

# REAL TIME CLUSTERING MODEL

J. Cheng, M. R. Sayeh

*Department of Electrical and Computer Engineering  
Southern Illinois University Carbondale, Carbondale, IL 62901, U.S.A.*

M. R. Zargham

*Department of Computer Science, Southern Illinois University Carbondale, Carbondale, IL 62901, U.S.A.*

Keywords: ODE, Clustering, Vector quantization, Real time.

Abstract: This paper focuses on the development of a dynamic system model in unsupervised learning environment. This adaptive dynamic system consists of a set of energy functions which create valleys for representing clusters. Each valley represents a cluster of similar input patterns. The system includes a dynamic parameter for the clustering vigilance so that the cluster size or the quantizing resolution can be adaptive to the density of the input patterns. It also includes a factor for invoking competitive exclusion among the valleys; forcing only one label to be assigned to each cluster. Through several examples of different pattern clusters, it is shown that the model can successfully cluster these types of input patterns and form different sizes of clusters according to the size of the input patterns.

## 1 INTRODUCTION

As stated in (Jain, 1988), "Cluster analysis is the process of classifying objects into subsets that have meaning in the context of a particular problem." In other words, clustering is a process of grouping a set of unlabeled data. As shown in Figure 1, in general, clustering can be grouped into two types: non-overlapping (exclusive) and overlapping (nonexclusive). In non-overlapping, each object input will be assigned to only one cluster whereas in overlapping an object can be assigned to more than one cluster. In this paper we only consider non-overlapping clustering. Non-overlapping clustering could lie either intrinsic or extrinsic. In the intrinsic approach, also called unsupervised learning, a proximity matrix is the only criteria used. (Proximity matrix represents relationship between the objects; if the objects are patterns such matrix could represent the distance between the patterns). The extrinsic approach, also called supervised learning, in addition to proximity matrix, it also uses category labels on the objects. To notice the difference between these two approaches, let's consider a set of data representing health condition of normal and overweight children. Using intrinsic approach, we

can group these children based on these factors and then try to determine whether overweight plays a role in academic status. Taking an extrinsic approach allow us to study the way of separating normal and overweight children by considering their health conditions.

We consider intrinsic approaches only. The intrinsic methods can be divided into five types: hierarchical, partitional, grid-based, artificial neural networks, and evolutionary (Jain, 1988; MacQueen, 1967; Grossberg, 1976; and Kohonen, 1982).

Comparison of these clustering methods is hard to do using simulation because of different implementation of the methods and the data that is used. It is also hard to do theoretically comparison of them because they are almost impossible to model mathematically (Jain, 1988). Furthermore, the existing models impose architectural complexity and/or time complexity which prevent them of having real time response time. To overcome the real time mathematical modeling problems, we have proposed a new method which depends solely on ordinary differential equations (ODE) (Cheng, 2006). There is no need for IF/THEN logical statements. Therefore it can be easily implemented on the analog type devices to take advantage of high-speed electronics or photonics technologies.

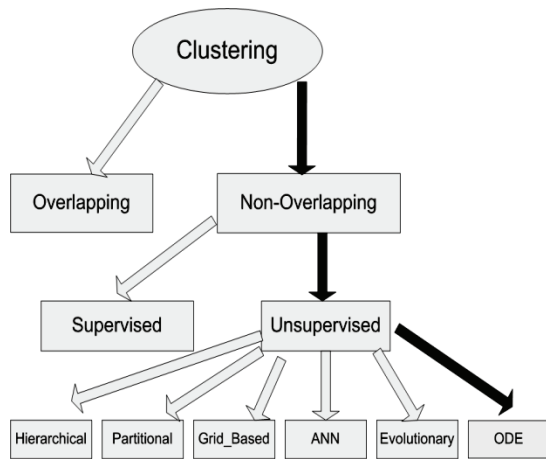


Figure 1: Different Clustering Methods.

The organization of this paper is as follows: Section 2 describes the adaptive dynamical model and devise an additional state for the dynamics of vigilance parameter  $\lambda$ , which plays an important role in our system; Section 3 illustrates the performance of the proposed model on several examples; Finally, Section 4 presents the conclusion.

## 2 THE CLUSTERING MODEL

The idea behind our model is to store input patterns on the surface of an energy function of a dynamic system. The clusters are represented as valleys on the surface of an energy function. To demonstrate this concept, let's consider an input  $U$  in one dimensional space as shown in Figure 2. Also let  $W$  be a representative of a cluster and is randomly placed on the surface of the energy function as shown in Figure 2 by a black dot. The energy function is represented by the black line, and the input  $U$  is denoted by a gray dot. As the dynamic progresses, the  $W$  moves toward  $U$  and forms a complete valley when it reaches to  $U$ . Similarly, in cases where there are more than one input pattern, the valleys are created for patterns that are close to each other.

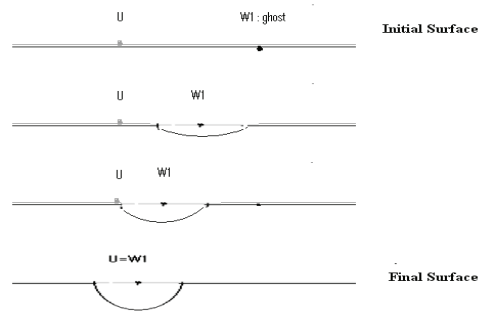


Figure 2: One-Dimensional Example of Process of Clustering forming by our model.

Generally, considering an  $N$  dimensional space for input patterns, the  $V$  energy function is constructed to represent  $M$  valleys centered at locations  $W_j = (w_{1j}, w_{2j}, \dots, w_{Nj})^T$ , for  $j = 1$  to  $M$ , representing  $M$  clusters for  $P$  input patterns. The  $V$  energy function is represented as

$$V = \sum_{p=1}^P \left[ \gamma - \sum_{j=1}^M x_{jp} \right]^2$$

Here,

$$x_{jp} = \frac{1}{1 + \lambda_j \| U_p - W_j \|^2}$$

where  $\| \bullet \|$  is the Euclidean norm,

$W_j = (w_{1j}, w_{2j}, \dots, w_{Nj})^T$ , for  $j = 1$  to  $M$ , represent center of valley  $j$ ,

$U_p$  is the input pattern  $p$ ,

$M$  is the number of clusters, and

$P$  is the number of input patterns.

$W$  and  $U$  refer to vectors when  $N > 1$ .

The constant  $\gamma$  encourages a number of valleys (clusters) to be formed and the factor  $\lambda$  (vigilance parameter) approximately reflects the radius of the generated valley.

In order to invoke competitive exclusion among the valleys, the following competitive C-energy function is used

$$C = \sum_{p=1}^P \sum_{i=1}^M \sum_{s \neq i}^M x_{ip} x_{sp}$$

This function guarantees only one of the valleys encodes the patterns by achieving its minimal value.

Based on the above two functions, the total R-energy function is constructed, which is the summation of the V-energy function and C-energy function.

$$R = \sum_{p=1}^P \left[ \gamma - \sum_{j=1}^M x_{jp} \right]^2 + \rho \sum_{p=1}^P \sum_{i=1}^M \sum_{s \neq i}^M x_{ip} x_{sp}$$

where  $\rho$  is a balancing factor.

The  $\lambda_j$ ,  $j=1, \dots, M$ , (vigilance parameter) approximately reflects the radius of the generated valley  $j$ . As  $\lambda_j$  gets smaller, valley  $j$  becomes wider.

The  $x_{jp}$  ( $j=1, \dots, M$ ), which has a value between 0 and 1, represents the depth of valley  $j$ . As the distance between  $U_p$  and  $W_j$  decreases, the value of  $x_{jp}$  moves toward 1 indicating  $W_j$  as a cluster for  $U_p$ . This model is a gradient dynamical system with  $R$ ; thus

$$\frac{\partial R}{\partial w_{ij}} = -4\lambda_j \sum_{p=1}^P \left[ \gamma - \sum_{q=1}^M x_{qp} + \sum_{s \neq q}^M x_{sp} \right] * x_{jp}^2 (u_{ip} - w_{ij})$$

$$x_{jp} = \frac{1}{1 + \lambda_j \left( \sum_{i=1}^N (u_{ip} - w_{ij})^2 \right)}$$

where  $x_{jp}$  is explicitly given as

Thus, the dynamics of the system can be expressed as:

$$\frac{dw_{ij}}{dt} = 4\lambda_j \sum_{p=1}^P \left[ \gamma - \sum_{q=1}^M x_{qp} + \sum_{s \neq q}^M x_{sp} \right] * x_{jp}^2 (u_{ip} - w_{ij})$$

Now let's consider system state for the vigilance parameter  $\lambda$  so that the cluster size can be adaptive to the size of the input patterns;  $\lambda_{adjust}$  is a function which can be expressed as

$$\lambda_{adjust} = \sum_{j=1}^M \left( \frac{P * \lambda_{initial}}{\gamma} - \lambda_j \sum_{p=1}^P x_{jp} \right)^2$$

Here,  $\lambda_{initial}$  is the initial value of  $\lambda$ , usually specified by the user.

The  $\lambda$  dynamics can be derived as

$$\frac{\partial \lambda_{adjust}}{\partial \lambda_j} = 2\lambda_j \sum_{p=1}^P \sum_{j=1}^M \left[ \frac{P * \lambda_{initial}}{\gamma} - \lambda_j \sum_{q=1}^P x_{jq} \right] * x_{jp}^2 \sum_{k=1}^N (u_{kp} - w_{kj})$$

### 3 CLUSTERING PERFORMANCE

This section demonstrates simulation results of our model on two sets and four sets of 2D input patterns as shown in figures 3, 5, 7, and 10. In Figure 3, there are two sets of input patterns where each one includes 10 input patterns. In Figure 5, there are two sets of input patterns, one contains 20 and the other contains 10 inputs. In Figures 7 and 10, there are four sets of input patterns, containing 5, 10, 15 and 20 inputs. Figures 3, 5, and 7 include 4  $W$ s while Figure 10 includes 6  $W$ s. Input patterns are denoted as squares (gray color) and  $W$ s represented as diamonds (black color). The coordinate one and coordinate two represent the first and second coordinates of input patterns ( $U$ ) and weights ( $W$ ). Each set of inputs are randomly generated having a value between 0 and 1, while weights are placed at some specific points. Figures 4(a), 6(a), 8(a), 9(a), and 11(a) demonstrate tracking paths and final positions of the  $W$ s in figures 3, 5, 7, 7 and 10, respectively. The small circles represent the final position of  $W$ s; the ones with grey color represent the center of the clusters formed by winner  $W$ s; and the ones with white color represent final position of  $W$ s, which cannot win the competition. Figures 4(b), 6(b), 8(b), 9(b), and 11(b) are the final contour images of the energy function for each of these cases. The surfaces of these images represent the  $V$ -energy function in terms of different input values. Each generated valley is represented by a set of nested rings with different shades representing different levels of valley's depth (black represents the deepest point and the dark gray represents the surface). Figure 4 (a) demonstrates the set of obtained clusters for initial  $\lambda$  equal to 20, and  $\gamma$  equal to 2. As can be seen, two  $W$ s win the competition to form clusters for each set of inputs and the other two  $W$ s move away from the input patterns. As can be observed from the contour image in Figure 4 (b), two valleys are formed representing two different clusters.

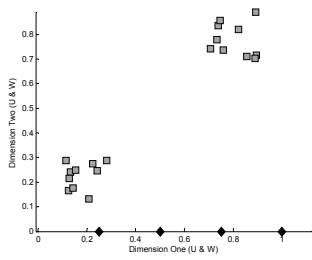


Figure 3: Two clusters; there are 20 input patterns (represented by squares) and 4  $W$ s (represented by the diamonds) positioned at the bottom of the inputs.

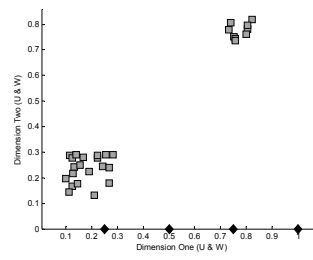
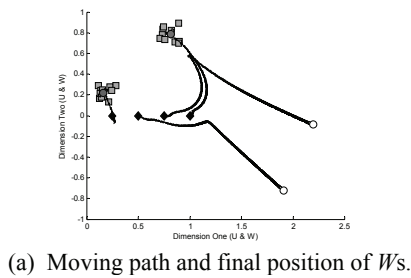
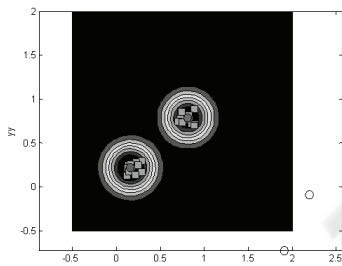


Figure 5: Two clusters; there are 30 input patterns (represented by squares) and 4  $W$ s (represented by the diamonds) positioned at the bottom.

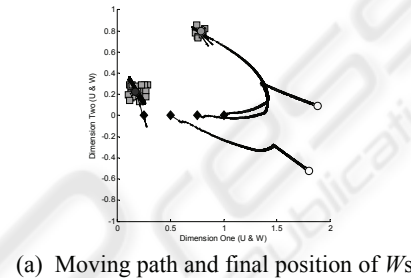


(a) Moving path and final position of  $W$ s.

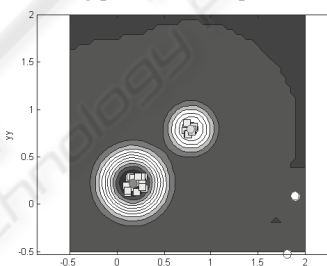


(b) Contour representation.

Figure 4: Final positions of  $W$ s and contour representation of the  $V$ -energy function.



(a) Moving path and final position of  $W$ s.



(b) Contour representation.

Figure 6: Final positions of  $W$ s and contour representation of the  $V$ -energy function.

Figure 5 (a) demonstrates the set of obtained clusters for initial  $\lambda$  equal to 20, and  $\gamma$  equal to 2. As can be seen, two  $W$ s win the competition and form clusters. The other two  $W$ s move away. In this simulation,  $\lambda$  values from initial value 20 change to 15.7, 351.5, 260.2, and 29.3 for each of four  $W$ s. Two winning  $W$ s have two different  $\lambda$  values 15.7 and 29.3 since the two sets of input patterns have different size. The other two runaway  $W$ s have  $\lambda$  values as 351.5 and 260.2. The bigger set of input patterns corresponds to smaller  $\lambda$  values (resulting bigger cluster), and the smaller set of input patterns corresponds to bigger  $\lambda$  values (resulting smaller cluster).

Figure 8 (a) demonstrates the set of obtained clusters for initial  $\lambda$  equal to 20, and  $\gamma$  equal to 4. As can be seen, four  $W$ s form clusters for each set of inputs. After simulation stops,  $\lambda$  values from initial value 20 change to 10.1, 11.9, 29.7, and 17.5 forming four different size clusters.

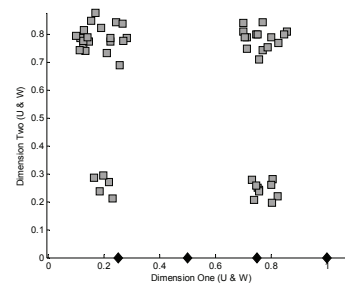
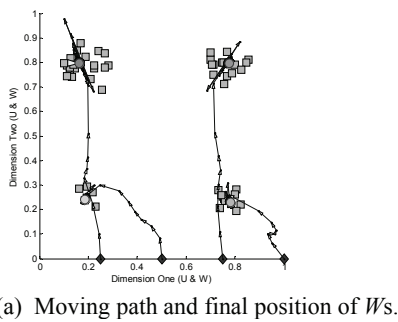
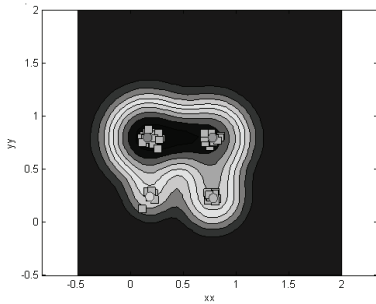


Figure 7: Four clusters; there are 50 input patterns (represented by squares) and 4  $W$ s (represented by the diamonds) positioned at the bottom.



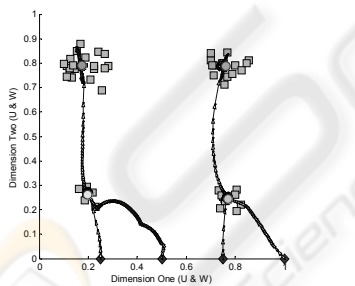
(a) Moving path and final position of  $W_s$ .



(b) Contour representation.

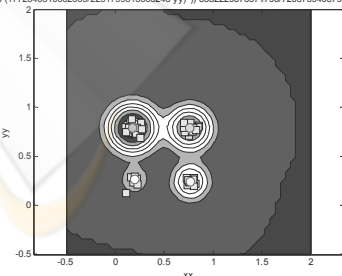
Figure 8: Final positions of  $W_s$  and contour representation of the  $V$ -energy function.

Figure 9 (a) demonstrates the same experience as Figure 8 (a) except in this case the initial value for  $\lambda$  is set equal to 40. After simulation stops,  $\lambda$  values from initial value 40 change to 24.2, 30.0, 84.4 and 43.8.



(a) Moving path and final position of  $W_s$

5328 (1772340310682865/2251799813685248-yy<sup>3</sup>)<sup>3</sup>-853222987571793/72057594037927936(1+)



(b) Contour representation

Figure 9: Final positions of  $W_s$  and contour representation of the  $V$ -energy function.

Figure 11 (a) demonstrates the set of obtained clusters for initial  $\lambda$  equal to 25, and  $\gamma$  equal to 4. As can be seen, out of 6  $W_s$ , four  $W_s$  win and form clusters. After simulation stops,  $\lambda$  values for these winning  $W_s$  become 13.3, 41.1, 15.9, and 23.4.

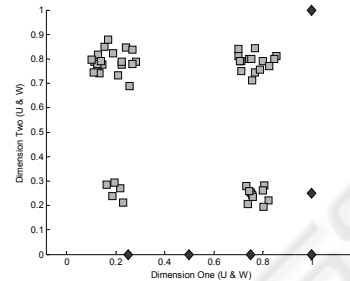
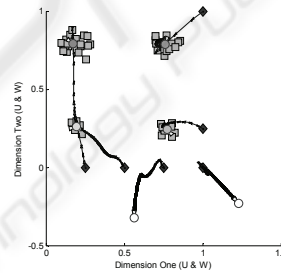
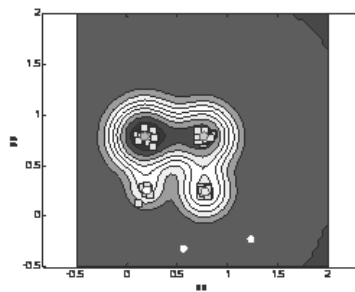


Figure 10: Four clusters; there are 50 input patterns (represented by squares) and 4  $W_s$  (represented by the diamonds) positioned at the right lower corner.



(a) Moving path and final position of  $W_s$ .



(b) Contour representation.

Figure 11: Final positions of  $W_s$  and contour representation of the  $V$ -energy function.

## 4 CONCLUSIONS

The main purpose of this paper was the introduction of a novel dynamical system for clustering which has potential for real-time device realization. We also devise an additional system state for the vigilance parameter  $\lambda$  so that the cluster size or the quantizing resolution can be adaptive to the size of

the input patterns. This reduces the burden of re-tuning the vigilance parameter for a given input pattern set and it will also better represent the input pattern space. These discussions are furthermore visualized by simulation examples. As shown from simulation results, our dynamic system can successfully cluster different input patterns by dynamically adjust  $\lambda$  values according to the size of the input patterns in order to form different sizes of clusters. In this system, self-organizing properties can be implicitly coded within the system trajectory structure using only ODE's. These ODE's can be directly implemented in hardware through feed-back networks by analog electronic or optical implementation.

## REFERENCES

- Block, H.-H., 2002. Clustering Methods: From Classical Models to New Approaches, *Statistics in Transition*, Vol. 5, No. 5, October 2002, pp. 725-758.
- Grossberg, S. 1976. Adaptive pattern classification and universal recoding, II: feedback, expectation, olfaction, and illusions, *Biological Cybernetics*, 23, 187-202.
- Jain, A. K., & Dubes, R. C. 1988. *Algorithms for Clustering Data*. New Jersey: Prentice Hall, Englewood Cliffs.
- Kohonen, T. 1982. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43, 59-69.
- MacQueen, J. B. 1967. Some methods for classification and analysis of multivariate observations, *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, University of California Press, Berkeley, 1:281-297.
- Cheng, J., Sayeh, M.R., & Zargham, M.R. 2006. Neural Net Based Models for Clustering. *International Journal of Computational Intelligence Theory and Practice*, 1(2), 91-102.