Adv Data Anal Classif (2013) 7:83–108
DOI 10.1007/s11634-013-0125-7

REGULAR ARTICLE

Random walk distances in data clustering and applications

Sijia Liu · Anastasios Matzavinos · Sunder Sethuraman

Received: 28 September 2011 / Revised: 24 May 2012 / Accepted: 30 September 2012 /
Published online: 6 March 2013

Abstract In this paper, we develop a family of data clustering algorithms that combine the strengths of existing spectral approaches to clustering with various desirable properties of fuzzy methods. In particular, we show that the developed method "Fuzzy-RW," outperforms other frequently used algorithms in data sets with different geometries. As applications, we discuss data clustering of biological and face recognition benchmarks such as the IRIS and YALE face data sets.

Keywords Spectral clustering · Fuzzy clustering methods · Random walks · Graph Laplacian · Mahalanobis · Face identification

Mathematics Subject Classification (2000) 60J20 · 62H30

1 Introduction

Clustering data into groups of similarity is well recognized as an important step in many diverse applications (see, e.g., Snel et al. 2002; Liao et al. 2009; Bezdek et al. 1997; Chen and Zhang 2004; Shi and Malik 2000; Miyamoto et al. 2008). Well known clustering methods, dating to the 70's and 80's, include the K-means algorithm

S. Liu · A. Matzavinos (✉)
Department of Mathematics, Iowa State University, Ames, IA 50011, USA
e-mail: tasos@iastate.edu

S. Liu
e-mail: sijialiu@iastate.edu

S. Sethuraman
Department of Mathematics, University of Arizona,
617 N. Santa Rita Ave., Tucson, AZ 85721, USA
e-mail: sethuram@math.arizona.edu

(Macqueen 1967) and its generalization, the Fuzzy C-means (FCM) scheme (Bezdek et al. 1984), and hierarchical tree decompositions of various sorts (Gan et al. 2007). More recently, spectral techniques have been employed to much success (Belkin and Niyogi 2003; Coifman and Lafon 2006). However, with the inundation of many types of data sets into virtually every arena of science, it makes sense to introduce new clustering techniques which emphasize geometric aspects of the data, the lack of which has been somewhat of a drawback in most previous algorithms.[1]

In this article, we consider a slate of "random-walk" distances arising in the context of several weighted graphs formed from the data set, in a comprehensive generalized FCM framework, which allow to assign "fuzzy" variables to data points which respect in many ways their geometry. The method we present groups together data which are in a sense "well-connected", as in spectral clustering, but also assigns to them membership values as in usual FCM. In particular, we introduce novelties, such as motivated "penalty terms" and "locally adaptive" weights, along with the "random-walk" distances, to cluster the data in different ways by emphasizing various geometric aspects. Our approach might be used also in other settings, such as with respect to the K-means algorithm for instance, although here we have concentrated on modifying the fuzzy variable setting of FCM.

We remark, however, our technique is different than say clustering by spectral methods, and then applying the usual FCM, as is used in the literature. It is also different than the FLAME (Fu and Medico 2007) and DIFFUZZY (Cominetti et al. 2010) algorithms which compute 'core clusters' and try to assign data points to them. In terms of results, it also differs from the classical FCM. Also, it is different from the "hierarchical" random walk data clustering method in Franke and Geyer-Schulz (2009). (See Sect. 3.3.3 for further discussion.)

We demonstrate the effectiveness and robustness of our method, dubbed "Fuzzy-Random-Walk (Fuzzy-RW)", for a choice of parameters, on several standard synthetic benchmarks and other standard data sets such as the IRIS and the YALE face data sets (Georghiades et al. 2001). In particular, we show in Sect. 5 that our method outperforms the usual FCM using the standard Euclidean distance, spectral clustering, and the FLAME algorithm on the IRIS data set, and also FCM and the spectral method using eigenfaces (Muller et al. 2004) dimensional reduction on the YALE data set, which are main points of the paper. We also observe that Fuzzy-RW performs well on the YALE data set with Laplacianface (He et al. 2005), a different dimensional reduction procedure.

The particular random walk distance focused upon in the article, among others, is the "absorption" distance, which is new to the literature (see Sect. 3 for definitions). We remark, however, a few years ago a "commute-time" random walk distance was introduced and used in terms of clustering (Yen et al. 2005). In a sense, although our technique Fuzzy-RW is more general and works much differently than the approach in Yen et al. (2005), our method builds upon the work in Yen et al. (2005) in terms of using a random walk distance. Moreover, Fuzzy-RW seems impervious to random seed initializations in contrast to Yen et al. (2005). (See Sect. 3.3.3 for more discussion.)

[1] For further discussion of the emerging role of data geometry in the development of data clustering algorithms (see, e.g., Chen and Lerman 2009; Haralick and Harpaz 2007; Coifman and Lafon 2006).

The plan of the paper is the following. First, in Sect. 2, we recall the classical FCM algorithm, and discuss some of its merits and demerits with respect to some data sets including a standard "three circle" data set. Then, in Sect. 3, we first introduce certain weighted graphs and the "random-walk" distances, before detailing our Fuzzy-RW method. In Sect. 4, we discuss other weight systems which emphasize different geometric features, both selected by the user and also "locally adapted". In Sect. 5, we discuss the performance of our method on the IRIS and YALE face recognition data sets, and in Sect. 6 we summarize our work and discuss possible extensions.

2 Centroid-based clustering methods

We introduce here some of the basic notions underlying the classical k-means and fuzzy c-means methods. In what follows, we consider a set of data

$$\mathcal{D} = \{x_1, x_2, \ldots, x_n\} \subset \mathbb{R}^m.$$

embedded in a Euclidean space. The output of a data clustering algorithm is a partition:

$$\Pi = \{\pi_1, \pi_2, \ldots, \pi_k\}, \tag{1}$$

where $k \leq n$ and each π_i is a nonempty subset of $\mathcal{D}$. Π is a partition of $\mathcal{D}$ in the sense that

$$\bigcup_{i \leq k} \pi_i = \mathcal{D} \quad \text{and} \quad \pi_i \cap \pi_j = \varnothing \quad \text{for all } i \neq j. \tag{2}$$

In this context, the elements of Π are usually referred to as clusters. In practice, one is interested in partitions of $\mathcal{D}$ that satisfy specific requirements, usually expressed in terms of a distance function $d(\cdot, \cdot)$ that is defined on the background Euclidean space.

The classical k-means algorithm is based on reducing the notion of a cluster π_i to that of a cluster representative or centroid $c(\pi_i)$ according to the relation

$$c(\pi_i) = \arg\min_{y \in \mathbb{R}^m} \sum_{x \in \pi_i} d(x, y). \tag{3}$$

In its simplest form, k-means consists of initializing a random partition of $\mathcal{D}$ and subsequently updating iteratively the partition Π and the centroids $\{c(\pi_i)\}_{i \leq k}$ through the following two steps (see, e.g., Kogan 2007):

(a) Given $\{\pi_i\}_{i \leq k}$, update $\{c(\pi_i)\}_{i \leq k}$ according to (3).
(b) Given $\{c(\pi_i)\}_{i \leq k}$, update $\{\pi_i\}_{i \leq k}$ according to centroid proximity, i.e., for each $i \leq k$,

$$\pi_i = \{x \in \mathcal{D} \mid d(c_i, x) \leq d(c_j, x) \text{ for each } j \leq k\}$$

In applications, it is often desirable to relax condition (2) in order to accommodate for overlapping clusters (Fu and Medico 2007). Moreover, condition (2) can be too restrictive in the context of filtering data outliers that are not associated with any of the clusters present in the data set. These restrictions are overcome by fuzzy clustering approaches that allow the determination of outliers in the data and accommodate multiple membership of data to different clusters (Gan et al. 2007).

In order to introduce fuzzy clustering algorithms, we reformulate condition (2) as:

$$u_{ij} \in \{0, 1\}, \quad \sum_{\ell=1}^{k} u_{\ell j} = 1, \quad \text{and} \quad \sum_{\ell=1}^{n} u_{i\ell} > 0, \tag{4}$$

for all $i \leq k$ and $j \leq n$, where u_{ij} denotes the membership of datum x_j to cluster π_i (i.e., $u_{ij} = 1$ if $x_j \in \pi_i$, and $u_{ij} = 0$ if $x_j \notin \pi_i$). The matrix $(u_{ij})_{i \leq k, j \leq n}$ is usually referred to as the data membership matrix. In fuzzy clustering approaches, u_{ij} is allowed to range in the interval [0, 1] and condition (4) is replaced by:

$$u_{ij} \in [0, 1], \quad \sum_{\ell=1}^{k} u_{\ell j} = 1, \quad \text{and} \quad \sum_{\ell=1}^{n} u_{i\ell} > 0, \tag{5}$$

for all $i \leq k$ and $j \leq n$ (Bezdek et al. 1984; Miyamoto et al. 2008). In light of Eq. (5), the matrix $(u_{ij})_{i \leq k, j \leq n}$ is sometimes referred to as a fuzzy partition matrix of $\mathcal{D}$. For each $j \leq n$, $\{u_{ij}\}_{i \leq k}$ defines a probability distribution with u_{ij} denoting the probability of data point x_j being associated with cluster π_i. Hence, fuzzy clustering approaches are characterized by a shift in emphasis from defining clusters and assigning data points to them to that of a membership probability distribution.

The prototypical example of a fuzzy clustering algorithm is the fuzzy c-means method (FCM) developed by Bezdek et al. (1984). The FCM algorithm can be formulated as an optimization method for the objective function J_p, given by:

$$J_p(U, C) = \sum_{i=1}^{k} \sum_{j=1}^{n} u_{ij}^{p} \, \|x_j - c_i\|^2, \tag{6}$$

where $U = (u_{ij})_{i \leq k, j \leq n}$ is a fuzzy partition matrix, i.e. its entries satisfy condition (5), and $C = (c_i)_{i \leq k}$ is the matrix of cluster centroids $c_i \in \mathbb{R}^m$. The real number p is a "fuzzification" parameter weighting the contribution of the membership probabilities to J_p (Bezdek et al. 1984). In general, depending on the specific application and the nature of the data, a number of different choices can be made on the norm $\|\cdot\|$. The FCM approach consists of globally minimizing J_p for some $p > 1$ over the set of fuzzy partition matrices U and cluster centroids C. The minimization procedure that is usually employed in this context involves an alternating directions scheme (Gan et al. 2007), which is commonly referred to as the FCM algorithm. A listing of the FCM algorithm is given in Appendix.

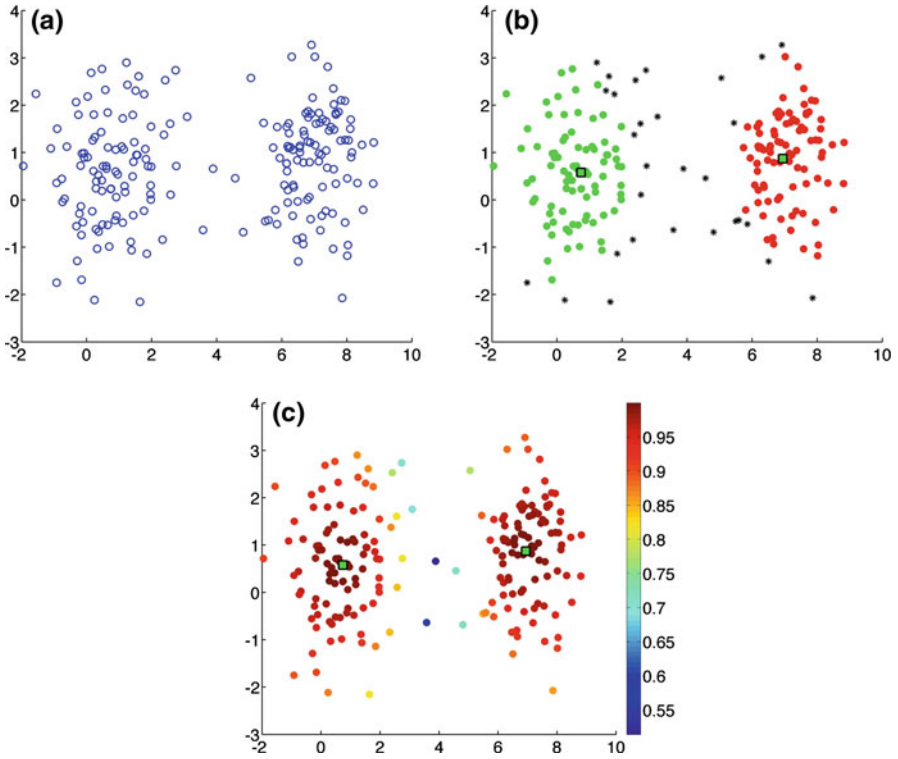

Fig. 1 **a** Figure showing a two-dimensional benchmark data set consisting of two linearly separable clusters. **b** Output of the FCM method (see, e.g., Eq. (6) in the text and Bezdek et al. 1984) applied to the data in **a**. The points *colored green* and *red* correspond to clusters for which the FCM-derived membership function attains values that are higher than threshold 0.9. The points in *black* are unassigned data points or outliers. **c** Figure showing the membership function computed by FCM. The *green squares* represent cluster centroids. (For interpretation of the references to *color in this figure legend*, the reader is referred to the web version of this article.)

This approach, albeit conceptually simple, works remarkably well in identifying clusters, the convex hulls of which do not intersect (Jain 2010; Meila 2006). A representative example is given in Fig. 1, where the data set under investigation is successfully clustered through the FCM algorithm using the Euclidean distance. However, for general data sets, J_p is not convex and, as we demonstrate below (see, e.g., Fig. 2), one can readily construct data sets $\mathcal{D}$ for which the standard FCM algorithm fails to detect the global minimum of J_p (Ng et al. 2002).

3 A new fuzzy clustering method

In the next two subsections, we discuss a weighted graph formed from the data set, and certain distances between data points. Using this framework, in the last subsection, we then develop our clustering method.

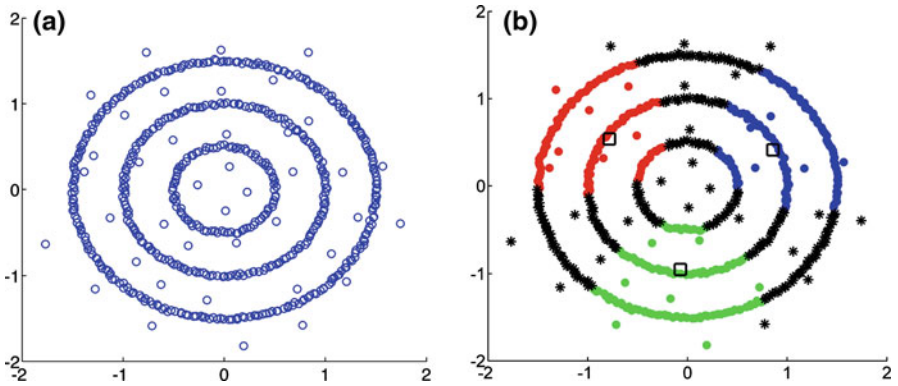

Fig. 2 **a** Dataset consisting of three core clusters and a uniform distribution of outliers. This geometric configuration leads to clusters which are not linearly separable, and it has been employed in the literature as an example of a data set for which the standard FCM method performs relatively poorly (Jain 2010; Ng et al. 2002). **b** Output of the FCM algorithm applied to the data in **a**. *The green squares* correspond to cluster centroids. The points *colored green*, *red*, and *blue* correspond to clusters for which the FCM-derived membership function attains values that are higher than threshold 0.8. The points in *black* are unassigned data with membership value <0.8. (For interpretation of the references to *color in this figure legend*, the reader is referred to the web version of this article.)

3.1 A random walk on the data

Given the data set $\mathcal{D}$ and the number k of clusters to be identified, we define a complete weighted graph $G = (V, E)$ with $V = \mathcal{D}$. The edges in E are weighted according to a weight matrix W with entries given by

$$W_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{\sigma}\right), \tag{7}$$

where $x_i, x_j \in \mathcal{D}$, and σ is a parameter which controls the spread in the weights. As usual, the choice of the norm $\|\cdot\|$ depends on the type of data considered. In what follows, $\|\cdot\|$ denotes the Euclidean norm.

This specific choice of the weight matrix is usually employed in standard, non-fuzzy spectral clustering approaches (see, e.g., Ng et al. 2002; Belkin and Niyogi 2003), where the optimal choice of parameter σ for a given data set is an active area of research (Coifman and Lafon 2006). Belkin and Niyogi (2003) in the spectral clustering context provide an extensive discussion of the advantages of these weights in the context of data clustering and dimensionality reduction. However, other choices have also been used in the literature (see, e.g., Coifman and Lafon 2006; Higham et al. 2007); in particular, later in this article, we introduce other weight matrices which will help detect some geometric features of the data set to be emphasized. Also, graphs G which are not complete have also been used in the literature (von Luxburg 2007; Belkin and Niyogi 2003), although in our treatment here, we will always assume G is the complete graph.

Given a weight matrix W, one can readily construct a random walk on G (Chung 1997) according to the transition matrix:

$$\mathcal{P} = D^{-1}W, \tag{8}$$

where D is the weighted degree matrix of G defined by

$$D_{ij} = \begin{cases} \sum_{\ell=1}^{n} W_{i\ell} & \text{if } i = j, \\ 0 & \text{if } i \neq j. \end{cases}$$

It is clear that $\mathcal{P}$ is a row stochastic matrix, i.e.,

$$0 \leq \mathcal{P}_{ij} \leq 1 \quad \text{and} \quad \sum_{\ell=1}^{n} \mathcal{P}_{i\ell} = 1,$$

for all $i,\ j \leq n$.

3.2 The absorption, commute-time and other distances

In the following, we define distance measures[2] on $\mathcal{D}$ that will eventually enable us to improve on the FCM machinery.

Consider a discrete time Markov chain $(X_n)_{n\geq 0}$ on the complete graph G with transition matrix $\mathcal{P}$ (8). Given $x_i, x_j \in \mathcal{D}$, we are interested in exploiting behaviors of the "random walk" X_n as it explores the geometry of the graph to construct a measure of the distance between x_i and x_j. Define the "hitting time" τ_j and "return time" τ_j^R of x_j as

$$\tau_j = \inf\{n \geq 0 \mid X_n = x_j\}$$
$$\tau_j^R = \inf\{n \geq 1 \mid X_n = x_j\}.$$

These two (random) times are the same if they start from $x_i \neq x_j$; however, starting from x_j, $\tau_j = 0$, but $\tau_j^R \geq 1$ is the time the random walk hits x_j after the first step.

3.2.1 Absorption distance

We first introduce the notion of the "absorption" distance between points x_i and x_j in the graph. This distance is built upon the idea that vertices x_i and x_j are distant if with large probability the random walk returns to x_i before "hitting" x_j. One is therefore interested in computing the probabilities $(P_i(\tau_j < \tau_i^R))_{i,j}$,

We now calculate the absorption probability $P_i(\tau_j < \tau_i^R)$, the chance the random walk, starting at x_i, "hits" x_j before returning to x_i. First note when $i = j$ that

[2] As it is usually the case in data clustering applications, the employed distance measures do not have to necessarily satisfy the properties of a metric (see, e.g., Chen and Zhang 2004).

$P_i(\tau_j < \tau_i^R) = 0$, and also when $i \neq j$ that $P_i(\{\tau_j < \tau_i^R\} \cap \{X_1 = x_i\}) = 0$ and $P_i(\{\tau_j < \tau_i^R\} \cap \{X_1 = x_j\}) = \mathcal{P}_{ij}$. Then, for $i \neq j$, from a first-step analysis (see, e.g., Brémaud 1999), write

$$P_i(\tau_j < \tau_i^R) = \mathcal{P}_{ij} + \sum_{k \neq i,j} \mathcal{P}_{ik} P_k(\tau_j < \tau_i), \tag{9}$$

as an average over possible first-step locations x_k. Next, for $k \neq i, j$, by a first-step analysis again, we observe

$$P_k(\tau_j < \tau_i) = \mathcal{P}_{kj} + \sum_{l \neq i,j} \mathcal{P}_{kl} P_l(\tau_j < \tau_i). \tag{10}$$

Define now the $(n-2)$-dimensional vector $V(i, j) = (P_k(\tau_j < \tau_i))_{k \neq i,j}$. Then, from (10)

$$V(i, j) = \mathcal{S}(i, j) + Q_{i,j} V(i, j)$$

where $Q_{i,j}$ is $(n-2) \times (n-2)$ submatrix of $\mathcal{P}$ with i, j rows and i, j columns removed, and $\mathcal{S}(i, j)$ is jth column of $\mathcal{P}$ with (i, j), (j, j) entries removed. One can readily solve

$$V(i, j) = (I - Q_{i,j})^{-1} \mathcal{S}(i, j).$$

Finally, noting (9), we have, for $i \neq j$, that

$$P_i(\tau_j < \tau_i^R) = \mathcal{P}_{ij} + \mathcal{R}(i, j) V(i, j),$$

where $\mathcal{R}(i, j)$ is ith row of $\mathcal{P}$ with (i, i), (i, j) entries removed.

In the remainder of the paper, we will use a "scaled" and symmetric form of the "absorption" expression. That is, we say the "absorption" distance between x_i and x_j in $\mathcal{D}$ is given by

$$T(x_i, x_j) = \left(1 - \frac{1}{2}(P_i(\tau_j < \tau_i^R) + P_j(\tau_i < \tau_j^R))\right)^{\gamma} \tag{11}$$

where the scaling parameter $\gamma \geq 0$ allows the user to control stratification of the distances between points in $\mathcal{D}$. One is also free to use another function of the absorption distance which takes advantage of its character.

3.2.2 Commute-time distance

We give now a version of the "commute-time" distance between points x_i and x_j based on the expected time $E_i[\tau_j]$ the random walk takes to move between them. Intuitively, points x_i and x_j separated by a large commute time may be understood as further apart then those bridged by a small commute time. In this way, the matrix of commute

times $(E_i[\tau_j])_{i,j}$ can serve as a distance measure between vertices $x_i, x_j \in \mathcal{D}$. Such a distance was first considered in Yen et al. (2005).

To compute these quantitites, first note, for any $x_j \in \mathcal{D}$, that $E_j[\tau_j] = 0$. Then, by a first-step analysis argument, $E_i[\tau_j]$ for $i \neq j$ is given by:

$$\begin{aligned} E_i[\tau_j] &= \sum_{\ell} E_i[\tau_j, X_1 = x_\ell] \\ &= E_i[\tau_j, X_1 = x_j] + \sum_{\ell \neq j} E_i[\tau_j, X_1 = x_\ell] \\ &= 1 \cdot \mathcal{P}_{ij} + \sum_{\ell \neq j} \mathcal{P}_{i\ell}(1 + E_\ell[\tau_j]). \end{aligned} \tag{12}$$

With respect to vector $A = (E_i[\tau_j])_{i \neq j}$, matrix $B = (\mathcal{P}_{i\ell})_{i \neq j, \ell \neq j}$ obtained by deleting the ith row and jth column from $\mathcal{P}$, and vector $R = (\mathcal{P}_{ij})_{i \neq j}$ which is the jth column of $\mathcal{P}$ with the (j, j) entry removed, Eq. (12) can be written as

$$A = R + B(\mathbf{1} + A),$$

where $\mathbf{1} = (1, 1, \ldots, 1)^T \in \mathbb{R}^{n-1}$. Hence, the vector of commute times $A = (E_i[\tau_j])_{i \neq j} \in \mathbb{R}^{n-1}$ is given by:

$$A = (I - B)^{-1}(R + B\mathbf{1}). \tag{13}$$

In what follows, we refer to the symmetric version of (13) as the "commute time" distance:

$$T_1(x_i, x_j) = \frac{1}{2}(E_i[\tau_j] + E_j[\tau_i]).$$

3.2.3 Other distances and discussion

One can of course build many other random-walk distances which might exploit differently the data geometry. For instance, let $g : \mathbb{R}^m \to \mathbb{R}_+$ be a function. Define

$$T_2(x_i, x_j) = \frac{1}{2}\left(E_i\left[\sum_{l=1}^{\tau_j} g(X_l)\right] + E_j\left[\sum_{l=1}^{\tau_i} g(X_l)\right]\right),$$

with the convention that an empty sum vanishes. When $g(x) \equiv 1$, T_2 reduces to the commute time distance above, $T_2 = T_1$. However, one may choose $g \not\equiv 1$ to emphasize parts of the background space $\mathbb{R}^m$ in assigning distance from x_i to x_j.

One might combine the "absorption" and "commute-time" distances to form

$$T_3(x_i, x_j) = \frac{1}{2}\left(E_i\left[\mathbb{1}(\tau_j < \tau_i^R)\sum_{l=1}^{\tau_j} g(X_l)\right] + E_j\left[\mathbb{1}(\tau_i < \tau_j^R)\sum_{l=1}^{\tau_i} g(X_l)\right]\right).$$

When $g(x) \equiv 1$, T_3 is the average of the expected commute times between points on those paths not returning to the starting point.

We now compare and contrast the "absorption" and "commute-time" distances. In the data sets we consider, both distances seem to separate points in the same manner, albeit with different parameter values. A main difference though from their definitions is that the commute-time distance gives more weight, in computing the expectation, to those random walk paths which take longer times, while the absorption distance does not emphasize such paths and only considers trajectories which do not return to the starting point.

We also mention, in this context, studies (von Luxburg et al. 2010; Alamgir and von Luxburg 2011) and references therein which point out that the tendency of the commute-time distance to weight long temporal paths may yield spurious distance values. However, the rankings given by the distance seem to be relevant, and von Luxburg et al. (2010) propose an 'amplified' commute-time distance which allows to discern the ranking structure more effectively.

In terms of implementation, although the commute time distance performs faster with computational complexity on order $O(n^3)$ for one step (to invert the matrix), the tendency in dense data sets is for the distance values to be quite high, and this forces sometimes extreme parameter values in the method to recognize ranking of the data. In this respect, modifications of the commute-time distance, with better properties, have been considered in von Luxburg et al. (2010) and references therein. On the other hand, although using the absorption distance in our technique has complexity $O(n^5)$ for one step [to invert $O(n^2)$ matrices], there is better spread in the ranked distances, with respect to the commute-time distance, which allows better parameter selection. It would be of interest to improve these cost estimates.

However, as the absorption time distance weights paths differently, it may have a more robust behavior than the commute-time distance. Of course, it would be of interest to make a more precise study of its properties. Since the absorption distance is new and unexplored, and to illustrate the possibilities in several data sets with different geometries, we have concentrated on this distance in the article.

3.3 Improving on FCM

3.3.1 Penalization and the absorption distance

We now introduce a family of fuzzy clustering algorithms that build on the FCM technology. The proposed methods will be collectively referred to as Fuzzy-RW, given their approach is to modify FCM by using random walk distances to measure data similarity.

As demonstrated in the computational experiment of Fig. 2, the performance of the usual FCM method is drastically reduced when applied to data sets characterized by a nested geometry (Ng et al. 2002; Cominetti et al. 2010). In this context, one approach to avoid such problems is to use a distance measure intrinsic to the geometry of the underlying data, instead of the Euclidean norm. Of course, to specify the 'intrinsic' distance requires prior knowledge of the data geometry.

However, in the absence of prior knowledge of the data set, a random walk on a graph formed from the data set can be employed to extract geometric information by randomly exploring the data landscape. This idea can be formalized in various ways, and the absorption distance, and other distances defined in the previous section serve this purpose. Hence, a natural modification of the FCM algorithm is to replace the objective function J_p of Eq. (6) with

$$J_p(U, C) = \sum_{i=1}^{k} \sum_{j=1}^{n} u_{ij}^p T^2(c_i, x_j), \tag{14}$$

where we recall T is the absorption distance defined in (11).

However, interestingly, the approach of minimizing $J_p(U, C)$ over the set of fuzzy partition matrices U and cluster centroids C does not lead to a successful identification of the core clusters in the example of Fig. 2. Indeed one can easily see that, in this example, the absorption distance is minimized over the circles. Hence, whenever two cluster centroids are located in the same circle, computations indicate the proposed method converges to a local, but not global minimum.

This latter phenomenon can be avoided by penalizing J_p with a term favoring large distances between cluster centroids. Hence, instead of minimizing J_p, the proposed algorithm minimizes F_p, where F_p is given by:

$$F_p(U, C) = J_p(U, C) + K \sum_{c_i \neq c_j} \frac{1}{T^2(c_i, c_j)}, \tag{15}$$

for some parameter K. Here, the centroids $\{c_i\}$ is a subset of size k of the vertices of the graph G composed of data points. The minimization then above of (15) consists of searching over these $\binom{n}{k}$ subsets, according to a convergence criterion as mentioned in Appendix.

The role of K is to balance effects of the penalty term with respect to the J_p term in (15). With respect to the absorption distance, in practice, it appears K can be chosen $K = O(n)$ with good success.

The rationale behind this approach is to ensure different centroids capture "appropriate" clusters. For instance, in Fig. 2b without the penalty term introduced, the procedure does not distinguish the three circles as distinct clusters, an outcome which seems desirable. However, if there are many "outliers", one of these might be selected as a centroid in a given run, and form a single-point cluster as in Fig. 3a. In the following subsection, we introduce a modification which discourages this phenomenon.

3.3.2 Using information on data density

To treat data sets which might have many outliers, and to avoid phenomena like the one shown in Fig. 3a, where the penalization term in (15) drives a cluster centroid to an outlier data point, the weight matrix W can be modified to include information on data density. In particular, we introduce another system of weights that lessens the similarity of relatively isolated data to the rest of the data set.

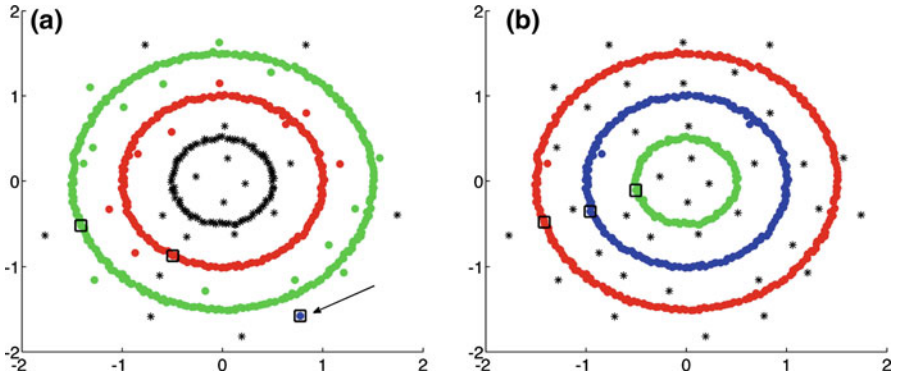

Fig. 3 **a** Output of minimizing the objective function (15) on the data of Fig. 2a. Here, one of the centroids is driven to an outlier datum. Parameters are $\sigma = 6 \times 10^{-4}, \gamma = 20, k = 3$ and $K = n$ the number of data points. **b** Output of Fuzzy-RW using approach described in Sect. 3.3.2, when applied to the same data set with parameters $\sigma = 6 \times 10^{-4}, r = 6 \times 10^{-2}, s = 2, \gamma = 20, k = 3$ and $K = n$ the number of data points. We have used threshold 0.85 in the figures. The *color code* is as in Fig. 2. *The black squares* indicate the locations of the cluster centroids. (For interpretation of the references to *color in this figure legend*, the reader is referred to the web version of this article.)

We start by introducing a hierarchy of neighborhoods on $\mathcal{D}$ as follows. For every $x_j \in \mathcal{D}$ and $r \in \mathbb{R}$, we define the neighborhood of x_j of radius r as

$$N_1(x_j, r) = \{x_i \in \mathcal{D} : \|x_i - x_j\| \leq r\}.$$

We will be referring to $N_1(x_j, r)$ as the first step neighborhood of x_j, and we define recursively the s-th step neighborhood of x_j as

$$N_s(x_j, r) = \{x_i \in \mathcal{D} : \|x_i - x_\ell\| \leq r \quad \text{and} \quad x_\ell \in N_{s-1}(x_j, r)\}.$$

Finally, we let $n_s(x_j, r)$ denote the (s, r)-density of $x_j \in \mathcal{D}$, defined by

$$n_s(x_j, r) = \text{number of elements of } N_s(x_j, r).$$

We remark, by construction, the neighborhood $N_s(s_j, r)$ groups together "fingers" or elongated aspects of the data set. Given a radius r and an integer step s, we modify the weight matrix used in computing the random-walk distance, using the same notation W, by introducing the density term $\kappa(i, j) = n_s(x_i, r)n_s(x_j, r)$ in (7) as follows:

$$W_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{\kappa(i, j)\sigma}\right). \tag{16}$$

Here, if one of x_i, x_j is somewhat isolated, then $\kappa(i, j)$ is smaller than if both data points belonged to dense neighborhoods, and accordingly the modified weight $W_{i,j}$ is biased to a lower value than would occur without the modification. In terms of the random-walk distance using the modified weight matrix, if one of x_i, x_j is isolated,

as it would be more difficult to travel between x_i and x_j, $T(x_i, x_j)$ would also be larger, and therefore an isolated centroid is less likely to be found in the minimization of (15).

We note that one could have also put another penalty term to attract centroids into dense regions instead of modifying the weights as we have done. We see now that this approach identifies in Fig. 3b, with many outliers, the three circles of data as the core clusters.

3.3.3 Further comparisons

As alluded to in the introduction, in Yen et al. (2005), a random walk distance based on the commute time between two points is used, specifically using the K-means objective function with distance $2T_1$ instead of T. There, after several (20) runs of K-means using the commute time distance, clustering which minimizes the objective function in these runs is chosen. However, there is no guarantee, even after several runs, when the initial centroids are chosen at random, that the "correct" clustering is achieved. On the other hand, in Fuzzy-RW, on any *single* run, no matter how the centroids are initialized, the penalty terms and the underlying graph weighting scheme, over subsequent iterations, drive the centroids away from each other so that optimal clustering is obtained.

Fuzzy-RW also differs from spectral clustering in the following way: In Fuzzy-RW, membership values are assigned to every data point so that low membership outlier points can be filtered out, which however in spectral clustering would be seen as core clusters themselves. Also, we point out Fuzzy-RW is not the same as running a spectral clustering method and then the classical FCM to assign membership values as it is performed in the literature (Tziakos et al. 2009). The main difference is that the penalizations and weighting scheme of the underlying graph introduced in Fuzzy-RW, in Sects. 3.3.2, 4.1 and 4.2, give more control on how a user might like to cluster data. In using FCM on data first spectrally clustered, one might run into similar problems with initializing centroids as discussed above near Eq. (15).

Moreover, Fuzzy-RW, which finds the "centroids" by optimizing an intrinsic objective function, differs from other fuzzy clustering approaches which first compute 'core clusters' and then try to assign data points to them. Two such examples are the FLAME (Fu and Medico 2007) and DIFFUZZY (Cominetti et al. 2010) algorithms. More specifically, FLAME identifies core clusters as relative dense parts of the data set and subsequently computes membership values through the general assumption that neighboring data points must have similar cluster memberships, whereas DIFFUZZY identifies core clusters by constructing a hierarchy of (Euclidean) neighborhood graphs and solving a discrete optimization problem and then assigns membership values to the data set by using a diffusion distance similar to the one employed by Coifman and Lafon (2006). Also, in this respect, Fuzzy-RW differs from other "hierarchical" random-walk methods in Franke and Geyer-Schulz (2009), where informally a pair of data points is assigned to a cluster at a certain level depending on when a type of random-walk, on an underlying graph formed from the data points, moves across the edge formed from the pair.

In the next section, we discuss more modifications of the weight structure to capture fine properties of the data set in some standard examples which might be useful in further distinguishing cluster geometry.

4 Elaborating on the notion of a cluster

Often one would like to emphasize various geometric features in clustering a data set. In the next two subsections, we show how to modify the weight structure so that directions both a priori specified or in some sense "locally adapted" are favored in computing the random-walk distance and in the clustering.

4.1 Intersecting linear subsets

As indicated in Sect. 3, the weight matrix W is the basic ingredient in the definition of the random walk (8) that we employ for determining distances and similarities between data in $\mathcal{D}$. In this section, we indicate another modification of W which allows Fuzzy-RW to cluster data with specific geometric requirements.

We start with the problem of identifying subsets of the data set on $\mathcal{D}$ that are embedded in lower dimensional linear manifolds, or affine spaces, of a specific orientation.[3] Variations of this problem are of interest in a number of applications, and different approaches have been suggested in the literature (Bock 1974, 1987; Späth 1985; Haralick and Harpaz 2005; Chen and Lerman 2009). For simplicity in the presentation, we discuss clustering data embedded in linear manifolds in $\mathbb{R}^2$. Nonetheless, the approach developed in this section can be readily generalized to higher-dimensional settings.

Let us suppose that we are interested in identifying clusters which are well approximated by a straight line in the direction of $v \in \mathbb{R}^2$. Hence, we are interested in defining a similarity matrix W that assigns high weight to edges (x_i, x_j) with the property that the vector joining x_i and x_j is approximately parallel to v. This can be readily achieved by replacing the Euclidean norm in (16) with the Mahalanobis distance (Abonyi and Feil 2007). In particular, consider the ellipsoid axes

$$V = \left[\tfrac{a}{a+1} v \quad \tfrac{1}{a+1} v^{\perp} \right],$$

where $a \in \mathbb{R}$ can be considered a "scale" which emphasizes the axis in direction of v and $v^{\perp}$ is orthogonal to v; specifically, if $v = (v_1, v_2)^T$, then $v^{\perp} = (-v_2, v_1)^T$. Let also $C = VV^T$ be the covariance matrix of V. Then, the Mahalanobis distance d_M in $\mathbb{R}^2$ is defined as

$$d_M^2(x, y) = (x - y)^T C^{-1} (x - y).$$

[3] Data clustering on linear manifolds, or affine spaces was first introduced by Bock (1974). Adopting the terminology of Haralick and Harpaz (2007), we say that L is a linear manifold in a vector space V if for some vector subspace S of V and some translation $t \in V$, $L = \{t + s \mid s \in S\}$.

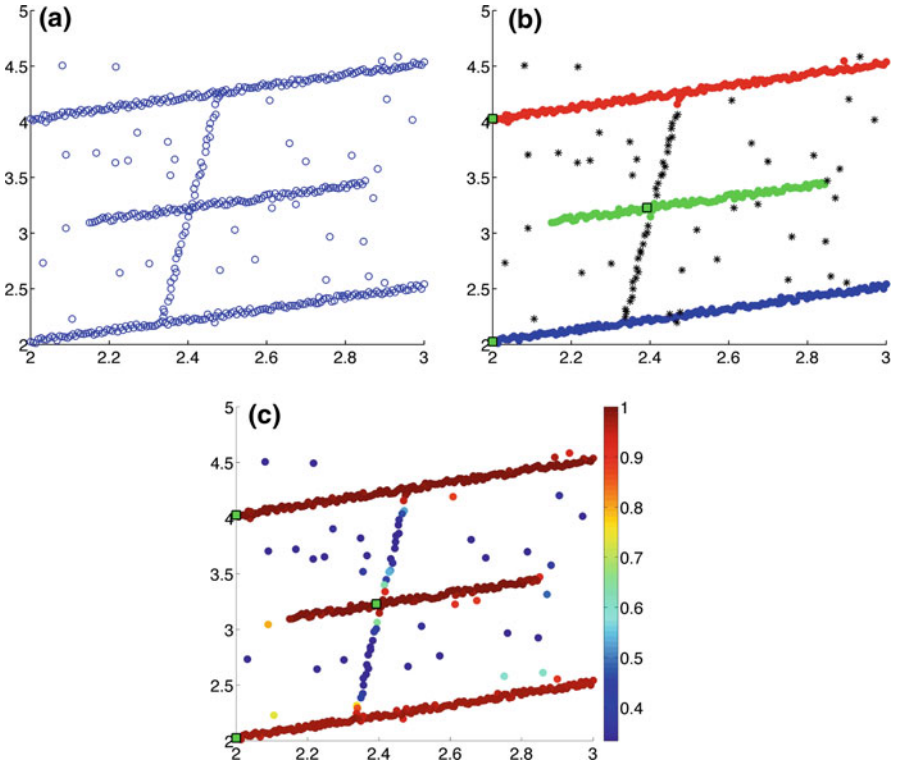

Fig. 4 **a** Benchmark data set that tests the ability of Fuzzy-RW, with threshold 0.8, to identify the points aligned with the three parallel lines in **a** as clusters (see Sect. 4.1). **b** Cluster assignments of data points (*color code* as in Fig. 2). **c** Membership values to corresponding clusters. The parameter values used are $\sigma = 4 \times 10^{-3}, r = 4 \times 10^{-2}, s = 4, a = 1.5, \gamma = 2, k = 3$ and $K = n$ where n is the number of data points. (For interpretation of the references to *color in this figure legend*, the reader is referred to the web version of this article.)

Hence, a possible choice for a weight matrix W that gives higher weight to pairs (x_i, x_j) nearly parallel to v is provided by

$$W_{ij} = \exp\left(-\frac{d_M^2(x_i, x_j)}{\kappa(i, j)\sigma}\right). \tag{17}$$

Here, W_{ij} depends on the direction v and the scale parameter a through the distance d_M.

In Fig. 4 the underlying data geometry is composed of intersecting lines, traditionally a quite difficult figure to cluster. However, with the above weight structure, favoring a particular direction, the random-walk distance is now less with respect to pairs of points parallel to v than otherwise.

In particular, we see that the method works fairly well to distinguish the parallel lines as separate clusters, ignoring the transversal.

In the next subsection, instead of prescribing a priori the directional bias, we introduce a weight structure which is "adaptive" in that it emphasizes directions according to local fits.

4.2 PCA and local linear approximations

For a specified radius r and step s, performing a local principal component analysis (PCA) on the s-th step neighborhood of each data point in $\mathcal{D}$ (see Sect. 3.3.2) provides the means to capture the local geometric structure of $\mathcal{D}$. As we demonstrate below, incorporating this information in the definition of the weight matrix W leads the Fuzzy-RW family of algorithms to behave more robustly on data sets that involve clusters of mixed dimensions.

A prototypical example of data of mixed dimensions is shown on Figs. 5 and 6, where each cluster involves a two-dimensional globular configuration of data along with some of the data embedded in a one-dimensional manifold. Data sets which involve geometric configurations of different intrinsic dimensions appear naturally in applications, and specialized methods have been developed recently for analyzing them (Arias-Castro et al. 2010).

Our approach however consists in finding a locally adapted coordinate system from which a weight matrix can be made. More specifically, for a data point $x_j \in \mathcal{D}$, consider its s-step neighborhood $N_s(x_j, r)$ as defined in Sect. 3.3.2. By performing PCA on the centered neighborhood, one computes m principal components $\{v_i\}_{i \le m}$ and corresponding eigenvalues of the covariance matrix $\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_m \ge 0$.

These principal components, which depend on x_j, locally approximate $\mathcal{D}$ in terms of an affine space $\mathcal{A}(x_j)$ with the property that the variance of the local projection of $\mathcal{D}$ onto $\mathcal{A}(x_j)$ is maximized. Hence, $\{v_i\}_{i \le m}$ can be thought of as a set of orthogonal axes from the point of view of data point x_j

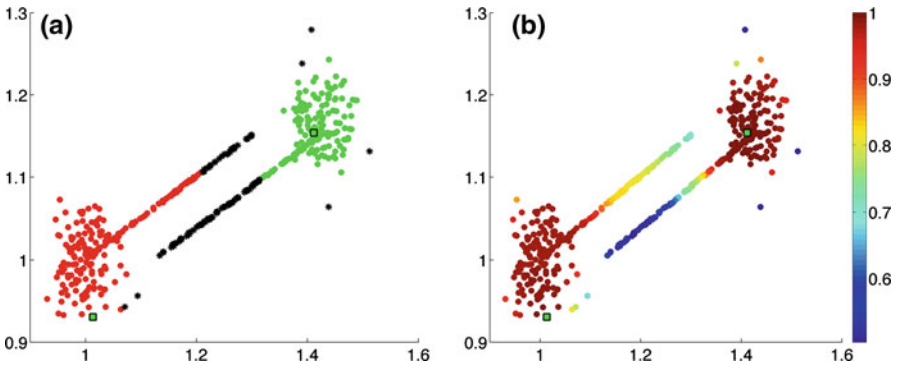

Fig. 5 Clustering of a data set of 'mixed dimensions' using Fuzzy-RW, with threshold 0.8, along with a "Gaussian-Euclidean" weight kernel (7). *Left* and *right* figures are the clustering output and membership values with respect to color codes as in Fig. 2. Parameter values are set as follows: $\sigma = 5 \times 10^{-4}$, $\gamma = 5, k = 2$ and $K = n/2$ where n is the number of data points. (For interpretation of the references to *color in this figure legend*, the reader is referred to the web version of this article.)

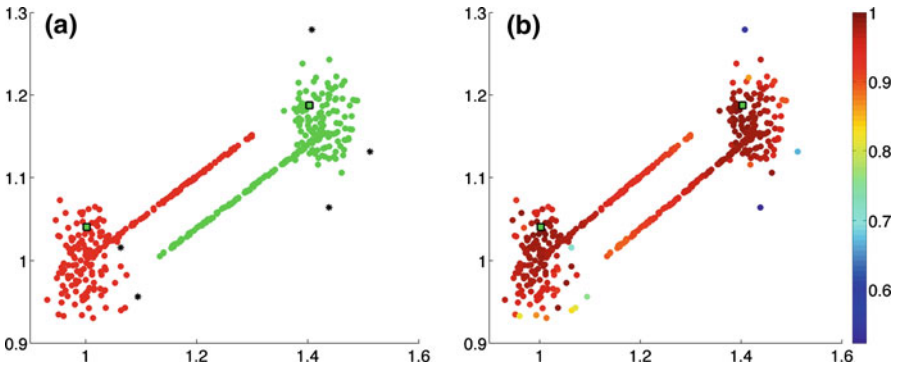

Fig. 6 Clustering of the data set in Fig. 5 with Fuzzy-RW combined with a locally adaptive weighting scheme, as described in Sect. 4.2. *Left* and *right* figures are the clustering output and membership values with respect to color codes as in Fig. 2. Parameter values are set as follows: $\sigma = 5 \times 10^{-4}$, $r = 0.017$, $s = 2$, $c = 6$, $\gamma = 5$, $k = 2$ and $K = n$. (For interpretation of the references to *color in this figure legend*, the reader is referred to the web version of this article.)

As in the previous subsection, we now choose scales $\mathbf{a} = \langle a_1, \ldots, a_m \rangle$, where $\sum_{i=1}^{m} a_i = 1$, to emphasize the axes, and form a Mahalanobis distance between data points x and y,

$$d_M^2(x, y) = (x - y)^T C^{-1}(x - y) \tag{18}$$

where $C = VV^T$ and $V = [a_1 v_1 \ldots a_m v_m]$. We note here the distance constructed is *not* symmetric in its arguments as $C = C(x)$ depends on the point x. Now, using weight formulation (17) with respect to distance (18), we construct a 'locally adapted' weight matrix, which we note again is *not* symmetric. In effect, W assigns directed weights across the various edges. With this constructed weight matrix, one forms the random walk distance (11) which is forced to be symmetric.

We remark, rather than choosing a priori given scales as in Sect. 4.1, the scales could also be taken as functions of the eigenvalues themselves, and in this way be themselves "locally adapted." For instance, when $m = 2$, one might use $a_i = a_i(\lambda_1, \lambda_2)$ for $i = 1, 2$ where $a_1 = \frac{c}{c+1} \mathbb{1}\left(\frac{\lambda_1}{\lambda_2} > c\right) + \frac{\lambda_1}{\lambda_1+\lambda_2} \mathbb{1}\left(\frac{\lambda_1}{\lambda_2} \leq c\right)$ and $a_2 = 1 - a_1$, which allows the scale to be given by truncated proportional eigenvalues in terms of a parameter c.

In Figs. 5 and 6, $m = 2$ and the scales chosen are such that $c = 6$.

5 Applications

We now apply our method to two standard data sets. The first benchmark is a biological data set, the IRIS data set, and the second is the YALE face recognition data set.

5.1 The Iris benchmark data set

In this subsection we evaluate the performance of the Fuzzy-RW method using regularized objective function (15) and weight matrix (7). We demonstrate that this

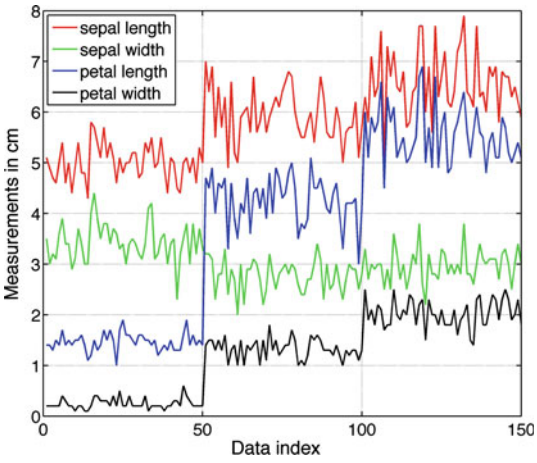

Fig. 7 Figure showing the Iris benchmark data set

Table 1 Comparison of numbers of true positives (TP) and false positives (FP) for different clustering methods

	FCM		Spectral		FLAME		Fuzzy-RW	
Index	TP	FP	TP	FP	TP	FP	TP	FP
1	50	0	50	0	50	0	50	0
2	47	13	50	15	50	11	47	2
3	37	3	35	0	37	0	48	3

implementation of Fuzzy-RW outperforms two well-known fuzzy clustering algorithms when applied to the Iris data set, shown in Fig. 7. As is well known, the Iris data set is a benchmark commonly employed in pattern recognition analysis (Hathaway and Bezdek 2001). It contains three clusters (types of Iris plants: *Iris setosa*, *Iris versicolour* and *Iris virginica*) of 50 data points each in 4 dimensions (features): sepal length, sepal width, petal length and petal width.

The results of applying FCM Bezdek et al. (1984), spectral clustering (Coifman and Lafon 2006) and the bioinformatics-oriented FLAME method (Fu and Medico 2007) to identifying the three clusters embedded in the Iris data set are shown on Table 1. Clearly, in the context of this benchmark, Fuzzy-RW, with parameters $\sigma = 0.1$, $\gamma = 1, k = 3, K = n = 150$ and points assigned to clusters by maximal membership, outperforms the other three approaches.

5.2 Data clustering in face recognition

Data clustering methods have traditionally been applied to a variety of image analysis tasks. Examples include image segmentation (Chen and Zhang 2004; Bezdek et al. 1997; Tziakos et al. 2009), image registration (Tsao and Lauterbur 1998) and shape recognition (Desolneux et al. 2008; Cao et al. 2007, 2008), among others. Chen and Zhang (2004) discuss the importance of developing fuzzy clustering methods which

can handle images given in terms of irregular, 'non-Euclidean' structures in the corresponding feature space.

In this section, we evaluate the performance of the Fuzzy-RW, using weight matrix (7), in the context of the face recognition problem (Belhumeur et al. 1997; Georghiades et al. 2001; Lee et al. 2005), that is the task of matching a given face to a database of faces (Kimmel and Sapiro 2003). In particular, given a set of face images labeled with the person's identity (the training set) and an unlabeled set of face images from the same group of people, we are interested in identifying each person in the unlabeled set (Belhumeur et al. 1997). A slight generalization of this, which we will focus on here, is to allow the second set of face images to contain possibly images of persons not included in the training set.

There are a number of approaches to the face recognition problem with perhaps the most straightforward one being that of applying directly a nearest neighbor classifier (Brunelli and Poggio 1993; Georghiades et al. 2001). However, this approach has been criticized on the basis of its computational complexity and its performance in the presence of extreme light variations in the images to be analyzed (Georghiades et al. 2001). More recent approaches rely on combining dimensionality reduction algorithms, such as principal component analysis (PCA) or locality preserving projection (LPP) (He et al. 2005), with a nearest neighbor classifier or a more general clustering algorithm (Lee et al. 2005; Shental et al. 2009).

Below we briefly recall, for the convenience of the reader, the eigenface technology which determines a basis of 'eigenfaces' in which the whole data set of faces can be represented. Laplacian faces uses a similar method to determine a basis of 'Laplacianfaces' (see He et al. 2005 for more details). In a nutshell, eigenfaces determines a low dimensional representation of the data set by computing principal components with respect to a training set which maximize the variance of its projection. Laplacianfaces proceeds along similar lines but uses generalized eigenvectors with respect to a constructed Laplacian matrix which strongly weights nearest-neighbor edges, and so preserves more of the geometry of the data set. Then, we show, with respect to the YALE face recognition data set, that Fuzzy-RW performs better than other clustering methods with respect to eigenfaces dimensional reduction. We also make a comparison between clustering using Fuzzy-RW when the reduction is done on the one hand with eigenfaces and on the other hand with Laplacianfaces.

5.2.1 The eigenface technology

Eigenfaces facilitate the low-dimensional representation of face images. The basic idea is that given a training set $\mathcal{T}$ of face images, principal component analysis is used to compute the principal directions, or eigenfaces, of $\mathcal{T}$. Each image then can be approximated by a linear combination of a few eigenfaces (Kimmel and Sapiro 2003; Muller et al. 2004).

In particular, consider a set of grayscale face images $\mathcal{I} = \{x_i\}_{i \leq n} \in \mathbb{R}^{p \times q}$, where it is assumed that every x_i is pre-processed by applying some image registration algorithm, and hence each face is aligned within the image. Now, let $\mathcal{T} \subset \mathcal{I}$ be a training set (Muller et al. 2004) for our classification scheme. Then, the covariance matrix of $\mathcal{T}$ is given by

$$C = \sum_{x_i \in \mathcal{T}} (x_i - \mu_{\mathcal{T}})(x_i - \mu_{\mathcal{T}})^T,$$

where $\mu_{\mathcal{T}} \in \mathbb{R}^{p \times q}$ is the mean of all images in the training set.

We are interested in identifying a low-dimensional space S for which the variance of the projection of $\mathcal{T}$ into S is maximized. This is readily done by identifying an $\ell \times pq$ orthonormal matrix Q that maps $\mathbb{R}^{p \times q}$ into a space of dimension $\ell < pq$ and such that it maximizes the determinant $\det(QCQ^T)$. It can be shown that QCQ^T is the covariance matrix of the image of $\mathcal{T}$ under Q (Belhumeur et al. 1997; Muller et al. 2004). The rows of Q are the eigenvectors of the covariance matrix C that correspond to the ℓ largest eigenvalues, and in what follows they will be referred to as the eigenfaces of $\mathcal{T}$.

5.2.2 The Laplacianfaces approach

As with eigenfaces, given a set of pre-processed face images $\mathcal{I} = \{x_i\}_{i \leq n} \subset \mathbb{R}^{p \times q}$, we work with a traning set $\mathcal{T} \subset \mathcal{I}$ to find first a low-dimensional basis. Using principal component analysis, as in the eigenface construction, the images in $x_i \in \mathcal{T}$ are transformed $x_i \to y_i$. Then, a similarity matrix is defined for nodes $x_i, x_j \in \mathcal{T}$:

$$S_{ij} = \exp\left(-\frac{d(y_i, y_j)}{t}\right)$$

where $d(y_i, y_j)$ is the Euclidean distance when y_i and y_j are both within a certain number of Euclidean nearest-neighbors of each other (in our experiment, within 9 neighbors), and $d(y_i, y_j) = 0$ otherwise. Here, t is a parameter to be chosen (and in our later experiment $t = 77$). Now, we choose a set of non-trivial eigenvectors (say, the first 20), which we call the Laplacianfaces, of the following generalized problem:

$$MLM^T\omega = \lambda MDM^T\omega$$

where the ith row of M is y_i, and the 'Laplacian' matrix $L = D - S$ and D is the diagonal matrix with $D_{ii} = \sum_j S_{ij}$.

If S is a constant matrix of 1's, then MLM^T is a data covariance matrix analogous to that in the eigenface construction. The role of S is to weight nearest-neighbor data points and so preserve in the eigenvector calculation some of the geometry of the data set. At this point, all images in $\mathcal{I}$ are centered with respect to the mean image of the training set. These are then projected on the Laplacianfaces computed to form a transformed data set $\{z_i\}_{i \leq n}$ which are analyzed by the Fuzzy-RW method.

5.2.3 Fuzzy-RW in the context of face recognition

In this subsection, we discuss results with Fuzzy-RW using weight matrix (7), in the context of recognizing faces from the YALE data base (Georghiades et al. 2001).

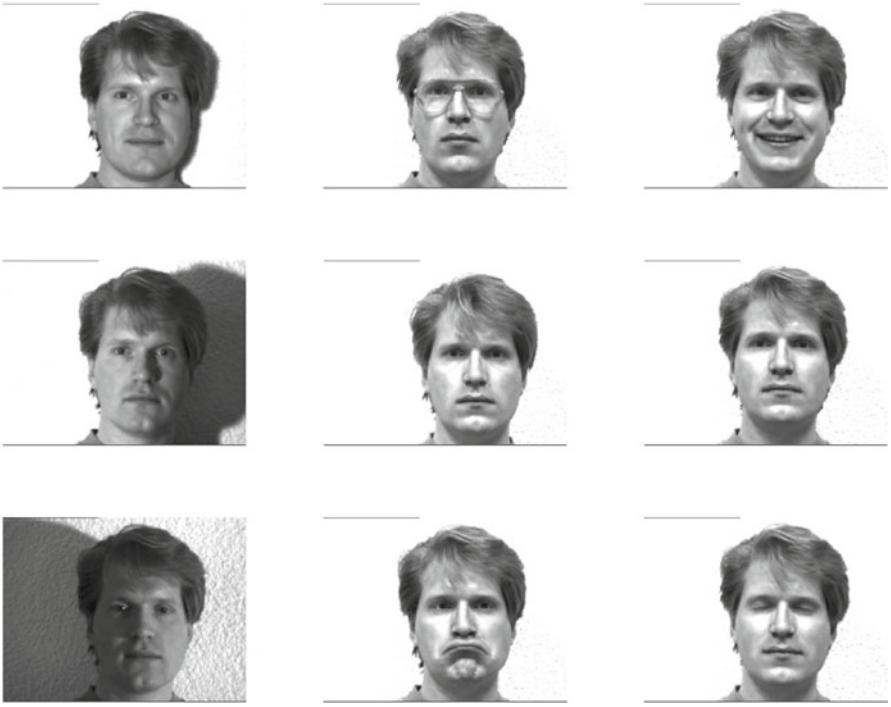

Fig. 8 The YALE Face Database (Georghiades et al. 2001) contains 165 *grayscale* images of 15 individuals. There are 11 images per subject, one per different facial expression or configuration. The *figure* shows *nine* such images for a specific individual. The complete database is available at http://cvc.yale.edu/projects/yalefaces/yalefaces.html

The YALE data base consists of 165 images of 15 individuals. Each individual has 11 images taken with different facial expressions or under different lighting conditions (see, e.g., Fig. 8). These correspond to the 11 inset bars per each of the 15 tick marks in the horizontal axes in Fig. 9. In our experiment, 5 out of 11 images per individual were taken to form a training set from which lower dimensional representatives of the YALE images are found through eigenfaces or Laplacianfaces techniques.

Of course, successful clustering of these representatives should distinguish the 15 groups of images corresponding to the 15 individuals. In Fig. 9, clustering results using FCM, the spectral method, and Fuzzy-RW are shown. Interestingly, Fuzzy-RW recognizes all 15 groups, and performs better than the other methods in terms of minimizing misassignments. Specifically, spectral clustering achieves a 53.3 % success rate in correctly assigning face images to individuals in the data base, whereas Fuzzy-RW with eigenfaces achieves a success rate of 69.7 %, and Fuzzy-RW with Laplacianfaces correctly assigns 71.5 % of the images to the corresponding individuals. In the context of this experiment, the classical FCM algorithm achieves a success rate of only 7.3 %.

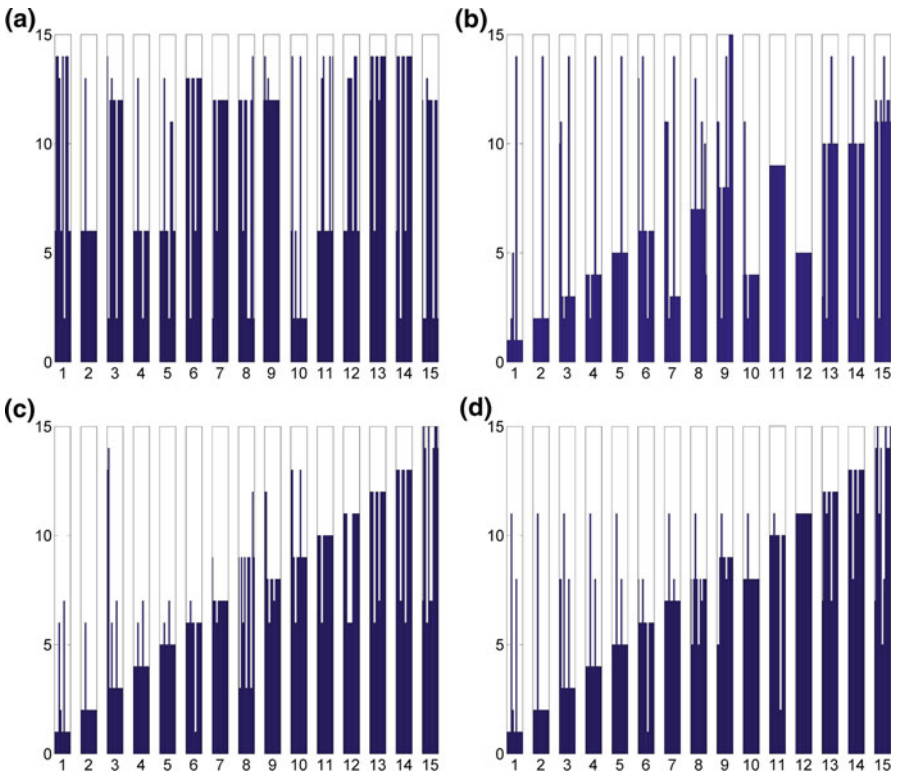

Fig. 9 Plots compare clustering of the YALE data set with respect to different methods. *The horizontal axes* correspond to the 15 individuals in the database, while *the vertical axes* correspond to the clustering assignments. Each individual has 11 images taken with different facial expressions, corresponding to the 11 *inset bars* per index in the *horizontal axis*. Plots (**a**–**d**) give the outputs of the following methods: **a** FCM with eigenfaces, **b** spectral clustering with eigenfaces, **c** Fuzzy-RW with eigenfaces, and **d** Fuzzy-RW with Laplacianfaces. The parameter values used in **c** are $\sigma = 184$, $\gamma = 3$, $k = 15$ where k is the number of clusters, and $K = n$ where $n = 165$ is the number of face images. In **d**, the values are $\sigma = 60$, $\gamma = 1$, $k = 15$, and $K = n$. Here, $n = 165$ images are assigned to clusters by maximal membership

6 Conclusion

We have presented a framework and methodology, namely the Fuzzy-RW method, which allows to cluster data sets difficult for current techniques. Specifically, we introduce several new weighted graphs formed from the data set, and distances defined through random walk step distributions on these graphs which respect more the data set geometry than Euclidean distances. In particular, the "absorption" distance introduced appears novel in the literature. In a nutshell, Fuzzy-RW modifies the classical FCM approach through use of motivated penalty terms and different graph weight schemes which emphasize geometric aspects of the data.

In terms of applications, we have shown on synthetic data sets and real-world data sets such as the IRIS and YALE face recognition data sets that for a choice of parameters our Fuzzy-RW technique outperforms many of the existing methods. Given

that the concept of "clustering" itself is not well-defined, and up to the user and the type of problem being considered, we note that our method is flexible, in terms of how weights are introduced, and can accomodate diverse geometric notions in segregating data, the main point explored in the article with respect to 5 geometrically different data sets [convex clusters (Fig. 1); circle-shaped clusters (Figs. 2, 3); intersecting linear manifolds (Sect. 4.1); Iris data (Sect. 5.1); face recognition data set (Sect. 5.2)]. In turn, we remark a cost for allowing a random-walk distance to explore the data (cf. Sect. 3.2.3) is in the complexity of the algorithm which is polynomial in n, and which would be of interest to improve. Parameter estimation is also a concern in large data sets, as for any clustering method, although one might use a stability or resampling method to identify relevant training values (Levine and Domany 2001; Ben-Hur et al. 2002). In the examples of the article, parameters were found readily through a few trials.

In comparison to other methods, as noted in the Sect. 1, although types of weighted graphs are also used in spectral clustering, our Fuzzy-RW technique is different, not only in the development, but also in that we assign membership values to data points so that outliers can be filtered. In some sense, the Fuzzy-RW method takes into account important features of classical FCM and spectral approaches. Fuzzy-RW is also different than FLAME (Fu and Medico 2007) and DIFFUZZY (Cominetti et al. 2010) which group data according to 'core clusters'. Moreover, Fuzzy-RW differs in particular from the random walk clustering technique in Yen et al. (2005), in that it is not sensitive to the initialization of the centroids.

Finally, we now ask some questions which could point to possible future directions with respect to the Fuzzy-RW method. At the heart of our clustering method is the underlying weighted graph formed from the data points. Might more efficient results be obtained from a 'sparsely' connected graph rather than the complete graph used here? In this context, known constraints in terms of similarity of some data points might also be incorporated into the edge weight structure. Also, could the edge weights and parameters used in the algorithm be 'learned' in some robust way? Could also the number of clusters which is adequate be learned (see, e.g., Cominetti et al. 2010)?

Acknowledgments The research of SL has been supported in part by an Alberta Wolfe Research Fellowship from the Iowa State University Mathematics department. The research of AM has been supported in part by the Mathematical Biosciences Institute and the National Science Foundation under Grant DMS 0931642. The research of SS is supported in part by NSF DMS-1159026.

Appendix

The procedure commonly employed in the literature for minimizing the FCM functional (6) is an alternating directions scheme, originally proposed by Bezdek et al. (1984). For completeness, we provide a listing of the algorithm below. More details can be found in Gan et al. (2007) and Bezdek et al. (1984), among others.

The FCM algorithm:

1: **initiate** the cluster centroids $\{c_i\}_{i \leq k}$.

2: **compute** the fuzzy partition matrix:

$$u_{ij} = \frac{\|x_j - c_i\|^{-\frac{2}{p-1}}}{\sum_{\ell \le k} \|x_j - c_\ell\|^{-\frac{2}{p-1}}}, \quad i \le k, j \le n \tag{19}$$

3: **repeat**
4: **update** the cluster centroids:

$$c_i = \frac{\sum_{j=1}^{n} u_{ij}^{p} x_j}{\sum_{j=1}^{n} u_{ij}^{p}}, \quad i \le k$$

5: **update** $\{u_{ij}\}_{i \le k, j \le n}$ according to (19).
6: **until** a convergence criterion is satisfied.
7: **return** $\{u_{ij}\}_{i \le k, j \le n}$, $\{c_i\}_{i \le k}$

The convergence criterion in line 6 is usually chosen to be of the form $\|U^{(r)} - U^{(r-1)}\| < \varepsilon$ for some pre-specified $\varepsilon > 0$ (Gan et al. 2007). Here, $U^{(r)}$ and $U^{(r-1)}$ denote the values of the fuzzy membership matrix $U = (u_{ij})_{i \le k, j \le n}$ in the r and $r-1$ iteration of the loop, respectively.

Now, the k clusters of data points may be decided in terms of thresholding with respect to the membership matrix, or sometimes data points can be assigned to clusters based on their maximal membership values.

References

Abonyi J, Feil B (2007) Cluster analysis for data mining and system identification. Birkhäuser, Basel
Alamgir M, von Luxburg U (2011) Phase transition in the family of p-resistances. In: Shawe-Taylor J, Zemel R, Bartlett P, Pereira F, Weinberger K (eds) Advances in neural information processing systems (NIPS), vol 24. http://books.nips.cc/papers/files/nips24/NIPS2011_0278.pdf
Arias-Castro E, Chen G, Lerman G (2010) Spectral clustering based on local linear approximations. arXiv:1001.1323v1
Belhumeur P, Hespanha J, Kriegman D (1997) Eigenfaces vs. fisherfaces: recognition using class specific linear projection. IEEE Trans Pattern Anal Mach Intell 19(7):711–720
Belkin M, Niyogi P (2003) Laplacian eigenmaps for dimensionality reduction and data representation. Neural Comput 16:1373–1396
Ben-Hur A, Elisseeff A, Guyon I (2002) A stability based method for discovering structure in clustered data. In: Pacific Symposium on Biocomputing, pp 6–17
Bezdek J, Ehrlich R, Full W (1984) FCM: the fuzzy c-means clustering algorithm. Comput Geosci 10: 191–203
Bezdek J, Hall L, Clark M, Goldgof D, Clarke L (1997) Medical image analysis with fuzzy models. Stat Methods Med Res 6:191–214
Bock H-H (1974) Automatische Klassifikation. Theoretische und praktische Methoden zur Gruppierung und Strukturierung von Daten (Clusteranalyse). Vandenhoek & Ruprecht, Göttingen (in German)
Bock H-H (1987) On the interface between cluster analysis, principal component clustering, and multidimensional scaling. In: Bozdogan H, Gupta A (eds) Multivariate statistical modeling and data analysis. Reidel, Dordrecht, pp 17–34
Brémaud P (1999) Markov chains: Gibbs fields, Monte Carlo simulation, and queues. Springer, New York
Brunelli R, Poggio T (1993) Face recognition: features vs. templates. IEEE Trans Pattern Anal Mach Intell 15(10):1042–1053

Cao F, Delon J, Desolneux A, Museé P, Sur F (2007) A unified framework for detecting groups and application to shape recognition. J Math Imaging Vis 27(2):91–119
Cao F, Lisani J-L, Morel J-M, Museé P, Sur F (2008) A theory of shape identification. Springer, Berlin
Chen G, Lerman G (2009) Spectral curvature clustering (SCC). Int J Comput Vis 81(3):317–330
Chen S, Zhang D (2004) Robust image segmentation using FCM with spatial constraints based on new kernel-induced distance measure. IEEE Trans Syst Man Cybern Part B 34(4):1907–1916
Chung F (1997) Spectral graph theory. CBMS, vol 92. American Mathematical Society, Providence
Coifman R, Lafon S (2006) Diffusion maps. Appl Comput Harmon Anal 21(1):5–30
Cominetti O, Matzavinos A, Samarasinghe S, Kulasiri D, Liu S, Maini P, Erban R (2010) Diffuzzy: a fuzzy clustering algorithm for complex data sets. Int J Comput Intell Bioinforma Syst Biol 1(4):402–417
Desolneux A, Moisan L, Morel J-M (2008) From gestalt theory to image analysis: a probabilistic approach. Springer, New York
Franke M, Geyer-Schulz A (2009) An update algorithm for restricted random walk clustering for dynamic data sets. Adv Data Anal Classif 3(1):63–92
Fu L, Medico E (2007) FLAME, a novel fuzzy clustering method for the analysis of DNA microarray data. BMC Bioinforma **8**(3). doi:10.11861471-2105-8-3
Gan G, Ma C, Wu J (2007) Data clustering: theory, algorithms, and applications. In: ASA-SIAM series on statistics and applied probability. SIAM, Philadelphia
Georghiades A, Belhumeur P, Kriegman D (2001) From few to many: Illumination cone models for face recognition under variable lighting and pose. IEEE Trans Pattern Anal Mach Intell 23(6):643–660
Haralick R, Harpaz R (2005) Linear manifold clustering. In: Perner P, Imiya A (eds) Machine learning and data mining in pattern recognition. Lecture notes in computer science, vol 3587. Springer, Berlin, pp 132–141
Haralick R, Harpaz R (2007) Linear manifold clustering in high dimensional spaces by stochastic search. Pattern Recognit 40(10):2672–2684
Hathaway R, Bezdek J (2001) Fuzzy *c*-means clustering of incomplete data. IEEE Trans Syst Man Cybern Part B 31(5):735–744
He X, Yan S, Hu Y, Niyogi P, Zhang H-J (2005) Face recognition using laplacianfaces. IEEE Trans Pattern Anal Mach Intell 27(3):328–340
Higham D, Kalna G, Kibble M (2007) Spectral clustering and its use in bioinformatics. J Comput Appl Math 204(1):25–37
Jain A (2010) Data clustering: 50 years beyond K-means. Pattern Recognit Lett 31(8):651–666
Kimmel R, Sapiro G (2003) The mathematics of face recognition. SIAM News 36(3). http://www.siam.org/news/news.php?id=309
Kogan J (2007) Introduction to clustering large and high-dimensional data. Cambridge University Press, New York
Lee K-C, Ho JM, Kriegman D (2005) Acquiring linear subspaces for face recognition under variable lighting. IEEE Trans Pattern Anal Mach Intell 27(5):684–698
Levine E, Domany E (2001) Resampling method for unsupervised estimation of cluster validity. Neural Comput 13(11):2573–2593
Liao C-S, Lu K, Baym M, Singh R, Berger B (2009) IsoRankN: spectral methods for global alignment of multiple protein networks. Bioinformatics 25(12):i253–i258
Macqueen J (1967) Some methods for classification and analysis of multivariate observations. In: Proceedings of the 5th Berkeley symposium on mathematical statistics and probability. University of California Press, pp 281–297
Meila M (2006) The uniqueness of a good optimum for *k*-means. In: Cohen W, Moore A (eds) Proceedings of the 23rd international conference on machine Learning, pp 625–632
Miyamoto S, Ichihashi H, Honda K (2008) Algorithms for fuzzy clustering: methods in c-means clustering with applications. Studies in fuzziness and soft computing, vol 229. Springer, Berlin
Muller N, Magaia L, Herbst B (2004) Singular value decomposition, eigenfaces, and 3D reconstructions. SIAM Rev 46(3):518–545
Ng A, Jordan M, Weiss Y (2002) On spectral clustering: analysis and an algorithm. In: Leen T, Dietterich T, Tresp V (eds) Advances in neural information processing systems, vol 14. MIT Press, Cambridge, pp 849–856
Shental N, Bar-Hillel A, Hertz T, Weinshall D (2009) Gaussian mixture models with equivalence constraints. In: Basu S, Davidson I, Wagstaff K (eds) Constrained Clustering: advances in algorithms, theory, and applications. Chapman & Hall, London, pp 33–58

Shi J, Malik J (2000) Normalized cuts and image segmentation. IEEE Trans Pattern Anal Image Segm 22(8):888–905

Snel B, Bork P, Huynen M (2002) The identification of functional modules from the genomic association of genes. PNAS 99(9):5890–5895

Späth H (1985) Cluster dissection and analysis. Ellis Horwood Ltd., Chichester

Tsao J, Lauterbur P (1998) Generalized clustering-based image registration for multi-modality images. Proc 20th Ann Int Conf IEEE Eng Med Biol Soc **20**(2):667–670

Tziakos I, Theoharatos C, Laskaris N, Economou G (2009) Color image segmentation using Laplacian eigenmaps. J Electron Imaging 18(2):023004

von Luxburg U (2007) A tutorial on spectral clustering. Stat Comput 17(4):395–416

von Luxburg U, Radl A, Hein M (2010) Getting lost in space: large sample analysis of the commute distance. In: Lafferty J, Williams CKI, Shawe-Taylor J, Zemel R, Culotta A (eds) Advances in neural information processing systems (NIPS), vol 23. http://books.nips.cc/papers/files/nips23/NIPS2010_0929.pdf

Yen D, Vanvyve F, Wouters F, Fouss F, Verleysen M, Saerens M (2005) Clustering using a random walk based distance measure. In: Verleysen M (ed) In: Proceedings of the 13th European symposium on artificial, neural networks (ESANN), pp 317–324