Experimental Study on Spectral Clustering

Ricardo Filipe Silva
ricardofcsasilva@gmail.com
Departamento de Informática
Faculdade de Ciências e Tecnologia
Universidade Nova de Lisboa

March 23, 2012

1 Introduction

Clustering is the process of grouping a set of objects into clusters in such a way that objects in the same cluster have high similarity with each other, but are as dissimilar as possible to objects located in other clusters [7]. Moreover, it is an unsupervised classification technique widely used (and useful) in exploratory data analysis like in data mining, pattern recognition and image processing.

In the last few years, a promising technique for grouping problems has emerged: spectral clustering. Spectral clustering algorithms perform data clustering using the eigenvectors of a similarity matrix (or affinity matrix) derived from the data [11]. This approach has been gaining more and more interest due to its high performance in data clustering and simplicity in implementation [6]. However, there are some open issues regard this technique. On one hand, the similarity matrix from which the spectral clustering approach works can be done from different ways using different parameters. This factor has a huge impact on clustering result and a little is known about theoretical implications [6]. That is, different datasets have different inherent characteristics and each type of similarity matrix deals better with some characteristics than others. So, the usage of the appropriate type of similarity matrix (given a dataset) is crucial in order to acquire good performance from spectral clustering algorithms.

On the other hand, in general, spectral clustering algorithms have the number of clusters as an input. However, as an unsupervised technique, a minimum domain knowledge input should be required from the user [5].

In this work, an experimental study was performed in order to evaluate: (*i*) the choice of an adequate similarity matrix setting as input of the NJW algorithm; (*ii*) the strategy to automatically determine the number of clusters in data proposed by the DiFUZZY algorithm [3]. This discussion will be supported by an experimental study that involves synthetic and real datasets.

This report is organized as follows. Section 2 presents a set of research literature related to spectral clustering. In Section 3 is presented the experimental study and the discussion associated. Section 4 concludes with some remarks about the study performed.

2 Related Work

In this section, an overview about spectral clustering is given, introducing different types of similarity matrices. At the end of this section, two spectral algorithms are presented which are used on the experimental study.

2.1 Similarity Graphs

Consider a set of data points $x_1, ...x_n$ and some notion of similarity $s_{ij} > 0$ between all pairs of data points x_i and x_j. In spectral clustering, they are initially organized as a graph $G = (V, E)$ where each vertex v_i represents a point x_i where $v_i \in V$, and each edge e_i between two vertices v_i and v_j carries a non-negative weight s_{ij} where $e_i \in E$.

Five types of similarity graphs are presented and discussed in [6]: (*i*) the ϵ-neighborhood graph which connects v_i and v_j if the distance between them is smaller than ϵ; (*ii*) the k-nearest neighbor graph (k-NN) connects v_i and v_j if v_j is among the k-nearest neighbors from v_i **or** v_i is among the k-nearest-neighbors from v_j; (*iii*) the mutual k-NN graph which is similar to the previous one but the condition use a conjunction instead of a disjunction; (*iv*) the fully connected graph simply connect all data points with positive similarity; in the euclidean space, the Gaussian Kernel similarity measure is a good candidate: $S_{ij} = exp(-||x_i - x_j||^2/2\sigma^2)$.

Using different similarity graphs result in different graphs for the same dataset and some of them express better the similarity between data points than others, affecting the clustering result.

When data have clusters with different densities or clusters within irregular background clutter, it is likely that using the ϵ-neighborhood graph or the Gaussian Kernel similarity may lead to a poor results, once the parameters ϵ and σ are global and they have similar role, i.e. controls the width of neighborhood which in such datasets it is not constant. On the other hand, the k-NN can connect points on different scales but depending on k parameter that characteristic can be advantageous or not. Plus, when the data have clusters within irregular background clutter, poor clustering results are probably obtained. Finally, the mutual k-NN graph is well-suited to detect clusters with different densities and with background cluttered. These conclusions are presented in [6], and some of them will be discussed in the experimental study performed (see section 3).

In order to outperform the previous similarity graphs some research is directed to develop new approaches for similarity matrices' construction.

Zelnik-Manor and Perona [10] proposed a new definition of Gaussian Kernel: $S_{ij} = exp(-d^2(x_i, x_j)/\sigma_i\sigma_j)$, i.e. instead of having a global parameter σ, they propose to calculate a local scaling parameter σ_i for each data point x_i, where $\sigma_i = d(x_i, x_K)$, denoting the distance from x_i to its k neighborhood.

With this modification, the authors argue the ability to deal with datasets with clusters at different scales and with irregular background cluttered.

In [11] a fuzzy similarity measure is proposed. The algorithm to construct the fuzzy similarity matrix take as input the cluster prototypes V and partition matrix U obtained by the fuzzy c-means (FCM) [2]. The matrix U contains all the information about the relative closeness of each point x_j with all the prototypes $V_i, i = 1, 2, ..., C$. The membership value in U denotes the degree that a data point belongs to a specific cluster. Therefore, through the membership degree of two data points to a cluster, the authors assume that the edges in the similarity graph have very low weights if the points do not have the same prototypes, and the edges have high weights if the points have the same prototypes. The algorithm to construct the fuzzy similarity graph is the follow:

Input: The partition matrix obtained by FCM, the number t of the nearest prototypes.

- Sort each column of partition matrix U according to the descending order of membership values, then obtain a new matrix U' .

- For each x_i and x_j
  If x_i and x_j has the same nearest prototype.
      s_{ij} = 1.
  Else if x_i and x_j has the same prototype in their t nearest prototypes
      $s_{ij} = max(max(U'_{li}))$ where $(0 < i < m, 0 < j < n)$
  Otherwise, $s_{ij} = 0$.
  Set $s_{ji} = s_{ij}$.

Output: Obtain the fuzzy similarity matrix of the dataset.

The spectral clustering algorithms take as input one similarity matrix. As discussed, there are several ways to construct one and for each type also exist parameters to set, which may have a strong influence on clustering result. In the follow section, it will presented two spectral clustering algorithms.

2.2 Spectral Clustering Algorithms

Recalling that, the goals of clustering is (*i*) to minimize the between-cluster similarity and (*ii*) to maximize the within-cluster similarity. Once we transform the initial data into a graph representation, the goal is to find a partition in the graph such that: the edges between different groups have very low weights and the edges within a group have high weights. One way to solve graph partitioning problems is through objective functions. The two most common objective functions that encode the goals of the clustering are [6]: RatioCut and NormalizedCut (Ncut). However, solve any of objective functions mentioned is a NP-hard problem. Spectral clustering is a way to solve relaxed versions of such problems.

In Spectral clustering, the Laplacian matrices plays a key role. The properties of such matrices allows us to change the representation of the original similarity matrix enhancing the cluster-properties in the data, so that clusters can be trivially detected in this new representation.

There are two types of Laplacian matrices: (*i*) unnormalized graph Laplacian (L); (*ii*) normalized graph Laplacians which in turn there are two types: symmetric matrix L_{sym} and the random walk matrix L_{rw}. The difference between the unnormalized and the normalized matrices is that the former encode the relaxed version of RatioCut and the latter encode the relaxed version of Ncut objective function. An important property of those objective functions is that the Ncut encode both goals of clustering unlike the RatioCut which only encode the first goal. For more details about these matrices see [6].

Two spectral algorithms will be used on experimental study. To discuss the choice of an adequate similarity matrix setting as input of a spectral clustering algorithm, it will be used the Ng-Jordan-Weiss (NJW) algorithm [9]. This algorithm takes as input a similarity matrix and the k number of clusters to construct. As an output returns each data point associated to one cluster. In addiction, this algorithm uses the L_{sym}.

The DifFUZZY [3] algorithm takes as input the original dataset, the parameter M (detailed further), and three more parameters. The last three parameters (γ_1, γ_2 and γ_3) are related to the fact the algorithm makes use of L_{rw} which has behind a theory that tries to find a partition of the graph such that the random walk stays long within the same cluster and occasionally jumps between clusters. As an output DiFUZZY return the *k* number of clusters and for each data point a set of C numbers representing the degree of membership to each cluster.

Unlike the NJW algorithm, the DifFUZZY algorithm has a strategy to automatically determine the number of clusters in data, based on the ϵ-neighborhood graph construction. It was defined the function $F(\epsilon)$ that denotes the number of components of the ϵ-neighborhood graph which contains at least M vertices. Hereupon, they compute the number of the clusters C in the following way:

$$C = \max_{\epsilon \in (0,\infty)} (F(\epsilon))$$

They start to compute $F(\epsilon)$ with ϵ near to zero, and then the ϵ is being incremented. For relatively large ϵ values, the $F(\epsilon)$ value will be one as the ϵ-neighborhood graph becomes fully connected, i.e. it only has one component. Therefore, from a near zero value until a relatively large value, the number of clusters C will be the maximum number of components computed from the several ϵ-neighborhood graphs.

Both algorithms previously presented use normalized Laplacian matrices, once they are preferable compared with unnormalized Laplacian matrices [6].

3 Experiments and Comparative Analysis

In this section, it will be present and discuss the experimental study performed using the spectral clustering algorithms presented in section 2. The goal of such study is twofold: (*i*) to perform a comparative study about the impact on clustering result, when we use different kinds of similarity matrices as input for the NJW algorithm; the strategies involved on the comparative study are: the fully connected graph by Gaussian Kernel which adopts a *data-driven* strategy proposed by [10] (from now on it will be mentioned as Gaussian Kernel), the k-NN, the mutual k-NN and the fuzzy similarity proposed by [11] (from now on it will be mentioned as the fuzzy similarity); (*ii*) to conduct an experimental study with the DifFUZZY algorithm [3] in order to evaluate the effectiveness of its approach to automatically determine the number of clusters and corresponding partitions in data.

Along the study all the results have been evaluated according to the Adjusted Rand Index (ARI) [4] to score the similarity between the generated and computed clusterings. The results and its discussion will be presented in the subsequent sections.

3.1 Study about the Impact on Clustering Result using Different Similarities Matrices

When it is used the NJW algorithm, we have to choose which approach to use in order to construct the similarity matrix. However, different results are obtained with different approaches and some of them can be undesired. The purpose of this study is to make a comparative study, setting different similarity matrices as input of the NJW algorithm in order to evaluate its effectiveness for different datasets with different underlying structures.

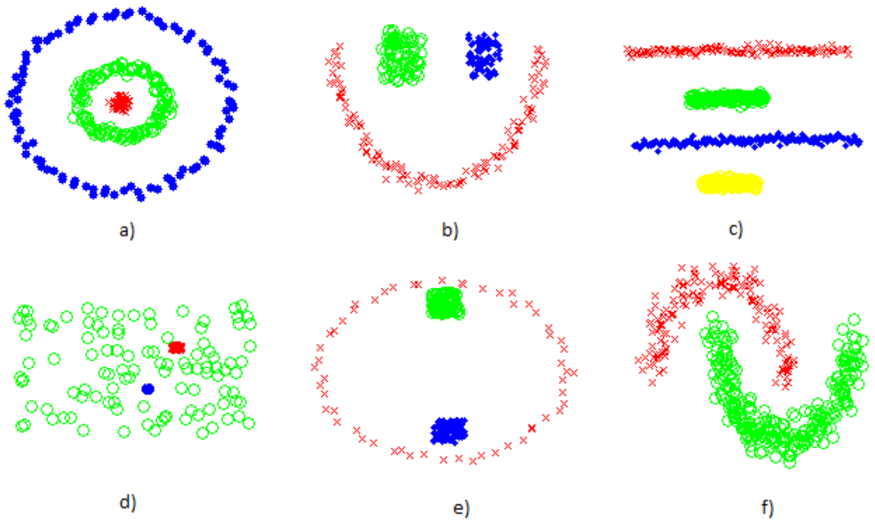

Figure 1: Some synthetic datasets involved in the study.

The study was started with typical synthetic datasets used in the literature. In the Figure 1, there are six datasets with different geometries which have been used in [10], excluding the dataset in the Figure 1f which was used in [6, 11]. In the Figure 1a, we have three circular geometries; regarding the Figure 1b, we have two squares and one semi-ellipse; concerning the Figure 1c, we have four lines with slightly different densities and width; in the Figure 1d, there is one dense square and circle, and irregular background clutter; in the Figure 1e, we have one concentric geometry and two similar squares (multi-scaled clusters); and, in the Figure 1f, we have two moons with also relatively different densities.

Table 1: The parameters used to obtain the results displayed in Table 2.

Parameters	**Gaussian Kernel**	**k-NN**	**mutual k-NN**	**Fuzzy similarity**
a)	k=3	k=7	k=9	t=4; C=51
b)	k=3	k=7	k=9	t=4; C=51
c)	k=3	k=7	k=9	t=4; C=51
d)	k=3	k=5	k=11	t=4; C=51*
e)	k=3	k=6	k=11	t=2; C=55*
f)	k=3	k=6	k=11	t=2; C=59*

Table 2: The ARI values after applying the NJW algorithm to the datasets shown in Figure 1.

dataset	**Gaussian Kernel**	**k-NN**	**mutual k-NN**	**Fuzzy similarity**
a)	1	1	1	1
b)	1	1	1	1
c)	1	1	1	1
d)	1	0.767	0.990	0.36*
e)	1	0,69	1	1*
f)	1	1	1	1*

In order to perform the comparative study regarding these datasets, the procedure was the follow: (*i*) it was applied the NJW algorithm setting different parameters for the construction of the similarities matrices (with $k = 1$ until $k = 80$ and at each iteration the k was incremented with one in case of Gaussian Kernel, k-NN and mutual k-NN); (*ii*) in case of the fuzzy similarity, the similar was done, i.e. for $t = 2$ until $t = 20$ (at each iteration the t was incremented with two) and for each t different C values were tested (from $C = 3$ until $C = 80$ incremented with 4), where C is the number of prototypes computed by the FCM..

The results were evaluated using the ARI. The best result obtained from each similarity matrix are shown in Table 2, and the corresponding parameters are displayed in Table 1. This was done so that the comparative analysis is performed based on the optimal result for each similarity strategy.

From the results obtained in Table 2, the Gaussian Kernel is the one which appears as the most robust, dealing with different geometries and different datasets using the same parameter, which is not true for the other strategies.

Furthermore, it was the only one which handle perfectly (at least so far) with irregular background clutter despite the mutual k-NN also provides good results. However, the k-NN does not deal very well with irregular background clutter, and when clusters with very different densities are close from each other which is the case illustrated in the Figure 1e.

Regarding the fuzzy similarity, it is obtained poor results when there is irregular background clutter in the dataset. For datasets d), e), f), if we apply the algorithm with the same parameters several times, different results are obtained (that is the reason for some results are tagged with a star). Recalling, the input to construct the fuzzy similarity matrix is the cluster prototypes V and partition matrix U obtained by the FCM. Even using the same parameters, the partition matrix obtained by the FCM is not always the same (once the initial prototypes are chosen randomly) which has a negative impact on the final clustering result for datasets with some inherent characteristics, namely multi-scaled clusters and irregular background clutter. Therefore, for such cases the fuzzy similarity strategy can be very instable. Thus, after some attempts, it was possible to obtain good results. However, they are unreliable.

So far, it was only used typical synthetic datasets used in the literature. In the Figure 2, it are proposed/designed two datasets. Regard the dataset illustrated in the Figure 2g, a clustering algorithm has to deal simultaneously with arbitrary shapes, different sizes and different densities; for the dataset in the Figure 2h, a clustering algorithm has to deal irregular background clutter, a normal cluster (the circle), and a cluster which itself has "areas" with different densities and it is a difficult geometry. The reason behind the latter dataset is because in the literature when it is tested datasets with irregular background clutter, the clusters are very dense and non-elongated.

For the next study, it is not consider the fuzzy similarity strategy as poor results were obtained for the reasons explain previously. Moreover, the procedure used was similar to previous one. The results obtained are in the Table 4 and the corresponding parameters in the Table 3.

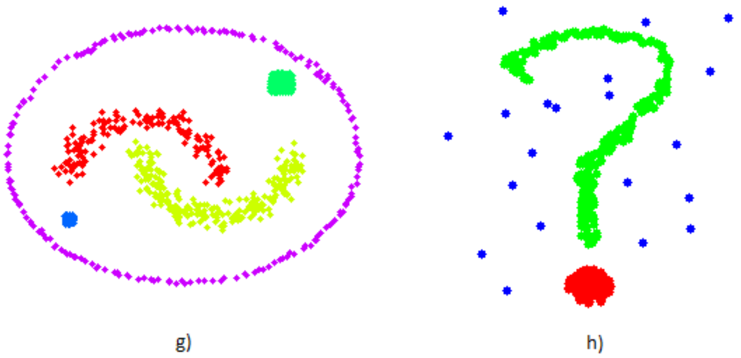

Figure 2: The synthetic datasets proposed.

From the results obtained in Table 4, concerning the dataset g), all the similarity approaches obtain good results. Yet, the Gaussian Kernel and mutual k-NN are more suitable. Therefore, in the absence of irregular background clut-

Table 3: The parameters used to obtain the results displayed in Table 4.

Parameters	Gaussian Kernel	k-NN	mutual k-NN
g)	k=7	k=7	k=10
h)	k=1	k=7	k=7

Table 4: The ARI values after applying the NJW algorithm to the datasets shown in Figure 2.

dataset	Gaussian Kernel	k-NN	mutual k-NN
g)	0,997	0,9343	0,997
h)	0,442	0,542	0,948

ter, such strategies are good dealing with arbitrary shapes, different sizes and different densities simultaneously. However, when the irregular background and arbitrary shapes (with different densities) characteristics are mixed in one dataset, the mutual k-NN reveals the most robust, as can be seen through the results obtained for dataset h).

For the previous datasets, it was essayed the usage of the NJW algorithm using the FCM where each data point was defuzzified so that every data point was assigned to its maximum membership cluster. The evaluation was performed also with ARI. Again, similar results were obtained, they are not included in the report.

It was done another comparative study, but this time the data was generated using the data generator developed. Specifically, the clusters of data points were generated from a bivariate spherical Gaussian distribution according to [8] and, the center of each cluster is located on bisectors of the quadrants of the Cartesian plane at the same distance from the origin ($Origin = (0, 0)$). This kind of dataset will be referred as *bivariate3* or *bivariate4* if the number of cluster is three or four, respectively.

Furthermore, like in [8], a scale parameter is used denoted by sn as a factor of clusters' proximity to the origin. When $sn = 0$, the clusters are more near to the center (high level of cluster intermix) and when $sn = 1$ the clusters are more distant from the center (low level of clusters intermix), as can be seen in the Figure 3.

Using the data generator, it was generated a *bivariate3* dataset with $sn = 1$ where all clusters have the same size, i.e. each cluster has 200 data points. Using this dataset and for each similarity matrix, it was applied the NJW algorithm several times: (*i*) with k from 1 to 80, and in each iteration the k was incremented with one, using k as input of Gaussian Kernel, k-NN and mutual k-NN; (*ii*) in case of the fuzzy similarity, the similar was done, i.e. it was applied the NJW algorithm with the fuzzy similarity using t from 2 until 20 (at each iteration the t was incremented with two) and for each t it was also used

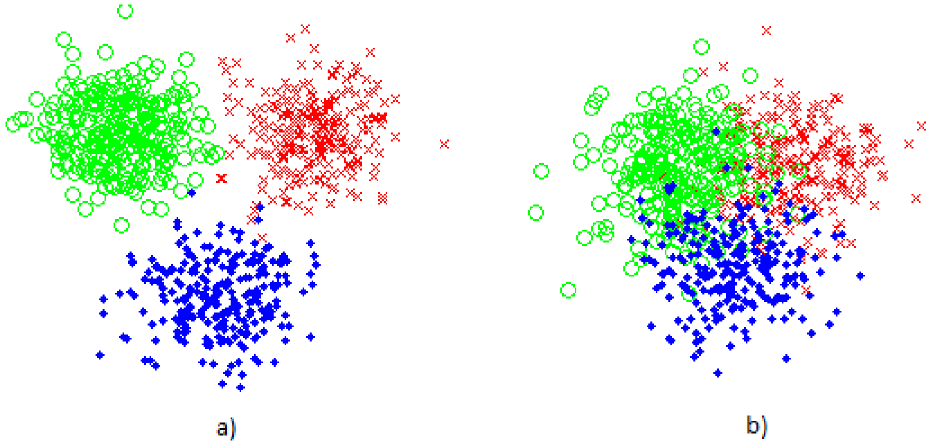

Figure 3: Bivariate3: a) $sn = 1$; b) $sn = 0$.

using different C values (from $C = 3$ until $C = 80$ incremented with 8). The results were evaluated using the ARI, and for the best results obtained from each similarity matrix, the parameter configurations were stored.

Then, it was performed the following: starting at $sn = 1$ and decreasing 0.3 value at each step, it was generated a *bivariate3.1* dataset (three clusters and each one with 200 data points), and it was applied to NJW algorithm, using the different similarity matrices and using the parameter configuration already computed; this was repeated five times and the average of ARI is presented in the table 5.

The previous process was repeated for *bivariate3.2* dataset, where the clusters have different cardinalities: 50, 200, 100 data points. The results are displayed in the Table 6.

Table 5: Bivariate3.1 (clusters with equal size) with different scale values of cluster intermix. The average of the ARI values.

	Gaussian Kernel	**k-NN**	**mutual k-NN**	**Fuzzy similarity**
sn/params	**k = 3**	**k=19**	**k=22**	**t=4 C=51**
1	0,971	0,968	0,963	0,969
0,7	0,923	0,913	0,921	0,913
0,4	0,818	0,821	0,818	0,629
0,1	0,635	0,629	0,623	0,528

Looking at the results displayed in the table 5 and 6, the NJW algorithm obtains good results using the different strategies when the value sn is equal to one. As we increase the level of intermix between the clusters and maintaining the same parameters, all the similarity strategies have the same behavior in both scenarios (clusters with equal and different sizes), i.e. the effectiveness on finding the correct partitions decrease. Yet, the fuzzy similarity have worse

Table 6: Bivariate3.2 (clusters with different size) with different scale values of cluster intermix. The average of the ARI values.

sn/params	Gaussian Kernel k = 3	k-NN k=7	mutual k-NN k=11	Fuzzy similarity t=8 C =51
1	0,963	0,949	0,925	0,970
0,7	0,894	0,844	0,868	0,895
0,4	0,764	0,590	0,502	0,531
0,1	0,585	0,424	0,486	0,502

results and the most robust for high level of intermix is the Gaussian Kernel.

However, the fine tuning is done considering $sn = 1$, in order to analyze the robustness of the similarity approaches as the level of intermix increases and the parameters are maintained.

By doing the fine tuning considering $sn = 0.1$, the results obtaining are presented in Table 7 and 8.

Table 7: Bivariate3 (clusters with equal size) with $sn = 0.1$. The ARI values.

sn/params	Gaussian Kernel k = 11	k-NN k=25	mutual k-NN k=29	Fuzzy similarity t=2 C =3
0,1	0,655	0,655	0,659	0,700

Table 8: Bivariate3 (clusters with different size) with $sn = 0.1$. The ARI values.

sn/params	Gaussian Kernel k = 2	k-NN k=27	mutual k-NN k=11	Fuzzy similarity t=2 C =24
0,1	0.549	0.412	0.503	0.635

As it can be seen, in both cases (clusters with equal and different sizes) the fuzzy similarity matrix not only improves the result comparing with the results previously obtained, but also outperforms the others strategies. Therefore, the fuzzy similarity matrix deals better with high level of clusters intermix, if we use the appropriate parameters.

The experiment with data points generated from a bivariate spherical Gaussian distribution with four clusters was essayed and similar results were obtained, but, globally, with a slightly weaker ARI values. Yet, the comparative analysis was the same.

To end this study, the NJW algorithm was applied to one widely used real dataset (from UCI Machine Learning Repository) in classification: Wisconsin Diagnostic Breast Cancer [1]. It contains two clusters (benign or malignant) of

357 and 212 data points correspondingly, of 30 dimensions (features). Plus, the classes are linearly separable using all 30 input features. For each similarity strategy, the parameters was computed as before and the ARI values obtained are shown in Table 9.

Table 9: Wisconsin Diagnostic Breast Cancer dataset . The ARI values.

Gaussian Kernel	k-NN	mutual k-NN	Fuzzy similarity
k = 2	k=4	k=66	t=6 C=30
0,737	0,81	0,7193	0,755*

The NJW algorithm returns relatively good results where the more suitable similarity strategy is the k-NN. On the other hand, the problem mentioned about the fuzzy similarity strategy concerning its instability is also verified for this dataset.

3.2 Study about the Effectiveness of DiFUZZY algorithm

In general, the number of clusters is an input parameter of spectral clustering algorithms. However, as it was mentioned previously, a minimum domain knowledge input should be required from the user [5], once the purpose of unsupervised technique is to find underlying structures from data.

It was evaluated the effectiveness the DiFUZZY algorithm and its approach to automatically infer the number of clusters in data (explained in section 2.2). In order to perform such task, it were used the major datasets used previously. The procedure was the follow: firstly, it were used the default values for γ_1, γ_2 and γ_3 (i.e. 0.3, 0.1 and 1 respectively), once they have been optimized and used successfully in all datasets according to [3]. For each dataset, it was applied the DiFUZZY algorithm for different M values, i.e from $M = 10$ until $M = 120$ (incremented at each step by 1); each data point undecided was defuzzified so that it was assigned to its maximum membership cluster. The results were evaluated according to the ARI and the best results obtained are displayed in Table 10.

For the datasets a), b), c), e), f) the DiFUZZY strategy was globally successful on finding the correct number of clusters and in the assignment of each data point to its correct cluster.

Concerning the dataset d), h) the algorithm cannot find the correct numbers of clusters. Recall, the datasets d), h) have irregular background clutter. Therefore, the DiFUZZY algorithm obtain poor results when the datasets have irregular background.

Regard the dataset g) is unable to recovered the correct number of clusters despite a good ARI result is obtained which means that the assignment of each data point to its correct cluster was done in a good way.

In order to test the ability of DiFUZZY to find the correct number of clusters, it was also tested using the *bivariate3.1* and *bivariate3.2* datasets with the sn equal to 0.1. For such datasets, the DiFUZZY was able to recovered the correct

Table 10: The ARI values after applying the DiFUZZY algorithm to the datasets of this study.

dataset	M	Number of Clusters	Number of Clusters Recovered	ARI
a)	61	3	3	1
b)	40	3	3	1
c)	50	4	4	0.9673
d)	40	3	2	0.4221
e)	20	3	3	0.891
f)	100	2	2	1
g)	90	5	4	0,849
h)	70	3	3	0.437
bivariate3.1	34	3	3	0.592
bivariate3.2	25	3	3	0.296
WBDC	15	2	2	0,723

number of clusters, however, the effectiveness of the results is lower comparing with the NJW algorithm. For higher values of sn, better results are obtained as occur with the NJW algorithm.

Finally, for the *wbdc* dataset (characterized by its high dimensionality), the DiFUZZY was also able to recovered the correct number of clusters. Yet, the effectiveness is lower than the best result obtained with the NJW algorithm.

4 Conclusions

In this report, it were perform two studies. Firstly, it was study how four different ways to construct the similarity matrix affects the effectiveness of the clustering result obtained by the NJW algorithm. From such study, the conclusions are: (*i*) the fuzzy similarity can be very unstable for datasets where there are clusters with arbitrary shapes, different densities and sizes, and if the dataset has irregular background clutter; (*ii*) the most robust similarity construction for datasets where there is simultaneously irregular background clutter and elongated geometries is the mutual k-NN; (*iii*) on the other hand, the fuzzy similarity is the most robust for bivariate spherical datasets considering high level of intermix between the clusters; (*iv*) the k-NN does not deal with irregular background clutter and with multi-scaled clusters, however, for real datasets present the best results; (*v*) the Gaussian Kernel is able to deal with different geometries and different datasets using the same parameters, i.e it is the one that presents less variation on the effectiveness when we change the k parameter; it present good results for dataset where there is simultaneously different arbitrary geometries, with different scales and sizes.

The other study performed evaluate the effectiveness of the DifFUZZY algorithm [3] and its approach to automatically determine the number of clusters and corresponding partitions in data. The conclusions from the study are: (*i*) the algorithm obtain poor results when the datasets have irregular background clutter; (*ii*) for bivariate spherical datasets, the algorithm can recover the correct number of cluster, however, the partitions obtain are not good, specially if the clusters have different sizes (considering high level of intermix between the clusters); (*iii*) for the dataset where there are clusters with arbitrary shapes, different densities and sizes, the algorithm was unable to recover the right number of clusters. Globally, its effectiveness is lower than the NJW algorithm.

References

[1] Wisconsin diagnostic breast cancer dataset description. `http://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/wdbc.names`

[2] Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Kluwer Academic Publishers, Norwell, MA, USA (1981)

[3] Cominetti, O., Matzavinos, A., Samarasinghe, S., Kulasiri, D., Liu, S., Maini, P.K., Erban, R.: Diffuzzy; a fuzzy clustering algorithm for complex datasets. Int. J. Comput. Intell. Bioinformatics Syst. Biol. 1, 402–417 (January 2010)

[4] Hubert, L., Arabie, P.: Comparing partitions. Journal of Classification 2, 193–218 (1985)

[5] Kolatch, E.: Clustering algorithms for spatial databases: A survey. Tech. rep. (2001)

[6] von Luxburg, U.: A tutorial on spectral clustering. Statistics and Computing 17, 395–416 (2007)

[7] Miller, H.J., Han, J.: Geographic Data Mining and Knowledge Discovery, Second Edition. CRC Press, 2 edn. (May 2009)

[8] Nascimento, S., Felizardo, R., Mirkin, B.: Thematic fuzzy clusters with an additive spectral approach. In: Antunes, L., Pinto, H.S. (eds.) EPIA. Lecture Notes in Computer Science, vol. 7026, pp. 446–461. Springer (2011)

[9] Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS. pp. 849–856. MIT Press (2001)

[10] Zelnik-Manor, L., Perona, P.: Self-tuning spectral clustering. In: NIPS (2004)

[11] Zhao, F., Liu, H., Jiao, L.: Spectral clustering with fuzzy similarity measure. Digital Signal Processing 21(6), 701 – 709 (2011)