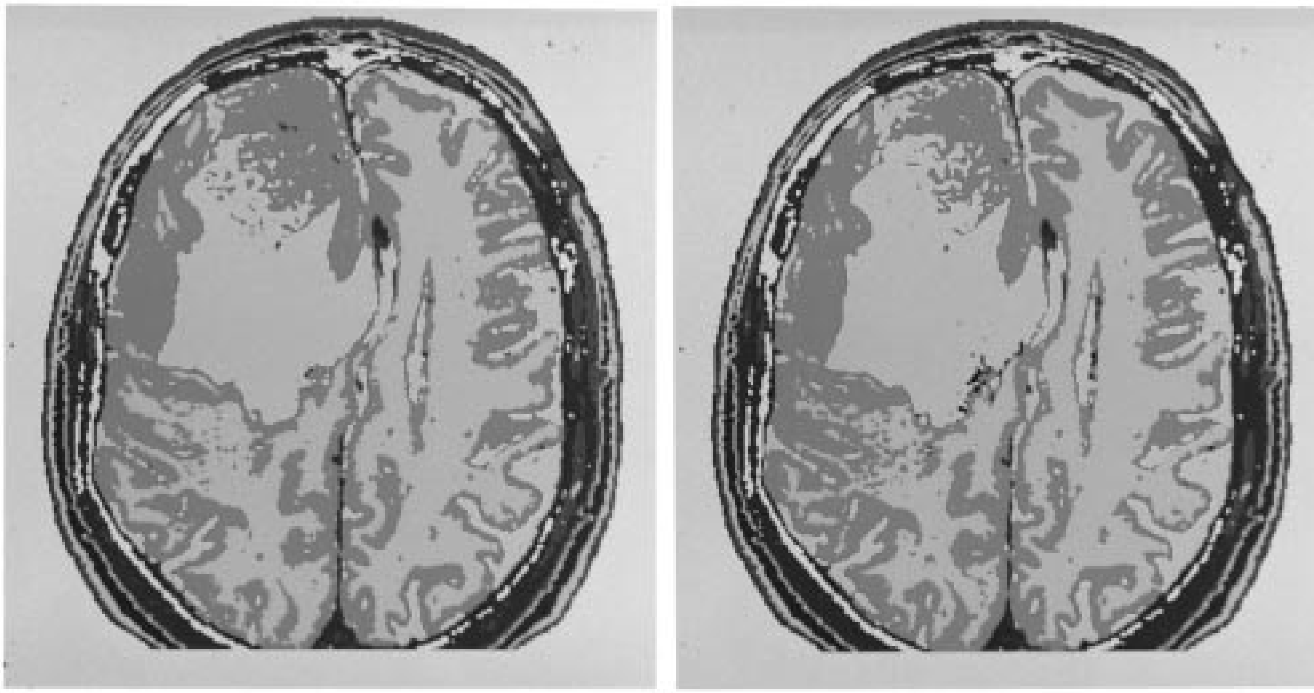


Fig. 4. Magnetic resonance (MR) image of the brain of an individual suffering from meningioma: (a) T1-weighted image, (b) T2-weighted image, (c) spin density image, and (d) T1-weighted image after the patient was given Gadolinium.

with $\alpha = 1$ and $\alpha = 0.1$, respectively. Fig. 7(a) and (b) shows the segmented images produced by the algorithms from the extended FALVQ 2 family for $n = 1$ with $\beta = 1$ and $\beta = 0.1$, respectively. Fig. 8(a) and 8(b) shows the segmented images produced by the algorithms from the extended FALVQ 3 family for $n = 1$ with $\gamma = 1$ and $\gamma = 0.1$, respectively. In all these experiments, the initial value of the learning rate was $\eta_0 = 0.001$ and the total number of iterations was $N = 100$. Clearly, the tumor and the surrounding edema are clearly identified by the algorithm from the extended FALVQ 1 family with $n = 1$ and $\alpha = 1$ (competition measures: $A_u = 0.306, C_u = 0.629$), the algorithm from the extended FALVQ 2 family with $n = 1$ and $\beta = 1$ (competition measures: $A_u = 0.264, C_u = 0.608$), and the algorithm from the extended FALVQ 3 family with $n = 1$ and $\gamma = 1$ (competition measures: $A_u = 0.167, C_u = 0.5$). However, the



(a) (b)

Fig. 5. Segmented MR images by the (unlabeled data) LVQ algorithm with $N = 100$and (a) $\eta_0 = 0.1$, (b) $\eta_0 = 0.9$.

tumor was not distinguished from surrounding tissue by the algorithm from the extended FALVQ 1 family with $n = 1$ and $\alpha = 0.1$ (competition measures: $A_u = 0.469, C_u = 0.661$), the algorithm from the extended FALVQ 2 family with $n = 1$ and $\beta = 0.1$ (competition measures: $A_u = 0.468, C_u = 0.661$), and the algorithm from the extended FALVQ 3 family with $n = 1$ and $\gamma = 0.1$ (competition measures: $A_u = 0.467, C_u = 0.661$). As indicated by the values of the corresponding competition measures, for these values of α, β, γ and n the nonwinning prototypes are not significantly updated to match the input vector of the network. In fact, for these values of α, β, γ and n the algorithms from the extended FALVQ 1, FALVQ 2, and FALVQ 3 families resemble the behavior of Kohonen's (unlabeled data) LVQ algorithm.

VII. Conclusions

This paper presented a new methodology for constructing FALVQ algorithms, which exploits the fact that the competition between the winning and nonwinning prototypes during the learning process is regulated by the interference functions. According to this methodology, the development of FALVQ algorithms begins with the selection of the interference function instead of the membership function. The proposed methodology allows for a more direct impact on the competition between the prototypes during the learning process and can be the basis for extending the existing families of FALVQ algorithms. This paper also introduced two quantitative measures that establish a relationship between the formulation that led to FALVQ algorithms and the competition between the prototypes during the learning process. The proposed competition measures can be used for selecting the parameters of FALVQ algorithms. Various algorithms from the extended FALVQ 1, FALVQ 2, and FALVQ 3 families were experimentally tested on the IRIS data set. In addition, FALVQ algorithms were used to perform segmentation of MR images of the brain. These experiments verified the validity of the proposed competition measures by testing the performance of FALVQ algorithms in the limit where they resemble the (unlabeled data) LVQ algorithm. These experiments also illustrated the efficiency of FALVQ algorithms used to perform the nontrivial vector quantization task involved in this application.

Appendix A
Proof of Proposition 1

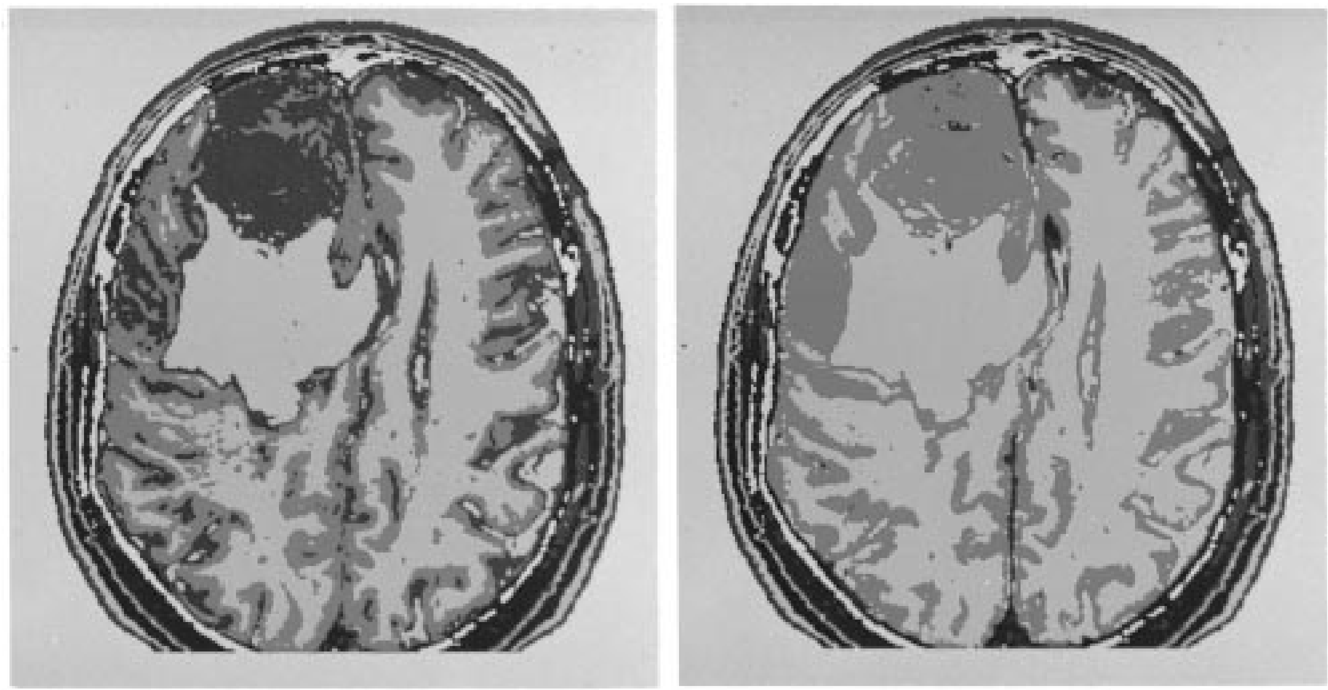
The extrema of the interference function (25) satisfy the condition

$$\frac{dw(x)}{dx} = \beta e^{-\beta x}\left(\frac{(\beta x)^n}{n!} - \frac{(\beta x)^{n-1}}{(n-1)!} - 1\right) = 0. \qquad \text{(A1)}$$

Since $\beta \neq 0$, the values of x which correspond to extrema of $w(x)$ can be found by solving the equation

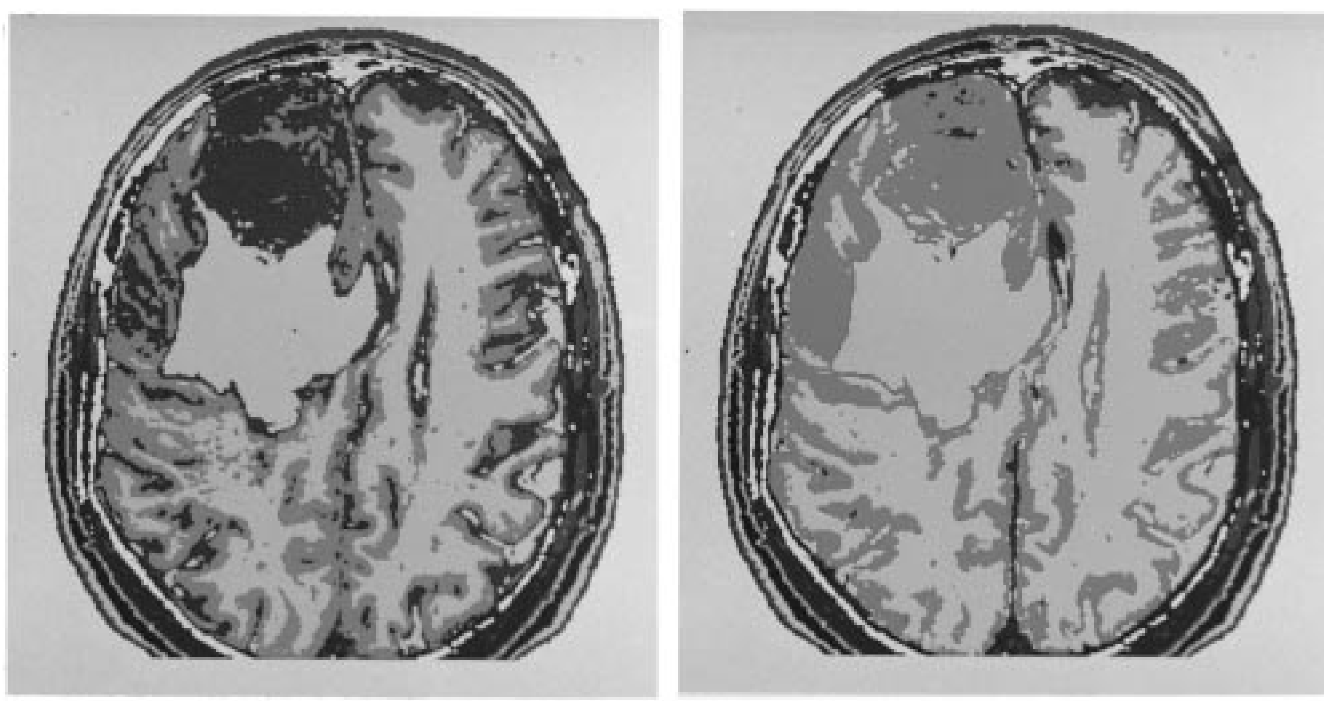
$$\frac{z^n}{n!} - \frac{z^{n-1}}{(n-1)!} - 1 = 0 \qquad \text{(A2)}$$

where $z = \beta x$. For $n = 1$, the only root of (A2) is $z_0 = 2$. For $n = 2$, the only positive root of (A2) is $z_0 = 1 + \sqrt{3} \in (1, 2)$. If $n \geq 3$, the roots of (A2) can be determined numerically. Nevertheless, it is shown here that, in the case were $n > 1$, (A2) has only one positive root $z_0 \in (n, n+1)$. Moreover, z_0 approaches asymptotically n from the right as the value of n increases.



(a) (b)

Fig. 6. Segmented MR images by algorithms from the extended FALVQ 1 family with $n = 1$ and (a) $\alpha = 1$, (b) $\alpha = 0.1$ ($N = 100$, $\eta_0 = 0.001$).



(a) (b)

Fig. 7. Segmented MR images by algorithms from the extended FALVQ 2 family with $n = 1$ and (a) $\beta = 1$, (b) $\beta = 0.1$ ($N = 100$, $\eta_0 = 0.001$).

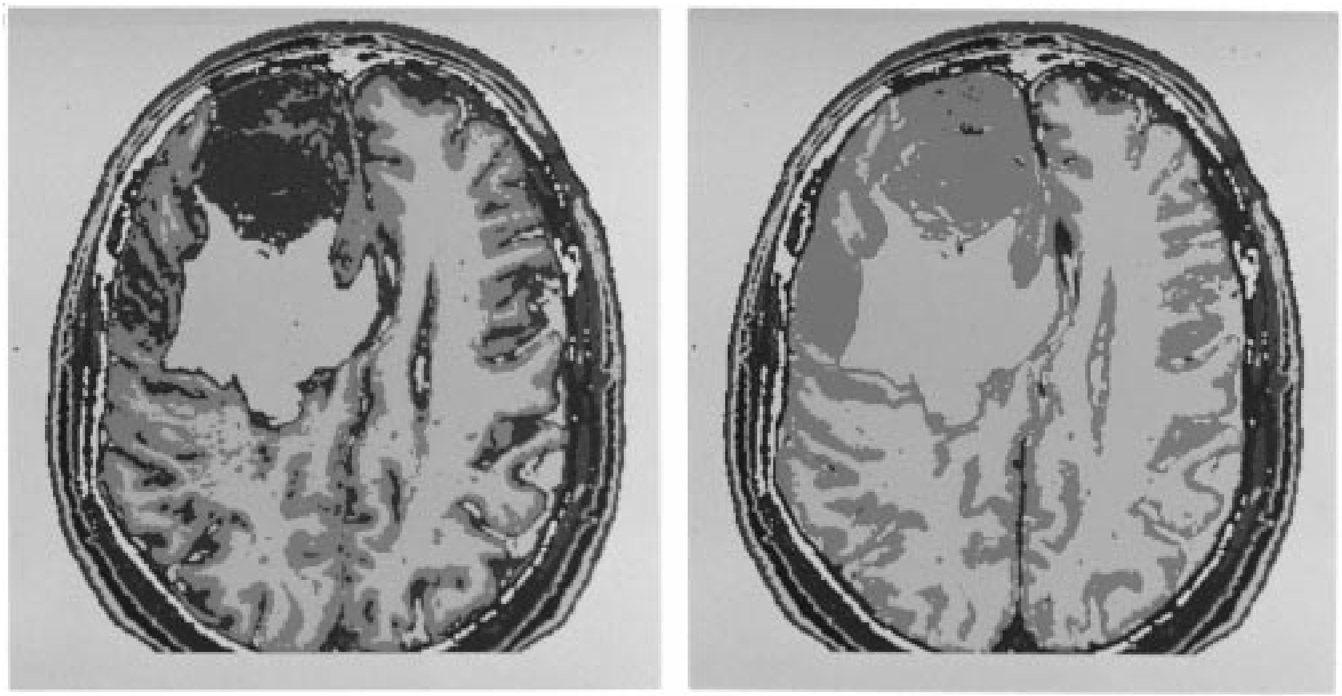
Consider the function

$$f(z) = \frac{z^n}{n!} - \frac{z^{n-1}}{(n-1)!} - 1 = \frac{z^{n-1}}{(n-1)!}\left(\frac{z}{n} - 1\right) - 1. \quad \text{(A3)}$$

Clearly, $f(0) = -1 \neq 0$. Thus, $z = 0$ is not a root of (A2). If $0 < z < n$, then $\frac{z}{n} - 1 < 0$. Since $z > 0$, $f(z) < 0$, $\forall z \in (0, n)$. Thus, there is no root of (A2) in the interval $[0, n)$.

Consider that $z > n + 1$. Since $\frac{z}{n} - 1 > \frac{1}{n}$, then

$$f(z) = \frac{z^{n-1}}{(n-1)!}\left(\frac{z}{n} - 1\right) - 1 > \frac{z^{n-1}}{n!} - 1 > \frac{(n+1)^{n-1}}{n!} - 1 > 0, \ \forall n > 1. \quad \text{(A4)}$$



(a) (b)

Fig. 8. Segmented MR images by algorithms from the extended FALVQ 3 family with $n = 1$ and (a) $\gamma = 1$, (b) $\gamma = 0.1$ ($N = 100$, $\eta_0 = 0.001$).

Thus, there is no root of (A2) in the interval $(n+1, \infty)$. For $z = n$, $f(n) = -1 < 0$. For $z = n+1$,

$$f(n+1) = \frac{(n+1)^{n-1}}{(n-1)!}\left(\frac{n+1}{n} - 1\right) - 1 = \frac{(n+1)^{n-1}}{n!} - 1 > 0, \forall n > 1. \quad \text{(A5)}$$

Since $f(z)$ changes sign as z takes values in the interval $[n, n+1]$, there exists at least one zero-crossing in this interval. It can easily be verified that the equation

$$f'(z) = \frac{z^{n-2}}{(n-2)!}\left(\frac{z}{n-1} - 1\right) = 0 \quad \text{(A6)}$$

has no roots in the interval $[n, n+1]$. This implies that $f(z)$ has no extrema in this interval. Thus, the zero-crossing is unique. As the value of n increases, $f(n) = -1$ while $f(n+1) = \frac{1}{n!}(n+1)^{n-1} - 1$ increases exponentially. Thus, the zero-crossing approaches asymptotically n from the right.

According to the above analysis, the positive root of $w'(x) = 0$ is determined by $\beta x = n + \mu$, where $\mu \in (0,1)$. Also, μ approaches zero as n increases. Therefore, $w(x)$ has a minimum at $x = \frac{1}{\beta}(n+\mu)$. In conclusion, $w(x)$ is a monotonically decreasing function in the interval $(0,1)$ if $\frac{1}{\beta}(n+\mu) > 1$ or, equivalently, if $\beta < n+\mu$. Since $\mu \in (0,1)$, this condition is satisfied if $\beta < n$.

Since $u'(x) = w(x)$, it can easily be verified that $u(x)$ possesses a maximum at $x = \frac{1}{\beta}(n!)^{\frac{1}{n}}$, which is the root of $(\beta x)^n - n! = 0$. Thus, $u(x)$ is a monotonically increasing function in the interval $(0,1)$ if $\frac{1}{\beta}(n!)^{\frac{1}{n}} > 1$ or, equivalently, if $\beta < (n!)^{\frac{1}{n}}$.

Appendix B
Proof of Proposition 2

The extrema of the interference function $w(\cdot)$ defined in (30) satisfy $dw(x)/dx = 0$. Since

$$\frac{dw(x)}{dx} = -\gamma n(1-\gamma x)^{n-2}[2 - (n+1)\gamma x] \quad \text{(B1)}$$

the extrema of $w(x)$ occur at $x = \frac{1}{\gamma}$ and $x = \frac{1}{\gamma}\frac{2}{n+1} < \frac{1}{\gamma}$. The second-order derivative of $w(x)$ is

$$\frac{d^2w(x)}{dx^2} = \gamma^2 n(n-1)(1-\gamma x)^{n-3}[3 - (n+1)\gamma x]. \quad \text{(B2)}$$

Clearly, $d^2w(x)/dx^2|_{x=\frac{1}{\gamma}} = 0$. In addition, for $n > 1$,

$$\frac{d^2w(x)}{dx^2}\Big|_{x=\frac{1}{\gamma}\frac{2}{n+1}} = \gamma^2 n(n-1)\left(\frac{n-1}{n+1}\right)^{n-3} > 0. \quad \text{(B3)}$$

Thus, $w(x)$ possesses a minimum at $x = \frac{1}{\gamma}\frac{2}{n+1}$. As a result, the interference function $w(x)$ is a monotonically decreasing function of x over the interval $(0,1)$ if $\frac{1}{\gamma}\frac{2}{n+1} > 1$ or, equivalently, if

$$\gamma < \frac{2}{n+1}. \quad \text{(B4)}$$

For $n = 1$, $w(x)$ decreases linearly for any $\gamma < 1$. Thus, the condition (B4) is valid for all $n \geq 1$.

The extrema of $u(\cdot)$ occur at $du(x)/dx = w(x) = 0$, that is, at $x = \frac{1}{\gamma}$ and $x = \frac{1}{\gamma}\frac{1}{n+1}$. It can easily be verified that $\frac{d^2u(x)}{dx^2}|_{x=\frac{1}{\gamma}} = \frac{dw(x)}{dx}|_{x=\frac{1}{\gamma}} = 0$. In addition, for $n \geq 1$,

$$\frac{d^2u(x)}{dx^2}\Big|_{x=\frac{1}{\gamma}\frac{1}{n+1}} = \frac{dw(x)}{dx}\Big|_{x=\frac{1}{\gamma}\frac{1}{n+1}} = -\gamma n\left(\frac{n}{n+1}\right)^{n-2} < 0. \quad \text{(B5)}$$

Thus, the membership function $u(\cdot)$ possesses a maximum at $x = \frac{1}{\gamma}\frac{1}{n+1}$. In conclusion, $u(x)$ is a monotonically increasing function in the interval $(0, 1)$ if $\frac{1}{\gamma}\frac{1}{n+1} > 1$ or, equivalently, if

$$\gamma < \frac{1}{n+1}. \tag{B6}$$

References

[1] E. Anderson, "The IRISes of the Gaspe Peninsula," *Bull. Amer. IRIS Soc.*, vol. 59, pp. 2–5, 1939.

[2] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum, 1981.

[3] J. C. Bezdek and N. R. Pal, "Two soft relatives of learning vector quantization," *Neural Networks*, vol. 8, no. 5, pp. 729–743, 1995.

[4] R. M. Gray, "Vector quantization," *IEEE ASSP Mag.*, vol. 1, no. 2 pp. 4–29, Apr. 1984.

[5] L. O. Hall, A. M. Bensaid, L. P. Clarke, R. P. Velthuizen, M. S. Silbiger, and J. C. Bezdek, "A comparison of neural network and fuzzy clustering techniques in segmenting magnetic resonance images of the brain," *IEEE Trans. Neural Networks*, vol. 3, pp. 672–682, 1992.

[6] R. J. Hathaway and J. C. Bezdek, "Optimization of clustering criteria by reformulation," *IEEE Trans. Fuzzy Syst.*, vol. 3, pp. 241–246, 1995.

[7] T. Huntsberger and P. Ajjimarangsee, "Parallel self-organizing feature maps for unsupervised pattern recognition," *Int. J. General Syst.*, vol. 16, pp. 357–372, 1989.

[8] N. B. Karayiannis, "Generalized fuzzy k-means algorithms and their application in image compression," in *SPIE Proc. vol. 2493: Applicat. Fuzzy Logic Technol. II*, Orlando, FL, Apr. 19–21, 1995, pp. 206–217.

[9] ——, "Generalized fuzzy c-means algorithms," submitted to *IEEE Trans. Syst., Man, Cybern*.

[10] N. B. Karayiannis and J. C. Bezdek, "An integrated approach to fuzzy learning vector quantization and fuzzy c-means clustering," *IEEE Trans. Fuzzy Syst.*, in press, 1997.

[11] N. B. Karayiannis, J. C. Bezdek, N. R. Pal, R. J. Hathaway, and P.-I Pai, "Repairs to GLVQ: A new family of competitive learning schemes," *IEEE Trans. Neural Networks*, vol. 7, pp. 1062–1071, 1996.

[12] N. B. Karayiannis and P.-I Pai, "A family of fuzzy algorithms for learning vector quantization," in *Intelligent Engineering Systems Through Artificial Neural Networks, vol. 4*, C. H. Dagli *et al.*, Eds. New York: ASME Press, 1994, pp. 219–224.

[13] ——, "Fuzzy algorithms for learning vector quantization: Generalizations and extensions," in *SPIE Proc. vol. 2492: Applicat. Sci. of Artificial Neural Networks*, Orlando, FL, Apr. 17–21, 1995, pp. 264–274.

[14] ——, "A family of fuzzy algorithms for learning vector quantization," submitted to *IEEE Trans. Neural Networks*.

[15] ——, "Fuzzy algorithms for learning vector quantization," *IEEE Trans. Neural Networks*, vol. 7, pp. 1196–1211, 1996.

[16] N. B. Karayiannis and M. Ravuri, "An integrated approach to fuzzy learning vector quantization and fuzzy c-means clustering," in *Intelligent Engineering Systems Through Artificial Neural Networks, vol. 5*, C. H. Dagli *et al.*, Eds. New York: ASME Press, 1995, pp. 247–252.

[17] N. B. Karayiannis and A. N. Venetsanopoulos, *Artificial Neural Networks: Learning Algorithms, Performance Evaluation, and Applications*. Boston, MA: Kluwer, 1993.

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

[19] ——, "The self-organizing map," *Proc. IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.

[20] N. R. Pal, J. C. Bezdek, and E. C.-K. Tsao, "Generalized clustering networks and Kohonen's self-organizing scheme," *IEEE Trans. Neural Networks*, vol. 4, pp. 549–557, 1993.

[21] E. C. -K. Tsao, J. C. Bezdek, and N. R. Pal, "Fuzzy Kohonen clustering networks," *Pattern Recognition*, vol. 27, no. 5, pp. 757–764, 1994.

[22] Y. Z. Tsypkin, *Foundations of the Theory of Learning*. New York: Academic, 1973.

Nicolaos B. Karayiannis (S'85–M'91) was born in Greece on January 1, 1960. He received the Diploma degree in electrical engineering from the National Technical University of Athens in 1983, and the M.A.Sc. and Ph.D. degrees in electrical engineering from the University of Toronto in 1987 and 1991, respectively.

He is currently an Assistant Professor in the Department of Electrical Engineering, University of Houston, TX. From 1984 to 1991 he worked as a Research and Teaching Assistant at the University of Toronto, Canada. From 1983 to 1984 he was a Research Assistant at the Nuclear Research Center "Democritos," Athens, Greece, where he was engaged in research on multidimensional signal processing. He has published more than 60 papers, including 25 in technical journals, and is the coauthor of the book *Artificial Neural Networks: Learning Algorithms, Performance Evaluation, and Applications* (Boston, MA: Kluwer, 1993). His current research interests include supervised and unsupervised learning, applications of artificial neural networks in image processing and analysis, applications of fuzzy logic in neural modeling, and learning vector quantization.

Dr. Karayiannis is a member of the International Neural Network Society (INNS) and the Technical Chamber of Greece. He is the recipient of the W. T. Kittinger Outstanding Teacher Award. He is also a corecipient of a Theoretical Development Award for a paper presented at the Artificial Neural Networks in Engineering'94 Conference. He is an Associate Editor of the IEEE TRANSACTIONS ON FUZZY SYSTEMS. He is also the General Chair of the 1997 International Conference on Neural Networks (ICNN '97), to be held in Houston, TX, on June 9–12, 1997.