2004 International Symposium on Nonlinear
Theory and its Applications (NOLTA2004)
Fukuoka, Japan, Nov. 29 - Dec. 3, 2004

The Stochastic Network Approach to Clustering

Thomas Ott[†], Willi-Hans Steeb[‡] and Ruedi Stoop[†]

[†]Institute of Neuroinformatics, ETH/UNIZH Zurich, Switzerland
[‡]Department of Applied Mathematics, RAU University Johannesburg, South Africa
Email: tott@ini.phys.ethz.ch, ruedi@ini.phys.ethz.ch

Abstract—We outline a general framework for the systematic and consistent description of clustering methods. Clustering is considered as a self-organized process, exhibited by stochastic neural networks in the ”thermal” equilibrium. Classes of similar data items are found by identifying groups of preferentially synchronized neurons. As illustrations of our clustering framework, four clustering methods are considered in detail.

1. Introduction

Clustering is one of the most fundamental information processing techniques for dealing with large amounts of data. Clustering provides a compact representation of information, and may thus be considered a prerequisite for cognitive skills, such as pattern recognition and effective communication. In the human cognition paradigm, some classification skills clearly rely on unsupervised clustering. Based on similarities among the items, we are able to group unfamiliar items without any external supervision. To mimic such unsupervised processes, it appears natural to use self-organizing systems. In this approach, clusters appear as emergent spatio-temporal characteristics exhibited by the whole system, although they result from the interactions among the system components. Also in artificial neural networks, groups of stochastic neurons that preferentially express the same firing state, form natural clusters. In this contribution, we therefore exploit the ansatz that the identification of groups of *synchronized* neurons as a cluster, provides us with the most general paradigm of clustering. Consequently, a variety of different clustering methods, inspired from magnetic systems, graph theory and optimization problems, will be reinterpreted from the neural network point of view. These methods will be treated as specific models of networks. This not only yields a consistent terminology, it also allows for a straightforward comparison among the methods and provides a theoretical basis for the introduction of novel and optimized methods.

We will start with some mathematical prerequisites needed for our approach. Then our general framework will be worked out. As an example, a novel method (called Hopfield clustering) will be derived and applied to a two-dimensional clustering problem. Finally, we will give a reinterpretation of three popular clustering methods within the presented framework.

2. The network framework for clustering

Let a set of n items be described by (usually high-dimensional) feature vectors x_i, $i = 1 \dots n$. Using an appropriate similarity measure (e.g., Euclidian distance), we can calculate values d_{ij} that express the similarity among pairs of items $\{i, j\}$. The goal of clustering is to partition the items into k different groups, such that the groups form in a sense natural entities of similar items. The number of groups k is a priori unknown, it emerges during the self-organization process of the network.

In the neural network paradigm, each neuron of the network represents exactly one data item. The synaptic weights generally depend on the values d_{ij} in a nonlinear way. The discrete time dynamics is governed by a, generally stochastic, update rule. The *architecture of the network* therefore comprises the update rules that, given the initial conditions, determine the synaptic weights. In this way, each clustering method is described by one particular network architecture. Below, we outline how to identify clusters of preferentially synchronized neurons, and develop general criteria how the different clustering approaches translate into specific update rules. Our terminology will be the following: **neurons** occupy a finite set Q of m possible firing states, i.e., $s_i \in Q = \{q^1, ..., q^m\}$, where s_i is the actual state of neuron i. If $s_i = s_j$, the two neurons i and j are called **synchronized**. $\mathbf{s}(t) \in S$ denotes the system's firing configuration at time t, where $S = Q^n$. By **structure** we will understand the array of synaptic weights $\mathbf{j} = (J_{12}, ..., J_{n-1n}) \in J$, where J is the space of all potential synaptic structures. For explicitly time or configuration dependent network structures, we will explicitly write $\mathbf{j} = \mathbf{j}(\mathbf{s}(t), t)$. The current state of the network, embracing firing configuration and synaptic structure, will be denoted by $\sigma(t) = (\mathbf{s}(t), \mathbf{j}(t)) \in \Sigma = S \times J$.

In general, the update rules determine the current transition probabilities P_t, i.e.,

$$\sigma(t{+}1) = \sigma' \text{ with probability } P_t(\sigma', \sigma(t), \sigma(t{-}1), ...). \quad (1)$$

During this process, in most models only the firing configuration $\mathbf{s}$ is changed. For some models, however, also the structure of the network $\mathbf{j}$ changes. We will refer to a structure change as *learning*. Update rules that change (s_i or J_{ij}) only due to local information, will be called *biologically suggestive*. However, update rules that rely on properties exhibited by the whole network, will be considered as well. Clusters of preferentially synchronized neurons are generally identifiable only after a long enough observation time M. Technically, for their identification we proceed as follows: For two neurons i and j, consider the *empiric* pair correlation function $G(i, j)$,

$$G(i, j) = \frac{1}{M} \sum_{t=1}^{M} \Pi(t) \delta_{s_i s_j}(t), \tag{2}$$

where $\Pi(t)$ is a step weighting function whose specific role will be discussed towards the end of the section. Let G denote the relation $\{\{i, j\} | G(i, j) > \Theta = 0.5\}$. Two neurons i, j then belong to a cluster C if $\exists\ k_1, ..., k_l$: $\{i, k_1\}, \{k_1, k_2\}, .., \{k_l, j\} \in G$, where Θ is a fixed threshold value (G-criterion). In order to obtain meaningful clustering solutions, the network dynamics should be chosen such that more "similar" neurons more often share the same state. Therefore, we assume that architectures come equipped with time independent functions H (called a *cost* function), that map a state σ into $\mathbb{R}$:

$$\sigma \in \Sigma \mapsto H(\sigma) = H_{\mathbf{j}}(\mathbf{s}) \in \mathbb{R}. \tag{3}$$

The idea is that $H_{\mathbf{j}}(\mathbf{s})$ shall be lowest for configurations that have many neurons (corresponding to similar items) in the same state.

We first consider the case of a **fixed structure** ($\mathbf{j}(\mathbf{s}, t) =$ *const*). Then the costs determine a one-parameter family of Gibbs distributions with parameter $T \in [0, \infty]$, i.e.,

$$p(H_{\mathbf{j}}(\mathbf{s}), T) = p_{\mathbf{j},T} = \frac{1}{Z_j(T)} e^{-H_{\mathbf{j}}(\mathbf{s})/T}, \tag{4}$$

with $\sum_{\mathbf{s}} p(H_{\mathbf{j}}(\mathbf{s}), T) = 1$. For networks with fixed structure, we demand that (4) determines the probability of a firing configuration $\mathbf{s}$. In this way, we come up with the following *criterion* for the update process: By an *update process* we shall understand a **Markov process** on the configuration space S with (4) as **steady state distribution**. In particular, this means that the empirical pair correlation (2) approximates $G(i, j) \approx \sum_s p(H_{\mathbf{j}}(\mathbf{s}), T) \delta_{s_i, s_j}$. For most clustering methods, $H_{\mathbf{j}}(\mathbf{s})$ is explicitly known and (1) can be derived from $p_{\mathbf{j},T}$. Sometimes, however, (1) is given and $H_{\mathbf{j}}(\mathbf{s})$ has to (and can) be constructed.

The Gibbs distribution is a natural choice from the network, as well as from the clustering point of view: (**I**) It is a natural choice from the network point of view, as it is the equilibrium distribution of the canonical ensemble. T can be interpreted as a (thermal) noise or fluctuation parameter. The system is in equilibrium with its environment according to a given noise level. Naturally, $p(\mathbf{j}, T)$ is smaller for higher costs and vice versa. (**II**) From the clustering point of view: For $T \to 0$, only the configuration(s) with the lowest cost survives. Thus, cost optimization methods are the $T = 0$ border cases and the update rule (1) can be any appropriate optimization procedure. Often, H is minimal if all neurons are in the same firing state (ferromagnetic models). In this case, T can be interpreted as a resolution (or attention) parameter which determines, how strongly similarities among items are weighted. For $T \to 0$, the resolution of the system is low, i.e., differences between items are smoothed out and all items are clustered into one single class. In fact, for ferromagnetic models, all neurons are perfectly synchronized at $T = 0$ and only one cluster is obtained. For increased T, the differences increase so that only very similar items are clustered together. Accordingly, for $T \to \infty$, all configurations are equally probable, indicating independent neurons and thus independent items. In this way, a kind of clustering hierarchy is generated, going from the one cluster level via intermediate levels, to unclustered points.

If the structures are no longer fixed ($\mathbf{j}(\mathbf{s}, t) \neq$ *const*), (1) generally is no longer a Markov process on S. Two cases are generic: (**I**) The structure $\mathbf{j}$ is strictly $\mathbf{s}$-dependent and (4) holds with $\mathbf{j} = \mathbf{j}(\mathbf{s})$. Then, as in the case $\mathbf{j} =$ *const*, (4) can still be approximated by a Markov process. (**II**) Learning, modifying the network structures, is used to provide optimized clustering solutions. After learning, one particular structure $\mathbf{j_0}$ is chosen and cannot be replaced. Technically, this requires setting $\Pi = 0$ as long as the network structure changes (i.e., in the transient phase). The relevant steady state distribution is $p_{\mathbf{j_0},T}$, which can again be approached by a Markov process. For models with a free parameter T, the determination of the best T is another, possibly nontrivial, issue of learning (see [1]). Obviously, $\Pi(t)$ in (2) helps to avoid an overestimation of the transients for the stationary distribution of (1) (e.g. $\Pi(t) = 0$ when $\mathbf{j}(t) \neq \mathbf{j_0}$).

3. Hopfield clustering

In our approach, a clustering method is uniquely defined by the cost function which depends on the architecture and vice versa. From this function, by means of the Gibbs ensemble, the probabilities of configurations emerge. To illustrate this concept, we consider a (non-standard) stochastic Hopfield network whose neurons can occupy either firing state $s_i \in \{-1, 1\}$. The underlying dynamics is sequential, i.e., at each update step t only one single, randomly drawn neuron k_t will be updated:

$$\begin{aligned} i \neq k_t : & \quad s_i(t+1) = s_i(t) \\ i = k_t : & \quad s_i(t+1) = sgn[\tanh[h_i(s(t))/T] + \eta_i(t)]. \end{aligned} \tag{5}$$

$\eta_i(t)$ is a independent random number (representing threshold noise) uniformly drawn from the interval $[-1, 1]$. T controls the noise impact. $h_i(s(t))$ is the local field calculated after

$$h_i(s(t)) = \sum_{j=1}^{n} J_{ij} s_j(t). \tag{6}$$

The time-invariant structure of the network is described by

$$J_{ij} = J_{ji} = \frac{1}{\widehat{K}} e^{\frac{-d_{ij}^2}{2a^2}}, \tag{7}$$

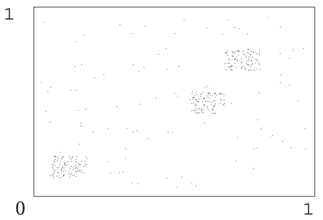

Figure 1: A two-dimensional clustering problem.

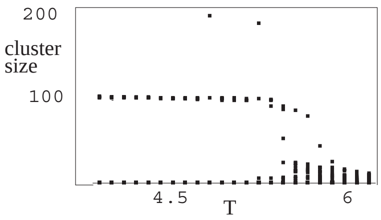

Figure 2: The size of the occurring clusters vs. T. The three (here overlapping) clusters of size 100 decay for large T.

if j is one of the k nearest neighbors of i, and $J_{ij} = 0$ otherwise. $\widehat{K}$ is the average number of coupled neighbors and a is the average distance between them.

(5) is biologically plausible, as only local information is required. It specifies the process (1) maintaining the structure fixed ($\mathbf{j} = const$). Consequently, (5) defines a Markov process with the stationary Gibbs distribution (4), whose cost function is the Ising model Hamiltonian

$$H(\mathbf{s}) = -\frac{1}{2}\sum_{i,j}^{n} J_{ij}s_is_j. \qquad (8)$$

Obviously, if two neurons enter the same state, the cost function is significantly lowered, due to the larger J_{ij}. Thus, such configurations are more frequently visited during the process (5). To find natural clusters, we use the G-criterion and scan the T-range for stable clusters. There is, however, a notable disadvantage when using the update rule (5). For practical values $M \ll \infty$, the system usually gets stuck in a basin of attraction of a local minimum of H. This can make the detection of independent clusters impossible, as in local minima neurons of different clusters can be in the same firing state (which happens necessarily for more than two clusters). To avoid trapping by minima, the configuration should intermittently be reset to a random configuration. This procedure corresponds to short spontaneous noise bursts in the network.

The clustering algorithm was tested on a two-dimensional toy system (see Fig.1) with three clusters of the same size (100). Fig.2 shows the identified clusters for a fixed T-range. Three clusters can be recognized. The background decays into singletons. The results are not very sensitive to the choice of k (number of nearest neighbors). However, for calculations with reasonable M, fluctuations can occur (e.g, with as an extrem effect, merging of two clusters), which complicates an automated cluster identification.

4. Superparamagnetic clustering

For data analysis, local update rules, such as (5), are slow and not particularly suitable for applications. Superparamagnetic clustering, introduced in [2], departs with the definition of the cost function H and uses a more efficient algorithm to converge towards the Gibbs distribution (4). Each neuron can take $m = 20$ possible firing states. H is defined similar to (8) as

$$H(\mathbf{s}) = \frac{1}{2}\sum_{i,j}^{n} J_{ij}(1 - \delta_{s_is_j}), \qquad (9)$$

where the J_{ij} are fixed and determined as in (7). For the calculation of (2), a global spin flip algorithm, the Swendsen-Wang algorithm, is used. For this algorithm, the first configuration can be chosen at random. The update (1) is performed as follows: First, coupled neurons ($J_{ij} \neq 0$) are frozen together with probability $P^f_{i,j} = 1 - \exp(-\frac{J_{ij}}{T}\delta_{s_is_j})$. Next, SW clusters are identified. A SW cluster contains all neurons that are connected via a path of frozen bonds. Finally, to all neurons of a SW cluster, an identical randomly drawn firing value is assigned. This procedure is more efficient and reliable than local updating, as local minima are avoided (resulting in a reduced autocorrelation time).

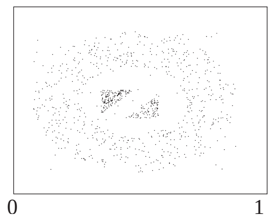

Figure 3: An example of a two-dimensional nontrivial clustering problem: different shapes (two triangular clusters (≈ 100/160) and a ring cluster (≈ 600)) and differing densities (including background noise) render this problem a nontrivial one.

The results (see Fig. 4) obtained for the nontrivial toy system of Fig.3, show three clearly distinguishable clusters, where the T-stability of the different clusters depends on their density. Thus, for the identification of clusters of different densities, a global clustering resolution, as given by T, is disadvantageous [1]. This can be remedied by optimizing $T = T_{ij}$ locally, on the cluster level. In order to maintain the principle (4), we will interpret this procedure as a network structure optimization (learning) with effective synaptic weights $J^{eff}_{ij} = J_{ij}/T_{ij}$.

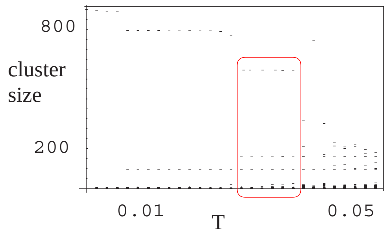

Figure 4: Size of detected clusters vs. temperature T. In the marked phase, the three clusters occur simultaneously.

5. Clustering by cost optimization

Pairwise clustering is an optimization problem, where the aim is to minimize the following (or a similar) cost function

$$H_{pc} = \sum_{\nu\le k}\left(|C_\nu| \sum_{\{i,j\}\in E_{\nu\nu}} \frac{d_{ij}}{2|E_{\nu\nu}|}\right), \qquad (10)$$

where $E_{\alpha\beta} = \{\{i,j\}|x_i \in C_\alpha \wedge x_j \in C_\beta\}$ and C_ν denotes a cluster. The cost minimum balances between the size of the clusters and their compactness. (10) corresponds to a cost function of an all-to-all network, where the structure $\mathbf{j}$ depends on the firing configurations $\mathbf{s}$, i.e., $\mathbf{j} = \mathbf{j}(\mathbf{s})$:

$$H_{pc}(s) = \sum_{\{i,j\}} J_{ij}(\mathbf{s})\delta_{\mathbf{s_i s_j}}, \qquad (11)$$

with $J_{ij}(\mathbf{s}) = d_{ij}/(|s_i| - 1)$. $|s_i| = \sum_{j=1}^{n} \delta_{s_i s_j}$ is the number of neurons in the state s_i with $s_i \in \{1, 2, ..., k\}$. To achieve the cost minimum $\min_{\mathbf{s}} H_{pc}(\mathbf{s})$, annealing techniques are frequently used. Generally, cost optimization is a $T \to 0$ process, i.e., the clusters are frozen groups of synchronized neurons. Unlike in ferromagnetic models, $T = 0$ usually does not lead to one single big cluster. Instead, the result yields as many clusters as possible (confined by k). This is achieved by avoiding configurations with synchronized 'dissimilar neurons', by rendering them suboptimal in terms of the cost function H.

K-means is a clustering approach that is based upon a fixed number of clusters. In the network picture, k central neurons are given, to which all other neuron are connected. The central neurons have *fixed*, and distinct, firing states $(q_1, ..., q_k)$. The goal is to find the optimal firing configuration and structure, so that the following cost function is minimized:

$$H_{km}(s) = \sum_{i\le n}\sum_{\nu\le k} J_{i\nu}\delta_{s_i q_\nu} = \sum_{\{i,\nu\}} J_{i\nu}(\mathbf{s})\delta_{s_i q_\nu}, \qquad (12)$$

where

$$J_{i\nu} = |x_i - y_\nu|^2 \text{ and } y_\nu = \sum_j x_j/|s_j|\delta_{s_j q_\nu}. \qquad (13)$$

The corresponding learning rule can be interpreted in terms of (1) as a deterministic or a $T = 0$ process: First, each neuron j adopts the firing state of the 'closest' central neuron (initially chosen by random assignment): $s_j = q_i$ with $i = arg \min_\nu J_{j\nu}$. Then, the network structure is changed, i.e., the $J_{i\nu}$ are adapted according to (13). In ideal cases, the repetition of the procedure quickly leads to the minimum.

6. Discussion

We have introduced a framework for the coherent description of clustering methods. In the framework, each clustering method entrains its own specific neural network architecture. During clustering, each data item is represented by a neuron and clusters are identified by groups of preferentially synchronized neurons. The latter emerge from a self-organized process, mediated by an architecture-specific cost function over the firing configurations. The costs determine the probability of a configuration in the dynamical steady state situation. We have pointed out that the Gibbs distribution is a natural choice for the stationary distribution, where the cost function takes the role of the Hamiltonian. As the stationary distribution fully characterizes the clustering solution, an appropriate Markov process will give rise to the formation of clusters. Two specific families of methods have been discussed: (**I**) In the ferromagnetic methods (e.g., Hopfield, or superparamagnetic clustering), coupled neurons intrinsically try to synchronize. This tendency is opposed by the noise which comes in the guise of the temperature T in the Gibbs distribution. Noise renders the synchronization of two neurons the less probable the weaker the coupling is. As a consequence, groups of strongly coupled neurons form clusters of synchronized neurons over a whole range of temperatures T. Robust and unbiased clusters are the results. (**II**) Cost optimization methods (e.g., pairwise or K-means clustering) are described as $T \to 0$ border cases. Reasonable clustering results require the network structure to depend on the firing configurations. As the choice of the number of possible firing states strongly influences the solution, these methods are naturally more biased. Stochastic variants using $T > 0$ can also be interpreted as members of this class. Other deterministic methods (linkage methods, e.g.) can be included in this framework as $T = 0$ border cases.

References

[1] T. Ott, A. Kern, A. Schuffenhauer, M. Popov, P. Acklin, E. Jacoby, R.Stoop, "Sequential superparamagnetic clustering for unbiased classification of high-dimensional chemical data," *J. Chem. Inf. Comput. Sci.*, vol.44(4), pp. 1358-1364, 2004.

[2] M. Blatt, S. Wiseman, E. Domany, "Superparamagnetic clustering of data," *Phys. Rev. Lett.*, vol.76, pp. 3251–3254, 1996.