SOME STUDIES ON LVQ MODELS

A dissertation submitted in partial fulfilment of the requirements for the
M. Tech. (Computer Science) degree of the Indian Statistical Institute

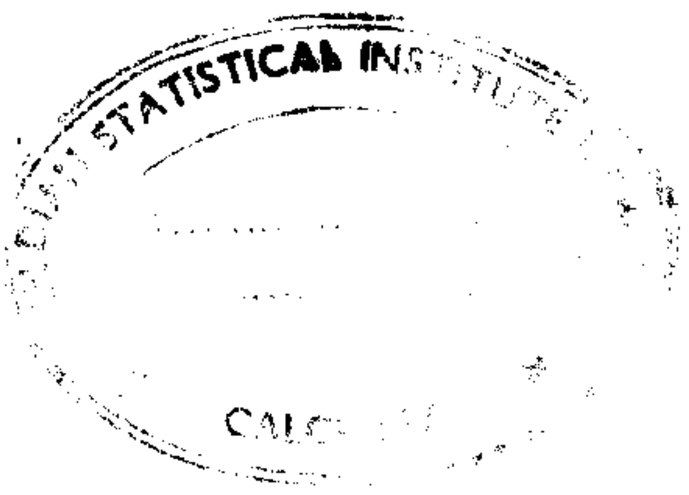
By

Sanjoy Bose

Under The Supervision of

Dr. C.A. Murthy,
Machine Intelligence Unit.
INDIAN STATISTICAL INSTITUTE
203, Barrackpore Trunk Road,
Calcutta-700035

July 10, 1996



Indian Statistical Institute
203, B.T. Road,
Calcutta- 700 035.

Certificate of Approval

This is to certify that the thesis titled **SOME STUDIES ON LVQ MODELS** submitted by **Sanjoy Bose**, towards partial fulfillment of the requirements for the degree of M. Tech. in Computer Science at the Indian Statistical Institute, Calcutta, embodies the work done under my supervision.

Dr. C. A. Murthy
Machine Intelligence Unit,
Indian Statistical Institute,
Calcutta-700 035.

Acknowledgements

The very first person whom I'd like to thank and whom I feel myself honoured to have worked with is my supervisior Dr. C.A. Murthy, who is both an excellent teacher and a nice human being, and who has taught me a lot during my stay of two years.

I am also grateful to Prof. S.K. Pal for reviewing my work in the middle of the semester and also giving valuable suggestions.

I thank all C.S.S.C. staffs for helping me to avail the computing facilities and also giving their valuable advices, whenever I faced any problem.

I also grateful to Mr. Paramartha Dutta for helping me with LaTeX, when I was writing this report.

I am also thankful to Mr. B. Uma Shankar for helping me to take the print outs of different images.

Lastly, I should thank all my classmates, especially Mr. Bishwadeep Biswas, Mr. Eluri Vijay Kumar for helpful discussion, and Mr. Rajarshi Chowdhury for helping me in drawing different figures. I take the oppurtunity to thank all of them for making my two year's stay in ISI a memorable one.

Calcutta.
July 10,1996

(Sanjoy Bose)

Abstract

A new LVQ model has been proposed and listed here. It's performance in relation to other existing models has been studied experimentally with the help of artiicial data set as well as IRIS data. Finally the proposed algorithm is applied on a satellite image data. The perfomance of the proposed model has been found satisfactory.

Contents

Chapter 1

Introduction

Clustering is an important technique used in discovering the inherent structure present in the set of objects. Clustering algorithms attempt to organize unlabelled pattern vectors into clusters or **"natural groups"** such that the points within a cluster are more similar to each other than to points belonging to different clusters. **Vector Quantization** is essentially a clustering process which attempts to subdivide a random set of vectors into subsets, or clusters, which are pairwise disjoint, all nonempty and reproduce the original set of vectors via union[3].

Let the set of patterns $\mathbf{X}$ be $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, x_i, \ldots, \mathbf{x}_n\}$ where $\mathbf{x}_i$ is the ith pattern vector, $\mathbf{X} \subseteq \mathcal{R}^p, \mathbf{X}$ is finite. Let the number of clusters be k. If the clusters are represented by $C_1, C_2, \ldots, C_k$ then

i) $C_i \neq \phi$ for $i = 1, \ldots, k$
ii) $C_i \bigcap C_j = \phi \quad i \neq j$
iii) $\bigcup_{i=1}^{k} C_i = \mathbf{X}$

where ϕ represents null set and $k \geq 2$[2].
The goal of clustering is therefore to minimize the sum of squares of within cluster distances, i.e.:

$$\sum_{i=1}^{k} \sum_{\mathbf{x} \in C_i} \|\mathbf{x} - \mathbf{v}_i\|^2$$

where $\mathbf{x}$ is the input vector. The set of vectors $\mathbf{V} = (\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k)$ is called the codebook. $\mathbf{v}_i$ is the representative vector for class C_i. Thus class C_i is quantized by the vector $\mathbf{v}_i$. The process of designing the codebook is

called **Vector Quantization**. Many techniques of vector quantization use clustering approach[9].

Neural Networks have been employed in many clustering problems. Among the existing models, the **Self Organizing Feature Map** of Kohonen [11],[4] find the topological structure hidden in the input data. Kohonen's **Learning Vector Quantizer** (LVQ) [10] network can perform clustering when the number of clusters present in the data set is known apriori.

In the following sections, we deal with unsupervised pattern classification using neural network approach. In the unsupervised algorithms, no information concerning the correct class is provided to the nets. Each new pattern is presented only once and the weights are modified after each presentation. We assume everywhere that the number of clusters, k is known apriori. The learning process used is known as **Competitive Learning Scheme**. The basic idea underlying what is called competitive learning is roughly as follows:

Assume a sequence of samples of vector $\mathbf{x} = \mathbf{x}(t) \in \mathcal{R}^p$, where t is the time coordinate, $x(n+1) = x(n)$ and a set of variable reference vectors $\{\mathbf{v}_i(t) : \mathbf{v}_i \in \mathcal{R}^p, i = 1, 2, \ldots, k\}$. Assume that $\mathbf{v}_i(0)$'s have been initialized in some proper way(random selection will suffice). If $\mathbf{x}(t)$ is compared with each $\mathbf{v}_i(t)$ at each successive instant t (taken here to be $t = 1, 2, \ldots$) then the best matching $\mathbf{v}_i(t)$ is obtained by some distance measure $d(\mathbf{x}(t), \mathbf{v}_i(t))$. If $i = c$ be the best matching reference vector then

$$d(\mathbf{x}(t), \mathbf{v}_c(t)) = \min_i d(\mathbf{x}(t), \mathbf{v}_i(t))$$

Then $\mathbf{v}_c(t)$ is to be updated, so that it moves closer to $\mathbf{x}(t)$. In the neural network model the neigbouring cells in the output layer compete in their activities such that in the process, vectors tend to become specifically "tuned" to different domains of the input variable $\mathbf{x}$ [5].

Chapter 2

Existing Algorithms

2.1 Learning Vector Quantization

In the **VQ** the objective has been to find vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k$ $(k \geq 2)$ such that,

$$\sum_{\mathbf{x} \in \mathbf{X}} f(\mathbf{x}, \mathbf{v}_i)$$

is minimized. Here $\mathbf{v}_i$ is closest to $\mathbf{x}$ and $f(\mathbf{x}, \mathbf{v}_i)$ is a function of the distance between $\mathbf{x}$ and $\mathbf{v}_i$. In the neural network based **LVQ** models, where at any single instant only one input vector $\mathbf{x}$ from $\mathbf{X}$ is under consideration, the researchers have tried to achieve the stated aim by modifying the vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k$ at each instant taking help of $f(\mathbf{x}, \mathbf{v}_i)$ [where $\mathbf{v}_i$ is the winning prototype for $\mathbf{x}$] and $f(\mathbf{x}, \mathbf{v}_r)$ [where $\mathbf{v}_r$ is the non-winning prototype for $\mathbf{x}$].

LVQ has been associated in literature with a neural network architecture that has been shown in Fig.1 . If $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} \subseteq \mathcal{R}^p$ denote the unlabelled data, k denotes the number of clusters, then the input layer of the network contains p nodes and the output layer contains k nodes. The input layer is connected directly to the competition layer or the output layer. The ith node in the output layer is associated with a weight vector $\mathbf{v}_i$. The p components $\{\mathbf{v}_{ji}\}$ of $\mathbf{v}_i$ are often regarded as weights or connection strengths of the edges that connect the p inputs to the node i.

The prototypes $\mathbf{V} = (\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k), \mathbf{v}_i \in \mathcal{R}^p$ for $1 \leq i \leq k$ are the unknown vector quantizers we seek. In this context learning refers to finding the values for the $\{\mathbf{v}_{ji}\}$. When an input vector $\mathbf{x}$ is submitted to this

network, distances are computed between $\mathbf{x}$ and each $\mathbf{v}_i$. The output node i in the output layer is the distance between $\mathbf{x}$ and $\mathbf{v}_i$. The output nodes compete, a winner node (minimum distance) say c is found, and the corresponding $\mathbf{v}_c$ is then updated using an update rule.

The **LVQ** algorithm is given below:

step 1. Given unlabelled data set $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} \subseteq \mathcal{R}^p$ and number of clusters k.

Fix N = maximum number of updating steps, and $\epsilon > 0$ where ϵ is the termination condition.

step 2. Initialize $\mathbf{V}_0 = (\mathbf{v}_{1,0}, \ldots, \mathbf{v}_{k,0})$ where each $\mathbf{v}_{i,0} \in \mathcal{R}^p$, and learning rate $\alpha_0 \in (0, 1)$.

For $t = 1, 2, \ldots, N$:

For $j = 1, 2, \ldots, n$

a. Find

$$\min_i \|\mathbf{x}_j - \mathbf{v}_{i,t-1}\| \tag{1}$$

Let $\mathbf{v}_{c,t-1}$ be such that $\|\mathbf{x}_j - \mathbf{v}_{c,t-1}\| = min_i \|\mathbf{x}_j - \mathbf{v}_{i,t-1}\|$

b. Update the winner $\mathbf{v}_{c,t-1}$:

$$\mathbf{v}_{c,t} = \mathbf{v}_{c,t-1} + \alpha_{t-1}(\mathbf{x}_j - \mathbf{v}_{c,t-1}) \tag{2}$$

Next j

step 3. Compute

$$E_t = \sum_{l=1}^{p} \sum_{r=1}^{k} |v_{lr,t} - v_{lr,t-1}|$$

step 4. if $E_t \leq \epsilon$ stop;Else adjust learning rate $\alpha_t \leftarrow \alpha_0(1 - t/N)$.

Next t.

step 5. For each $\mathbf{x}$ in $\mathbf{X}$ if

$$\|\mathbf{x} - \mathbf{v}_{c,t-1}\| = \min_j \|\mathbf{x} - \mathbf{v}_{j,t}\|$$

and mark $\mathbf{x}$ with label c

The update scheme used for modifying the winner has a simple geometric interpretation which is shown in Fig.2 .

The winning prototype $\mathbf{v}_{c,t-1}$ is moved along the vector $(\mathbf{x}_j - \mathbf{v}_{c,t-1})$ towards $\mathbf{x}_j$. The amount by which $\mathbf{v}_{c,t-1}$ is shifted to arrive at $\mathbf{v}_{c,t}$ depends on the value of the learning rate parameter α_{t-1} where $\alpha_t \in [0, 1)$.

2.1.1 Initialization

$\mathbf{V_0} = (\mathbf{v}_{1,0}, \mathbf{v}_{2,0}, \ldots, \mathbf{v}_{k,0}) \in \mathcal{R}^p$ have to be initialized. There are several initialization schemes. An initialization scheme used in the existing algorithms, and also in the proposed method is described below:
For data set $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2 \ldots, \mathbf{x}_n\} \subseteq \mathcal{R}^p$. Let data point q and the initial prototype i be $\mathbf{x}_q = (x_{1q}, x_{2q}, \ldots, x_{pq})'$ and $\mathbf{v}_i = (v_{1i}, v_{2i} \ldots v_{pi})'$ respictively. Compute the feature ranges :
Minimum of feature

$$j : m_j = \min_q \{x_{jq}\} : \quad j = 1, 2, \ldots, p \tag{3}$$

Maximum of feature

$$j : M_j = \max_q \{x_{jq}\} : \quad j = 1, 2, \ldots, p \tag{4}$$

with this compute the jth component of the ith initial prototype v_{ji} as:

$$v_{ji} = m_j + (i-1)\left(\frac{M_j - mj}{k-1}\right) \quad i = 1, 2, \ldots, k; \; j = 1, 2, \ldots, p \tag{5}$$

Formula (4) disperses initial prototype values uniformly along each feature range.

2.2 GLVQ

LVQ attempts to minimize an objective function that places all its emphasis on the winning prortotype for each data point. This is reflected in eqn.(2) which alters only the winner. This, however ignores global information about the geometric structure of the data that is represented in the remaining $(k-1)$ distances from $\mathbf{x}$ to the non-winner prototypes. In this section **Generalized Learning Vector Quantization algorithm** is discussed. The algorithm is associated with the same neural network architecture, where the feature vectors $\mathbf{x}$ provide the inputs to the map and the weight vectors play the role of the prototypes $\mathbf{v}_i$. The learning rule associated with **GLVQ** is obtained by minimizing a cost function which measures a locally weighted error of the input with respect to the winning prototype. Mathematically this is explained below.

Let $x \in \mathcal{R}^p$ be an input vector. Let $\mathcal{J}$ be the cost function which measures the locally weighted mismatch of x with respect to the winner.

$$\mathcal{J}(\mathbf{x}; \mathbf{v}_1, \ldots, \mathbf{v}_k) = \sum_{r=1}^{k} g_r \|\mathbf{x} - \mathbf{v}_r\|^2 \tag{6}$$

$$g_i = \left\{ \begin{array}{ll} 1 & \text{if } i = \min_k \|\mathbf{x} - \mathbf{v}_k\| \\ \frac{1}{\sum_{j=1}^{k} \|\mathbf{x}-\mathbf{v}_j\|^2} & \text{otherwise} \end{array} \right\} 1 \leq i \leq k \tag{7}$$

Where $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ is a set of inputs. The objective of the GLVQ is to find a set of k $\mathbf{v}_r$'s, say $\mathbf{V} = \{\mathbf{v}_r\}$, such that the locally weighted error functional $\mathcal{J}$ is minimized.

The update rules for solving eqn.(3) based on minimization of $\mathcal{J}$ [6] are: for winner prototype i,

$$\mathbf{v}_{i,t} = \mathbf{v}_{i,t-1} + \alpha_{t-1} \left(\frac{D^2 - D + \|\mathbf{x} - \mathbf{v}_{i,t-1}\|^2}{D^2}\right) (\mathbf{x} - \mathbf{v}_{i,t-1}) \tag{8a}$$

and for non-winner node j,

$$\mathbf{v}_{j,t} = \mathbf{v}_{j,t-1} + \alpha_{t-1} \left(\frac{\|\mathbf{x} - \mathbf{v}_{i,t-1}\|^2}{D^2}\right) (\mathbf{x} - \mathbf{v}_{j,t-1}) \quad for\ j = 1, 2, \ldots, k \quad j \neq i \tag{8b}$$

where **x** is the current input vector and

$$D = \sum_{r=1}^{k} \|\mathbf{x} - \mathbf{v}_{r,t-1}\|^2$$

Tables 1,2 show the result of $T = 500$ iterations of GLVQ with the initial learning rate $\alpha_0 = 0.6$ on **IRIS** and **IRIS/10** respectively, where in IRIS/10 the feature vectors of IRIS are scaled by a factor of 10.

The resullts show that the **GLVQ** algorithm doesn't work properly for the normalized data. For some scaling of data, it may happen that the change in the winner prototype is less than changes that are made to the other $(k-1)$ prototypes. So the non-winner prototypes will be pulled towards the data more strongly than the winner prototypes. This results in all prototypes migrating to the same point in $\mathcal{R}^p$, as they did for **IRIS/10** [6],[7].

2.3 Fuzzy algorithms for learning vector quantization

The algorithms are based on the minimization of a fuzzy objective function, formed as the weighted sum of the squared euclidean distances between each input vector and the prototypes. Assuming that **x** is the input vector, $\mathbf{v}_i$ is the winning prototype, and k is the number of clusters, the update equation for the prototypes can be derived by minimizing,

$$\mathcal{J} = \sum_{r=1}^{k} u_{ir} \|\mathbf{x} - \mathbf{v}_r\|^2 \tag{9}$$

where $u_{ir} = u_{ir}(x), r = 1, 2, \ldots, k$ is a set of generalized membership functions, which regulate the competition between the prototypes, $\mathbf{v}_r$, $r = 1, 2, \ldots, k$ for the input **x**. The term generalized membership functions is used to indicate that their form can be selected apriori according to some intuitively reasonable criteria [7].

The development of genuinely competitive learning vector quantization algorithm requires the selection of the generalized membership functions

assigned to the prototypes. A fair competition among the prototypes is guaranteed if the generalized memberhsip function assigned to each prototype:

- is invariant to the magnitude of input vectors.
- is equal to unity if the prototype is the winner.
- takes the value between 1 and 0 if the prototype is not a winner.
- approaches zero if the prototype is not a winner and its distance from the input vector approaches infinity [7].

Some fuzzy algorithms for LVQ are described below.

2.3.1 FALVQ

Assuming $\mathbf{x}$ is the input vector and $\mathbf{v}_i$ is the winning prototype, i.e.,

$$\|\mathbf{x} - \mathbf{v}_i\|^2 \; < \; \|\mathbf{x} - \mathbf{v}_r\|^2 \;\; \forall \; \mathbf{v}_r \neq \mathbf{v}_i$$

the above mentioned conditions are satisfied by the objective function defined by eqn.(9) with

$$u_{ir} = \begin{cases} 1 & \text{if } r = i \\ \frac{1}{1+\frac{\|\mathbf{x}-\mathbf{v}_r\|^2}{\|\mathbf{x}-\mathbf{v}_i\|^2}} & \text{if } r \neq i \end{cases} \tag{10}$$

According to this definition, u_{ir} decreases from a value close to $\frac{1}{2}$ to 0 as $\|\mathbf{x} - \mathbf{v}_r\|^2$ increases from a value slightly higher than $\|\mathbf{x} - \mathbf{v}_i\|^2$ to infinity.

The objective function $\mathcal{J}$ is, therefore:

$$\mathcal{J} \; = \; \|\mathbf{x} - \mathbf{v}_i\|^2 \; + \; \sum_{r \neq i}^{k} \left(\frac{1}{1 + \frac{\|\mathbf{x}-\mathbf{v}_r\|^2}{\|\mathbf{x}-\mathbf{v}_i\|^2}}\right) \|\mathbf{x} - \mathbf{v}_r\|^2 \tag{11}$$

The FALVQ updation rules are derived by minimizing the above objective function using the gradient descent method [7]. If $\mathbf{x}$ is the input vector, the

winning prototype $\mathbf{v}_i$ can be updated by :

$$\Delta \mathbf{v}_i \;=\; \alpha\,(\mathbf{x} - \mathbf{v}_i)\,(1 \;+\; \sum_{r \neq i}^{k} (1 - u_{ir})^2\,) \qquad (12a)$$

While the non-winning prototypes $\mathbf{v}_j \neq \mathbf{v}_i$ can be updated by :

$$\Delta \mathbf{v}_j \;=\; \alpha\,(\mathbf{x} - \mathbf{v}_j)\,u_{ij}^2 \qquad (12b)$$

The adaption of the prototype during the learning process depends on the learning rate $\alpha \in [0, 1)$, which is a monotonically decreasing function of the number of iterations t defined as $\alpha = \alpha(t) = \alpha_0(1 - t/N)$, where α_0 is the initial value of the learning rate and N the total number of iterations, predetermined for the learning process.

2.3.2 Other Fuzzy Algorithms

As seen in the previous section, $u_{ir} = 1$ if $\mathbf{v}_r = \mathbf{v}_i$, where $\mathbf{v}_i$ is the winning prototype, that is, $\|\mathbf{x} - \mathbf{v}_i\|^2 = min_{\mathbf{v}_j \in \mathbf{V}} \|\mathbf{x} - \mathbf{v}_j\|^2$. If $\mathbf{v}_r \neq \mathbf{v}_i$, then $\frac{\|\mathbf{x}-\mathbf{v}_r\|^2}{\|\mathbf{x}-\mathbf{v}_i\|^2} > 1 \;\; \forall \mathbf{v}_r \neq \mathbf{v}_i$ and therefore $u_{ir} < \frac{1}{2}$. Since $u_{ir} \in (0, \frac{1}{2}) \; \forall r \neq i$ the function u_{ir} described in eqn.(10) favours rather strongly the winning prototype and hece there is a bias inherent in the definition of u_{ir} towards the winner. So, the weight of $\|\mathbf{x} - \mathbf{v}_r\|$ in $\mathcal{J}$ lies between 0 and $\frac{1}{2}$. In other words, the contribution of $\|\mathbf{x} - \mathbf{v}_r\|^2$ towards $\mathcal{J}$ is restricted, reducing the competitive effect of the non-winner.

The non-winning prototypes can be made more competitive by introducing a new set of generalized membership functions such that, if $\mathbf{v}_r \neq \mathbf{v}_i$, u_{ir} takes the values in the interval $(0, \beta_r)$, where $\beta_r \geq \frac{1}{2}$.
This is done by introducing a new set of generalized membership functions of the form

$$u_{ir} = \begin{cases} 1 & \text{if } r = i \\ \dfrac{1}{1 + \frac{\|\mathbf{x}-\mathbf{v}_r\|^2}{D(\mathbf{x}, \mathbf{v}_j \in \mathbf{V})}} & \text{if } r \neq i \end{cases} \qquad (13)$$

where $D(\mathbf{x}, \mathbf{v}_j \in \mathbf{V})$ is a differentiable function of $\|\mathbf{x} - \mathbf{v}_j\|^2$, $\mathbf{v}_j \in \mathbf{V}$ such that $D(\mathbf{x}, \mathbf{v}_j \in \mathbf{V}) \geq min_{\mathbf{v}_j \in \mathbf{V}} \|\mathbf{x} - \mathbf{v}_j\|^2$ where $min_{\mathbf{v}_j \in \mathbf{V}} \|\mathbf{x} - \mathbf{v}_j\|^2 = \|\mathbf{x} -$

$\mathbf{v}_i\|^2$, if $\mathbf{v}_i$ is the winning prototype. u_{ir} in eqn.(13) can also be written as,

$$u_{ir} = \begin{cases} 1 & \text{if } r = i \\ \frac{1}{1+\frac{\|\mathbf{x}-\mathbf{v}_r\|^2}{\|\mathbf{x}-\mathbf{v}_i\|^2}\cdot\frac{\|\mathbf{x}-\mathbf{v}_i\|^2}{D(\mathbf{x},\mathbf{v}_j\in\mathbf{V})}} & \text{if } r \neq i \end{cases}$$

Thus the value of β_r depends on the ratio of $\frac{\|\mathbf{x}-\mathbf{v}_i\|^2}{D(\mathbf{x},\mathbf{v}_j\in\mathbf{V})}$.
The three existing algorithms [7] using this concept are discussed below:

Harmonic FALVQ

Here $\frac{1}{D(\mathbf{x},\mathbf{v}_j\in\mathbf{V})} = \frac{1}{D_H(\mathbf{x},\mathbf{v}_{j\in\mathbf{V}})} = \frac{1}{k}\sum_{j=1}^{k}\frac{1}{\|\mathbf{x}-\mathbf{v}_j\|^2}$
The updation rule obtained using the same objective function as (9) are :

$$\Delta\mathbf{v}_i = \alpha\,(\mathbf{x}-\mathbf{v}_i)\,(1+\frac{1}{k}\sum_{r\neq i}^{k} u_{ir}^2\,(\frac{\|\mathbf{x}-\mathbf{v}_r\|^2}{\|\mathbf{x}-\mathbf{v}_i\|^2})^2) \qquad (14a)$$

for the winning prototype and

$$\Delta\mathbf{v}_j = \alpha\,(\mathbf{x}-\mathbf{v}_j)\,(u_{ij}^2+\frac{1}{k}\sum_{r\neq i}^{k} u_{ir}^2\,(\frac{\|\mathbf{x}-\mathbf{v}_r\|^2}{\|\mathbf{x}-\mathbf{v}_j\|^2})^2) \qquad (14b)$$

for the non-winning prototypes $\mathbf{v}_j \neq \mathbf{v}_i$.

Geometric FALVQ

Here $D(\mathbf{x},\mathbf{v}_j\in\mathbf{V}) = D_G(\mathbf{x},\mathbf{v}_j\in\mathbf{V}) = (\prod_{j=1}^{k}\|\mathbf{x}-\mathbf{v}_j\|^2)^{\frac{1}{k}}$

The update equations are :

$$\Delta\mathbf{v}_i = \alpha(\mathbf{x}-\mathbf{v}_i)\,(1+\frac{1}{k}\sum_{r\neq i}^{k} u_{ir}(1-u_{ir})\frac{\|\mathbf{x}-\mathbf{v}_r\|^2}{\|\mathbf{x}-\mathbf{v}_i\|^2}) \qquad (15a)$$

for the winner prototype and

$$\Delta\mathbf{v}_j = \alpha(\mathbf{x}-\mathbf{v}_j)\,(\mathbf{u}_{ij}^2+\frac{1}{k}\sum_{r\neq i}^{k} u_{ir}(1-u_{ir})\frac{\|\mathbf{x}-\mathbf{v}_r\|^2}{\|\mathbf{x}-\mathbf{v}_i\|^2}) \qquad (15b)$$

for the non-winner prototype $\mathbf{v}_j \neq \mathbf{v}_i$

Arithmetic FALVQ

Here $D(\mathbf{x}, \mathbf{v}_j \in \mathbf{V}) = D_A(\mathbf{x}, \mathbf{v}_j \in \mathbf{V}) = \frac{1}{k} \sum_{j=1}^{k} \|\mathbf{x} - \mathbf{v}_j\|^2$

The update equations are:

$$\Delta \mathbf{v}_i = \alpha(\mathbf{x} - \mathbf{v}_i) \left(1 + \frac{1}{k} \sum_{r \neq i}^{k} (1 - u_{ir})^2\right) \tag{16a}$$

for the winner prototype and

$$\Delta \mathbf{v}_j = \alpha(\mathbf{x} - \mathbf{v}_j) \left(u_{ij}^2 + \frac{1}{k} \sum_{r \neq i}^{k} (1 - u_{ir})^2\right) \tag{16b}$$

for the non-winner prototype $\mathbf{v}_j \neq \mathbf{v}_i$

The analysis of the algorithms using harmonic, geometric and arithmetic mean are done in [7]. In both the cases of Harmonic FALVQ and Geometric FALVQ, updation equations are such that, if $\|\mathbf{x} - \mathbf{v}_r\|^2 \gg \|\mathbf{x} - \mathbf{v}_i\|^2$, the updation of the winning prototype towards input $\mathbf{x}$ is increased, while on the other hand if $\|\mathbf{x} - \mathbf{v}_r\|^2 \approx \|\mathbf{x} - \mathbf{v}_i\|^2$ the effect of the input x on the winning prototype is inhibited i.e. the updation of the winner towards x is decreased.Again $u_{ir}(1-u_{ir})$ in eqn(15a) is an increasing function if $0 \leq u_{ir} \leq \frac{1}{2}$ and decreasing function if $u_{ir} > \frac{1}{2}$. Since in the geometric FALVQ $u_{ir} > \frac{1}{2}$ when $\|\mathbf{x} - \mathbf{v}_r\|^2$ is sufficiently close to $\|\mathbf{x} - \mathbf{v}_i\|^2$, $u_{ir}(1-u_{ir})$ decreases as the non-winning prototype $\mathbf{v}_r$ approaches $\mathbf{v}_i$,while u_{ir}^2 in Harmonic FALVQ is monotonically increasing function of u_{ir}. Thus the non-winning prototypes close to the winning prototype v_i results in stronger inhibition of its adaption when the Geometric FALVQ is used and so Geometric FALVQ results in stronger competition. In Arithmetic FALVQ, though each input $\mathbf{x}$ has a stronger effect on winner than the non winner(updation of the winning prototype is greater than the updation of non-winner prototype), however difference between u_{ij}^2 and 1 is not significant, so Arithmetic FALVQ1 can't discriminate between prototypes which are similar.

Chapter 3

Proposed Method

The generalized membership functions used in the algorithms HFALVQ, GFALVQ, AFALVQ are of the form shown in eqn.(13), where the value of u_{ir} lies between $(0, \beta_r)$ for $r \neq i$ and $\beta_r \geq \frac{1}{2}$. The value of β_r depends on the ratio of $\frac{\|\mathbf{x}-\mathbf{v}_i\|^2}{D(\mathbf{x},\mathbf{v}_j \in \mathbf{V})}$. However, the ratio varies from problem to problem and cannot be made equal to zero. So, the value of β_r cannot reach 1 and hence there always exists a bias, inherent in the definition of u_{ir}. Infact u_{ir} is not a continuous function in all these cases. Again, it was experimentally found that these existing algorithms, though work sufficiently well for the IRIS data set , where there are 3 equal classes of 50 data each, they fail for the data set where unequal sized classes present. For these cases the cluster centers produced by the algorithms are very much different from the physical cluster centers.
So a new algorithm has to be developed that works equally well for the data set having equal sized classes and at the same time, for unequal classes produce a better cluster centers which are close to the physical cluster centers.

To remove the bias in u_{ir} completely a new set of generalized membership functions is chosen that satisfies the four conditions for a genuinely competitive learning vector quantization. The new generalized membership function is :

$$u_{ir} = exp(1 - \frac{\|\mathbf{x} - \mathbf{v}_r\|^2}{\|\mathbf{x} - \mathbf{v}_i\|^2}) \tag{17}$$

This is a continuous function in $(0, 1]$.

The objective function $\mathcal{J}$ is taken to be same as before:

$$\mathcal{J} = \sum_{r=1}^{k} u_{ir} \|x - v_r\|^2$$

where $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k \in \mathcal{R}^p$ are the set of prototypes, k is the number of clusters and $k \geq 2$

The update equations for the winning prototypes and the non-winning prototypes are derived by minimizing $\mathcal{J}$ and using the u_{ir} as given in the eqn(17).

The derivation of the updation equation for the winning prototype:

$$\mathcal{J} = \sum_{r=1}^{k} u_{ir} \|\mathbf{x} - \mathbf{v}_r\|^2 = \|\mathbf{x} - \mathbf{v}_i\|^2 + \sum_{r \neq 1}^{k} u_{ir} \|\mathbf{x} - \mathbf{v}_r\|^2$$

Differentiating, $\mathcal{J}$ with respect to the winning prototype $\mathbf{v}_i$ gives

$$\begin{aligned}
\frac{\partial \mathcal{J}}{\partial \mathbf{v}_i} &= -2(\mathbf{x} - \mathbf{v}_i) + \frac{\partial}{\partial \mathbf{v}_i}\left(\sum_{r \neq i}^{k} u_{ir} \|\mathbf{x} - \mathbf{v}_r\|^2\right) \\
&= -2(\mathbf{x} - \mathbf{v}_i) + \sum_{r \neq i}^{k} \left(\frac{\partial}{\partial \mathbf{v}_i} u_{ir} \|\mathbf{x} - \mathbf{v}_r\|^2\right) \\
&= -2(\mathbf{x} - \mathbf{v}_i) + \sum_{r \neq i}^{k} u_{ir} \left(\frac{\partial}{\partial \mathbf{v}_i}\left(-\frac{\|\mathbf{x} - \mathbf{v}_r\|^2}{\|\mathbf{x} - \mathbf{v}_i\|^2}\right)\right) \|\mathbf{x} - \mathbf{v}_r\|^2 \\
&= -2(\mathbf{x} - \mathbf{v}_i) + \sum_{r \neq i}^{k} u_{ir} \left(\frac{\partial}{\partial \mathbf{v}_i}\left(-\frac{1}{\|\mathbf{x} - \mathbf{v}_i\|^2}\right)\right) \|\mathbf{x} - \mathbf{v}_r\|^4 \\
&= -2(\mathbf{x} - \mathbf{v}_i) - \sum_{r \neq i}^{k} u_{ir} \frac{2(\mathbf{x} - \mathbf{v}_i)}{\|\mathbf{x} - \mathbf{v}_i\|^4} \|\mathbf{x} - \mathbf{v}_r\|^4 \\
&= -2(\mathbf{x} - \mathbf{v}_i) - 2\sum_{r \neq i}^{k} u_{ir} \left(\frac{\|\mathbf{x} - \mathbf{v}_r\|^2}{\|\mathbf{x} - \mathbf{v}_i\|^2}\right)^2 (\mathbf{x} - \mathbf{v}_i) \\
&= -2\left(1 + \sum_{r \neq i}^{k} u_{ir} \left(\frac{\|\mathbf{x} - \mathbf{v}_r\|^2}{\|\mathbf{x} - \mathbf{v}_i\|^2}\right)^2\right)(\mathbf{x} - \mathbf{v}_i)
\end{aligned}$$

Derivation of the updation equation ofthe non-winning prototype:
Differentiating $\mathcal{J}$ with respect to non-winning prototypes $\mathbf{v}_j, j \neq i$,gives

$$\begin{aligned}
\frac{\partial \mathcal{J}}{\partial \mathbf{v}_j} &= \frac{\partial}{\partial \mathbf{v}_j}(\|\mathbf{x}-\mathbf{v}_i\|^2 + \textstyle\sum_{r\neq i}^{k} u_{ir}\|\mathbf{x}-\mathbf{v}_r\|^2) \\
&= \frac{\partial}{\partial \mathbf{v}_j}(u_{ij}\|\mathbf{x}-\mathbf{v}_j\|^2 + \textstyle\sum_{r\neq i,j}^{k} u_{ir}\|\mathbf{x}-\mathbf{v}_r\|^2) \\
&= (\frac{\partial}{\partial \mathbf{v}_j}u_{ij})\|\mathbf{x}-\mathbf{v}_j\|^2 + u_{ij}\frac{\partial}{\partial \mathbf{v}_j}(\|\mathbf{x}-\mathbf{v}_j\|^2) \\
&= u_{ij}(\frac{\partial}{\partial \mathbf{v}_j}\frac{\|\mathbf{x}-\mathbf{v}_j\|^2}{\|\mathbf{x}-\mathbf{v}_i\|^2})\|\mathbf{x}-\mathbf{v}_j\|^2 - 2u_{ij}(\mathbf{x}-\mathbf{v}_j) \\
&= 2u_{ij}\frac{(\mathbf{x}-\mathbf{v}_j)}{\|\mathbf{x}-\mathbf{v}_i\|^2}\|\mathbf{x}-\mathbf{v}_j\|^2 - 2u_{ij}(\mathbf{x}-\mathbf{v}_j) \\
&= u_{ij}(\mathbf{x}-\mathbf{v}_j)(\frac{\|\mathbf{x}-\mathbf{v}_j\|^2}{\|\mathbf{x}-\mathbf{v}_i\|^2} - 1)
\end{aligned}$$

So,the updation equation of the winning prototype is:

$$\mathbf{v}_i(t+1) = \mathbf{v}_i(t) + \Delta\mathbf{v}_i$$

where

$$\Delta\mathbf{v}_i = -\alpha'\frac{\partial \mathcal{J}}{\partial \mathbf{v}_i}$$

or,

$$\Delta\mathbf{v}_i = \alpha'(1 + \sum_{r\neq i}^{k} u_{ir}(\frac{\|\mathbf{x}-\mathbf{v}_r\|^2}{\|\mathbf{x}-\mathbf{v}_i\|^2})^2)(\mathbf{x}-\mathbf{v}_i) \qquad (18a)$$

The updation equation for the non-winning prototype is:

$$\mathbf{v}_j(t+1) = \mathbf{v}_j(t) + \Delta\mathbf{v}_j$$

where

$$\Delta\mathbf{v}_j = \alpha'\frac{\partial \mathcal{J}}{\partial \mathbf{v}_i}$$

Here, the sign is taken positive to make the updated $\mathbf{v}_j$ move closer to $\mathbf{x}$.
So,

$$\Delta\mathbf{v}_j = \alpha' u_{ij}(\frac{\|\mathbf{x}-\mathbf{v}_j\|^2}{\|\mathbf{x}-\mathbf{v}_i\|^2} - 1)(\mathbf{x}-\mathbf{v}_j) \qquad (18b)$$

where $\alpha' = \frac{\alpha}{k-1}$ and α is the learning rate.

The comparison between the updation equation of the winning and the non-winning prototype is based on the observation that,

$$
\begin{aligned}
u_{ij}\left(\frac{\|\mathbf{x}-\mathbf{v}_j\|^2}{\|\mathbf{x}-\mathbf{v}_i\|^2}\right) &< u_{ij}\left(\frac{\|\mathbf{x}-\mathbf{v}_j\|^2}{\|\mathbf{x}-\mathbf{v}_i\|^2}\right) + \textstyle\sum_{r\neq i,j}^{k} u_{ir}\left(\frac{\|\mathbf{x}-\mathbf{v}_r\|^2}{\|\mathbf{x}-\mathbf{v}_i\|^2}\right) \\
\Rightarrow \quad u_{ij}\left(\frac{\|\mathbf{x}-\mathbf{v}_j\|^2}{\|\mathbf{x}-\mathbf{v}_i\|^2}\right) &< \textstyle\sum_{r\neq i,j}^{k} u_{ir}\left(\frac{\|\mathbf{x}-\mathbf{v}_r\|^2}{\|\mathbf{x}-\mathbf{v}_i\|^2}\right)^2 \\
\Rightarrow \quad u_{ij}\left(\frac{\|\mathbf{x}-\mathbf{v}_j\|^2}{\|\mathbf{x}-\mathbf{v}_i\|^2}\right) - u_{ij} &< \textstyle\sum_{r\neq i,j}^{k} u_{ir}\left(\frac{\|\mathbf{x}-\mathbf{v}_r\|^2}{\|\mathbf{x}-\mathbf{v}_i\|^2}\right)^2 + 1 \\
\Rightarrow \quad u_{ij}\left(\left(\frac{\|\mathbf{x}-\mathbf{v}_j\|^2}{\|\mathbf{x}-\mathbf{v}_i\|^2}\right) - 1\right) &< 1 + \textstyle\sum_{r\neq i} u_{ir}\left(\frac{\|\mathbf{x}-\mathbf{v}_r\|^2}{\|\mathbf{x}-\mathbf{v}_i\|^2}\right)^2 \qquad (19)
\end{aligned}
$$

Clearly, eqn.(19) indicates that the input vector x has a more significant effect on the winning prototype i.e. the updation for the winner is greater than the updation for the non-winner.

The adaptation of the winning prototype $\mathbf{v}_i$ can be investigated by studying the term $\sum_{r\neq i}^{k} u_{ir}\left(\frac{\|\mathbf{x}-\mathbf{v}_r\|^2}{\|\mathbf{x}-\mathbf{v}_i\|^2}\right)^2$ in eqn.(18a) which represents the effect of the non-winning prototypes. Assume that $\mathbf{v}_r$ is the non-winning prototype such that $\|\mathbf{x}-\mathbf{v}_r\|^2 \gg \|\mathbf{x}-\mathbf{v}_i\|^2$. According to the defn. of u_{ir}, in this case, when $r \neq i$, $u_{ir} \rightarrow 0$. Hence $u_{ir}\left(\frac{\|\mathbf{x}-\mathbf{v}_r\|^2}{\|\mathbf{x}-\mathbf{v}_i\|^2}\right)^2 \rightarrow 0$. So, $\Delta\mathbf{v}_i$ given by eqn.(18a) is small. However, if $\|\mathbf{x}-\mathbf{v}_r\|^2 \approx \|\mathbf{x}-\mathbf{v}_i\|^2$, then $u_{ir} \rightarrow 1$. So, $\Delta\mathbf{v}_i$ increases. In summary, the presence of $\mathbf{v}_r$, such that $\|\mathbf{x}-\mathbf{v}_r\|^2 \approx \|\mathbf{x}-\mathbf{v}_i\|^2$ increases thè updation of the winning prototype towards the input $\mathbf{x}$, while the presence of $\mathbf{v}_r$, such that such that $\|\mathbf{x}-\mathbf{v}_r\|^2 \gg \|\mathbf{x}-\mathbf{v}_i\|^2$decreases the updation of the winning prototype towards $\mathbf{x}$. This method of competition is intuitively reasonable.

The algorithm can be summarized as folows:

1. Select the codebook of size k; fix α_0, N; set $t = 0$;
   randomly generate the initial set of prototypes $\mathbf{V} = (\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k)$
2. Calculate $\alpha = \alpha_0(1 - \frac{t}{N})$
3. For each input vector $\mathbf{x}$:
   - find $\|\mathbf{x} - \mathbf{v}_i\|^2 = min_{\mathbf{v}_r \in \mathbf{V}}\{\|\mathbf{x} - \mathbf{v}_r\|^2\}$
   - evaluate u_{ir} according to eqn.(17).
   - update winning prototype $\mathbf{v}_i$ and the non-winning prototypes $\mathbf{v}_j \neq \mathbf{v}_i$ according to eqn.(18a) and (18b) respectively.
4. if $t < N$, then $t = t + 1$ and go to step 2.

The performance of the of the existing algorithms for the data set having unequal sized classes is poor (chapter 4). The cluster centers produced by them are away from the class means. This is because the updation for the non-winner prototypes for these methods, given by equation (12b),(14b),(15b),(16b) are comparatively large. So, in the initial phase of the learning process the total updation for the prototype vector of the smaller class, as the non-winner(when $\mathbf{x}$ belonging to the larger class is processed) is much larger than the total updation, as the winner(when $\mathbf{x}$ corresponding to the smaller class is processed). So the prototype corresponding to the smaller class is being pulled away from the physical class mean, towards the larger class.

We show below, that the non-winner updation for the proposed method is comparatively smaller than those for the existing methods, thus reducing the displacement of the cluster center for the smaller class from its physical class mean.

For the proposed method $u_{ij}(\frac{\|\mathbf{x}-\mathbf{v}_j\|^2}{\|\mathbf{x}-\mathbf{v}_i\|^2} - 1)$ in eqn.(18b) with u_{ij} specified by eqn.(17) has maximum value of $\frac{1}{e} \approx 0.36$.During the initial phase of the learning process, $\|\mathbf{x} - \mathbf{v}_r\|^2 \approx \|\mathbf{x} - \mathbf{v}_i\|^2$ and so, in this phase the value of u_{ir} for Harmonic FALVQ, Geometric FALVQ and Arithmetic FALVQ will be greater than $\frac{1}{2}$. So, in $\Delta\mathbf{v}_j$ given by eqn.(14b),(15b),(16b) the term $u_{ij}^2 > \frac{1}{4} = .25$. So, in many cases, it happens that for these algorithms, the value of the term contributing for the updation of the non-winner is

greater than .36 or approximately $\frac{1}{e}$, that is, $\Delta \mathbf{v}_j$ for these is greater than the $\Delta \mathbf{v}_j$ for the proposed method. Due to this, the non-winner prototype will be pulled more towards $\mathbf{x}$ by these algorithms than the proposed one. In FALVQ $u_{ir} \in (0, \frac{1}{2})$. So, $u_{ir}^2 < \frac{1}{4}$. Hence the difference between $\Delta \mathbf{v}_j$ for FALVQ and the proposed method is not very significant. However, as already said the bias present in the definition of u_{ir} of FALVQ detoriorates the performance of FALVQ.

The proposed method is also not affected by scaling of data. Hence this is scale invariant in the sense of the following proposition.

Proposition :
Let $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} \subseteq \mathcal{R}^p$ and
let $\mathbf{V}_0 = (\mathbf{v}_{1,0}, \mathbf{v}_{2,0}, \ldots, \mathbf{v}_{k,0}), \mathbf{v}_{i,0} \in \mathcal{R}^p$ be a set of initial prototypes.
Let τ be fixed positive number and define sets of scaled data and initial prototypes by:

$$\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n\} = \tau\mathbf{X} = \{\tau\mathbf{x}_1, \tau\mathbf{x}_2, \ldots, \tau\mathbf{x}_n\} \; and$$

$$\mathbf{W}_0 = (\mathbf{w}_{1,0}, \mathbf{w}_{2,0}, \ldots, \mathbf{w}_{k,0}) = \tau\mathbf{V}_0 = (\tau\mathbf{v}_{1,0}, \tau\mathbf{v}_{2,0}, \ldots, \tau\mathbf{v}_{k,0})$$

Then applying the proposed method to $\mathbf{X}$, initialized by $\mathbf{V}_0$, is equivalent to applying the proposed method to $\mathbf{Y}$, initailized by $\mathbf{W}_0$, in the sense that

$$\mathbf{w}_{j,t} = \tau\mathbf{v}_{j,t} \;\; : j = 1, 2, \ldots, k \; and \; t = 0, 1, \ldots$$

Proof: We initially show that the membership values $\{u_r\}$ for $\mathbf{X}$ and $\mathbf{Y}$ are identical, because they involve ratios of norm values, so the scaling factor τ cancels out,just as in

$$exp(1 - \frac{\|\mathbf{y} - \mathbf{w}_r\|^2}{\|\mathbf{y} - \mathbf{w}_i\|^2}) = exp(1 - \frac{\|\tau\mathbf{x} - \tau\mathbf{v}_r\|^2}{\|\tau\mathbf{x} - \tau\mathbf{v}_i\|^2})$$

$$= exp(1 - \frac{\|\mathbf{x} - \mathbf{v}_r\|^2}{\|\mathbf{x} - \mathbf{v}_i\|^2})$$

We use induction to prove the proposition. The result holds for $t = 0$ by hypothesis. We now assume it to hold for arbitrary $t - 1$, and show, it holds for t.

By eqn.(18a)

$$\mathbf{w}_{i,t} = \mathbf{w}_{i,t-1} + \alpha'(1 + \sum_{r \neq i}^{k} u_{ir}(\frac{\|\mathbf{y} - \mathbf{w}_{r,t-1}\|^2}{\|\mathbf{y} - \mathbf{w}_{i,t-1}\|^2})^2)(\mathbf{y} - \mathbf{w}_{i,t-1})$$

for the winner prototype.
So,

$$\begin{aligned}
\mathbf{w}_{i,t} &= \mathbf{w}_{i,t-1} + \alpha'(1 + \textstyle\sum_{r \neq i}^{k} u_{ir}(\frac{\|\tau\mathbf{x} - \tau\mathbf{v}_{r,t-1}\|^2}{\|\tau\mathbf{x} - \tau\mathbf{v}_{i,t-1}\|^2})^2)(\tau\mathbf{x} - \tau\mathbf{v}_{i,t-1}) \\
&= \mathbf{w}_{i,t-1} + \alpha'(1 + \textstyle\sum_{r \neq i}^{k} u_{ir}(\frac{\|\mathbf{x} - \mathbf{v}_{r,t-1}\|^2}{\|\mathbf{x} - \mathbf{v}_{i,t-1}\|^2})^2)\tau(\mathbf{x} - \mathbf{v}_{i,t-1}) \\
&= \tau\mathbf{v}_{i,t-1} + \alpha'\tau(1 + \textstyle\sum_{r \neq i}^{k} u_{ir}(\frac{\|\mathbf{x} - \mathbf{v}_{r,t-1}\|^2}{\|\mathbf{x} - \mathbf{v}_{i,t-1}\|^2})^2)(\mathbf{x} - \mathbf{v}_{i,t-1}) \\
&= \tau[\mathbf{v}_{i,t-1} + \alpha'(1 + \textstyle\sum_{r \neq i}^{k} u_{ir}(\frac{\|\mathbf{x} - \mathbf{v}_{r,t-1}\|^2}{\|\mathbf{x} - \mathbf{v}_{i,t-1}\|^2})^2)(\mathbf{x} - \mathbf{v}_{i,t-1})]
\end{aligned}$$

since u_{ir} is independent of τ

$$\mathbf{w}_{i,t} = \tau\mathbf{v}_{i,t}$$

By eqn.(18b)

$$\mathbf{w}_{j,t} = \mathbf{w}_{j,t-1} + \alpha' u_{ij}(\frac{\|\mathbf{y} - \mathbf{w}_{j,t-1}\|^2}{\|y - \mathbf{w}_{i,t-1}\|^2} - 1)(\mathbf{y} - \mathbf{w}_{j,t-1})$$

for non-winner prototype $j \neq i$
So,

$$\begin{aligned}
\mathbf{w}_{j,t} &= \mathbf{w}_{j,t-1} + \alpha' u_{ij}(\frac{\|\tau\mathbf{x} - \tau\mathbf{v}_{j,t-1}\|^2}{\|\tau\mathbf{x} - \mathbf{v}_{i,t-1}\|^2} - 1)(\tau\mathbf{x} - \tau\mathbf{v}_{j,t-1}) \\
&= \tau\mathbf{v}_{j,t-1} + \alpha' u_{i,j}\tau(\frac{\|\mathbf{x} - \mathbf{v}_{j,t-1}\|^2}{\|\mathbf{x} - \mathbf{v}_{i,t-1}\|^2} - 1)(\mathbf{x} - \mathbf{v}_{j,t-1}) \\
&= \tau[\mathbf{v}_{j,t-1} + \alpha' u_{i,j}(\frac{\|\mathbf{x} - \mathbf{v}_{j,t-1}\|^2}{\|\mathbf{x} - \mathbf{v}_{i,t-1}\|^2} - 1)(\mathbf{x} - \mathbf{v}_{j,t-1})]
\end{aligned}$$

since u_{ij} is independent of τ

$$\mathbf{w}_{j,t} = \tau \mathbf{v}_{j,t} \qquad for\ j \neq i$$

So, for each t,

$$\mathbf{w}_{i,t} = \tau \mathbf{v}_{i,t}$$

$$\mathbf{w}_{j,t} = \tau \mathbf{v}_{j,t} \quad j \neq i$$

Hence the proposition.

Chapter 4

Experimental Results and Comparison

In order to judge the performance of the algorithms, we have used the following measures:

- number of misclassification.
- a measure Z (named as "total distortion") and it is defined as,

$$Z = \sum_{\mathbf{x} \in \mathbf{X}} \sum_{r=1}^{k} u_{ir} \|\mathbf{x} - \mathbf{v}_r\|^2$$

(Note that $u_{ir} = 1$ for $r = i$ for all the algorithms. When $r \neq i$ each algorithm provides its own value of u_{ir}.)

4.1 Results on IRIS data

The GLVQ, FALVQ, Harmonic FALVQ, Geometric FALVQ, Arithmetic FALVQ and the proposed algorithms were tested using Anderson's IRIS data set, which has extensively been used for evaluating the performance of the pattern classification algorithms[1].This data set contains 150 feature vectors of length four, which belong to 3 classes representing different IRIS subspecies. Each class contains 50 feature vectors. One of the 3 classes is well seperated from the other two, which are not easily seperable due to the existence of similar vectors. The performance of the algorithms, tested on

this data set is evaluated by counting the number of the classification errors, i.e., the number of feature vectors that are assigned to a wrong cluster by the algorithms[6],[7].

The raw IRIS data were classified by the GLVQ, different FALVQ algorithms, and the proposed algorithms with $N = 500$ and different initial values of the learnimg rate. Table 1 shows the corresponding results.

The GLVQ, FALVQ, Harmonic FALVQ, resulted in 16 or 17 classification error when they are applied on the raw IRIS data set. This is typical when the IRIS data set is classified by the unsupervised algorithms. The proposed method also has 16 or 17 classification errors, with the data set.Geometric FALVQ has a slightly better performance, as it misclassified 12 feature vectors. The Arithmetic FALVQ algorithm is clearly inferior to the others, since in this case it has a classification error 23.

Scaled IRIS data(IRIS/10) is also used for testing the algorithms. The GLVQ algorithm being sensitive to scaling of data, resulted in 50 classification errors with $\alpha = 0.6$. While the FALVQ, Harmonic FALVQ, Geometric FALVQ, Arithmetic FALVQ are scale invariant. The proposed algorithm, as already proved is also scale invariant. Hence their performance is not affected using scaled IRIS data .

The performance of these algorithms are also tested using data set containing unequal sized classes. From the IRIS data set, 3 classes are formed, where class 1, class 2, class 3 contain 50, 30, 10 feature vectors respectively taken from the corresponding class 1, class 2,class 3 feature vectors of the IRIS data set. So the number of feature vectors in this data set is 90. The algorithms are tested on this data set of unequal sized class with $N = 500$ and $\alpha_0 = 0.05$. Proposed method along with FALVQ gives the best performance, as it misclassified only 2 feature vectors. Harmonic FALVQ misclassified 4 feature vectors. The performance of the Arithmetic FALVQ is worst since it misclassified 26 feature vectors. Table 3 shows the corresponding results.

4.2 Results on Artificial Data Sets generated in $\mathcal{R}^2$

The performance of the algorithms are also tested using Artificial Data sets. The artificial data set contains two classes which are of unequal sizes. Two classes are generated using 1000 points. The points are uniformly distributed in each class. The class 1 has an apriori probability

of 0.8, while the class 2 has an apriori probability of 0.2. So class 1 has larger number of points than the class 2. The number of points in class 1 is 790, whereas in class 2 it is 210. Figures 3, 10, 17 shows the two classes with interclass distances 0.5, 0.2, 0.01 unit respectively. The class 1 has a radius of 2 unit with center at $(0, 0)$. Class 2 has a radius of 1 unit with center at (3+interclass distance,0). The physical class means of the classes are therefore, their centers.

The performances of the algorithms on these data sets are shown in tables 4, 5, 6, with $N = 500$ and $\alpha_0 = 0.005$. For interclass distance=0.5, the proposed method classified all feature vectors correctly. Also for interclass distances 0.2 and 0.01, the proposed method gives the best performance by misclassifying 4 and 22 feature vectors respectively. As can be seen from the tables 4, 5, 6 numbers of misclassification by the proposed method is much less than those of the other algorithms. For all these data sets, the next best performance is given by FALVQ and the worst performance is given by Arithmetic FALVQ. Tables 7, 8, 9 show the list of the cluster centers obtained by the algorithms for the artificial data sets. with $N = 500$ and $\alpha_0 = 0.005$. Figures. 24, 25, 26 give the graphical representation of the same cluster centers. In the graph, cluster centers produced by an algorithm is marked by a number. The left position of the number represents the center for the class 1, while the right position is the center for class 2.

It can be seen that, for the existing methods the cluster center for class 2 has been pulled towards the larger class, while the proposed algorithm obtains the cluster center for class 2 closest to its class means.

The performance of the algorithms are also tested using the distortion measure Z. Table 10 shows the total distortion obtained by different algorithms on different data set. The values are obtained after $N = 500$ and with $\alpha_0 = 0.005$. The distortion obtained using the proposed algorithm is minimum among all algorithms, for each of the data set.

4.3 Results on IRS Imagery

The performance of the proposed algorithm is tested on IRS Imagery. IRS stands for Indian Remote Sensing Satellite. The data used for this work, is taken from the satellite IRS-1B. The satellite is equipped with 2 different sensors-LISS I and LISS II. Data used for this work is from LISS II sensor. LISS II has a focal length of 324.4 meter with a spectral range between 0.45 -0.86 micrometer. The whole spectral range has been divided into 4 bands,

namely Blue ($0.45-0.52\ \mu m$), Green ($0.52-0.59\ \mu m$), Red ($0.62-0.68\mu m$), Infrared ($0.77-0.86\mu m$).

The scene used for evaluating the performance of the algorithm is Calcutta scene. 256 x 256 image for each of the four bands are taken. Figures 27, 28, 29, 30 show the corresponding Band 1, Band 2, Band 3, Band 4 images. The region primarily consists of 6 different types of landcovers. The 6 classes are Clear water, Turbid Water, Concrete Structures, Habitation, Vegetation and Open space.
The constituents of these classes are described below.

1. Pure Water: This class contains pond water.
2. Turbid Water: This class contains rivers.
3. Concrete: This class contains buildings, railway lines, roads.
4. Habitation: This class basically consists of suburban and rural habitation i.e. concrete structures but comparatively less in density.
5. Vegetation: This class represents the crop area and the forest area.
6. Open Space: This class contains Barren land , sand.

The proposed algorithm is used for clustering the pixels in this IRS image, with number clusters k taken to be 6,5,4 and 3. The best results are obtained with $k = 3$. Reconstructed image with $k = 3$ is shown in the fig.31. Each of the 3 classes in this image is shown seperately in Figures 32, 33, 34. It has been possible to label 2 clusters among 3 clusters in the images. These 2 classes which can be clearly identified are Water and Land, with the third class consisting of very few pixels (noise pixels). The results with $k = 4$, $k = 5$, $k = 6$ are however not satisfactory.

Chapter 5

Conclusions

Here we have presented a new fuzzy learning vector quantization algorithm. The algorithm uses a membership function which is continuous on $(0, 1]$ and hence, unlike the other algorithms it removes the inherent bias towards the winner. This helps in increasing the competitive effect among the prototypes. The large non-winner updation, that were present in other algorithms, is also eliminated. Hence the performance of this algorithm for a data set having unequal sized classes is much better. The algorithm is tested using different number of iterations, and also different learning rates. It is also tested with different initialization. The results obtained in all these cases are same. Total distortion obtained by this algorithm is also the least. The proposed algorithm assumes that the number of clusters present in the data set is greater than=2. Experiments are done with data sets which are non-overlapping, where the algorithm performs better than all other existing methods. Unlike the previous algorithms, the proposed algorithm requires slightly more computations as it requires to compute exponential membership functions.

A possible way for the improvement in all these algorithms is to find the concrete mathematical setup with theorems and proofs which judge the performance of the algorithms and use the results for further modifications. This however is a difficult task.

Chapter 6

Tables and Figures

Algorithm	α_0	Classification Errors
GLVQ	0.5	17
	0.6	17
	0.05	17
FALVQ	0.05	17
	0.005	16
Harmonic FALVQ	0.05	16
	0.005	16
Geometric FALVQ	0.05	12
	0.005	12
Arithmetic FALVQ	0.05	23
	0.005	23
Proposed Method	0.05	17
	0.005	16

TABLE 1: Performance of the algorithms on the IRIS data set, N=500.

Algorithms	α_0	Classification Errors
GLVQ	0.6	50
FALVQ	0.005	16
HFALVQ	0.005	16
GFALVQ	0.005	12
AFALVQ	0.005	23
Proposed Method	0.005	17

TABLE 2: Performance of the algorithms on the scaled IRIS data(IRIS/10), N=500.

Algorithms	Classification Error
GLVQ	2
FALVQ	2
Harmonic FALVQ	4
Geometric FALVQ	18
Arithmetic FALVQ	26
Proposed Method	2

TABLE 3:Performance of the algorithms on 3 unequal sized classes, with feature vectors taken from IRIS data set, N=500
class 1 contains 50 vectors.
class 2 contains 30 vectors.
class 3 contains 10 vectors.
$\alpha_0 = 0.05$

Algorithms	Classification Errors
GLVQ	22
FALVQ	22
Harmonic FALVQ	32
Geometric FALVQ	62
Arithmetic FALVQ	128
Proposed Method	0

TABLE 4: Performance of the algorithms on artificial data set with 2 classes of unequal size, interclass distance = 0.5, $\alpha_0 = 0.005$, N=500.

Algorithms	Classification Errors
GLVQ	47
FALVQ	45
Harmonic FALVQ	53
Geometric FALVQ	99
Arithmetic FALVQ	150
Proposed Method	4

TABLE 5: Performance of the algorithms on artificial data set with 2 classes of unequal size, interclass distance = 0.2, $\alpha_0 = 0.005$, N=500.

Algorithms	Classification Errors
GLVQ	92
FALVQ	74
Harmonic FALVQ	85
Geometric FALVQ	11
Arithmetic FALVQ	167
Proposed Method	22

TABLE 6: Performance of the algorithms on artificial data set with 2 classes of unequal size, interclass distance = 0.01, $\alpha_0 = 0.005$, N=500.

Algorithms	Cluster centers	
	cluster 1	cluster 2
GLVQ	(-0.038,0.027)	(3.24,0.027)
FALVQ	(-0.064,0.014)	(3.27,0.028)
Harmonic FALVQ	(-0.119,0.011)	(3.164,0.029)
Geometric FALVQ	(-0.058,0.002)	(2.62,0.026)
Arithmetic FALVQ	(0.14,-0.003)	(1.628,0.051)
Proposed Method	(0.18,0.02)	(3.81,0.022)
Actual Class Center	(0,0)	(3.5,0)

Table 7 : Actual class centers and cluster centers produced by different algorithms on the artificial data set with inter class distance = 0.5, $N = 500$, $\alpha_0 = 0.005$

Algorithms	Cluster centers	
	cluster 1	cluster 2
GLVQ	(-0.087,0.013)	(2.81,0.037)
FALVQ	(-0.108,0.011)	(2.88,0.033)
Harmonic FALVQ	(-0.15,0.0108)	(2.79,0.03)
Geometric FALVQ	(-0.086,-0.004)	(2.25,0.044)
Arithmetic FALVQ	(0.10,-0.004)	(1.49,0.050)
Proposed Method	(0.18,0.02)	(3.48,0.002)
Actual Class Centers	(0,0)	(3.2,0)

Table 8 : Actual Class Centers and cluster centers produced by different algorithms on the artificial data set with inter class distance = 0.2, $N = 500$, $\alpha_0 = 0.005$

Algorithms	Cluster centers	
	cluster 1	cluster 2
GLVQ	(-0.168,-0.021)	(2.44,0.085)
FALVQ	(-0.154,0.002)	(2.6,0.052)
Harmonic FALVQ	(-0.18,0.003)	(2.53,0.04)
Geometric FALVQ	(-0.09,-0.007)	(2.07,0.05)
Arithmetic FALVQ	(0.07,0.005)	(1.404,0.036)
Proposed Method	(0.131,0.019)	(3.12,0.010)
Actual Class centers	(0,0)	(3.01,0)

Table 9 : Actual class centers and cluster centers produced by different algorithms on the artificial data set with inter class distance = 0.01, $N = 500$, $\alpha_0 = 0.005$

Algorithms	IRIS	Artificial data set		
		$\delta = 0.5$	$\delta = 0.2$	$\delta = 0.01$
GLVQ	225.4	2073.96	2033.79	1997.93
FALVQ	216.48	2131.24	2065.13	2010.04
Harmonic FALVQ	369.21	2648.21	2527.72	2438.23
Geometric FALVQ	752.97	3446.47	3157.055	2978.22
Arithmetic FALVQ	1162.73	4024.84	3620.17	3382.28
Proposed Method	102.88	1360.21	1409.49	1432.13

Table 10: Distortion produced by different algorithms, N=500, $\alpha_0 = 0.005$, δ is the interclass distance between two classes (unequal sizes) in the artficial data set.

Algorithms	Number of Iterations, N		
	N=300	N=500	N=800
GLVQ	22	22	22
FALVQ	22	22	22
Harmonic FALVQ	32	32	32
Geometric FALVQ	62	62	62
Arithmetic FALVQ	128	128	128
Proposed Method	0	0	0

Table 11: Number of misclassification produced by different algorithms on different number of iterations, $\alpha_0 = 0.005$

Algorithms	learning rate, α_0		
	$\alpha_0 = 0.003$	$\alpha_0 = 0.005$	$\alpha_0 = 0.007$
GLVQ	22	22	22
FALVQ	22	22	22
Harmonic FALVQ	32	32	32
Geometric FALVQ	62	62	62
Arithmetic FALVQ	128	128	128
Proposed Method	0	0	0

Table 11: Number of misclassification produced by different algorithms on different learning rates, $N = 500$

Algorithm	Trial				
	#1	#2	#3	#4	#5
Proposed Method	0	0	0	0	0

Table 12: Number of misclassifications on different initialization, $N = 500$, $\alpha_0 = 0.005$.

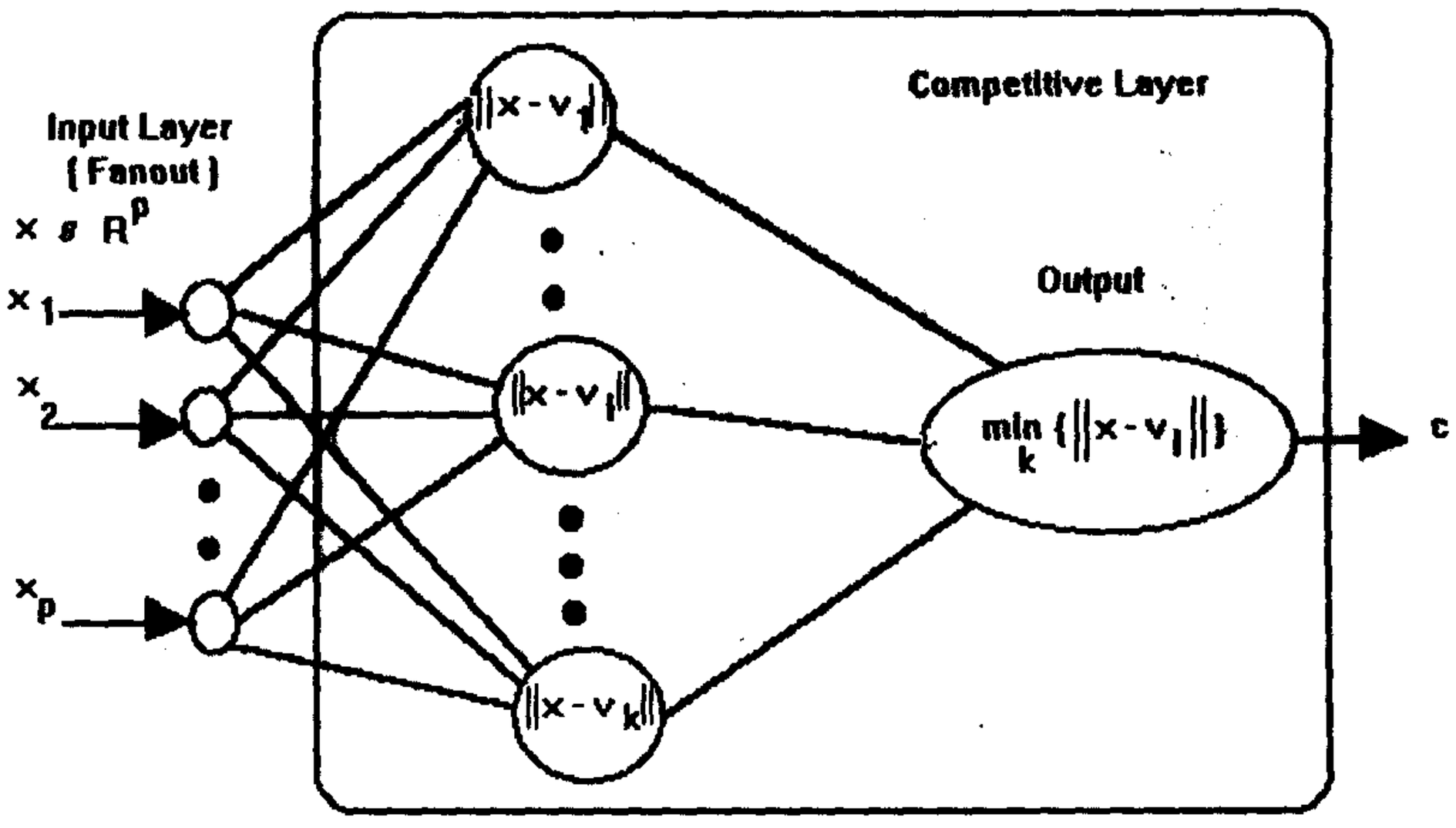


Figure 1. The LVQ Competetive Learning Network

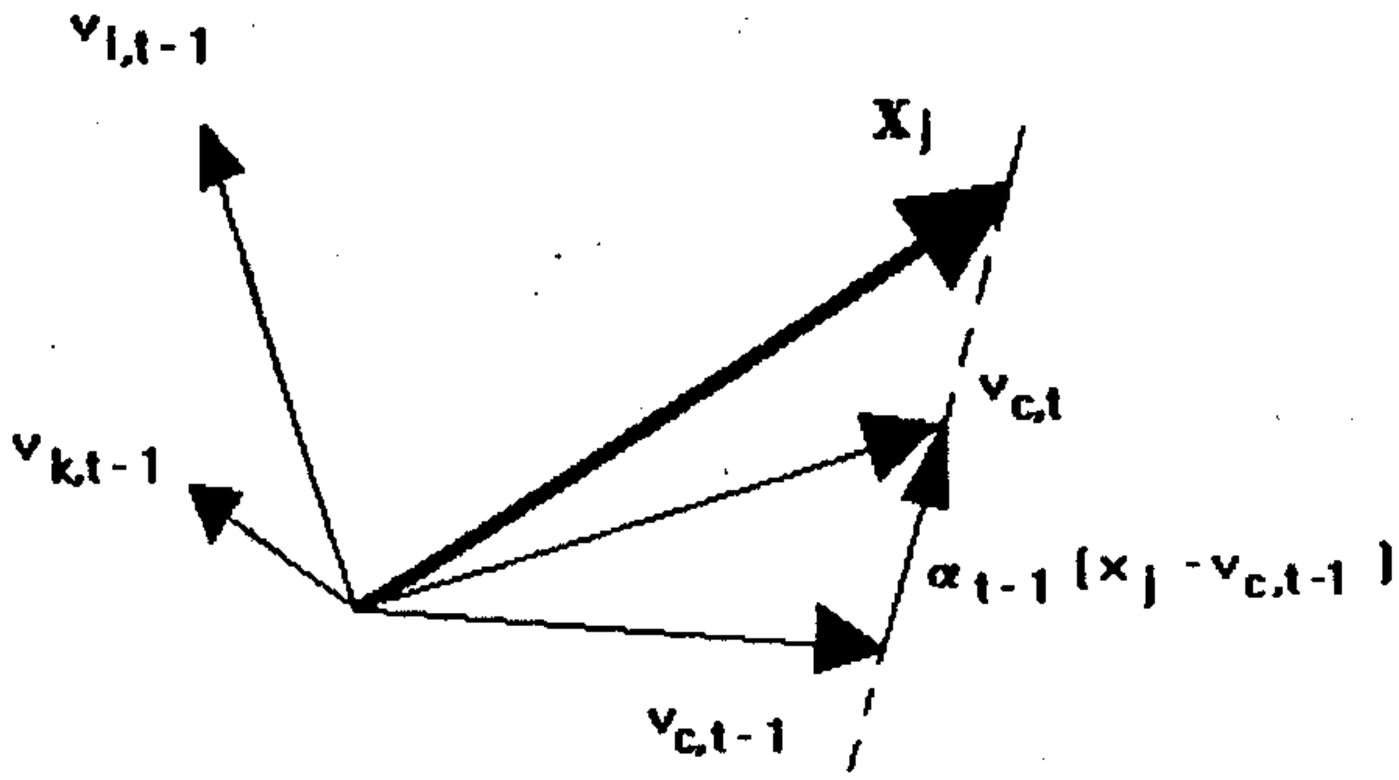


Fig 2. Updating the winning LVQ Prototype

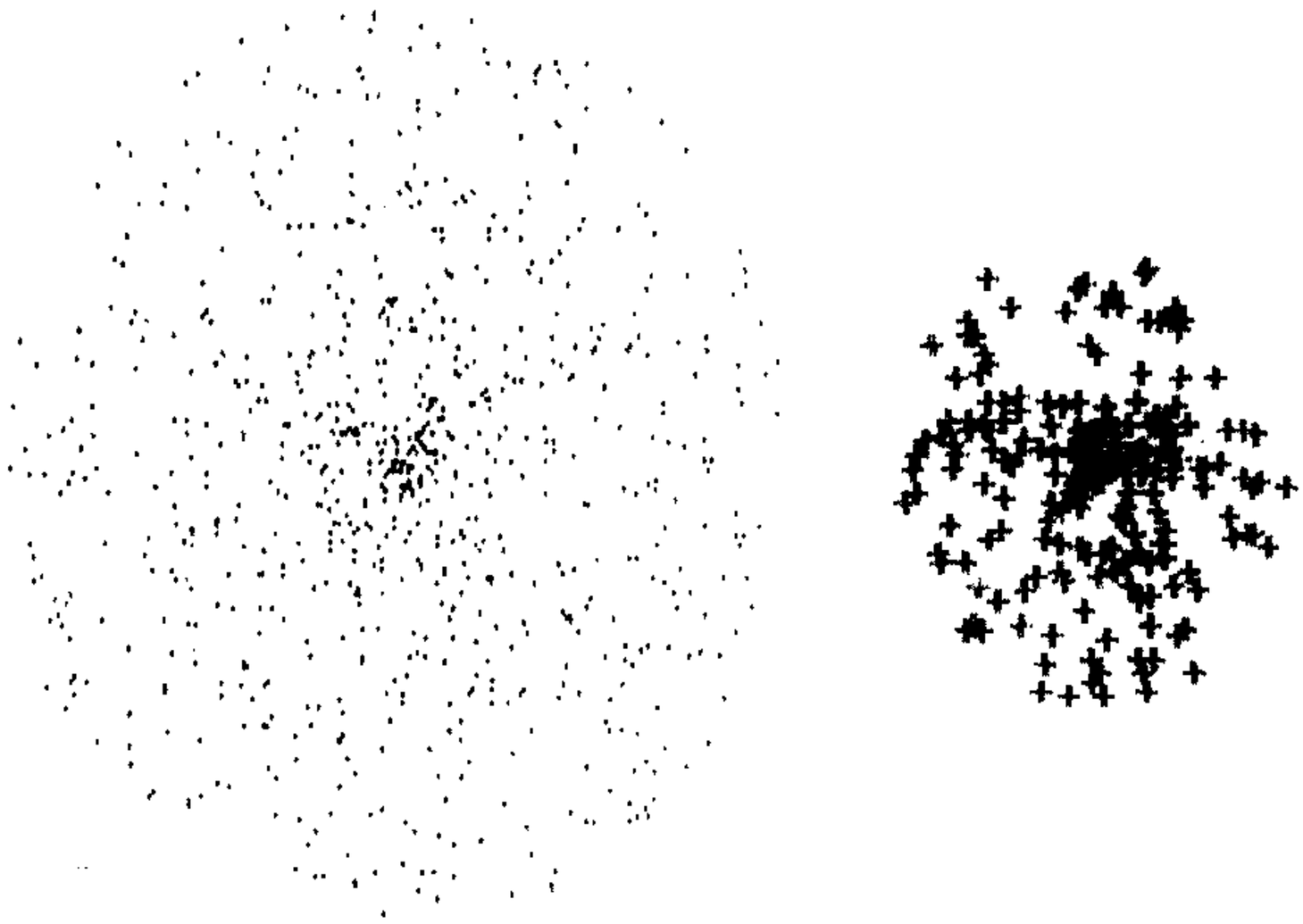


Fig 3: Artificial data set with two classes of unequal size.

interclass distance = 0.5

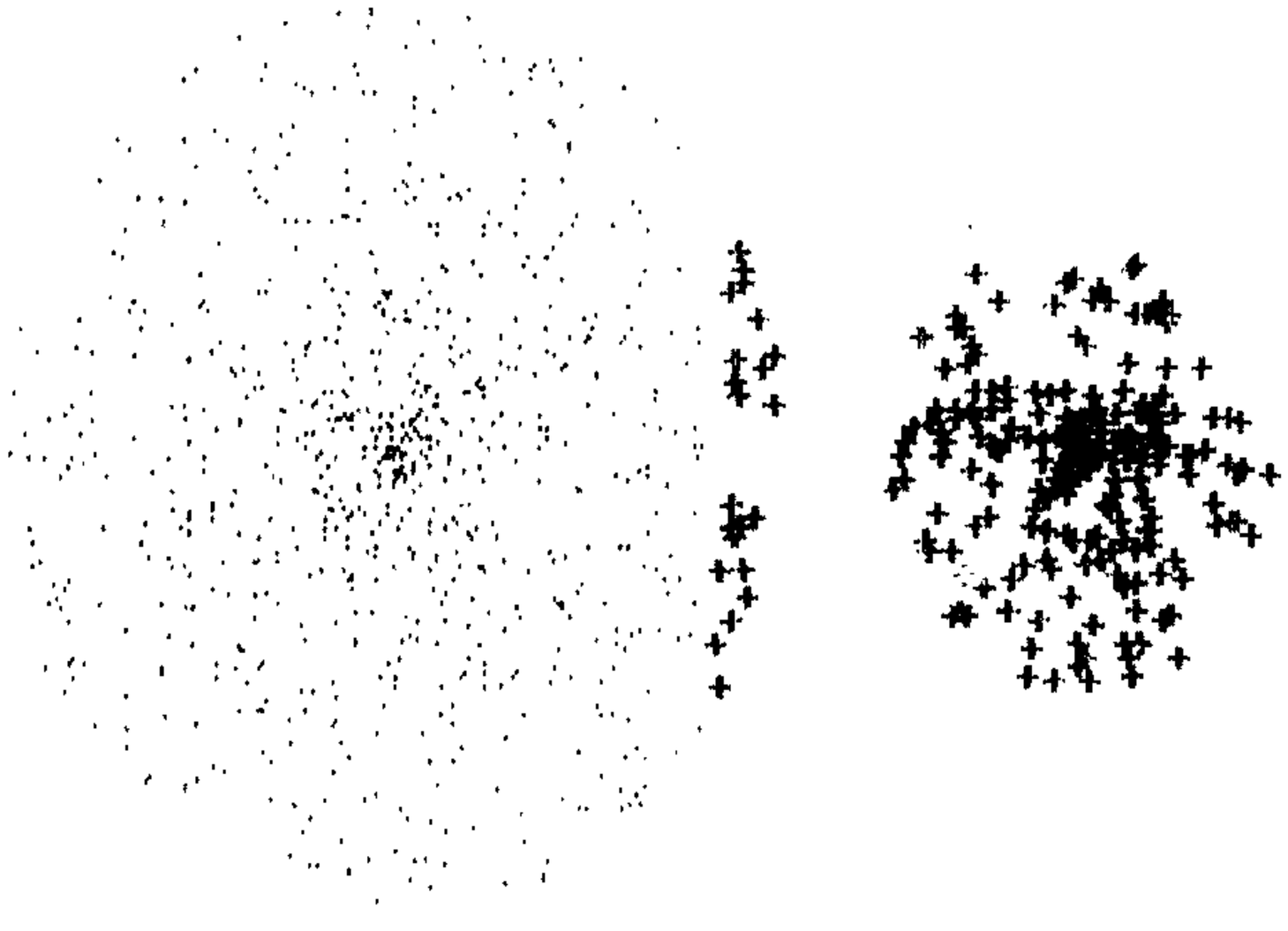


Fig 4: Performance of GLVQ algorithm.

interclass distance = 0.5

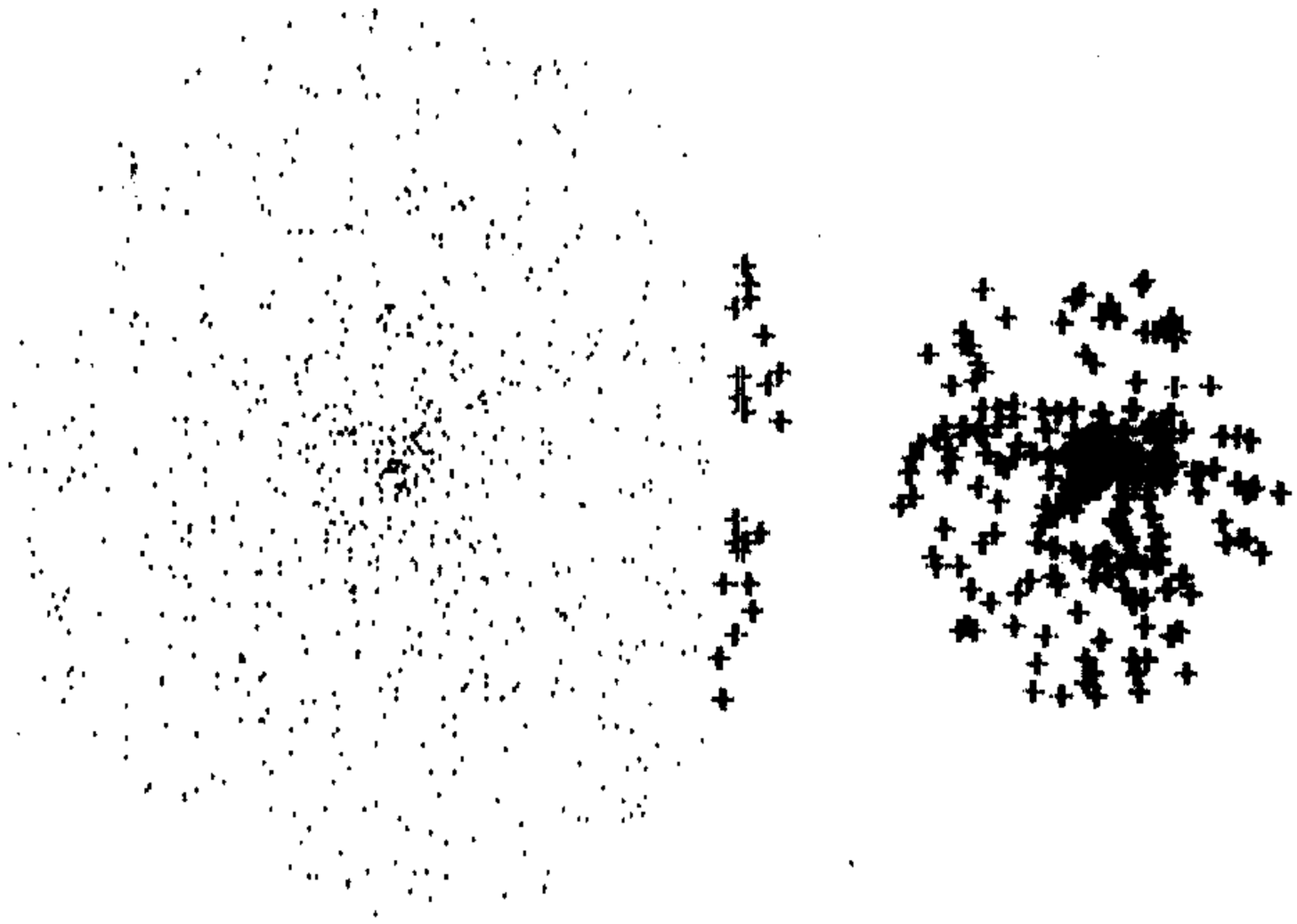


Fig 5: Performance of FALVQ algorithm.

interclass distance = 0.5

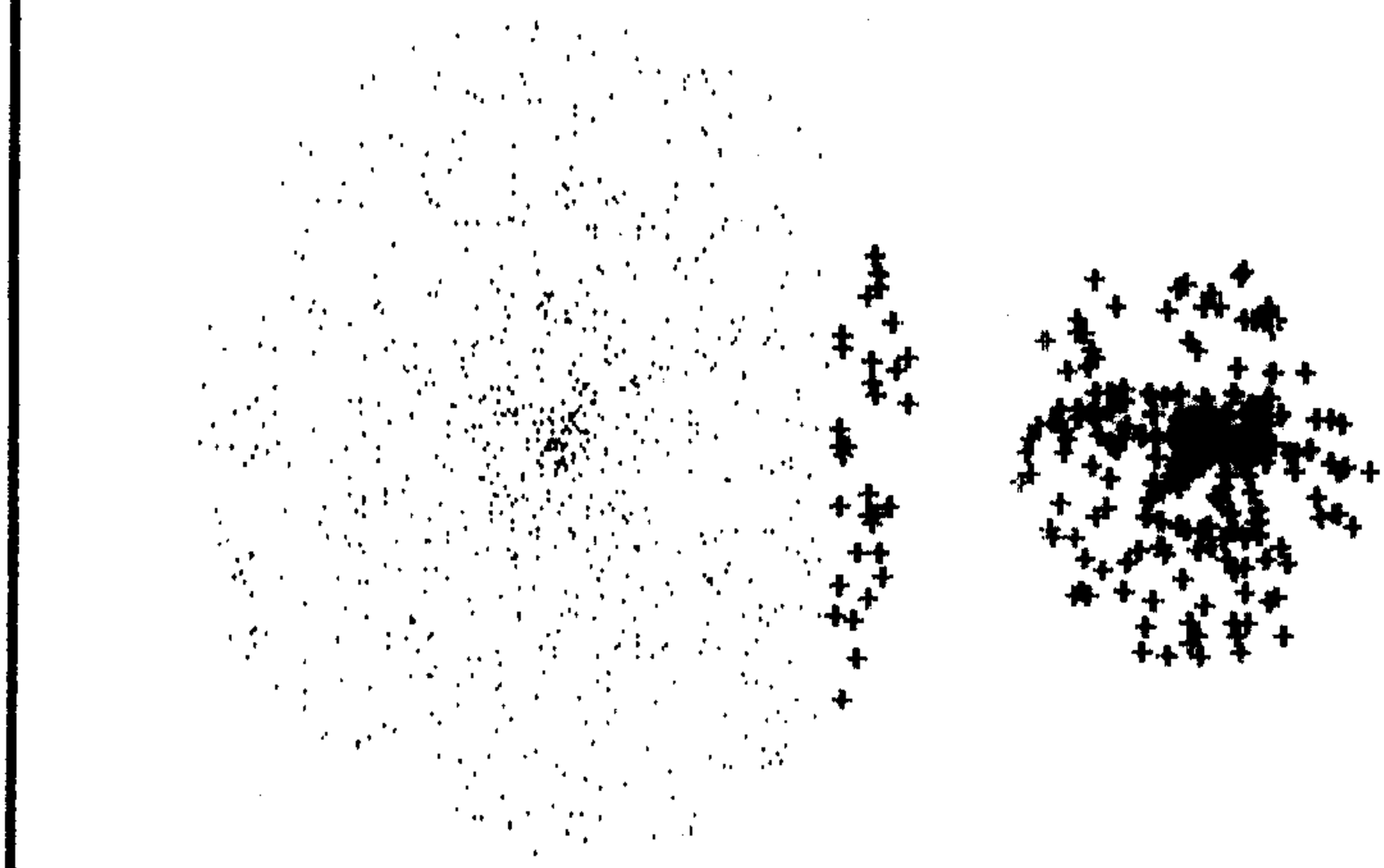


Fig 6: Performance of Harmonic FALVQ algorithm.

interclass distance = 0.5

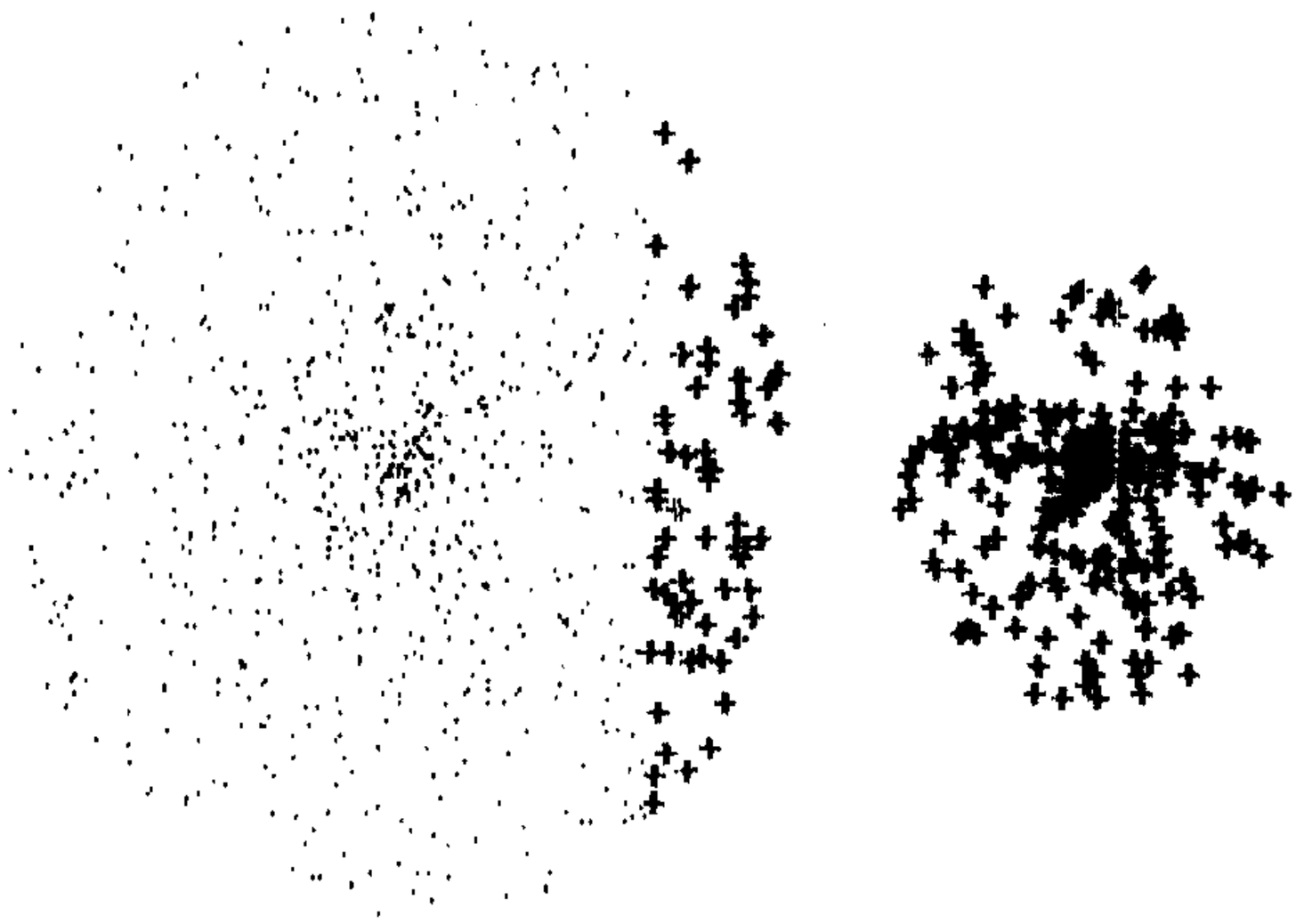


Fig 7: Performance of Geometric FALVQ algorithm.

interclass distance = 0.5

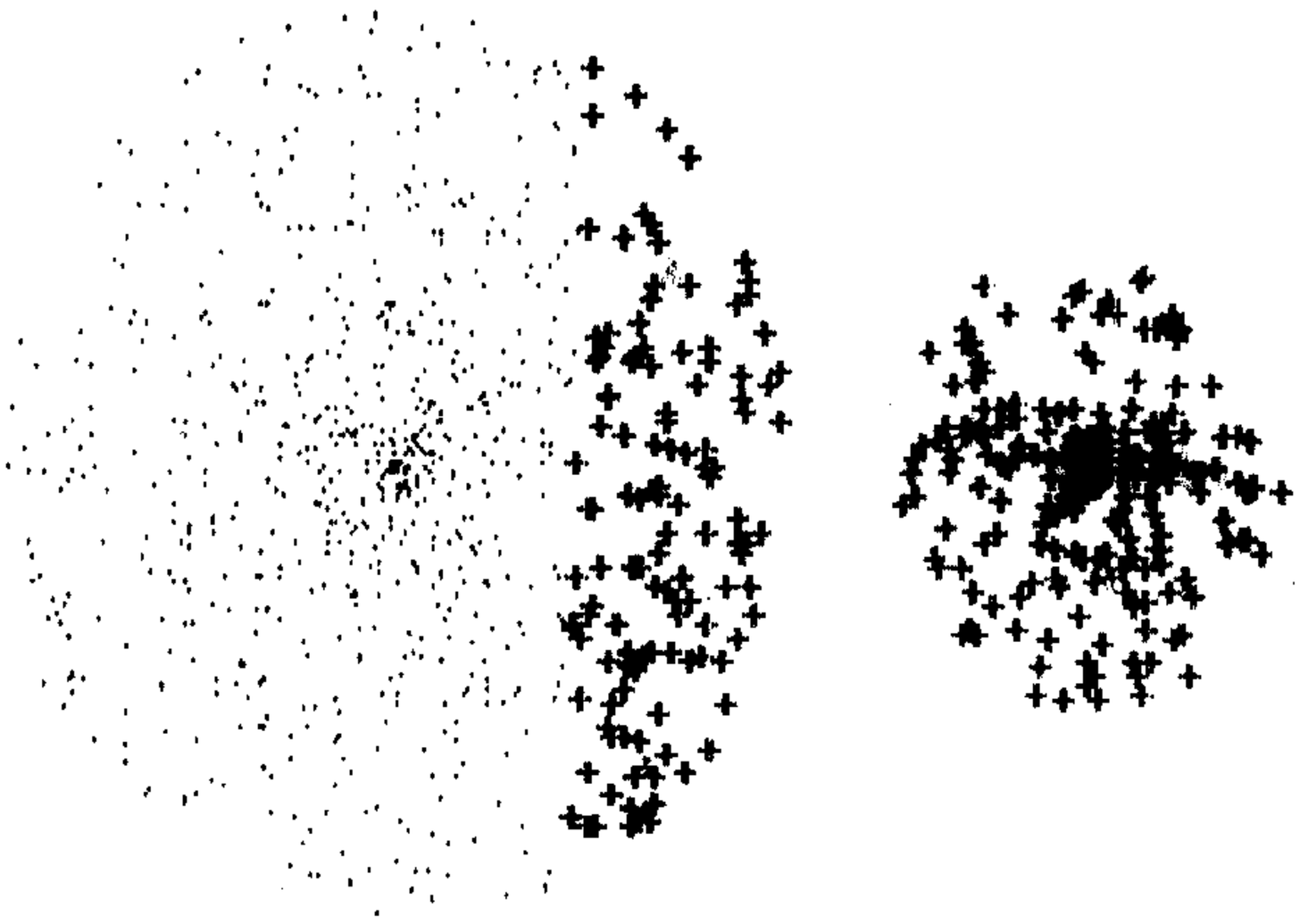


Fig 8: Performance of Arithmetic FALVQ algorithm.

interclass distance = 0.5

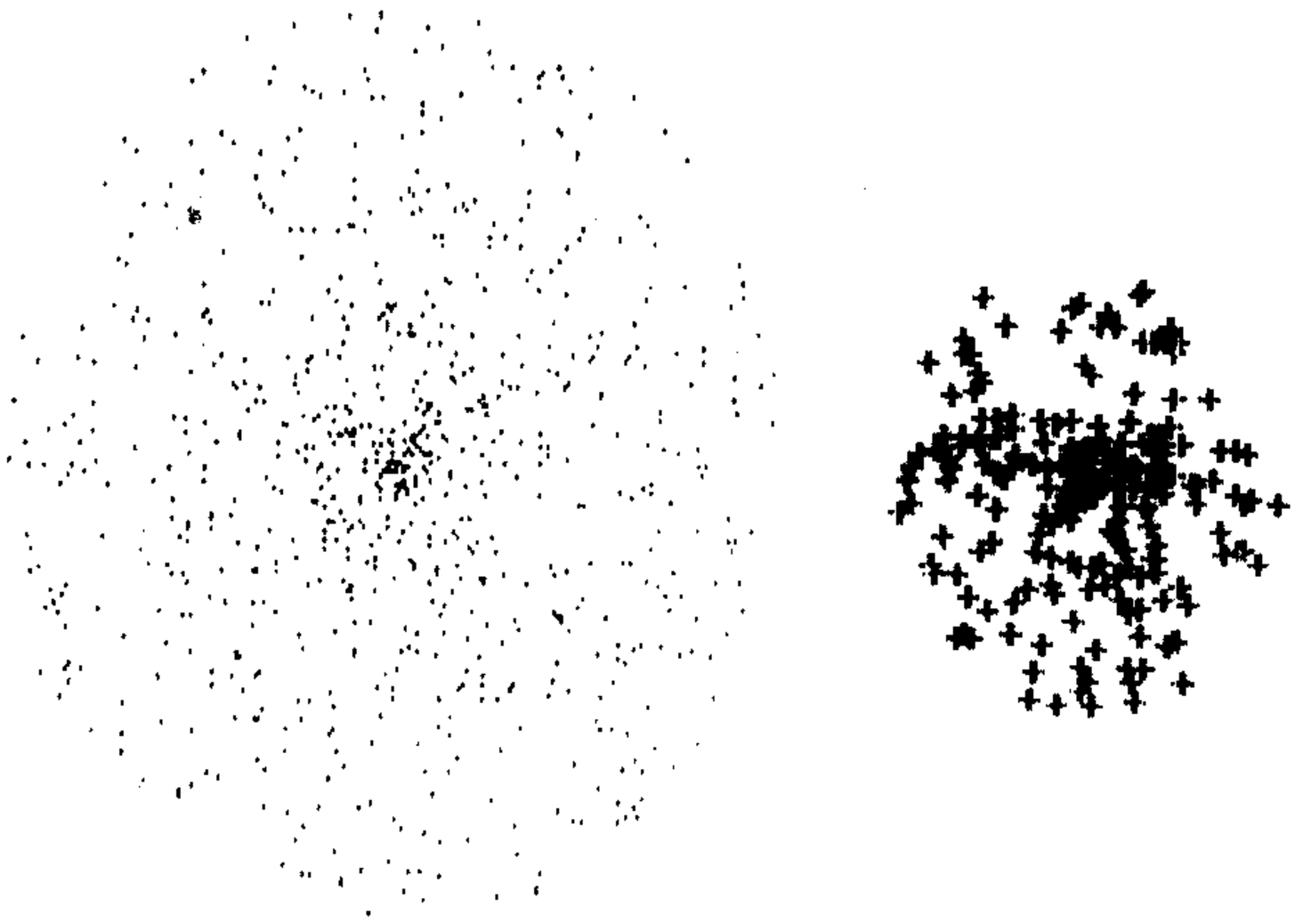


Fig 9: Performance of the proposed algorithm.

interclass distance = 0.5

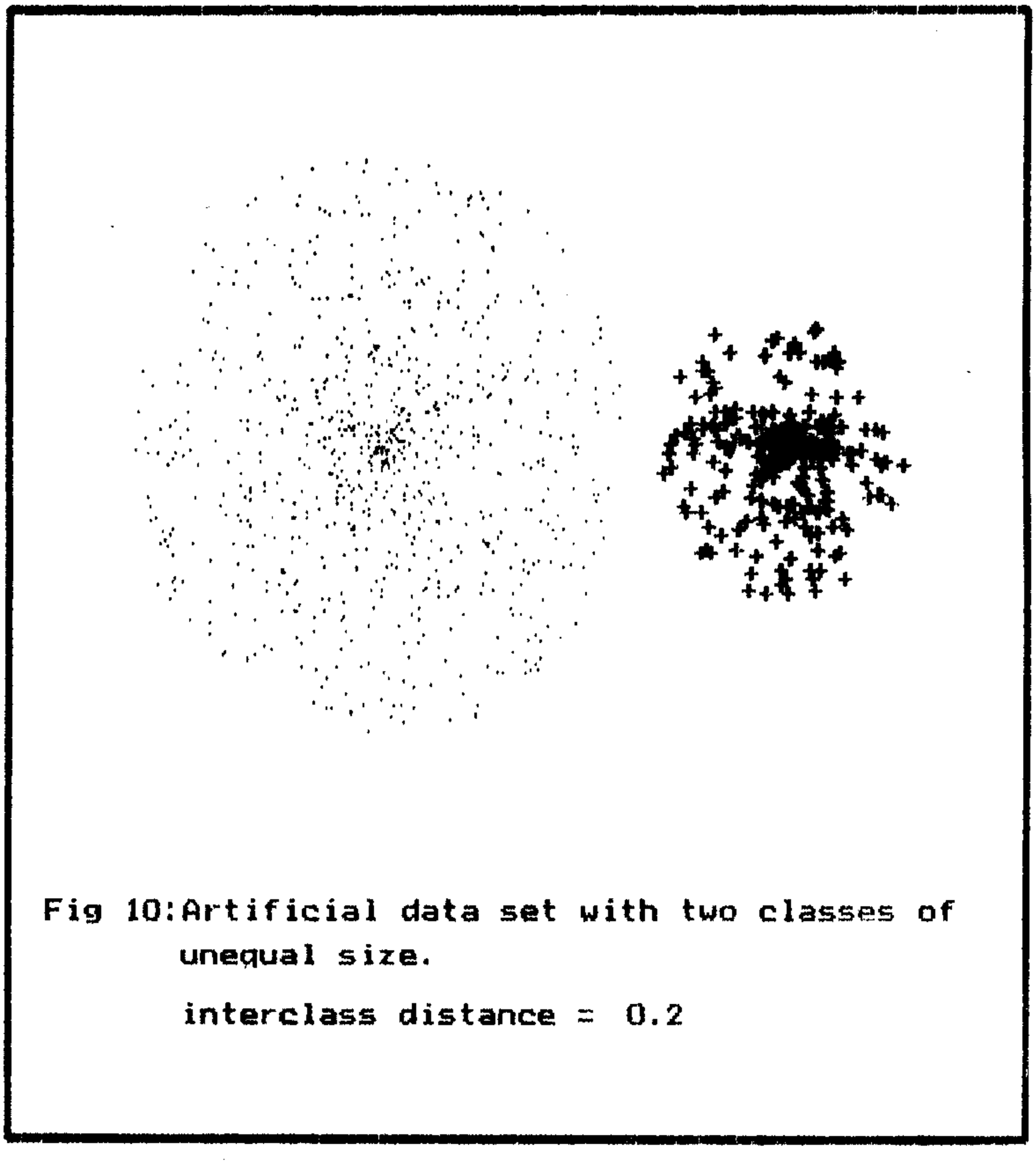


Fig 10:Artificial data set with two classes of unequal size.

interclass distance = 0.2

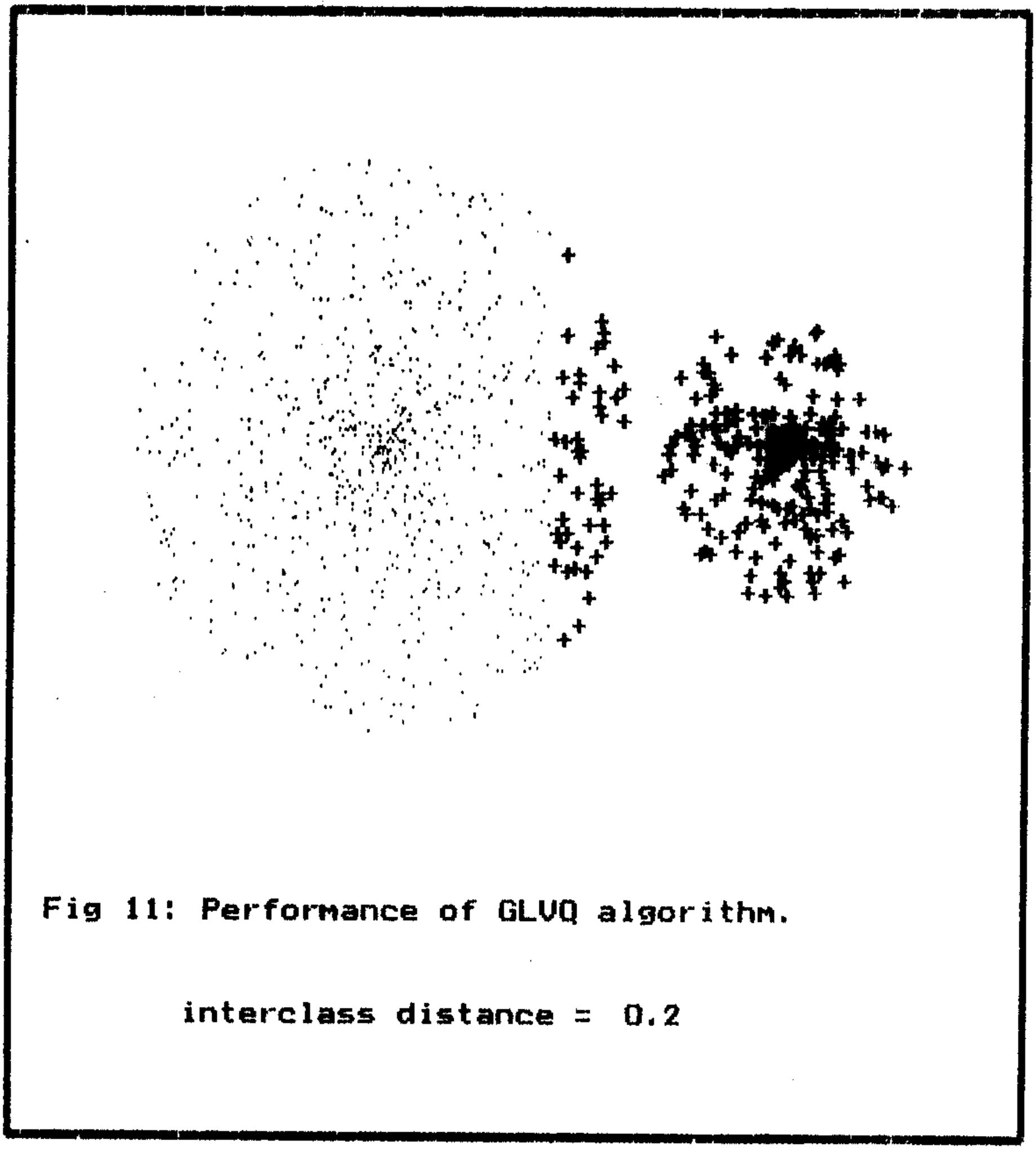


Fig 11: Performance of GLVQ algorithm.

interclass distance = 0.2

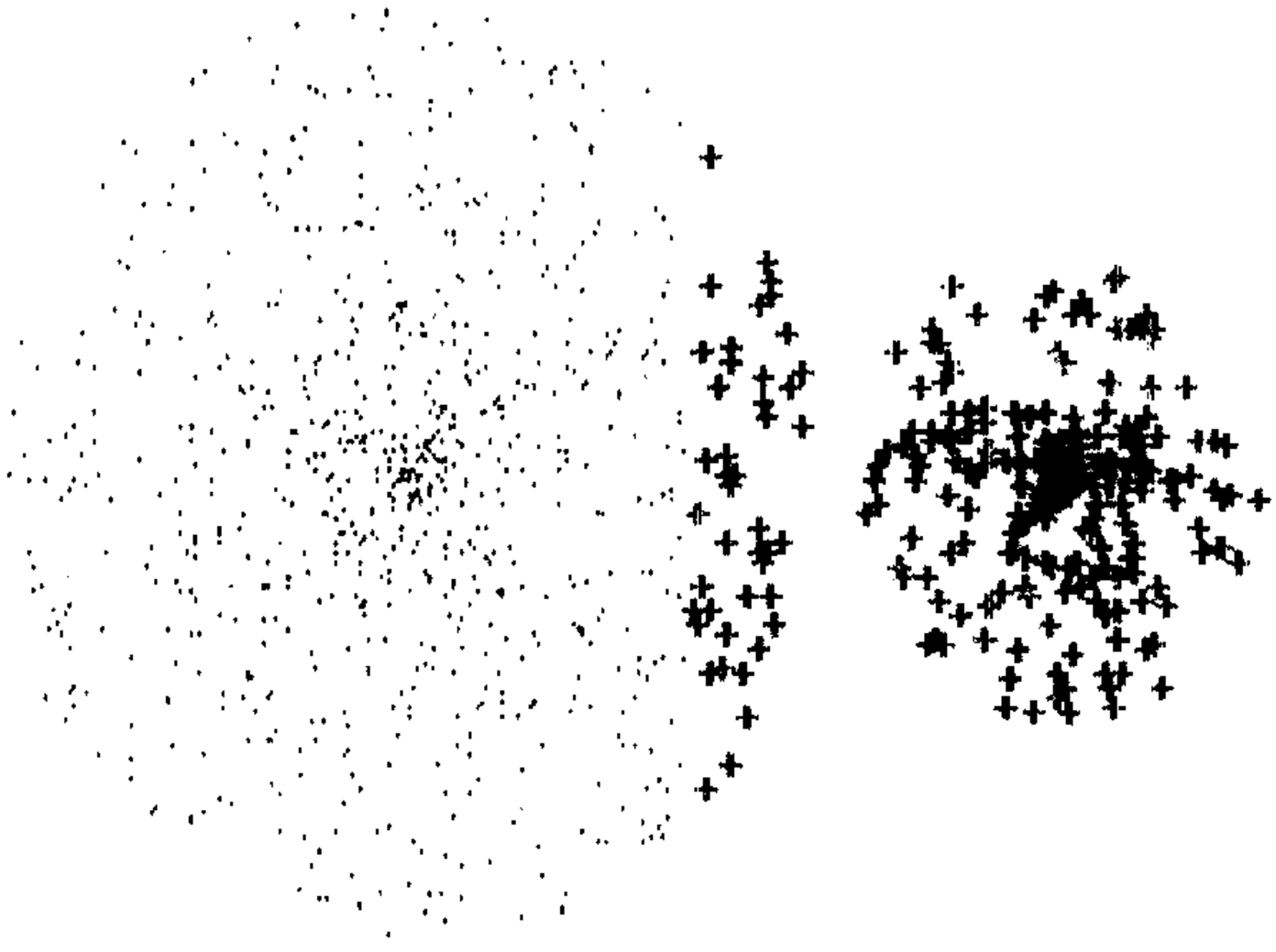


Fig 12: Performance of FALVQ algorithm.

interclass distance = 0.2

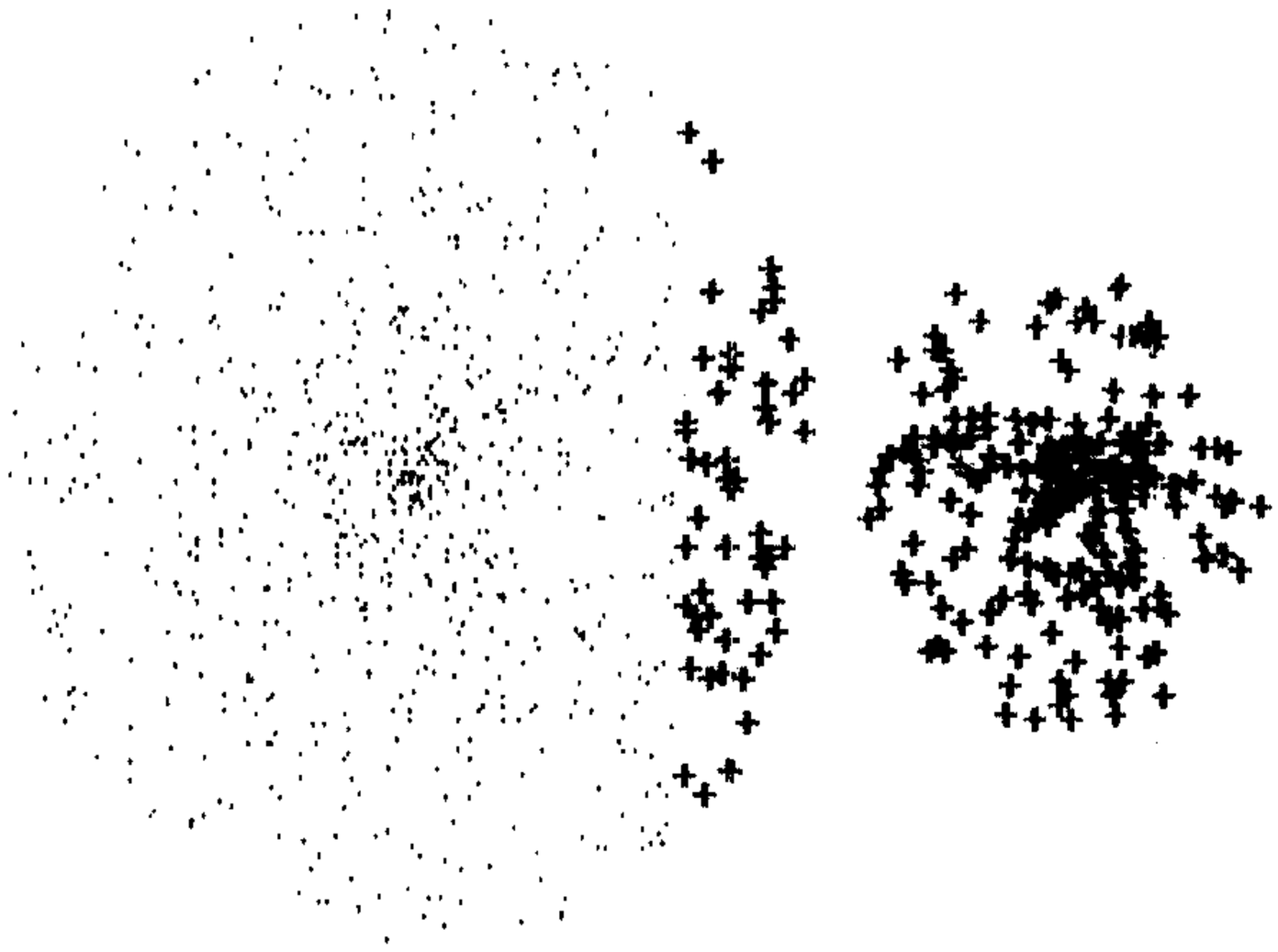


Fig 13: Performance of Harmonic FALVQ algorithm.

interclass distance = 0.2

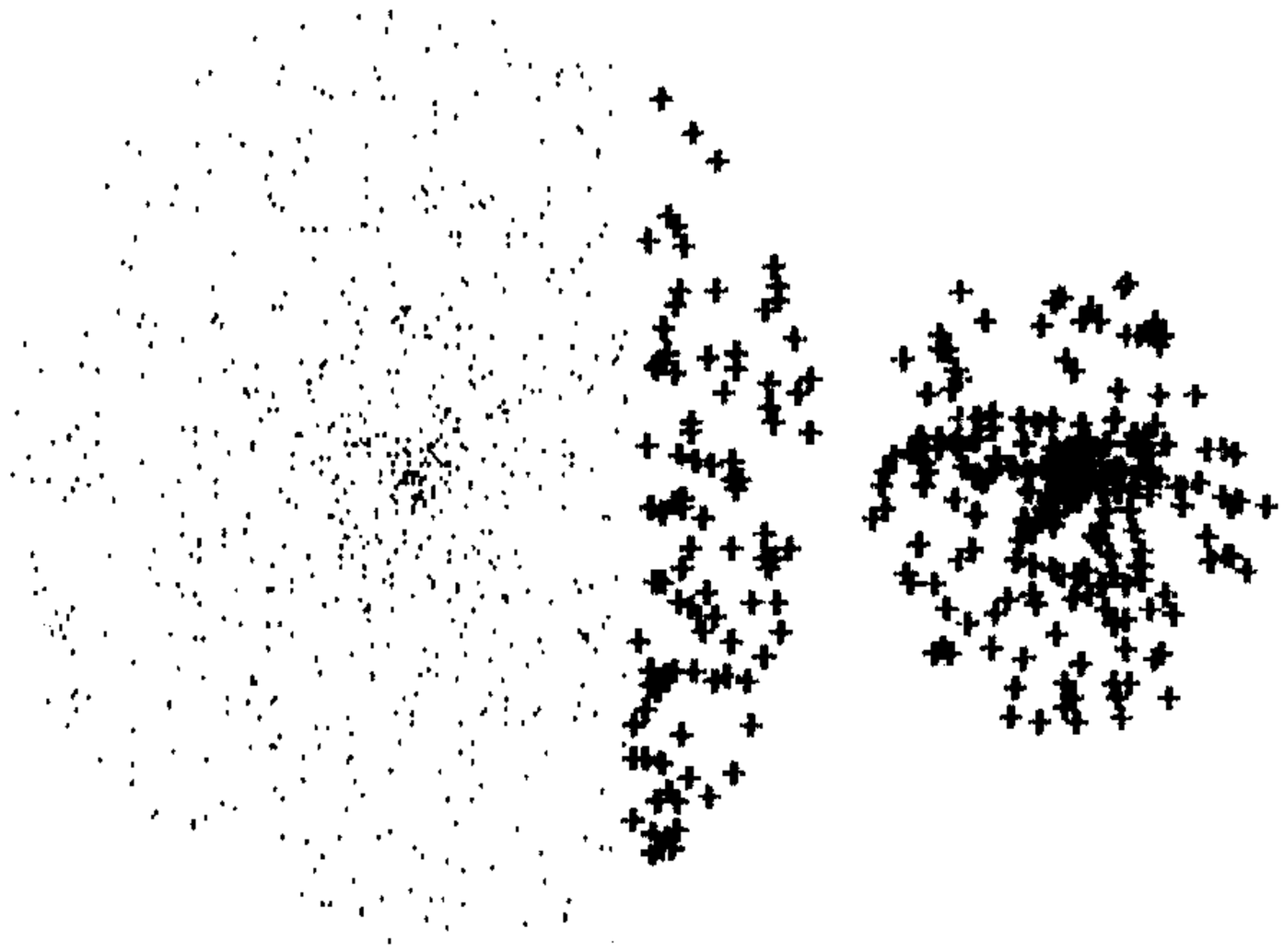


Fig 14: Performance of Geometric FALVQ algorithm.

interclass distance = 0.2

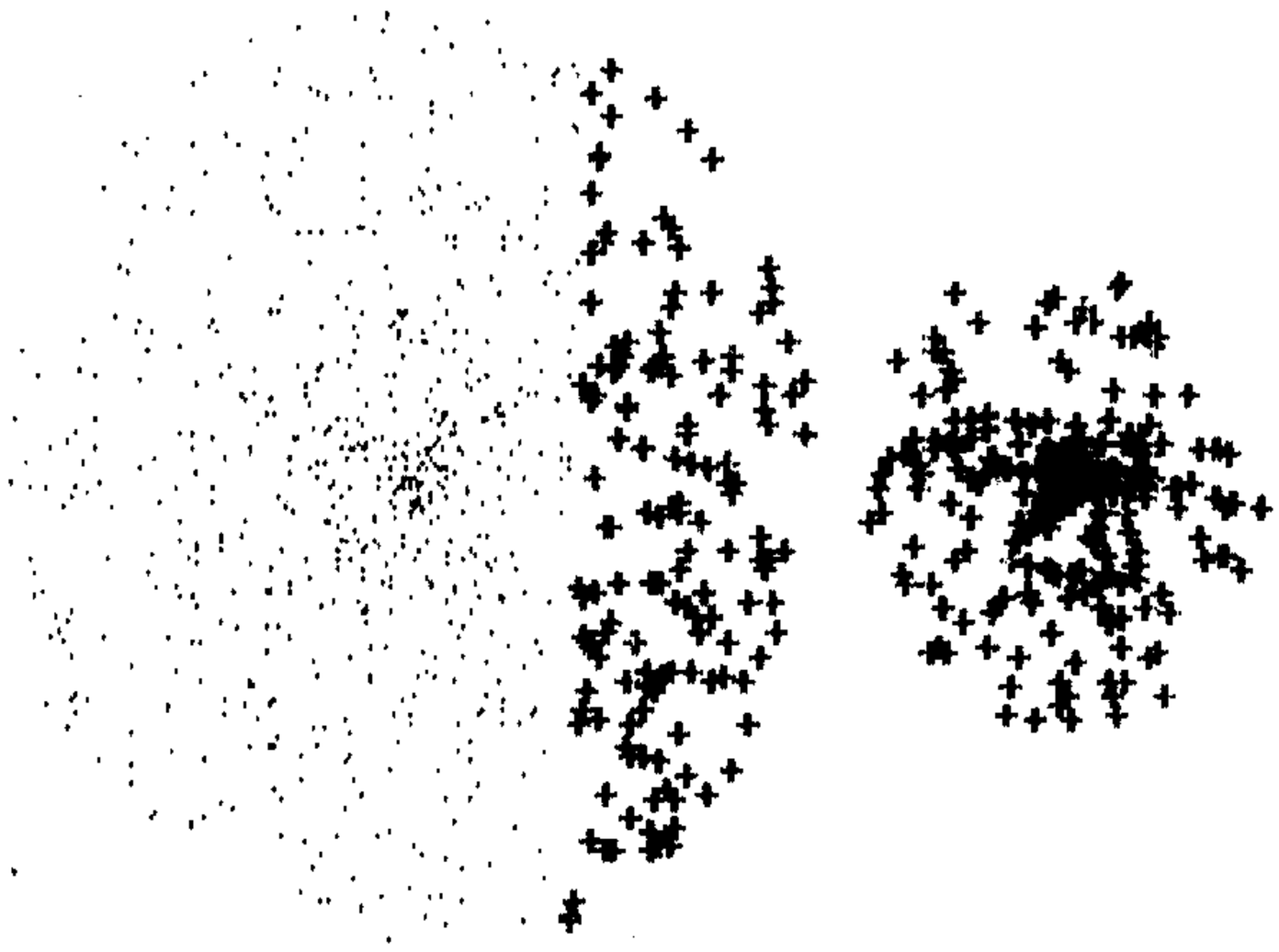


Fig 15: Performance of Arithmetic FALVQ algorithm

interclass distance = 0.2

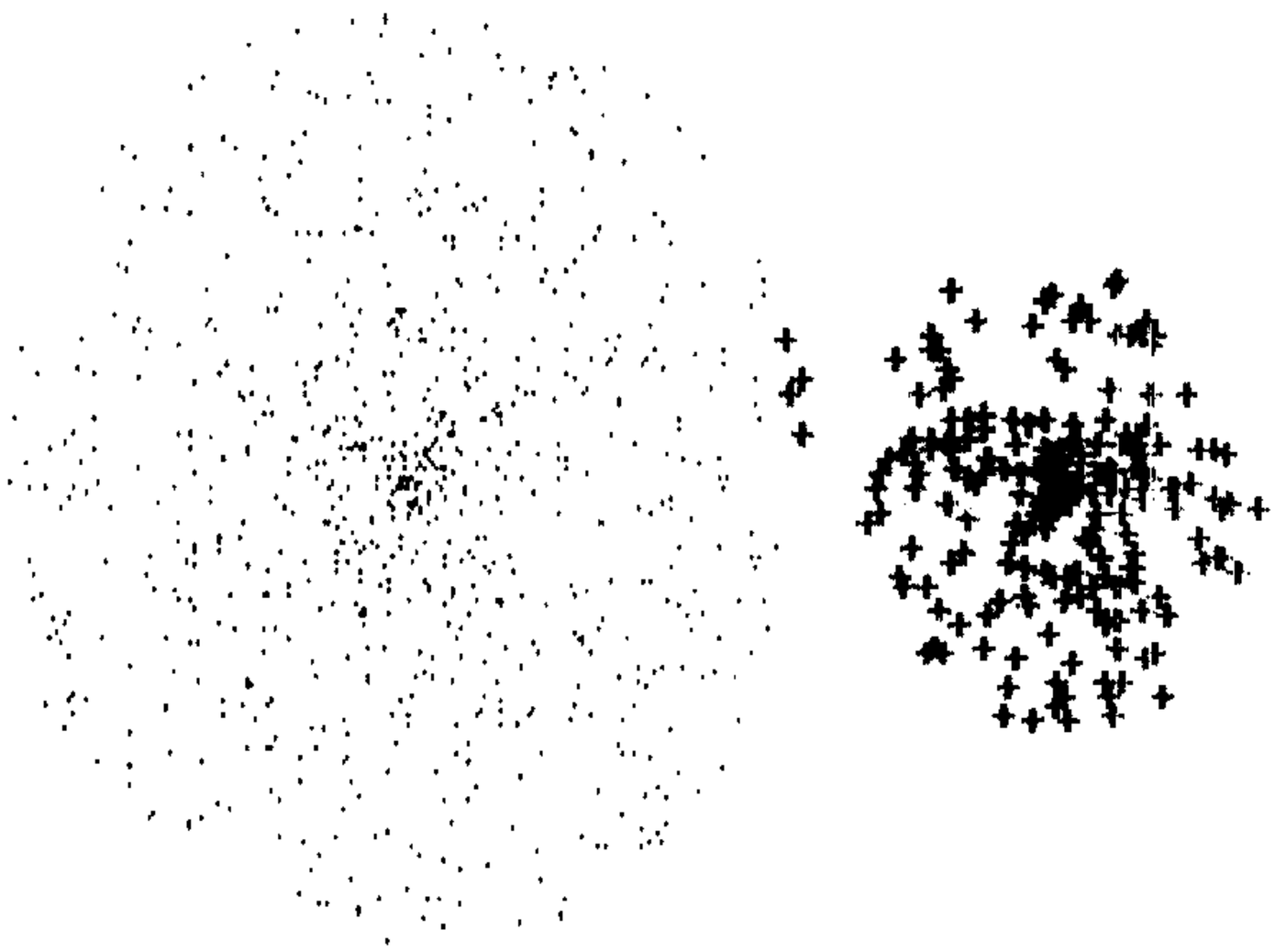


Fig 16: Performance of the proposed algorithm.

interclass distance = 0.2

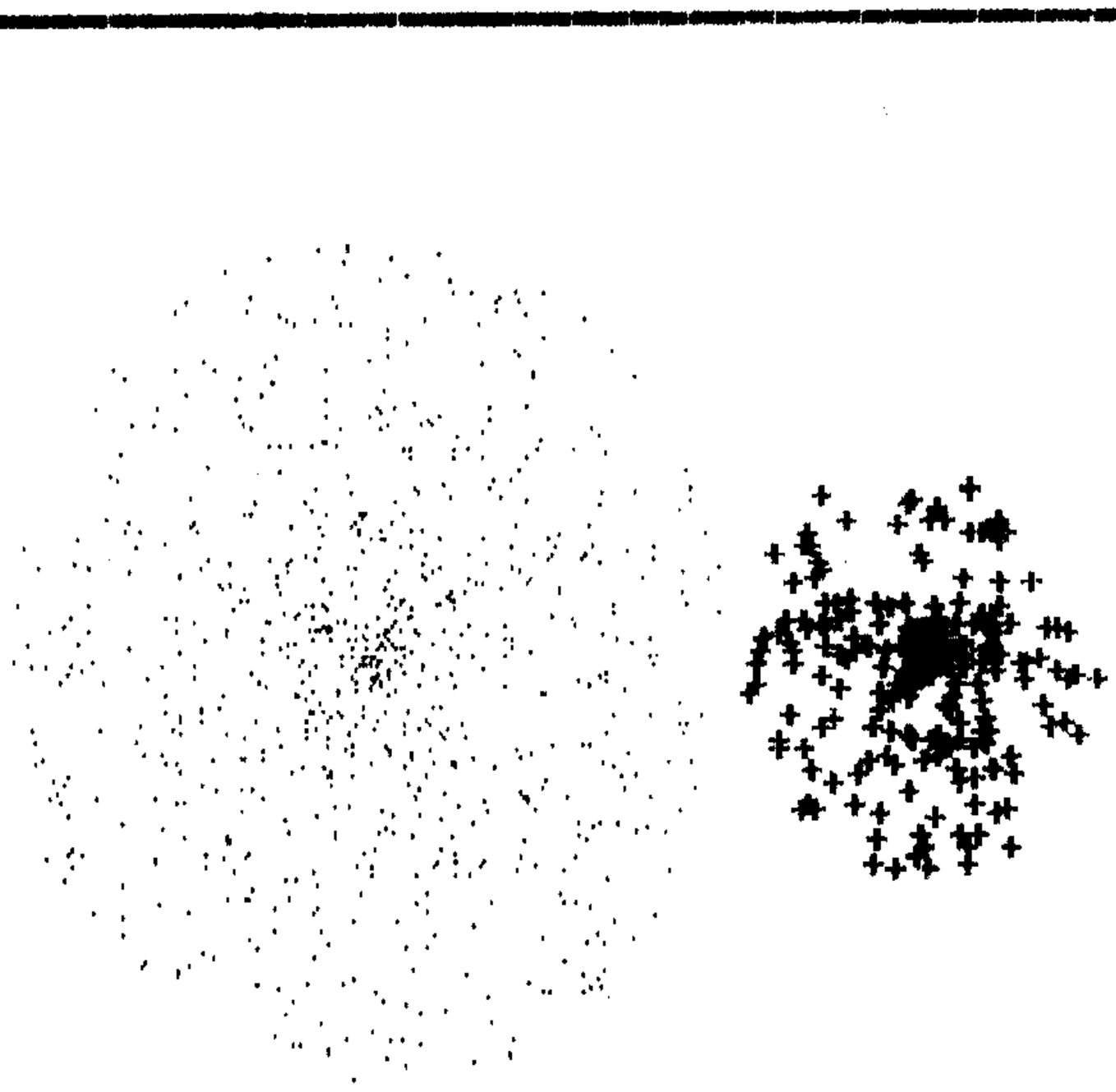


Fig 17:Artificial data set with two classes
of unequal size.

interclass distance = 0.01

Fig 18: Performance of GLVQ algorithm.

interclass distance = 0.01

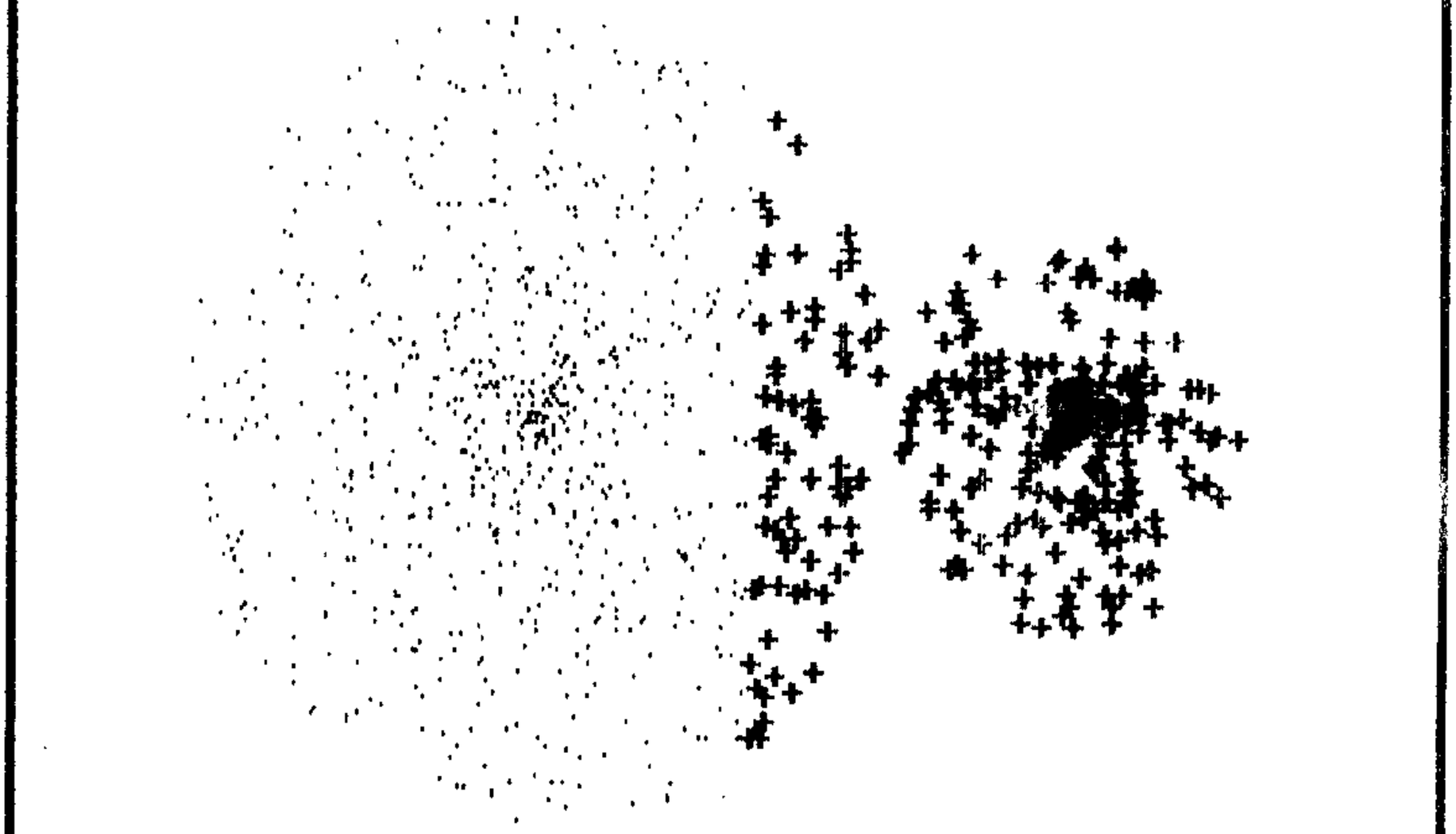


Fig 19: Performance of FALVQ algorithm.

interclass distance = 0.01

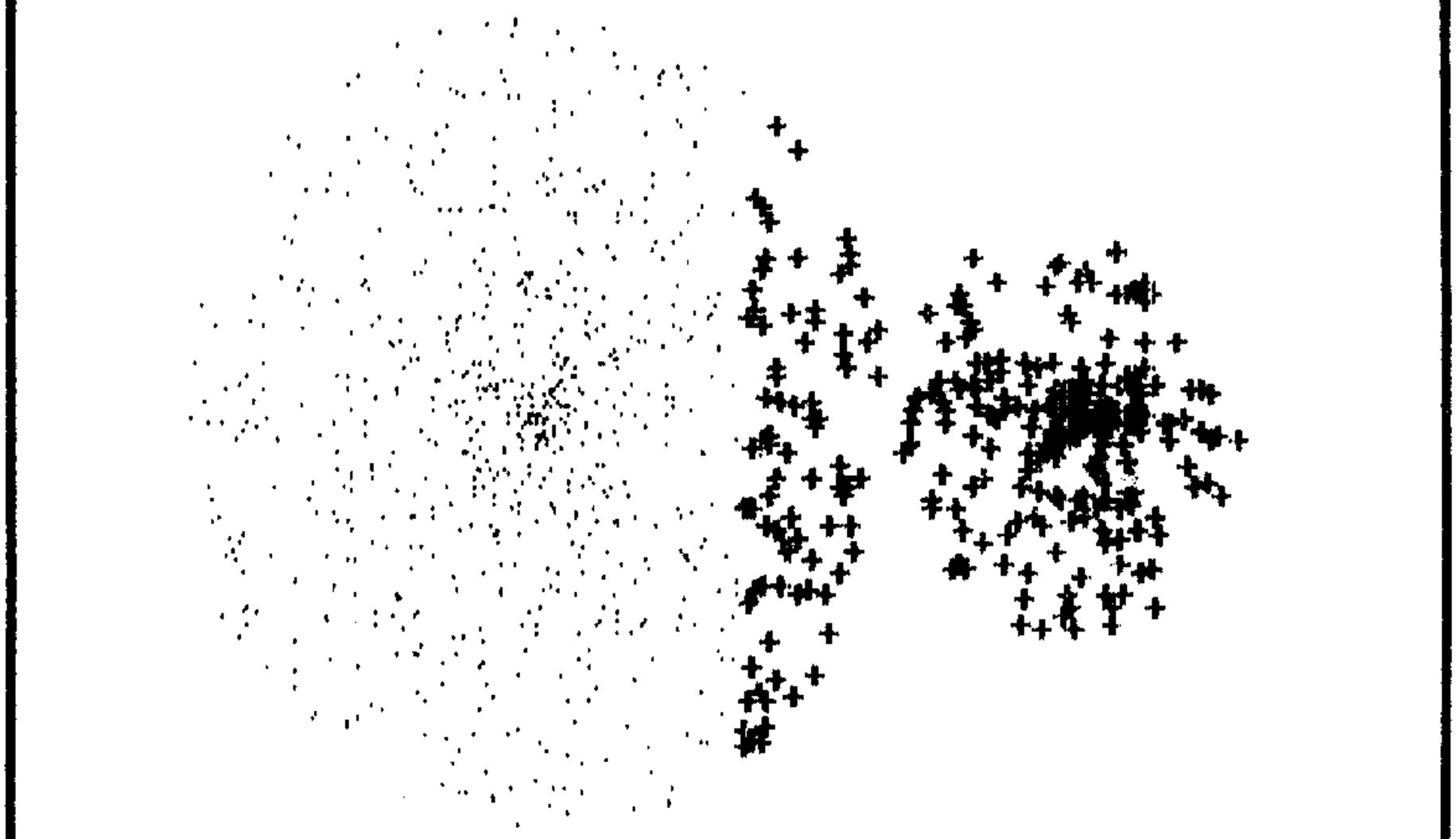


Fig 20: Performance of Harmonic FALVQ algorithm.

interclass distance = 0.01

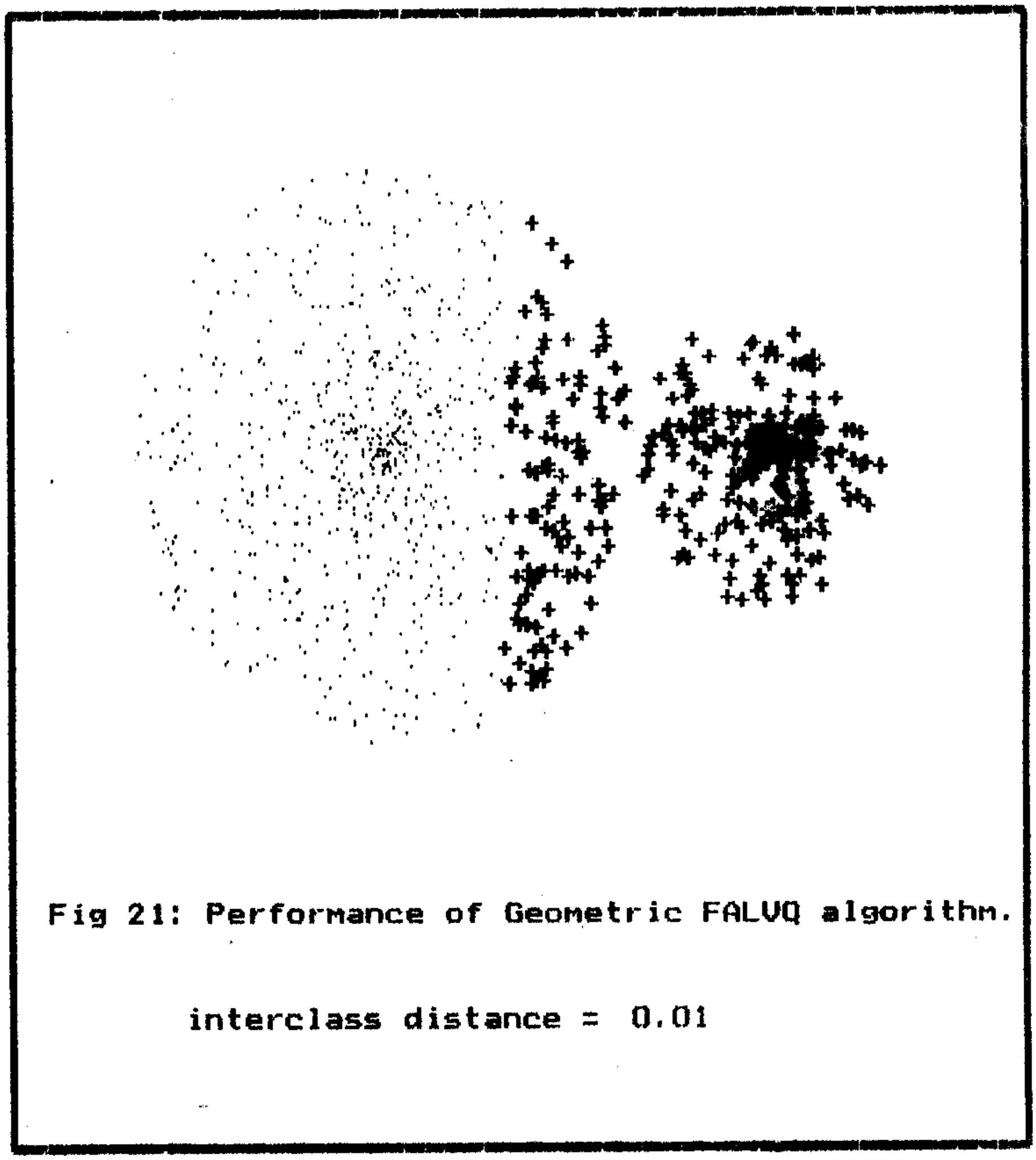


Fig 21: Performance of Geometric FALVQ algorithm.

interclass distance = 0.01

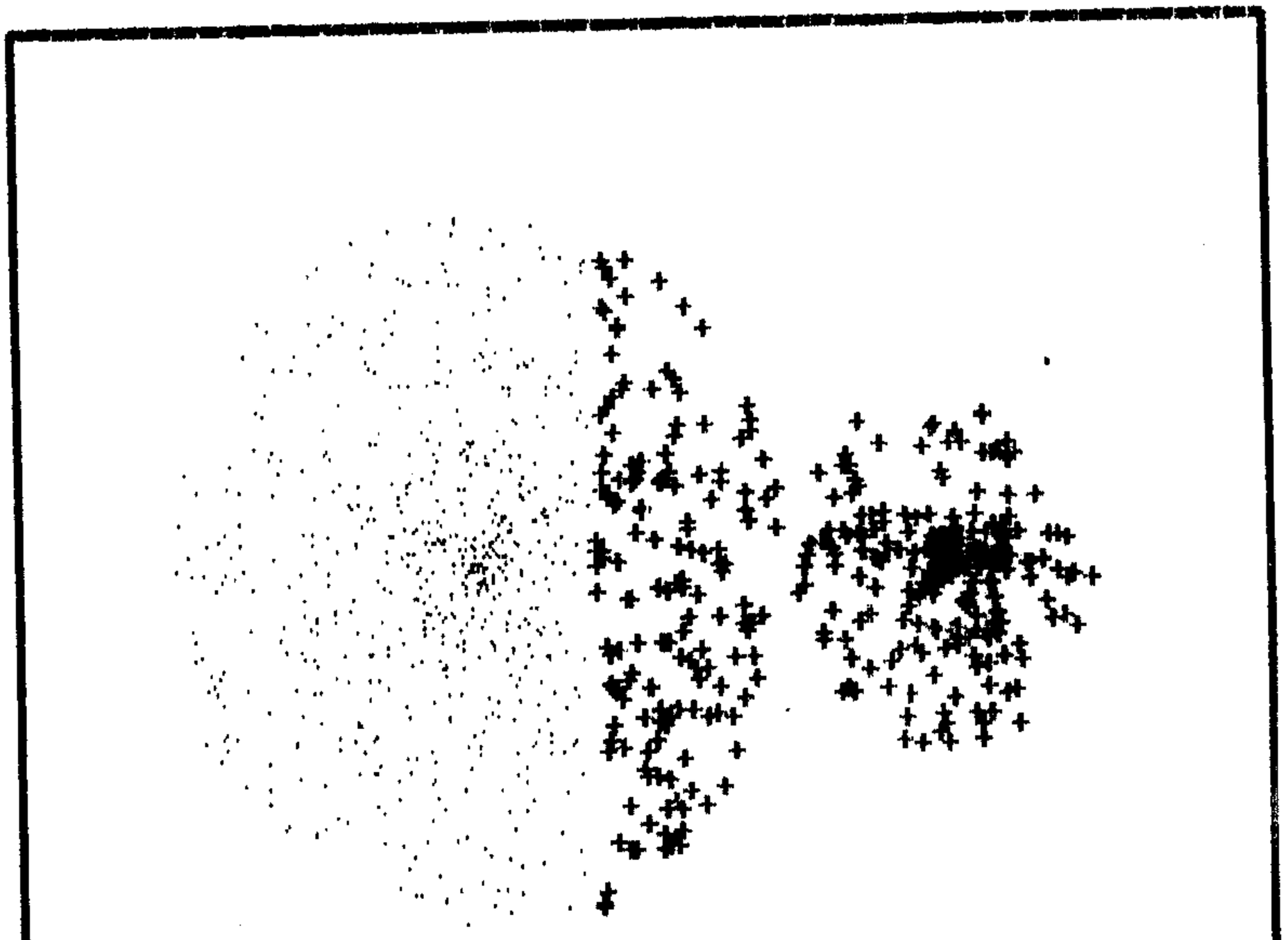


Fig 22:Performance of Arithmetic FALVQ algorithm.

interclass distance = 0.01

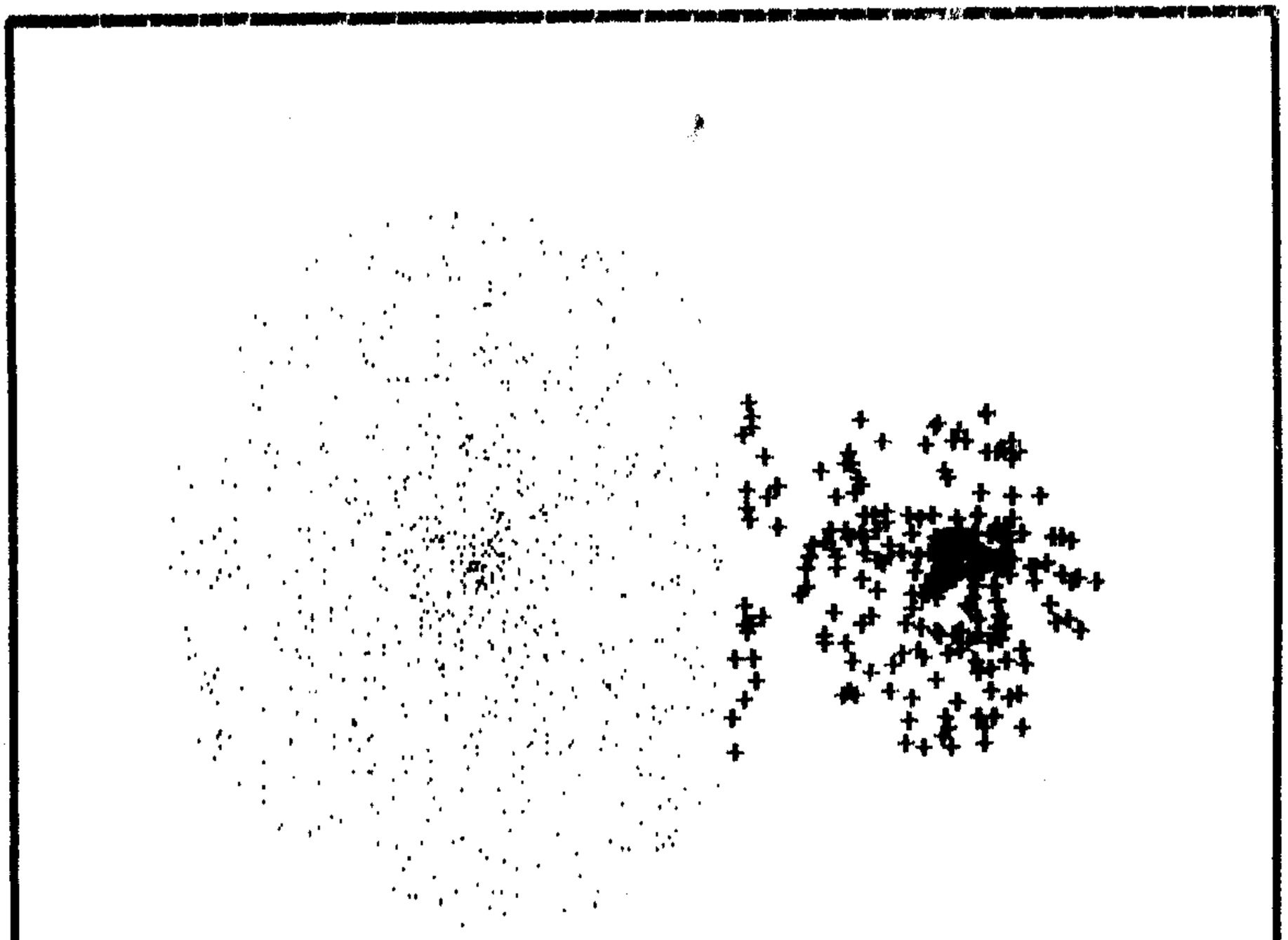


Fig 23: Performance of the proposed algorithm.

interclass distance = 0.01

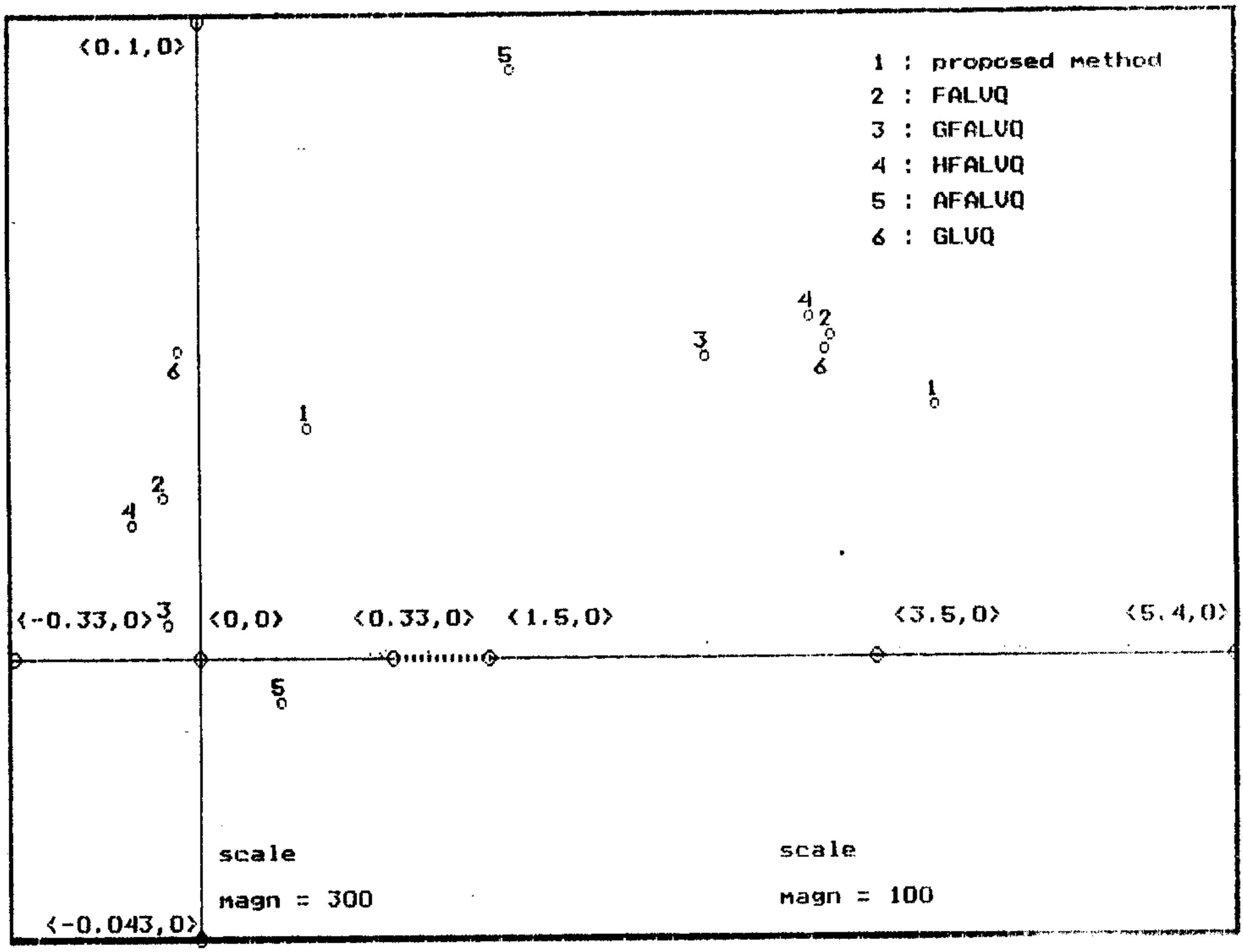


Fig. 24 :
Cluster centers produced for artificial data set by different algorithms

Interclass distance = 0.5

Physical cluster center : class 1 --- (0,0) class 2 --- (3.5,0)

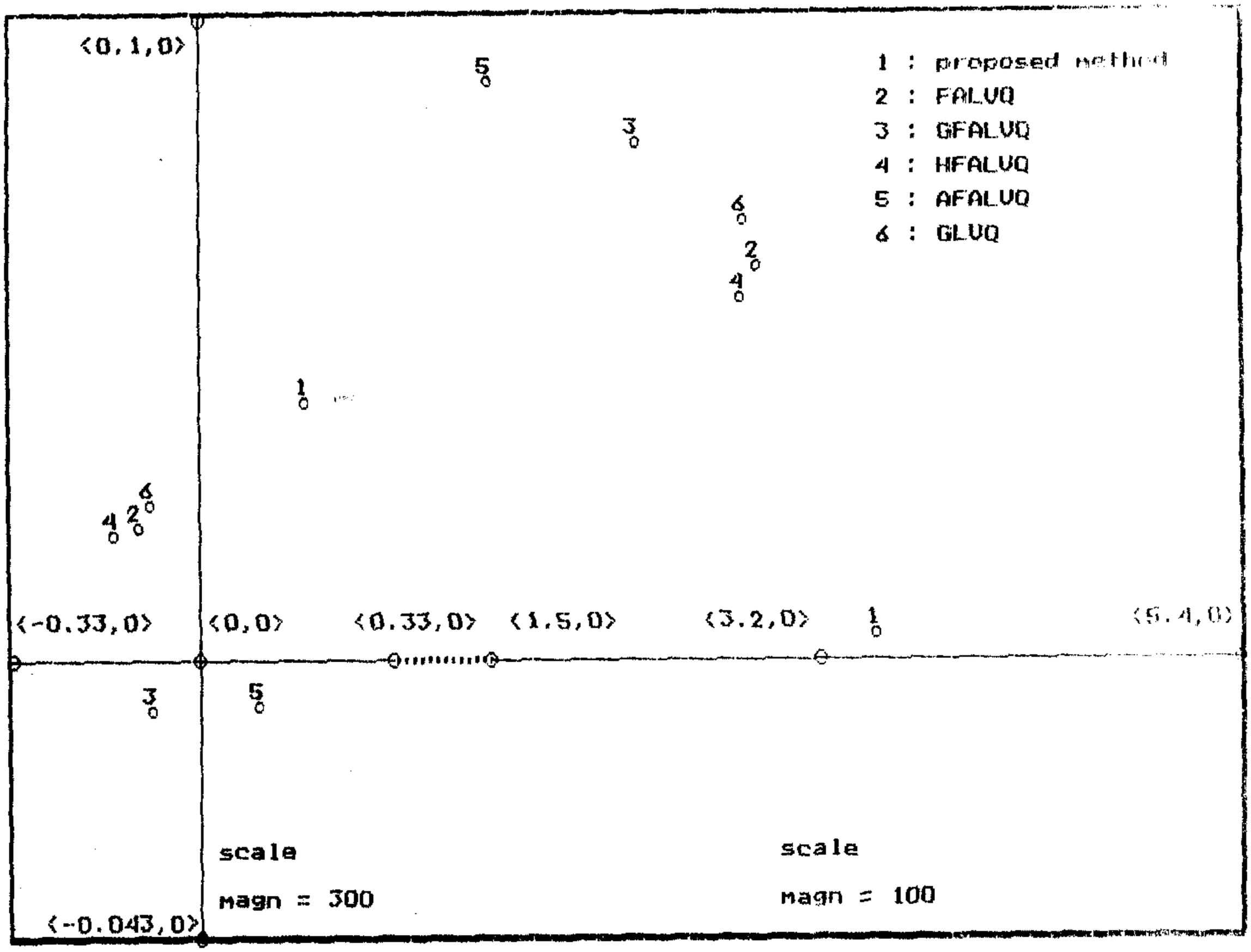


Fig. 25 :
Cluster centers produced for artificial data set by different algorithms

Interclass distance = 0.2

Physical cluster center : class 1 --- <0,0> class 2 --- <3.2,0>

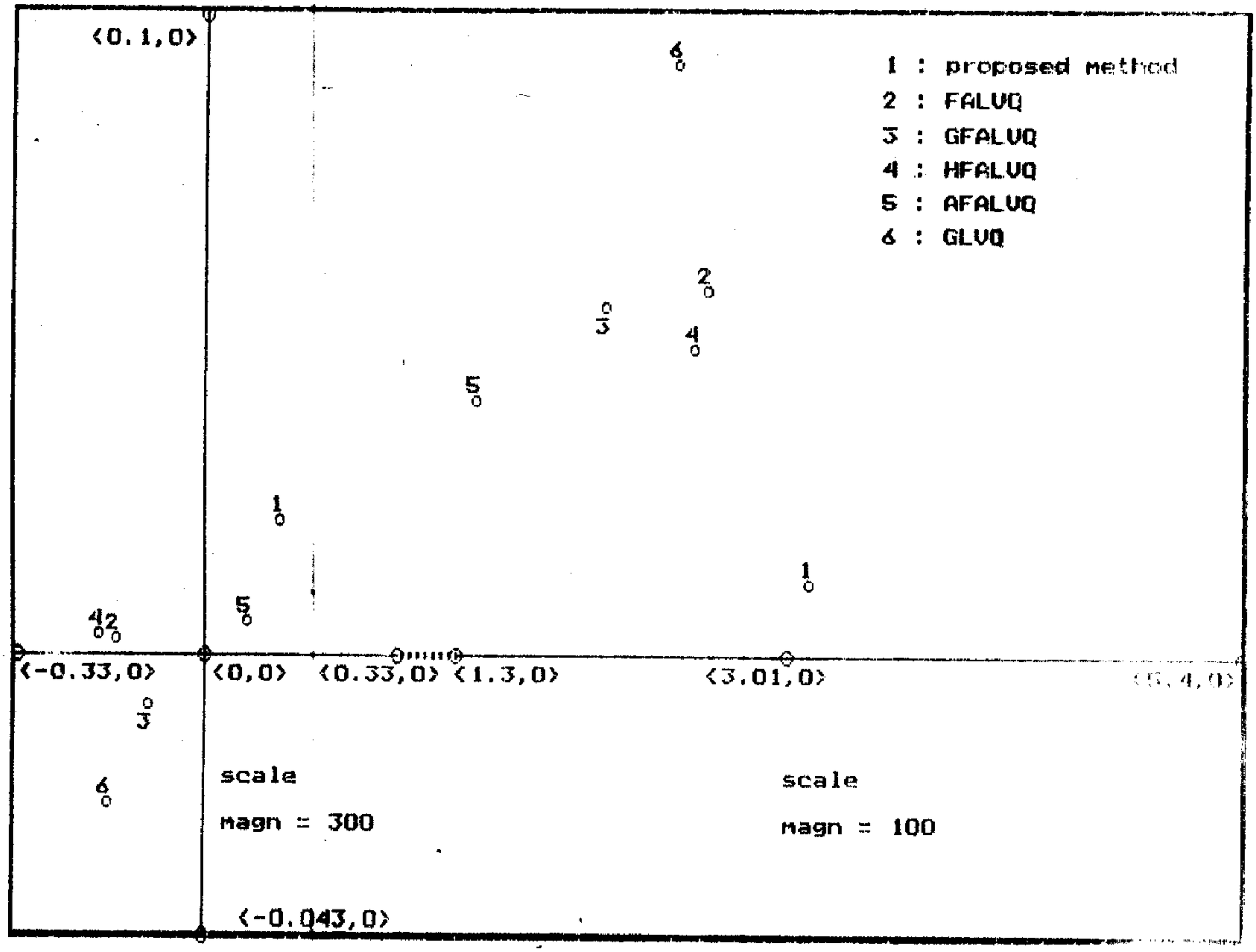


Fig. 26 :
Cluster centers produced for artificial data set by different algorithms

Interclass distance = 0.01

Physical cluster center : class 1 --- ⟨0,0⟩ class 2 --- ⟨3.01,0⟩

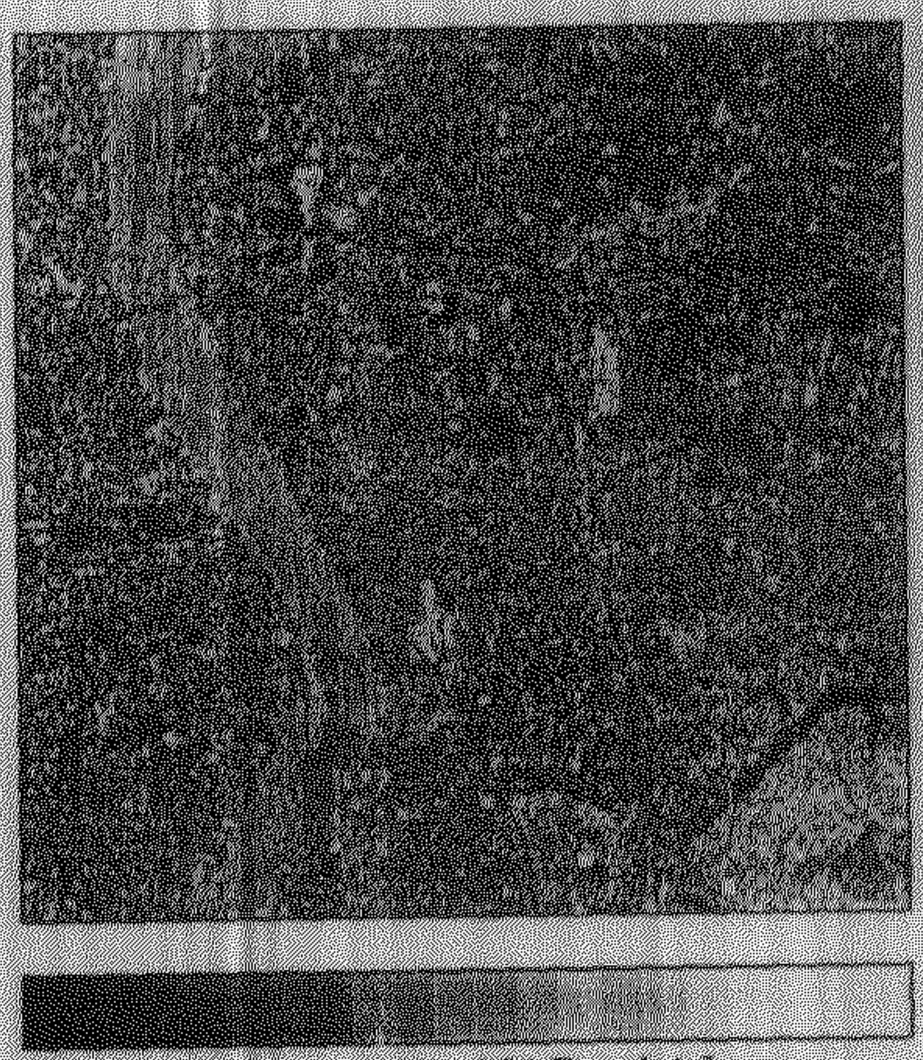


Fig. 27: Original 256 x 256 Band-1 Image

Fig. 28: Original 256 x 256 Band-2 Image

Fig. 29: Original 256 x 256 Band-3 Image

Fig. 30: Original 256 x 256 Band-4 Image

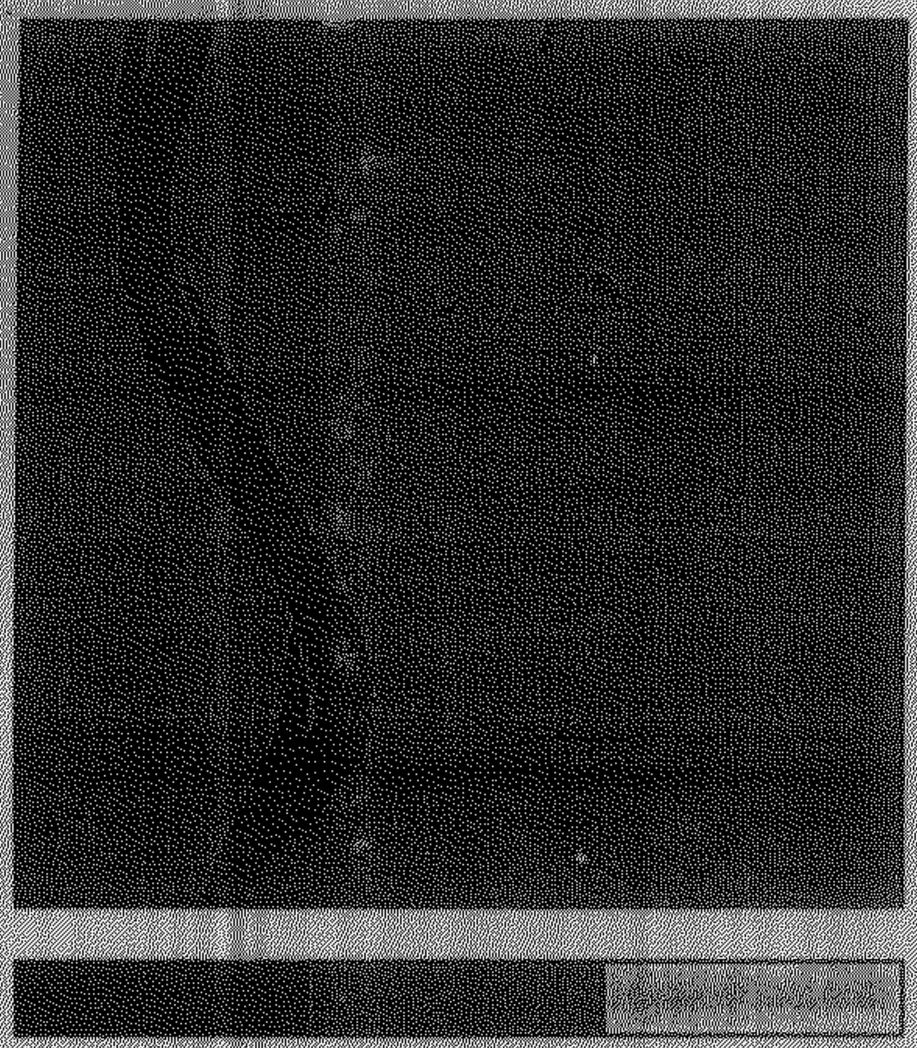


Fig. 31: Reconstructed 256 x 256 image with k=3

Fig. 32: "Water" class obtained with k=3

Fig. 33: "Land" class obtained with k=3

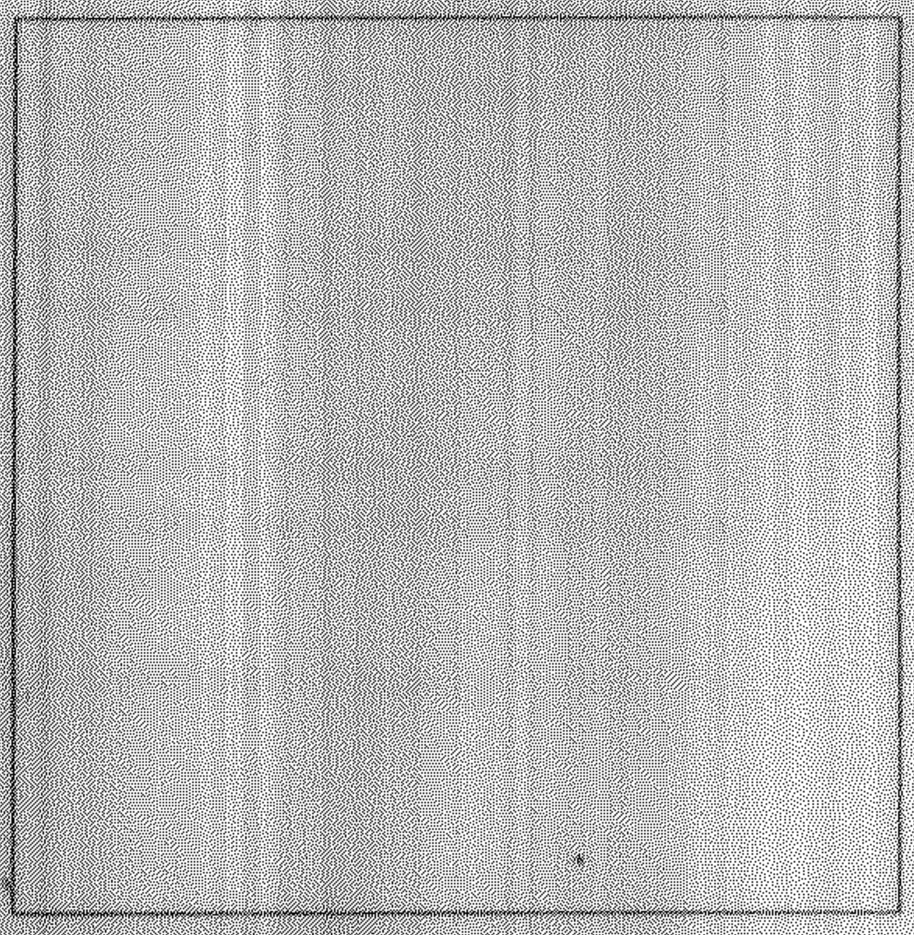


Fig. 34: Noise obtained in the third class with k=3

References

1. E. Anderson, " The IRISes of the Gaspe Peninsula ", Bulletin of American IRIS Society, vol. 59,pp.2-5,1939.

2. N. Chowdhury, C.A. Murthy, S.K. Pal, " Cluster Detection Using Neural Networks ".

3. R.M. Gray, " Vector Quantization ", IEEE ASSP Magazine, vol. 1, no. 2, pp. 4-29, April 1984.

4. T. Kohonen, " An Introduction to Neural Computing ", Neural Networks, vol. 1, pp. 3-16, 1988.

5. T. Kohonen, " The Self-Organization Map ", Proceeding of the IEEE, vol. 78, no. 9, pp. 1464-1480, 1990.

6. N.B. Karayiannis, J. Bezdek, N.R. Pal, R.J. Hathaway, Pin-I Pai, " Repairs to GLVQ : a new family of competitive learning schemes ".

7. N.B. Karayiannis, Pin-I Pai, " A Family of Fuzzy Algorithms for Learning Vector Quantization " , Systems, Neural Nets, and Computing, Technical Report No. 94-07, July 1994.

8. N.B. Karayiannis, Pin-I Pai, " Fuzzy Vector Quantization Algorithms and Their Application in Image Compression ", IEEE Transactions on Image Processing, vol. 4, no. 9, September 1995.

9. N.M. Nasrabadi, Robert A. King, " Image Coding Using Vector Quantization: A Review ", IEEE Transactions on Communications, vol. 36, no. 8, August 1988.

10. T. Kohonen, " Improved Versions of Learning Vector Quantization ", Proceedings of the International Joint Conference on Neural Networks, vol. I, San Diego, pp. 545-550, June, 1990.

11. T. Kohonen, " Self-Organization and Associative Memory ", 3rd Edition, Springer Verlag, Berlin, 1989.