2010 International Symposium on Nonlinear Theory and its Applications
NOLTA2010, Krakow, Poland, September 5-8, 2010

A Classification System based on Collaboration of Adaptive Resonance Theory Maps and Learning Vector Quantization

Yoko Enosawa†, Haruna Matsushita†, and Toshimichi Saito†

†EE Dept., Hosei Univ., Tokyo 184-8584, Japan
yoko.enosawa.w4@gs-eng.hosei.ac.jp, {haruna.matsushita.pd, tsaito}@k.hosei.ac.jp

Abstract—This paper studies a novel classification system with unsupervised learning. First, the adaptive resonance theory map is used to make categories for input sate. After that the learning vector quantization decides the category borders. In elementary classification problems, algorithm works better as the problem complexity increases.

1. Introduction

Learning vector quantization (LVQ) is a simple and universal classification algorithm [1] [2]. The LVQ relates deeply to self-organizing maps: depending of problem complexity, the LVQ can realize flexible classification function that is impossible linear algorithm such as regression analysis and is suitable for many applications. However, the LVQ is a supervised learning algorithm and can not work for data set without category information of each teacher signal. If we can add labeling function to the LVQ, the performance and flexibility of the LVQ can be improved further.

This paper considers collaboration of adaptive resonance theory map (ART) and LVQ , where the ART plays to make categories to help the LVQ. We add a little improvement to the ART, and, after the ART subroutine, the LVQ tries to improve the classification and decides the category borders in the feature space. The ART-LVQ can operate as classification system with unsupervised learning. As is well known, the ART is flexible unsupervised learning algorithm with many applications [3]-[5]. In our previous works, the ART has been used effectively to classify input space for parallel processing and have contributed to improve performance of Self-Organizing Maps (SOM) and ant colony optimizers (ACO) [6]-[8].

In order to consider the algorithm performance, we apply the ART-LVQ algorithm to elementary classification problems. The basic numerical results suggest that (1) the ART-alone is sufficient for simple classification problems and (2) classification function of ART-LVQ becomes better as the problem complexity increases. The ART-LVQ is novel and can be developed into efficient unsupervised learning system for classification problems. Also the ART-LVQ may help parallel processing of many algorithms including SOM and ACO.

2. Algorithm

Our classification system consists of an improved version of adaptive resonance theory map (IART) and LVQ. As a set of input data is given, the IART subroutine makes category information and the LVQ subroutine makes borders of the input space.

2.1. IART for labeling

Let the input data consist of N pieces of 2-dimensional points (x_i, y_i), $i = 1 \sim N$. In the IART, the i-th category at discrete time t is characterized by a circle at center $(x_i,\ y_i)$ with radius r_i:

$$\boldsymbol{W}_i(t) = (x_i(t), y_i(t), r_i(t)),\ i = 1 \sim N_c(t)$$

where $N_c(t)$ is the number of categories at time t. The IART subroutine is defined in the following 6 steps.

Step 1 (Initialization): Let $t = 0$, $N_c(t) = 1$, $r_i(t) = 0$ and let $(x_i(t),\ y_i(t))$ be selected randomly from some two-dimensional distribution $P(x, y)$.

Step 2 (Selection): An input (X, Y) is selected randomly from $P(X, Y)$. If the input belongs to some category then goto Step 5, otherwise we find the closest category to the input.

$$\begin{aligned} &\boldsymbol{W}_c(t) = (x_c(t), y_c(t), r_c(t)) \\ &T_c = \min_i(T_i(X, Y)),\ T_i(X, Y) \\ &\quad = \sqrt{(X - x_i(t))^2 + (Y - y_i(t))^2} - K \times r_i(t) \\ &T_c \equiv \sqrt{(X - x_c(t))^2 + (Y - y_c(t))^2} - K \times r_c(t) \end{aligned} \quad (1)$$

where T_i is the choice function of the i-th category. $-1 \leq K \leq 1$ is the distance parameter that is a key to control the number of categories as suggested in Fig. 1. If $T_c > \gamma$ then goto Step 3 where $\gamma \in [0, 1]$ is the vigilance parameter. If $T_c \leq \gamma$ then goto Step 4.

Step 3 (Birth of a new category): A new category is born at position of the input $(X, Y) \equiv P_j$ as shown in Fig. 2(a) and $N_c(t) = N_c(t) + 1$. The radius of the new category is zero and the suffix $N_c(t)$ is assigned to the new category.

$$\boldsymbol{W}N_c(t) = (x_{N_c(t)}, y_{N_c(t)}, 0) \quad (2)$$

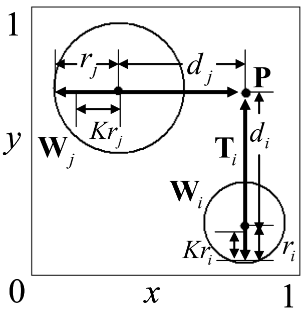

Figure 1: Choice function.

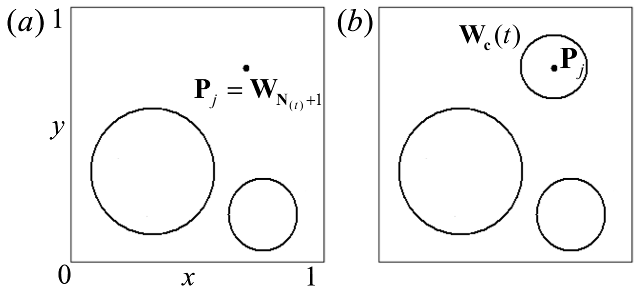

Figure 2: Learning algorithm. (a) Birth of a new category. (b) Category enlargement.

Step 4 (Category enlargement): The selected category $W_c(t)$ is enlarged such that the input (X, Y) is included in the border as shown in Fig. 2(b) and goto Step 5.

Step 5 (Iteration update): Let $t = t + 1$. Goto Step 2 and repeat until the time $t = N$ where N is the number of inputs. At time $t = N$, the $N_c(t) \equiv N_c$ is declared as the number of categories.

Step 6 (Clean-up): If plural categories are overlapped then the overlapped region is divided by lines through the intersections. Using this division line, we can recognize the data category easily. This step does not exist in our previous work [8].

2.2. LVQ for border decision

In the LVQ, a weight vector $\boldsymbol{w}_i$ corresponds to a category and is updated depending on input data. Let $\boldsymbol{w}_i(t) = (x_i, y_i)$ be the weight vector at position (x_i, y_i) at discrete time t and let N_c be the number of categories: $i = 1 \sim N_c$. The LVQ is defined by the following 5 steps.

Step 1 (Initialization): Let the center (x_i, y_i) of the i-th category $\boldsymbol{W}_i(t)$ of ART be changed into the position of the i-th weight vector $\boldsymbol{w}_i(t) \equiv (x_i, y_i)$. Let $t = 0$.

Step 2 (Input): An input $(X, Y) \equiv P_t$ is applied. We select

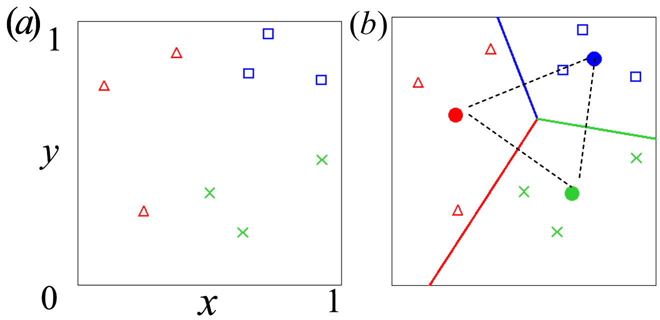

Figure 3: Learning algorithm. (a) input. (b) Decition border.

the category $\boldsymbol{w}_c(t)$ that is the closest to the input:

$$\boldsymbol{w}_c(t) = \min_i(\boldsymbol{w}_i(t) - P_t). \quad (3)$$

Step 3 (Update of category): The selected category $\boldsymbol{w}_c(t)$ is updated as the following:

$$\boldsymbol{w}_c(t)^{(new)} = \begin{cases} \boldsymbol{w}_c(t)^{(old)} + a_t(P_j - \boldsymbol{w}_c(t)^{(old)}), & (\boldsymbol{w}_c(t) = \boldsymbol{w}_t) \\ \boldsymbol{w}_c(t)^{(old)} - a_t(P_j - \boldsymbol{w}_c(t)^{(old)}), & (\boldsymbol{w}_c(t) \neq \boldsymbol{w}_t) \end{cases} \quad (4)$$

where $\boldsymbol{w}_t$ is the category of the input given by the IART. $a_t \in [0, 1]$ is a learning parameter that is linearly monotone decreasing for time t:

$$a_t = -a_0(t/t_{total} - 1)$$

where $0 < a_0 < 1$ is a positive parameter and t_{total} is a total number of learning iterations.

Step 4 (Border decision): Let $t = t + 1$, goto Step 2 and repeat until the maximum time limit t_{tmax}. At time $t = t_{tmax}$, the borders of categories are decided by perpendicular bisector of the category position as shown in Fig. 3.

Step 5 (Category reassignment): If the LVQ-based category is different from the ART-based category, the LVQ-based category is to be valid.

3. Numerical Experiments

In order to consider the classification function, we apply the ART-LVQ algorithm to elementary problems whose the input space is given by union of three Gaussian distributions: $N_i(\mu_1, \mu_2; \sigma_1^2, \sigma_2^2)$, $i = 1 \sim 3$ where μ and σ is the mean and the standard deviation, respectively. In the experiments, we have used the following two problems based on three Gaussians ($N_c = 3$) as shown in Fig. 4(a):

$$\begin{aligned} P1 &: \begin{cases} N_1(0.4, 0.55; 0.1^2, 0.1^2) \\ N_2(0.5, 0.2; 0.1^2, 0.1^2) \\ N_3(0.75, 0.45; 0.1^2, 0.1^2) \end{cases} \\ P2 &: \begin{cases} N_1(0.4, 0.55; 0.1^2, 0.1^2) \\ N_2(0.5, 0.2; 0.08^2, 0.08^2) \\ N_3(0.75, 0.45; 0.12^2, 0.12^2) \end{cases} \end{aligned} \quad (5)$$

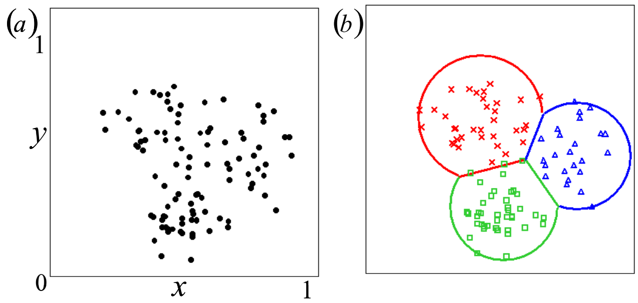

Figure 4: An example of $P2$. (a) Input data for IART (N = 100). (b) Classification results.

3.1. Categories extraction by IART

In the numerical experiments, the parameters are fixed after trial-and-errors:

$$\gamma = 0.35, \ K = 0$$

Fig. 4(b) shows an example of results for P_2. 100 inputs are selected randomly from $N_1 \cup N_2 \cup N_3$ and are applied to the IART. The IART has succeeded to extract three categories. In order to evaluate such results, we introduce two measures. The first one is the coincidence rate of the ART-based category with the right answer:

$$CR1 = \frac{\text{\# input data with correct category for ART}}{\text{\# input data}} \quad (6)$$

The second one is the extraction rate of the categories:

$$CR2 = \frac{\text{\# categories extracted by ART}}{\text{\# categories}} \quad (7)$$

Table 1 summarizes the results where $CR1$ is given in average, min and max for 37 trials where the 3 categories are extracted. $CR2$ is calculated for t_{max} trials. We can see that the case $N = 100$ gives the best $CR2$ for both $P1$ and $P2$. It suggests that over-learning disturbs category extraction ability. The case $N = 500$ gives the best $CR1$ in average: this $CR1$ is a criterion to consider the ART-LVQ function. It should be noted that the three Gaussian functions overlap to each other and higher $CR1$ is hard even if the classification is optimal.

3.2. Categories extraction by ART-LVQ

The parameters are fixed after trial-and-errors:

$$\gamma = 0.35, \ K = 0, \ a_0 = 0.6, \ t_{total} = 10^3$$

Fig. 5 shows an example of results for P_2. Here we introduce the third measure that is the coincidence rate of the ART-LVQ-based category with the right answer:

$$CR3 = \frac{\text{\# input data with correct category for ART-LVQ}}{\text{\# input data}} \quad (8)$$

Table 1: Results of IART.

	N	$CR1_{avg}$	$CR1_{min}$	$CR1_{max}$	CR2
$P1$	100	58.0	5.0	91.0	76.9
	300	69.7	5.0	91.7	36.3
	500	68.4	9.4	90.4	7.4
$P2$	100	55.4	12.0	90.0	79.0
	300	56.6	6.7	90.7	69.0
	500	60.8	6.8	90.6	50.2

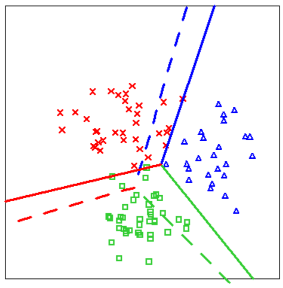

Figure 5: An example result of ART-LVQ for $N = 100$ and $P2$. The broken line is the correct border and the solid line is the boder by the ART-LVQ. The difference between two borders is measured by $CR3$.

Fig. 6 and Fig. 7 show dependence of $CR3$ on learning time t of the LVQ. In P_1, $t = t_{max}$ is sufficient to give good value of $CR3$. In P_2, $t = N$ is sufficient to give good value of $CR3$ The table 2 summarizes the results. $CR3$ are calculated for 37 trials of three categories by IART. In $P1$, $N = 300$ gives the best $CR3$ in average. In $P2$, $N = 500$ gives the best $CR3$ in average, however, $N = 100$ can realize reasonable improvement of $CR3$ for ART. It may suggest that the ART-LVQ can realize faster classification for $P2$ that is harder problem than $P1$.

4. Conclusions

We have presented the ART-LVQ algorithm that can realize unsupervised learning for classification. Performing elementary numerical experiment, the algorithm performance has been considered. We can suggest the following.

(1) The IART can play good classification performance for simple problems such as $P1$. However, over-learning may reduce the performance.

(2) The ART-LVQ is effective in the case of relatively complex problems such as $P2$. Also, fast classification is possible in that case.

Future problems are many, including the following: finding optimal algorithm parameter values, analysis of the learning process, and application to practical problems.

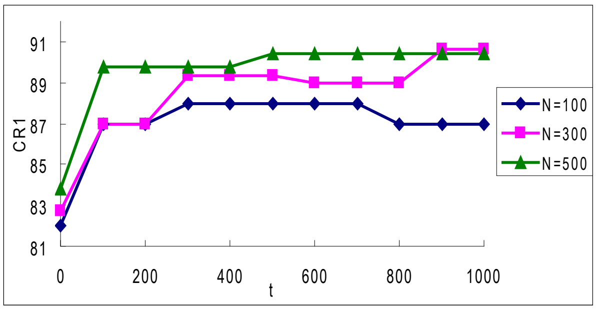

Figure 6: Time-dependence of $CR1$ for P1.

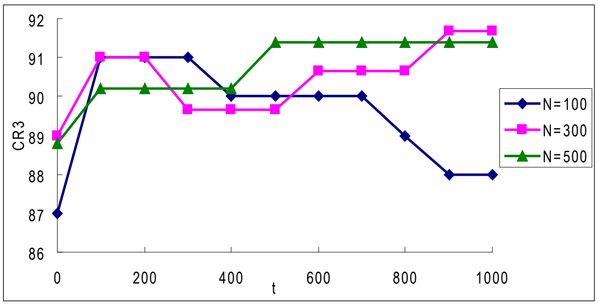

Figure 7: Time-dependence of $CR3$ for P2.

Table 2: Learning result $CR3$ of ART-LVQ. Repeat until the time $t_{max} = N$.

	N	$CR3_{avg}$	$CR3_{min}$	$CR3_{max}$
$P1$	100	57.5	1.0	88.0
	300	71.2	3.3	92.0
	500	69.2	5.8	91.8
$P2$	100	56.3	1.0	91.0
	300	56.8	5.0	91.3
	500	62.8	5.6	91.6

References

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

[2] M. Grbovic and S. Vucetic, Learning Vector Quantization with Adaptive Prototype Addition and Removal, Proc. of IEEE-INNS/IJCNN, pp. 994-1001, 2009.

[3] G. C. Anagnostopoulos and M. Georgiopoulos, Ellipsoid ART and ARTMAP for Incremental Clustering and Classification, IEEE Trans. Neural Networks, pp. 1221-1226, 2001.

[4] O. Parsons and G. A. Carpenter, ARTMAP neural networks for information fusion and data mining: map production and target recognition methodologies, Neural Networks, 16, pp. 1075-1089, 2003

[5] M. Ohki, H. Torikai and T. Saito, A simple radial basis ART network: basic learning characteristics and application to area measurement, Proc. NOLTA, pp. 262-265, 2005.

[6] T. Oshime, T. Saito and H. Torikai, ART-based parallel learning of growing SOMs and its application to TSP, Lecture Note on Computational Science, 4232, Springer, Neural Information Processing, I, pp. 1004-1011, 2006.

[7] M. Takanashi, H. Torikai and T. Saito, An approach to fusion of growing self-organizing maps and adaptive resonance theory maps, IEICE Trans. Fundamentals, E90-A, 9, pp. 2047-2050, 2007.

[8] H. Koshimizu and T. Saito, Parallel Ant Colony Optimizers with Local and Global Ants, Proc. of IEEE-INNS/IJCNN, pp. 1655-1659, 2009.