Additive Spectral Method for Fuzzy Cluster Analysis of Similarity Data Including Community Structure and Affinity Matrices

Boris Mirkin[a], Susana Nascimento[b]

[a]*Department of Computer Science*
Birkbeck University of London
London WC1E 7HX, UK
and Division of Applied Mathematics and Informatics, State University - Higher School of Economics
Moscow, Russian Federation

[b]*Department of Computer Science and Centre for Artificial Intelligence (CENTRIA)*
Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa
2829-516 Caparica, Portugal

Abstract

An additive spectral method for fuzzy clustering is proposed. The method operates on a clustering model which is an extension of the spectral decomposition of a square matrix. The computation proceeds by extracting clusters one by one which makes the spectral approach quite natural. The iterative extraction of clusters, also, allows us to draw several stopping rules to the procedure. This applies to several relational data types differently normalized: network structure data (the first eigen-vector subtracted), affinity between multidimensional vectors (the pseudo-inverse Laplacian transformation), and conventional relational data including in-house data of similarity between research topics according to working of a research center. We experimentally compare the performance of our method with that of several recent techniques and show its competitiveness.

Preprint submitted to Elsevier *April 26, 2011*

Keywords: spectral fuzzy clustering, additive fuzzy clustering, one-by-one clustering, Lapin transformation, community structure, research activity structure

1. Introduction

This work is motivated by the problem of clustering research topics within a Computer Science research organization according to the similarity between topics derived on the basis of the efforts by researchers engaged in them. Our similarity measure makes it natural to consider it a result of additive action of fuzzy clusters representing the main directions of the organization's research. Our additive model is a natural extension of the crisp additive clustering model [33, 32], which itself is an extension of the principal component analysis based on the spectral decomposition of square matrices. Therefore, we extend the spectral clustering approach, along with its Laplacian data normalization options [34], to the model.

Because of the general nature of the developed method, we compare it with existing approaches to fuzzy clustering. In particular, we apply it to four similarity/dissimilarity data types: (a) ordinary graphs of community structure, (b) affinity similarity data derived from feature based information, (c) small real-world benchmark dissimilarity datasets, and (d) similarity data including the similarity between research topics. The method appears to be completive in our experiments. Moreover, because it is model based there are innate stopping criteria that can help in determining the number of clusters.

The remainder of the paper is organized as follows. Section 2 describes the additive fuzzy clustering model and a Fuzzy ADDItive Spectral clustering method FADDIS derived from it in two versions, depending on the set of eigenvectors utilized for finding clusters. It also describes the normalized Laplace pseudo-inverse transformation, Lapin, which can be usefully applied sometimes to sharpen the similarity data structure. Section 3 describes experiments on application of FADDIS to the four data types above along with comparing it with other fuzzy clustering developments, including those rather recent ones. Section 4 puts FADDIS in the context of the published work in related areas of relational fuzzy clustering, additive clustering, spectral clustering, and community structure detection. Section 5 concludes the paper.

2. Additive Fuzzy Clustering Model and Spectral Clusters

2.1. Additive model and iterative extraction of clusters

Consider a similarity matrix $A = (a_{tt'})$ between elements t, t' of an N-element set T. The structure of this matrix will be represented by a set of fuzzy clusters.

Assume that a fuzzy cluster on T is represented by a fuzzy membership vector $\mathbf{u} = (u_t)$, $t \in T$, such that $0 < u_t < 1$ for all $t \in T$, and an intensity $\mu > 0$ that expresses the extent of significance of the pattern corresponding to the cluster according to the similarity scale.

Our additive fuzzy clustering model involves K fuzzy clusters that reproduce the similarities up to additive errors according to the following equations:

$$a_{tt'} = \sum_{k=1}^{K} \mu_k^2 u_{kt} u_{kt'} + e_{tt'}, \tag{1}$$

where $\mathbf{u}_k = (u_{kt})$ is the membership vector of cluster k, and μ_k its intensity.

The model defines that product $\mu_k^2 u_{kt} u_{kt'}$ expresses that part of the similarity $a_{tt'}$ between elements t, t' that is supplied by cluster k, which depends on both the cluster's intensity and the membership values. The value μ_k^2 summarizes the contribution of intensity and will be referred to as the cluster's weight.

Intuition for model (1) comes from the area of our interest, the analysis of research activities in terms of key-words of a taxonomy of the domain such as the ACM Computing Classification System (ACM-CCS) [1]. Consider, for example, the following list of topics from ACM-CCS classification:

- D.4.2. Operating systems: storage management
- D.4.7 Operating systems: organization and design
- I.2.4. Knowledge representation
- I.2.10 Vision and scene understanding
- I.3.5 Computational geometry and object modeling
- I.4.1 Digitization and image capture

as that of subjects in which a Computer Science department conducts research.. Specifically, assume that a group undertakes research in Operating Systems, cluster OS = {D.4.2, D.4.7}, another group is doing Image Analysis, IA = {I.2.10, I.3.5, I.4.1}, and the third group relates to Hierarchical Structures and their applications, HS = {D.4.2, D.4.7, I.2.4, I.2.10, I.3.5}. Assume that the group research intensities differ, say, are equal to 4, 3, and

2, respectively. Let us assume as well a background similarity between the topics, due to the fact that all belong to the area of Computer Science, as equal to 1. Then it is natural to define the similarity between topics D.4.2 and D.4.7 as the sum of intensities of the clusters they simultaneously belong to: 4 according to OS cluster and 2 according to HS plus the background intensity 1, leading to 4+2+1=7. Similarly, the similarity between topics D.4.2 and I.3.5 will be 2+1=3, and between topics D.4.2 and I.4.1, just the background similarity 1. A similarity matrix can be clearly derived from the clusters. Yet the problem is whether we are able to reconstruct the clustering from this matrix.

The model in (1) extends this approach to fuzzy clusters of research topics. Taking the product of values μu_t and $\mu u_{t'}$ to express the extent of similarity between t and t' in this model, reflects the interpretation of fuzzy memberships as action forces and, also, makes it mathematically convenient. Of course, a different definition of the extent of similarity due to a cluster, involving operations more convenient in the fuzzy logics perspective, such as of maximum or minimum, rather than multiplication, can also be taken without much changing the computational structure of our approach. Yet the formulation in (1) is much convenient because of its mathematical structure as will be seen in the next section.

The problem of fitting model (1) can be formalized by using the least-squares criterion: given matrix $A = (a_{tt'})$, find K fuzzy clusters $\mathbf{u}_k$ along with their intensities μ_k to minimize the sum of squares of the errors, $\sum_{t,t'} e_{tt'}^2$.

The model (1) much resembles the celebrated spectral decomposition of matrix A. Moreover, as it is well known, provided that A is definite semi-positive, the first K eigenvalues and corresponding eigenvectors form a solution to the least-squares problem if no constraints on vectors $\mathbf{u}_k$ are imposed. This can lead to a viable two-step clustering strategy, analogous to that of some forms of spectral clustering [28, 18]. According to this strategy, K elements of the spectral decomposition of A are to be found first, and then these are to be projected onto nonnegative normed vectors to form an admissible solution to model (1). Unfortunately, our preliminary experiments with such a method have not been successful; the method fails to recover even simple cluster structures from similarity matrices.

Therefore, we apply another approach, the one-by-one principal component analysis strategy of iterative extraction for finding one cluster at a time. It has been applied at the case of crisp clusters to produce provably tight clusters assigned with additive contributions to the data scatter [19, 23]. Here we extend this strategy to fuzzy clustering.

Specifically, at each step, we consider the problem of minimization of the one-cluster least-squares criterion

$$E = \sum_{t,t' \in T} (w_{tt'} - \xi u_t u_{t'})^2 \tag{2}$$

with respect to unknown positive ξ weight, so that the intensity μ is the square root of ξ, and fuzzy membership vector $\mathbf{u} = (u_t)$, given similarity matrix $W = (w_{tt'})$.

At the first step, W is taken to be equal to A. Then the matrix changes by subtracting from it the part of similarities accounted for by the found cluster, due to the additivity of model (1). The residual similarity matrix for obtaining the next cluster is defined as

$$W - \mu^2 \mathbf{u}\mathbf{u}'$$

where μ and $\mathbf{u}$ are the intensity and membership vector of the found cluster. In this way, A indeed is additively decomposed according to formula (1) and the number of clusters K can be set during the process, depending on the contributions to the data scatter, rather than beforehand.

2.2. Finding one fuzzy cluster

To see how criterion (2) works, let us specify an arbitrary membership vector $\mathbf{u}$ and find the value of ξ minimizing (2) at this $\mathbf{u}$. Obviously, criterion (2) is a convex function of ξ so that the first-order condition of optimality should solve the problem:

$$\frac{\partial E}{\partial \xi} = -2 \sum_{t,t' \in T} (w_{tt'} - \xi u_t u_{t'}) u_t u_{t'} = 0.$$

This implies that

$$\xi = \frac{\sum_{t,t' \in T} w_{tt'} u_t u_{t'}}{\sum_{t \in T} u_t^2 \sum_{t' \in T} u_{t'}^2}$$

In matrix terms the optimal ξ is

$$\xi = \frac{\mathbf{u}' W \mathbf{u}}{(\mathbf{u}'\mathbf{u})^2} \tag{3}$$

which is obviously non-negative if matrix W is semi-positive definite.

By putting this ξ in equation (2), one can easily derive that

$$E = \sum_{t,t' \in T} w_{tt'}^2 - \xi^2 \sum_{t \in T} u_t^2 \sum_{t' \in T} u_{t'}^2 = S(W) - \xi^2 \left(\mathbf{u}'\mathbf{u}\right)^2,$$

where $S(W) = \sum_{t,t' \in T} w_{tt'}^2$ is the similarity data scatter.

Let us denote the last item by $G(\mathbf{u})$. Then, according to (3),

$$G(\mathbf{u}) = \xi^2 \left(\mathbf{u}'\mathbf{u}\right)^2 = \left(\frac{\mathbf{u}'W\mathbf{u}}{\mathbf{u}'\mathbf{u}}\right)^2, \tag{4}$$

so that the similarity data scatter can be represented as the sum of $G(\mathbf{u})$ and E representing, respectively, its explained and unexplained parts:

$$S(W) = G(\mathbf{u}) + E. \tag{5}$$

Since $S(W)$ (5) is constant, the optimal cluster is to maximize the explained part $G(\mathbf{u})$ (4) or its square root,

$$g(\mathbf{u}) = \xi\mathbf{u}'\mathbf{u} = \frac{\mathbf{u}'W\mathbf{u}}{\mathbf{u}'\mathbf{u}}, \tag{6}$$

The value $g(\mathbf{u})$ in (6) is the celebrated Rayleigh quotient; its maximum is known to be the maximum eigenvalue of matrix W reached at the corresponding eigenvector $\mathbf{u}$, if $\mathbf{u}$ is not constrained.

Also, the formulas above allow us to clear the issue of normalization of the membership vector $\mathbf{u}$. Indeed, the additive fuzzy clustering model in (1) or (2) makes use of the product $\mu\mathbf{u}$, without specifying which part of it is μ and which is $\mathbf{u}$. This is somewhat alleviated by the expression (3) for $\xi = \mu^2$ that relates μ to the scale of W. Yet to attend to the conventional view of independent membership scores, $\mathbf{u}$ has to be normalized so that individual membership values do not exceed 1. The structure of the formulas above suggests a normalization of $\mathbf{u}$ by the Euclidean norm, so that its square, $\mathbf{u}'\mathbf{u} = \sum_t u_t^2 = 1$. This normalization is accepted from now on. It makes the cluster weight simply equal to $\xi = \mathbf{u}'W\mathbf{u}$ and $G(\mathbf{u}) = \xi^2$. The

Euclidean normalization fits well into the spectral approach which uses the same normalization for eigenvectors.

This shows that the spectral clustering approach is a natural way of action in the given context. According to this approach, one should first solve the unconstrained problem of maximization of $g(\mathbf{u})$ and then take its projection to the set of nonnegative fuzzy membership vectors. Our projection operator $\mathcal{P}(\mathbf{z})$ is defined as follows:

$$\mathcal{P}(\mathbf{z}) = \mathbf{u} / \|\mathbf{u}\|, \tag{7}$$

where $\mathbf{u} = (u_t)$ is defined by

$$u_t = \begin{cases} 0, & \text{if } z_t \leq 0; \\ z_t, & \text{if } 0 < z_t < 1; \\ 1, & \text{if } z_t \geq 1. \end{cases} \tag{8}$$

It should be noted that testing $z_t \geq 1$ in the operator $\mathcal{P}$ is redundant because of the assumption that the eigenvector $\mathbf{z}$ is normed so that no component of $\mathbf{z}$ can be greater than 1.

This spectral method, that will be referred to as Fuzzy ADDItive Spectral clustering algorithm FADDIS, can be easily extended to the case when fuzzy clusters are required to form a fuzzy partition so that $\sum_{k=1}^{K} u_{kt} = 1$ for each $t \in T$. To make this constraint working, after each cluster extraction step k, $k = 1, 2, \cdots K-1$, the cumulative belongingness $\alpha_{kt} = \sum_{l=1}^{k} u_{lt}$ should be taken into account in the operator $\mathcal{P}(\mathbf{z})$. For each t, the unity in the definition of u_t (8), should be changed for $1 - \alpha_{kt}$ – this will warrant that $\sum_{k=1}^{K} u_{kt} \leq 1$.

If $\mathbf{z}$ is an eigenvector of W corresponding to an eigenvalue λ, so is $-\mathbf{z}$, which implies that one should consider both $\mathbf{u} = \mathcal{P}(\mathbf{z})$ and $\mathbf{u}^- = \mathcal{P}(-\mathbf{z})$ as candidates for projecting to the set of fuzzy membership vectors. Obviously, $\mathcal{P}(-\mathbf{z})$ picks up the absolute values of the negative components of $\mathbf{z}$. This raises an issue of which one of $\mathbf{u}$ or $\mathbf{u}^-$ should be taken, along with its intensity which is $\mu = \mathbf{u}'W\mathbf{u}$ or $\mu^- = \mathbf{u}^{-'}W\mathbf{u}^-$, respectively. We address this by taking into account the criterion of maximization of contribution $G = \xi^2$ in (5): that one that makes the fourth power of the intensity μ or μ^- greater.

The principle of maximization of the contribution $G = \xi^2$ can be further extended to all the eigenvectors, not only those corresponding to the maximum eigenvalue. Indeed, as will be seen further, for some matrices W, value μ or μ^- computed for a non-maximal eigenvalue λ can be greater than those for the maximum eigenvalue.

Therefore we arrive at two versions of FADDIS differing by the way in which a fuzzy cluster is selected:

(m) from projections of the eigenvectors corresponding to the maximum eigenvalue only;

(a) from projections of all the eigenvectors corresponding to all positive eigenvalues.

FADDIS algorithm

Input: Symmetric similarity matrix A, threshold of the contribution of an individual cluster $\epsilon > 0$, threshold of the total clusters contribution $\tau > 0$.

Output: The number of fuzzy clusters K, cluster membership vectors $\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_K$ as well as their intensity values $\mu_1, \mu_2, ..., \mu_K$ and contributions $G_1(\mathbf{u}_1), G_2(\mathbf{u}_2), ..., G_K(\mathbf{u}_K)$ where indexes k at $G(\mathbf{u}_k)$ reflect the fact that they have been computed at different residual similarity matrices.

0 Initialization: Set $k = 1$ and $W = A$; compute the data scatter $S = \sum_{i,j} w_{ij}^2$.

1 Spectral: Find the set of all positive eigenvalues $\Lambda = \{\lambda\}$ and corresponding normed eigenvectors $Z = \{\mathbf{z}_\lambda\}$ for matrix W.

2 Stop-condition: If Λ is empty, computation stops and outputs whatever clusters, along with their intensities and contributions, have been found so far.

3 Fuzzy cluster projection in either m or a version:

(m) Take the normed eigenvector $\mathbf{z}$ and its negation $-\mathbf{z}$ corresponding to maximum $\lambda \in \Lambda$, use operator $\mathcal{P}$ to compute their fuzzy projections $\mathbf{u}$ and $\mathbf{u}^-$, and take that one that maximizes the contribution, $G(\mathbf{u})$ or $G(\mathbf{u}^-)$, as $\mathbf{u}_k$ along with corresponding $\mu_k = \mathbf{u}_k' W \mathbf{u}_k$ and $G(\mathbf{u}_k)$.

(a) Take eigenvectors $\mathbf{z}$ and $-\mathbf{z}$ corresponding to all $\lambda \in \Lambda$, use operator $\mathcal{P}$ to compute their fuzzified projections, and take that one of them that maximizes the contribution, $G(\mathbf{u})$, as $\mathbf{u}_k$ along with corresponding $\mu_k = \mathbf{u}_k' W \mathbf{u}_k$ and $G(\mathbf{u}_k)$.

4 Stop-condition: Check whether $G(\mathbf{u}_k)/S < \epsilon$ or $\sum_{l=1}^{k} G(\mathbf{u}_l)/S > \tau$. If either is, or both are, true, the computation stops, k is taken as K,

and all found clusters are output. Otherwise, add 1 to k, set W equal to $W - \mu^2 \mathbf{u}_k \mathbf{u}_k'$ and go to step 1.

2.3. Properties of FADDIS algorithm

There are a number of properties of FADDIS procedure:

1. The residual matrix W can get all the eigenvalues negative even if the initial matrix A is semi-positive definite, which may bring the procedure to a halt because formula (3) would lead to a negative ξ in this case – this is reflected in Step 2 of FADDIS.

2. Any matrix A can be equivalently substituted by its symmetric version $\tilde{A} = (A + A')/2$, to exclude complex-valued eigenvalues.

3. The cluster contributions are additive so that each can be expressed as a proportion of the initial similarity data scatter S.

4. The cluster contributions tend to decrease at each step, but they do not necessarily form a monotone decreasing sequence, because the spectral cluster does not necessarily globally minimize criterion (2).

5. The procedure converges so that the total cluster contribution increases at each step.

Most items among the above are based on experimental evidence, but some can be proven in a mathematically rigorous way as follows.

Assertion 1. *At any* $\mathbf{u}$*, the value of the criterion* $g(\mathbf{u})$ *does not change if* $W^s = (W + W')/2$ *is put instead of* W*.*

Proof: Indeed, when $w_{tt'}$ is changed for $w^s_{tt'}$ the only items affected are in the numerator, the sum $w_{tt'}u_tu_{t'} + w_{t't}u_{t'}u_t$. But $w^s_{tt'}u_tu_{t'} + w^s_{t't}u_{t'}u_t = (w_{tt'}+w_{t't})u_tu_{t'} + (w_{t't}+w_{t't})u_{t'}u_t)/2 = w_{tt'}u_tu_{t'} + w_{t't}u_{t'}u_t$, which proves the statement.

Thus, the optimizers of $g(\mathbf{u})$ do not change if W^s is used in (6), which proves the assertion.

Following this statement, it is always assumed in the remainder that the data matrix W has been symmetrized with the transformation $W^s = (W + W')/2$ to warrant all the eigenvalues real.

Assertion 2. *For the iteratively extracted clusters $\mu_k\mathbf{u}_k$ with optimal $\mu_k = \sqrt{\xi_k}$ in (3), even if $\mathbf{u}_k$ are not optimal ($k = 1, 2, ..., K$), their contributions to the original similarity data scatter are additive so that*

$$S(A) = G_1(\mathbf{u}_1) + G_2(\mathbf{u}_2) + ...G_K(\mathbf{u}_K) + E_K, \qquad (9)$$

where E_K is the scatter of the final residual matrix.

Proof: We prove the formula for $K = 2$, which is easy to extend to other K values by induction. Indeed, the scatter of the residual matrix $W_1 = W - \mu_1^2\mathbf{u}_1\mathbf{u}_1'$ after subtraction of the first cluster is but the value of E_1 in equation (5): $S(W) = G_1(\mathbf{u}_1) + E_1$ where $E_1 = S(W_1)$. A similar decomposition $S(W_1) = G_2(\mathbf{u}_2) + E_2$ holds for the second cluster. After substituting this equation for $S(W_1)$ for E_1 in the former equation, that becomes $S(W) = G_1(\mathbf{u}_1) + G_2(\mathbf{u}_2) + E_2$ which proves the statement.

These properties substantiate the following criteria for halting the process of iterative extraction of fuzzy clusters used in FADDIS procedure:

1. The maximum value of ξ (3) for the spectral fuzzy cluster is negative.

2. The contribution of a single extracted cluster is too low, less than a pre-specified $\epsilon > 0$ value. For example, for a network like Karate club of about 30 members in section 3.1, a cluster should contribute at least as much as an average entity, so that $\epsilon = 1/30$ should be considered a fair choice in this problem.

3. The residual scatter E becomes smaller than a pre-specified $1-\tau$ value, say less than 5% of the original similarity data scatter, which means that the total cluster contribution has become greater than τ, that is 95% in the example.

4. A pre-specified number $K_{\max}$ of clusters is reached – in some real-world problems such a number can be set indeed.

2.4. Laplace transformation and its adaptation to the additive model

The spectral approach to clustering similarity data, along with the so-called Laplacian normalization, became popular after publication by Shi and Malik [34]. This paper proposed a very successful normalized cut criterion proven to be related to the problem of minimization of the Rayleigh quotient for the Laplace matrix rather than the similarity matrix itself.

Given a similarity matrix W, its Laplace matrices are defined as follows. First, an $N \times N$ diagonal matrix D is computed, with (t,t) entry equal to $d_t = \sum_{t' \in T} w_{tt'}$, the sum of t's row of W. Then combinatorial Laplacian and normalized Laplacian are defined with equations $L = D - W$ and $L_n = D^{-1/2} L D^{-1/2}$, respectively. Both matrices are semipositive definite and have zero as the minimum eigenvalue. The minimum non-zero eigenvalues and corresponding eigenvectors of the Laplacian matrices are utilized then as relaxations of combinatorial partition problems [34, 28, 39, 18]. Of these two Laplace matrices, the normalized Laplacian in general is considered superior [18].

Yet the Laplacian normalizations cannot be used in our approach as is, because FADDIS relies on maximum rather than minimum eigenvalue. To pass over this issue, the authors of [28] utilized a complementary matrix $M_n = D^{-1/2}WD^{-1/2}$ which relates to L_n by equation $L_n = I - M_n$ where I is the identity matrix. This means that M_n has the same eigenvectors as L_n, whereas the respective eigenvalues relate to each other as λ and $1 - \lambda$, so that matrix M_n can be used for our purposes as well.

Yet we prefer using the Laplace Pseudo INverse transformation, Lapin for short, defined as

$$L_n^+ = \tilde{Z}\tilde{\Lambda}^{-1}\tilde{Z}'$$

where $\tilde{\Lambda}$ and $\tilde{Z}$ are defined by the spectral decomposition $L_n = Z\Lambda Z'$ of matrix L_n in the following way. First, set T' of indices of elements corresponding to non-zero elements of Λ is determined, after which the matrices are taken as $\tilde{\Lambda} = \Lambda(T', T')$ and $\tilde{Z} = Z(:, T')$. The Lapin transformation leaves the eigenvectors of L_n unchanged while inverting the non-zero eigenvalues $\lambda \neq 0$ to those $1/\lambda$ of L_n^+. Then the maximum eigenvalue of L_n^+ is the inverse of the minimum non-zero eigenvalue λ_1 of L_n, corresponding to the same eigenvector. The inverse of a λ near 0 could make the value of $1/\lambda$ quite large and greatly separate it from the inverses of other near zero eigenvalues of L_n. Consider, for example, $\lambda_1 = 0.05$ and $\lambda_2 = 0.2$ so that their complements to unity are 0.95 and 0.8 while the inverses are 20 and 5 – the growth of the gap between the values, from 0.15 to 15, is impressive indeed. The latter gap suits the FADDIS' one-by-one approach much better.

This ability allows the Lapin transformation to manifest clusters according to human intuition, such as presented in Fig. 1 where two clusters, heap in the center and ring around it, cannot be separated by variance based algorithms such as K-Means or EM for mixtures of Gaussian distributions but are easily separated by spectral clustering [18]. This is caused by the fact that after the Lapin transformation the similarity structure becomes clear-cut with all the positive Lapin similarities within the two intuitive clusters and all the negative Lapin similarities between them. Yet there can be cases, as will be seen further, at which Lapin transformation does not work at all (see also [30]).

3. Experiments in Application and Comparison

In this section, we consider four types of similarity data that have some differences, not always clearly understood, that have been analyzed in the literature with different approaches. These data types are:

1. Ordinary graphs with a "flat" similarity structure; they have been intensely used in the problem of detection of community structure.

2. Small real-world dissimilarity data that have been subject of analysis in founding papers on relational fuzzy clustering.

3. Affinity data that are obtained by transforming coordinate based data with a, typically Gaussian, kernel. Because of the high sensitivity of the Gaussian kernel, these data manifest high versatility in similarities,

which make the affinity data a sound target for the spectral clustering approach.

4. Similarity data either derived from in-house surveys of research activities or obtained in psychological experiments.

We are going to use these types of data for both testing FADDIS and comparing it to other fuzzy clustering techniques.

3.1. Application to Finding Community Structure

The research in finding community structure in ordinary graphs has been revitalized recently by M. Newman and others, with the usage of the so-called modularity criterion and reformulating it within the spectral cluster analysis framework (see, for example, [27, 26, 36, 18]). The graph with a set of vertices T is represented by the similarity matrix $A = (a_{tt'})$ between graph vertices such that $a_{tt'} = 1$ if t and t' are connected by an edge, and $a_{tt'} = 0$, otherwise. Then matrix A is symmetrized by the transformation $(A + A')/2$ after which all diagonal elements are made zero, $a_{tt} = 0$ for all $t \in T$. We assume that the graph is connected; otherwise, its connected components are treated separately.

We first apply FADDIS algorithm, in both -m and -a versions, to Zachary karate club network data, which serves as a prime test bench for community finding algorithms. This ordinary graph consists of 34 vertices, corresponding to members of the club and 78 edges between them - the data and references can be found, for example, in [27, 39]. The members of the club are divided according to their loyalties toward the club's two prominent individuals: the administrator and instructor. Thus the network is claimed to consist of two communities, with 18 and 16 differently loyal members respectively.

Applied to this data, both versions of FADDIS lead to the same three fuzzy clusters to be taken into account. Indeed, the fourth cluster both times accounts for just 2.4% of the data scatter, which is less than the inverse of the number of entities $\tau = 1/34$ suggested above as a natural threshold value. Some characteristics of the found solution(s) are presented in Table 1.

All the membership values of the first cluster are positive - as mentioned above, this is just the first eigenvector; the positivity means that the network is well connected. The second and third FADDIS clusters match the claimed structure of the network: they have 16 and 18 positive components, respectively, corresponding to the two observed groupings.

Let us compare our results with those of a recent spectral fuzzy clustering method developed in [39]. The latter method finds three fuzzy clusters, two of them representing the groupings, though with a substantial overlap between them, and the third, smaller, cluster consisting of members 5,6,7,11,17 of just one of the groupings – see [39], p. 487. We think that this latter cluster may have come up from an eigenvector embracing the members with the largest numbers of connections in the network. It seems for certain that FADDIS outperforms the method of [39] on Zachary club data.

To test the performance of FADDIS algorithm in detecting community structure on a larger scale, we devised an experiment in randomly drawing a community network. This network comprises two communities, each consisting of a random number of members from 6 to 15; the connecting edges are drawn uniform randomly with probability p within each community and probability q between the communities. Although the uniform distributions do not necessarily reflect those in real world networks [27, 26], this seems an appropriate bench-mark for testing a general clustering algorithm such as FADDIS.

After a network is generated, a version of FADDIS is run; then the first membership vector is discarded, and the following two types of errors are recorded over the two entity sets corresponding to the positive membership values in the second and third membership vectors, after identifying that of the generated communities they correspond to:

- the confusion error, which is the number of entities wrongly assigned between the two clusters, related to the total number of entities generated;
- the omission error, which is the number of entities not assigned to clusters 2 and 3 at all, related to the total number of entities generated;

At these data sets, the results of FADDIS-a did not differ from those of FADDIS-m, indicating that the largest eigenvalue always leads to the best contributing clusters in this setting. Table 2 presents averages and standard deviations of each of the confusion and omission error values over a thousand data generation runs. Each cell in it corresponds to a pair (p, q), $p = 0.6, 0.7, 0.8, 0.9$ and $q = 0.1, 0.2, 0.3, 0.4$; each of the mean values is accompanied with the standard deviation after slash. Within every cell, the confusion error is on top with the omission error underneath.

As one can see, the errors are rather high at $p = 0.6$, reaching its minimum of total 27.4 % at $q = 0.2$. At each q, the errors decrease with the growth of internal connections p. One would expect the greatest error at the worst conditions, the smallest internal links at $p = 0.6$ and the largest external links at q=0.4, which is the case indeed. Yet a non-trivial feature of the error, that has been always observed over many series of 1000 runs of the data generation routine, is the lack of monotonicity of the errors with respect to q. The error appears to be always smaller at $q = 0.2$ than at $q = 0.1$. Moreover, with the growth of p the minimum error moves to even greater q values. For example, at $p = 0.9$, the error, totaling to 4.5%, is the smallest at $q = 0.3$.

Overall, the results show that the method is consistent, and in fact, efficient in discovering the two-community structure. Its performance when there are more communities in the graph remains to be tested.

3.2. Fuzzy clustering affinity data

The affinity data is a relational similarity data obtained from a feature based dataset using a semi-positive definite kernel, usually the Gaussian one. Specifically, given an $N \times V$ matrix $Y = (y_{tv})$, $t \in T$ and $v = 1, 2, ..., V$, non-diagonal elements of the similarity matrix W are defined by equation

$$w_{tt'} = exp(-\frac{\sum_{v=1}^{V}(y_{tv} - y_{t'v})^2}{2\sigma^2}),$$

with the diagonal elements made equal to zero, starting from founding papers [34, 28]. The value $ss = 2\sigma^2$ is a user-defined parameter, that is pre-specified to make the resulting similarities $w_{tt'}$ spread over interval [0,1].

To see how this approach works, we adapt an example from [25]: two 2D clusters are generated, one from a normal distribution corresponding to a small ball whereas the other from a uniformly distributed strip, which is much longer. When y-axis difference between the clusters is small, separating them is of an issue for spectral clustering algorithms [25]. By changing the distance between the clusters, one can test the consistency of a clustering method. Specifically, a hundred points are randomly generated by using Gaussian distribution N(1,0.5) over both axes to make cluster 1, and two hundred points are generated uniform randomly in a strip taking the fragment of x-axis from 0 to 50 while maintaining its width over y-axis equal to one.

This is illustrated in Fig. 2: the strip is put at $y = 3.5$ on part (a) and at $y = 0.5$ on part (c) of it. Then the data are standardized with the conventional z-scoring: by subtracting grand means from each of the coordinates and dividing the results by the feature's standard deviation.

Because of a combined use of Gaussian kernel and Lapin transformation, the final similarities are much diverse so that the very first fuzzy cluster here covers no general similarity between entities but should correspond to a meaningful grouping in the data.

Table 3 presents the averaged results of FADDIS algorithm over a hundred runs of data generation at three different ratios of the cluster sizes. No suffix -m or -a is attached to the name of the algorithm because both lead to the same results at these datasets. The number of points generated is always 300, but the cluster distribution differs: only 100 entities belong to the ball in the left column, 150 in the middle column, and 200 in the column on the right. The clusters are defuzzified at 0 level and compared with those generated. The same two types of errors that have been defined in Section 3.1 are registered here: the error of confusion that occurs if a point belonging to one cluster is identified by the algorithm as belonging to the other, and the error of omission occurring if a point belongs to neither of the two first clusters.

In general, the errors are consistent with the expectations. They are monotone decreasing with the growth of y and almost disappear at $y = 3.0$ or greater. However the character of the monotonicity is different at different cluster size distributions. At 100/200 ratio, the error is high at $y = 0.5$, but almost disappears starting from $y = 2.5$; moreover, the error of omission is rather low here. But at the opposite, 200/100, ratio, at small y values, the errors of omission are rather high while the confusion errors are relatively small, and the errors keep appearing even at higher degrees of separation of the clusters.

By changing the threshold of defuzzification the errors can be significantly decreased. Specifically, at the threshold of defuzzification 0.2, the confusion errors entirely disappear, while omission errors become lower, at small y values, and zero, at larger y values, as clearly seen in Table 4.

To compare our approach with other methods for fuzzy clustering of affinity data, we pick up an example from Brouwer [5]. This example concerns a two-dimensional data set, that we refer to as Bivariate4, comprising four clusters generated from bivariate spherical normal distributions with the same standard deviation 950 at centers (1000, 1000), (1000,4000), (4000, 1000), and (4000, 4000), respectively. The data forms a cloud presented in Fig. 3.

This data was analyzed in [5] by using the matrix D of Euclidean distances between the generated points. Five different fuzzy clustering methods have been compared, three of them relational, by Roubens [31], Windham [35] and NERFCM [12], and two of fuzzy c-means (FCM) with different preliminary pre-processing options of the similarity data into the entity-to-feature format, FastMap and SMACOF [5]. Of these five different fuzzy clustering methods, by far the best results have been obtained with method FCM applied to a five-feature set extracted from D with FastMap method [5]. The adjusted Rand index [15] of the correspondence between the generated clusters and those found with the FCM over FastMap method is equal on average, of 10 trials, 0.67 according to [5]; no standard deviation is reported.

To compare FADDIS with these, we apply Gaussian kernel to the data generated according to the Bivariate4 scheme and pre-processed by the z-score standardization so that similarities, after z-scoring, are defined as $a_{ij} = exp(-d^2(y_i, y_j)/0.5)$ where d is Euclidean distance. This matrix then is Lapin transformed to the matrix W to which FADDIS is applied.

To be able to perform the computation using a PC MatLab, we reduce the respective sizes of the clusters, 500, 1000, 2000, and 1500 totaling to 5000 entities altogether in [5], tenfold to 50, 100, 200 and 150 totaling to 500 entities. The issue is of doing a full spectral analysis of the square similarity matrices of the entity set sizes, which we fail to do with our PC MatLab versions at a 5000 strong dataset. We also experimented with fivefold and twofold size reductions. This should not much change the results because of the properties of smoothness of the spectral decompositions [14].

Indeed, one may look at a 5000 strong random sample as a combination of two 2500 strong random samples from the same population. Consider a randomly generated $N \times 2$ data matrix X of N bivariate rows, thus leading to Lapin transformed $N \times N$ similarity matrix W. If one doubles the data matrix by replicating X as $XX = [X; X]$, in MatLab notation, which is just a $2N \times 2$ data matrix consisting of a replica of X under X, then its Lapin transformed similarity matrix will be obviously equal to

$$WW = \begin{bmatrix} W & W \\ W & W \end{bmatrix}$$

whose eigenvectors are just doubles $(\mathbf{z}, \mathbf{z})$ of eigenvectors $\mathbf{z}$ of W. If the second part of the double data matrix XX slightly differs from X, due to sampling errors, then the corresponding parts of the doubled similarity matrix and eigenvectors also will slightly differ from those of WW and $(\mathbf{z}, \mathbf{z})$. Therefore, the property of stability of spectral clustering results [14] will hold for thus changed parts. This argument equally applies to the case when the original sample is supplemented by four or nine samples from the same population.

In our computations, five consecutive FADDIS clusters have been extracted for each of randomly generated ten Bivariate4 datasets. The very first cluster has been discarded as reflecting just the general connectivity information, and the remaining four were defuzzified into partitions so that every entity is assigned to its maximum membership class. The average values of the adjusted Rand index, along with the standard deviations at Bivariate4 dataset versions of 500, 1000, and 2500 generated bivariate points are presented in Table 5 for both cases of FADDIS, -a and -m. The results support our view that the data set size is not important if the proportions of the cluster structure do not change. According to the tables, both FADDIS-m and FADDIS-a methods outperform the results obtained by the five fuzzy clustering methods reported in [5].

One can see, also, that FADDIS-a provides a slightly better recovery of the Bivariate4 cluster structure than FADDIS-m. This is caused by the fact that FADDIS-m tends to halt, because of negative eigenvalues, at getting just three clusters rather than four: the two smallest clusters are merged in one by FADDIS-m. This tendency of FADDIS-m in discovering a coarse cluster structure before halting will be observed at other data types too.

A remark:

The entity-to-feature format of the Bivariate4 data suggests that relational cluster analysis is not necessarily the best way to analyze it; a genuine data clustering method such as K-Means may bring better results. Indeed, an application of the "intelligent" K-Means method from [21] to the original data size of $N = 5000$ has brought results with the average adjusted Rand index of 0.75 (the standard deviation 0.045), which is both higher and more consistent than the relational methods applied here and in [5].

3.3. Experiments with benchmark dissimilarity data

We take on three small real-world datasets that have been used extensively by researchers in fuzzy clustering: Windham's dissimilarity data [35], Davé-Sen Country dissimilarity data [7] and celebrated Iris data [9].

Analysis of Windham's dissimilarity data

A matrix of dissimilarity values d_{ij} between eleven objects is presented in Table 6 in which there are clear groupings of objects 1-5 and 7-11, whereas object 6 has close connections with objects 5 and 7. This dataset has been considered by Windham in [35] and then used as a clear-cut structure example in [11, 3, 12, 7, 5].

We transform this matrix into a similarity matrix W by applying Gaussian kernel transformation $w_{ij} = exp(-d_{ij}^2/100)$. Then we apply FADDIS-a to both W and its Lapin transformation L_n^+.

At W, there have been four clusters whose contributions have been greater than one thousandth (see Table 7). The first of them covers all entities to reflect the general connectivity. The other three match the structure of the data rather well so that cluster II corresponds to the grouping 7-11, cluster III, the grouping 1-5, and cluster IV, to the grouping 5-7, though this latter cluster contributes just 2% of the data scatter.

At the Lapin transformed matrix, only two clusters have been extracted before the value (3) became negative bringing the process to a halt. Positive components of these correspond to each of the two groupings, 1-5 and 7-11.

These results go along with the results of application of fuzzy clustering methods described in [31, 3, 12, 7, 5]. All of them require pre-specifying the number of clusters, at $K = 2$, and they all, except for the method by [31] that merged all in one cluster (see [5]), produce two clusters at which the five-element groupings have high membership values and object 6 is shared between them. Our result with the three meaningful clusters over matrix W in Table 7, in which object 6 does not belong to the groupings but forms one of its nearest neighbors, seems adequate too. Another feature of FADDIS clustering results is a rather sharp separation in the membership values: many are zeros, which is not the case with more conventional approaches which require special efforts for the defuzzification.

Analysis of Davé-Sen Country dissimilarity data

The Country dataset, taken from [7], Table II, presents dissimilarities between 12 countries obtained by averaging the results of a survey among students in political science (see Table 8).

Table 9 shows the clustering results found by applying three reference algorithms from the literature of relational fuzzy clustering: RFC[7], NERFCM[12], and FastMap[5] (see discussion in Section 4). Each of the algorithms finds three clusters, with 'Egypt' moving from cluster $c3$ to cluster $c2$ in the case of FastMap.

Table 10 shows the results of applying algorithms FADDIS-m and FADDIS-a to the data transformed to the similarity matrix W by subtracting all the dissimilarities from their maximum value. In this case, four clusters were found, with the fourth cluster separating 'Egypt' from the countries 'Brazil', 'India' and 'Zaire'. The lowest dissimilarity between 'Egypt' and the other countries, 4.67, may justify the separation.

The clustering structure resulting from the application of FADDIS to Lapin transformed matrix W is meaningless along with the very low contributions of each of the five clusters to the explanation of data scatter (see, for example, FADDIS-a results in Table 11).

However, with the Gaussian kernel pre-processing applied to the dissimilarity matrix itself, followed by Lapin transformation, FADDIS leads to meaningful results. FADDIS-a leads to a number of fragmented clusters. Yet FADDIS-m halts after just two clusters extracted because of the negative eigen-values. These two clusters are presented in Table 12. It looks like the final similarity matrix here does capture the data structure, however rough, to merge smaller patterns with the two grand patterns, of more or less a free economy, the USA et al. to more or less a rigid one, the USSR et al.

Analysis of Fisher-Anderson's Iris data

Although the celebrated Iris dataset [9] is not in the relational data format, we have been tempted to analyze it here to see how FADDIS would fare along the other applications of fuzzy clustering techniques.

In order to apply the FADDIS algorithm to 150×4 Iris entity-to-feature data set, it has been transformed into a 150×150 dissimilarity matrix by applying the Euclidean distance metric, and then transformed to a similarity matrix W by subtracting it from the maximum distance.

Tables 13-14 show the confusion matrices of FADDIS-m/FADDIS-a followed by the clusters' contribution to the data scatter at matrix W as is (Table 13) of after applying to W the Lapin transformation (left part of Table 14 for FADDIS-m and right part of Table 14 for FADDIS-a). With the original W, both FADDIS-m and FADDIS-a provide the same result (Table 13): they find 3 clusters exactly (which corresponds to the original number of classes) with 10 (6.6%) misclassified cases. This favorably compares with results of other fuzzy clustering algorithms: "The typical result of comparing hardened FCM or HCM partitions to the physically correct labels of Iris is 14–17 errors" (see [29], p. 528).

With the Lapin transformation applied, FADDIS-m finds 3 clusters also but with a higher precision error of 12% plus 6,67% of entities not clustered at all (omission error). FADDIS-a results are even worse: it finds 5 clusters with a precision error of 36,7% and omission error of 3,33%. Yet it should be made clear that the conclusion of a bad working of FADDIS here can be drawn with no knowledge of the pre-defined clusters at all: just note how low are the contributions of FADDIS found clusters meaning that they are but noise.

Once again the use of affinity data obtained with a Gaussian kernel here leads to even worse results, except for the FADDIS-m applied after Lapin transformation of the affinity data. It finds a coarse picture of just two clusters (once again the algorithm halts because of negative eigenvalues), one coinciding with the first Iris class and the other merging Iris second and third classes together. This result concurs with the claims [4] that the Iris data set may consist of just two, not three, clusters and, more importantly, feeds in at the capacities of FADDIS-m in discovering a coarse structure of the data.

3.4. Application to genuine similarity data

The potential single distinction of the genuine relational, or similarity, data from the affinity data is in handling the diagonal. It is made zero at the affinity data so that the entire focus is on the relations between the entities. Yet at the genuine similarity data the diagonal may bear an important distinction between the entities, which may affect the results. Indeed, zeroing the diagonal may change the results, because of changes in the summary values d_t in the denominator (while leaving L-values in the numerator unaffected).

Consider, for example, a typical genuine similarity dataset in Table 15, that presents the frequency of human confusion between different segmented numerals (such as those in Fig. **??**); the greater the confusion, the greater the similarity. The diagonal dominates the data and shows, for example, that humans tend to identify 1 and 0 better than 8 and 9.

Consider results of applying FADDIS at two options – one with the diagonal unchanged (u), the other with the diagonal zeroed (z). Defuzzified clusters at threshold 0.3 are shown for the first five FADDIS-a membership vectors in Table 16. One can see that two clusters at which (u) and (z) results agree are groupings of numerals {1,4,7} and {6, 8, 0}. These are exactly the clusters that have been found in [20] by a hierarchical aggregation algorithm maximizing the chi-squared coefficient of the aggregate table. The other clusters are significantly differ, except perhaps the clusters {3, 5, 9} at (z), and {3,5,6,9} at (u), both closely resembling cluster {3,5,9} from [20]. Yet some may say that these differences are not quite important because of low contributions to the data scatter.

Consider now the similarities between research topics derived from our survey of researchers in a University department or research center. These motivate the additive model in (1) as described in Section 2.1. The authors developed a publicly available tool ESSA for e-surveying of members of Computer Science Research organizations (see [22] and https://copsro.di.fct.unl.pt/). This tool is used to obtain a data table whose columns correspond to a set of V individuals or project teams in the organization ($v = 1, 2, \cdots, V$), and rows to (some of) research topics taken to be leaves of the ACM-CCS taxonomy ([1]). The (t, v) entry in the table is the score f_{tv} given by member v to the topic t, to express the share of their total research effort devoted to topic t; f_{tv} is greater than 0 but smaller than 1, and the column v sums up to unity - a property which suggests a specific normalization weight assigned to each of the columns, as explained below.

Also, the estimates f_{tv} can be derived from the body of documents posted on web, though this method can be applied only to organizations whose members do post English-written documents of their research on the Internet.

Then the similarity $a_{tt'}$ between topics t and t' can be defined as the inner product of vectors of scores $\mathbf{f}_t = (f_{tv})$ and $\mathbf{f}_{t'} = (f_{t'v})$, $v = 1, 2, \cdots, V$. Since all the individual scores sum up to unity, $\sum_{t \in T} f_{tv} = 1$ for each v, the scores of individuals bearing more topics tend to be smaller than those of individuals engaged in fewer numbers of topics. To make up for this, the inner product is moderated by a natural weighting factor, the ratio of the number of topics marked by individual v, n_v, and n_{max}, the maximum n_v over all $v = 1, 2, ..., V$,

$$a_{tt'} = \sum_{v=1}^{V} \frac{n_v}{n_{max}} f_{tv} f_{t'v}. \tag{10}$$

The similarity measure (10) has the following properties:

- The similarity matrix is definite semipositive.
- The similarity between two topics can be positive if and only if there is at least one researcher that is engaged in both.
- The greater the individual membership values, the greater the similarity.
- Given a pair of topics, the greater the set of researchers engaged in them, the greater the similarity.

Let us describe in brief FADDIS results obtained for topic-to-topic similarity matrices corresponding to two real-world Computer Science organizations, one a research center labeled here as A, the other a University department labeled here as B. The matrices can be found in [24].

First, some technical characteristics of the results:

1. Clustering results do not much depend on the diagonal entries, either left untouched or zeroed: there is no difference between the versions in membership values at the first three decimals at all.

2. FADDIS versions -m and -a lead to different results. Whereas FADDIS-a brings forth a number of clusters with declining, however sharply, contributions, FADDIS-m abruptly halts at both these data sets not because of low contributions but because the continuation of the process becomes impossible: the next spectral cluster membership vector gives a negative value to the weight (3), which goes along with the idea FADDIS-m revealing a coarse cluster structure in the dataset. We accept FADDIS-m results as those found without any use of knowledge of the domain.

Because one of the organizations, A, is a research center whereas the other, B, is a university department, one should expect that the total number of research topics in A is smaller than that in B, and, similarly, the number of clusters in A should be less than that in B. Indeed, research centers are usually created for a limited set of research goals, whereas university departments must cover a wide range of topics in teaching, which necessarily affects the research efforts. Both of these appear to be true: the number of ACM-CCS topics scored in A is 46 versus 54 in B. Also, the number of clusters in A is two, whereas in B it is four.

The clusters found at both research center A and university department B have a more or less clear meaning and are consistent with the informal assessment of the research conducted in each of the research organizations. Moreover, the sets of research topics that have been chosen by individual members at the ESSA survey follow the cluster structure rather closely, falling mostly within one of them. The FADDIS results for the two data (ESSA) surveys can be consulted in [24].

4. Related Work

This paper crosses several lines of research, of which the following will be mentioned here in sequence: relational fuzzy clustering, additive clustering, spectral clustering, and detection of community structure.

Relational fuzzy clustering

Relational fuzzy clustering is an activity of deriving fuzzy clusters from a relation, that is, a matrix of a dissimilarity index on T, $(d(t,t')), t, t' \in T$. This can be divided in two major streams: one utilizing the fuzzy logics operations such as minimum or plus but no operation of division, and the other involving all the numeric operations, including division. The former is rather thin and less developed (see, for instance, [37] and [10]). Our approach falls in the latter stream, which can be traced to papers [31] and [35] that utilized, essentially, the sum $\sum_{k=1}^{K}\sum_{t,t'} u_{tk}^2 u_{t'k}^2 d(t,t')$ as the criterion to minimize over unknown membership vectors $\mathbf{u}_k, k = 1, ..., K$. A similar criterion, proven to be equivalent to the criterion of popular fuzzy c-means method [4], was utilized by [11] to derive their RFCM algorithm, that works in two-phase iterations similar to c-means, including a relational analogue to the concept of cluster centroid. Specifying the so-called "fuzzifying" constant at the level of 2, the RFCM criterion is the sum over $k = 1, ..., K$ of items $\sum_{t,t'} u_{tk}^2 u_{t'k}^2 d(t,t')/\sum_t u_{tk}^2$ where $d(t,t')$ must be the squared Euclidean distance - otherwise, RFCM may lead to negative membership values. But even in this format, RFCM appears to be superior to Windham's assignment-prototype algorithm [3]. Later this restriction was relaxed, initially, by modifying RFCM into NERFCM algorithm to include the addition of a positive number to all off diagonal distances [12] and, more recently, by directly imposing the non-negativity constraint for membership values [7]. The latter paper also extended the concept of fuzzy clustering to include the so-called "noise" cluster to hold the bulk of membership values for entities that are far away of the K clusters being built. Paper [5] makes use of a two-stage procedure in which the first stage, such as FastMap mentioned above, supplies the entities with a few distance-approximating features so that the second stage utilizes a conventional algorithm such as fuzzy c-means for building fuzzy clusters in thus produced feature space.

FADDIS differs from these, first of all, in that it does not require the cluster membership values to form a fuzzy partition so that, for any entity t, the sum of its cluster membership values does not necessarily sum up to 1. Moreover, in our setting, the membership vector goes along with the cluster intensity so that the entries in the resulting index $\mu\mathbf{u}$, although non-negative, are not necessarily less than or equal to unity. This may seem to be a step too far, yet it is perfectly fitting the concept of fuzzy set introduced in [38]. An advantage of such an approach is that there is no need to introduce the concept of noise cluster [7] - the odd entities just get all membership values equal to zero. Another convenience is a natural definition of the validity of a cluster and set of clusters in our setting - by using the concept of contribution to the data scatter; the greater the better.

One more difference, the sequential character of FADDIS, makes it somewhat natural to address the problem of the number of clusters, which is impenetrable in the convenient settings. It is probably these features that make FADDIS that competitive in cluster recovery as shown above.

It is worth mentioning that some authors refer to fuzzy clusters whose membership values not necessarily sum up to 1 as possibilistic clusters (see, for example, [29, 8]). FADDIS clusters can be considered possibilistic too, albeit additional conditions that each cluster membership vector is normed and supplied with the cluster intensity value. In contrast to possibilistic clustering algorithms, though, FADDIS involves no additional parameters such as the reference distance in [8] to be adjusted.

The difference of FADDIS clustering criterion, apart from the fact that it applies to similarities rather than dissimilarities, should not be overstated though. There is a striking similarity between the RFCM criterion for a single cluster and our $g(\mathbf{u})$ criterion in (5). Indeed, denote $z_{tk} = u_{tk}^2$, then the former becomes $\sum_{t,t'} z_{tk} z_{t'k} d(t,t')/\sum_t z_{tk}$ which differs from the Rayleigh quotient $g(\mathbf{z})$ by the denominator only, it is the sum of $\mathbf{z}$'s rather than of their squares.

From the computational point of view, FADDIS is straightforwardly linked to a repetitive finding the matrix spectral decomposition, especially heavy in version (a) in which all eigenvectors are tested. That effectively limits the sizes of computationally feasible datasets to about three-four thousand entities, in a PC located environment with built-in spectral operations, like MatLab. In this aspect, the two-stage method by [5] is much better suitable to larger datasets.

Additive clustering

The additive clustering of similarity data has been introduced, in English, by Shepard and Arabie [33] in the setting involving cluster membership vectors $\mathbf{u}_k$ constrained to be just 1/0 binary vectors. Paper [19] referred to even earlier publications, in Russian, and proposed the iterative cluster extraction framework in that setting. However, the additive clustering model had not been extended to relational fuzzy clustering until a simplified version of model (1) was considered in [32]. This model involves a constant, not cluster-specific, intensity, cites no specific applications, and uses the Newton's descent method for fitting it. Newtons method involves many initialization parameters that need to be pre-specified, which is not what an innocent user would be willing to do. Thus, this paper appears to be the first treatise to properly extend the additive model to fuzzy clustering.

Spectral clustering

With respect to additive clustering model in (1), the spectral approach seems a most natural way to go because the equation is an extension of the spectral decomposition of matrix A onto fuzzy membership values. Yet, applied as is, by taking the first K eigenvectors and projecting them to cluster membership vectors, the approach, according to our experiments (not reported), fails to discover the clusters even in rather simple data structures. The spectral approach to clustering has gained popularity after Shi and Malik's change of the setting to, first, just one eigenvector, for a single cut, and, second, Laplacian data normalization [34].

The idea of computation of Laplace matrix as a normalization step appears tremendously effective, along with Gaussian kernel, at discovery of clusters of elongated geometry such as image segments or circular clusters [34, 28, 18]. The meaning of the pseudo-inverse Laplacian is currently under intense mathematical study in terms of conductivity of linear electric circuits (see, for example, [6, 30]) as well as the meaning of Gaussian kernel affinity transformation (see, for example, [25]).

It should be mentioned that, in our setting, the pseudo-inverse Laplace (Lapin) transformation is a device to fit into the nature of FADDIS clustering criterion, which has nothing to do with its electric network interpretation. In our experiments, Lapin transformation works, quite well, at the affinity and genuine similarity data; yet it fails on the innate dissimilarity data and ordinary graphs; the latter seems to have theoretical underpinnings [30].

Community detection

The single cut idea was extended, within the framework of community detection, to other normalizations by Newman and Girvan [27, 26]. Their idea of normalization comes from interpretation of the similarity data, even if an ordinary graph, as a manifestation of interactions between items $t, t' \in T$. To see the "real modularity structure" behind the interactions, the random interaction part, proportional to $d_t d_{t'}$ for each pair (t, t'), is to be subtracted first. After this, the spectral clustering approach should be applied [26]. This paper also can be put in that category, as well as any other method within the sequential extraction approach, since FADDIS makes just one cluster extracting step at a time. Moreover, in the context of community detection problem, FADDIS can be viewed as a further advancement into the approach of removal of random interactions from the similarities. Indeed, for a connected interaction graph, the first eigenvector is all positive, thus, equal to the first FADDIS cluster membership function. This first eigenvector $\mathbf{z}_1$, as is well known [2], is a further elaboration of the summary values d_t taken to represent the "random interaction force" in the modularity transformation – $\mathbf{z}_1$ takes into account not only direct interactions but indirect interactions as well. That means that subtraction from the data the similarities $\mu_1 z_{1t} z_{1t'}$ according to the eigenvector makes a better cleaning of the similarities from the background interactions. It is probably this feature that makes FADDIS competitive in the context of community structure analysis.

5. Conclusion

The major feature that puts FADDIS aside from the relational clustering approaches [3, 4, 5, 7, 11, 16, 31, 35, 37, 39] is that the cluster membership values directly contribute to the similarities, in an additive way, according to model (1). This comes with the price of imposing another novel feature, the cluster's intensity, to account for the similarity index scale. This somewhat blurs the meaning of a fuzzy membership value as proportion or probability which must never exceed the unity. Yet, along with the least squares criterion, this sharpens the found clusters and puts much more zeros in the membership vectors than in fuzzy clusters found by other methods.

Another feature of our approach is a natural setting for incomplete clustering: the method can and do get some of the entities not clustered at all - those with zero membership values to all the clusters. One more feature is that each cluster is accompanied with its weight, the contribution to the data scatter that basically accounts for the similarities between entities most belonging to the cluster: the larger the weight the greater within cluster similarities. On one hand, this characteristic can be used as another measure of clustering quality in addition to those introduced in [5]. On the other hand, the weights are utilized in FADDIS as natural stopping criteria, along with the most definite criterion of non-negativity of the maximum eigenvalue.

The presented material shows that FADDIS correctly clusters benchmark data, shows consistency over experimentally generated datasets, and is competitive over other approaches.

There are several issues that remain to be addressed:

1. Difference between FADDIS (m) and (a) versions.

   In our experimental settings, the case of negative maximum eigenvalue for stopping has occurred only at (m) version and never at (a) version. Moreover, even at (m) version, it works in most analyses of similarity between ACM-CCS items and very rarely at other cases. The propensity of FADDIS-m to capturing coarse structures of datasets after Lapin transformation with an abrupt halt because the residual matrix becomes negative definite should be further explored.

2. Data normalization.

   As we have seen, different data types may require different data normalization strategies. According to our experiments, community data should get just the symmetrization and diagonal removal, but not a Laplacian transformation. When we did apply the normalized Laplacian transformation to the random test data with two communities described in Section 3.1, the confusion error grew on average to about 30% and the omission error about 25%, which is much greater than errors at the unnormalized data. Explanation of this effect remains a task for the future (see [30]). However, the normalization with the pseudo-inverse Laplacian transformation works quite well at the affinity or similarity data. Specifics of these data types should be further explored.

3. Scalability.

FADDIS takes as many spectral decompositions as the number of clusters. In the version (m) only first eigenvector is needed, but version (a) uses all of them. This makes the scalability of the approach heavily linked to the scalability of the spectral decomposition, which leaves us with moderate, up to several thousand entities, data sizes. A breakthrough may come with the progress of approximation techniques; a step in this direction is the usage of maximum spanning trees for approximating Lapin normalization [13].

Acknowledgments

The materials for this paper have been developed within the framework of COPSRO project funded by the Portuguese Foundation for Science & Technology (grant PTDC/EIA/69988/2006 to SN). The support of the individual research project 09-01-0071 "Analysis of relations between spectral and approximation clustering" to BM by the "Science Foundation" Programme of the State University – Higher School of Economics, Moscow RF, is acknowledged. The authors express their gratitude to L. Moniz Pereira who initiated the work on the analysis of similarity between research topics in the context of mapping that to the ACM taxonomy, and Igor Guerreiro for developing the code for the ESSA e-survey tool. The help of members of Centre for Artificial Intelligence (CENTRIA) (Universidade Nova de Lisboa, Lisboa, Portugal) and the Department of Computer Science and Information Systems (Birkbeck University of London, London, UK) is appreciated. The authors are indebted to the reviewers for numerous comments that made us work harder to improve the presentation.

[1] The ACM Computing Classification System (1998), url= http://www.acm.org/class/1998/ccs98.html.

[2] C. Berge, The Theory of Graphs and Its Applications, Wiley (1964).

[3] J.C. Bezdek, R.J. Hathaway, M.P. Windham, Numerical comparisons of the RFCM and AP algorithms for clustering relational data, Pattern Recognition, 24 (1991) 783-791.

[4] J. Bezdek, J. Keller, R. Krishnapuram, T. Pal, Fuzzy Models and Algorithms for Pattern Recognition and Image Processing, Kluwer Academic Publishers, 1999.

[5] R.K. Brouwer, A method of relational fuzzy clustering based on producing feature vectors using FastMap, Information Sciences 179 (2009) 3561-3582.

[6] H. Chen, F. Zhang, Resistance distance and the normalized Laplacian spectrum, Discrete Applied Mathematics, 155(5) (2007) 654-661.

[7] R.N. Davé, S. Sen, Robust fuzzy clustering of relational data, IEEE Transactions on Fuzzy Systems, 10 (2002) 713-727.

[8] M. De Cáceres, F. Oliva, K. Font, On relational possibilistic clustering, Pattern Recognition, 39 (2006) 2010-2024.

[9] A. Frank, A. Asuncion, UCI Machine Learning Repository, Irvine, CA: University of California, School of Information and Computer Science (2010) [http://archive.ics.uci.edu/ml].

[10] Y.Y. Guh, M.S. Yang, R.W. Po, E.S. Lee, Establishing performance evaluation structures by fuzzy relation-based cluster analysis, Computers and Mathematics with Applications, 56 (2008) 572-582.

[11] R.J. Hathaway, J.W. Davenport, J.C. Bezdek, Relational duals of the c-means algorithms, Pattern Recognition, 22 (1989) 205-212.

[12] R.J. Hathaway, J.C. Bezdek, NERF c-means : Non-Euclidean relational fuzzy clustering, Pattern Recognition, 27 (1994) 429-437.

[13] M. Herbster, M. Pontil, S. R. Galeano, Fast prediction on a tree, NIPS (2008), 104, url: citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.143.6985.pdf (accessed 19 June 2010).

[14] L. Huang, D. Yan, M.I. Jordan, N. Taft, Spectral clustering with perturbed data, In: D. Koller, D. Schuurmans, Y. Bengio, L. Bottou (Eds.): Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems (Vancouver), MIT Press, 2009, pp. 705-712.

[15] L.J. Hubert, P. Arabie, Comparing partitions, Journal of Classification 2 (1985) 193-218.

[16] K. Inoue, K. Urahama, Sequential fuzzy cluster extraction by a graph spectral method, Pattern Recognition Letters 20 (1999) 699-705.

[17] D. Klein, M. Randic, Resistance distance, Journal of Mathematical Chemistry 12 (1993) 81-95.

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

[19] B. Mirkin, Additive clustering and qualitative factor analysis methods for similarity matrices, Journal of Classification 4 (1987) 7-31; Erratum 6 (1989) 271-272.

[20] B. Mirkin, Mathematical Classification and Clustering, Kluwer, 1996.

[21] B. Mirkin, Clustering for Data Mining: A Data Recovery Approach, Chapman and Hall / CRC Press, 2005.

[22] B. Mirkin, S. Nascimento, L. Moniz Pereira, Representing a Computer Science research organization over ACM Computing Classification System, In: P. Eklund, O. Haemmerlé (Eds.), Proc. 16th International Conference on Conceptual Structures, ISSN 1613-0073 (CEUR-WS.org), Toulouse, France, 2008, pp. 57-65.

[23] B. Mirkin, The iterative extraction approach to clustering, In: A.N. Gorban, B. Kegl, D.C. Wunsch, and A. Zinovyev (Eds.), Principal Manifolds for Data Visualization and Dimension Reduction, LNCSE 58, Springer-Verlag, 2008, pp. 151-177.

[24] B. Mirkin, S. Nascimento, Analysis of Community Structure, Affinity Data and Research Activities using Additive Fuzzy Spectral Clustering, Technical Report 6, School of Computer Science, Birkbeck University of London, 2009.

[25] B. Nadler, M. Galun, Fundamental limitations of spectral clustering, Neural Information Processing Systems Conference-06, Vol. 19, 2007.

[26] M.E.J. Newman, Finding community structure in networks using the eigenvectors of matrices, Phys. Rev. E 74 036104 (2006).

[27] M. Newman and M. Girvan, Finding and evaluating community structure in networks, Phys. Rev. E 69 026113 (2004).

[28] A. Ng, M. Jordan and Y. Weiss, On spectral clustering: analysis and an algorithm, In: T.G. Ditterich, S. Becker, Z. Ghahramani (Eds.), Advances in Neural Information Processing Systems, 14, MIT Press, Cambridge Ma., 2002, pp. 849-856.

[29] N.R. Pal, K. Pal, J.M. Keller and J.C. Bezdek, A possibilistic fuzzy c-means clustering algorithm, IEEE Transactions on Fuzzy Systems 13 (2005) 517-530..

[30] A Radl, U von Luxburg, M Hein, The resistance distance is meaningless for large random geometric graphs, Workshop on Analyzing Networks and Learning with Graphs (2009) http://snap.stanford.edu/nipsgraphs2009/papers/radl-paper.pdf.

[31] M. Roubens, Pattern classification problems and fuzzy sets, Fuzzy Sets and Systems 1 (1978) 239-253.

[32] M. Sato, Y. Sato, L.C. Jain, Fuzzy Clustering Models and Applications, Physica-Verlag, Heidelberg, 1997.

[33] R.N. Shepard and P. Arabie, Additive clustering: representation of similarities as combinations of overlapping properties, Psychological Review 86 (1979) 87-123.

[34] J. Shi and J. Malik, Normalized cuts and image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence (2000) 888-905.

[35] M.P. Windham, Numerical classification of proximity data with assignment measures, Journal of Classification, 2 (1985) 157-172.

[36] S. White and P. Smyth, A spectral clustering approach to finding communities in graphs, SIAM International Conference on Data Mining, 2005.

[37] M.S. Yang, H.M. Shih, Cluster analysis based on fuzzy relations, Fuzzy Sets and Systems 120 (2001) 197-212.

[38] L.A. Zadeh, Fuzzy sets, Information and Control, 8 (1965) 338-353.

[39] S. Zhang, R.-S. Wang and X.-S. Zhang, Identification of overlapping community structure in complex networks using fuzzy c-means clustering, Physica A 374 (2007) 483-490.

Vitae

Boris Mirkin's bio: Boris Mirkin holds a position of Professor of Computer

Science, in the School of Business, Economics, and Informatics, Birkbeck University of London, UK, and Professor in the Division of Applied Mathematics and Informatics, State University - Higher School of Economics, Moscow, RF. He is interested in mathematical models, computational algorithms and programs for clustering data in such applications as genomics, sociology, and text analysis. He published a hundred of refereed papers and several books on these; the latest monograph is “Clustering in data mining: A data recovery approach” (2005).

Susana Nascimento’s bio: Susana Nascimento received a Ph.D. degree in Computer Science from Universidade Nova de Lisboa in 2002. She is an Assistant Professor in the Department of Computer Science, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, Lisboa, Portugal. Her main research interests include fuzzy cluster analysis and classification with applications to data mining problems in general and remote sensing imagery.

Figures' Captions

1. Figure 1: Two intuitively obvious clusters: stars in the middle and dots in the ring.

2. Figure 2: Two versions of clusters of different shapes: that on (a) corresponds to distance 3.5 between them over y-axis, and on (c), distance 0.5 over y-axis. Figures (b) and (d) present the clusters after z-scoring of the data.

3. Figure 3: Digits: Styled digits formed by segments of the rectangle.

4. Figure 4: Bivariate4: the data of four bivariate clusters generated from Gaussian distributions according to [5].

Tables

Table 1: Characteristics of Karate club clusters found with FADDIS.

Cluster	Contribution, %	λ_1	Weight	Intensity
I	29.00	3.36	3.36	1.83
II	4.34	2.49	1.30	1.14
III	4.19	2.00	0.97	0.98

Table 2: The average confusion and omission errors of FADDIS clusters, along with their standard deviations, at different probabilities of the within community links (in rows) and between community links (in columns) resulting from a thousand data generation runs. The confusion error and its standard deviation are on top in every cell.

	0.4	0.3	0.2	0.1
0.6	0.329/0.110	0.237/0.136	0.160/0.140	0.166/0.151
	0.173/0.112	0.146/0.120	0.114/0.119	0.140/0.131
0.7	0.227/0.132	0.141/0.129	0.103/0.125	0.128/0.145
	0.123/0.118	0.088/0.110	0.082/0.117	0.121/0.138
0.8	0.110/0.111	0.072/0.100	0.061/0.096	0.098/0.135
	0.064/0.103	0.051/0.102	0.050/0.104	0.089/0.131
0.9	0.043/0.069	0.031/0.059	0.036/0.071	0.074/0.125
	0.019/0.067	0.014/0.058	0.025/0.082	0.062/0.131

Table 3: Average confusion and omission errors, along with their standard deviations (after slash), after a hundred of data generation runs at each of the different values of the y coordinate of the strip cluster, from $y = 0.5$ to $y = 3.5$ – the cases presented in Fig. 2. The columns refer to different ratios of the cluster cardinalities.

y	100/200	150/150	200/100
0.5	0.209/0.051	0.135/0.040	0.099/0.059
	0.064/0.116	0.069/0.111	0.249/0.172
1.0	0.151/0.027	0.080/0.024	0.040/0.040
	0.049/0.030	0.072/0.046	0.152/0.099
1.5	0.087/0.033	0.042/0.013	0.037/0.045
	0.034/0.018	0.021/0.015	0.052/0.081
2.0	0.016/0.010	0.010/0.006	0.020/0.041
	0.005/0.007	0.003/0.004	0.018/0.088
2.5	0.001/0.002	0.003/0.004	0.010/0.028
	0.000/0.001	0.000/0.001	0.007/0.066
3.0	0.000/0.001	0.001/0.002	0.002/0.003
	0.000/0.000	0.000/0.000	0.000/0.001
3.5	0.000/0.000	0.000/0.000	0.001/0.001
	0.000/0.000	0.000/0.000	0.000/0.000

Table 4: Average confusion and omission errors, along with their standard deviations (after slash), after the defuzzification at threshold 0.2. The rows correspond to different values of the y coordinate of the strip cluster, from $y = 0.5$ to $y = 3.5$ – the cases presented in Fig. 2. The columns refer to different ratios of the cluster cardinalities.

y value	100/200 th=0.2	150/150 th=0.2	200/100 th=0.2
0.5	0.000/0.000	0.000/0.000	0.000/0.000
	0.047/0.078	0.110/0.135	0.225/0.175
1.0	0.000/0.000	0.000/0.000	0.000/0.000
	0.058/0.035	0.081/0.039	0.154/0.082
1.5	0.000/0.000	0.000/0.000	0.000/0.000
	0.034/0.022	0.019/0.012	0.05/0.077
2.0	0.000/0.000	0.000/0.000	0.000/0.000
	0.005/0.007	0.003/0.004	0.006/0.006
2.5	0.000/0.000	0.000/0.000	0.000/0.000
	0.000/0.000	0.000/0.000	0.001/0.002
3.0	0.000/0.000	0.000/0.000	0.000/0.000
	0.000/0.000	0.000/0.000	0.000/0.001
3.5	0.000/0.000	0.000/0.000	0.000/0.000
	0.000/0.000	0.000/0.000	0.000/0.000

Table 5: Adjusted Rand Index values for FADDIS-m -a at different sizes of Bivariate4 dataset

	FADDIS-m clusters		FADDIS-a clusters	
Size	mean	std	mean	std
500	0.69	0.06	0.70	0.04
1000	0.71	0.06	0.70	0.03
2500	0.75	0.01	0.73	0.01

Table 6: A table of dissimilarity index between eleven objects (Table 1 from [5]).

Object	1	2	3	4	5	6	7	8	9	10	11
1	0	6	3	6	11	25	44	72	69	72	100
2	6	0	3	11	6	14	28	56	47	44	72
3	3	3	0	3	3	11	25	47	44	47	69
4	6	11	3	0	6	14	28	44	47	56	72
5	11	6	3	6	0	3	11	28	25	28	44
6	25	14	11	14	3	0	3	14	11	14	25
7	44	28	25	28	11	3	0	6	3	6	11
8	72	56	47	44	28	14	6	0	3	11	6
9	69	47	44	47	25	11	3	3	0	3	3
10	72	44	47	56	28	14	6	11	3	0	6
11	100	72	69	72	44	25	11	6	3	6	0

Table 7: FADDIS results for data objects of Table 6 for two data normalizations: Gaussian kernel transformation (W) and Lapin transformation (L_n^+).

Objects	Clusters at W				Clusters at L_n^+	
	I	II	III	IV	I	II
1	0.2460	0	0.4553	0	0.47	0
2	0.2662	0	0.4322	0	0.45	0
3	0.3490	0	0.5464	0	0.50	0
4	0.2662	0	0.4322	0	0.45	0
5	0.3567	0	0.3473	0.3947	0.36	0
6	0.3120	0	0	0.7606	0	0
7	0.3567	0.3924	0	0.5155	0	0.35
8	0.2662	0.4266	0	0	0	0.45
9	0.3490	0.5447	0	0	0	0.50
10	0.2662	0.4266	0	0	0	0.45
11	0.2460	0.4305	0	0	0	0.48
Contri	0.4875	0.1076	0.1094	0.0233	0.24	0.24
Intens	2.0594	1.4116	1.4176	0.9634	2.46	2.46

Table 8: Country Dissimilarity (CD) data from [7].

Country	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
C1-Belgium	0	5.58	7.00	7.08	4.83	2.17	6.42	3.42	2.50	6.08	5.25	4.75
C2-Brazil	5.58	0.00	6.50	7.00	5.08	5.75	5.00	5.50	4.92	6.67	6.83	3.0
C3-China	7.00	6.50	0.00	3.83	8.17	6.67	5.58	6.42	6.25	4.25	4.5	6.08
C4-Cuba	7.08	7.00	3.83	0.00	5.83	6.92	6.00	6.42	7.33	2.67	3.75	6.67
C5-Egypt	4.83	5.08	8.17	5.83	0.00	4.92	4.67	5.00	4.50	6.00	5.75	5.00
C6-France	2.17	5.75	6.67	6.92	4.92	0.00	6.42	3.92	2.25	6.17	5.42	5.58
C7-India	6.42	5.00	5.58	6.00	4.67	6.42	0.00	6.17	6.33	6.17	6.08	4.83
C8-Israel	3.42	5.50	6.42	6.42	5.00	3.92	6.17	0.00	2.75	6.92	5.83	6.17
C9-USA	2.50	4.92	6.25	7.33	4.50	2.25	6.33	2.75	0.00	6.17	6.67	5.67
C10-USSR	6.08	6.67	4.25	2.67	6.00	6.17	6.17	6.92	6.17	0.00	3.67	6.50
C11-Yugoslavia	5.25	6.83	4.5	3.75	5.75	5.42	6.08	5.83	6.67	3.67	0.00	6.92
C12-Zaire	4.75	3.00	6.08	6.67	5.00	5.58	4.83	6.17	5.67	6.50	6.92	0.00

Table 9: Country Dissimilarity (CD) data: results from application of the algorithms RFC[7], NERFCM[12] and FastMap[5].

	RFC	**NERFCM**	**Fast Map**
C1	$\{China, Cuba, USSR, Yugoslavia\}$	$\{China, Cuba, USSR, Yugoslavia\}$	$\{China, Cuba, USSR, Yugoslavia\}$
C2	$\{Belgium, France, Israel, USA\}$	$\{Belgium, France, Israel, USA\}$	$\{Belgium, Egypt, France, Israel, USA\}$
C3	$\{Brazil, Egypt, India, Zaire\}$	$\{Brazil, Egypt, India, Zaire\}$	$\{Brazil, India, Zaire\}$

Table 10: Country Dissimilarity (CD) data: results from application of the algorithms FADDIS-m and FADDIS-a without Lapin transformation.

	FADDIS-m/-a no Lapin	
	cluster	contrib
C1	$\{China, Cuba, USSR, Yugoslavia\}$	0.733
C2	$\{Belgium, France, Israel, USA\}$	0.069
C3	$\{Brazil, India, Zaire\}$	0.025
C4	$\{Egypt\}$	0.032

Table 11: Country Dissimilarity (CD) data: no good clusters with FADDIS-a applied after Lapin transformation.

	FADDIS-a after Lapin	
	cluster	contrib
C1	$\{China, Cuba, USSR\}$	0.090
C2	$\{Belgium, France, USA\}$	0.053
C3	$\{Egypt, India\}$	0.062
C4	$\{Brazil, Zaire\}$	0.055
C5	$\{Israel, Yugoslavia\}$	0.025

Table 12: Country Dissimilarity (CD) data Gauss-Lapin transformed: FADDIS-m results

	FADDIS-m after Gauss-Lapin	
	cluster	contrib
C1	$\{Brazil, Cuba, India, USSR, Yugoslavia, Zaire\}$	0.546
C2	$\{Belgium, China, Egypt, France, Israel, USA\}$	0.063

Table 13: FADDIS-m and FADDIS-a confusion matrix for the Iris data set, pre-processed with standard normalization; no Lapin transformation applied.

		Predicted Clusters		
		1	2	3
Original classes	1	50	0	0
	2	0	46	4
	3	0	6	44
Contrib		0.9071	0.0392	0.0128

Table 14: FADDIS-m confusion matrix for the Iris data set, pre-processed with standard normalization followed by Lapin transformation.

		FADDIS-m			FADDIS-a				
		1	2	3	1	2	3	4	5
Original classes	1	50	0	0	50	0	0	0	0
	2	0	23	17	4	19	9	5	12
	3	0	1	49	0	11	21	6	8
Contrib		0.0067	0.0055	0.0029	0.0067	0.0066	0.0066	0.0064	0.0064

Table 15: The Keren and Baggen (1981) data on confusion of the segmented numeral digits in an identification experiment [20].

	Response									
Stimulus	1	2	3	4	5	6	7	8	9	0
1	877	7	7	22	4	15	60	0	4	4
2	14	782	47	4	36	47	14	29	7	18
3	29	29	681	7	18	0	40	29	152	15
4	149	22	4	732	4	11	30	7	41	0
5	14	26	43	14	669	79	7	7	126	14
6	25	14	7	11	97	633	4	155	11	43
7	269	4	21	21	7	0	667	0	4	7
8	11	28	28	18	18	70	11	577	67	172
9	25	29	111	46	82	11	21	82	550	43
0	18	4	7	11	7	18	25	71	21	818

Table 16: FADDIS results at Digits data with the diagonal unchanged, u, or zeroed, z (both defuzzified at 0.3 threshold).

Numeral	Cluster1		Cluster2		Cluster3		Cluster4		Cluster5	
	u	z	u	z	u	z	u	z	u	z
1	+	+								
2					+			+		+
3						+	+			+
4	+	+							+	
5						+	+	+		
6			+	+			+	+		
7	+	+								
8			+	+						
9						+	+			
0			+	+						+
Contribution,%	20.3	23.2	8.3	9.5	8.3	5.2	2.3	1.0	3.5	0.6
Intensity	2.38	1.38	1.90	1.04	1.89	0.90	1.38	0.60	1.53	0.52

Figures

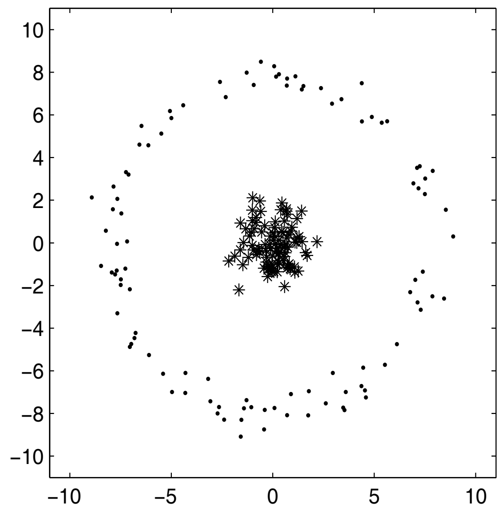

Figure 1: Two intuitively obvious clusters: stars in the middle and dots in the ring.

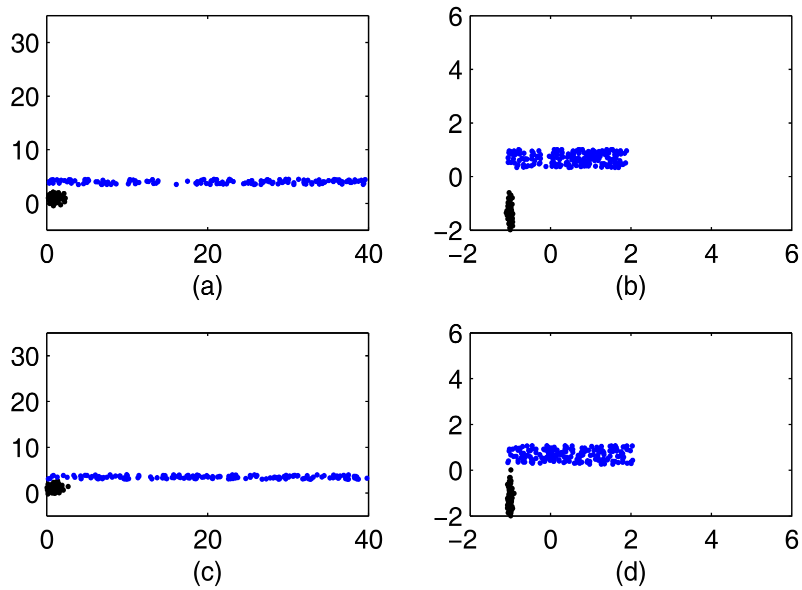

Figure 2: Two versions of clusters of different shapes: that on (a) corresponds to distance 3.5 between them over y-axis, and on (c), distance 0.5 over y-axis. Figures (b) and (d) present the clusters after z-scoring of the data.

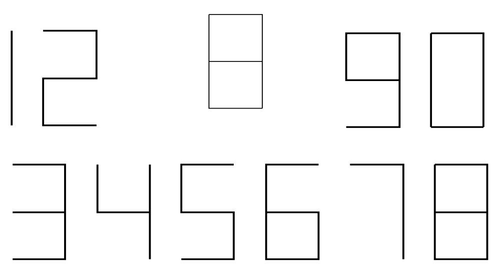

Figure 3: Digits: Styled digits formed by segments of the rectangle.

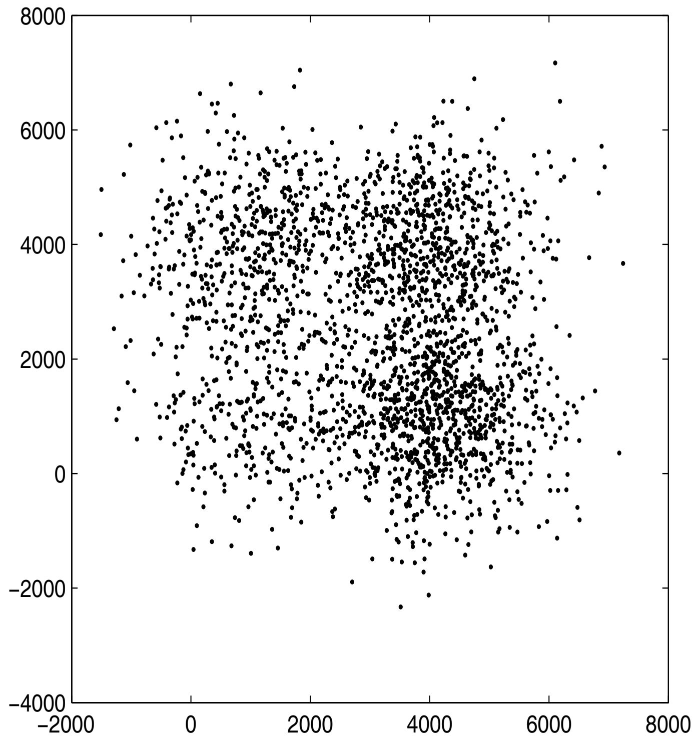

Figure 4: Bivariate4: the data of four bivariate clusters generated from Gaussian distributions according to [5].