A New Clustering Approach on the Basis of Dynamical Neural Field

Dequan Jin  
dqjin@yahoo.cn

Jigen Peng  
jgpeng@mail.xjtu.edu.cn

Department of Applied Mathematics, School of Science, Xi’an Jiaotong University, Xi’an 710049, China

Bin Li  
bin-li@uiowa.edu

Applied Mathematical and Computational Sciences, University of Iowa, Iowa City, IA 52242, U.S.A.

In this letter, we present a new hierarchical clustering approach based on the evolutionary process of Amari’s dynamical neural field model. Dynamical neural field theory provides a theoretical framework macroscopically describing the activity of neuron ensemble. Based on it, our clustering approach is essentially close to the neurophysiological nature of perception. It is also computationally stable, insensitive to noise, flexible, and tractable for data with complex structure. Some examples are given to show the feasibility.

1 Introduction

Data mining is a major procedure of knowledge discovery, aiming at extracting useful knowledge from massive amounts of data. Since computer and Internet techniques change rapidly, large amounts of information are represented and stored as data, leading to difficulty in extracting useful knowledge. Classifying or grouping data into different categories or clusters is required in these problems. It is called data clustering if such a process is executed without prior knowledge or supervision. Clustering is targeted at searching for ways to classify a data set into subsets according to similarity without labeling the data. Therefore, clustering algorithms should be able to explore and separate data structures (Xu & Wunsch, 2005).

Hierarchical clustering is an important clustering technique. Data usually show different structures if they are observed at different scales. Hierarchical clustering works by grouping data objects into a tree of clusters, with each clustering a partition of data. It creates a hierarchical decomposition that preserves the diversity of structures under different scales. One
class of hierarchical clustering, agglomerative clustering, is a successive merging process of clusters and can be classified into two types: nested and nontested. Nested hierarchical clustering keeps data in the same cluster derived at small scales together. Otherwise, it is nontested (Leung, Zhang, & Xu, 2000; Xu & Wunsch, 2005; Yip, Ding, & Chan, 2006; Han & Kamber, 2006; Hastie, Tibshirani, & Friedman, 2009).

Clustering is one of the most primitive human activities. Many clustering algorithms are presented by simulating the structures or mechanisms of the human perceptual system, such as the clustering algorithms based on Gestalt psychology, scale space theory, and neural networks. Originated in the late nineteenth century, Gestalt psychology is mainly about the formation and organization processes of human perception (Sternberg, 2003). Using Gestalt psychology, Zahn (1971) presented a graphical clustering algorithm with a minimal spanning tree. It has a significant influence on clustering, and many algorithms have been presented with a similar idea.

Motivated by phenomena derived from the investigation of biological vision and biophysics, scale space theory models the blurred effect of lateral interaction on the retina. Considering a data set as an image and each datum as a light point, Leung et al. (2000) presented a clustering approach on the basis of scale space theory by applying gaussian filtering.

Neural networks are popular models for reproducing neural system structures. Anderson, Gately, Penz, and Collins (1990) used an energy-minimizing network (the brain-state-in-a-box, BSB, model) for data clustering. Energy minima (i.e., attractors of dynamical neural networks) are formed by a generalized Hebb rule. Data that converge to the same attractor are considered to be in one cluster. The number of clusters that the network supports depends on its storage capacity, which Hopfield (1982, 1984) has investigated.

This letter develops a new hierarchical clustering approach based on the evolutionary process of Amari’s neural field. Our approach is biologically reasonable, highly consistent with human perception, robust to noise, computationally stable, and feasible.

The letter is organized as follows. In section 2, we briefly introduce Amari’s neural field theory and in section 3 establish a new hierarchical clustering approach on the basis of Amari’s model. The kernel and parameter selections are discussed in section 4. To show the feasibility of our approach, we provide some numerical examples in section 5. We conclude in section 6.

2 Dynamical Neural Field Theory

In the 1970s, theories on describing large-scale activation dynamics of cortical neurons as a continuous neural field were developed (Wilson & Cowan, 1972, 1973; Feldman & Cowan, 1975a, 1975b; Amari, 1977; Ermentrout & Cowan, 1979; Kishimoto & Amari, 1979). Amari’s (1977) neural field
model is one of many significant developments. Providing a theoretical framework that successfully interprets large amounts of important phenomena and problems of psychophysics, perception, and cognition (Giese, 1999), Amari’s model is mathematical analyzable and biologically reasonable. Therefore, the dynamical properties of his model have been extensively investigated (Kishimoto & Amari, 1979; Ermentrout & Cowan, 1979; Werner & Richter, 2001; Wennekers, 2002; Taylor, 1999, 2003; Laing & Troy, 2003; Owen, Laing, & Coombes, 2007; Kubota, Hamaguchi, & Aihara, 2009) and applied to motor planning, cognition problems, and autonomous robot path planning (Engels & Schöner, 1995; Schöner, Dose, & Engels, 1995; Erhagen & Bicho, 2006; Faubel & Schöner, 2008; Simmering, Schuttea, & Spencer, 2008; Johnson, Spencer, & Schöner, 2008, 2009).

In Amari’s model, a temporally and spatially continuous neural activation function \( u(z, t) \) is supposed to describe the dynamics of the activation of cortical neurons. \( z \in \Omega \) called behavior parameter. \( \Omega \) is called perceive space. In fact, the coordinate \( z \) mostly represents relevant perceptual parameters such as feature dimensions or movement parameters rather than the spatial coordinates over cortex.

Amari’s neural field is usually described by

\[
\tau \dot{u}(z, t) = -u(z, t) + \int_{\Omega} w(z, z')\theta(u(z', t)) \, dz' + s(z, t) - h, \tag{2.1}
\]

where \( \tau \) is a positive time constant, \( h \) is the resting level of the neural field, and \( s(z, t) \) is the input signal distribution. The region

\[
\{z \in \Omega : u(z, t) > 0\}
\]

is called the excited region. \( \theta(u) \) is a nonlinear monotone increasing threshold function, satisfying \( \lim_{u \to -\infty} \theta(u) = 0 \) and \( \lim_{u \to +\infty} \theta(u) = 1 \). It describes the neural field feedback of each excited point to its neighboring positions in \( \Omega \) with an interaction strength determined by interaction function \( w(z, z') \).

The lateral interaction is introduced by the integration term

\[
\int_{\Omega} w(z, z')\theta(u(z', t)) \, dz'
\]

on the right side of equation 2.1.

Amari’s neural field is usually assumed to be homogeneous. As a result, the interaction function \( w(z, z') \) can be written as \( w(z - z') \). Then \( w(z) \) is also called the interaction kernel. Approximating the lateral interaction of neurons, the lateral interaction of neural field is usually assumed to be locally excitatory and globally inhibitory (Amari, 1977). Typical interaction kernels are:
- Gaussian function with constant inhibitory interaction, given by
  \[ w(z) = g(z, \sigma) - H, \]  
  (2.2)
  inducing global inhibition
- The difference of gaussian (DoG) functions, given by
  \[ w(z) = Ag(z, \sigma) - Bg(z, \gamma \sigma), \]  
  where \( \gamma > 1 \)
- Rectangular Mexican hat shape kernel, given by
  \[
  w(z) = \begin{cases} 
  a, & \|z\| \leq r \\
  -a, & r < \|z\| \leq R \\
  0, & \text{else}
  \end{cases}
  \]  
  (2.4)

\( g(z, \sigma) \) is a gaussian function given by
\[ g(z, \sigma) = \exp\left(-\frac{\|z\|^2}{\sigma^2}\right). \]
(2.5)

Assuming \( s(z, t) \) to be time invariant that \( s(z, t) = s(z) \), the discrete description of Amari’s model 2.1 is given by
\[
\tau \dot{u}_i(t) = -u_i(t) + \sum_{j=1}^{n} w_{i,j} \theta(u_i(t)) + s_i - h,
\]  
(2.6)

which looks similar to Hopfield-type neural networks (Hopfield, 1982, 1984). However, they are essentially different. One major difference is in storing data. For neural networks like Hopfield type, patterns are stored by their attractors formed by training neural networks with learning rules. Data converging to the same attractor are considered to be in the same class. The number of patterns that a network is able to support depends on its storage capacity. However, for Amari’s model, patterns are stored by excited regions of neural field. Data in the same connected excited region are considered to be in the same cluster (Engels & Schöner, 1995; Schöner et al., 1995; Erhagen & Bicho, 2006; Faubel & Schöner, 2008; Simmering et al., 2008; Johnson et al., 2008, 2009). The number of attractors in a Hopfield network is always finite, while the number of attractors in a neural field is infinite because it has neutrally stable line (field) attractors. Attractors in a Hopfield network should be designed before input data are given, while attractors in a neural field are self-organized by input data. In other words, a bubble solution (local excitation) of a neural field is neutrally stable, having an infinite number of local excitation solutions (line attractor or field attractor).
Due to the input data, there can exist a number of local excitations, showing clusters existing in the data.

Another important difference is that in Amari’s model, the activation function \( u(z, t) \) represents the corresponding neural activation in \( \Omega \). Each point of \( \Omega \) corresponds to a neuron. Points in the input space of Hopfield-type neural networks do not have such correspondence to neurons. Moreover, in Hopfield-type neural networks, the synaptic interconnection strength \( w_{i,j} \) from neuron \( j \) to neuron \( i \) is determined by learning rules (Anderson et al., 1990), while \( w(z, z') \) is obtained by simulating the lateral interaction of neurons without any learning rule in Amari’s model.

There are three important types of stationary solution of Amari’s model 2.1: \( \phi \)-solution, the bubble solution, and the \( \infty \)-solution:

- A stationary solution \( u^*(z) \) is called a \( \phi \)-solution if \( u^*(z) \leq 0 \) for all \( z \in \Omega \).
- A stationary solution \( u^*(z) \) is called a bubble solution if \( u^*(z) > 0 \) in a bounded region of \( \Omega \).
- A stationary solution \( u^*(z) \) is called an \( \infty \)-solution if \( u^*(z) > 0 \) for all \( z \in \Omega \).

In the absence of external input, some results on these solutions are given for two-dimensional Euclid space (Taylor, 1999; Werner & Richter, 2001). Suppose \( R > 0 \). Let

\[
D = \{ z \in \Omega : \| z \| \leq R \}
\]

and

\[
W(\| z \|, R) = \int_D w(x - x') dx'. \tag{2.7}
\]

When \( \| z \| = R \), we have \( W(\| z \|, R) = W(R, R) \). Let

\[
G(R) = W(R, R) = \int_D w(z - z') dz'. \tag{2.8}
\]

Define \( G_\infty = \lim_{R \to \infty} G(R) \) and \( G_{\text{max}} = \max_{R \geq 0} G(R) \). When the interaction kernel \( w(z) \) is selected as a Mexican hat shape that \( G(R) \) has two monotone intervals, the following results are obtained (Taylor, 1999):

**Theorem 1.** Suppose \( s(z, t) = 0 \). Then

1. There exists a \( \phi \)-solution if and only if \( h > 0 \).
2. There exists an \( \infty \)-solution if and only if \( G_\infty > h \).
3. There exists a bubble solution if and only if \( h > 0 \) and \( R > 0 \) satisfies

\[
G(R) - h = 0. \tag{2.9}
\]
Theorem 2. The stability of the various solutions for different parameters is:

1. $G_{\text{max}} < h$: $\phi$-solution is monostable.
2. $0 < G_\infty < h < G_{\text{max}}$ or $G_\infty < 0 < h < G_{\text{max}}$: $\phi$-solution and a bubble solution are bistable when there exists another unstable bubble solution.
3. $G_\infty > h$: $\infty$-solution is stable.

For data clustering, many cases are high-dimensional. There are few approaches to dynamical properties of neural fields whose dimension is higher than 2. However, generally the dynamical properties of neural field should be qualitatively carried out.

3 Hierarchical Clustering Approach on the Basis of the Evolutionary Process of Dynamical Neural Field

Suppose that $X = \{x_i \in \mathbb{R}^n : i = 1, 2, \ldots, N\}$ is a given data set. As did Leung et al. (2000), we obtain the data distribution $I(z)$ by

$$I(z) = \frac{1}{N} \sum_{i=1}^{N} \delta(z - x_i), \quad (3.1)$$

where

$$\delta(z) = \lim_{\sigma \to 0} g(z, \sigma). \quad (3.2)$$

g(z, \sigma) is a gaussian function defined by equation 2.5. $I(z)$ contains $N$ isolated blobs corresponding to $x_i \in X, i = i, 2, \ldots, N$, respectively.

The data distribution $I(z)$ is transformed into the neural input signal $s(z)$ in the following way:

$$s(z) = \int_{\Omega} I(z - z', t) g(z', \sigma_{\text{input}}) d\Omega \quad (3.3)$$

where $\sigma_{\text{input}} > 0$ is small.

Consider the neural field given by

$$\tau u(z, t) = -u(z, t) + \int_{\Omega} w(z - z')\theta(u(z', t)) d\Omega' + s(z) - h. \quad (3.4)$$

The input $s(z)$ introduces activation peaks to the neural field. When the neural field evolves, these peaks continuously merge until the field reaches its steady state. Several neural activation peaks may be sustained, and the number and range depend on both the input $s(z)$ and the neural field.
Grouping the entire element \( x_i \in X \) in the same connected excited region into a cluster, we can obtain clustering.

To generate hierarchical clustering results on Amari’s model, an intuitive way is to gradually increase the scale \( \sigma \) of \( w(z) \). However, there are some problems. First, the clustering results are very sensitive to the parameter \( \sigma \). As shown in Figure 1, a very small perturbation of \( \sigma \) may induce a large change in clustering results. Second, as Taylor (1999) shows, obtaining a stable bubble solution has some requirements for the neural field. This indicates that some \( \sigma \) may not be able to generate a stable bubble solution when other parameters are fixed. Therefore, we have to construct a neural field that can sustain a bubble solution for \( \sigma \) in a wide interval, as well as balance the range and strength of lateral interaction at the same time. All of these make the parameter selection very complicated. Third, it is computationally costly. In fact, to obtain a hierarchical clustering result in this way, we have to solve a neural field equation numerically over again for any new \( \sigma \), which costs too much time.

To avoid these difficulties in obtaining a proper bubble solution, we develop a new approach. Notice that the process of the activation in the neural field evolves from its initial state to steady state. The initial positive
activation is induced from the input distribution \( s(z) \). As soon as the evolution starts, the positive activation in neural field diffuses. Proximate peaks begin to merge and compete with those in their inhibitory range until the neural field reaches its steady state. Assume the input scale is small enough to isolate every datum. If we choose a kernel whose scale is large enough, the evolutionary process is then a continuous merging process, and all data would merge into one excited region eventually. Therefore, for any given \( t \geq 0 \), by grouping data in the same excited region into a cluster, a clustering is obtained. These clusters continuously merge as \( t \) increases. Finally, a family of clusterings is obtained.

On the basis of this idea, we present a nested hierarchical clustering algorithm as follows.

1. Let \( \sigma_{\text{input}} > 0 \) be small to ensure that every blob in \( s(z) \) contains only one datum in \( X \).
2. Choose a large-scale \( \sigma \) and time constant \( \tau > 0 \). Choose an increasing time sequence \( \{t_1, t_2, \ldots\} \). Let \( i = 1 \).
3. Start the evolutionary process of neural field 3.4, and compute the corresponding state \( u(z, t_i) \). If there are \( p \) excited regions, group all the data locating in the same excited region into one cluster \( C^j_i, j = 1, 2, \ldots, p \). If there are unlabeled data, go to step 4; else, go to step 5.
4. If the unlabeled data belong to a positive activation peak, group them with the cluster whose corresponding connected region contains this peak; else, group the unlabeled data in the same negative activation peak into a new cluster \( C^p_i + k, k = 1, 2, \ldots, m \). In this way, every datum is labeled. Let \( p = p + m \). Go to step 5.
5. Let \( Sc_i = \{C^i_j\}_{j=1}^p \) and \( Sc_i \) is the clustering result at \( t_i \). If \( p > 1 \), let \( i = i + 1 \) and go to step 3; else, let \( n = i \) and go to step 6.
6. Let \( Sc_h = \{Sc_i\}_{i=1}^n \). Then \( Sc_h \) gives a hierarchical clustering result.

By this algorithm, we can obtain a hierarchical clustering sequence. A problem is how to find the clustering closest to the data structure in this sequence. To measure the goodness of the cluster and clustering, we give the following definitions:

**Definition 1.** The lifetime of a cluster is the length of the time interval in which the cluster survives in the evolutionary process of the neural field.

**Definition 2.** The lifetime of a clustering is the length of the time interval in which the clustering survives in the evolutionary process of the neural field, which equals the shortest lifetime of clusters in it.

Since a real cluster should be more stable in the evolutionary process, a cluster with a longer lifetime is preferred to a cluster with a shorter lifetime, and a clustering with a longer lifetime is also preferred to a clustering with a shorter lifetime.
Because a cluster changes as soon as new points merge into it, the lifetime of clustering may be very short when there is massive noise. Therefore, the above strategy is less efficient for data with noise. To reduce the impact of noise, we give another definition of the lifetime of clustering, called $T$-lifetime:

**Definition 3.** Let $\pi(t)$ be the number of excited regions of a neural field at evolutionary time $t$. Suppose $Sc_t$ is a clustering obtained at $t$ with $\pi(t) = m$. The $T$-lifetime of $Sc_t$ is the maximum length $T_m$ of the time interval in which $\pi(t) = m$.

The $T$-lifetime of clustering depends on the number of excited regions. Since most noise is suppressed by the lateral inhibitory effect of a neural field and therefore cannot generate positive peaks, $T$-lifetime is less sensitive to noise and therefore more suitable for data with noise. Similarly, a clustering with a longer $T$-lifetime is also preferred to a clustering with a shorter $T$-lifetime.

There is no direct relationship between the lifetime and dissimilarity of the clusters. Generally, clusters with a small dissimilarity may have a similar lifetime, but the opposite is not true: clusters with a similar lifetime may not have a small dissimilarity.

### 4 Kernel and Parameter Selection

In our approach, $w(z)$ is a DoG function given by

$$w(z) = A g(z, \sigma) - B g(z, \gamma \sigma),$$

providing enough room to coordinate the lateral interaction of neural field. $\sigma$, $\lambda$, $A$, $B$, and $h$ are important parameters. $\sigma$ determines the lateral excitatory range and the diffusing speed of positive activation. When $\sigma$ is small, the lateral excitatory range is small, inducing a small, excited region. When $\sigma$ increases, the lateral excitatory range also increases, inducing a larger excited region. $\gamma$ cooperates with $\sigma$, determining the range of the lateral suppressing effect. Increasing $\gamma$ induces a larger suppressing range of the neural field, which is important when there is noise. However, if there coexist small and large clusters, small clusters may be suppressed by large ones if $\gamma$ is too large. $A$ and $B$ determine the lateral excitatory and inhibitory strength, where $A - B$ should be positive. If $A - B$ is large, the lateral excitatory strength is strong, which also induces a lateral excitatory range. $B$ decides the lateral inhibitory strength inducing by kernel. Large $B$ induces a large suppressing effect. Different from the suppressing effect induced by $w(z)$, resting level $h$ determines the global inhibitory level of neural field.
As shown in theorem 1, Taylor (1999) proves that in the absence of input $s(z)$, a bubble solution exists if and only if there exist $R$ and $h$ satisfying equation 2.9:

$$G(R) - h = 0,$$  \hspace{1cm} (4.2)

with two solutions $R_1 > R_2 > 0$. $R_1$ has been proved to be the radius of a stable bubble solution; the neural field is able to sustain a positive activation bubble with radius $R_1$, and $R_2$ is the radius of an unstable bubble solution. However, when there are two or more bubbles, these bubbles may interact with each other and merge into larger ones if they are proximate or suppress each other if they are far away.

For instance, there is a gaussian distribution with covariance matrix

$$
\begin{pmatrix}
0.2 & 0 \\
0 & 0.2
\end{pmatrix}
$$

containing 1000 points in Figure 2. Let $\sigma = 0.12$, $\gamma = 50$, $A = 1.2$, $B = 0.01$, and $h = 0.05$. By solving equation 2.9, we obtain two solutions, and the large radius $R_1$ is about 1.38. As shown in Figure 2, the actual size of the
resulting cluster induced by stationary solution $u^*(z)$ is very close to its theoretical value.

With the same parameters, when there are two similar gaussian distributions whose distance of their centers is about 2.2, as shown in Figure 3, we obtain two bubbles. Since they locate in the inhibitory regions of each other, both of them are smaller than the one shown in Figure 2. It is more significant when there are three gaussian distributions, as shown in Figure 4, where the resulting clusters are even smaller than those in Figure 3.

As for most applications of Amari’s model, a stable bubble solution is preferred; the $\infty$-solution is usually supposed to be ill posed and should be avoided. Therefore, the bubble solution is also preferred in our approach. However, our approach is based on the evolutionary process of the neural field instead of the steady state. As a result, the $\infty$-solution and monostable $\phi$-solution are also compatible with our approach. In fact, the only case that we should avoid is a too strong lateral inhibitory effect, which makes it difficult to activate the neural field.

5 Numeric Illustration

To show the feasibility and efficiency of our approach, we give several numeric examples on 3-gaussian (without and with noise), double-C (without
and with noise), double-sine (with noise), and double-ring (with noise) data sets. Figures 5 through 10 are the clustering results by our approach.

In Figure 5, the thick dashed lines are the boundaries of the clusters derived by $u(z, 0.035)$. The thin solid line is the boundary of clusters derived by $u(z, 0.1)$. Our approach succeeds in distinguishing all three gaussian distributions when $t = 0.035$. When $t = 0.1$, almost all data are in the same cluster. When there is noise, as shown in Figure 6, our approach is also able to derive good clustering results. The three gaussian distributions are indicated by a thin solid line at $t = 0.04$. The main data are grouped into one cluster at $t = 0.1$, keeping most of noise out, with the same parameters.

In Figure 7, thick dashed lines are the boundaries of clusters derived by $u(z, 0.03)$. The thin solid line is a boundary of the cluster derived by $u(z, 0.2)$. We can see that the two arcs are indicated at $t = 0.03$, which are grouped into one cluster at $t = 0.2$. When there is noise, similar clustering results can be obtained at $t = 0.04$ and $t = 0.2$ with the same parameters as shown in Figure 8.

In Figures 9 and 10, thick lines are the boundaries of the clusters obtained by our approach for double-sine and double-ring data sets. As shown in these figures, our approach is able to discover underlying clusters in data.
Figure 5: Boundaries of clusters derived by $u(z, 0.035)$ (thick dashed lines) and $u(z, 0.1)$ (thin solid line) where $\tau = 0.1, \sigma = 0.35, \gamma = 2, A = 1, B = 0.2, h = 0.16$.

Figure 6: Boundaries of clusters derived by $u(z, 0.04)$ (thick dashed lines) and $u(z, 0.1)$ (thin solid line) where $\tau = 0.1, \sigma = 0.35, \gamma = 2, A = 1, B = 0.2, h = 0.16$. 
Figure 7: Boundaries of clusters derived by \( u(z, 0.03) \) (thick dashed lines) and \( u(z, 0.2) \) (thin solid line) where \( \tau = 0.1, \sigma = 0.17, \gamma = 20, A = 1.2, B = 0.01, h = 0.08 \).

Figure 8: Boundaries of clusters derived by \( u(z, 0.04) \) (thick dashed lines) and \( u(z, 0.2) \) (thin solid line), where \( \tau = 0.1, \sigma = 0.17, \gamma = 20, A = 1.2, B = 0.01, h = 0.08 \).
Figure 9: Boundaries of clusters derived by $u(z, 0.03)$ (thick lines), where $\tau = 0.1, \sigma = 0.2, \gamma = 50, A = 1.2, B = 0.01, h = 0.001$.

Figure 10: Boundaries of the clusters derived by $u(z, 0.05)$ (thick lines), where $\tau = 0.1, \sigma = 0.2, \gamma = 50, A = 1.2, B = 0.01, h = 0.001$. 
The input scale is selected as $\sigma_{\text{input}} = 0.01$, which is not small enough for our algorithm because the data are too dense. If we want to generate sufficiently small neural peaks that isolate every datum, the computation of evolutionary process would become unacceptable. Moreover, the main purpose here is to show that our approach is able to get proper clusterings at certain scales, so we do not care too much about the clusterings at small scales. Such an input scale $\sigma_{\text{input}}$ is tolerant.

For comparison, we employ other clustering methods, including the K-means method (Hastie et al., 2009), the BSB model (with 400 neurons, $\alpha = 0.5$, $\gamma = 0.9$, $\delta = 0$, and the limits for thresholds 2 and $-2$) (Anderson et al., 1990), the gaussian density-based clustering method (Han & Kamber, 2006) with some of the above data sets. Figures 11 and 12 show the clustering results obtained by the K-means method and the BSB model, for double-C without noise. Figures 13 to 15 show the clustering results obtained by the K-means algorithm, the BSB model, and the gaussian density-based clustering method ($\sigma = 5 \times 10^{-3}$) for double-C with noise, respectively. Figures 16 to 18 show the clustering results obtained by the K-means algorithm, BSB model, and gaussian density-based clustering method ($\sigma = 8 \times 10^{-3}$) for the double sine data set with noise, respectively. Figures 19 to 21 show the clustering results obtained by the K-means algorithm, the BSB model and
Figure 12: Clustering obtained by BSB model. Two clusters are obtained and denoted by · and ×.

Figure 13: Clustering obtained by K-means method. Two clusters are obtained and denoted by · and ×.
Figure 14: Clustering obtained by the BSB model. Two clusters are obtained and denoted by ⋅ and ×.

Figure 15: Clustering obtained by a gaussian density-based clustering method with $\sigma = 5 \times 10^{-3}$. 
Figure 16: Clustering obtained by K-means method. Two clusters are obtained and denoted · and ×.

Figure 17: Clustering obtained by BSB model. Two clusters are obtained and denoted by · and ×.
Figure 18: Clustering obtained by gaussian density-based clustering method with $\sigma = 8 \times 10^{-3}$.

Figure 19: Clustering obtained by K-means method. Two clusters are obtained and denoted by · and ×.
Figure 20: Clustering obtained by the BSB model. Two clusters are obtained and denoted by · and ×.

Figure 21: Clustering obtained by gaussian density-based clustering method with $\sigma = 8 \times 10^{-3}$. 
the gaussian density-based clustering method ($\sigma = 8 \times 10^{-3}$) for “double-ring” with noises, respectively. We see that our approach obtains better results than these clustering methods.

The K-means algorithm partitions a data set into $k$ clusters by minimizing the squared error function according to data similarity, which is measured by the mean values of the objects in the clusters (Hastie et al., 2009). The K-means algorithm works well when clusters are compact and convex. However, it, like other partitioning methods like K-medoids and SOM, requires users to specify the number of clusters. Moreover, these clustering approaches are not effective for discovering clusters with nonconvex shapes (Han & Kamber, 2006), as shown in Figures 11, 15, and 19.

Clustering methods using neural networks like the BSB model and the Hopfield type are based on an energy-minimizing strategy (Anderson et al., 1990). Data converging to the same attractor of neural network (i.e., locating in the same domain of attraction) are grouped into one cluster. Therefore, the clustering capability of neural networks depends on their attractors and corresponding domain of attractions, which are usually obtained by training neural networks with learning rules like Hebb. Theoretically, neural networks are able to discover arbitrarily shaped clusters. However, since it is not easy to train a dynamical neural network to possess a complex-shape domain of attraction in practice, neural networks like the BSB model and Hopfield type are not suitable for some special data sets, as shown in Figures 14, 17, and 20, which present the clustering results obtained by the BSB model.

Gaussian density-based clustering method is suitable for discovering arbitrarily shaped clusters. But when there exists noise, as shown in Figures 15, 18, and 21, the gaussian density-based clustering method requires additional strategy to identify noise and outliers (Leung et al., 2000; Han & Kamber, 2006). In our approach, noise also generates activation peaks in neural field. However, the peaks are suppressed by the excited regions corresponding to real clusters. Therefore, most of the peaks generated by noise are negative, as shown in Figures 6, 8, 9, and 10, which are much easier to identify.

Nearest-neighbor clustering is an important technique. It can be found in various forms in Hastie et al. (2009) and is suitable for discovering arbitrarily shaped clusters. To show the difference between our approach and nearest-neighbor clustering, another example is given (see Figure 22). In this example, two isolated gaussian distributions are connected by a band of points. The points in the band are so dense that nearest-neighbor clustering cannot group the two gaussian distributions into two different clusters. In our approach, the two gaussian distributions can be grouped into two different clusters, as shown in Figure 22.

When all the parameters, including the iterative step length in solving the neural field equation, are fixed, our approach can generate a unique dendrogram. An example is given in Figure 23, which portrays the dendrogram
Figure 22: Boundaries of clusters derived by $u(z, 0.2)$ (thick lines), where $\tau = 0.1$, $\sigma = 0.12$, $\gamma = 50$, $A = 1.2$, $B = 0.01$, $h = 0.05$.

Figure 23: Evolutionary tree (dendrogram) of hierarchical clustering obtained by our approach, where $\tau = 0.1$, $\sigma = 0.35$, $\gamma = 2$, $b = 0.1$, $h = 0.16$. 
obtained by our approach. We can see that as the evolutionary time $t$ increases, the small clusters merge into larger one. By adding some noise to this data set, the dendrogram obtained by our approach is portrayed in Figure 24. We can see that the dendrogram is not disturbed much by this noise. As shown in Figures 23 and 24, clusters in the same gaussian distribution have a small dissimilarity and similar lifetimes. However, the noise at different corners has similar lifetimes but large dissimilarity.

6 Conclusion

In this letter, we propose a hierarchical clustering approach on the basis of the evolutionary process of Amari’s dynamical neural field model. Since Amari’s dynamical neural field theory gives a theoretical framework that is essentially close to the neurophysiological nature of perception, our approach is computationally stable, biologically reasonable, highly consistent with human perception, and significantly robust to noise. In our approach, distinct from clustering methods based on neural networks like the BSB model (Anderson et al., 1990) and the Hopfield type, clusters are determined by the excited regions in the activation distribution of the neural field. The activation distribution is derived by a self-formation process of the neural field without any learning step. Since the excited regions can
be of arbitrary shape, our approach is suitable for discovering arbitrarily shaped clusters. Therefore, our approach is able to obtain good clustering results for various data sets as shown in the examples.

Nevertheless, our proposed approach is an attempt. Since Amari’s model contains a convolution on its right-hand side, it requires much computation. Therefore, though our approach is theoretically suitable for high-dimensional and mass data, employing dimensional-reduction techniques and feature extraction approaches can significantly add to computational efficiency.

Acknowledgments

We are grateful to the reviewers for their valuable and profound comments. This work is supported by the National Basic Research Program of China (2007CB311002) and by NCET.

References


Received March 31, 2010; accepted January 15, 2011.