International Journal of Uncertainty,
Fuzziness and Knowledge-Based Systems
Vol. 22, No. 6 (2014) 913–940
© World Scientific Publishing Company
DOI: 10.1142/S0218488514500482

Relational Fuzzy Self-Organizing Maps for Cluster Visualization and Summarization

Mohammed A. Khalilia
Computer Science Department,
University of Missouri, Columbia, MO 65211, USA
mohammed.khalilia@gmail.com

Mihail Popescu
Health Management and Informatics Department,
University of Missouri, Columbia, MO 65212, USA
popescum@missouri.edu

Received 27 August 2013
Revised 3 September 2014

The notion of Best-Matching Unit (BMU) in the proposed Fuzzy Relational Self-Organizing (FRSOM) algorithm is replaced by a membership function where every neuron has a certain degree of matching to an input object. The FRSOM is an extension of the relational self-organizing map. In the proposed FRSOM we incorporate a monotonically increasing fuzzifier and a monotonically decreasing neighborhood kernel. Initially, FRSOM assigns winning neurons. However, as time progresses adjacent neurons begin communicating and sharing information about the stimulus received. The amount of information being shared at a given time is governed by the fuzzifier and the number of neurons sharing information is controlled by the neighborhood kernel. Additionally, in this paper we show that FRSOM is the relational dual of Fuzzy Batch SOM (FBSOM) followed by experimental results comparing both FBSOM and FRSOM on synthetic datasets. Then we will demonstrate the visualization and summarization capabilities of FRSOM on two real relational datasets, Gene Ontology and a patient data consisting of Activity of Daily Living score trajectories.

Keywords: Kohonen networks, relational data, self-organization, unsupervised learning.

1. Introduction

Data visualization and exploration tools are about helping us understand and communicate ideas that can eventually drive action. For instance, such tools can assist in confirming the assumptions we make about the data or explore other datasets that exist in high dimensions. Fortunately, many algorithms were developed to handle the visualization and dimensionality reduction of high dimensional data. The class of such algorithms includes, but is not limited to, Principle Component Analysis (PCA),[1] Multi-dimensional Scaling (MDS),[2] Isomap,[3] Locally Linear Embedding (LLE)[4] and Laplacian Eigenmaps.[5] Some of those techniques, such as Laplacian Eigenmaps, construct a weighted graph of n nodes, where every node represents a point in higher dimensions.[5] Weighted edges connect adjacent nodes. The weights are defined using what is called a heat kernel, which assigns weights based on the distance between two points in the high dimensional space. Similar to the heat kernel, Kohonen Self-Organizing Maps (SOM) employs a neighborhood kernel that assigns weights for neurons based on their proximity in lower dimensions. Also, contrary to LLE where the number of nodes is equal to n, SOM maps n points to a predefined c neurons.

SOM is an unsupervised learning technique aimed at data exploration, clustering and visualization. It projects an s-dimensional input space into a low dimensional, usually two dimensional, lattice or grid of neurons. SOM is a powerful algorithm and has been used in many applications such as the system used to analyze the Sydney 2000 Olympic results using Viscovery SOMine software[6] or the SOM embedded in an Android device used for fall detection.[7] Another application is omeSOM software and the biological SOM which are used in biological sciences for data visualization[8] and the WEBSOM that is used for information retrieval and document clustering.[9] These are only a handful of numerous possible applications and different implementations of SOM that are tailored to address various problems.

Other variations of SOM are the Fuzzy SOM (FSOM) algorithms. The general idea of FSOM is to integrate fuzzy set theory into neural networks to give SOM the capabilities of handling uncertainty in the data. For instance, a FSOM algorithm for object data was proposed in Ref. 10 which is, in some sense, a regularization of the Fuzzy c-Means (FCM) algorithm. The FSOM in Ref. 10 is based on a cost function that is derived by introducing two modifications to the generalized FCM. First, the code vectors are distributed on a regular, low dimensional grid as in SOM, and a penalty term is added to guarantee a smooth distribution of the codebook vector values on the grid to help preserve the topological structure of the data. In Ref. 11 fuzzy object SOM based on fuzzy inputs and fuzzy weights for market segmentation of credit cards was proposed. FCM is applied for fuzzy clustering to identify the ambiguous sampled data located near the border between the clusters. In Ref. 12, a fuzzy SOM was developed by replacing the neurons of the original SOM with fuzzy rules, which are composed of fuzzy sets. The output of each rule is a singleton. For that reason, the algorithm maps the s-dimensional input space to a one dimensional output space. In Ref. 13, a hybrid SOM is proposed to predict overlapping clusters of high dimensional data and to detect the uncertainty that comes from the overlapping data. This approach is based on rough set theory to generate soft clustering. In Ref. 14, the same authors proposed a variation to Ref. 13 in which a two-level stage SA-Rough SOM (Simulated Annealing Rough Self-Organizing Map) was proposed.

Another fuzzy online Kohonen clustering networks was proposed.[15] The authors address major problems of SOM, such as the termination criteria, convergence and the SOM dependency on the sequence of the input data. To address these problems, FCM model is integrated into the learning rate allowing the neuron weights update function to be inversely proportional to their distance from the kth data point, o_k. As the fuzzifier gets smaller, updating the weights reverts back to Hard c-Means (winner take-all). As it gets larger, the weights are updated with lower individual learning rates. A fuzzy SOM algorithm was proposed in Ref. 16 where the FCM membership function was used to compute the degree of belongingness between a neuron and an object. However, the author abandoned the neighborhood function since it increases the computational complexity of SOM.

SOM was also extended to handle relational data. As we know, objects can be described in feature vectors or as relations. Object data $X = \{x_1, \dots, x_n\} \subset R^s$ consists of s-tuples of numerical feature vectors, x_k, that describe the objects o_k. On the other hand, relational data is presented as an $n \times n$ matrix R. Every element in R, r_{jk}, measures the relationship (similarity or dissimilarity) between objects o_j and o_k. For example, a dissimilarity relation satisfies the following conditions:

$$r_{jj} = 0 \qquad \forall\, j = 1, \dots, n, \tag{1a}$$

$$r_{jk} \geq 0 \qquad \text{for } k = 1, \dots n \text{ and } j = 1, \dots, n, \tag{1b}$$

$$r_{jk} = r_{kj} \quad \text{for } k = 1, \dots n \text{ and } j = 1, \dots, n. \tag{1c}$$

Few extensions to SOM were proposed to handle relational data.[17–19] In Ref. 17, a Self-Organizing Map for dissimilarity data was proposed. The authors described each neuron by a codebook that represents a subset of input vectors. The codebooks are then updated using a cost function that resembles the c-means objective function and accounts for both the relational data and the neighborhood topology. In Ref. 18, the authors extended Neural Gas (NG) and SOM to relational data based on the relational dual of the c-means clustering algorithm derived in Refs. 20 and 21. Every neuron is represented as a coefficient vector $\alpha_i \in R^s$ (this algorithm will be discussed in detail in Sec. 3).

An Ontological Self-Organizing Map (OSOM) was proposed to visualize and summarize datasets composed of words.[19] Ontological based similarity measures, such as the generalized outer product and ordered weighted average, in addition to the relational clustering distance measure, were integrated into SOM. Unlike the RSOM, where the coefficient vectors $\alpha_i \in R^s$, in OSOM, a prototype is a weight vector of fuzzy membership representation of all the terms in the dataset. Hence, the dimensions of the prototype vector can be less than s. For instance, the authors used Gene Ontology dataset of 194 gene products to test their algorithm. The dataset contains 64 terms; therefore, the length of the weight vector is 64 rather than 194 as would be the case in RSOM.

Regardless of the SOM algorithm being used, be it fuzzy, relational or ontological, the goal is to produce a low dimensional map, usually two-dimensional, to visualize the data. One type of maps that is widely used is the Unified Distance Matrix (U-matrix) which visualizes the topology of the data. A U-matrix contains valleys representing the clusters separated by mountain ranges that act as boundaries between the valleys. A good SOM produces a nontrivial U-matrix.[22] U-matrix is "nontrivial" when its watershed order or the number of distinct catchment basins is greater than one, but much less than n. This paper demonstrates the ability of FRSOM to produce a "nontrivial" U-matrix that can preserve the data topology. The reader is referred to Ref. 22 for more details about the concept of "nontrivial" U-matrix and its significance.

In this paper, we present the theoretical framework of FRSOM, and extend and elaborate on the FRSOM algorithm concept we proposed in Ref. 23. Here, we present a new Fuzzy Batch Self-Organizing Maps (FBSOM), which employs a monotonically increasing fuzzifier. Based on FBSOM, we then derive its fuzzy relational dual. In both FBSOM and FRSOM the notion of the Best Matching Unit (BMU) no longer exists. Instead, by using a monotonically increasing fuzzifier and a monotonically decreasing neighborhood size we give neurons the ability to share and communicate information about the stimulus. To our knowledge, this is the first attempt to apply fuzzification to the relational SOM. We demonstrate the benefits of FRSOM with extensive evaluations and comparisons to the FBSOM using multiple synthetic and real datasets.

The rest of the paper is organized as follows: Section 2 provides a brief overview of the Online and Batch SOM (BSOM) algorithm. Section 3 briefly introduces the RSOM algorithm. Section 4 presents the first contribution of the paper which is the Fuzzy Batch SOM (FBSOM). Section 5 presents the second contribution of the paper where we derive the relational dual of FBSOM algorithm and sets a few theorems that link FRSOM to RSOM. Section 6 justifies the use of a monotonically increasing fuzzifier and the ability of neurons to share information. Section 7 addresses the criteria used to evaluate the proposed algorithm. Section 8 briefly explains the technique used for SOM summarization. Section 9 presents results obtained on synthetic and real dataset which demonstrates the effectiveness of FRSOM. Finally, Sec. 10 concludes with analysis, remarks and future work.

2. Self-Organizing Map

SOM is an unsupervised learning technique that has been widely used in data visualization, exploration and clustering. SOM performs dimensionality reduction from a high-dimensional data space, $\mathbb{R}^s$, to a lower dimensional lattice or a map, usually two-dimensional. This feature allows us to visualize the cluster tendency of high-dimensional data. SOM forms a network structure that can be two-dimensional square, hexagonal grid or toroidal. Every node or neuron in the structure is connected to its neighbor using a neighborhood kernel, which gives SOM the topology preserving characteristic.

The online SOM algorithm starts by drawing a random data point, x_k, from the input data which causes the weight vectors, m_i, to move closer towards x_k according to a neighborhood function, h, and a learning rate, η (notations are summarized in Table 1). The learning rate $\eta(t)$ determines the amount of influence x_k has on every neuron at iteration t, while $h(t)$ determines the amount of influence based on the proximity of the neuron to the BMU (proximity is defined as the distance between neuron's i coordinate a_i and neuron's j coordinate a_j in 2D). BMU, denoted by w_k, is the closest neuron to the input x_k and therefore is influenced the most. In other words, SOM assigns a full membership for o_k to the winning neuron i $(u_{ik} = 1)$. However, if the entire dataset is available, one can use batch SOM (BSOM). BSOM can be significantly faster and does not require the specification of the learning parameter η.[24] Algorithm 1 outlines the BSOM procedure.

Table 1. Notations.

Symbol	Explanation
c	Number of neurons in the lattice
i	Refers to the ith neuron, $1 \leq i \leq c$
a_i	Neuron's i position in 2D space
x_k	Feature vector representing o_k, where $x_k \subset \mathbb{R}^s$
m_i	The weight vector of the ith neuron in the non-relational SOM
h_{ip}	Neighborhood function between neuron i and p
η	Learning rate
w_k	Refers to the position or index of the winning neuron of o_k
u_{ik}	The membership grade of o_k in neuron i
n	Number of objects in the dataset
o_k	The kth object in the dataset, $1 \leq k \leq n$
α_i	The coefficient vector of the ith neuron in the relational SOM
r_{jk}	The distance between o_j and o_k
d_{ik}	The distance between m_i and o_k
σ_0	Initial neighborhood size or radius
σ_f	Final neighborhood size or radius
q_0	Initial fuzzifier
q_f	Final fuzzifier
$q = q(t)$	Fuzzifier value at time t
t_{max}	Number of training epochs
N_i	Set of immediate neighboring neurons to i

3. Relational Self-Organizing Map

SOM is a very effective technique when the objects in the dataset are represented by feature vectors. However, when objects are described in relational form, one needs to use the Relational Self-Organizing Map (RSOM).[18] In RSOM, it is not necessary to know the vectorial representation of the input data to compute the cluster prototypes or weight vectors. Instead, a weight vector, m_i, is expressed as a linear combination of the input data points $m_i = \sum_{k=1}^{n} \alpha_{ik} x_k$ where $\sum_{k=1}^{n} \alpha_{ik} = 1$. The dissimilarity between a weight vector m_i and object o_k is computed based on the coefficients α and the dissimilarity matrix R (8). The goal in RSOM is to minimize the following objective function[18]

Algorithm 1. Batch Self-Organizing Map (BSOM)

```
►Input: Data $O = \{o_1, \dots, o_n\}$, where $o_k$ is represented by a feature vector $x_k$, map
        size, c neurons, initial radius $\sigma_0$, final radius $\sigma_f$
►Initialize random weight vectors $m \subset \mathbb{R}^s$
►for $t = 1$ to $t_{max}$
    Calculate the distance between neuron $i$ and $o_k$
        $d_{ik} = \|m_i - x_k\|^2;\ \forall\ 1 \le i \le c \text{ and } 1 \le k \le n.$    (2)
    Assign the winning neuron
        $w_k = \arg\min_i d_{ik};\ \forall\ 1 \le k \le n.$    (3)
    Compute the neighborhood function
        $h_{i,w_k}(t) = \exp\left(\frac{-\|a_i - a_{w_k}\|^2}{2\sigma^2(t)}\right);\ \forall\ 1 \le i \le c \text{ and } 1 \le k \le n.$    (4)
    Update the weight vectors
        $m_i(t+1) = \sum_{k=1}^{n} h_{i,w_k} x_k \Big/ \sum_{k=1}^{n} h_{i,w_k};\ \forall\ 1 \le i \le c.$    (5)
    Update the neighborhood radius
        $\sigma(t+1) = \sigma_0 \cdot (\sigma_f/\sigma_0)^{t/t_{max}}.$    (6)
    $t \leftarrow t + 1$
```

$$L(U;R) = \sum_{i=1}^{c} \frac{1}{2\sum_{k=1}^{n} h_{i,w_k}} \sum_{k=1}^{n} \sum_{k'=1}^{n} h_{i,w_k} h_{i,w_{k'}} r_{kk'}. \tag{7}$$

RSOM algorithm is outlined below:

Algorithm 2. Relational Self-Organizing Map (RSOM)

```
►Input: $n \times n$ relational data matrix $R$, map size, $c$, $\sigma_0$, $\sigma_f$
►Initialize random coefficient vectors $\alpha \subset \mathbb{R}^n$
►for $t = 1$ to $t_{max}$
    Calculate the distance between neuron $i$ and $o_k$
        $d_{ik} = \|m_i - x_k\|^2 = (R \cdot \alpha_i)_k - \frac{(\alpha_i^t \cdot R \cdot \alpha_i)}{2}$;
            $\forall\, 1 \le i \le c$ and $1 \le k \le n$.                                  (8)
    Assign the winning neuron using (3)
    Compute the neighborhood function using (4)
    Update the coefficients vector values
        $\alpha_{ik}(t+1) = \frac{h_{i,w_k}}{\sum_{k=1}^{n} h_{i,w_k}}$; $\forall\, 1 \le i \le c$ and $1 \le k \le n$.      (9)
    Update the neighborhood radius using (6)
    $t \leftarrow t + 1$
```

4. Fuzzy Batch Self-Organizing Map

Numerous fuzzy SOM algorithms were proposed. This section presents a new Fuzzy Batch SOM (FBSOM) algorithm which will serve as a foundation for the next section, the Fuzzy Relational SOM.

Contrary to the "winner take-all" paradigm, as in BSOM and RSOM, FBSOM gives neurons the ability to share a stimulus with their neighboring neurons. The amount of sharing is controlled by the fuzzifier and the number of neurons involved in sharing is governed by the neighborhood kernel. More discussion about the fuzzifier and information sharing will be presented in Sec. 6. The goal of FBSOM is to find a fuzzy partition $U \in M_{fcn}$, where

$$M_{fcn} = \left\{ U \in \mathbb{R}^{c \times n} \,\middle|\, \begin{array}{c} u_{ik} \in [0,1], \sum_{k=1}^{n} u_{ik} > 0, \\ \sum_{i=1}^{c} u_{ik} = 1, \forall\, 1 \le i \le c \text{ and } 1 \le k \le n \end{array} \right\} \tag{10}$$

and codebook vectors $m = \{m_1, \dots, m_c\}$, $m_i \in \mathbb{R}^s$, that minimizes the objective function

$$J_q(U, m; X) = \sum_{i=1}^{c} \sum_{k=1}^{n} u_{ik}^q \sum_{j=1}^{c} h_{ij}^q \, d_{jk}. \tag{11}$$

The objective function (11) is similar to the probabilistic SOM proposed in Ref. 25. And in contrast to fuzzy clustering methods where the fuzzifier, q, remains constant with time, in FRSOM, q changes at every iteration. For reasons we will discuss in Sec. 6, q uses a monotonically increasing function (15) where q varies within a range $[q_0 \; q_f]$, for example, $q_0 = 1$ and $q_f = 2$. The fuzzifier q in (11) and other equations we will encounter refers to the fuzzifier value at time t, $q(t)$. Therefore, the notations q and $q(t)$ are equivalent and we will use q for convenience.

Theorem 1. *Let $X = \{x_1, \dots, x_n\} \subset \mathbb{R}^s$ be the set of feature vectors, $m = \{m_1, \dots, m_c\}$ be the c neurons codebook vectors, $\|x_k - m_j\|^2$ is the distance between feature vector x_k and codebook vector m_j ($k = 1, \dots, n; j = 1, \dots, c$), and $U \in M_{fcn}$. Then the set (U, m) might be a minimizer of J_q only if*

$$u_{ik} = 1 / \sum_{j=1}^{c} \left(\frac{\sum_{a=1}^{c} h_{ia}^q d_{ak}^2}{\sum_{b=1}^{c} h_{jb}^q d_{bk}^2} \right)^{1/q-1} \quad \text{and} \quad m_j = \frac{\sum_{i=1}^{c} \sum_{k=1}^{n} h_{ij}^q u_{ik}^q x_k}{\sum_{i=1}^{c} \sum_{k=1}^{n} h_{ij}^q u_{ik}^q}.$$

Proof. To derive the necessary conditions and membership update equations, we set Lagrange optimization problem to minimize (11) under the constraint $\sum_{i=1}^{c} u_{ik} = 1, \forall k$, taking the derivative with respect to u_{ik} and set it equal to 0. We find that U might be a minimum of J_q only if

$$u_{ik} = 1 \Bigg/ \left[\sum_{j=1}^{c} \left(\frac{\sum_{a=1}^{c} h_{ia}^q d_{ak}^2}{\sum_{b=1}^{c} h_{jb}^q d_{bk}^2} \right)^{\frac{1}{q-1}} \right]. \tag{12}$$

Also, differentiating (11) with respect to m_j and setting it to 0 leads to codebook update equation

$$m_j = \sum_{i=1}^{c} \sum_{k=1}^{n} h_{ij}^q u_{ik}^q x_k \Bigg/ \sum_{i=1}^{c} \sum_{k=1}^{n} h_{ij}^q u_{ik}^q. \tag{13}$$

As you can see, o_k is assigned a partial or fuzzy membership in multiple neurons. Consequently, the notion of "best-matching unit" is no longer applicable to FBSOM and it is replaced by the membership degree. In fact, one can think of every neuron i as the winning neuron (BMU) of object o_k with degree u_{ik}. For this reason, none of the neurons in the proposed algorithm is empty.

By having $U \in M_{fcn}$, FBSOM leads to a smoother topology. The smoother topology is mainly due to eliminating empty neurons and the fact that adjacent neurons share similar memberships to object o_k. The importance and practicality of a smoother topology becomes more apparent when they are incorporated into the U-matrix. For instance, a less distorted and smoother U-matrix may help in increasing the accuracy of the U-matrix segmentation which is one of the methods used for the clustering of SOM.[26,27] The FBSOM is summarized in Algorithm 3.

Algorithm 3. Fuzzy Batch Self-Organizing Map (FBSOM)

```
►Input: Data $O = \{o_1, \ldots, o_n\}$, where $o_k$ is represented by a feature vector $x_k$, map
        size, $c$, initial fuzzifier $q_0$, final fuzzifier $q_f$, $\sigma_0$, $\sigma_f$
►Initialize Random weight vectors $m \subset \mathbb{R}^s$
►for $t = 1$ to $t_{max}$
    Calculate the distance between neuron $i$ and $o_k$ using (2)
    Compute the membership values, $u_{ik}$ using (12)
    Update the neighborhood function
        $h_{ij} = \exp\left(\frac{-\|a_i - a_j\|^2}{2\sigma^2(t)}\right).$   (14)
    Update weight vectors using (13)
    Update the neighborhood radius using (6)
    Update the fuzzifier value
        $q(t+1) = q_0 \cdot (q_f/q_0)^{t/t_{max}}.$   (15)
    $t \leftarrow t + 1$
```

5. Fuzzy Relational Self-Organizing Map

In this section, we will derive FRSOM theoretical framework, which is the major contribution of this paper. We will derive the membership and coefficient update equations and the fuzzy relational dual of the FBSOM. As mentioned earlier, in relational clustering it is not necessary to know the vectorial representation of the input data to compute the cluster prototypes or weight vectors. Instead, the codebook vector m_i is expressed as linear combination of the input data points $m_i = \sum_{k=1}^{n} \alpha_{ik} x_k$, $\sum_{k=1}^{n} \alpha_{ik} = 1$, where α_{ik} is computed as

$$\alpha_{ik} = \delta_{ik} \Big/ \sum_{k'=1}^{n} \delta_{ik'} \qquad (16)$$

and δ_{ik} is defined as

$$\delta_{ik} = \sum_{j=1}^{c} h_{ij}^{q} \cdot u_{jk}^{q}. \tag{17}$$

Using Eqs. (11)–(13), (16) and (17), we can re-formulate the fuzzy optimization scheme in terms of the relational data, R, which we will do next.

Definition. Let $U \in M_{fcn}$ be a fuzzy partition of R, the FRSOM functional K_q is defined as

$$min\left\{K_q(U;R) = \sum_{i=1}^{c} \frac{1}{2\sum_{k=1}^{n} \delta_{ik}} \sum_{k=1}^{n}\sum_{k'=1}^{n} \delta_{ik} \cdot \delta_{ik'} \cdot r_{kk'}\right\}. \tag{18}$$

The objective function (18) attempts to minimize the sum of pairwise distances among objects belonging to the same neuron. To better understand the FRSOM objective function, we will decompose it into its basic elements:

$h_{ij}^{q} \cdot u_{jk}^{q} =$	grade of membership of o_k contributed to neuron i by j, weighted by the neighborhood function;
$\delta_{ik} = \sum_{j=1}^{c} h_{ij}^{q} \cdot u_{jk}^{q} =$	sum of memberships of o_k contributed to i from the surrounding neurons;
$\sum_{k=1}^{n}\sum_{k'=1}^{n} \delta_{ik} \cdot \delta_{ik'} \cdot r_{kk'} =$	within neuron i sum of pairwise distances among objects weighted by the contributed memberships of o_k and $o_{k'}$ from neighboring neurons.

Notice that the neurons in FBSOM and FRSOM become less competitive than the “winner take-all” SOM. And while o_k has a membership in neuron i, o_k enjoys an additional membership in i, contributed by its neighbors.

Theorem 2. *Let* $q > 1$, $\|\cdot\|$ *be a norm induced by inner product on* $\mathbb{R}^s$, $X = \{x_1, \dots, x_n\} \subset \mathbb{R}^s$ *be a set of feature vectors, and* $R = [r_{jk}] = \left[\|x_j - x_k\|^2\right]$ *be the set of corresponding relational distance data. Then* $U \in M_{fcn}$ *is a minimizer of* K_q *if and only if* (U, m) *is a minimizer of* J_q *in* $U \in M_{fcn}$.

Proof. By substituting (16) in (8) which in turn is substituted in (11) we can derive the FRSOM objective function K_q as follows:

$$
\begin{aligned}
K_q(U,m;R) &= \sum_{i=1}^{c}\sum_{k=1}^{n}\sum_{j=1}^{c} u_{ik}^{q} h_{ij}^{q} (R\cdot\alpha_i)_k - \frac{(\alpha_i^t\cdot R\cdot\alpha_i)}{2} \\
&= \sum_{i=1}^{c}\sum_{k=1}^{n}\sum_{j=1}^{c} u_{ik}^{q} h_{ij}^{q} \left(R\cdot\frac{\sum_{j'=1}^{c} h_{ij'}^{q} u_{j'}^{q}}{\sum_{j'=1}^{c}\sum_{k'=1}^{n} u_{ij'}^{q} h_{j'k'}^{q}}\right)_k \\
&\quad - \frac{1}{2}\sum_{i=1}^{c}\sum_{k=1}^{n}\sum_{j=1}^{c}\left(\frac{\sum_{j'=1}^{c} h_{ij'}^{q} u_{j'}^{q}}{\sum_{j'=1}^{c}\sum_{k'=1}^{n} u_{j'k'}^{q} h_{ij'}^{q}}\cdot R\cdot\frac{\sum_{j'=1}^{c} h_{ij'}^{q} u_{j'}^{q}}{\sum_{j'=1}^{c}\sum_{k'=1}^{n} u_{j'k'}^{q} h_{ij'}^{q}}\right) \\
&= \sum_{i=1}^{c}\frac{\left(\sum_{j=1}^{c} u_j^{q} h_{ij}^{q}\right) R \left(\sum_{j'=1}^{c} u_{j'}^{q} h_{ij'}^{q}\right)}{\sum_{j'=1}^{c}\sum_{k'=1}^{n} u_{ij'}^{q} h_{j'k'}^{q}} \\
&\quad - \frac{1}{2}\sum_{i=1}^{c}\frac{\left(\sum_{j=1}^{c} u_j^{q} h_{ij}^{q}\right) R \left(\sum_{j'=1}^{c} u_{j'}^{q} h_{ij'}^{q}\right)}{\sum_{j'=1}^{c}\sum_{k'=1}^{n} u_{ij'}^{q} h_{j'k'}^{q}} \\
&= \sum_{i=1}^{c}\frac{\left(\sum_{j=1}^{c} u_j^{q} h_{ij}^{q}\right) R \left(\sum_{j'=1}^{c} u_{j'}^{q} h_{ij'}^{q}\right)}{2\sum_{j'=1}^{c}\sum_{k'=1}^{n} u_{ij'}^{q} h_{j'k'}^{q}} \\
&= \sum_{i=1}^{c}\frac{\sum_{k=1}^{n}\sum_{k'=1}^{n}\left(\sum_{j=1}^{c} u_j^{q} h_{ij}^{q}\right)\left(\sum_{j'=1}^{c} u_{j'k'}^{q} h_{ij'}^{q}\right) r_{kk'}}{2\sum_{j'=1}^{c}\sum_{k'=1}^{n} u_{ij'}^{q} h_{j'k'}^{q}}
\end{aligned}
$$

By setting $\delta_{ik} = \sum_{j=1}^{c} u_{jk}^{q} h_{ij}^{q}$ we can re-write K_q as:

$$
min\ \left\{K_q(U;R) = \sum_{i=1}^{c}\frac{1}{2\sum_{k=1}^{n}\delta_{ik}}\sum_{k=1}^{n}\sum_{k'=1}^{n}\delta_{ik}\cdot\delta_{ik'}\cdot r_{kk'}\right\}.
$$

From the above proof, we can see that $K_q(U;R) = J_q(U,m;X)$. By Theorem 1, we proved that $m = \{m_1,\dots,m_c\} \subset \mathbb{R}^s$ uniquely minimizes $J_q(U,m;X)$ for a fixed $U \in M_{fcn}$. It follows that U is a minimizer of $K_q(U;R)$ if and only if U is a minimizer of $J_q(U,m;X)$.

Theorem 2 establishes the close connection between the functionals K_q and J_q and in theory, we should be able to find the same partitions of objects using K_q in the relational scheme that is found using the object based FBSOM. Also, we can show the close connection between the functionals K_q and L which states that K_q is a generalization of L (7).

Corollary 1. If $\{q_0 = q_f\} \overset{+}{\rightarrow} 1$, the rate at which neurons share information decreases and the sharing stops when $q_0 = q_f = 1$. At that point, the FRSOM reduces to the "winner take-all" scheme as in the RSOM algorithm.

Proof. When $q_0 = q_f$, the fuzzifier q remains constant throughout the iterations, which means $q = q_0 = q_f$. Also, the limit property of (12) states the following:

$$\lim_{q\to 1} u_{ik} = \begin{cases} 1, & \sum_{a=1}^{c} h_{ia}^{q} d_{ak}^{2} < \sum_{b=1}^{c} h_{jb}^{q} d_{bk}^{2} \ \forall i \neq j \\ 0, & otherwise \end{cases};$$

$$\forall 1 \leq i \leq c; 1 \leq k \leq n.$$

Therefore, at $q_0 = q_f = 1$, δ_{ik} (17) converges to the neighborhood function,

$$\lim_{q\to 1} \delta_{ik} = \lim_{q\to 1} \sum_{j=1}^{c} h_{ij}^{q} \cdot u_{jk}^{q} = h_{i,w_k}.$$

Combining those results, we conclude that when $q_0 = q_f = 1$, $K_1(U;R) = L(U;R)$.

Corollary 1 states that RSOM is a special case of the FRSOM and it can be reduced to the "winner-take-all" scheme by using a constant fuzzifier and setting $q = 1$.

Theorem 3. *When σ is large, δ_{ik} is influenced by the memberships of o_k in the neighboring neurons to i. However, as neighborhood size shrinks with time and approaches its final value and as $\sigma_f \to 0$, δ_{ik} converges to u_{ik}^{q}.*

Proof. The limit property of (14) states the following:

$$\lim_{\sigma\to 0} h_{ij} = \begin{cases} 1, & \forall i = j \\ 0, & otherwise \end{cases} \forall 1 \leq i,j \leq c.$$

Therefore, we conclude that δ_{ik} converges to u_{ik}^{q} as $\sigma \to 0$

$$\lim_{\sigma\to 0} \delta_{ik} = \lim_{\sigma\to 0} \sum_{j=1}^{c} h_{ij}^{q} \cdot u_{jk}^{q} = u_{ik}^{q}.$$

Note that these theorems also apply to the FBSOM discussed earlier in Sec. 4. The complete FRSOM algorithm is outlined below:

Algorithm 4. Fuzzy Relational Self-Organizing Map (FRSOM)

```
►Input: $n \times n$ relational data matrix $R$, map size, $c$, $q_0$, $q_f$, $\sigma_0$, $\sigma_f$
►Initialize Random coefficient vectors $\alpha_i \subset \mathbb{R}^n, \sum_{k=1}^{n} \alpha_{ik} = 1$
►for $t = 1$ to $t_{max}$ do
    Compute the distance, $d_{ik} = \|m_i - x_k\|^2 = (R \cdot \alpha_i)_k - \frac{(\alpha_i^t \cdot R \cdot \alpha_i)}{2}$; $\forall i, k$.
    Update the membership value, $u_{ik} = 1 \Big/ \left[\sum_{j=1}^{c} \left(\frac{\sum_{a=1}^{c} h_{ia}^{q} d_{ak}^{2}}{\sum_{b=1}^{c} h_{jb}^{q} d_{bk}^{2}}\right)^{\frac{1}{q-1}}\right]$; $\forall i, k$.
    for $i = 1$ to $c$ do
        Update neighborhood function, $h_{ij} = \exp\left(\frac{-\|a_i - a_j\|^2}{2\sigma^2(t)}\right)$; $\forall j$.
        Update coefficient values, $\alpha_{ik} = \delta_{ik} / \sum_{k'=1}^{n} \delta_{ik'}$, $\delta_{ik} = \sum_{j=1}^{c} h_{ij}^{q} \cdot u_{jk}^{q}$; $\forall k$.
    $\sigma(t+1) = \sigma_0 \cdot (\sigma_f/\sigma_0)^{t/t_{max}}$.
    $q(t+1) = q_0 \cdot (q_f/q_0)^{t/t_{max}}$.
    $t \leftarrow t + 1$
```

6. Neurons Sharing Information

The remaining question is why does FRSOM employ a monotonically increasing fuzzifier? Why not simply use a constant fuzzifier similar to some clustering algorithms such as Fuzzy c-Means? (Note that this Section references FRSOM only, but the same argument applies to the FBSOM).

There are two reasons for employing this kind of fuzzifier. First, FRSOM has more degrees of freedom, map size, number of neurons initial and final radius. Introducing a new parameter, q, can cause undesirable interactions with some of the existing parameters. Indeed, both q and σ are related and one has to exercise extra care when setting them. Therefore, it is important to understand the behavior of the neighborhood function in relation to the fuzzifier. Remember this: SOM starts with a larger neighborhood radius σ_0 which decreases with time until it reaches σ_f and as $\sigma_0 \to \infty$ the neighborhood function $h_{ij} \to 1$ (14) causing the membership function (12) $u_{ik} \to 1/c$ (assuming a constant fuzzifier, i.e. $q = 2$). The experiments have shown that $\sigma_0 = 2$ is large enough to cause this problem. Thus, to alleviate this issue and prevent $u_{ik} \to 1/c$, FRSOM has to balance q and σ by employing a monotonically increasing fuzzifier and a monotonically decreasing radius. FRSOM will start as “winner take-all” with large radius and small fuzzifier $(q = 1)$ and the membership values become fuzzifier as q increases with time.

Second, a monotonically increasing fuzzifier allows the neurons to share information about the sensed stimulus. As a stimulus o_k is sensed at $t = 1$, a winning neuron i is assigned to that stimulus. However, neuron i loses the unity membership at $t > 1$ as it starts sharing and communicating information with its neighbors about o_k. That is when crisp memberships start becoming fuzzier. The harmony between the monotonically increasing fuzzifier and the monotonically decreasing neighborhood kernel governs and limits the sharing of information. The fuzzifier restricts the amount of information being shared with other neurons; as the fuzzifier increases, the membership values become more distributed and the amount of information communicated increases. On the other hand, the neighborhood kernel limits the intensity and the number of neurons sharing information. For instance, a Gaussian neighborhood kernel limits the sharing of information among distant neurons, so by the time FRSOM converges, only a small number of adjacent neurons should have a high firing strength for some stimulus. Hence, as time progresses more information is communicated among a smaller subset of neurons. In the experimental results, we will show an example demonstrating information sharing.

7. Evaluation Criteria

In this section we present the evaluation criteria used to assess FRSOM performance. For the evaluation we use: quantization error, topographic error and visualization based approach (U-matrix).

7.1. *Quantization error*

Quantization error (qe), a widely used SOM evaluation measure, is defined as the average distance between the objects in the dataset and their corresponding winning neurons.[28] A good map is expected to have a small qe.

$$qe = \frac{1}{n}\sum_{i=1}^{c}\sum_{k=1}^{n} u_{ik}^{q}\, d_{ik}. \tag{19}$$

Equation (19) weights the distance d_{ik} with the fuzzy membership u_{ik}. When $q_0 = q_f = 1$, (19) reduces to the original qe formulation.[28]

7.2. *Topographic error*

The quantization error is a good overall measure, but it does not reflect the topological preservation of the map. Different approaches for measuring the topology preservation were proposed Ref. 28. In this paper we use the topographic error (te) to measure the topology preservation. te measures the proportion of the input vector for which the first and second BMU are not adjacent neurons.

$$te = \frac{1}{n}\sum_{k=1}^{n} adj(o_k). \tag{20}$$

$$adj(o_k) = \begin{cases} 0, if\ the\ two\ closest\ neurons\ to\ o_k\ are\ adjacent \\ 1, if\ the\ two\ closest\ neurons\ to\ o_k\ are\ not\ adjacent \end{cases}.$$

To evaluate FRSOM, however, we do not use the two closest neurons, but rather the two neurons with the highest membership of o_k. Similar to qe, a good map is expected to yield a small te.

7.3. *Visualization based approach based on U-matrix*

Although the U-matrix is not a quantitative evaluation technique, nonetheless, it provides a quick overall qualitative assessment of how the algorithm performed. A U-matrix is calculated in the weight vector space ($\mathbb{R}^s$) or coefficient space in relational algorithms and displayed in the lattice space, which is usually two-dimensional. A U-height of a neuron i, $uh(m_i)$, is defined as the sum of distances from m_i to the neighboring neurons of i, N_i.[29] For instance, in a rectangular grid, N_i refers to the four immediate neighbors and the U-Height for neuron i is computed as:

$$uh(m_i) = \sum_{m_j \in N_i} \left\| m_i - m_j \right\|^2. \tag{21}$$

However, in relational SOM, such as FRSOM and RSOM, $\left\| m_i - m_j \right\|^2$ is calculated in terms of α and R as

$$\left\|m_i - m_j\right\|^2 = \alpha_j^t R \alpha_i - \frac{1}{2}\alpha_j^t R \alpha_j - \frac{1}{2}\alpha_i^t R \alpha_i. \tag{22}$$

A U-matrix is generated when the U-height of every neuron is calculated at that neuron's coordinates. The matrix can be displayed in 2D as a planar or in 3D to visualize the topology. When the U-matrix is visualized in 3D, the "mountain ranges" point to cluster boundaries and "valleys" refer to cluster centers.

8. SOM Summarization

Several techniques exist to summarize the results of SOM algorithms. One can either cluster the neurons and then identify which objects belong to which cluster,[30] or use image processing techniques to segment the U-matrix to find its distinct regions.[31,32] In this paper, we chose the latter technique and used the MATLAB implementation of the Watershed algorithm.[33] Once the Watershed algorithm is applied on the U-matrix, it returns multiple regions. Every region represents a catchment basin encompassing a group of similar neurons. While it is not always the case that we will identify the exact number of regions since Watershed algorithm can oversegment the U-matrix,[34] for the purposes of SOM summarization, oversegmentation is not an issue.

Once the regions are identified, we can uncover the similar neurons grouped in every region. For every neuron, we find the most representative object (the object with the highest weight or membership value to that neuron). Once every neuron has a representative object, a majority voting among neurons in that region is performed to vote for an overall representative label for that region. The label is then placed in the centroid of the region in question. Therefore, to label the region it is assumed that every object has an assigned label. For instance, an object in Hepta dataset is assigned a label $l \in \{1, 2, 3, 4, 5, 6, 7\}$, while patients in the ADL dataset are labeled with their own and unique ADL trajectory. As we will see later in the results section, every region is assigned an identifier as $R\#: l$, where # corresponds to the region index or number and l is the region label.

9. Experimental Results

To evaluate the proposed algorithm, FRSOM is compared to the FBSOM using synthetic and real datasets (Fig. 1), which are summarized in Table 2. For vectorial datasets, the dissimilarities among objects, r_{ij}, are calculated using the Euclidean distance. The well-separated three Gaussians (WS3G) and the overlapping three Gaussians (O3G) datasets are two-dimensional and they differ by the inter-cluster variance. Lines dataset contains three parallel lines. The congressional voting record dataset is published by the University of California-Irvine (UCI) Machine Learning Depository. Hepta dataset is part of the Fundamental Clustering Problem Suite (FCPS),[27] and it is used to demonstrate the behavior of the FBSOM and FRSOM.

In addition to the synthetic datasets that allow us to compare FRSOM to FBSOM, we test FRSOM using two real datasets, the Gene Ontology (GPD194) and the Activity of Daily Living (ADL). GPD194 is a pure relational dataset containing pairwise distances among Gene terms measured using fuzzy distance measure,[19] and ADL is a relational dataset containing pairwise distances among 3,963 patients measured using Dynamic Time Warping (DTW).[35] Notice that FRSOM expects an Euclidean relational matrix R and non-Euclidean matrix needs to be converted into an Euclidean one using techniques such as the β-Spread Tranformation[20]. However, this tranformation is necessary and important to perform if equation (8) results in negative values.[20] This situation was not encounter for GPD194 and ADL datasets and therefore no tranformation was necessary.

Table 2. Datasets.

	Name	Size	Dimensions	No. of Classes
A	Hepta	212	3	7
B	Well Separated Gaussians (WS3G)	1,500	2	3
C	Overlapping Gaussians (O3G)	1,500	2	3
D	Parallel Lines	300	2	3
F	Congressional Voting Record (CVR)	435	16	2
G	Gene Ontology (GPD194)	194	–	–
H	Activity of Daily Living (ADL) Patients	3,963	–	–

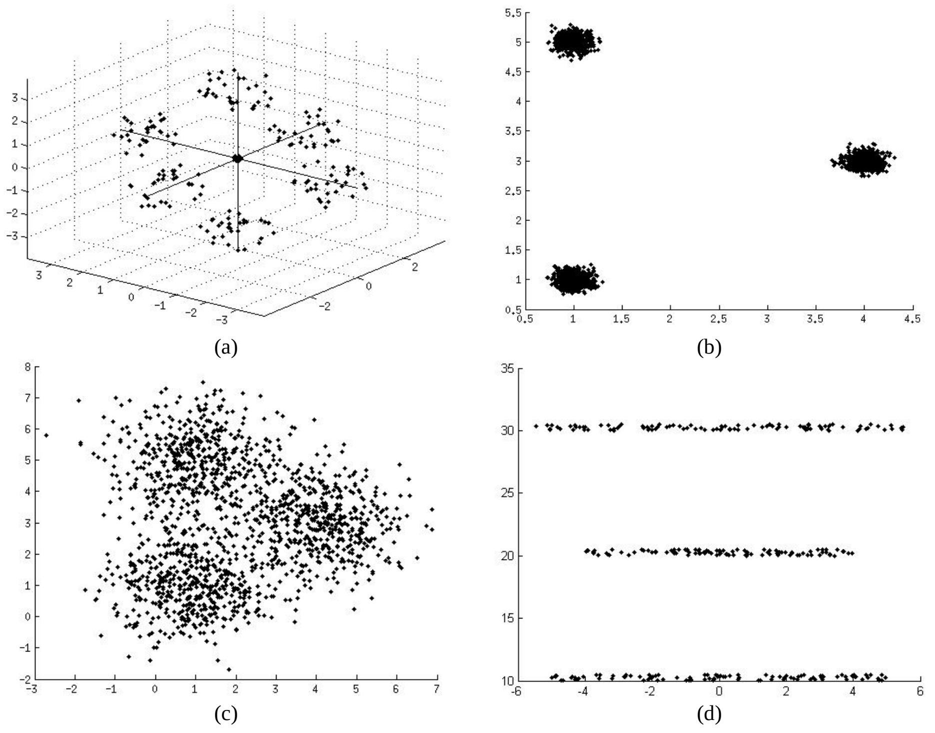


Fig. 1. Synthetic datasets (a) Hepta (b) WS3G, (c) O3G and (d) Parallel Lines.

The average and standard deviation of the qe, te and objective function values over 10 runs were calculated. For most datasets we used the parameters listed in Table 3. Notice that σ_0 is not fixed across all datasets, rather it is dataset specific. σ_0 is the only parameter that was varied and it was determined by trial and error. Overall, we found $\sigma_0 \in [1, 3]$ to give reasonable results for all datasets presented in this paper.

Table 3. Algorithm parameters.

Parameter	Value
q_0	1.01
q_f	2
σ_0	1–3
σ_f	0.5
C	400
Map size	20×20
Neighborhood function	Gaussian
Training length	10 epochs

9.1. *Hepta dataset*

The Hepta dataset contains 212 data points divided into the seven classes of 30 points each and two additional points in the center group.[27] The center group of points is about twice as dense as any of the six groups (Fig. 1(a), Table 2A, which refers to row A in Table 2). The goal of the Hepta dataset is to validate if the clustering algorithm can find the clusters with varying densities. The maps generated by FBSOM and FRSOM are shown in Figs. 2(a)–2(b), respectively. Every map is summarized by displaying the most representative labels for the catchment basins. The average qe, te and objective function values are presented in Table 7A.

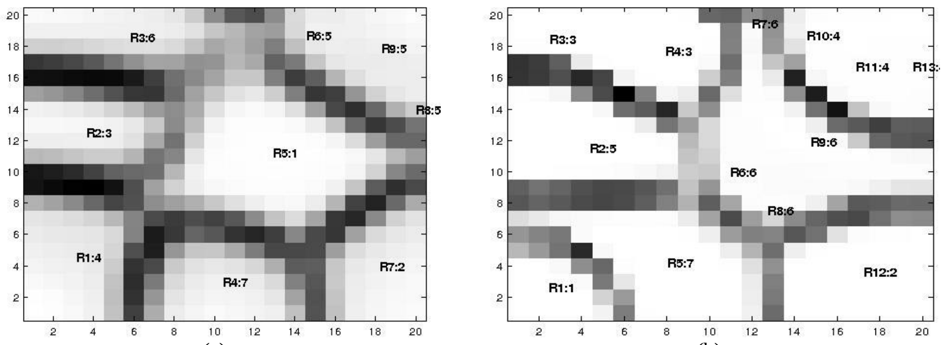
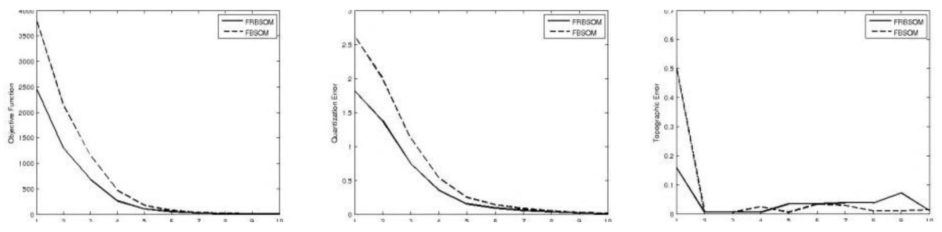


Fig. 2. Topographic map for Hepta dataset (a) FBSOM, (b) FRSOM.

The behavior of the objective function, quantization error and topographic error is shown in Figs. 3(a)–3(c) for one specific run. FRSOM starts with lower values (continuous line in Fig. 3), but as time progresses both algorithms behave similarly. In fact, these properties were observed in every dataset throughout the experiments (see Table 7). Thus, figures describing the objective function, qe and te, are only shown for this dataset.



(a) (b) (c)

Fig. 3. FBSOM (dashed line) and FRSOM (continuous line) behavior on Hepta dataset over one run of 10 iterations: (a) Objective function, (b) quantization error and (c) topographic error.

As mentioned before, for a given stimulus all neurons are winners to some degree. Neurons with high firing strength to stimulus will have the highest membership while neurons that have a weaker response will have a lower membership. The first few iterations of FRSOM will assign crisp memberships since q is very small. In other words, FRSOM's first few iterations resemble the RSOM behavior, meaning we start with a winning neuron. As time progresses and the fuzzifier value increases the memberships become fuzzier.

To illustrate this phenomenon for a given stimulus, we need to have a good understanding of the membership distribution among neurons. To do that, we will construct what we call the HL-matrix. One HL-matrix is constructed for one stimulus and has the same dimensions of the U-matrix. Given a stimulus o_k, the value of the ith neuron in the HL-matrix is computed as follows:

$$HL(i) = \sum_{j \in N_i} \left| u_{ik} - u_{jk} \right| \tag{23}$$

This calculation resembles the U-matrix, except we are now looking at the membership level. Of course, the goal is to minimize the values of the HL-matrix. Smaller values mean that adjacent neurons share similar memberships with the stimulus. The first iteration of FRSOM will assign crisp memberships similar to RSOM which result in an HL-matrix as shown in Fig. 4(a). Figure 4(a) demonstrates the "winner take-all" at $t = 2$: one neuron (top right corner) has a membership close to 1 and every other neuron has membership close to 0 (slight information sharing since $q = 1.16$). At $t = 5$ (Fig. 4(b)), q increases to 1.4 and the winning neuron starts sharing information about the stimulus with its neighbors and the membership value of that stimulus gets divided among those adjacent neurons. At $t = 8$ (Fig. 4(c)), where $q = 1.7$ a boundary begins to form. This separates the region in which neurons have high firing strength to the stimulus (see the region in the upper right corner of Fig. 4(c)) from the remaining neurons that have a weaker response. These two regions and the boundary become more defined in the last iteration at $t = 10$, where $q = 2$. We clearly see the L region (region with low active neurons) and the H region (region with high active neurons) separated by the boundary. Hence, communication among neurons intensifies with time and the maximum sharing of information occurs in the last iterations (Fig. 4(d)). The similarities between Fig. 4(d) and the U-matrix in Fig. 2(b) are evident. Figure 4(d) shows the catchment basin containing the set of neurons that responded to that stimulus, which corresponds to the catchment basin labeled R10, R11, and R13 in Fig. 2(b).

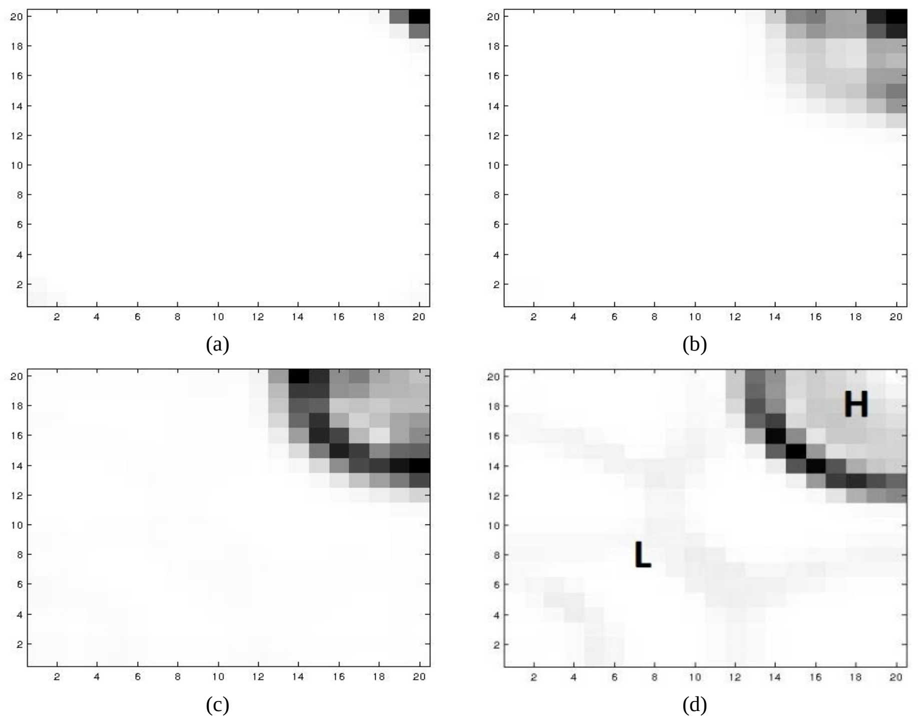


Fig. 4. FRSOM HL-matrix for a given stimulus at various iterations/times (a) $t = 1$ and $q = 1.16$, (b) $t = 5$ and $q = 1.4$, (c) $t = 8$ and $q = 1.7$ and (d) $t = 10$ and $q = 2$.

We can verify that two neighboring neurons represent and sense similar stimuli by inspecting their coefficient vectors. For instance, let us select two neurons from the upper right corner of Fig. 4(d), more specifically, neuron (18, 18) and (19, 16) whose coefficient vectors are shown in Figs. 5(a)–5(b), respectively. It is clear both neurons represent the same stimuli, but with varying weights.

9.2. *Well-Separated Three Gaussians* (*WS3G*)

The WS3G dataset contains three classes that are well separated; this is a fairly easy dataset to cluster (Fig. 1(b), Table 2B). The main goal of this dataset is to demonstrate the behavior of the FBSOM and FRSOM when the inter-cluster distance is large. Properties of the WS3G datasets are presented in Table 4.

Table 4. WS3G dataset properties.

No. of Points	Mean, μ	Std. Dev., σ
500	(1,1)	0.1
500	(1,5)	0.1
500	(4,3)	0.1

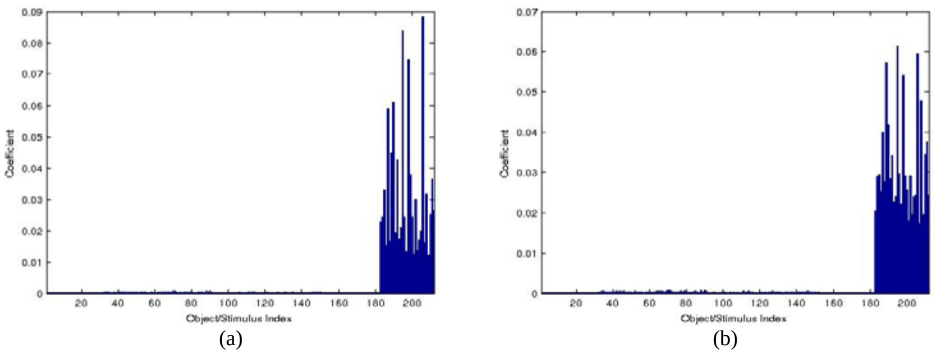


Fig. 5. Neuron coefficients for map location (a) (18, 18) and (b) (19, 16) in Fig. 4(d).

The topographic maps for FBSOM and FRSOM are shown in Figs. 6(a)–6(b), respectively. Both algorithms were able to successfully identify the three clusters. The average qe, te and objective function value are shown in Table 7B. Overall, the average values calculated are very close in both algorithms.

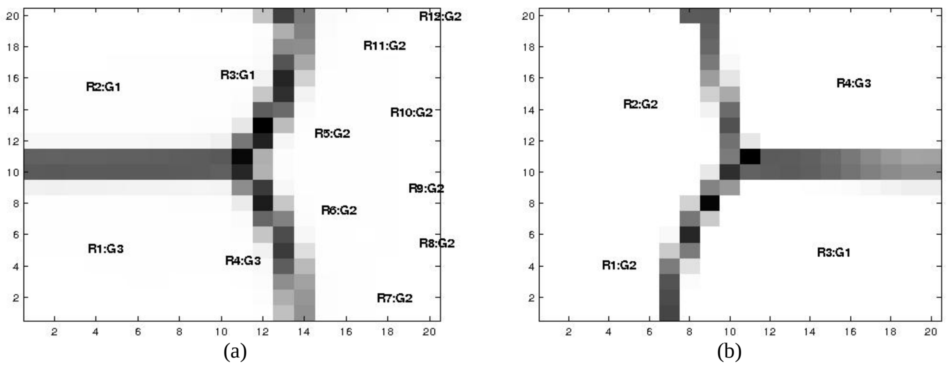


Fig. 6. Topographic maps for WS3G generated by (a) FBSOM, (b) FRSOM.

9.3. *Overlapping Three Gaussians* (*O3G*)

O3G is a similar dataset to WS3G; however, the clusters in O3G have larger variances which causes overlapping. Properties of the O3G datasets are presented in Table 5.

Table 5. O3G dataset properties.

No. of Points	Mean, μ	Std. Dev., σ
500	(1,1)	1
500	(1,5)	1
500	(4,3)	1

The topographic maps produced by FBSOM and FRSOM are shown in Figs. 7(a)–7(b), respectively. Since the three Gaussian clouds overlap, we expect more fuzziness in the maps. Both algorithms identified the three Gaussian clouds correctly as seen in Fig. 7. However, in FBSOM boundaries between the clusters are fuzzier compared to FRSOM. Contrary to the WS3G dataset where we observed how close the errors in both algorithms are, on the O3G the qe, te and the objective function value are a little higher in FBSOM than in FRSOM which may explain the difference between the topographic maps.

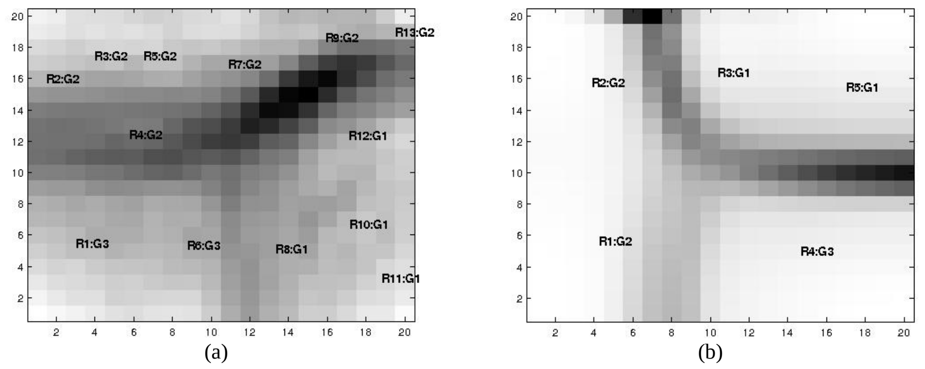


Fig. 7. Topographic maps for O3G generated (a) FBSOM, (b) FRSOM.

9.4. ***Parallel lines***

The lines dataset consists of three parallel lines of 100 points each (Fig. 1(d), Table 2D). The purpose of this dataset is to test whether or not FBSOM and FRSOM would preserve the topology of the data. Indeed, both algorithms are capable of preserving the topology of the lines dataset (Fig. 8). The measured qe and objective function values are close and topographic errors are identical (Table 7D).

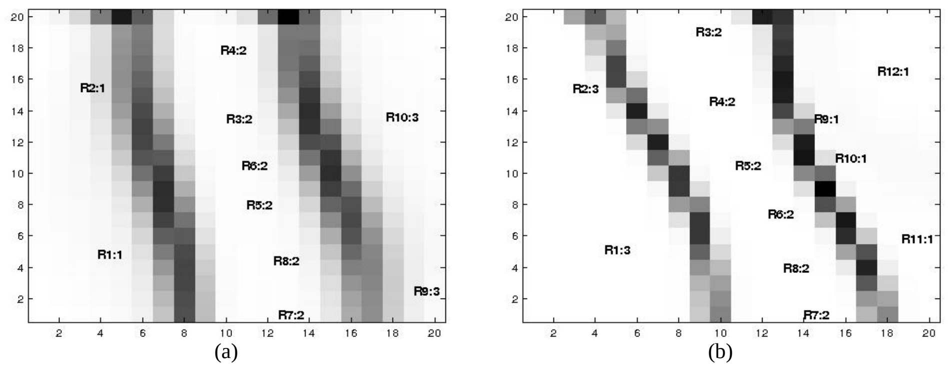


Fig. 8. Topographic maps for the Parallel Lines generated by (a) FBSOM and (b) FRSOM.

9.5. *Congressional Voting Record* (*CVR*)

The CVR dataset, obtained from UCI machine learning repository, contains 435 records. Each record corresponds to a congressman's vote on 16 issues. The class label for a record is either Democrat (D) or Republican (R).

The topographic maps generated by FBSOM and FRSOM are shown in Figs. 9(a)–9(b), respectively. One can say that the maps are almost identical. Also, the errors produced from both maps are very close (Table 7E). In fact, the average and the standard deviation of the topographic errors of both algorithms are identical (see topographic error column in Table 7E).

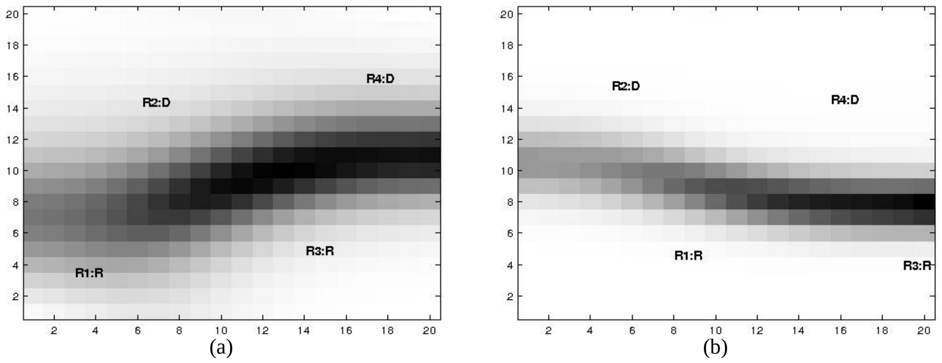


Fig. 9. Topographic maps for the Congressional Voting Record dataset generated by (a) FBSOM and (b) FRSOM.

9.6. *Gene Ontology Dataset* (*GPD194*)

The dataset discussed thus far originally existed in feature vectors and converted to relational data using a distance measure. This is not the case for the GPD194 dataset. It contains 194 sequences of human gene products and was obtained from ENSEMBL 2009[36] and used by Havens *et. al.* to test the ontological SOM (OSOM).[19] Table 6 describes the characteristics of the GPD194 dataset.[37]

Table 6. Charactertistics of the GDP194 Dataset.

ENSEMBL Family ID	F_i = Protein Family	Gene Symbols	No. of Genes	No. of Sequences
339	Myotubularin	MTMR1÷4, MTMR1÷4	7	21
73	Receptor Precursor	FGFR1÷4, RET, TEK, TIE1	7	87
42	Collagen Alpha Chain	COL1A2, COL21A2, COL24A2, COL27A2, COL2A1, COL3A1, COL4A1, COL4A2, COL4A3, COL4A6, COL5A3, COL9A1, COL9A2	13	86

The relational data GPD194 was produced using fuzzy measure similarity (FMS), which is based on Sugeno λ measure. Describing the FMS is outside the scope of this paper, and the reader is referred to Ref. 37 for more details about the GDP194 dataset and the FMS.

FRSOM was able to identify three main regions (Fig. 10), each representing one of the Protein families described in Table 6. The lower right corner of Fig. 10 corresponds to *myotubularin*, the lower left region represents *receptor precursor*, and the upper region corresponds to *collagen alpha chain*. Notice that the *collagen alpha chain* is further divided into three sub-regions: fibril forming collagens, type IV collagens, and fibril associated collagens with interrupted triple helices. Those regions are also observed and discussed in Popescu *et. al.*,[37,38] which provides a validation of the results obtained by FRSOM.

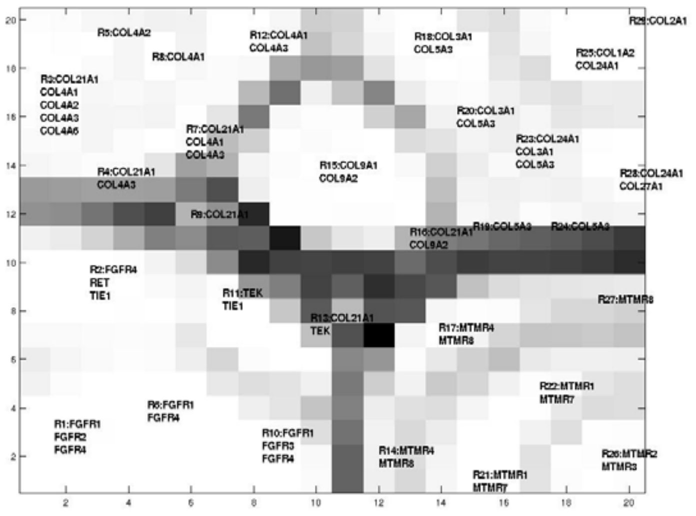


Fig. 10. Topographic map of GPD194 generated by FRSOM.

9.7. *Activity of Daily Livings* (*ADL*)

ADL dataset contains 3,963 patients originated from the 2006–2007 Minimum Data Set (MDS). Every patient has seven individual ADL item scores, each measuring the performance of a given activity. Those activities are: self-performance of bed mobility; transfer between surfaces; locomotion on the nursing unit; dressing; eating; toilet use; and personal hygiene. Patients' ADL scores are generally assessed every three months. For our analysis, we will not address the individual ADL scores, instead focusing on the seven-item ADL score, which is the sum of the individual ADL scores. The score varies from 0 to 28, where 0 indicates complete independence in all seven activities and 28 indicates complete dependence on others for all seven activities. For convenience we will refer to this score as ADL.

Not all patients were assessed every three months. Therefore, the length of the patients' trajectories varies depending on the number of times they were assessed. Some patients may have four ADL scores while others may have 12. Hence, measuring the distance between patients is more difficult than computing the Euclidean distance. So, instead, we need to measure the distances among patients' trajectories represented as time series with uneven length and irregular time steps. To do that, we will use Dynamic Time Warping (DTW) distance measure.[35]

DTW starts with two sequences (patients) $p_1 = p_{11}, p_{12}, \cdots, p_{1a}$ and $p_2 = p_{21}, p_{22}, \cdots, p_{2b}$ of lengths a and b, respectively. We construct an $a \times b$ matrix D where the element (i, j) corresponds to the distance between $p_{1\mathrm{i}}$ and $p_{2\mathrm{j}}$. For instance, $d(p_{1\mathrm{i}}, p_{2\mathrm{j}}) = |p_{1\mathrm{i}} - p_{2\mathrm{j}}|$. Therefore, each element in D corresponds to the alignment between points $p_{1\mathrm{i}}$, $p_{2\mathrm{j}}$ in the sequence p_1 and p_2, respectively. For example, in Fig. 11(a) patient i with four ADL scores is represented on the x-axis and patient j with six ADL scores is on the y-axis. The matrix in Fig. 12(a) shows the pairwise distances among the ADL scores of both patients. The goal of DTW is to minimize the cost of the warping path, which is done by computing a cumulative distance between points $p_{1\mathrm{i}}$, $p_{2\mathrm{j}}$ of the sequences (Fig. 11(b)), which is computed as follows[35]

$$\gamma(p_{1i}, p_{2j}) = d(p_{1i}, p_{2j}) + \min\left(\gamma(p_{1i-1}, p_{2j-1}), \gamma(p_{1i-1}, p_{2j}), \gamma(p_{1i}, p_{2j-1})\right) \quad (24)$$

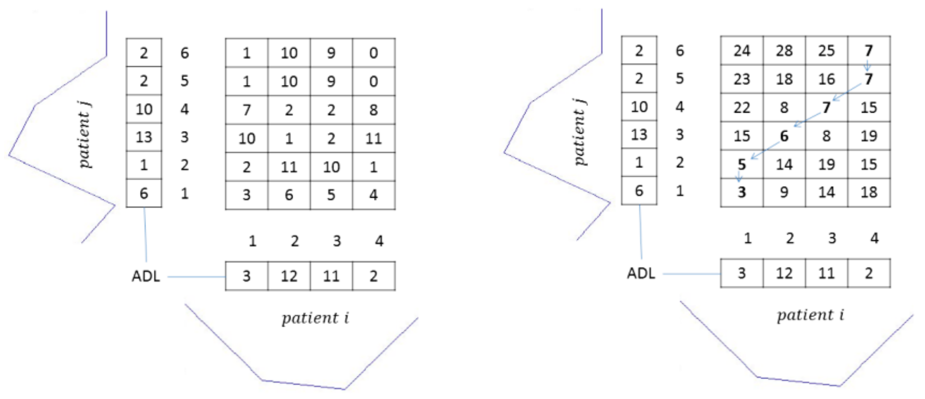


Fig. 11. Distance matrix produced by DTW. (a) Shows the pairwise distance between ADL scores belonging to the two patients (b) is the cumulative DTW distance matrix with the final distance $\gamma(p_4, q_6) = 7$.

Based on the relational data produced using the DTW distance measure, FRSOM generated the topographic map as shown in Fig. 12. The map shows four distinct regions, each region representing a unique set of ADL trajectories. For instance, the upper left corner of Fig. 12 (R2 and R3) represents 471 patients whose ADL trajectory increased before it starting to decline. R5 contains 828 patients whose ADL trajectory decreased and it appears to stabilize. The lower left corner (R1 and R4) represents 1,328 patients whose trajectory consistently increases. R7 contains 780 who exhibit characteristics similar to patients in R1 and R4 except for an apparent improvement in their ADL score after the sudden increase. Lastly, the upper right corner (R6, R8 and R9) represents 556 patients whose ADL trajectory seems to be unstable and fluctuating.

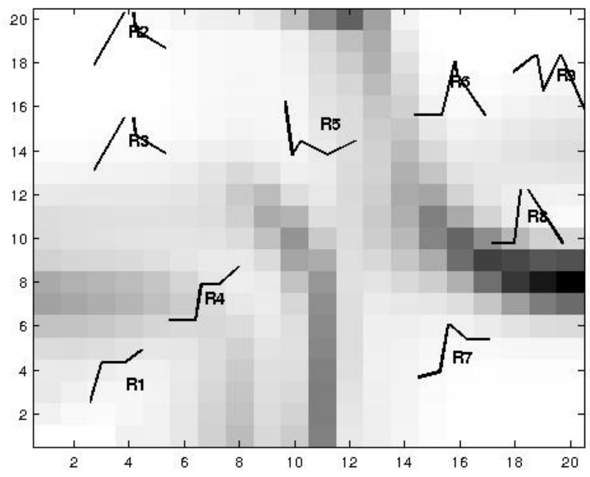


Fig. 12. Topographic map generated by FRSOM using the relational data produced using DTW.

Table 7. Results.

	Dataset	Quantization Error		Topographic Error		Objective Function	
		FBSOM	FRSOM	FBSOM	FRSOM	FBSOM	FRSOM
A	Hepta	0.0123 (1×10^{-4})	0.0087 (4×10^{-5})	0.0198 (0.0187)	0.0292 (0.0341)	2.9 (0.0275)	2.09 (0.0083)
B	Well Separated Gaussians (WS3G)	0.0018 (7×10^{-6})	0.0013 (3×10^{-6})	0.0023 (0.0015)	0.003 (0.0011)	3.18 (0.0145)	2.29 (0.0037)
C	Overlapping Gaussians (O3G)	0.0107 (9×10^{-5})	0.0082 (6×10^{-6})	0.0297 (0.0247)	0.0135 (0.0091)	18.02 (0.1433)	13.72 (0.0072)
D	Parallel Lines	0.0345 (0.0042)	0.0197 (0.0002)	0.0177 (0.0244)	0.0177 (0.0104)	11.54 (1.41)	6.51 (0.088)
E	Congressional Voting Record (CVR)	0.0102 (1×10^{-5})	0.0071 (2×10^{-6})	0.0002 (0.0007)	0.0002 (0.0007)	4.96 (0.0063)	3.44 (0.0004)
F	Gene Ontology (GDP194)	–	0.00004 (2×10^{-5})	–	0.534 (0.316)	–	0.013 (0.0029)
G	Activity of Daily Living (ADL)	–	0.1204 (0.0003)	–	0.171 (0.172)	–	465.96 (7.774)

A summary of the FBSOM and FRSOM performance, the average quantization and topographic errors and objective function value are estimated from 10 runs of the algorithms. The value in parentheses represents the standard deviation.

10. Conclusion

The FRSOM algorithm proposed in the paper introduced few extensions to the RSOM.[18] First, FRSOM incorporates a monotonically increasing fuzzifier and monotonically decreasing neighborhood kernel. We demonstrated this concept using the HL-matrix, where neurons that exhibit strong responses to a given stimuli are separated by a boundary from the neurons that display a weaker response. Second, FRSOM includes visualization and summarization capabilities, which we demonstrated using two real relational datasets.

FRSOM acts on relational data that describes the pairwise dissimilarities among objects, but contrary to RSOM or the classical SOM, by incorporating fuzziness to the neurons, the notion of BMU no longer exists in FRSOM. Instead, every object is associated with a neuron with varying grade of membership. FRSOM's initial iterations resemble the classical crisp SOM since the fuzzifier is small. In other words, FRSOM begins as a "winner take-all" paradigm and as time progresses and the value of the fuzzifier increases, neighboring neurons begin to share and communicate information about the stimuli they sense.

We employed five datasets (some from FCPS and the UCI repository) to test the performance of FBSOM and FRSOM algorithms. We found that both the FBSOM and FRSOM have similar topographic maps and very close error rates. In few cases, the error rates were identical. These results are expected since FRSOM is the relational dual of the FBSOM. Additionally, we tested the performance of the FRSOM using two real relational datasets, Gene Ontology and Activity of Daily Living.

Relational data clustering and visualization, such as FRSOM, are two of the effective approaches to handle data that do not exist in vectorial form or if it is impractical to represent objects as feature vectors. Healthcare data is one example where describing patients in feature vectors is not practical. Patient medical records contain large amount of information such as diagnosis and procedure codes, medications and lab results. Encoding those information using feature vectors will results in a sparse dataset since every patient has only few diagnosis, procedures, etc. A more practical alternative is to measure patient distance using approaches, such as ontology based distances,[37,39] then using algorithms such as FRSOM that utilizes the pre-computed distances for clustering and visualization. However, algorithms such as FRSOM are inherently complex, computationally expensive, and most importantly they lack, the ability to handle large relational data. For instance, the time complexity of FRSOM is $O(c^2n^3t)$, where c is the number of neurons, n is the number of objects, and t is the number of iterations. The memory complexity is $O(n^2 + (c * n)^3)$, where n^2 is the memory complexity of the relational matrix R, $(c * n)^3$ is the complexity for storing the distance matrix (8), partition matrix (12) and the coefficient vectors (16). For example, if $n = 10{,}000$ and $c = 400$, the memory usage is estimated to be 2.9GB. Although RSOM was proposed for large dissimilarity datasets,[40] direct applicability of that technique to FRSOM is not trivial since FRSOM does not assign winning neurons to objects. That said, the FRSOM is a practical algorithm for small to medium size $(n < 10^6)$[41] relational datasets. Therefore, the issue of adapting FRSOM to big relational data remains an important topic to be addressed in future work.

References

1. S. Wold, K. Esbensen and P. Geladi, Principal component analysis, *Chemom. Intell. Lab. Syst.* **2** (1987) 37–52.
2. M. Davison, Introduction to multidimensional scaling and its applications, *Appl. Psychol. Meas.* (1983). at <http://cda.psych.uiuc.edu/mds_509_2013/readings/davidson.pdf>
3. J. B. Tenenbaum, V. de Silva and J. C. Langford, A global geometric framework for nonlinear dimensionality reduction, *Science* **290** (2000) 2319–2323.

4. S. T. Roweis and L. K. Saul, Nonlinear dimensionality reduction by locally linear embedding, *Science* **290** (2000) 2323–2326.
5. M. Belkin and P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, *Neural Comput.* **15** (2003) 1373–1396.
6. L. Churilov and A. Flitman, Towards fair ranking of Olympics achievements: the case of Sydney 2000, *Comput. Oper. Res.* **33** (2006) 2057–2082.
7. W. Kurdthongmee, *Applications of Self-Organizing Maps* (InTech, 2012). doi:10.5772/3464.
8. G. Stegmayer, M. Gerard and D. Milone, Data mining over biological datasets: An integrated approach based on computational intelligence, *IEEE Comput. Intell. Mag.* **7** (2012) 22–34.
9. S. Kaski, T. Honkela, K. Lagus and T. Kohonen, WEBSOM – Self-organizing maps of document collections, *Neurocomputing* **21** (1998) 101–117.
10. R. D. Pascual-Marqui, A. D. Pascual-Montano, K. Kochi, and J. M. Carazo, Smoothly distributed fuzzy c-means: a new self-organizing map, *Pattern Recogn.* **34** 2395–2402 (2001).
11. S.-C. Chi, R.-J. Kuo and P.-W. Teng, A fuzzy self-organizing map neural network for market segmentation of credit card, in *SMC 2000 Conf. Proceedings, 2000 IEEE Int. Conf. Syst. Man Cybern. Cybernetics Evol. to Syst. Humans, Organ. their Complex Interact.* (*Cat. No.* 00CH37166) **5** (IEEE) 3617–3622.
12. P. Vuorimaa, Fuzzy self-organizing map, *Fuzzy Sets Syst.* **66** (1994) 223–231.
13. M. Sap and E. Mohebi, *Hybrid Self Organizing Map for Overlapping Clusters* (Springer-Verlag *Proc. CCIS* (2008), at <http://citeseerx.ist.psu.edu/viewdoc/download? doi = 10.1.1.177.4641&rep=rep1&type=pdf>
14. E. Mohebi and M. N. M. Sap, Hybrid Kohonen self organizing map for the uncertainty involved in overlapping clusters using simulated annealing. in *11th Int. Conf. Comput. Model. Simul. 2009* (IEEE, 2009), pp. 53–58, doi:10.1109/UKSIM.2009.28.
15. E. C.-K. Tsao, J. C. Bezdek and N. R. Pal, Fuzzy Kohonen clustering networks, *Pattern Recogn.* **27** (1994) 757–764.
16. M. Karabulut and T. İbrikci, A fuzzy self-organizing map algorithm for biological pattern recognition, *Expert Syst.* (2010), doi:10.1111/j.1468-0394.2010.00560.x
17. A. Golli, B. Conan-Guez and F. Rossi, A self-organizing map for dissimilarity data, *Classif. Clust. Data Min. Appl.* (2004) 61–68.
18. A. Hasenfuss and B. Hammer, Relational topographic maps. *Adv. Intell. Data Anal. VII* (2007). at <http://www.springerlink.com/index/D0664R20V2L83MX5.pdf>
19. T. C. Havens, J. M. Keller and M. Popescu, Computing with words with the ontological self-organizing map, *Fuzzy Syst. IEEE Trans.* **18** (2010) 473–485.
20. R. J. Hathaway and J. C. Bezdek, Nerf c-means: Non-Euclidean relational fuzzy clustering. *Pattern Recogn.* **27** (1994) 429–437.
21. I. Sledge, J. Bezdek, T. Havens and J. Keller, A relational dual of the fuzzy possibilistic c-means algorithm, in *Int. Conf. Fuzzy Syst.* (IEEE, 2010), pp.1–9, at <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5584846>.
22. A. Ultsch, Emergence in self-organizing feature maps, *Self-Organizing Maps* (*WSOM'07*), *Bielefeld* (2007). at <http://en.scientificcommons.org/38258322>.
23. M. Khalilia and M. Popescu, Fuzzy relational self-organizing maps, in *2012 IEEE Int. Conf. Fuzzy Syst.* (IEEE, 2012), pp. 1–6, doi:10.1109/FUZZ-IEEE.2012.6250833.
24. T. Kohonen, The self-organizing map, *Neurocomputing* (1998), at <http://www.science-direct.com/science/article/pii/S0925231298000307>
25. T. Heskes, Self-organizing maps, vector quantization, and mixture modeling, *IEEE Trans. Neural Netw.* **12** (2001) 1299–1305.
26. F. Moutarde and A. Ultsch, U*F clustering: a new performant "cluster-mining" method based on segmentation of self-organizing maps, in *Work. Self-Organizing Maps* (2005), at <http://hal.archives-ouvertes.fr/hal-00435726>
27. A. Ultsch, Clustering with SOM: U* C. *Proc. Work. Self- Organ. Maps* 75–82 (2005), at <http://www.citeulike.org/group/2572/article/1304144>

28. K. Kiviluoto, Topology preservation in self-organizing maps, in *Proc. Int. Conf. Neural Networks* **1** (IEEE) 294–299.
29. A. Ultsch, Maps for the visualization of high-dimensional data spaces, in *Proc. Work. Self Organ. Maps* (2003), at <http://www.informatik.uni-marburg.de/~databionics/papers/ ultsch-03maps.pdf>
30. J. Vesanto and E. Alhoniemi, Clustering of the self-organizing map, *IEEE Trans. Neural Netw.* **11** (2000) 586–600.
31. J. A. F. Costa and M. L. de Andrade Netto, A new tree-structured self-organizing map for data analysis, in *IJCNN'01. Int. Jt. Conf. Neural Networks. Proc.* (Cat. No.01CH37222) **3** (IEEE) 1931–1936.
32. J. A. F. Costa and M. L. de Andrade Netto, Cluster analysis using self-organizing maps and image processing techniques, in *IEEE SMC'99 Conf. Proceedings, 1999 IEEE Int. Conf. Syst. Man, Cybern.* (Cat. No.99CH37028) **5** (IEEE) 367–372.
33. F. Meyer, Topographic distance and watershed lines. *Signal Processing* **38** (1994) 113–125.
34. L. Shafarenko, M. Petrou and J. Kittler, Automatic watershed segmentation of randomly textured color images, *IEEE Trans. Image Process.* **6** (1997) 1530–1544.
35. D. Berndt and J. Clifford, Using dynamic time warping to find patterns in time series. in *AAAI Work. Knowl. Discov. Databases* (1994) 359–370.
36. T. J. P. Hubbard, *et al.*, Ensembl 2009, *Nucleic Acids Res.* **37** (2009) D690–7.
37. M. Popescu, J. M. Keller and J. A. Mitchell, Fuzzy measures on the gene ontology for gene product similarity, *IEEE/ACM Trans. Comput. Biol. Bioinform.* (2006) 263–274.
38. T. C. Havens, J. M. Keller, M. Popescu and J. C. Bezdek, Ontological self-organizing maps for cluster visualization and functional summarization of gene products using Gene Ontology similarity measures, in *2008 IEEE Int. Conf. Fuzzy Syst.* (*IEEE World Congr. Comput. Intell.*) (IEEE, 2008), pp. 104–109, doi:10.1109/FUZZY.2008.4630351.
39. G. B. Melton *et al.*, Inter-patient distance metrics using SNOMED CT defining relationships, *J. Biomed. Inform.* **39** (2006) 697–705.
40. B. Hammer and A. Hasenfuss, Topographic mapping of large dissimilarity data sets, *Neural Comput.* **22** (2010) 2229–2284.
41. T. C. Havens, J. C. Bezdek, C. Leckie, L. O. Hall and M. Palaniswami, Fuzzy c-means algorithms for very large data, *IEEE Trans. Fuzzy Syst.* **20** (2012) 1130–1146.