

# A New Clustering Method for Improving Plasticity and Stability in Handwritten Character Recognition Systems

Javad Sadri

Ching Y. Suen

Tien D. Bui

CENPARMI (Center for Pattern Recognition and Machine Intelligence)

Computer Science and Software Engineering Department, Concordia University

1455 de Maisonneuve Blvd. West, Montreal, Quebec, Canada, H3G 1M8

Tel: (514)-848-2424 Ext:7950, Fax: (514)-848-2830

{j\_sadri, suen, bui}@cse.concordia.ca

## Abstract

*This paper presents a new online clustering algorithm in order to improve plasticity and stability in handwritten character recognition systems. Our clustering algorithm is able to automatically determine the optimal number of clusters in the input data. An incremental learning technique similar to Adaptive Resonance Theory (ART) is used to determine the best cluster for new data. Our technique also allows the previously learned clusters to be merged whenever the newly arrived data points push their centers close together. We also developed new features and similarity measures in order to describe and compare the shapes of handwritten digits to be used in our clustering algorithm. Results of our algorithm on clustering the shapes of the handwritten numerals from the CENPARMI isolated digit database are shown. Our method can incrementally learn new handwriting styles of digits, without forgetting the previous ones, therefore it can improve plasticity and stability.*

## 1. Introduction

The determination of a set of representative prototypes to be used by a pattern recognition system is a very important design step [1]. In the case of handwritten character recognition, finding representative prototypes can be very complex because of the very large variability exhibited by handwritten characters. Therefore, a machine learning technique is required to automate this task. Machine learning techniques such as Neural Networks (NN) and Support Vector Machines (SVM), because of their very good generalization power, have received a great deal of attention over the last decade. However, one of the main limitations of these approaches is their low incremental learning capacity. Conventional neural networks, or support vector machines,

in order to learn new patterns, must be retrained. However, by retraining/learning new prototypes, these machines normally forget the old learned prototypes. In order to maintain the general performance of the system at a relatively high level, the retraining process should involve all the historical samples besides the new samples. So, we have to combine those samples into one huge data set and use it for retraining the system (classifier). However, this is not efficient in terms of both time and space. This can be considered as one of the main challenges in the design of evolutionary, efficient and robust handwritten character recognition systems.

Normally people's handwriting styles change with time, or new data samples/shapes of writing of characters are introduced and continuously added to already huge databases. One way to tackle this challenge is to introduce a recognition system that operates incrementally (that is able to learn new data continuously), and adapts to new writing styles without forgetting the previous ones. Learning new patterns continuously is called plasticity, and not forgetting the previously learned patterns is called stability. Adaptive Resonance Theory (ART) was proposed in order to resolve the plasticity-stability dilemma in machine learning [2].

In this paper, in order to achieve plasticity and stability in handwritten character recognition systems, we took the basic idea of the ART clustering algorithm [2], and we modified it. In our method, in order to improve the generalization in learning, whenever the centers of the neighboring clusters become very close to each other, (based on some similarity or distance measure), they will be merged into one cluster. Also in this paper, we introduce a new two dimensional feature representation, which enables us to capture the basic shapes of the handwritten digits. Based on these features, we also introduce a similarity measure in order to compare the shapes and structure of the digits. Using our clustering algorithm, features, and the similarity measure, we cluster

handwritten digits based on their shapes, and we find the centers of different clusters for each digit. These centers of clusters (basic shapes) will function as representative prototypes which can model each digit class, efficiently. Finally, a K-Nearest Neighbor (K-NN) classifier, utilizing our similarity measure, is used to assign class labels to new samples based on their similarity to the prototype shapes of each digit class. This system is able to incrementally learn the new shapes/styles of handwritten characters, without forgetting the previously learned ones.

The remaining part of this paper is organized as follows. Section 2 describes our proposed feature extraction and the similarity measure, while Section 3 introduces our proposed clustering algorithm. Section 4 describes the experimental results, while conclusions are drawn in Section 5.

## 2. Feature extraction and similarity measure

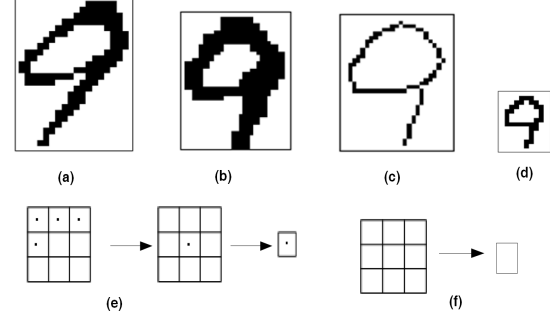
Feature extraction and similarity measure are two essential components of any clustering method. They will be described in this section and applied in Section 3.

### 2.1. Feature extraction

The images of isolated digits should be pre-processed prior to their feature extraction or recognition in order to remove noise or to smooth edges. Here, the method in [3] is used to remove the noise and to smooth the edges of the digits. Then the slants of the isolated digits are corrected according to [4], and their sizes are normalized into a matrix size of 45 by 45, using a moment-based normalization method [5]. Also, the center of gravity of each digit is transferred to the center of the image (45 by 45). Then, for feature extraction, the skeleton of each normalized and slanted digit is taken, and it is divided into 15 by 15 zones, such that each zone is a window of 3 by 3 pixels. If there is at least one black pixel in a zone, its center pixel is set to black; otherwise its center pixel will remain white. Then, all the pixels in a zone except for the center pixel will be removed. This transformation can greatly reduce the variations of the pixels in the skeletons of the handwritten digits, and it can extract and represent their basic shapes/structures. The details of this transformation and our feature extraction are shown in Figure 1. The transformed skeleton in Figure 1-d has 15 by 15 pixels, which is considered as a two dimensional feature vector representation.

### 2.2. Similarity measure

Similarity or dissimilarity (distance) measures also play an important role in clustering, or classification. Researchers have investigated similarity/dissimilarity measures for a century, and currently many similarity/distance



**Figure 1. An example of feature extraction: (a) Original image, (b) Pre-processed, slant corrected, and normalized image (45 by 45 pixels), (c) Skeleton of part b, (d) Reducing the resolution of the skeleton in horizontal and vertical directions by 1/3; the resulting image is considered as a feature vector. (e, and f) are two examples of our transformation. All the black pixels inside the windows are represented by a single pixel in the center of the window; white pixels around the center are removed.**

measures are available in the literature such as Euclidean distance, Inner product, Hamming distance, Rogers-Tanimoto similarity [6], etc. In this paper, we took Rogers-Tanimoto similarity measure ( $S_{R-T}(X, Y)$ ) in Equation 1, and we modified it a little bit and denoted it by ( $MS_{R-T}(X, Y)$ ) in Equation 2, as follows:

$$S_{R-T}(X, Y) = \frac{X^t Y + \bar{X}^t \bar{Y}}{X^t Y + \bar{X}^t \bar{Y} + 2X^t \bar{Y} + 2\bar{X}^t Y} \quad (1)$$

$$MS_{R-T}(X, Y) = \frac{\alpha X^t Y + \beta \bar{X}^t \bar{Y}}{\alpha X^t Y + \beta \bar{X}^t \bar{Y} + 2X^t \bar{Y} + 2\bar{X}^t Y} \quad (2)$$

where  $\alpha \geq 1$ ,  $\beta \geq 0$ , and  $0 \leq MS_{R-T}(X, Y) \leq 1$

Here  $X$ , and  $Y$  are two binary feature vectors (where zeros stand for white pixels, and ones stand for black pixels), and  $\alpha$ , and  $\beta$  are two adjustable weights (credits), and  $X^t Y$  stands for inner product. Rogers-Tanimoto similarity measure is always between 0 and 1. If two pattern vectors,  $X$  and  $Y$ , are exactly the same, their similarity, based on this measure, is equal to 1. Whenever the two pattern vectors are completely dissimilar (for example, logical complements of each other), their similarity measure is equal to 0. In addition to these advantages, our modified measure in Equation 2 introduces a family of similarity measures which enable us to apply different weights or larger credits for positive matching (black to black pixels matching:

$X^t Y$ ) or negative matching (white to white pixels matching:  $\overline{X^t Y}$ ). We can say that Rogers-Tanimoto similarity is a special case of this family where both  $\alpha$  and  $\beta$  are taken equal to 1.

### 3. Clustering algorithm

Clustering aims at discovering groups of similar patterns, and identifying meaningful structures or useful patterns in large datasets. There has been a lot of research on clustering; see [7] for a survey of clustering algorithms. Since the ART algorithm has incremental learning capabilities, we took the basic idea of this algorithm, and we modified it as below:

- 0 adjust the minimum similarity threshold ( $\delta$ )
- 1 cluster\_centers\_list =  $\phi$
- 2 read the next pattern.
- 3 find the most similar cluster center to the pattern in cluster\_centers\_list with similarity greater than  $\delta$   
 if found: assign the pattern to that cluster; Adjust cluster center, and check if this new center becomes close (with similarity greater than  $\delta$ ) to any other cluster center, then merge the corresponding clusters as one cluster.  
 if not found: build a new cluster, and insert the input pattern to cluster\_centers\_list as a new cluster center.
- 4 repeat steps in 2-3 for all the input patterns

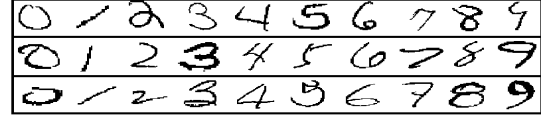
**Figure 2. Our clustering algorithm**

As seen in this algorithm,  $\delta$  (minimum similarity threshold) is the only parameter that must be adjusted before running the algorithm. Unlike off line clustering algorithms (such as K-Means, here, we do not specify the number of clusters a priori. Also, in order to find the centers of the clusters, our algorithm scans (visits) all the input patterns just one time. These properties of the algorithm makes it suitable for the clustering of online incoming data.

### 4. Experimental results

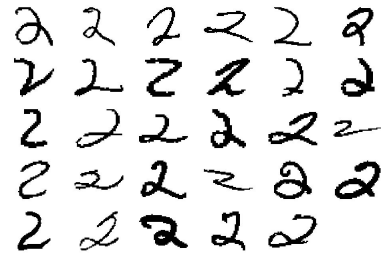
For our experiments we used the CENPARMI (Center for Pattern Recognition and Machine Intelligence) handwritten digit database. This database has 4000 training samples (400 samples per digit) and 2000 testing samples (200 samples per digit). Samples in the CENPARMI database show more variations in their shapes, compared to similar databases such as MNIST, we selected therefore this database for our experiments. Some samples of this database are shown in Figure 3.

In our experiments, we took  $\alpha$ , and  $\beta$  (parameters of the similarity measure) both equal to 2.5, and  $\delta$  (parameter of the clustering algorithm) equal to 0.70. We applied our clustering algorithm using the features and the similarity measure described in Section 2 to training samples of



**Figure 3. Samples from the CENPARMI isolated handwritten digit database.**

each digit in the CENPARMI database (400 training samples per digit). Afterwards, we found the centers of the clusters for each digit, and we considered those centers as candidate prototypes. As an example, Figure 4 shows all cluster centers (29 prototypes) obtained for digit 2.



**Figure 4. Centers of clusters (29 prototypes) of digit 2 in the training set of the CENPARMI isolated handwritten digit database.**

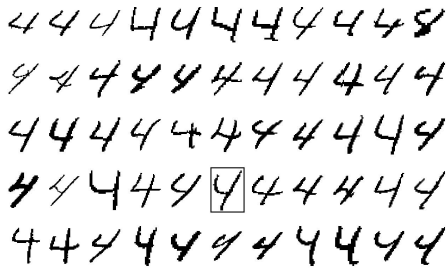
Table 1 shows the number of cluster centers obtained per digit class. With the above values of the parameters ( $\alpha$ ,  $\beta$ , and  $\delta$ ), our clustering algorithm totally was able to find 342 clusters of different shapes, which is much smaller than the number of training samples in the CENPARMI database (4000). This greatly reduces the computation time, and memory space required for the system, yet it maintains the high performance, using fewer prototypes.

**Table 1. Number of clusters per each digit class in the training set of the CENPARMI database**

Digits	0	1	2	3	4	5	6	7	8	9	Total
Number of clusters	12	4	29	46	43	32	41	44	56	35	342

As another example, one of the clusters of digit 4 is shown in Figure 5. The center of this cluster, obtained by our algorithm, has been identified by a rectangular bounding box. In our method, the center of a cluster is a pattern that has a basic shape which contains most of the common structural parts of the other samples in the cluster. By adding new samples to a cluster, its center can move.

However, at any case algorithm consider center as a pattern which has maximum similarity to all the members of its corresponding cluster. As seen by these examples, our clustering algorithm for new shapes of different digits (or new shapes of the same digits) which are dissimilar (having similarity less than  $\delta$ ) makes new clusters. So, it is able to incrementally learn those new shapes (plasticity), at the same time it does not forget the old clusters, and it maintains the previously learned knowledge (stability).



**Figure 5. One of the clusters of digit 4. This cluster has 55 samples, and our clustering algorithm has identified one of the patterns of this cluster as its center (representative, or prototype).**

The classifier used in our experiments is based on K-Nearest Neighbor rule: an input digit is compared against all the prototypes (centers of all the clusters for all digits) and the most K similar ones vote for its classification. In our experiments, we took K=3. We applied our classifier to all testing samples (200 samples per class) in the CENPARMI digit Database, and the results are shown in Table 2. Our overall recognition result on the CENPARMI test set is %98.23, which is a little bit lower than the results reported by [8] (%99.05: using SVM\_rbf kernels). However, our goal in this paper was not to reach the maximum recognition rate, but instead to design a system with higher plasticity and stability.

## 5. Conclusions and future works

In this paper, we presented a new clustering algorithm for improving plasticity and stability of handwritten character recognition systems. Our method can incrementally learn new handwritten styles of characters, without forgetting the previously learned ones. It is able to automatically determine the optimal number of clusters. Also, it adaptively allows the learned clusters to be merged to make broader concepts. By adjusting its parameter, our algorithm can even find outlier patterns. Unlike, off line clustering algorithms, our method scans the input data just one time, so it is normally faster, and it can be used for clustering on line input

data. In the future, we would like to expand our system, in order for it to learn handwritten mathematical symbols, and also characters from other languages (Farsi/Arabic,...) to be able to recognize multi-language handwritten documents.

**Table 2. Recognition rates per each digit class and overall recognition rate in the testing set of the CENPARMI database.**

Digits	Recog. Rate (%)	Digits	Recog. Rate (%)	Digits	Recog. Rate (%)
0	99.71	4	97.64	8	98.78
1	99.82	5	99.05	9	98.84
2	98.74	6	99.12	<b>Overall</b>	<b>98.23</b>
3	98.30	7	98.10	—	—

## References

- [1] L. P. Cordella, C. De Stefano, A. Della Cioppa, and A. Marcelli. "A New Evolutionary Learning Model for Handwritten Character Prototyping," Proc. ICIAP'99 (10th International Conference on Image Analysis and Processing), pp. 830-835, 1999.
- [2] G. A. Carpenter and S. Grossberg. "The ART of Adaptive Pattern Recognition by a Self-Organizing Neural Network," IEEE Computer, 21(3), pp. 77-88, 1988.
- [3] N. W. Strathy, C. Y. Suen, and A. Krzyzak, "Segmentation of Handwritten Digits Using Contour Features," Proc. 2<sup>nd</sup> ICDAR, pp. 577-580, Oct. 1993.
- [4] A. D. S. Britto JR., R. Sabourine, E. Lethelier, F. Bortolozzi, and C. Y. Suen, "Improvement in Handwritten Numeral String Recognition by Slant Correction and Contextual Information," Proc. 7<sup>th</sup> IWFHR, pp. 323-332, Sept. 2000.
- [5] C. L. Liu, K. Nakashima, H. Sako, and H. Fujisawa, "Handwritten Digit Recognition: Investigation of Normalization and Feature Extraction Techniques," Pattern Recognition, Vol. 37, pp. 265-279, 2004.
- [6] D. J. Rogers and T. T. Tanimoto, "A Computer Program For Classifying Plants," Science, 132(3434), pp. 1115-1118, 1960.
- [7] R. Xu, and D. Wunsch II, "Survey of Clustering Algorithms," IEEE Trans. On Neural Networks, Vol. 16, NO. 3, May 2005.
- [8] C.L. Liu, K. Nakashima, H. Sako, and H. Fujisawa, "Handwritten Digit Recognition: Benchmarking of State-the-Art techniques," Pattern Recognition, Vol., 36, pp. 2271-2285, 2003.