

USING A DIFFUSION-LIKE PROCESS FOR CLUSTERING

O. Yadid-Pecht and M. Gur
Julius Silver Institute for Biomedical Sciences
Department of Biomedical Engineering
Technion – Israel Institute of Technology
Haifa 32000, Israel

ABSTRACT

A simple clustering method using a neural net, which implements a diffusion-like, process is suggested. The implementation requires basic elements, numbered as the number of pixels, that work in parallel. The units can be viewed as simple "neurons", requiring only a small number of local connections. In spite of its simplicity, this implementation has several advantages over commonly used fuzzy clustering methods. Specifically, it is not dependent on initial conditions and it provides the "typicality" notion that is lacking in the well known Fuzzy C means (FCM) and its derivatives.

1. INTRODUCTION

Clustering is a well known problem in numerous applications. Fuzzy clustering has proven to be advantageous and is of considerable importance in research and implementation. Neural nets are widely used in performing low-level operations, such as feature extraction in pattern recognition, and are very easy to implement.

We propose using neural networks for the clustering problem and further suggest "probability" clustering, in which each data point has a degree of probability of belonging to the cluster. This is implemented by simple "neurons" with a small number of local connections. The "heart" of the system is a "diffusion" neural net. This net implements a process which uses lateral excitation to spread activity locally, generating long-range interactions over time. Depending on the extent of the diffusion process, i.e. the time elapsed, it generates peaks over areas of growing size. A maxima selection net uses lateral inhibition to localize the apex of those local peaks. These processes are implemented in neural networks and can yield a "fuzzy" clustering system. The main advantage using our approach is that simple local calculations, performed in parallel, result in the desired clustering.

2. GENERAL DESCRIPTION

The suggested method (for C clusters) can be summarized as follows:

1. Determine an n-dimensional grid of dots where each dimension presents one attribute used for clustering. Each grid point location corresponds to the attributes

- of a point in the feature space. Initialize to 1 all grid points corresponding to data points 1. All other dots are initialized to 0.
2. Perform the diffusion process according to the heat equation (see below). In this process each point "diffuses" its value to its neighbors. Activation starts spreading from dots with initial value 1 to their surroundings. After a while, new local maxima emerge as local activations are summed. Since diffusion continues, those local maxima can be viewed as centers of clusters of growing size. Eventually, only one point will be a local maximum – the one which is the centroid of all data. We are interested in the local maxima preceding this single one. Local maxima that stay in their locations for several iterations indicate that the clusters they represent are "stabilized" enough, i.e. no new dots which change its center location are added to the cluster.
 3. Find C highest local maxima which are sufficiently "stabilized", since the larger the value the more dots it represents – for a specified radius.
 4. For each local maximum, all points within a radius r are assigned to this cluster. Their memberships are a function of the diffusion values.
 5. Go to step 2 until the required "zone of influence" for every cluster is achieved. It can be shown that the radius of influence and the number of iterations are connected.

2.1. Diffusion Dynamics

The authors use the heat flow equation for solving the clustering problem: specifically, a diffusion simulation of the heat flow process. Without losing generality, discussed here is the one dimensional heat flow equation, which is given by:

$$U_t = U_{xx} \quad (t > 0), \quad U(x, 0) = f(x) \quad (-\infty < x < \infty) \quad (1)$$

where $U(x,t)$ is the function that has to be determined. The fundamental solution to the heat flow problem is given by:

$$u(x, t) = \frac{1}{2\sqrt{\pi t}} e^{-(x-\xi)^2/(4t)} \quad (2)$$

which can be interpreted as the probability density of a random variable ξ that is normally distributed with a mean x and a standard deviation $\sqrt{2t}$. This can also be regarded as the probability distribution of the position at time t for a particle known to have started from $\xi=x$ at time $t=0$ [1].

In order to simulate the formation of the problem in terms of finite differences, a spacing h in the x direction and a spacing k in the t direction was introduced by Hilderbrand [2], such that

$$U_{m,n} = U(x_m, t_n) = U(mh, nk) \quad (3)$$

After some approximations, the simulating problem consists of determining $U_{m,n}$ such that the "partial difference equation"

$$U_{m,n+1} = \rho U_{m+1,n} + (1 - 2\rho) U_{m,n} + \rho U_{m-1,n} \quad (4)$$

holds for $m = 0, \pm 1, \pm 2, \dots$ and $n = 0, 1, 2, \dots$; subject to the initial conditions:

$$U_{m,0} = f_m = f(x_m) \quad (5)$$

prescribed for $m = 0, \pm 1, \pm 2, \dots$

We simulate the diffusion process in a neural net. Two dimensional diffusion can be implemented as a Cartesian frame in which two neighboring neurons are connected with a weight of strength k_{diff} – the diffusion constant (denoted as ρ in the one-dimensional case). Each neuron inhibits itself with a weight of $4k_{diff}$ (See Fig. 1), while k_{diff} satisfies the stability condition. The approximation for the two dimensional case is:

(6)

$$\frac{\partial U(x, y)}{\partial \tau} = k_{diff}[U(x-1, y) + U(x, y-1) + U(x+1, y) + U(x, y+1) - 4U(x, y)]$$

where k_{diff} is the diffusion constant, and $U(x,y,t)$ is to be determined.

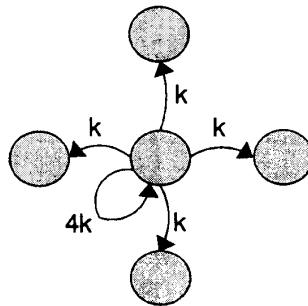


Figure 1: A possible implementation of the diffusion process is a neural network with very simple elements in which each neuron is connected to its neighbors as described.

The successive arrays corresponding to the use of successive simultaneous relaxation processes can be interpreted as specifying approximate successive temperature distributions on N at different times (see Fig. 2).

At each time-step, the largest peaks are selected as cluster centers. This is also implemented in a neural net; local maxima detectors precede a modified Maxnet [4] in which the C largest local maxima are selected.

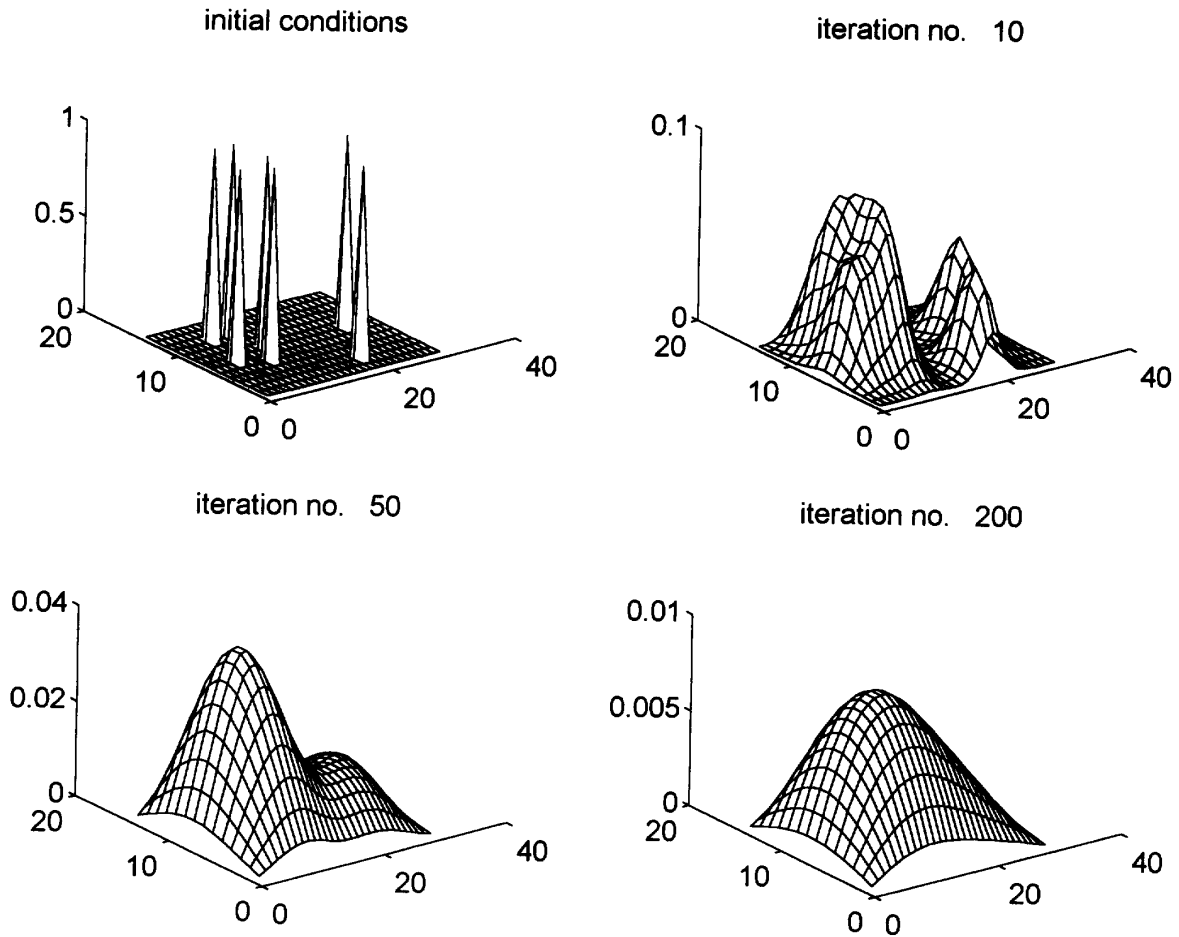


Figure 2: Evolving clusters during the diffusion process.

3. EXPERIMENTAL RESULTS

When using the FCM algorithm, presence of noise in the data (see Fig. 3a) can drastically influence the estimates of the class prototypes. "Dumping" all the outliers into a noise cluster is a solution but to perform this, one needs the means to identify the outliers, i.e. to consider cluster validity. However, validity-based methods are computationally expensive since it is necessary to perform the clustering for a range of C values. Even more important, validity based methods do not give membership values that can be interpreted as degrees of compatibility.

Results of implementing the diffusion process on the input set (Fig. 3a) are described in Fig. 3b. If only 2 clusters are desired, it can be seen that the proposed clustering can

identify and eliminate noise outliers from a data set in a natural way, by choosing the two largest maxima, thus reducing the risk of biasing the results.

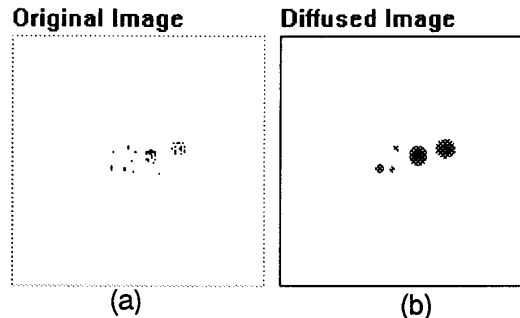


Figure 3: (a) A data set consisting of two classes with noise added. (b) Results of applying the diffusion- like process to the data set of figure 3a.

Another problem encountered using FCM is that membership values which are unrepresentative of the degree of belonging can be assigned [3]. Consequently, membership values cannot distinguish between a moderately atypical member and an extreme atypical member (see Fig. 4a).

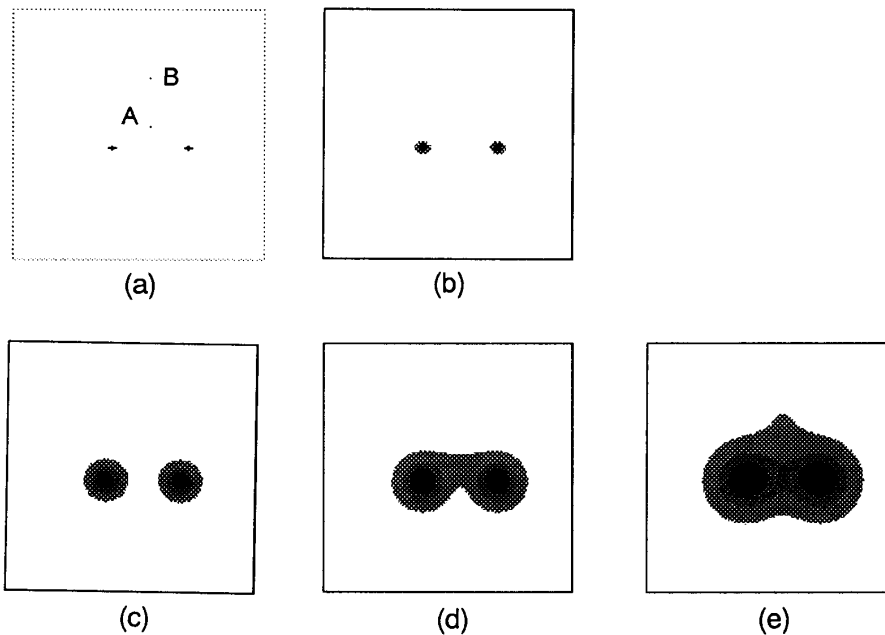


Figure 4: (a) Example of a data set in which the memberships of points A and B resulting from the FCM algorithm are approximately 0.5., even though point A is a "better" member of both clusters. Diffusion results at 20 (b), 200 (c), 400 (d) and 800 (e) iterations.

Implementing the diffusion process for clustering the data set results in that the two clusters are formed quite quickly and are very stable, i.e. their centers remain static for

quite a long period (see Fig. 4b,4c). After many iterations (see Fig. 4d), point A can be included in one of the two clusters, although its membership value is very small. Point B still remains an outlier. In Fig. 4e both points may be not considered outliers, yet point B, naturally, has a lower membership value than point A.

4. DISCUSSION

There are a number of benefits in using the proposed method.

First, it requires only simple local calculations, performed in parallel while the Euclidian distance is implicitly embedded. In contrast, commonly used clustering algorithms require a very large number of computations.

Second, it offers the notion of "typicality". This feature was already shown to be useful [3] and does not always exist in the FCM algorithm.

Third, the results do not depend on initialization, contrary to most FCM algorithms and their derivatives.

Fourth, the process is less sensitive to noise. In commonly used fuzzy C means algorithms the noise points, which are often quite distant from the clusters, can drastically influence the estimated prototype point.

Fifth, cluster validity is "built in", since when the "primary" clusters are selected, data on the other clusters are available, and can be checked if required.

Finally, the framework can be enlarged to get information concerning the history of the points stabilization to decide on the number of clusters in the data.

5. ACKNOWLEDGMENTS

The authors thank Amir Geva for helpful discussions concerning fuzzy clustering.

6. REFERENCES

- [1] Forsythe G. *Finite Difference Methods for Partial Differential Equations*, J. Wiley and Sons, New York, 1960.
- [2] Hildebrand F. *Finite Difference Equations and Simulations*, Prentice Hall, 1968.
- [3] Krishnapuram R. and Keller J. M. A possibilistic approach to clustering, *IEEE Trans. on Fuzzy Systems*, Vol. 1, No. 2, May 1993, pp. 98–110.
- [4] Yadid-Pecht O. and Gur M. A biologically inspired modified Maxnet, submitted.