Evolution of Graphs for Unsupervised Learning

Christina Chrysouli and Anastasios Tefas

Abstract In this paper, we propose a novel method which adopts evolutionary techniques so as to optimize a graph structure. The method that was developed has been applied in clustering problems, where spectral graph clustering technique has been used. In order to use evolutionary algorithms initial population has been created consisting of nearest neighbor graphs and variations of these graphs, which have been properly altered in order to form chromosomes. Since it was observed that initial population is crucial for the performance of the algorithm, several techniques have been considered for the creation. A fitness function was used in order to decide about the appropriateness of the chromosomes. The major advantage of our approach is that the algorithm is generic and can be used to all problems that are, or can be, modeled as graphs, such as dimensionality reduction and classification. Experiments have been conducted on a traditional dance dataset and other multidimensional datasets, providing encouraging results.

Keywords Spectral clustering · Similarity graphs · Evolutionary algorithms

1 Introduction

The aim of clustering is to discover the natural grouping of a dataset, such that similar samples are placed in the same group, while dissimilar samples are placed into different ones. Clustering has been used in order to solve a diversity of problems, including bioinformatics, data mining, image analysis, information retrieval etc. A detailed survey on clustering applications can be found in [1] and a more recent study in [2].

C. Chrysouli · A. Tefas (✉)
Department of Informatics, Aristotle University of Thessaloniki,
University Campus, 54124 Thessaloniki, Greece
e-mail: tefas@aiia.csd.auth.gr

C. Chrysouli
e-mail: chrysouli@aiia.csd.auth.gr

J.J. Merelo et al. (eds.), *Computational Intelligence*,
Studies in Computational Intelligence 620, DOI 10.1007/978-3-319-26393-9_2

Spectral graph clustering is widely used and have received a lot of attention nowadays, as it can be applied to a wide variety of practical problems, such as computer vision and speech analysis. Spectral graph clustering [3] 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 clusters. In all clustering techniques, and thus also in spectral clustering, nodes that originate from the same cluster should have high similarity values, whereas nodes from different clusters should have low similarity values. In [4] the authors summarize some of the applications of spectral graph clustering.

So far, some evolutionary-based approaches to the problem of clustering have been proposed throughout the years. In [5] the authors proposed a genetic algorithm in order to search for the cluster centers by minimizing a clustering metric, while in [6] authors aim to find the optimal partition of the data, using a genetic algorithm, without searching all possible partitions. A more detailed survey of evolutionary algorithms for clustering is presented in [7].

In the proposed approach, similarity graphs are evolved, which have been transformed properly in order to play the role of the chromosomes in the employed genetic algorithm [8]. In order to use evolutionary algorithms we construct the initial population with the aid of k-nearest neighbor graphs which, then, are transformed to one-dimensional binary strings and undergo genetic operators.

The remainder of this paper is organized as follows. In Sect. 2, the problem that we attempt to solve is stated and some general aspects that concern the algorithm are discussed, including similarity graph construction, and spectral clustering issues. Section 3, presents the proposed evolutionary algorithm in detail. In Sect. 4, experimental results of the algorithm are described. Finally, in Sect. 5, conclusions are drawn and future work is discussed.

2 Problem Statement

Clustering is the process of partitioning a usually large dataset into groups, according to a similarity (or dissimilarity) measure. The goal is to place samples that have a small distance from each another, to the same cluster, whereas samples that are at a large distance from each another are placed to different clusters. 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 of 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. If

we assume that each vertex v_i represents a data sample, then two nodes v_i, 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 $s_{i,j}$. The problem of clustering may now 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.

Before constructing a similarity graph, we need to define a similarity function on the data. The most common similarity function **S** is the Gaussian similarity function (heat kernel). Heat kernel between two graph nodes is defined as:

$$\mathbf{S} = \mathbf{h_{i,j}} = \mathbf{exp}\left(-\frac{\left\|\mathbf{v_i} - \mathbf{v_j}\right\|^2}{\sigma^2}\right), \tag{1}$$

where σ is a parameter that defines the width of the neighborhood.

Generally, the most common choice of similarity graphs are k-nearest neighbor graphs (to be called k-nn graphs) because of their simplicity as well as their sparsity. The aim of a k-nn graph **A** is to connect node v_i with node v_j if v_j is among the k nearest neighbors of v_i. This results in a directed graph which is easily transformed to an undirected by simply ignoring the directions of the edges. In the proposed method, an undirected graph was used, in order to construct the similarity graph.

However, it is well known that spectral clustering is very sensitive to the choice of the similarity graph that is used for constructing the Laplacian [9]. Indeed, selecting a fixed k parameter for the k-nn graph is very difficult and different values lead to dramatically different clusterings. Optimizing the clustering 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 optimize specific clustering criteria, that are considered as fitness functions, with respect to the underlying graph, which is transformed to a chromosome solution.

2.2 Spectral Graph Clustering

Spectral graph clustering [3], 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 many clustering applications [4].

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

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

where **W** is the adjacency matrix, and **v**, λ are the eigenvectors and eigenvalues respectively.

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

Fig. 1 The **S** matrix represents the full similarity matrix constructed using (1). The **A** matrix represents a k-nn graph, which has undergone genetic operators. The $\odot$ operator performs element-wise multiplication, resulting in a sparse matrix **W**, which only contains elements in places where **A** matrix contains elements

Although many variations of graph Laplacians exist [9], we focus on the normalized graph Laplacian **L** [10] defined as:

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

where **W** is the adjacency matrix, with $w_{i,j} = w_{j,i} \geq 0$, **D** is the degree matrix and **I** is the identity matrix. The smallest eigenvalue of **L** is 0, which corresponds to the eigenvector $\mathbf{D}^{-1/2}\mathbf{1}$. The **L** matrix is always positive semi-definite and has n non-negative real-valued eigenvalues $\lambda_1 \leq \cdots \leq \lambda_n$. 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 **L** matrix, where **W** is defined as:

$$\mathbf{W} = \mathbf{S} \odot \mathbf{A}, \tag{4}$$

S represents the full similarity matrix obtained using (1) and **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 **W**, only containing elements in places where **A** matrix contains elements. An example of the $\odot$ operator is illustrated in Fig. 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 Algorithm

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 optimize the structure of the graph, by optimizing 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.

Let J be a clustering criterion that depends on the similarity graph $\mathbf{W}$. However, the optimization 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 (4), the optimization problem is defined as:

$$\underset{\mathbf{A}}{optimize}\ J(\mathbf{A}), \tag{5}$$

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

3.1 Construction of Initial Population

In our algorithm, we use the sparse matrices that originate from k-nn graphs, resulting in an initial population that consists of matrices with binary elements. 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, thus, several, arbitrary rules exist; in the proposed method, we have used multiples of the data diameter.

First, we calculate k-nearest neighbor matrices $\mathbf{A}$, with $k = 3, \ldots, 8$, which constitute the backbone of the initial population. Next step is to enrich the population with nearly k-nearest neighbor matrices. In order to achieve that, we alter the k-nearest neighbor matrices that have already been calculated, by converting a small proportion of 0's, from $\mathbf{A}$ matrices, to 1's and vice versa. This process guarantees that the proportion of 1's and 0's will remain the same in the new matrix. 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, 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.

From the various experiments conducted, we have concluded that the selection of the parameter k of the nearest neighbor graphs is crucial to the clustering results, as illustrated in Fig. 2. Figure 2a presents a dataset that consists of two classes with each one having a different color. Figure 2b, c represent the clustering results when a 3 and a 5-nearest neighbor graph were used, respectively.

Before proceeding to the 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 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, forming the one-dimensional vector (Fig. 3).

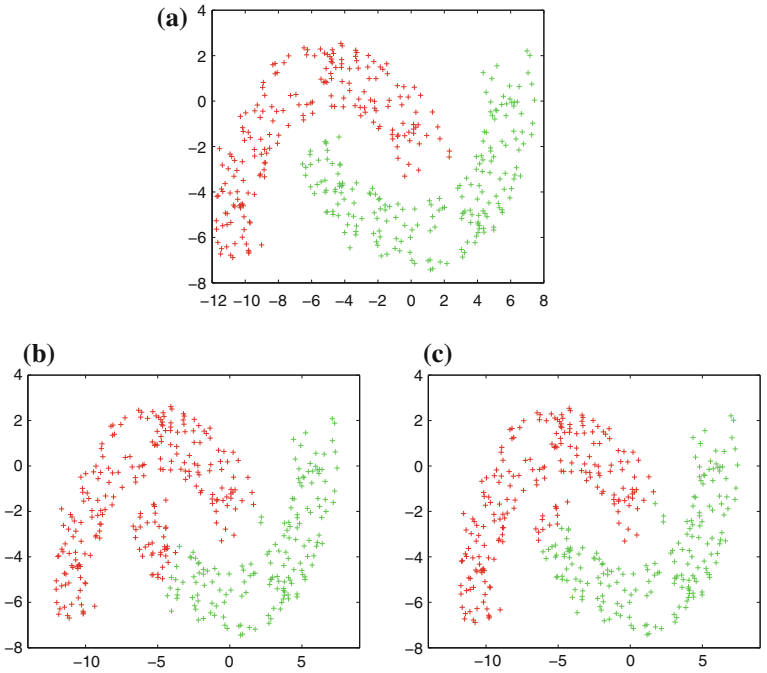

Fig. 2 The effect of k-nearest neighbor graphs in clustering. In Fig. 2a the two classes of the dataset are presented. Figure 2b, c represent the clustering results when a 3 and a 5-nearest neighbor graph were used, respectively. Notice the difference in clustering results especially when the data are close to both classes

$$\begin{bmatrix} 1&0&0&0&1&0\\ 0&1&0&1&0&0\\ 0&0&1&1&1&0\\ 0&1&1&1&0&1\\ 1&0&1&0&1&0\\ 0&0&0&1&0&1 \end{bmatrix} \rightarrow \begin{bmatrix} 0&0&0&1&0\\ 0&1&0&0\\ 1&1&0\\ 0&1\\ 0 \end{bmatrix} \rightarrow \begin{bmatrix} 000100100110010 \end{bmatrix}$$

Fig. 3 The way a k-nn graph $\mathbf{A}$ is transformed into a, one-dimensional vector, chromosome. We only keep the elements of the *upper* diagonal, as the matrix is constructed to be symmetric, resulting in a matrix like the one in the *middle*. Then, this matrix is accessed horizontally, in order to obtain the desirable result, the chromosome

3.2 Optimization of the Solutions

The novelty of the proposed algorithm is based on the way that we select to optimize the solutions of the problem, by optimizing a clustering criterion J, as previously defined in (5). 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 depend on quantities and features inherent

to the dataset, whereas calculation of external criteria implies that we have some knowledge about the dataset in advance (i.e. ground truth).

In the recent literature, many different clustering criteria [11] have been proposed. Some of the most common internal criteria are Calinski-Harabasz index [12], Davies-Bouldin index [13] and Dunn's index [14], whereas some external criteria are purity [15], F-measure [16], a measure based on Hungarian algorithm [17] and normalized mutual information [18]. Some of the aforementioned internal criteria have been used both for optimization and evaluating the performance of the algorithm, whereas the external criteria only for evaluation.

As the value of such criteria cannot be optimized, without the use of derivatives, we have employed evolutionary techniques in order to solve this problem. The optimization 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

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 [19], 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 ith chromosome to be selected, if f_i is its fitness value is defined as:

$$p_i = \frac{f_i}{\Sigma_{j=1}^{N} f_j}. \tag{6}$$

Next, we combine the selected chromosomes, based on the crossover rate which was set to 0.7, in order to produce new ones. In the proposed algorithm, a single crossover point is randomly selected for every set of chromosomes and the subsequences that are formed are exchanged respectively. Then, we randomly choose a small proportion of the chromosomes, based on the mutation rate which was set to 0.4, to undergo mutation, that is the random change of some elements of a chromosome. In order to guarantee that the newly produced chromosomes will not have been altered too much we perform mutation by converting 1 % of 0's to 1's and vice versa.

After the application of genetic operations to the chromosomes, the new generation has been formed. In order to perform spectral clustering (Sect. 2.2), we need to reconstruct the k-nearest neighbor 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 Fig. 1. Spectral clustering [10] may now be performed on $\mathbf{L}$ as in (3).

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 optimized criterion has not been altered for 5 consecutive generations.

4 Experiments

In order to evaluate the proposed algorithm, we have conducted several experiments using 3 different datasets and exploiting several input parameters. The characteristics of the datasets used, are described in Table 1.

Datasets "Movie 1" and "Movie 2" consist mainly of facial images originate from movies, detected using a face detector. 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. 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, from which histograms were extracted according to [20]. An example of the dataset is illustrated in Fig. 4.

The size of the populations remained unchanged for all the experiments conducted and was set to 200 chromosomes. Every experiment was executed 3 times, so 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 optimized. 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

Table 1 Datasets used

Dataset	Duration	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
Folk dances	–	5	1012	1000

Fig. 4 An example of Ramna dance, from the "Folk dances" dataset

values as if they were being optimized themselves. Instead, their values depend on the clustering obtained by optimizing the criterion c. Moreover, the optimization 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 tables presented here, we have attempted to summarize 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. For Tables 2, 3 and 4 the criterion being optimized is highlighted. The σ parameter is the heat kernel parameter as in (1) and C is the

Table 2 Folk dances dataset. Optimizing Calinski-Harabasz criterion

σ	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 3 Movie 1. Optimizing Calinski-Harabasz criterion

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**

Table 4 Movie 2. Optimizing Calinski-Harabasz criterion

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

number of clusters. We observe that in almost all cases the external criteria agrees with the internal optimized criterion that the clustering that was performed did actually grouped the data better than if the 5-nn graph would have been employed. In most cases, the other internal criterion also agrees to this conclusion.

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 neighbor 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 neighbor graphs as initial population appears to yield satisfactory results, in terms of both internal and external criteria.

In the future, we aim to improve the proposed evolutionary algorithm, by optimizing 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 neighbor graphs.

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. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM Comput. Surv. (CSUR) 31 **3**, 264–323 (1999)
2. Jain, A.K.: Data clustering: 50 years beyond K-means. Pattern Recognition Letters **31**(8), 651–666 (2010)
3. Jordan, M.I., Bach, F.R.: Learning spectral clustering. In: Advances in Neural Information Processing Systems 16 (2003)

4. Satu S.E.: Graph clustering. Comput. Sci. Rev. 1 **1**, 27–64 (2007)
5. Maulik, U., Bandyopadhyay, S.: Genetic algorithm-based clustering technique. Pattern Recognit. **33**(9), 1455–1465 (2000)
6. Murthy, C.A., Chowdhury, N.: In search of optimal clusters using genetic algorithms. Pattern Recognit. Lett. **17**(8), 825–832 (1996)
7. Eduardo H.R., Campello, R.G.B., Freitas, A.A., De Carvalho, A.C.P.L.F.: A survey of evolutionary algorithms for clustering. IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. **39**(2), 133–155 (2009)
8. John H.H.: Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artific. 211 (1992)
9. Von, U.L.: A tutorial on spectral clustering. Stat. Comput. **17**(4), 395–416 (2007)
10. Andrew Y.Ng., Jordan, M.I., Yair, W.: On spectral clustering: analysis and an algorithm. Adv. Neural Inf. Process. Syst. **2**, 849–856 (2002)
11. Vendramin, L., Campello, R.J.G.B., Hruschka, E.R.: On the Comparison of Relative Clustering Validity Criteria, In: SDM, pp. 733–744 (2009)
12. Caliński, T., Harabasz, J.: A dendrite method for cluster analysis. Commun. Stat. Theory Methods **3**(1), 1–27 (1974)
13. Davies, D.L., Bouldin, D.W.: A cluster separation measure. IEEE Trans. Pattern Anal. Mach. Intell. **2**, 224–227 (1979)
14. Dunn, J.C.: Well-separated clusters and optimal fuzzy partitions. J. Cybern. **4**(1), 95–104 (1974)
15. Ying, Z., Karypis, G.: Criterion functions for document clustering: experiments and analysis. Technical Report (2001)
16. Eissen, Z., Meyer, B.S.S., Wißbrock, F.: On cluster validity and the information need of users, pp. 216–221. ACTA Press (2003)
17. Munkres, J.: Algorithms for the assignment and transportation problems. J. Soc. Ind. Appl. Math. **5**(1), 32–38 (1957)
18. He, Z., Xu, X., Deng, S.: K-ANMI: a mutual information based clustering algorithm for categorical data (2005)
19. De Jong, K.A.: An analysis of the behavior of a class of genetic adaptive systems. PhD diss., PhD thesis, University of Michigan, Dissertation Abstracts International. **36**(10) (1975)
20. Iosifidis, A., Tefas, A., Ioannis, P.: Minimum class variance extreme learning machine for human action recognition. IEEE Trans. Circ. Syst. Video Technol. **23**(11), 1968–1979 (2013)

http://www.springer.com/978-3-319-26391-5

Computational Intelligence
International Joint Conference, IJCCI 2014 Rome, Italy, October 22-24, 2014 Revised Selected Papers
Merelo, J.J.; Rosa, A.; Cadenas, J.M.; Dourado, A.; Madani, K.; Filipe, J. (Eds.)
2016, XIX, 412 p. 111 illus., 58 illus. in color., Hardcover
ISBN: 978-3-319-26391-5