Fuzzy Relational Self-Organizing Maps

Mohammed Khalilia
Computer Science
University of Missouri
Columbia, Missouri
makkx9@mail.missouri.edu

Mihail Popescu
Health Management and Informatics
University of Missouri
Columbia, Missouri
PopescuM@missouri.edu

***Abstract*—In this paper we propose a novel fuzzy relational self-organizing map algorithm (FRSOM) that can be used to map a set of *n* objects described by pair-wise dissimilarity values to a two dimensional lattice structure. FRSOM generates a fuzzy membership matrix replacing the crisp best-matching unit matrix in the regular relational SOM (RSOM). We found that FRSOM discovers hard to find substructures in the data that present a challenge to the crisp relational SOM. Furthermore, we observed a triple relationship that seems to exist among the number of data points in the training data, map size and the fuzzifier *m*. We compare FRSOM and RSOM using several synthetic datasets.**

Keywords: Fuzzy relational self-organizing maps; self organizing maps; relational data; clustering; fuzzy sets

I. INTRODUCTION

NUMEROUS clustering techniques have been proposed in the literature to handle relational data. Relational data describe the relations (dissimilarity or similarity) between objects, rather than characterizing the objects themselves. In [1], Hathaway and Bezdek developed Non-Euclidean Relational Fuzzy *c*-Means (NERFCM) to cluster a set of *n* objects described by pair-wise dissimilarity values. Another fuzzy relational clustering algorithm, any relational clustering algorithm (ARCA), based on fuzzy *c*-means is proposed in [2]. Essentially, ARCA treats relational data as object data and employs a fuzzy *c*-means to obtain the clusters. Consequently, ARCA does not impose any restrictions on the relational data matrix and it is reportedly less prone to the dimensionality curse than NERFCM. In addition to relational clustering, data visualization techniques, such as Self-Organizing Map (SOM), were also adapted to relational data.

SOM [3] is an unsupervised learning technique aimed at data exploration and visualization. SOM projects an *s*-dimensional input space into a low dimensional, usually two dimensional, lattice or grid of neurons. Several crisp relational SOM algorithms were developed [4, 5]. In [4], the authors described each neuron by a codebook that represents a subset of input vectors. Essentially, the input data is clustered in *c* clusters, where *c* is the number of SOM neurons. The codebooks (clusters) are updated using a cost function that resembles the c-means objective function and accounts for both the relational data and the neighborhood topology. In [5], the authors extended Neural Gas (NG) and SOM to relational data given by pair-wise dissimilarity values using the relational dual of fuzzy c-means clustering derived in [1, 6]. Both RSOMs proposed in [4, 5] are based on a batch update approach. Our fuzzy extension to RSOM presented in this paper is based on the approach presented in [5].

A fuzzy SOM algorithm for object data was proposed in [7], which is in some sense a regularization of the fuzzy c-means (FCM) algorithm. The fuzzy SOM in [7] is based on a cost function that is derived by introducing two modifications to the generalized FCM [8-9]. 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 and help preserve the topological structure of the data. A fuzzy SOM based on fuzzy inputs and fuzzy weights for market segmentation of credit cards is proposed in [10]. FCM is applied for fuzzy clustering to identify the ambiguous sampled data located near the border between the clusters. In [11], 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 the each rule is a singleton, thus the algorithm maps the *s*-dimensional input space to a one dimensional output space. In [12], a hybrid SOM is proposed for overlapping clusters to predict 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 [13], the same authors proposed a variation to [12] in which a two-level stage SA-Rough SOM (Simulated Annealing Rough Self-Organizing Map) was proposed.

The algorithm proposed in this paper, FRSOM, is a combination of the relational SOM developed in [5] and NERFCM presented in [1]. In [5] the original SOM was reformulated to resemble, to some extent, the c-means clustering algorithm. The main SOM features, i.e. the updating of the winning neuron neighborhood and its "push" toward the input vector were preserved. However, similar to c-means, SOM will use a crisp best-matching unit (BMU) matrix that we will refer to as "the membership matrix", u.

The remainder of this article is structured accordingly: Section II describes the FRSOM algorithm. Section III investigates some of the properties of the proposed method and Section IV gives some conclusions and ideas for future research.

II. METHODS

A. Crisp and fuzzy memberhip matrices

For a set of n objects $\{o_k\}_{k\in[1,\ n]}$ we define the crisp membership in c clusters as:

$$u_{ik} \in \{0,1\} \text{ for } 1 \le i \le c \text{ and } 1 \le k \le n \quad (1)$$

$$\sum_{i=1}^{c} u_{ik} = 1, \qquad \forall k \quad (2)$$

where u_{ik} is the membership of object o_k in cluster i. In the crisp SOM case, i represents an individual neuron and c is the total number of neurons in the lattice. The use of crisp memberships forces object k to either be assigned ($u_{ik} = 1$) or not ($u_{ik} = 0$) to neuron i.

By preserving the constraint in (2), we can relax the constraint from (1)

$$u_{ik} \in [0,1] \text{ for } 1 \le i \le c \text{ and } 1 \le k \le n \quad (3)$$

to describe a set of fuzzy c-partitions for the n objects, where u_{ik} represents the degree to which object o_k belongs to cluster or neuron i. FRSOM uses the fuzzy partition matrix u described in (3).

B. Object and relational data

In clustering, one can encounter two main types of datasets: object and the relational. The object data, O, can be described as:

$$O = \{o_1, \dots, o_n\} \subset R^s, \quad (4)$$

where each feature vector o_k is s-tuple of real numbers. The other format of numerical data is an $n \times n$ relational data matrix $D = [D_{ij}]$, where D_{ij} is a measure of the relationship between o_i and o_j. The relationship can be either a similarity or dissimilarity relation. The dissimilarity relation, D, satisfies the following conditions:

$$D_{jj} = 0, \forall\, j = 1, \dots, n \quad (5)$$

$$D_{jk} > 0, \text{for } k = 1, \dots n \text{ and } j = 1, \dots, n \quad (6)$$

$$D_{jk} = D_{kj}, \text{for } k = 1, \dots n \text{ and } j = 1, \dots, n \quad (7)$$

The FRSOM algorithm is a modification of the original SOM presented in [3]. First, it is adapted to handle relational data as proposed by [5]. And second, the crisp membership matrix is substituted with the fuzzy membership matrix based on the formulation of the NERFCM algorithm [1].

C. Batch self-organizing map algorithm (BSOM)

BSOM consists of $c \times n$ neurons located on a two dimensional grid, where each neuron i is represented by an s-dimensional weight vectors $\boldsymbol{w} = \{w_1, \dots, w_c\}$, where s is the number of dimensions of the input data. Unlike the online version of SOM, in the batch version the whole dataset is presented to the map before any updates to the weights take place. The main steps of the BSOM algorithm are as follows:

(BSOM1) Objects $O = \{o_1, \dots, o_n\}$ are presented to the map all at the same time.

(BSOM2) Initialize the weight vectors, $\boldsymbol{w}$. For instance, w can be assigned randomly generated values, by selecting random objects from the dataset, or by selecting the c objects that have the furthest distance from each other.

(BSOM3) Distances between the objects, O, and the weight vectors, $\boldsymbol{w}$, are computed. The neuron with the minimum distance to object o_k is assigned as the wining neuron and it is denoted with $I^*(o_k)$.

$$I^*(o_k) = \arg\min_i \|w_i - o_k\| \quad (8)$$

$\|.\|$ is the distance measure which is usually the Euclidean distance. The assignment of the winning neuron is constraint with (1) and (2), where the degree of membership between w_i and o_k is given by

$$u_{ik} = \begin{cases} 1 & \text{if } I^*(o_k) = i \\ 0 & \text{otherwise} \end{cases}, \quad (9)$$

which is similar to c-means membership assignment.

(BSOM4) After repeating the step (BSOM3) for all the input dataset, all the weight vectors are updated using (10).

$$w_i(t+1) = \frac{\sum_{k=1}^{n} h\big(n(I^*(o_k), i)\big) * o_k}{\sum_{k=1}^{n} h\big(n(I^*(o_k), i)\big)} \quad (10)$$

Where

$-t$ denotes the training step.

$-n(I^*(o_k), i)$ is the distance of a path connecting the winning neuron index, $I^*(o_k)$, and neuron i in the lattice structure.

$-h\big(n(I^*(o_k), i)\big)$ is the neighborhood function around the wining neuron $I^*(o_k)$, which is defined as:

$$h\big(n(I^*(o_k), i)\big) = \eta(t) * \exp\left(\frac{-n(I^*(o_k), i)}{2\sigma^2(t)}\right) \quad (10)$$

$\eta(t) \in [0,1]$ is the learning rate and $\sigma(t)$ is the neighborhood radius, which decreases with time.

(BSOM5) Repeat the steps (BSOM3) and (BSOM4) t_{max} times.

D. Fuzzy relational self-organizing map algorithm (FRSOM)

In this section we will explain the FRSOM in details and highlight the differences between BSOM and FRSOM. The BSOM will be modified so that it can visualize relational data and compute fuzzy membership assignments. More specifically, FRSOM will differ in the (BSOM1) step where the input is relational data, D, instead of the object data O. In (BSOM3) step the Euclidean distance will substituted with the distance given in (12) because in relational clustering we assume that there exist points x_k such that $D_{ik} = \|x_i - x_k\|^2$, where D_{ik} is a measure of the dissimilarity between the points x_i and x_k. In (BSOM4) step the fuzzy memberships will be

incorporated in (10) to update the new weight vectors. In addition to those changes, fuzzy memberships will be computed to replace the crisp memberships (9). The main steps of the FRSOM algorithm are given below.

(FRSOM1) Relational input vectors $D = \{D_1, \dots, D_n\}$ are presented to the map all at the same time, where D_1 is the first row of the relational data.

(FRSOM2) As in the BSOM, FRSOM starts by initializing the codebook vectors, $\boldsymbol{w}$. Initialization can be done by randomly selecting random vectors from the relation D, or by randomly generating codebook vectors. However, just as in any other clustering algorithms, random initialization tends to produce inconsistent results. One can mitigate this problem by initializing the codebook vectors with data points that are furthest from each other.

(FRSOM3) In addition to the codebook vectors, $\boldsymbol{w}$, cluster prototypes or cluster centers, $\boldsymbol{\alpha}$, are initialized where α_i is a normalized w_i as:

$$\alpha_i = \frac{w_i}{\sum_{k=1}^{n} w_{ik}} \tag{11}$$

(FRSOM4) Distances between the relational vectors, D, and the cluster prototypes, $\boldsymbol{\alpha}$, are computed. Since the data is given in terms of pair-wise dissimilarity, the distance $\|w_i - x_k\|^2$ is computed by [1, 5]

$$\|w_i - x_k\|^2 = d_{ik} = (D \cdot \alpha_i)_k - \frac{(\alpha_i^t \cdot D \cdot \alpha_i)}{2}, \quad \text{for } 1 \le i \le c \text{ and } 1 \le k \le n. \tag{12}$$

Interested readers are referred to [5] for a complete proof of (12).

(FRSOM5) Calculate the degree of association, u_{ik}, between w_i and x_k using the NERFCM approach [1], which is

$$u_{ik} = \frac{1}{\sum_{j=1}^{c} \left(\frac{d_{ik}}{d_{jk}}\right)^{\frac{2}{m-1}}} \quad \text{for } 1 \le i \le c \text{ and } 1 \le k \le n. \tag{13}$$

where $m > 1$ is the fuzzifier.

(FRSOM6) The neuron with the minimum distance to D_k (14) or with the maximum membership to D_k (15) is assigned as the wining neuron and it is denoted with $I^*(x_k)$.

$$I^*(x_k) = \arg\min_i d_{ik} \tag{14}$$

$$I^*(x_k) = \arg\max_i u_{ik} \tag{15}$$

(FRSOM7) After repeating steps (FRSOM4) and (FRSOM5) for every training data point and for every neuron, the new weights are updated based on the membership, u_{ik}, as follows:

$$w_i(t+1) = \frac{\sum_{k=1}^{n} h\big(n(I^*(x_k), i)\big) * (u_{ik}^m * D_k)}{\sum_{k=1}^{n} h\big(n(I^*(x_k), i)\big) * u_{ik}^m} \tag{16}$$

(FRSOM8) Update the cluster prototypes, α, using the newly computed weight vectors

$$\alpha_i(t+1) = \frac{w_i(t+1)}{\sum_{k=1}^{n} w_{ik}(t+1)} \tag{17}$$

(FRSOM9) Repeat the steps (FRSOM4) through (FRSOM8) t_{max} times.

A summary of the FRSOM algorithm is given below (Algorithm 1).

Algorithm 1: Fuzzy Relational SOM Algorithm (Batch)

►**Data:** $D_j, j = 1, \dots, n$ where D_j is the j-th vector the relational data.

►**Initialization:**
Set $m > 1$
$t \leftarrow 0$
Initialize the codebook weight vectors:
$w = \{w_1, \dots, w_c\} \in R$
Initialize the prototypes, σ, with (11)

►**while** $t < t_{max}$ **do**
for k=1 **to** n **do**
Compute the distance, d_{ik}, between D_k and α_i, $\forall i$ with (12)
Find the winning neuron, $I^*(x_k)$, with (14) or (15)
Compute the membership, u_{ik}, between D_k and α_i, $\forall i$ with (13)
endfor

Update the codebook weight vectors, w, with (16)
Update the prototypes vectors, σ, with (17)
$t \leftarrow t + 1$

endwhile

III. Experimental Results

In this section we present FRSOM results on two synthetic datasets. First, we will demonstrate the performance of FRSOM while varying the fuzzifier, m. This experiment helps choose the value of m that will be used in subsequent experiments. Second, we will discuss the relationship among the number of training data points, map size and the fuzzifier m. Third, we compare crisp vs. fuzzy relational SOM using a non-overlapping and overlapping Gaussian clouds. The FRSOM algorithm has been implemented based on the SOM Toolbox for MATLAB [14].

A. FRSOM behavior for different fuzzifiers

Just as in any other fuzzy clustering algorithms, lower values of m will produce crisp cluster memberships while higher values of m will result in fuzzier membership values. In this section we show, for several values of the fuzzifier m={1.03, 1.5, 1.8, 2}, that FRSOM follows the same behavior.

In our experiments we used two synthetic Gaussian datasets: 3 well separated Gaussians (denoted as WS3G, see Tables I and Fig. 3.a) and 3 overlapping Gaussians (denoted as O3G, see Table II and Fig. 3.b).

Also, FRSOM is initialized to codebook weight vectors, *w*, using the 100 data points that are furthest from each other and with a 10 x 10 hexagonal grid. The prototypes, α, for FRSOM are initialized using (11).

Fig. 2 shows the FRSOM maps obtained for various values of *m* on WS3G data. We see that the lower the *m* is the more crisp the membership and the SOM maps are (Fig. 2.a and 2.b). The reason for the high similarity between the maps of m = 1.03 and m = 1.5 could be due to the high clusterness of the data. However, as the value of *m* increases the membership and the SOM map start getting fuzzier (Fig. 2.c and 2.d). Although Fig. 2.d is somewhat fuzzy, we can still see two blue patches on the upper and lower left corner and another blue patch spanning the right side of the map. The high fuzziness at m = 2 might be attributed to the SOM map size. For instance, maybe it is more appropriate to choose a larger map size when m = 2. However, the map can't be larger than a certain fraction of the number of data points. Thus, one has to find a balance among the fuzzifier, map size and number of data points in the training dataset. In our case, we found that the value *m*=1.1 produces the most consistent results and we will use it for subsequent experiments.

B. *Relationship among map size, fuzzifier and number of data points*

In the crisp SOM one has to strike a balance between the number of data points in the training data and the map size. The number of neurons, *c*, must be at least smaller than the number of data points, *n*. In FRSOM, another dimension to this relationship is added, which is the fuzzifier, *m*. The map size is likely to be a function of *m* and *n.*

$$mapsize = f(m,n) \tag{14}$$

To illustrate this relationship, observe Fig. 2.b and Fig. 1. In Fig. 1.b the map size is 10 × 10 and m = 1.5 while in Fig. 2 the map size is 20 × 20 and the m = 1.5. In this case, we see that as the number of neurons approaches the number of data points the SOM map becomes fuzzier. Intuitively, this behavior is similar to a map stretching effect. However, we leave the complex relationship between *n*, *m* and map size for further research. For now, we will use a 10 × 10 map size in subsequent experiments.

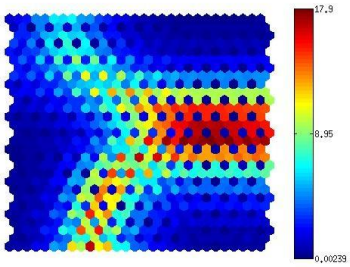


Figure 1. FRSOM on WS3G with grid size of 20x20 and *m*=1.5

C. *Three Gaussian clouds*

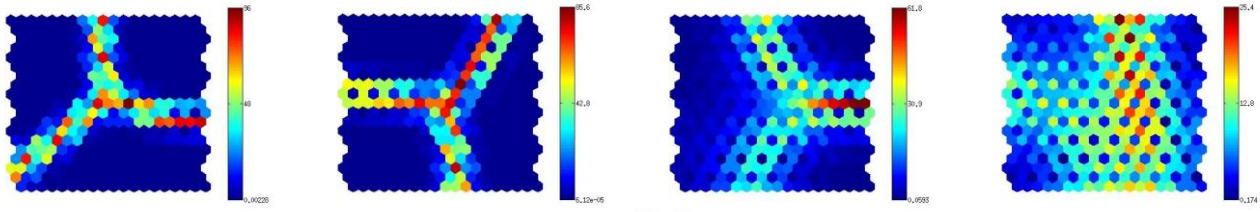
In this section we compare the crisp relational SOM [5] and the fuzzy relational SOM proposed in this paper. We will show that both algorithms have similar performance when the data has high clusterness (WS3G). However, FRSOM performs better when the data has low clusterness (O3G).

1) *High clusterness data (WS3G data)*

To illustrate the performance similarity between the crisp and fuzzy relational SOM we used the well separated three Gaussian clouds, WS3G. Table 1 outlines the properties of each cloud in the WS3G data.

TABLE I. WELL SEPARATED THREE GAUSSIAN CLOUDS PROPERTIES, WS3G

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



(a) (b) (c) (d)

Figure 2. RFSOM Behavior for different fuzzifiers, (a) *m*=1.03 (b) *m*=1.5 (c) *m*=1.8 (d) *m*=2

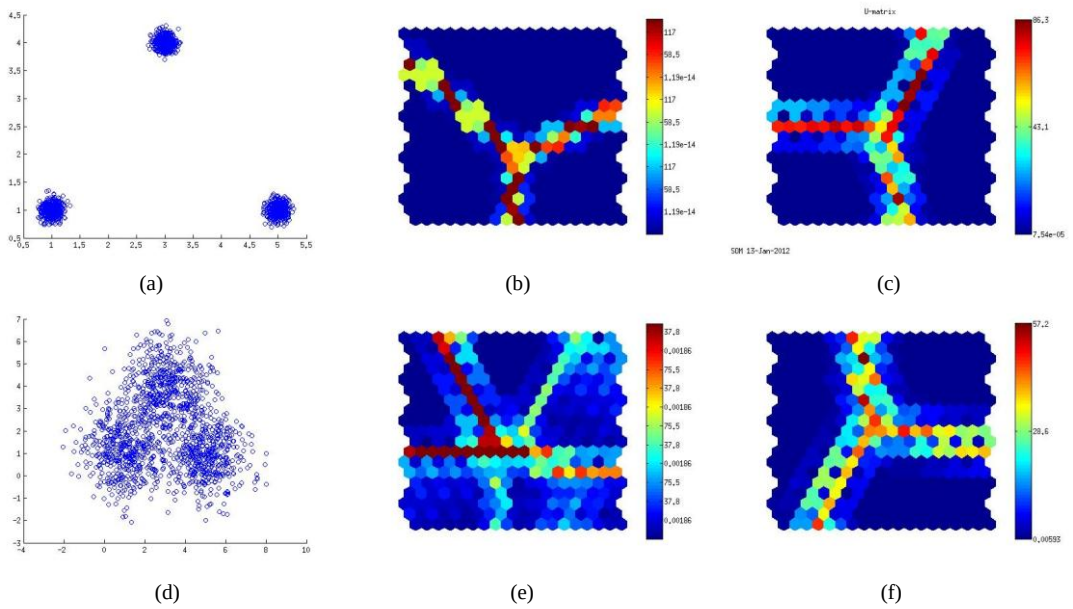


Figure 3. Crisp vs. Fuzzy Relational SOM (a) well separated three Gaussian clouds, WS3G (b) performance of RSOM on WS3G data (c) performance of FRSOM on WS3G data (d) overlapping three Gaussian clouds clouds, O3G (e) performance of RSOM on O3G data (f) performance of FRSOM on O3G data

The FRSOM algorithms are trained on objects represented by the relational data, $D = [D_{ij}]_{1500x1500}$, where D consists of the squared Euclidean distance of the Gaussian clouds dataset. Fig. 3.b and Fig. 3.c show images of the trained crisp relational SOM and fuzzy relational SOM, respectively. Both algorithms are initialized to the same codebook weight vectors using the 100 data points that are furthest from each other and with a 10 × 10 hexagonal grid. Additionally, the prototypes are initialized using (11) and the fuzzifier for FRSOM is set to m=1.1. Both algorithms are able to correctly show the groupings of the data points. The other observation to make here is that FRSOM at m=1.01 reduces to the crisp relational SOM.

2) Low clusterness data (O3G)

In this experiment we use the overlapping Gaussian data, O3G. Table 2 outlines the properties of each cloud in the O3G data.

TABLE II. OVERLAPPING THREE GAUSSIAN CLOUDS PROPERTIES, O3G

No. of points	Mean, μ	Std. dev., σ
500	(1,1)	1
500	(1,5)	1
500	(4,3)	1

Figure 3.e and Fig. 3.f show images obtained for crisp and fuzzy relational SOM, respectively. It is clear the FRSOM was able to capture the structure of the data more accurately than the crisp relational SOM. This experiment suggests that FRSOM can discover subtle substructures in relational data.

IV. CONCLUSIONS

In this paper we presented a fuzzy relational SOM (FRSOM) based on the crisp relational SOM found in [5] and the NERFCM formulation [1]. We have studied the behavior of SOM on two different synthetic datasets for different fuzzifier values. We also suggested that there might be an interesting relationship among the size of the data set, map size and the fuzzifier m.

From the experiments in Section III, we can conclude the following. First, when $m \to 1$ FRSOM reduces to the crisp relational SOM, which shows that it is a specific type of FRSOM. This observation was also pointed out in the NERCM algorithm [1]. Second, the crisp and fuzzy relational SOM with m= 1.1 performed similar when presented with dataset that is well separated. However, when the dataset has overlapping clusters, FRSOM outperformed its crisp version. We suggest that the more subtle substructures in the data might be visualized with FRSOM.

However, we still have challenging FRSOM problems ahead of us to solve. One of these problems is determining the map size as function of the number of the training data patterns and the fuzzifier. Another area that needs attention in relational clustering is the size of the relational data matrix. Relational matrices, especially for real biomedical datasets, tend to be large which rapidly increase the need for computational resources. One way to address this issue is by applying dimensionality reduction techniques such as kernel clustering proposed in [15]. And lastly, we are planning on utilizing FRSOM with real datasets encountered in our medical informatics research.

REFERENCES

[1] R. J. Hathaway and J. C. Bezdek, "NERF c-means: Non-Euclidean relational fuzzy clustering," Pattern recognition, vol. 27, pp. 429-437, 1994.

[2] P. Corsini, B. Lazzerini, and F. Marcelloni, "A new fuzzy relational clustering algorithm based on the fuzzy C-means algorithm," *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, vol. 9, pp. 439-447, 2005.

[3] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, pp. 1464-1480, 1990.

[4] A. Golli, B. Conan-Guez, and F. Rossi, "A self-organizing map for dissimilarity data," *Classification, Clustering, and Data Mining Applications*, pp. 61-68, 2004.

[5] A. Hasenfuss and B. Hammer, "Relational topographic maps," Advances in Intelligent Data Analysis VII, pp. 93-105, 2007.

[6] R. J. Hathaway, J. W. Davenport, and J. C. Bezdek, "Relational duals of the c-means clustering algorithms," *Pattern recognition*, vol. 22, pp. 205-212, 1989.

[7] R. Pascual-Marqui, A. Pascual-Montano, K. Kochi, and J. Carazo, "Smoothly distributed fuzzy c-means: a new self-organizing map," *Pattern recognition*, vol. 34, pp. 2395-2402, 2001.

[8] J. C. Bezdek, "Fuzzy mathematics in pattern classification," 1973.

[9] J. C. Bezdek, Pattern recognition with fuzzy objective function algorithms: Kluwer Academic Publishers, 1981.

[10] S. C. Chi, R. J. Kuo, and P. W. Teng, "A fuzzy self-organizing map neural network for market segmentation of credit card," 2000, pp. 3617-3622 vol. 5.

[11] P. Vuorimaa, "Fuzzy self-organizing map," Fuzzy Sets and Systems, vol. 66, pp. 223-231, 1994.

[12] M. Sap and E. Mohebi, "Hybrid Self Organizing Map for Overlapping Clusters," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 1, pp. 11-20, 2008.

[13] E. Mohebi and M. Sap, "Hybrid kohonen self organizing map for the uncertainty involved in overlapping clusters using simulated annealing," 2009, pp. 53-58.

[14] Neural Networks Research Centre, Helsinki University of Technology, SOM Tool-box, http://www.cis.hut.fi/projects/somtoolbox/

[15] R. Chitta, R. Jin, T. C. Havens, and A. K. Jain, "Approximate kernel k-means: solution to large scale kernel clustering," 2011, pp. 895-903.