New Learning Strategy for Prototypes in Linear Vector Quantization

Ghiorghe Puşcaşu
Faculty of Automatic Control, Computers, Electrical Engineering and Electronics, "Dunarea de Jos" University of Galati, gheorghe.puscasu@ugal.ro
Bogdan Codreş
Faculty of Automatic Control, Computers, Electrical Engineering and Electronics, "Dunarea de Jos" University of Galati, bogdan.codres@ugal.ro

Alexandru Stancu
School of Electrical and Electronic Engineering, Faculty of Engineering and Physical Sciences, The University of Manchester, Sackville Street Building, E1d, Manchester, M13 9PL, United Kingdom, Alexandru.Stancu@manchester.ac.uk
Alexandru Codreş
School of Electrical and Electronic Engineering, Faculty of Engineering and Physical Sciences, The University of Manchester, Sackville Street Building, E1d, Manchester, M13 9PL, United Kingdom, Eduard.Codres@postgrad.manchester.ac.uk

***Abstract*—This paper proposes a new strategy for prototypes initialization in linear vector quantization algorithm (LVQ). Three principles which must be satisfied during the learning stage are shown in the paper. These principles are essential to guarantee appropriate learning for the LVQ algorithm. However, all versions of LVQ algorithms try to answer to one of the principles, but unfortunately contradicting with the other ones. The new strategy proposed in the paper aims to solve this issue and consists of two steps: (1) analyse the a-priori data set and (2) apply a pre-learning algorithm to initialize the prototypes. The pre-learned prototypes resulted from step 2 are used by the LVQ algorithm in the learning process. The examples presented in the case study and the criterion used to assess the training performance of the prototypes reinforce that the training strategy of the prototypes proposed in the paper provides better results in certain situations compared to classical LVQ algorithms.**

***Keywords*—LVQ algorithms, pattern recognition, learning, Self-Organizing Map.**

I. INTRODUCTION

The LVQ algorithm [1] is a supervised learning algorithm that learns to classify data based on determining the optimal position of the prototypes. LVQ can be included in a broad family of learning algorithms based on Stochastic Gradient Descent [2]. LVQ has applications in different fields, such as: signal processing [3], fault diagnosis [4, 5], feature extraction [6,7], and many others. The LVQ algorithm has been improved during the years by many authors [8 , 9].

The main idea of the LVQ algorithm is to build a quantized approximation for the spatial distribution of the training vectors using a finite number of prototypes. After the learning process is finished, each prototype represents a subset of the data that has the same degree of similarity. The LVQ algorithm requires several factors to be defined before the learning process starts[5], and they include: (1) the objective function, (2) the appropriate distance measure, (3) the optimization algorithm, (4) the optimal number of prototype vectors, and (5) the initial positions of the prototype vectors to avoid suboptimal solutions.

The LVQ algorithm and its three versions LVQ1, LVQ2, and LVQ3 were proposed for the first time by Kohonen in [1]. The LVQ2 and LVQ3 algorithms use the concept of *window* to update the prototypes during the learning process. One of the conditions for updating the prototypes is that the new input data, or pattern, presented to the algorithm, must fall into the *window*. If the new pattern is outside the *window*, the pattern will be ignored and blocked in the process of prototypes learning. This is a method of selecting the patterns which will affect the prototypes during the learning process.

For improving the performance of the LVQ network, a new weight-updating formula for prototypes is presented in [10]. In [11] the authors introduce an explicit cost function where the learning rule attempts to minimize such function. Selecting a subset from the training data set to update LVQ prototypes is introduced in [12], where the proposed method selects an updated set composed by a subset of points considered to be at the risk of being captured by another prototype from a different class.

In the prototype learning process, it is necessary to take into account the following three principles:

1. Specialization of prototypes. After learning, each prototype will get similar characteristics with a subset of patterns belonging to the class represented by the prototype.

2. Every pattern from the training data set must participate in the prototype update (learning process). Otherwise, the characteristics of the non-participating patterns will not be transferred to the prototypes.

3. Minimize as much as possible prototype removal if the pattern and the prototype belong to different classes. In this case the prototype is no longer updated if the patterns from the training set satisfy certain conditions. For instance (for LVQ2 and LVQ3 algorithms), if the pattern is outside the window, it will not be used for prototype learning.

In some situations the third principle is in contradiction with the first and the second principle. All versions of LVQ algorithms try to answer to one of the principles, but unfortunately contradicting with another. For instance, the *window* method used for the LVQ2 and LVQ3 algorithms satisfies, in some situations, the third principle, but, in the same time, does not satisfy the second principle.

However, it is well-known that all LVQ algorithms and variations of the original LVQ-learning algorithm strongly depend on the initialization of the prototypes [1, 3, 13]. The prototypes initialization and the training data set can often create a conflict among the three principles. The learning rules for the prototypes, without a-priori analysis of the training data set, lead to the impossibility of satisfying all three principles, at the same time.

The new lemma and new theorem proposed in this paper will set the conditions that can guarantee a proper prototype allocation when the LVQ1 algorithm is applied. Although these conditions may not be fulfilled for complex data sets, the lemma and the theorem are important for finding instances when prototype training provides unacceptable solutions. Some of these cases are illustrated in the paper and, as a solution for these situations, a new method of prototype initialisation and pre-learning is implemented.

The paper is organized as follows: in Section 2, the prototypes' dynamic during the learning process is analysed, and a new lemma and theorem are introduced to guarantee optimal or acceptable solution for the LVQ algorithm. Based on the theoretical results from Section 2, a new algorithm for LVQ1 is proposed in Section 3, and its advantages are illustrated by several examples with simulated data (academic data set). In Section 4 the paper is concluded and some comments about future works are presented.

II. Ease of Use The prototypes' dynamic during the learning process

This section presents new theoretical results regarding the prototypes' dynamic during the learning process with LVQ1 algorithm. The prototypes' dynamic strongly depends on the structure of the training data set. In most cases, the training data set has an irregular structure where it is impossible to consider all possible cases. However, without loss of generality, some representative situations will be considered in this section in order to illustrate the prototypes' dynamic.

Lemma. *Consider X^1 and X^2 the training sets for two classes which contain n_1 patterns and n_2 patterns respectively. The sets X^1 and X^2 are included in the convex domains DC_1 and DC_2 ($X^1 \subset DC_1$, $X^2 \subset DC_2$). For LVQ1 algorithm, assume that the following hypotheses are true:*

1) *The learning rate is considered to be bounded in the interval* [0 1], *i.e.* $0 < k(t) \leq 1$; (1)

2) $\max\limits_{x,y \in DC_1} \{d(x,y)\} < \min\limits_{\substack{x' \in DC_1 \\ y' \in DC_2}} \{d(x',y')\}$ (2)

$\max\limits_{x,y \in DC_2} \{d(x,y)\} < \min\limits_{\substack{x' \in DC_1 \\ y' \in DC_2}} \{d(x',y')\}$ (3)

3) *The prototypes Z^1 and Z^2 are initialized with patterns of both classes. Then, during the learning process, for the set of prototypes Z^1 and Z^2, the following relations hold:*

$\max\limits_{x \in X^1, y \in Z^1} \{d(x,y)\} < \min\limits_{x' \in X^2, y \in Z^1} \{d(x',y')\}$ or equivalently,

$Z^1 \subset DC_1$ (4)

$\max\limits_{x \in X^2, y \in Z^2} \{d(x,y)\} < \min\limits_{x' \in X^2, y \in Z^1} \{d(x',y')\}$ or equivalently, $Z^2 \subset DC_2$ (5)

Proof. The LVQ1 algorithm is based on relations (1) and (2). Consider the following notations:

$X^1 = \{x_i^1\}, i = \overline{1,n_1}$ are patterns from class 1, $X^2 = \{x_i^2\}, i = \overline{1,n_2}$ are patterns from class 2 and $Z^1 = \{z_i^1\}, i = \overline{1,np_1}$ are prototypes associated to class 1, $Z^2 = \{z_i^2\}, i = \overline{1,np_2}$ are the prototypes associated to class 2. These prototypes will be initialized with patterns from each corresponding class.

The following inclusions result from hypothesis 3:

$$\left\{\{z_i^1\}_{i=\overline{1,np_1}}\right\} \subset DC_1 \quad (6)$$

$$\left\{\{z_i^2\}_{i=\overline{1,np_2}}\right\} \subset DC_2 \quad (7)$$

Considering hypothesis 2 (the inequalities (2) and (3)) and the inclusions (6) and (7) one can conclude that before first use of the LVQ1 algorithm, hypothesis 3 is satisfied, i.e. the inequalities (4) and (5) are satisfied.

Now, assume that the pattern $\{x_i^1(t)\} \in X^1$ is used in the learning process by the LVQ1 algorithm. The inequality (4) is satisfied, so the closest prototype $\{z_r^1\}$ to the pattern $\{x_i^1(t)\}$ of class 1 also belongs to class 1. Consequently, eq. (8) is used for the prototype update:

$$z_r^1(t+1) = z_r^1(t) + k(t) \cdot [x_i^1(t) - z_r^1(t)] \quad (8)$$

According to the lemma hypothesis (1) ($k(t) \in (0,1]$), the update prototype $\{z_r^1(t+1)\}$ belongs to the line segment from $z_r^1(t)$ to $x_i^1(t)$. Because $x_i^1(t)$ and $z_r^1(t)$ belong to the convex domain DC_1 , after the learning process, the prototype associated with class 1 ($z_r^1(t+1)$) will also belong to the convex domain DC_1. In this case, inequality (4) remains true after the learning process.

Similarly, it can be proven that the prototypes of class 2 satisfy ineq. (5) after every stage of the learning process.

Because the pattern $\{x_i^1(t)\}$ was arbitrary chosen, we can conclude that inequalities (4) and (5) are true for each prototype and throughout the entire learning process.

Observation: The new Lemma introduced in this paper proves that the prototypes will satisfy relations (6) and (7) after each learning stage during the entire learning process. Fig. 1 illustrates the case when the patterns and the prototypes of both classes satisfy the lemma hypotheses. In this case the prototypes will never leave the domains in which they belong before the learning process.

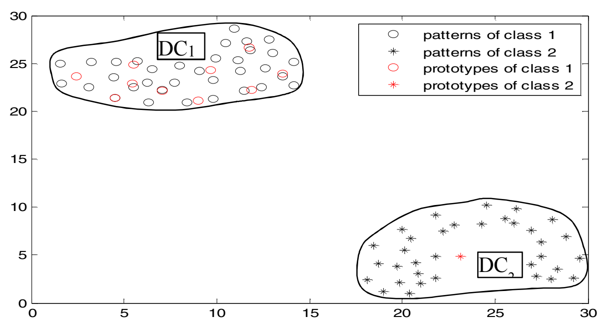

Fig. 1. The situation when the patterns and the prototypes of both classes satisfy the lemma hypotheses
DC_1 – convex domain which includes the sets X^1,Z^1,
DC_2 – convex domain which includes the sets X^2,Z^2,

***Theorem**. Consider X^1 and X^2 the training set vectors for two classes which contain patterns in $\mathbb{R}^n$. The set $X^1 \subset DC1$, where $DC1$ is a convex domain. The set X^2 contains two subsets $X^{21} \subset DC_{21}$, $X^{22} \subset DC_{22}$, with DC_{21} and DC_{22} convex domains.*

Assuming the following hypotheses for the LVQ1 learning algorithm:

***1)** The learning rate $k(t)$ is always positive and is decreasing monotonically from 1, i.e. in the interval (0, 1]:*

$k(t) = a - b \cdot t$, *a=1, b>0* , $t \in (0, \frac{a}{b}]$.

***2)** a.* $$\max_{x,y \in DC_1} \{d(x,y)\} < \min_{\substack{x' \in DC_1 \\ y' \in DC_{21}}} \{d(x',y')\} \quad (9)$$

$$\max_{x,y \in DC_1} \{d(x,y)\} < \min_{\substack{x' \in DC_1 \\ y' \in DC_{22}}} \{d(x',y')\} \quad (10)$$

b. $$\max_{x,y \in DC_{21}} \{d(x,y)\} < \min_{\substack{x' \in DC_{21} \\ y' \in DC_1}} \{d(x',y')\} \quad (11)$$

$$\max_{x,y \in DC_{21}} \{d(x,y)\} < \min_{\substack{x' \in DC_{21} \\ y' \in DC_{22}}} \{d(x',y')\} \quad (12)$$

c. $$\max_{x,y \in DC_{22}} \{d(x,y)\} < \min_{\substack{x' \in DC_{22} \\ y' \in DC_1}} \{d(x',y')\} \quad (13)$$

$$\max_{x,y \in DC_{22}} \{d(x,y)\} < \min_{\substack{x' \in DC_{22} \\ y' \in DC_{21}}} \{d(x',y')\} \quad (14)$$

3) $$\max_{\substack{x \in DC_{22} \\ y \in DC_{21}}} \{d(x,y)\} < \min_{\substack{x' \in DC_{22} \\ y' \in DC_1}} \{d(x',y')\} \quad (15)$$

***4)** The prototypes of class 1 are initialized only with patterns belonging to set $X^1$and the prototypes of class 2 are initialized only with patterns belonging to subset X^{21}. Then, after the learning process, the patterns of the training set which belong to the subset X^{22} will only be represented by one prototype.*

Proof. Fig. 2 illustrates a situation when the theorem hypotheses are fulfilled.

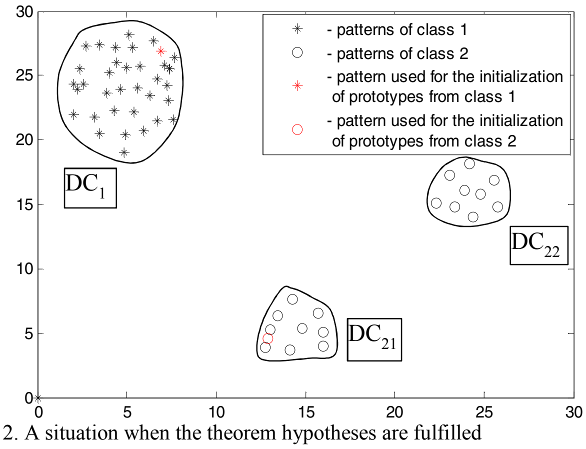

Fig. 2. A situation when the theorem hypotheses are fulfilled

Consider the following notations:

$X^1 = \{x_i^1\}, i = \overline{1, n_1}$ the patterns of the training data set which belong to class 1 and $X^{21} = \{x_i^{21}\}, i = \overline{1, n_{21}}$, $X^{22} = \{x_i^{22}\}, i = \overline{1, n_{22}}$ the patterns of the training data set which belong to class 2.

The prototypes associated with both classes are $Z^1 = \{z_i^1\}, i = \overline{1, np_1}$ and $Z^2 = \{z_i^2\}, i = \overline{1, np_2}$, respectively. The set of initial prototypes under the hypothesis 4 of the theorem satisfy the relations:

$$Z^1 \subset X^1 \quad (16)$$

$$Z^2 \subset X^{21} \quad (17)$$

Considering the theorem hypotheses and relations (16) and (17) the patterns and the prototypes of both classes verify the following relations:

$$\left\{\{x_i^1\}_{i=\overline{1,n_1}} \cup \{z_i^1\}_{i=\overline{1,np_1}}\right\} \subset DC_1 \quad (18)$$

$$\left\{\{x_i^{21}\}_{i=\overline{1,n_{21}}} \cup \{z_i^{21}\}_{i=\overline{1,np_2}}\right\} \subset DC_{21} \quad (19)$$

$$\left\{\{x_i^{22}\}_{i=\overline{1,n_{22}}}\right\} \subset DC_{22} \quad (20)$$

For the first stage of the learning process, the learning rate is $k(t) = 1$ $(t = 0)$ for all training set patterns used for prototypes' updating by the learning algorithm.

Now assume that during the learning process, the pattern $x_i^1 \in X^1$ is used by the learning algorithm to update the prototypes. Taking into account relations (9), (10), (18), and Lemma one can conclude that if the closest prototype to the pattern x_i^1 belongs to class 1 before the learning stage, it will also belong to the same class after the learning stage. The prototype also belongs to the convex domain DC_1.

Consider the special case when the pattern x_1^{22} (the first pattern of the set X^{22}) is used by the algorithm to update the prototypes. Consider hypothesis 3 of the Theorem (i.e. the relation (15)), the closest prototype to the pattern x_1^{22} is one of the prototypes associated with the set X^{21} . We note this prototype with $z_c^*(t)$.

Eq. (21) is used for the prototype update:

$$z_c^*(t+1) = z_c^*(t) + 1 \cdot [x_i^{22} - z_c^*(t)] = x_i^{22} \quad (21)$$

where the learning rate is 1 because this is the first learning stage of the learning process. After that, the subset X^{22} will only contain one associated prototype. By considering relations (13), (14) and lemma, if the algorithm uses any other pattern from the subset X^{22} for the learning process, then the closest prototype to the pattern will also be $z_c^*(t)$. In conclusion, the subset X^{22} will only be represented by one prototype after the learning process: $z_c^*(t) \subset DC_{22}$.

Observation 1: The prototype $z_c^*(t)$ is always updated when a pattern of the subset X^{22} is used by the learning algorithm. After every learning stage, the updated prototype will also belong to the same domain DC_{22}.

A situation when the theorem hypotheses are satisfied is presented in Fig. 3. The initialization of the prototypes for class 2 is randomized by using patterns from the subset 21.

Fig. 3 shows that the subset 22 has only one prototype.

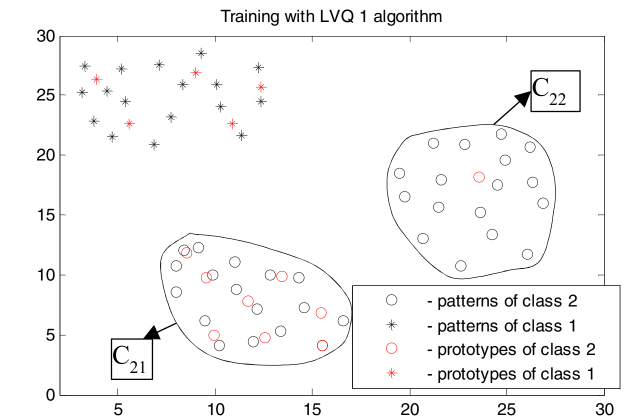

Fig. 3. Academic data set which verifies the theorem hypotheses

Observation 2: If the training data set contains a large number of patterns, randomizing initial prototypes cannot guarantee that all sets of patterns with similar characteristics have allocated prototypes. For instance, in voice recognition applications, it is possible that the same speaker may have different subsets in the training data set.

Observation 3: If hypothesis 3 of the theorem is not satisfied, the patterns of the subclass 22 are closer to the prototypes of class 1 than to the prototypes of the subclass 21. Thus, when the patterns from the subclass 22 are used in the learning process by the LVQ1 algorithm, the prototypes of class 1 are updated with relation (2) and they will be gradually removed from the patterns of the subclass 22. This situation is illustrated in fig. 4.

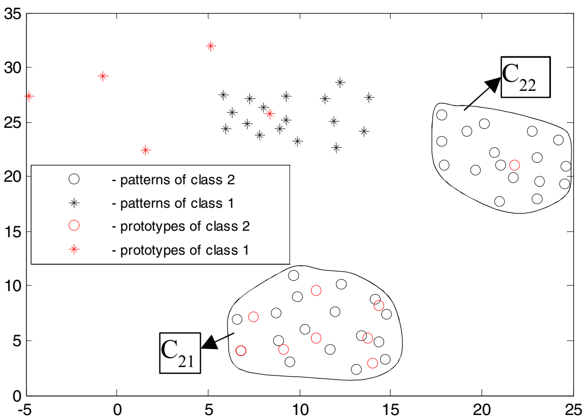

Fig. 4. Academic data set which does not verify hypothesis 3 of the theorem

III. A NEW STRATEGY FOR PROTOTYPE LEARNING USING LVQ1 ALGORITHM. CASE STUDY

As it was shown in section 2, the structure of the training data set plays an important role in the prototype learning stage. The structure of the learning data set can lead to unfavourable situations for the prototypes updating. In order to avoid this situation for cases when the lemma or theorem hypotheses are not fulfilled, a new algorithm is proposed in the following of this section. The main idea of this algorithm is to analyse the classes structure first, and then, to use Self-Organizing Map algorithm [1] for a-priori prototypes updating for each of the subclasses. The proposed algorithm will divide the classes which can create problems in subclasses. Consequently, the prototypes degree of representativeness will significantly increase.

By analysing the structure of the training data set and by initialising the prototypes with patterns which belong to each subclass, it is possible to obtain correct results regarding the prototypes placement in the pattern space. A pre-learning strategy by using a procedure similar with Self-Organizing Map will avoid the instability of the LVQ1 algorithm.

Consider a training data set with $\Omega_1 ... \Omega_n$, n classes, each containing $M_1 ... M_n$ patterns. The new proposed strategy for prototype updating is as follows:

1. The structure of each class is analysed by determining the number of subclasses corresponding to the patterns for all n classes.

2. Pre-learning phase for the prototypes corresponding to each subclass using Self-Organizing Map. For each subclass, the number of prototypes is proportional to the number of patterns which belong to the sub-group, considering the total number of the patterns. By dividing the training data set into smaller subclasses we address the issues presented above which arise when the hypotheses of Theorem are not fulfilled. In this case, by proper allocation of the prototypes, only the conditions of the Lemma need to be satisfied to guarantee the performance of the LVQ1 algorithm. But even for complex training data sets, when neither Theorem nor Lemma conditions are fulfilled, the pre-learning phase can achieve better classification results for the LVQ1 training algorithm.

3. The LVQ1 algorithm is applied to the prototypes initialized with the values obtained in step 2.

The algorithm that implements the new learning strategy is shown in the following page.

Observation:
The patterns clustering of the class Ωi into a number of subclasses (Step 1.1) is determined by either using the Basic ISODATA algorithm [15] or by minimising the sum of the square errors.

After training the prototypes with LVQ type algorithms, a test stage is required to verify that the patterns are correctly classified by the trained prototypes.

For the evaluation of the learning algorithm performance a criterion is proposed that measures the accuracy degree with which the prototypes estimate the pattern repartition in their space.

Suppose that a certain class Ω with $n\Omega$ patterns is represented by np prototypes, and each prototype represents ng_j patterns, j=1, ..., np. The patterns' number is represented by the np prototypes and the following equation is verified

$$\sum_{j=1}^{np} ng_j = n\Omega. \qquad (22)$$

The estimation accuracy of the patterns' repartition in their space is given by the equation:

$$V_C = \sum_{j=1}^{np} \sum_{i=1}^{ngj} \|\omega_i - p_j\| = n\Omega, \qquad (23)$$

where ω_i represents the pattern that is the nearest of the p_j prototype.

If the p_j prototypes occupy a more central position inside the patterns space, then the value of the criterion decreases. The learning strategy proposed in this section is exemplified with an academic data set. Comparative results between the proposed strategy and LVQ type algorithms are presented.

Fig. 5a shows a case when the training data set does not fulfil the Lemma or Theorem hypotheses, and the training was made with the LVQ1 algorithm. The prototypes could be removed from the patterns they were initially assigned with. Fig. 5a shows a situation where there is a subclass with no prototype because of the complex structure of the training data set. However, neither LVQ2 algorithm (fig. 5b), nor LVQ3 algorithm (fig. 5c), leads to satisfactory results since the subclass remains without prototypes. In this example the patterns of class 2 are distributed in space, occupying a non-compact area. In this case some patterns from class 2 are close to class 1 but, in the same time, some are away. This spatial distribution of the patterns, together with relation (2) will contribute to removing some prototypes of class 1 and class 2 as it is shown in fig. 5a. Fig. 5d shows the training result after the new proposed algorithm is used.

Algorithm for the new proposed strategy

1. The index number of the class is initialized: i=1.

1.1 The patterns from the class Ω_i are divided in an adequate number of subclasses. Consider that for the Ω_i class, which has M_i patterns, we obtain g_i subclasses noted with $\{SC_i^k\}_{k=\overline{1,g_i}}$, each subclass having M_i^k patterns.

1.2. The index number of the subclass is initialized: k=1.

1.2.1 For each subclass SC_i^k we attach prototypes $\{Z_q^k\}_{q=\overline{1,np_i^k}}$ a-priori established, taking into account the appearance probability of the subclasses' patterns in the global training data set. The prototypes initialisation is made with patterns belonging to the subclass SC_i^k , randomly selected.

1.2.2 The iteration number N for the pre-learning stage is set.

1.2.2.1 the pre-learning step t is initialized: t=1;

1.2.2.2 the learning rate is updated: $k(t) = 1 - t/N$

1.2.2.3 the index number of patterns from subclass SC_i^k is initialized: $r = 1$;

1.2.2.4 the pattern $x_r \in SC_i^k$ is presented to the algorithm;

1.2.2.5 the prototype z_c which verifies the following relation is found:

$$\min_{q=\overline{1,np_i^k}}\{\|x_r - z_q\|\} = \|x_r - z_c\|$$

and then the prototype is updated with the relation

$$z_c(t+1) = z_c(t) + k(t)\cdot[x_r(t) - z_c(t)];$$

1.2.2.6 $r = r + 1$;

1.2.2.7 if $r \le M_i^k$, jump at 1.2.2.4{we move to the next pattern from subclass SC_i^k}, if not – carry on;

1.2.2.8 $t = t + 1$;

1.2.2.9 if $t < N$, jump at 1.2.2.2{we move to the next pre-learning stage }, if not – carry on.

1.2.3 $k = k + 1$.

1.2.4 If $k \le ng_i$, jump at 1.2.1 {we move to the next subclass }, if not – carry on.

1.3 $i = i + 1$.

1.4 If $i \le n$, jump to 1.1{we move to the next class }, if not – carry on.

2. The classes' prototypes are initialized with the subclasses' prototypes obtained at the previous step.

3. We apply the classical Kohonen algorithm, using the prototypes initialized at the previous step.

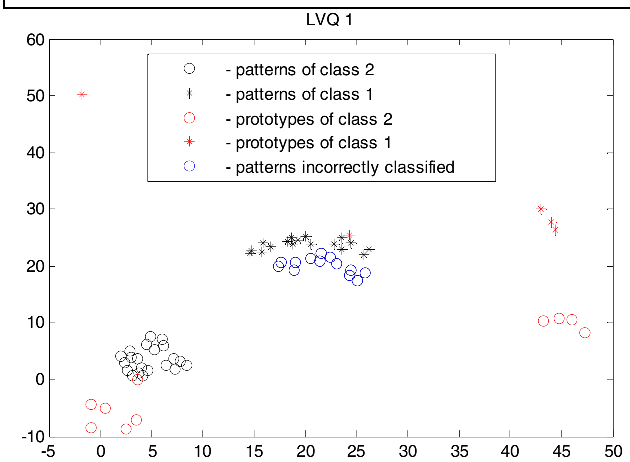

Fig. 5a. Classification results using LVQ1 algorithm

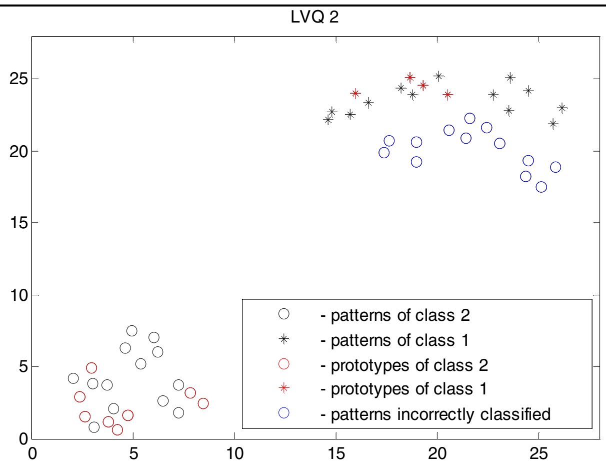

Fig. 5b. Classification results using LVQ2 algorithm

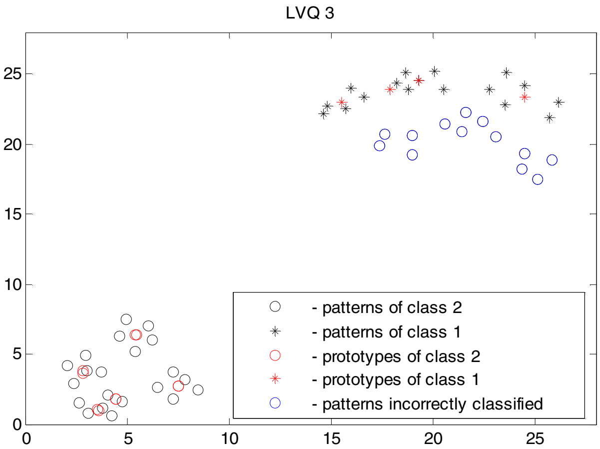

Fig. 5c. Classification results using LVQ3 algorithm

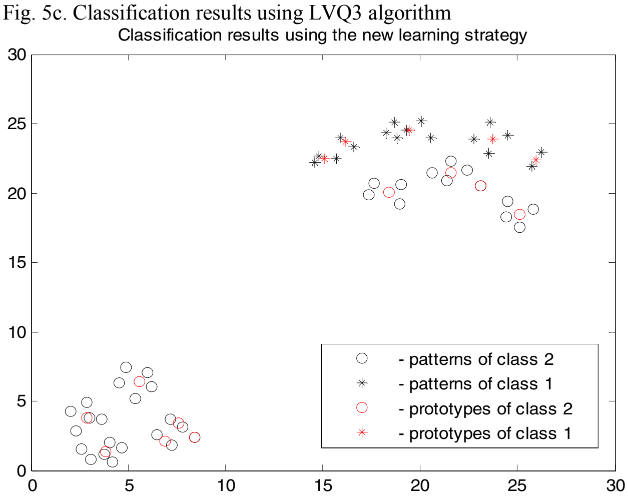

Fig. 5d. Classification results using the new learning strategy

Analytical evaluation of the prototype training with algorithms: LVQ1, LVQ2, LVQ3 and the proposed strategy is shown in the table 1.

It is necessary to mention that the proposed criterion is useful in case the patterns belong to R^q ($q > 3$), in which situation a visual analysis cannot be performed.

TABLE 1. ANALYTICAL EVALUATION OF THE PROTOTYPE TRAINING WITH LVQ ALGORITHMS (ACADEMIC DATA SET)

Algorithm	**Classification 100 %**	**Number of patterns incorrectly classified**	**Value of the criterion**
LVQ1	no	13	262.8358
LVQ2	no	13	116.7772
LVQ3	no	13	86.4451
Proposed Algorithm	yes	0	39.4040

In step 1, using the BASIC Isodata algorithm, the class 2 is divided in two subclasses, subclass 2.1 and subclass 2.2 as it is shown in fig. 5d. For subclass 2.2 and class 1 the hypotheses of theorem 1 are not fulfilled. Because the patterns of subclass 2.1 are grouped, in this case we will not have the problems shown in fig. 5a, when the patterns of the class 2 were distributed in space, occupying a large area. In step 2 the prototypes were allocated in the pattern space. In step 3, which in fact is LVQ1 algorithm, the prototypes for subclass 2.2 are updated because the theorem's hypotheses are fulfilled. The prototypes for class 1 and subclass 2.1 are updated using classical LVQ1 updating rules [1] in order to avoid the situation when a prototype that belongs to a class becomes closer to the patterns from another class. In practical applications, the patterns have multimodal distribution [13,14] and in this case it is necessary to analyse the training data set prior to the learning process. It should be mentioned that in the previous examples, when the prototypes were removed from their corresponding class, the prototype learning process principles proposed in section 1 were not fulfilled. In this case we need to address the third principle in order to improve prototype learning using the LVQ1 algorithm.

IV. CONCLUSIONS AND FURTHER RESEARCH

The paper presents an efficient algorithm for prototypes learning with LVQ algorithm when the patterns have a multimodal and irregular distribution. Three principles, to be satisfied during the learning process, were proposed. To support these principles a new lemma and a new theorem were proposed in the paper. The lemma and the theorem prove that the data set structure and the prototypes initialization play a crucial role for obtaining acceptable results. In this context, Section 3 proposes a new algorithm for prototypes updating. With the new algorithm, in step 1 the learning data set structure is analysed. In step 2 the Self-Organizing Map algorithm for prototypes (a pre-learning stage for each of the subclasses resulted in step 1) is used. A comparison between the new strategy and the classic LVQ algorithms was illustrated in Sections 3 and 4, highlighting that using the proposed algorithm unexpected situations during LVQ training are avoided.

Also in Section 3 we proposed a criterion to evaluate the training stage by computing the estimation accuracy of the pattern repartition in their space.

Future research will focus on two directions: one direction will be to analyse the class' structure in order to establish an appropriate number of subclasses, in which the class may be divided; another direction will be the evaluation of both the learning phase and the learning results.

REFERENCES

[1] T. Kohonen, The Self-organizing Map, *Proceedings of the IEEE*, **78**(9) (1990), pp. 1464-1480.

[2] L. Bottou, Stochastic Learning, Advanced Lectures on Machine Learning, eds. Olivier Bousquet and Ulrike von Luxburg, *Lect. Notes in Artif. Intellig.*, LNAI 3176 (Springer Verlag, Berlin, 2004), pp. 146–168.

[3] M. Blachnika and W. Duch, LVQ algorithm with instance weighting for generation of prototype-based rules, *Neural Networks*, **24**(8) (2011), pp. 824–830.

[4] J. Liu, K. Zheng, H. Zhang and D. Penga, Comparative Research on Power Transformer Fault Diagnosis Based on Several Artificial Neural Networks, *J. of Comput. Inform. Systems*, **9**(18) (2013), pp. 7501–7508.

[5] T. Xu, Research on sensor fault diagnosis method based LVQ neural network and clustering analysis, *7th World Congress on Intelligent Control and Automation*, (Chongqing, 2008), pp. 6017 – 6020.

[6] L. Fan, Q. Wu, C. Ruan and Z. Zhuo, A feature extraction algorithm based on 2D complexity of gabor wavelets transform for facial expression recognition, *5th International Congress on Image and Signal Processing* (Chongqing, Sichuan, China, 2012), pp. 392 – 396.

[7] S. Kutluk and B. Gunsel, Tissue density classification in mammographic images using local features, *21st. Signal Processing and Commun. Applic. Conf.* (Haspolat, 2013), pp. 1 – 4.

[8] F.-M. Schleif, Th. Villmann, B. Hammer and P. Schneider, Efficient kernelized prototype based classification, *Int. J. of Neural Systems*, 21(6) (2011), pp. 443-457.

[9] D. Nova, P. A. Estévez, A review of learning vector quantization classifiers, *Neural Computing and Applications*, **25**(3-4) (2014), pp. 511-524.

[10] M.T. Vakil-Baghmisheh and N. Pavesic, Premature Clustering Phenomenon and New Training Algorithms for LVQ, *Pattern Recognition*, **36**(8) (2003), pp. 1901-1912.

[11] A. S. Sato and K. Yamada, Generalized Learning Vector Quantization, *Advances in Neural Information Processing Systems*, eds. G. Tesauro, D. Touretzky and T. Leon, **7** (1995), pp. 423-429.

[12] C. E. Pedreira, Learning Vector Quantization with Training Data Selection, *IEEE Trans. Pattern Analysis and Machine Intellig.*, **28**(1) (2006), pp. 157-162.

[13] B. Hammer, M. Strickert and T. Villmann, Learning Vector Quantization for Multimodal Data, in *Lecture Notes in Comp. Science*, Vol. 2415 (Springer Berlin Heidelberg, 2002), pp. 370-376.

[14] H. Iwamida, S. Katagiri and E. McDermott, A hybrid speech recognition system using HMMs with an LVQ-trained codebook, *Int. Conf. on Acoustics, Speech, and Signal Processing*, (1990), vol. 1, pp. 489 – 492.

[15] N. Memarsadeghi, D. M. Mount, N. S. Netanyahu and J. Le Moigne, A fast implementation of the ISODATA clustering algorithm, *Int. J. of Computational Geometry and Applications*, **17**(1) (2007), pp. 71-103.