Spectral Clustering and Semi-Supervised learning using Evolving Similarity Graphs

Christina Chrysouli[a,*], Anastasios Tefas[a]

[a]Department of Informatics, Aristotle University of Thessaloniki, University Campus 54124, Thessaloniki, Greece

Abstract

Spectral graph clustering has become very popular in recent years, due to the simplicity of its implementation as well as the performance of the method, in comparison with other popular ones. In this article, we propose a novel spectral graph clustering method that makes use of genetic algorithms, in order to optimise the structure of a graph and achieve better clustering results. We focus on evolving the constructed similarity graphs, by applying a fitness function (also called objective function), based on some of the most commonly used clustering criteria. The construction of the initial population is based on nearest neighbour graphs, some variants of them and some arbitrary ones, represented as matrices. Each one of these matrices is transformed properly in order to form a chromosome and be used in the evolutionary process. The algorithm's performance greatly depends on the way that the initial population is created, as suggested by the various techniques that have been examined for the purposes of this article. The most important advantage of the proposed method is its generic nature, as it can be applied to several problems, that can be modeled as graphs, including clustering, dimensionality reduction and classification problems. Experiments have been conducted on a traditional dances dataset and on other various multidimensional datasets, using evaluation methods based on both internal and external clustering criteria, in order to examine the performance of the proposed algorithm, providing promising results.

Keywords: Spectral Clustering, Similarity Graphs, Evolutionary Algorithms

1. Introduction

Clustering is an unsupervised learning process that aims at discovering the natural grouping of a set of data, such that similar samples are placed in the same group, while dissimilar samples are placed into different ones. The problem of clustering is a very challenging problem due to the assumption that no labels are attached to the data. Clustering has been used in a wide variety of applications, including bioinformatics [1],

[*] Corresponding author

Email addresses: chrysouli@aiia.csd.auth.gr (Christina Chrysouli), tefas@aiia.csd.auth.gr (Anastasios Tefas)

[2], data mining [3], image analysis [4], information retrieval [5], [6], etc. A detailed survey on clustering applications can be found in [7] and a more recent study in [8]. In [9] the authors attempt to briefly review a few core concepts of unsupervised and semi-supervised clustering.

Spectral graph clustering [10] refers to a class of graph techniques, that rely on eigenanalysis of the Laplacian matrix of a similarity graph, aiming to divide graph nodes in disjoint groups (or clusters). In spectral clustering, as in all clustering techniques, nodes that originate from the same cluster should have high similarity values, whereas nodes from different clusters should have low similarity values. Spectral analysis can be applied to a variety of practical problems, including face clustering [11] [12], speech analysis [13, 14] and dimensionality reduction [15], and, as a result, spectral clustering algorithms have received increasing interest. More clustering applications of spectral graph clustering are reviewed in [16].

In the last years, evolutionary-based approaches have been extensively applied to clustering problems due to their ability to adapt to very different problems with only few changes [17]. In [18] the authors proposed a genetic algorithm in order to search for the cluster centers by minimising a clustering metric, while in [19] authors aim to find the optimal partition of the data, using a genetic algorithm, without searching all possible partitions. In another, more recent work [20] the authors propose a new grouping genetic algorithm applied to clustering, emphasising to the proposed encoding and different modifications of crossover and mutation operators. A more detailed survey of evolutionary algorithms for clustering is presented in [21].

Evolutionary algorithms are probabilistic algorithms, which might not guarantee the optimal solution, but are likely to return a good one, in a reasonable period of time. The most important advantage of evolutionary algorithms is that they do not require any auxiliary knowledge, but only a fitness function. This makes them good candidates for optimising different kinds of criteria. Moreover, other advantages of evolutionary algorithms are the simplicity of the method, their adaptability and the fact that they can cope with multi-modal functions.They are also particularly well suited for difficult problems, where little is known about the underlying search space.

In the proposed approach, spectral graph clustering has been employed and applied on evolving similarity graphs, which have been transformed in such a way so as to play the role of the chromosomes in the genetic algorithm [22]. The initial population, for the needs of the genetic algorithm, is constructed with the aid of k-nearest neighbour graphs, represented as matrices, which are, then, transformed to one-dimensional binary strings and undergo genetic operators. The evolutionary algorithm is performed based on the value of the employed fitness function, that uses some of the most common clustering criteria. In the proposed method, we make use of spectral graph clustering in order to find logical grouping of the dataset.

The remainder of this paper is structured as follows. In section 2, we state the problem that we are dealing with in detail, as well as some of the general aspects that concern the proposed algorithm. We also discuss the way that the similarity graph is created and some spectral clustering issues. In Section 3, the proposed evolutionary algorithm is presented in detail. In Section 4, experimental results of the algorithm are presented and described. Finally, in Section 5, conclusions are drawn and future work is discussed.

2. Problem statement

Clustering is the process of partitioning a usually large dataset into groups (or clusters), according to a similarity (or dissimilarity) measure. It is an unsupervised learning process that no labels are provided and, also, no information of the number of clusters is given. The aim of any clustering algorithm is to place in the same cluster samples that have a small distance from each other, whereas samples that are placed in different clusters are at a large distance from each other. If we assume that we have a dataset X, defined as $X = \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, ...$, which consists of all the data that we want to place into clusters, then we define a clustering of X in m clusters $C_1, ..., C_m$, in such a way that the following conditions apply:

- $C_i \neq \emptyset$, $i = 1, ...m$
- $\cup_{i=0}^{m} C_i = X$
- $C_i \cap C_j = \emptyset, \ i, j = 1, ..., m, \ i \neq j$

Clustering is usually not a trivial task, as the only information we have about the data, is the data itself. In order to obtain some information about the structure of the data, we usually construct similarity matrices.

2.1. Similarity Functions and Similarity Graphs

Similarities between data samples can be represented as a similarity graph $G = (V, E)$, where V, E represent vertices (or nodes) and edges of the graph, respectively. Assuming that each vertex $\mathbf{v}_i$ represents a single data sample, then two nodes $\mathbf{v}_i$, $\mathbf{v}_j$ are connected if the similarity $s_{i,j}$ between them is positive or larger than a threshold, and the edge is weighted by $s_{i,j}$. Thus, the problem of clustering can be reformulated as finding a partition of the graph such that the weights within a cluster have high values, whereas weights between different clusters have low values. In an ideal situation, all data that belong to the same cluster should be placed in a compact group, which lie away from other compact groups.

Before constructing a similarity graph, we need to define a similarity function on the data. This means that we should take under consideration that points which are "similar" in the dataset should also be similar after applying the similarity function. The most common similarity function $\mathbf{S}$ is the Gaussian similarity function (heat kernel) [23]. Heat kernel between two graph nodes is defined as:

$$[\mathbf{S}]_{i,j} = h(\mathbf{v}_i, \mathbf{v}_j) = exp\left(-\frac{\left\|\mathbf{v}_i - \mathbf{v}_j\right\|^2}{\sigma^2}\right), \tag{1}$$

where σ is a parameter that defines the width of the neighbourhood. The value of σ plays an important role to the performance of the algorithm, thus, there are several, arbitrary, rules concerning its value. One of them is to choose σ using variants of the data diameter, for example the exact or multiples of the diameter.

The purpose of a similarity graph is to connect data points that belong to local neighbourhoods, in a way that each point corresponds to a node in the graph. Local neighbourhoods can be expressed in many different ways, leading to many types of similarity graphs (e.g. k-nearest neighbour graphs, ε-neighbourhood graphs, fully connected graphs) and node connections. In general, experience shows that constructing a similarity graph is not a trivial task [24], and little is known on the effect of the various constructions. Moreover, it affects, significantly, the performance of the algorithm; different choices of the similarity graph lead to entirely different results.

One of the most common choices when constructing a similarity graph [24], is k-nearest neighbour graphs (to be called k-nn graphs) because of their simplicity as well as their sparsity. The aim of a k-nn graph $\mathbf{A}$ is to connect node $\mathbf{v}_i$ with node $\mathbf{v}_j$ if $\mathbf{v}_j$ is among the k nearest neighbours of $\mathbf{v}_i$, which results in a directed graph [24]. In the proposed method, an undirected graph was used, obtained by simply ignoring the directions of the edges, meaning that we connect $\mathbf{v}_i$ and $\mathbf{v}_j$ if $\mathbf{v}_i$ is among the k-nearest neighbours of $\mathbf{v}_j$ or if $\mathbf{v}_j$ is among the k-nearest neighbours of $\mathbf{v}_i$ (symmetric k-nearest neighbour graph).

It is well known that spectral clustering is very sensitive to the choice of the similarity graph that is used for constructing the Laplacian [24]. Indeed, selecting a fixed k parameter for the k-nn graph is very difficult and different values lead to dramatically different clusterings.

Indeed, the standard approach so far was to deal only with k-nn graphs and the optimisation of the parameter k was dealt with trial and error or cross-validation approaches. However, there is no justification of the use of k-nn graphs when someone wants to optimise a generic clusterability criterion. In the most generic case any arbitrary graph could be the fittest solution to the specific optimisation problem. Moreover, when we consider distributions that are not unimodal or they are mixtures of distributions with different density from region to region, it is not reasonable to consider that a single parameter k can work appropriately for the entire data space.

A second approach, which also restricts the solutions is to consider a threshold for building the neighbourhoods (ε-ball neighbourhood). However, in that case the problem still remains that the solution is sub-optimal and considers that a single parameter ε will fit to the entire dataset.

Optimising the clustering results over the graph structure is not a trivial task, since the clustering criteria are not differentiable with respect to the graph structure. Thus, we propose in this paper to use evolutionary algorithms in order to optimise specific clustering criteria, that are considered as fitness functions, with respect to the underlying graph, which is transformed to a chromosome solution.

The proposed approach is by far the most generic one since it allows for any arbitrary graph to become the fittest given that the initial population and the genetic transformations are appropriately designed. To this end, we use the intuitively good solutions of the k-nn graphs in the initial population and also random graphs that will give us the diversity in the population.

2.2. Spectral Graph Clustering

Spectral graph clustering [10], refers to a class of graph techniques, which rely on the eigenanalysis of a matrix, in order to partition graph nodes in disjoint clusters and is commonly used, in recent years, in many clustering applications [16]. As in all clustering techniques, in spectral graph clustering nodes that belong to the same cluster should have high similarity values, whereas nodes from different clusters should have low similarity values. Before proceeding to spectral clustering algorithm, it is crucial to define the Laplacian matrix.

Let $\mathbf{D}$ be a diagonal $N \times N$ matrix having the sum $d_{ii} = \sum_j W_{i,j}$ on its main diagonal. Then, the generalised eigenvalue problem is defined as:

$$(\mathbf{D} - \mathbf{W})\mathbf{v} = \lambda \mathbf{D}\mathbf{v}, \tag{2}$$

where $\mathbf{W}$ is the adjacency matrix, and $\mathbf{v}$, λ are the eigenvectors and eigenvalues respectively.

Although many variations of graph Laplacians exist [24], we focus on the normalised graph Laplacian $\mathbf{L}$ [25], which is a symmetric matrix, and can be defined as:

$$\mathbf{L} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2} \tag{3}$$

$$= \mathbf{D}^{-1/2}(\mathbf{D} - \mathbf{W})\mathbf{D}^{-1/2} \tag{4}$$

$$= \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2} \tag{5}$$

where $\mathbf{W}$ is the adjacency matrix, with $w_{i,j} = w_{j,i} \geq 0$, $\mathbf{D}$ is the degree matrix and $\mathbf{I}$ is the identity matrix. The smallest eigenvalue of $\mathbf{L}$ is 0, which corresponds to the eigenvector $\mathbf{D}^{-1/2}\mathbf{1}$. The $\mathbf{L}$ matrix is always positive semi-definite and has n non-negative real-valued eigenvalues $\lambda_1 \leq ... \leq \lambda_n$. The computational cost of spectral clustering algorithms is quite low when matrices are sparse. Luckily, we make use of k-nn graphs which are in fact sparse.

In the proposed method, we perform eigenanalysis on $\mathbf{L}$ matrix, where $\mathbf{W}$ is defined as:

$$\mathbf{W} = \mathbf{S} \odot \mathbf{A}, \tag{6}$$

where $\mathbf{S}$ represents the full similarity matrix obtained using (1) and $\mathbf{A}$ represents an undirected k-nn matrix, which is a sparse matrix. The $\odot$ operator performs element-wise multiplication. This process results in a sparse matrix $\mathbf{W}$, only containing non-zero elements in places where $\mathbf{A}$ matrix contains non-zero elements. An example of the $\odot$ operator is illustrated in Figure 1. Eigenvalues are always ordered increasingly, respecting multiplicities, and the first k eigenvectors correspond to the k smallest eigenvalues. Once the eigenanalysis has been performed and the new representation of the data has been obtained, the k-means algorithm is used in order to attach a cluster to every data sample.

3. The proposed evolutionary technique for clustering

In order to partition a dataset into clusters, spectral graph clustering has been applied on evolving k-nn similarity graphs. In more detail, we evolve a number of k-nn similarity graphs with the aid of a genetic algorithm, in order to optimise the structure of the graph, by optimising a clustering criterion. In this paper, clustering criteria were employed as fitness functions. Moreover, k-nn similarity graphs are transformed properly into chromosome solutions, in order to be used in the genetic algorithm.

$$\overset{\mathbf{S}}{\begin{bmatrix} 1 & 0.1 & 0.4 & 0.6 & 0.8 & 0.7 & 0.7 & 0.3 \\ 0.1 & 1 & 0.5 & 0.8 & 0.1 & 0.4 & 0.6 & 0.7 \\ 0.4 & 0.5 & 1 & 0.6 & 0.9 & 0.5 & 0.2 & 0.5 \\ 0.6 & 0.8 & 0.6 & 1 & 0.6 & 0.9 & 0.1 & 0.2 \\ 0.8 & 0.1 & 0.9 & 0.6 & 1 & 0.2 & 0.2 & 0.7 \\ 0.7 & 0.4 & 0.5 & 0.9 & 0.2 & 1 & 0.8 & 0.4 \\ 0.7 & 0.6 & 0.2 & 0.1 & 0.2 & 0.8 & 1 & 0.7 \\ 0.3 & 0.7 & 0.5 & 0.2 & 0.7 & 0.4 & 0.7 & 1 \end{bmatrix}} \odot \overset{\mathbf{A}}{\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}} = \overset{\mathbf{W}}{\begin{bmatrix} 1 & 0 & 0 & 0 & 0.8 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0.8 & 0 & 0 & 0 & 0.7 \\ 0 & 0 & 1 & 0.6 & 0.9 & 0 & 0 & 0 \\ 0 & 0.8 & 0.6 & 1 & 0 & 0.9 & 0 & 0 \\ 0.8 & 0 & 0.9 & 0 & 1 & 0 & 0 & 0.7 \\ 0 & 0 & 0 & 0.9 & 0 & 1 & 0.8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.8 & 1 & 0.7 \\ 0 & 0.7 & 0 & 0 & 0.7 & 0 & 0.7 & 1 \end{bmatrix}}$$

Figure 1: An example of $\odot$ operator which performs element-wise multiplication.

Let J be a clustering criterion that depends on the similarity graph $\mathbf{W}$. However, the optimisation problem is not convex and, moreover, the fitness function is not differentiable with respect to $\mathbf{W}$. Since $\mathbf{S}$ is considered constant after selecting a specific similarity function and through the definition of $\mathbf{W}$ in (6), the optimisation problem is defined as:

$$\underset{\mathbf{A}}{optimise}\, J(\mathbf{A}), \tag{7}$$

where $\mathbf{A}_{i,j} \in 0,1$ is a k-nn graph.

3.1. Construction of Initial Population

In order to create the initial population, we do not make use of the full similarity matrix S, mainly for time and space efficiency reasons. Instead, we use the sparse matrices that originate from k-nn graphs, resulting in an initial population that consists of matrices with binary elements. The decision of employing of the k-nn graphs, for the construction of the initial population, was based on the observation that their structure was already good (also they are sparse graphs). The aim of the proposed algorithm is to find a new structure of these k-nn graphs, so as to obtain better clustering results. Also, efforts to use only random sparse matrices, as initial population, have been made in order to gain completely different structures of the graphs, which only led to worse results, thus, not presented here.

In this method, a Gaussian function has been employed as a similarity measure, in order to obtain the similarity matrix $\mathbf{S}$, which is calculated pairwise for all the data in a database of our choice, using (1). Our experiments showed that the value of σ has a decisive role to the performance of the algorithm. In the proposed method, we have used multiples of the data diameter.

The process of construction the initial population begins with the calculation of k-nearest neighbour matrices $\mathbf{A}$, with $k = 3,...,8$, which constitute the backbone of the initial population. Next step is to enrich the population with nearly k-nearest neighbour matrices. In order to achieve that, we alter the k-nearest neighbour matrices that have already been calculated, by converting a small proportion of 0's, from $\mathbf{A}$ matrices, to 1's and vice versa. In more detail, in order to choose which elements of the matrix are going to be inverted, we first select randomly, with probability of 1%, T elements to be inverted from ones to zeros, and then, using the exact same number of elements we invert randomly T zeros to ones.

This process guarantees that the proportion of 1's and 0's will remain the same in the new matrix, thus the new chromosomes will have almost equal number of zeros and ones. It is important not to alter the k-nn graphs completely, so as to keep all the good properties. Finally, a small proportion of completely random matrices are added to the set of matrices, in order to increase the population diversity, in which the number of 1's are equal to the number of 1's that a 5-nn graph would have. Approximately 10% of the initial population is comprised of random graphs.

From the various experiments conducted, we have concluded that the selection of the parameter k of the nearest neighbour graphs is crucial to the clustering results, as illustrated in Figure 2. Figure 2a presents a dataset that consists of two classes with each one having a different colour. Figures 2b and 2c represent the clustering results when a 3 and a 5-nearest neighbour graph were used, respectively. We should highlight the difference between the clustering results, especially when the elements are close to both classes.

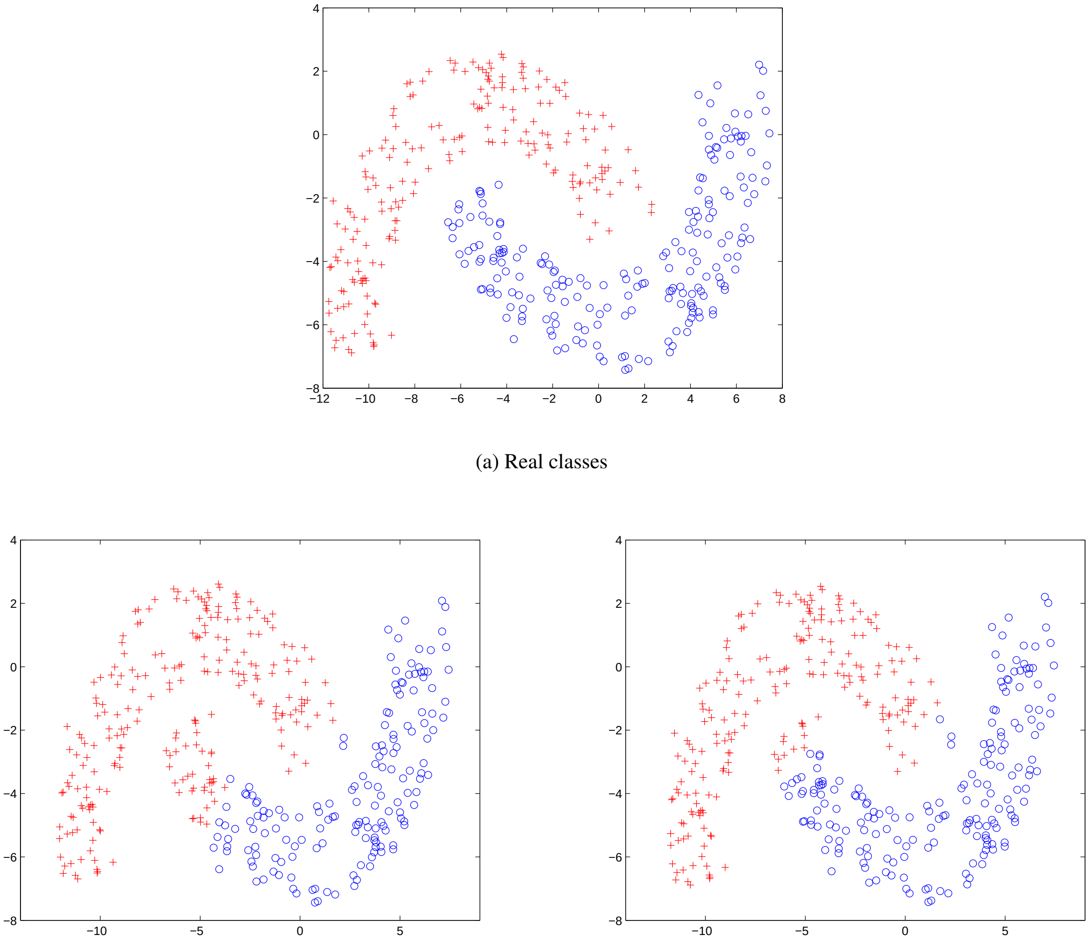
Before proceeding to the proposed algorithm, we must define the way that the k-nn matrices, and variants of these matrices, in the initial population are transformed into chromosomes, thus, we need to define how a square matrix, like a similarity described earlier, becomes a one-dimensional vector. As the k-nn graphs $\mathbf{A}$ are constructed in such a way to be symmetrical, we may only keep the elements of the upper triangular matrix, with no loss of information. Then, the remaining elements are accessed in rows sequentially, forming the one-dimensional vector. The procedure of reforming a square matrix in order to obtain the one dimensional chromosome is illustrated in Figure 3.

3.2. Optimisation of the Solutions

The novelty of the proposed algorithm is based on the way that we select to optimise the solutions of the problem, by optimising a clustering criterion J, as previously defined in (7). Since clustering criteria are not differentiable we make use of genetic algorithms in order to optimise them. Before focusing on how this is achieved, we need to define clustering criteria.

Clustering criteria are divided into two main categories, internal and external criteria. The calculation of internal criteria implies that we have no prior knowledge about the data and we can only rely on quantities and features inherent to the dataset, whereas calculation of external criteria implies that we have some knowledge about the dataset in advance. More specifically, in order to use an external criterion, we should already have a representation of human reference clustering, called ground truth. Usually, in real problems, we do not have any prior information about the dataset, thus it is difficult to use external criteria in such problems. Nevertheless, external criteria are more representative of the cluster structures.

In the literature, many different criteria have been proposed [26], [27], that can be used in order to measure the fitness of the clusters produced by clustering algorithms. Some of the most widely used internal criteria are Calinski-Harabasz index [28], Davies-Bouldin index [29] and Dunn's index [30], whereas some external criteria are F-measure [31], purity [32], normalised mutual information [33] and a measure based on hungarian algorithm [34]. All the aforementioned criteria have been used in the proposed algorithm, some of them both for optimisation and evaluating the performance of the algorithm and some only for evaluation. Starting with internal criteria:



(a) Real classes

(b) 3-nearest neighbour graph

(c) 5-nearest neighbour graph

Figure 2: The effect of k-nearest neighbour graphs in clustering. In 2a the two classes of the dataset are presented.

- Calinski-Harabasz index [28] can be defined as:

$$CH = \frac{trace\,\{\mathbf{S}_B\}}{trace\,\{\mathbf{S}_W\}} \times \frac{N-k}{k-1}, \tag{8}$$

where N is the number of the elements in the dataset examined (that is if a dataset consists of 100 images, then $N = 100$), k is the number of the disjoint clusters after the partition, $\mathbf{S}_B$ and $\mathbf{S}_W$ are the between-cluster scatter and within-cluster

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ & & 0 & 0 & 1 & 0 & 0 & 1 \\ & & & 1 & 0 & 0 & 0 & 0 \\ & & & & 0 & 1 & 0 & 0 \\ & & & & & 0 & 0 & 1 \\ & & & & & & 1 & 0 \\ & & & & & & & 1 \\ & & & & & & & \end{bmatrix} \rightarrow$$

$$\begin{bmatrix} 1001000001001100000100001101 \end{bmatrix}$$

Figure 3: The way a k-nn graph $\mathbf{A}$ is transformed into a, one-dimensional vector, chromosome.

scatter matrices respectively:

$$\mathbf{S}_B = \sum_{i=1}^{k} N_i \left(\mathbf{m}_i - \mathbf{m}\right) \left(\mathbf{m}_i - \mathbf{m}\right)^T , \tag{9}$$

and

$$\mathbf{S}_W = \sum_{i=1}^{k} \sum_{p=1}^{N_i} \left(\mathbf{x}_i(p) - \mathbf{m}_i\right) \left(\mathbf{x}_i(p) - \mathbf{m}_i\right)^T \tag{10}$$

where N_i is the number of objects assigned to the i-th cluster, $\mathbf{x}_i(p)$ is the p-th element assigned to the i-th cluster, $\mathbf{m}_i$ is the vector of element means within the i-th cluster, and $\mathbf{m}$ is the vector of overall element means. Generally, we expect compact and well separated clusters to have high values of $trace\{\mathbf{S}_B\}$ and low values of $trace\{\mathbf{S}_W\}$. Therefore, the better the data clustering the higher the value of the ratio between $trace\{\mathbf{S}_W\}$ and $trace\{\mathbf{S}_B\}$.

- Davies-Bouldin index [29] is a criterion also based on within and between cluster similarities and is defined as:

$$DB = \frac{1}{k} \sum_{i=1}^{k} max_{i \neq j} \frac{\overline{d}_i + \overline{d}_j}{d_{i,j}}, \tag{11}$$

where k denotes the number of the disjoint clusters after the partition, i, j are cluster labels, $\overline{d}_i$ is the average within-group distance for the i-th cluster and $d_{i,j}$ is the inter-group distance between clusters i and j. If the Euclidean norm is used, then $\overline{d}_i$, $d_{i,j}$ take the form of equation (9) and (10), respectively. The smaller the value of DB index, the better the partition.

- Another internal clustering criterion is Dunn's index [30], which is based on separability and compactness of the clusters, and can be defined as:

$$D = min_{p,q \in \{1,\ldots,k\}, p \neq q} \left\{ \frac{d_{p,q}}{max_{i \in \{1,\ldots,k\}} \delta_i} \right\}, \tag{12}$$

where δ_i is the diameter of the i-th cluster and $d_{p,q}$ is the set distance between p and q clusters. The diameter δ_i of a cluster i is defined as the maximum distance between a pair of elements within that cluster. The set distance $d_{p,q}$ can be defined as the minimum distance between a pair of elements between two different clusters p and q. Large values of D imply a better partition of the data.

In the bibliography, there are also many variations of all the aforementioned criteria [26], which are based on different definitions of distance, diameter etc.

External criteria assume that we, somehow, have a representation of the human reference clustering.

- F-measure is a widely used external criterion that makes use of precision and recall measures [31]. Precision, is defined as the ratio of correct instances to the total number of elements that actually belong to the cluster, whereas recall is defined as the ratio of the correct elements to the total number of instances that the algorithm returned to belong in the cluster. Then, precision of a cluster j with respect to cluster i can be defined as:

$$prec(i,j) = \frac{\left|C_j \cap C_i^*\right|}{\left|C_j\right|}, \tag{13}$$

while the recall of a cluster j with respect to cluster i is defined as:

$$rec(i,j) = \frac{\left|C_j \cap C_i^*\right|}{|C_i^*|}. \tag{14}$$

where $C = C_1,...,C_n$ is a clustering of a set and $C^* = C_1^*,...,C_m^*$ is a representation of human reference clustering of the same set (i.e. ground truth). Then, F-measure is defined as:

$$F(i,j) = 2 \times \frac{prec(i,j) \times rec(i,j)}{prec(i,j) + rec(i,j)}, \tag{15}$$

while the overall F-measure is given by:

$$F = \sum_{i=1}^{m} \frac{|C_i^*|}{|S|} \times \max_{j=1,...,n} F(i,j). \tag{16}$$

where $|S|$ is the cardinality of the elements in the set and $|C_i^*|$ is the cardinality of the elements in the human reference clustering set.

Generally, a large precision value implies that most of the elements in the set were correctly clustered, but we might not have derived all the elements that belong to this cluster. On the other hand, a large recall value implies that we have found almost all elements that belong to a cluster, but we might also have a lot of elements that do not belong to this cluster. It must be noted that, F-measure, punishes more clusters that contain samples from different ground truth classes, than splitting a ground truth class into two clusters which, nevertheless, contains samples of only one ground truth cluster. In order to have better partitioning results, we need to maximise the F-measure.

- Another external criterion that is commonly used is purity [32], which can be defined as:

$$Pur = \sum_{i=1}^{k} \frac{n_i}{n} \left(\frac{1}{n_i} max_j(n_i^j) \right), \tag{17}$$

  where n_i is the size of the cluster i, k is the number of clusters, n is the total number of elements in the dataset, and (n_i^j) is the number of elements in cluster i with the label j. Higher values of Pur, imply better clustering results.

- Normalised mutual information [33] is also used as an external criterion for evaluating clustering algorithms, and is defined as:

$$NMI(X;Y) = \frac{H(X) + H(Y)}{H(X,Y)}. \tag{18}$$

  where $H(X)$, $H(Y)$ denote the entropy of X and Y probability distributions respectively and $H(X,Y)$ is the entropy:

$$H(X) = -\sum p(x) \log p(x), \tag{19}$$

$$H(X,Y) = -\sum p(x,y) \log p(x,y), \tag{20}$$

  where $p(x)$ is the marginal probability of X and $p(x,y)$ is the joint probability of X and Y. Generally, the higher the NMI value in (18), the better the partition.

- Another external criterion is the hungarian measure Hun [34]. The aim of this measure is to match every cluster, which the algorithm returned, to the best possible class, as it was defined by ground truth. In more detail, if we suppose to have k clusters to which we want to assign l classes, on a one-to-one basis, and we also know the cost of assigning a given cluster to a given class, then we want to know the optimal assignment, the one that minimises the total cost. The hungarian method is an algorithm that finds an optimal assignment for a given cost matrix C. The higher the Hun value the better the clustering result.

As the value of such criteria cannot be optimised, without the use of derivatives, we have employed evolutionary techniques in order to solve this problem. The optimisation is performed by altering the chromosomes or, else, by altering the k-nn similarity matrices $\mathbf{A}$ as in (2).

3.3. The Genetic Cycle

As we have already defined how the initial population is formed and how the chromosome evaluation is performed, we may now define the details of the genetic algorithm. Some of the main operators are single-point crossover, multi-point crossover and mutation operators which are illustrated in Figure 4. In a single-point crossover operator a random crossover point is selected, in a corresponding place in both the chromosome-parents. The offsprings are produced by mutually exchanging the subsequences of the chromosome-parents, in the crossover point. The multi-point crossover

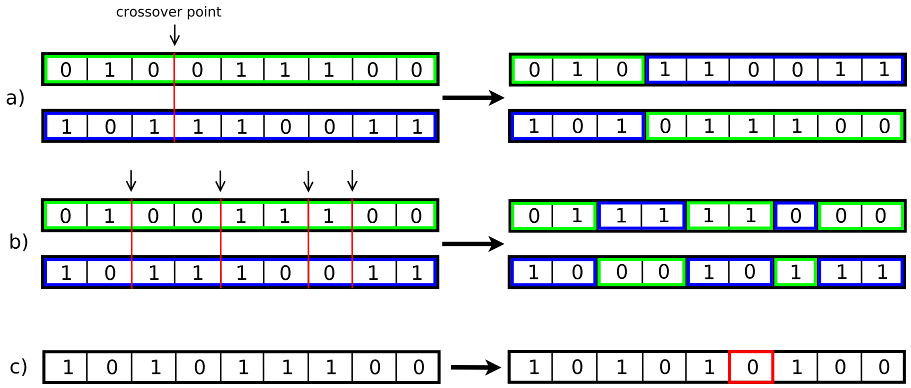


Figure 4: a) Single-point crossover operator. b) Multi-point crossover operator. c) Mutation operator. We have emphasised the element of the chromosome that is mutated (marked with a red border).

operator is the general case, where multiple crossover points are chosen and are mutually being exchanged, sequentially. Mutation operator is the random alternation that happens in one or more points of the bit string. In all 4 a), b), c) the chromosome-parents are presented on the left, while the chromosome-children are presented on the right.

Evolutionary algorithms solve problems based on operators inspired from biology. The first step of the genetic algorithm is to select the chromosomes which will undergo the crossover operator. For this purpose, a roulette wheel method has been employed [35], where a probability is associated with each chromosome, based on the value of the fitness function: the higher the value, the higher the probability to be selected. The probability p_i of the i-th chromosome to be selected, if f_i is its fitness value, is defined as:

$$p_i = \frac{f_i}{\Sigma_{j=1}^{N} f_j}. \tag{21}$$

This technique is based on the notion that the fittest chromosomes are more likely to produce fitter offsprings when combined, than if some less fitter were combined. The number of the offspring-chromosomes, that is chosen to undergo crossover operator, is usually the same as the parents-chromosomes in the population. Nevertheless, it is very usual a chromosome with a high fitness value to be chosen for crossover more than once and, also, a chromosome with a low fitness value not to be chosen at all.

Next, we combine the selected chromosomes, some of which, it is reasonable to expect that they will produce chromosomes with a higher fitness value. The crossover rate defines if chromosomes will finally undergo crossover operator. The crossover rate takes values between 0 and 1 but, usually, a value closer to 1 is used; in our experiments we have used a value of 0.7. In the proposed algorithm, a single crossover point is randomly selected for every set of chromosomes and the sub-sequences that are formed are exchanged respectively.

Then, we randomly choose a proportion of the chromosomes to undergo mutation, that is the random change of some elements of a chromosome. This proportion is based on the mutation rate, which was set to 0.4 in all of our experiments, that takes values from 0 to 1, as the crossover rate did. In more detail, for every chromosome, a random number r between 0 and 1 is generated; if r is larger than the mutation rate, then this chromosome will undergo mutation, else it will remain unchanged. A small value of the mutation rate is usually preferable. Moreover, in order to guarantee that the newly produced chromosomes will not have been altered too much, after the mutation, mutation is performed by converting a number of 0's to 1's and vice versa. More precisely, we first select randomly, with probability of 1%, T elements to be inverted from ones to zeros, and then, using the exact same number of elements we invert randomly T zeros to ones.

After the application of genetic operators to the chromosomes, the new generation has been formed. In order to perform spectral clustering (Section 2.2), we need to reconstruct the k-nearest neighbour matrix $\mathbf{A}$, which will consist of binary digits, from the one-dimensional vector chromosome. Then we apply the similarity matrix $\mathbf{S}$ on $\mathbf{A}$ using the $\odot$ operator, in order to obtain the $\mathbf{W}$ as illustrated in Figure 1. Spectral clustering [25] may now be performed on $\mathbf{L}$ as in (5).

The next step is to calculate the fitness values of all the newly produced chromosomes, and place them along with the parent-chromosomes. Then, elitism is performed: we sort all chromosomes, with the fittest being on the top, and we keep only those chromosomes with the highest fitness value, so as the number of the chromosomes kept to remain unchanged after every generation.

The proposed algorithm terminates when a maximum of 50 generations has been reached, or when the optimised criterion has not been altered for 5 consecutive generations.

To the best of our knowledge this is the first paper that tries to optimise the generic graph structure in spectral clustering. Indeed, the proposed approach is very promising since it gives a generic solution to several machine learning algorithms that use the data structure in terms of their adjacency matrices as regularizers. The application of the proposed approach to supervised learning is straightforward and it will lead to novel classifiers with improved performance. Laplacian least squares, Laplacian support vector machines, graph embedding, etc. are some of the examples of algorithms that can benefit from the proposed approach.

3.4. Semi-supervised Learning

Traditionally, learning has been approached either in an unsupervised manner, for example clustering, where all the data are unlabeled, or in a supervised manner, for example classification and regression, where all the data are labeled. Generally, it is time consuming, and thus expensive, to collect labeled data, while unlabeled ones are easier to gather. The aim of semi-supervised learning is to combin labeled and unlabeled data in such a way in order to change the learning behavior, and design algorithms that take advantage of such a combination [36].

It is natural for many practical problems to consider that we only possess a proportion of labels in a dataset. Then, the problem of clustering can be transformed into how this small proportion of labels can be used in order to obtain a better clustering of the data. Semi-supervised learning [37], in machine learning, is a class of techniques which uses both labeled and unlabeled data, usually a small set of the former and a large set of the latter, in order to obtain clusters. Thus, both labeled and unlabeled data are used in the learning process.

Semi-supervised learning has been widely used during the recent years in clustering. It has been applied on many practical problems, including image classification [38], person identification [39] and large scale text applications [40].

Following the notion of semi-supervised learning, we assume that our data follow a specific probability distribution P. Then, according to this definition, we can redefine each data sample, to which has been attached a label, as a pair of (x,y), where x represents the data sample and y represents the labels, and which are created based on the probability P. Moreover, the data with no labels are represented as x, which are created based on the marginal probability P_x of P. Semi-supervised learning is based on the fact that this marginal probability P_x can be used in order to better learn any function (e.g. in clustering or classification). In this paper, we aim to learn how to produce chromosomes that improve clustering after every generation, by optimising a clustering criterion. In other words, the clustering criterion plays the role of the function being oprimised.

Usually, we need to make some assumptions about the underlying structure of the dataset distribution. Often, semi-superised algorithms make use at least one of the following assumptions [41]:

- Smoothness assumption. This assumption is based on the notion that two points, close to each other, in a high density area, is probable to share the same label.
- Cluster assumption. It is based on the notion that, generally, data tend to create distinct clusters, and those that are in the same cluster have also a high probability to share the same label.
- Manifold assumption. This hypothesis is based on the notion that the data are of low dimensionality. With this assumption, we aim to avoid the problem of very high data dimensionality.
- Low density separation. We usually assume that the boundaries of a cluster are selected to be in low density areas.

In this paper, semi-supervised learning has been used in clustering, in order to optimise an external criterion. In more detail, in terms of the proposed algorithm, for some of the experiments presented we have assumed that we possess a small proportion of labels l of the dataset, which are selected randomly once and, then, the same labeled data are used in every genetic cycle. The proportion of labels that we have used in our experiments ranged between 5% and 20% of the total labels of each dataset. Then, using only these l labels, we have computed the fitness value f of the population, by using one of the external criteria. The evaluation of the algorithm is performed using only the rest of the criteria (and not the one being optimised), which are also being calculated during every experiment. The overall value of a criterion is the value of an external criterion calculated as if we possessed the labels for the whole dataset. Thus, this technique uses both labeled and unlabeled data in order to obtain clusters.

Table 1: Datasets used.

Dataset	**duration or source**	**Classes**	**Size of dataset**	**# of features**
Movie 1	02 : 06 : 21	21	1,222	152×152
Movie 2	01 : 44 : 31	41	1,435	150×150
Libras Movement	UCI	15	360	90
Iris	UCI	3	150	4
MNIST	[13]	10	100	784
folk dances	[42]	5	1012	1000

Essentially, only a small proportion of labels was used in this method for obtaining the fitness values of chromosomes, while the rest of the procedure remained unchanged.

4. Experiments

In order to evaluate the proposed algorithm, we have conducted several experiments using 6 different multidimensional datasets and exploiting several input parameters. All of the datasets presented have several dimensions noted as # of features in Table 1. Although we own the ground truth for all of the datasets, it is only used to calculate external criteria and extract conclusions about the performance of the algorithm when used with internal criteria and semi-supervised clustering. The characteristics of the datasets used, are described in Table 1.

A more detailed description regarding the datasets is presented here. Datasets “Movie 1” and “Movie 2” consist mainly of facial images originate from movies, detected using a face detector. The face detector that has been used is the Viola-Jones variant included in the openCV library In the experiments, the images were scaled, in order to have the same size, considering all the detected facial images of the movie clip and using a mean bounding box, from all bounding boxes that the face detector provided. A problem that might arise is that of anisotropic scaling: the images returned by the detector might have different height and width, which is problematic when scaling towards a mean bounding box, thus we calculate the bigger dimension of the bounding box and then we take the square box that equals this dimension centered to the original bounding box center. Datasets “Libras Movement” and “Iris” originate from UCI [43] and consist of 360 and 150 elements, respectively. The dataset “MNIST handwritten dataset” [13] consists of a subset of 1000 images of the original dataset. The images represent handwritten digits from 0 to 9, and the choice of which samples to keep was a random selection of 100 samples per digit. In order to reduce the size of the dataset to 100 dimensions, Principal Component Analysis (PCA) was used. Lastly, the initial “Folk dances” dataset consists of videos of 5 different traditional dances: Lotzia, Capetan Loukas, Ramna, Stankena and Zablitsena with 180, 220, 220 201 and 192 videos respectively [42], from which histograms were extracted according to [44]. An example of the dataset is illustrated in Figure 5.

The goal of the algorithm is to obtain good clustering results, that satisfy two different requirements. The first is to manage to optimise the given internal clustering

Figure 5: An example of Ramna dance, from the ”Folk dances” dataset.

criterion, while the second is to manage to also induce optimisation to the rest of the internal and external clustering criteria. In an ideal experiment, all the criteria that are calculated in order to check the algorithms performance should also be optimised.

The size of the populations remained unchanged for all the experiments conducted and was set to 200 chromosomes. Every experiment was executed 3 times, thus the results presented here are the average of these runs. We should highlight here that, in every experiment, only one clustering criterion c is being optimised. The values of the rest of the criteria are also calculated during every experiment only for evaluation reasons. In other words, the values of the rest of the criteria are not their best values as if they were being optimised themselves. Instead, their values depend on the clustering obtained by optimising the criterion J. Moreover, the optimisation of a single criterion does not necessarily mean that the rest of the criteria will also be improved, especially when the way in which the criteria are calculated differs a lot.

In order to better assess the performance of the proposed approach we have conducted several comparisons against several other spectral clustering algorithms by analysing the behavior of the standard spectral clustering using different options to create the similarity matrix either using k-nn neighbourhoods or using ε-ball neighbourhoods. In the following Section we highlight the comparison of the proposed approach against different k-nn graphs and we also compare against ε-ball neighbourhood graphs.

4.1. Comparison with k-nn graph

The comparison here concerns the performance of the proposed algorithm compared to k-nn graphs as initial population. In Figure 6, results from “Movie 2” are

Table 2: Libras Movement. Optimising $F-$measure % criterion (internal criteria).

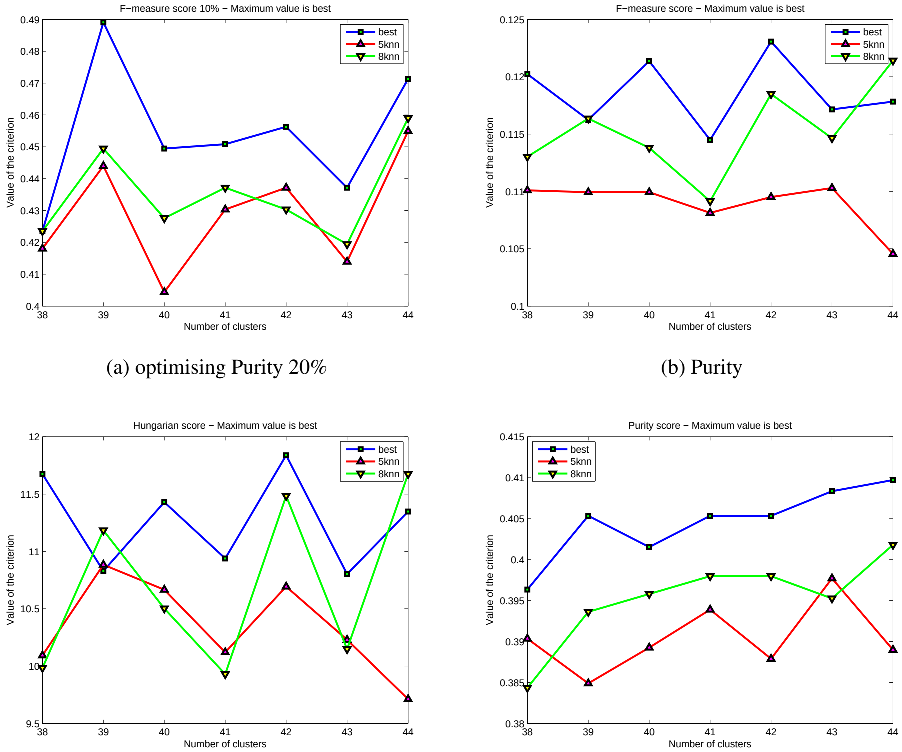
σ	labels%	C	CH		DB		Dunn	
			best	5nn	best	5nn	best	5nn
0.89	10	14	**161.47**	131.35	**0.67**	0.76	**0.10**	0.07
1.33	20	15	**167.21**	110.02	**0.61**	0.79	**0.08**	0.04
2.66	20	14	**141.89**	101.17	**0.69**	0.70	0.05	**0.08**
5.32	10	14	**127.39**	110.85	**0.70**	0.75	**0.12**	0.07

illustrated, with "Purity %" being the optimised criterion and assuming that we possess 20% of the total labels. Axes x, y represent the number of clusters and the value of each criterion, respectively. The "best" line, in the Figure 6a represents the values of this criterion after its optimisation, whereas in the rest figures of the criteria represents the value of the respective criterion (i.e. Purity, Hungarian, etc.) according to the best results of the optimised criterion (here, the "Purity%" criterion). The "5knn" and "8knn" lines represent the values of the criterion if clustering had been performed using the 5 and 8-nearest neighbour graph, respectively. The comparison with the results of the 5 and 8-nearest neighbour graphs is made as a baseline for our method, since, especially the 5-nearest neighbour graph, they are a common choice for data representation.

When optimising "Purity %" criterion, the rest of the external criteria are also improving. Moreover, optimisation of "Purity %" seems to improve the clustering when the number of clusters was set equal to the number of classes, according to internal criteria. Notice that, the way the internal criterion "Davies-Bouldin" is defined, low values mean better clustering has been performed. In Figure 7 the results of dataset "Libras Movement" are presented (according to Figure 6) where the criterion being optimised is "$F-$measure %". When optimising "$F-$measure %" criterion, the rest of the external criteria are also improving. In this example we can also notice that the optimisation of "$F-$measure %" improves the clustering when the number of clusters was set equal to the number of classes, according to internal criteria.

As the number of clusters is not a known parameter in clustering problems, we have experimented with different values of the number of clusters, in order to inspect the performance of the algorithm. In the presented figures we have set the lowest and highest number of clusters in such a way so as to illustrate the results of a wide range of clusters.

In Tables presented here, we have attempted to summarise some of the results of the datasets. The results of the proposed method are represented under the label "best", while "5nn" represent the results of the clustering if the 5-nn graph would have been employed to the data. Tables 2, 3, 4, 5, 6 represent the results of the algorithm, when "$F-$measure %" external criterion was being optimised. In more detail, in Tables 2 and 4 the intenal criteria are presented, while in Tables 3 and 5 the external ones. For Tables 7, 8 and 9 the criteria being optimised are highlighted in every sub-table (from top to bottom "Calinski-Harabasz", "$F-$measure %", "Purity %"). The σ parameter is the heat kernel parameter as in (1), C is the number of clusters, and "labels %" is the percentage of the labels we assumed to possess (only for external criteria).



(a) optimising Purity 20%

(b) Purity

(c) Hungarian

(d) F-measure

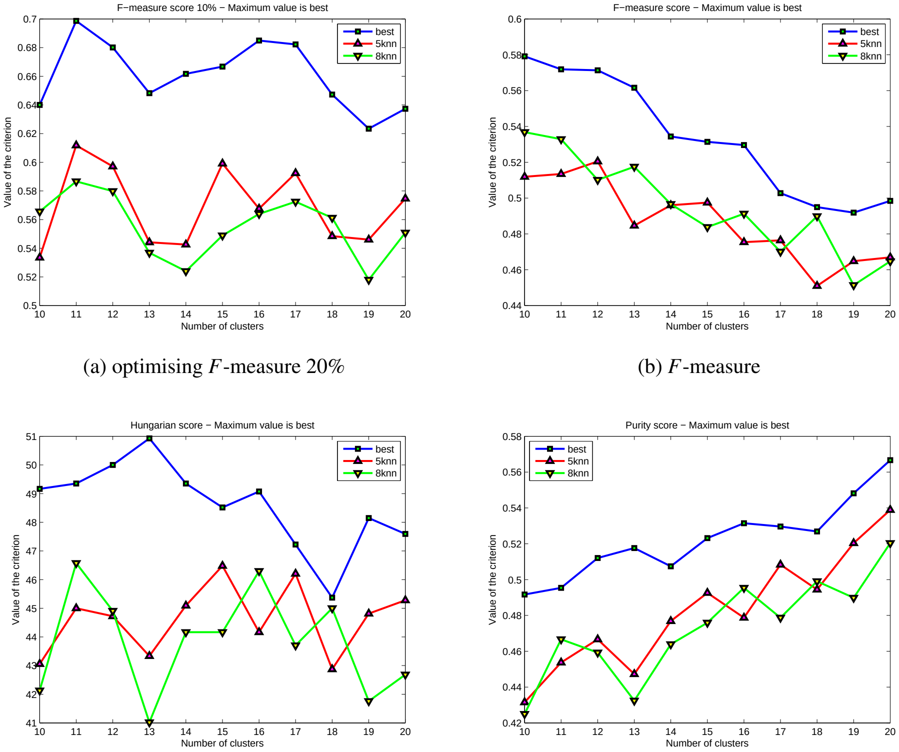
Figure 6: Results for dataset "Movie 2". In every plot axis x, y represent the number of clusters and the value of each criterion respectively. The parameter of heart kernel was set to $\sigma = 75$.

Table 3: Libras Movement. Optimising $F-$measure % criterion (external criteria).

σ	labels%	C	Hungarian		NMI		***F*-measure %**		Purity		$F-$measure	
			best	5nn	best	5nn	best	5nn	best	5nn	best	5nn
0.89	10	14	**48.06**	45.93	**0.64**	0.63	**0.68**	0.64	**0.50**	0.48	**0.51**	0.49
1.33	20	15	**45.93**	45.83	**0.63**	0.62	**0.59**	0.57	**0.50**	0.48	**0.50**	0.48
2.66	20	14	**47.41**	42.96	**0.62**	0.61	**0.58**	0.55	**0.50**	0.46	**0.51**	0.48
5.32	10	14	**45.28**	44.72	**0.63**	0.61	**0.69**	0.66	**0.48**	0.47	**0.51**	0.49

Table 4: Iris. Optimising F-measure % criterion, σ=3.83 (internal criteria).

σ	labels%	C	CH		DB		Dunn	
			best	5nn	best	5nn	best	5nn
3.83	5	3	**140.68**	134.05	**0.67**	0.74	**0.23**	0.13
3.83	10	3	**161.40**	82.17	**0.49**	0.89	**0.16**	0.04
3.83	All	3	**359.03**	162.73	0.60	**0.53**	**0.28**	0.07



(a) optimising F-measure 20%

(b) F-measure

(c) Hungarian

(d) Purity

Figure 7: Results for dataset "Libras Movement". In every plot axis x, y represent the number of clusters and the value of each criterion respectively. The parameter of heat kernel was set to $\sigma = 5$.

Table 5: Iris. Optimising F-measure % criterion, σ=3.83 (external criteria).

σ	labels%	C	Hungarian		NMI		***F*-measure %**		Purity		$F-$measure	
			best	5nn	best	5nn	best	5nn	best	5nn	best	5nn
3.83	5	3	**74.67**	70.67	**0.59**	0.59	**0.83**	0.78	**0.76**	0.74	**0.77**	0.75
3.83	10	3	**65.56**	58.22	**0.48**	0.33	**0.80**	0.71	**0.68**	0.60	**0.69**	0.62
3.83	All	3	**85.11**	69.78	**0.69**	0.49	-	-	**0.85**	0.72	**0.85**	0.72

Table 6: Mnist handwritten digits. Optimising F-measure % criterion, σ=5.

σ	labels%	C	CH		DB		Hungarian		***F*-measure %**		Purity		$F-$measure	
			best	5nn	best	5nn	best	5nn	best	5nn	best	5nn	best	5nn
5	10	9	**486.71**	478.16	0.84	**0.77**	**68.87**	62.17	**0.79**	0.71	**0.72**	0.66	**0.68**	0.63
5	10	10	340.36	**345.17**	**0.78**	0.86	**70.92**	62.25	**0.76**	0.65	**0.73**	0.67	**0.72**	0.65
5	10	11	350.05	**370.72**	0.77	**0.76**	**67.51**	60.04	**0.76**	0.63	**0.70**	0.65	**0.72**	0.66

Table 7: Folk dances dataset. Optimising Calinski-Harabasz criterion.

σ	labels%	C	**Calinski-Harabasz**		Davies-Bouldin		NMI		Purity	
			best	5nn	best	5nn	best	5nn	best	5nn
0.45		5	**77.803**	40.665	**2.116**	3.317	**0.32**	0.255	**0.468**	0.434
0.9		5	**71.026**	38.309	**2.745**	3.252	**0.281**	0.271	**0.441**	0.434
1.8		5	**74.923**	43.649	**2.292**	3.013	**0.312**	0.291	**0.469**	0.463

Table 8: Movie 1. Top to bottom optimising Calinski-Harabasz, F-measure %, Purity % criteria.

σ		C	**Calinski-Harabasz**		Davies-Bouldin		Hungarian		Purity	
			best	5nn	best	5nn	best	5nn	best	5nn
5000		21	**161.239**	121.659	1.165	**1.162**	**20.922**	20.758	0.468	**0.475**
15000		21	**161.011**	123.922	1.208	**1.103**	**21.495**	21.167	0.462	**0.477**
20000		21	**149.195**	121.413	1.169	**1.072**	**21.113**	20.404	0.459	**0.475**

σ	labels %	C	Hungarian		**F-measure %**		Purity		$F-$measure total	
			best	5nn	best	5nn	best	5nn	best	5nn
20000	10	22	**21.17**	19.42	**0.31**	0.29	**0.48**	0.46	**0.24**	0.22
10000	20	22	**21.79**	19.99	**0.29**	0.26	0.47	**0.48**	**0.23**	0.23
15000	20	22	**20.51**	20.51	**0.28**	0.26	0.47	**0.48**	**0.24**	0.23
20000	20	22	**20.73**	19.37	**0.29**	0.27	**0.49**	0.47	**0.24**	0.23

σ	labels %	C	Hungarian		**Purity %**		Purity		$F-$measure	
			best	5nn	best	5nn	best	5nn	best	5nn
5000	20	21	**20.786**	19.858	**0.493**	0.485	**0.487**	0.479	**0.232**	0.226
10000	20	20	**21.877**	21.304	**0.504**	0.493	**0.483**	0.473	**0.245**	0.240
15000	20	20	**21.086**	20.949	**0.503**	0.497	**0.477**	0.472	**0.241**	0.240

4.2. Comparison with ε-neighbourhood graph

This structure of graphs is created by connecting each node to all other nodes which have distance d_{ij} smaller than a threshold ε. Initial population was created using ε-neighbourhood graphs, in a similar way as the construction of the initial population using k-nn graphs. In fact, the value of ε was set as the mean value of the k-nn graph values. In more detail, the mean values of k-nn graph values (with $k = 3,..,8$) were used as the ε value of a ε-ball neighbourhood graph. These six new graphs constitute the backbone of the initial population and they will be referred as $3-8$-ball neighbourhood graphs although $3,4,..,8$ values do not refer to the ε value, for brevity. The rest of the procedure, in order to enrich the initial population, was exactly the same as described in Section 3.1.

In Tables 10 and 11 comparative results of the different approaches used to create the initial population are illustrated. In Table 10 the results of the F-measure optimised criterion are presented, where "k-nn graph" and "ε-ball neighbourhood graph" refer to the graph that has been used for the construction of the initial population, while"best" refers the results of the proposed algorithm. Moreover, "5nn" represents the results of the algorithm if clustering had been performed using the 5-nearest neighbour graph and "5-ball" represents the results of the algorithm if clustering had been performed using the 5-ball neighbour graph. Lastly, "bestInitial" refers to the results if the clustering would have been performed on the best initial population among the k-nn graphs ($k = 3,...,8$) and the best initial population among the ε-neighbourhood graphs ($3-8$-ball graphs), respectively.

Table 9: Movie 2. Top to bottom optimising Calinski-Harabasz, F-measure %, Purity % criteria.

σ		C	**Calinski-Harabasz**		Davies-Bouldin		Hungarian		Purity	
			best	5nn	best	5nn	best	5nn	best	5nn
25		40	**81.917**	70.737	1.240	**1.204**	**15.889**	15.447	**0.400**	0.398
50		41	**76.269**	69.302	**1.144**	1.257	**16.353**	15.819	**0.410**	0.408
75		41	**78.449**	66.245	1.226	**1.200**	**16.121**	15.981	0.401	**0.402**
150		40	**82.090**	66.393	**1.183**	1.248	**16.167**	15.772	**0.403**	0.391

σ	labels %	C	Hungarian		**$F-$measure %**		Purity		$F-$measure total	
			best	5nn	best	5nn	best	5nn	best	5nn
50	10	40	**16.19**	15.77	**0.33**	0.32	**0.41**	0.39	**0.17**	0.17
25	20	41	**15.96**	15.42	**0.26**	0.24	**0.40**	0.40	**0.17**	0.17
50	20	41	**16.26**	15.96	**0.25**	0.23	0.41	**0.41**	**0.17**	0.17
75	20	41	**16.33**	16.28	**0.25**	0.25	**0.40**	0.40	**0.17**	0.17

σ	labels %	C	Hungarian		**Purity %**		Purity		$F-$measure	
			best	5nn	best	5nn	best	5nn	best	5nn
50	20	41	**32.733**	32.706	**0.404**	**0.404**	**0.380**	0.378	0.458	**0.461**
75	20	41	**10.229**	10.120	**0.451**	0.430	**0.401**	0.394	**0.109**	0.108
150	20	41	**17.267**	16.667	**0.515**	0.497	**0.455**	0.440	**0.181**	0.178

Table 10: Libras Movement. Optimising $F-$measure 20% criterion using k-nn and ε-ball neighbourhood graphs

	best	5nn	bestInitial
k-nn graph	0.6667	0.5991	0.6046

	best	5-ball	bestInitial
ε-ball neighbourhood graph	0.5602	0.4815	0.4907

In Table 11 we keep the same notation as before, and present the results of the rest criteria when the F-measure criterion is being optimised. Again, we need to emphasise that the values in this table are not the best values of each criterion, but the values that each criterion takes according to the optimised criterion. We have also emphasised (bold words) the different methods used to evaluate the algorithm.

From Tables 10 and 11 we observe that the proposed approach is able to improve also the performance of the ε-neighbourhood populations for several internal and external criteria. It is also obvious that the results of the proposed algorithm, using the k-nn graphs, are superior to the results of the method that uses the ε-ball neighbourhood graphs. This superiority is not only evident to the optimising criterion, but also to the rest of the criteria presented in Table 10.

5. Conclusion

We have presented a novel algorithm that makes use of evolutionary algorithms in order to achieve good clustering results, with the aid of nearest neighbour graphs. It is important to remark that the algorithm is general and can be used to manipulate a wide variety of different problems, such as clustering and dimensionality reduction. The technique of using nearest neighbour graphs as initial population appears to yield satisfactory results, in terms of both internal and external criteria.

Table 11: Libras Movement. Rest of the criteria, optimising $F-$measure 20% criterion using k-nn and ε-ball neighbourhood graphs

	total F-measure	Hungarian	Purity	Calinski-Harabasz	Davies-Boulding
k-nn graph **best**	0.5314	48.52	0.5231	95.88	0.7818
k-nn graph **5knn**	0.48.75	46.48	0.4946	84.38	0.9231
k-nn graph **bestInitial**	0.4838	45.93	0.4815	92.84	0.8682
ε-ball neighbourhood graph **best**	0.3458	32.78	0.3444	102.3	0.8729
ε-ball neighbourhood graph **ε-ball**	0.3073	27.87	0.3056	85.92	1.085
ε-ball neighbourhood graph **bestInitial**	0.3577	32.04	0.3417	68.01	0.9481

In the future, we aim to improve the proposed evolutionary algorithm, by optimising even different criteria, or even use multiple of them in order to decide which chromosome is best. We shall also focus our efforts on creating an even better initial population, for example by including more than only random variations of the nearest neighbour graphs. Indeed, the authors are working towards incorporating the proposed evolutionary based graph optimisation approach to classification and dimensionality reduction.

Acknowledgments

This research has been co–financed by the European Union (European Social Fund - ESF) and Greek national funds through the Operation Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: THALIS–UOA–ERASITECHNIS MIS 375435.

References

[1] G. B. Fogel, D. W. Corne, Evolutionary computation in bioinformatics, Morgan Kaufmann, 2002.

[2] Y.-K. Shih, S. Parthasarathy, Identifying functional modules in interaction networks through overlapping markov clustering, Bioinformatics 28 (18) (2012) i473–i479. `doi:10.1093/bioinformatics/bts370`.

[3] D. Saravanan, S. Srinivasan, A proposed new algorithm for hierarchical clustering suitable for video data mining, Data Mining and Knowledge Engineering 3 (9) (2011) 569–572.

[4] J. F. Khan, S. M. Bhuiyan, R. R. Adhami, Image segmentation and shape analysis for road-sign detection, Intelligent Transportation Systems, IEEE Transactions on 12 (1) (2011) 83–96.

[5] S. C. Hoi, W. Liu, S.-F. Chang, Semi-supervised distance metric learning for collaborative image retrieval and clustering, ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP) 6 (3) (2010) 18.

[6] G. V. Mantena, X. Anguera, Speed improvements to information retrieval-based dynamic time warping using hierarchical k-means clustering., in: ICASSP, IEEE, 2013, pp. 8515–8519.

[7] A. K. Jain, M. N. Murty, P. J. Flynn, Data clustering: a review, ACM Computing Surveys 31 (3) (1999) 264–323.

[8] A. K. Jain, Data clustering: 50 years beyond k-means., in: ECML/PKDD (1), Vol. 5211, Springer, 2008, pp. 3–4.

[9] N. Grira, M. Crucianu, N. Boujemaa, Unsupervised and semi-supervised clustering: a brief survey, A review of machine learning techniques for processing multimedia content, Report of the MUSCLE European Network of Excellence (FP6).

[10] F. Bach, M. Jordan, Learning spectral clustering, Tech. rep., UC Berkeley (2003).

[11] S. Foucher, L. Gagnon, Automatic detection and clustering of actor faces based on spectral clustering techniques., in: CRV, IEEE Computer Society, 2007, pp. 113–122.

[12] A. Adler, M. Elad, Y. Hel-Or, Probabilistic subspace clustering via sparse representations., IEEE Signal Process. Lett. 20 (1) (2013) 63–66.

[13] F. R. Bach, M. I. Jordan, Learning spectral clustering, with application to speech separation., Journal of Machine Learning Research 7 (2006) 1963–2001.

[14] S. Makino, T.-W. Lee, H. Sawada, Blind speech separation, Springer, 2007.

[15] M. Belkin, P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation., Neural Computation 15 (6) (2003) 1373–1396.

[16] S. E. Schaeffer, Graph clustering., Computer Science Review 1 (2007) 27–64.

[17] A. Peiravi, H. R. Mashhadi, S. H. Javadi, An optimal energy-efficient clustering method in wireless sensor networks using multi-objective genetic algorithm., Int. J. Communication Systems 26 (1) (2013) 114–126.

[18] U. Maulik, S. Bandyopadhyay, Genetic algorithm-based clustering technique., Pattern Recognition 33 (9) (2000) 1455–1465.

[19] C. A. Murthy, N. Chowdhury, In search of optimal clusters using genetic algorithms., Pattern Recognition Letters 17 (8) (1996) 825–832.

[20] S. Salcedo-Sanz, S. Jiménez-Fernández, L. Carro-Calvo, J. Del Ser, J. Portilla-Figueras, et al., A new grouping genetic algorithm for clustering problems, Expert Systems with Applications 39 (10) (2012) 9695–9703.

[21] E. R. Hruschka, R. J. G. B. Campello, A. A. Freitas, A. P. L. F. De Carvalho, A survey of evolutionary algorithms for clustering, Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 39 (2) (2009) 133–155.

[22] J. H. Holland, Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence, MIT Press, Cambridge, MA, USA, 1992.

[23] F. Pérez-Cruz, O. Bousquet, Kernel methods and their potential use in signal processing, IEEE Signal Processing Magazine 21 (2004) 57–65.

[24] U. Luxburg, A tutorial on spectral clustering, Statistics and Computing 17 (4) (2007) 395–416.

[25] A. Y. Ng, M. I. Jordan, Y. Weiss, et al., On spectral clustering: Analysis and an algorithm, Advances in neural information processing systems 2 (2002) 849–856.

[26] L. Vendramin, R. J. G. B. Campello, E. R. Hruschka, On the comparison of relative clustering validity criteria., in: SDM, SIAM, 2009, pp. 733–744.

[27] H. S. Christopher D. Manning, Prabhakar Raghavan, Introduction to Information Retrieval, Cambridge University Press, 2008.

[28] T. Caliński, J. Harabasz, A dendrite method for cluster analysis, Communications in Statistics Simulation and Computation 3 (1) (1974) 1–27.

[29] D. L. Davies, D. W. Bouldin, A cluster separation measure, Pattern Analysis and Machine Intelligence, IEEE Transactions on (2) (1979) 224–227.

[30] J. C. Dunn, Well-separated clusters and optimal fuzzy partitions, Journal of cybernetics 4 (1) (1974) 95–104.

[31] B. S. S. M. zu Eissen, F. Wißbrock, On cluster validity and the information need of users, ACTA Press (2003) 216–221.

[32] Y. Zhao, G. Karypis, Criterion functions for document clustering: Experiments and analysis (2001).

[33] Z. He, X. Xu, S. Deng, K-anmi: A mutual information based clustering algorithm for categorical data, CoRR.

[34] J. Munkres, Algorithms for the assignment and transportation problems, Journal of the Society of Industrial and Applied Mathematics 5 (1) (1957) 32–38.

[35] K. A. De Jong, An analysis of the behavior of a class of genetic adaptive systems, Ph.D. thesis, University of Michigan, Ann Arbor, university Microfilms No. 76-9381 (1975).

[36] X. Zhu, A. B. Goldberg, Introduction to semi-supervised learning, Synthesis lectures on artificial intelligence and machine learning 3 (1) (2009) 1–130.

[37] O. Chapelle, B. Schölkopf, A. Zien, Semi-Supervised Learning, MIT Press, 2006.

[38] M. Guillaumin, J. Verbeek, C. Schmid, Multimodal semi-supervised learning for image classification, in: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE, 2010, pp. 902–909.

[39] M.-F. Balcan, A. Blum, P. P. Choi, J. D. Lafferty, B. Pantano, M. R. Rwebangira, X. Zhu, Person identification in webcam images: An application of semi-supervised learning.

[40] V. Sindhwani, S. S. Keerthi, Large scale semi-supervised linear svms, in: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, ACM, 2006, pp. 477–484.

[41] O. Chapelle, B. Schölkopf, A. Zien, et al., Semi-supervised learning, Vol. 2, MIT press Cambridge, 2006.

[42] I. Kapsouras, S. Karanikolos, N. Nikolaidis, A. Tefas, Feature comparison and feature fusion for traditional dances recognition, in: Engineering Applications of Neural Networks, Springer, 2013, pp. 172–181.

[43] K. Bache, M. Lichman, UCI machine learning repository (2013).
URL `http://archive.ics.uci.edu/ml`

[44] A. Iosifidis, A. Tefas, I. Pitas, Minimum class variance extreme learning machine for human action recognition, Circuits and Systems for Video Technology, IEEE Transactions on 23 (11) (2013) 1968–1979.