Noname manuscript No.
(will be inserted by the editor)

Spectral methods for growth curve clustering

Snježana Majstorović · Kristian Sabo · Johannes Jung · Matija Klarić

Received: date / Accepted: date

Abstract The growth curve clustering problem is analyzed and its connection with the spectral relaxation method is described. For a given set of growth curves and similarity function, a similarity matrix is defined, from which the corresponding similarity graph is constructed. It is shown that a nearly optimal growth curve partition can be obtained from the eigendecomposition of a specific matrix associated with a similarity graph. The results are illustrated and analyzed on the set of synthetically generated growth curves. One real-world problem is also given.

Keywords Curve clustering · Similarity graph · Laplacian matrix · Modularity matrix · Spectral methods

1 Introduction

Clustering or grouping a data set into conceptually meaningful clusters (Bezdek 1981) is a well-studied problem in recent literature (Kogan 2007; Su et al. 2010), and it has practical importance in a wide variety of applications such as computer vision, signal-image-video analysis, multimedia, networks, biology,

S. Majstorović
Trg Ljudevita Gaja 6, 31000 Osijek, Croatia
E-mail: smajstor@mathos.hr

K. Sabo
Trg Ljudevita Gaja 6, 31000 Osijek, Croatia
E-mail: ksabo@mathos.hr

J. Jung
Strasse des 17. Juni 135, 10623 Berlin, Germany
E-mail: jung.johannes.92@gmail.com

M. Klarić
Trg Ljudevita Gaja 6, 31000 Osijek, Croatia
E-mail: mklaric2@mathos.hr

medicine, geology, psychology, business, politics and other social sciences. Classification and ranking of objects are also becoming more and more interesting topics for researchers, decision makers and state administrations (Marošević et al. 2013; Turkalj et al. 2016).

Clustering algorithms identify clusters based on some measure of similarity between the objects of the data set. They can be divided into two main groups: hierarchical and partitional (Jain 2010). Hierarchical clustering algorithms recursively find nested clusters either in agglomerative mode (starting with each data point in its own cluster and merging the most similar pair of clusters successively to form a cluster hierarchy) or in divisive mode (starting with all data points in one cluster and recursively dividing each cluster into smaller clusters). They do not require any input parameters, only a similarity measure is needed. The most well-known hierarchical algorithms (Gan et al. 2007) are single-link, complete-link, average-link and Ward algorithm.

Compared to hierarchical clustering algorithms, partitional clustering algorithms find all the clusters simultaneously as a partition of the data and do not produce a hierarchical structure. Because only one set of clusters is the output of a typical partitional clustering algorithm, the user is required to input the desired number of clusters (usually called k). Generally, partitional clustering algorithms are faster than hierarchical clustering.
Partitional clustering algorithms can be divided into two classes, i.e., hard clustering, where each data belongs to only one cluster, and soft clustering, where every data point belongs to every cluster up to a certain degree. The most popular and the simplest hard clustering algorithm is the $k-$means algorithm. Well-known soft clustering methods include the Fuzzy $k-$means (Bezdek 1981), the Expectation Maximization algorithm (see, for example Duda et al. 2011), the smooth $k-$means algorithm that is based on the Euclidean l_2-norm (Kogan 2007), or on the l_1-norm (Sabo 2014), etc.

Spectral clustering is a general class of techniques for clustering a graph derived from the data by using eigenvectors of adjacency (or similarity) matrices. Eigenvectors are used to perform dimensionality reduction before clustering in fewer dimensions (see e.g. Luxburg 2007). These techniques are very useful in hard non-convex clustering problems since they provide new data representation in the low-dimensional space that can be easily clustered. Clustering algorithms that rely on spectral techniques can be used either for partitional clustering such that, after data are projected into a lower-dimensional space (the spectral/eigenvector domain) where they are easily separable, some type of the $k-$means algorithm can be applied, or for hierarchical clustering by repeatedly bipartitioning the subsets in this way. The most popular matrix used for spectral clustering is the well-known Laplacian matrix (and its modifications), but nowadays, with the development of a theory of complex networks, the modularity matrix has become very popular, especially for a very large set of data. Some theoretical results concerning spectral partitioning by using the modularity matrix can be seen in Bolla (2011).

Functional data analysis is an active topic in statistics with a wide range of applications. It extends the classical multivariate methods when data are

functions or curves. Clustering functional data, i.e. curve clustering, is a very challenging task and there are many different approaches to this problem (Jacques & Preda 2014). During the past several years various methods have been developed for curve clustering problem: Sangalli et al. (2010) presented the $k-$ means alignment algorithm, which both clusters and aligns the curves, while Zhang et al. (2015) derived an efficient Bayesian method to cluster curve data using the so-called elastic shape metric. Some very recent results concerning curve clustering methods can be found in Chamroukhi (2016) and Park & Ahn (2017).

In this paper, we study a special curve clustering problem known as growth curve clustering. We consider a nonparametric clustering method, which consists of defining a specific similarity function between curves, and then, after choosing a new point–based representation of each curve, we apply the algorithm for searching for a nearly global optimal partition. The clustering method is based on two spectral clustering techniques, one of them uses Laplacian, and the other uses modularity matrices. To the best of our knowledge, this method has not been used so far to solve the curve clustering problem, not even in the case when curves are of some particular type.

The paper is organized as follows: In Section 2, we define growth curves and give an overview of spectral clustering methods. Section 3 deals with a spectral approach to the problem of growth curve clustering, while in Section 4 we illustrate the possibilities of this method on both synthetically generated curves and a real-world example.

2 Theoretical basis

2.1 Growth curves

A growth curve or growth function is an empirical model of the evolution of a quantity over time. Growth curves are widely used in biology for quantities such as population size, individual body height or biomass. Our problem is motivated by modeling and analysis of pig growth (see Vincek et al. 2012 and Vincek et al. 2012). The growth is a significant physiological activity for all domestic animals, but it is of special interest when meat animals such as pigs, poultry, beef and others are concerned since the growth is nowadays considered as the material base of animal production. Other types of curves that often appear in animal production are lactation curves which describe the quantity of the milk produced over a lactation period. These curves are one of the most important indicators in dairy farm management, see Janković (2016).

In many practical applications, the most frequently used growth function simulating animal weight growth is the logistic function (Jukić & Scitovski 2003; Ratkowsky 1990; Vincek et al. 2012) with parameters $\mathbf{a} = (A, b, c)$ given by

$$f(t; \mathbf{a}) = \frac{A}{1 + e^{-b(t-c)}}, \ A, b > 0. \tag{2.1}$$

It is a solution of the differential equation

$$\frac{dy}{dt} = by\left(1 - \frac{y}{A}\right), \tag{2.2}$$

with initial condition $y(t_0) = y_0$, also known as the logistic growth model. Parameter b can be interpreted as the maximum possible rate of the animal weight growth (it determines the steepness of the logistic curve), parameter A is the upper limit of weight growth called carrying capacity and

$$c = t_0 + \frac{1}{b}\ln\frac{A - y_0}{y_0}$$

is a t-coordinate of the logistic curve inflection point. From (2.2) we can easily see that the early, unimpeded growth rate is modeled by the first term by. Later, as the animal weight grows, the modulus of the second term, which is after multiplication $-\frac{by^2}{A}$, becomes almost as large as the first. So in early stages the weight increases exponentially but levels off eventually and approaches its carrying capacity due to limited recourses. On the interval $(-\infty, +\infty)$, the graph of y is an "S"-shaped curve with the horizontal asymptotes $y = 0$ and $y = A$.

For the purpose of determining life cycle phases in animal weight growth the following growth functions are also considered: the generalized logistic function with parameters, the Gompertz function with parameters, and the von Bertalanffy growth function with parameters. General growth functions which combine several other growth functions can also be found in the literature, such as the well-known Richardson growth function (see Ratkowsky 1990).

2.2 Spectral clustering

Spectral clustering is a very popular clustering method. It can be easily implemented and efficiently solved by standard linear algebra methods. In order to partition a given set $\mathcal{S} = \{x_i : i = 1, \ldots, n\}$ of data points into k clusters by using spectral clustering methods, we need to construct a similarity graph $G = (V, E)$. Each vertex v_i in such graph represents a point x_i and two vertices are connected by an edge if some imposed condition on the calculated similarity between the corresponding data points is fulfilled. This is possible only if we are able to define a specific similarity function between data points in $\mathcal{S}$.

There are many types of similarity graphs and their main purpose is to model the local neighborhood relationships between data points. The most popular similarity graphs are as follows: the $\epsilon-$*neighborhood graph*, in which we connect all points whose pairwise distances are smaller than ϵ, the $k-$*nearest neighbor graph*, in which we connect vertex u with vertex v if v is among the $k-$nearest neighbors of u, and the *fully connected graph*, in which we connect

all points with positive similarity with each other (Luxburg 2007). For basic definitions and terminology in graph theory, see Diestel (2000).

After the similarity graph is constructed, the problem of clustering can be reformulated: we want to partition the similarity graph into two or more groups such that the edges between different groups have very low weights and the edges within a group have high weights. The best way to perform the partition is either to solve the min-cut problem (Luxburg 2007) or to maximize the Newman-Girvan modularity (Newman & Girvan 2004). Since these problems are in general NP-hard, it is necessary to study their relaxed version. This leads to spectral clustering methods which require the usage of some special matrices associated to the graph. Other examples of NP-hard problems can be seen in Ćwik (2017).

In what follows, we define several types of such matrices and give a brief description of their main properties. For basic definitions and terminology in matrix theory, see Golub & Van Loan (1996).

Let G be a undirected, weighted graph with vertex set $V = \{v_1, \ldots, v_n\}$ and weighted adjacency matrix $\mathbf{W}$, where $w_{ij} = w_{ji} \geq 0$ and $w_{ii} = 0$, $i, j = 1, \ldots, n$. The degree of a vertex v_i is defined as

$$d_i = \sum_{j=1}^{n} w_{ij}.$$

Let $\mathbf{D}$ be a diagonal matrix with vertex degrees d_i, $i = 1, \ldots, n$, on the diagonal.

• The unnormalized Laplacian matrix $\mathbf{L}$ associated with the graph G is defined as

$$\mathbf{L} = \mathbf{D} - \mathbf{W}.$$

Matrix $\mathbf{L}$ is symmetric positive semidefinite with the property that the sum of elements in each row or column is equal to zero. Consequently, $\mathbf{L}$ has a real valued spectrum, all cofactors are equal, the smallest eigenvalue is zero and the corresponding eigenvector is all-one vector $\mathbf{1}^\tau$. The number of connected components of G corresponds to the multiplicity of the eigenvalue 0 of $\mathbf{L}$.

• The symmetric normalized Laplacian matrix $\mathbf{L}_{sym}$ is defined as

$$\mathbf{L}_{sym} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}.$$

Matrix $\mathbf{L}_{sym}$ is a symmetric real matrix with ones on the main diagonal. Matrices $\mathbf{L}_{sym}$ and $\mathbf{L}$ are congruent matrices so by Sylvester's law of inertia they have the same numbers of positive, negative, and zero eigenvalues.

• The random walk normalized Laplacian matrix $\mathbf{L}_{rw}$ is defined as

$$\mathbf{L}_{rw} = \mathbf{D}^{-1}\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{W},$$

where $\mathbf{D}^{-1}\mathbf{W}$ is the transition matrix of a random walk on G. This matrix is non-symmetric, having ones on the main diagonal and the sum of elements in each row is equal to zero. Matrices $\mathbf{L}_{sym}$ and $\mathbf{L}_{rw}$ are similar matrices so they have the same spectrum. According to the Gershgorin circle theorem, all eigenvalues are contained in $[0, 2]$. It is easy to check that $(\lambda, \mathbf{u})$ is an eigenpair of $\mathbf{L}_{rw}$ if and only if $(\lambda, \mathbf{D}^{1/2}\mathbf{u})$ is an eigenpair of $\mathbf{L}_{sym}$.

• The modularity matrix $\mathbf{M}$ is defined as

$$\mathbf{M} = \mathbf{W} - \mathbf{d}\mathbf{d}^{\tau},$$

where $\mathbf{d} = (d_1, \ldots, d_n)^{\tau}$ is the degree vector comprising the main diagonal of $\mathbf{D}$ and, without loss of generality, $\mathbf{W}$ has an additional property that the sum of its elements is equal to one. Matrix $\mathbf{M}$ is a symmetric indefinite matrix with the sum of elements in each row or column equal to zero. Therefore, it has an eigenvalue zero with the corresponding eigenvector $\mathbf{1}^{\tau}$ (Majstorović & Stevanović 2014).

• The normalized modularity matrix $\mathbf{M}_D$ is defined as

$$\mathbf{M}_D = \mathbf{D}^{-1/2}\mathbf{M}\mathbf{D}^{-1/2} = \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2} - \sqrt{\mathbf{d}}\sqrt{\mathbf{d}}^{\,\tau},$$

where $\sqrt{\mathbf{d}} = (\sqrt{d_1}, \ldots, \sqrt{d_n})^{\tau}$. Matrix $\mathbf{M}_D$ is a symmetric real matrix with the eigenvalues in $[-1, 1]$. One of its eigenvalues is zero and the corresponding eigenvector is $\sqrt{\mathbf{d}}$. The relation between $\mathbf{M}_D$ and $\mathbf{L}_{sym}$ is the following:

$$\mathbf{M}_D = \mathbf{I} - \mathbf{L}_{sym} - \sqrt{\mathbf{d}}\sqrt{\mathbf{d}}^{\,\tau}.$$

Matrices $\mathbf{M}$ and $\mathbf{M}_D$ are congruent matrices.

Spectral clustering methods use these matrices to make a new representation of data set $\mathcal{S}$ in which clusters can be easily detected. For a given similarity matrix, we construct a weighted adjacency matrix $\mathbf{W}$ of the corresponding similarity graph. By using k suitable eigenvectors of one of the above mentioned matrices, vertices, i.e. data points, are projected from $n-$dimensional to $k-$dimensional space providing a new kind of representation. It was proved that the $k-$means algorithm can easily detect the clusters in such data representation (see Luxburg 2007). Beside $k-$means, any other type of the clustering algorithm can also be applied.

If we choose one of the three types of Laplace matrices, then we need to calculate k eigenvectors corresponding to its k smallest eigenvalues. This approach is a relaxed version of the well-known min-cut problem in graph theory.

For one of the mentioned modularity matrices we need to calculate k eigenvectors corresponding to its k largest eigenvalues and this is a relaxed version of the Newman-Girvan modularity maximization problem, nowadays very popular in the theory of complex networks.

Both approaches have the same purpose, i.e. to detect clusters in a graph. The only difference is that the min-cut approach is mainly focused on minimizing the sum of weights of edges between clusters, while the Newman-Girvan modularity approach is focused on maximizing the weights of edges inside clusters.

Spectral clustering algorithms for all types of matrices considered here are given in Section 3.1.

Remark 1 *The main difficulty of clustering algorithms is the estimation of k, where k is the number of clusters. For Laplacian matrices the appropriate tool is the eigengap heuristic. The optimal k is the one for which the k smallest eigenvalues $\lambda_1 < \cdots < \lambda_k$ are very small and λ_{k+1} is relatively large.*
For modularity matrices optimal k is the one for which $k = p + 1$, where p is the number of positive eigenvalues.

3 The proposed approach to growth curve clustering

Let $\mathcal{H} = \{f(t\,;\,\mathbf{a}_i)\,:\,i = 1, \ldots, n\}$ be the set of growth curves defined with (2.1) and let us suppose $\dfrac{df(t\,;\,\mathbf{a}_i)}{dt} \in L^2(\mathbb{R}, \mathbb{R})$, $\mathbf{a}_i \in \mathbb{R}^l$. Growth curve clustering is a problem of partitioning a set of growth curves into k subsets called clusters, $1 \leq k \leq n$, so that the curves inside each of them are very similar among themselves (equivalently, they are closest to each other) and as different as possible from the curves of the other clusters (equivalently, they are furthest from the curves of other clusters).

For a set $\mathcal{H}$ of growth curves, we are interested in the corresponding growth velocities. Therefore, we consider a dissimilarity function $d : \mathcal{H} \times \mathcal{H} \to \mathbb{R}_+$ between curves defined as

$$d(f(t;\mathbf{a}_i), f(t;\mathbf{a}_j)) = \int_\alpha^\beta \left(\frac{f'(t;\mathbf{a}_i)}{\sqrt{\int_\alpha^\beta (f'(t;\mathbf{a}_i))^2\,dt}} - \frac{f'(t;\mathbf{a}_j)}{\sqrt{\int_\alpha^\beta (f'(t;\mathbf{a}_j))^2\,dt}} \right)^2 dt, \tag{3.1}$$

$0 \leq \alpha < \beta < \infty$, which can be written as

$$d(f(t;\mathbf{a}_i), f(t;\mathbf{a}_j)) = 2(1 - s(f(t;\mathbf{a}_i), f(t;\mathbf{a}_j))),$$

with $s : \mathcal{H} \times \mathcal{H} \to \mathbb{R}_+$ being a similarity index between curves defined as

$$s(f(t;\mathbf{a}_i), f(t;\mathbf{a}_j)) = \int_\alpha^\beta \frac{f'(t;\mathbf{a}_i) \cdot f'(t;\mathbf{a}_j)}{\sqrt{\int_\alpha^\beta (f'(t;\mathbf{a}_i))^2\,dt} \cdot \sqrt{\int_\alpha^\beta (f'(t;\mathbf{a}_j))^2\,dt}}\,dt. \tag{3.2}$$

This index was introduced in Sangalli et al. (2009) as a cosine of the angle between first derivatives of the functions $f(t;\mathbf{a}_i)$ and $f(t;\mathbf{a}_j)$ with the inner product

$$\langle f(t;\mathbf{a}_i) | f(t;\mathbf{a}_j) \rangle = \int_\alpha^\beta f'(t;\mathbf{a}_i) \cdot f'(t;\mathbf{a}_j)\,dt.$$

By $\Pi(\mathcal{H}) = \{\pi_1, \ldots, \pi_k\}$ we denote a partition of the set $\mathcal{H}$ into k subsets $\pi_1, \ldots, \pi_k$, $1 \leq k \leq n$, i.e.

$$\begin{aligned} \bigcup_{i=1}^{k} \pi_i &= \mathcal{H}, \\ \pi_i \cap \pi_j &= \emptyset,\ i \neq j, \\ |\pi_j| &\geq 1,\ j = 1, \ldots, k. \end{aligned} \tag{3.3}$$

Elements $\pi_1, \ldots, \pi_k$ of such partition are called curve-clusters. To each cluster $\pi_j \in \Pi$ we can associate its curve-center $f(t; \mathbf{c}_j)$, where $\mathbf{c}_j$ is defined by

$$\mathbf{c}_j = \arg\min_{\mathbf{c}} \sum_{i=1}^{|\pi_j|} d(f(t; \mathbf{a}_i), f(t; \mathbf{c})). \tag{3.4}$$

If we define an objective function $\mathcal{F} \colon \mathcal{P}(\mathcal{H}, k) \to \mathbb{R}_+$ on the set of all partitions $\mathcal{P}(\mathcal{H}, k)$ of $\mathcal{H}$ containing k clusters by

$$\mathcal{F}(\Pi) = \sum_{j=1}^{k} \sum_{f(t;\mathbf{a}_i) \in \pi_j} d(f(t; \mathbf{c}_j), f(t; \mathbf{a}_i)), \tag{3.5}$$

then we can define an optimal curve-partition $\Pi^\star$, such that

$$\mathcal{F}(\Pi^\star) = \min_{\Pi \in \mathcal{P}(\mathcal{H}, k)} \mathcal{F}(\Pi).$$

Instead of minimizing the function $\mathcal{F}$ given by (3.5) directly, the well-known $k-$means algorithm for finding the locally optimal partition could be applied. In order to do this, we need to choose k arbitrary distinct initial curve-centers, that is, curves $f(t; \mathbf{c}_1), \ldots, f(t; \mathbf{c}_k)$. Then we need to define clusters

$$\pi_j := \{f(t; \mathbf{a}_i) \,:\, d(f(t; \mathbf{a}_i), f(t; \mathbf{c}_j)) \leq d(f(t; \mathbf{a}_i), f(t; \mathbf{c}_l)),\ l = 1, \ldots, k,\ l \neq j\}, \tag{3.6}$$

which means that we assign each function $f(t; \mathbf{a}_i)$ to the cluster based on the curve-center it is least dissimilar to in terms of the dissimilarity function (3.1). After all functions are assigned to clusters, the cluster centers are updated by the optimization procedure (3.4). $k-$means clustering of curves was considered in Tarpey & Kinateder (2003), while the $k-$means alignment algorithm which both clusters and aligns curves was proposed in Sangalli et al. (2010).

Optimization problem (3.4) is very difficult to solve since we deal with global optimization on a space of parameters. Therefore, instead of the optimization procedure, we define a similarity matrix from which we construct the corresponding similarity graph and use well-known spectral methods that allow us to make a new representation of the data set, i.e., the set of curves for which any simple clustering algorithm such as $k-$means can be applied. The most suitable choice of the similarity graph for the growth curve clustering problem is the fully connected graph, that is, the one in which every pair of vertices is connected by an edge. The edges are weighted with numbers

$s_{ij} = s(f(t;\mathbf{a}_i), f(t;\mathbf{a}_j))$, where $s : \mathcal{H} \times \mathcal{H} \to \mathbb{R}_+$ is defined by (3.2), that is, with similarities between the corresponding curves.
After the new point–based representation of each curve is obtained by using spectral techniques, we apply the algorithm for searching for a nearly global optimal partition, proposed in Scitovski & Scitovski (2013). We decided to use this algorithm as a substitute for the usual $k-$means algorithm because it generalizes already known incremental algorithms for the purpose of finding a good initial approximation for $k-$means. This algorithm locates either a globally optimal partition or a locally optimal partition close to the global one, and it requires significantly shorter CPU-time than other incremental algorithms.

Alternative approach

In most applications the functions are only observed at a finite number of time points. This motivates us to consider an alternative approach to growth curve clustering in which we ignore the functional nature of data set $\mathcal{H}$. Therefore, instead of considering a set $\mathcal{H}$ of growth curves defined with (2.1), we can consider the set $\mathcal{S}$ consisting of N-dimensional vectors with components corresponding to function values in N fixed time points. Spectral clustering of $\mathcal{S}$ can be performed as described in Section 2.2. Again, the most suitable choice of a similarity graph is the fully connected graph and the most common similarity function which models local neighborhoods is the Gaussian similarity function (Ng et al. 2001) defined as

$$g(x_i, x_j) = e^{-\frac{||x_i - x_j||^2}{2\gamma^2}}, \quad x_i, x_j \in \mathcal{S}, \tag{3.7}$$

where parameter γ controls the width of the neighborhoods.

3.1 Growth curve clustering algorithm

A big advantage of the spectral approach to growth curve clustering is its simplicity. The spectral algorithm is easy to implement in any software such as Matlab, Mathematica, etc. since it uses basic linear algebra theorems, and hence it is simple to solve.
First, we state the algorithm that uses one of the three types of Laplacian matrices:

INPUT: Set of growth curves $\mathcal{H}$, number k of clusters.

- Construct similarity matrix $S \in \mathbb{R}^{n\times n}$ based on a specific similarity measure for comparison of growth curves.
- Construct the similarity graph.
- Construct weighted adjacency matrix W of the constructed similarity graph.
- Compute matrix LAP which is one of the matrices L, L_{sym}, L_{rw}.

(↓ spectral clustering)
• Compute k eigenvectors $u_1, \ldots, u_k$ of LAP which correspond to the k smallest
eigenvalues and place them as columns in matrix $U \in \mathbb{R}^{n \times k}$.
(For L_{sym} we need an extra step which is a normalization of rows of U to a length of 1.)
• Cluster the points $(y_i)_{i=1,\ldots,n}$ in $\mathbb{R}^k$, which are vectors corresponding to the rows of U, with the algorithm for finding a nearly global optimal partition, proposed in Scitovski & Scitovski (2013), into clusters $C_1, \ldots, C_k$.
OUTPUT: Clusters $C_1, \ldots, C_k$.

For one of the two types of modularity matrices the algorithm is very similar to the previous one:

INPUT: Set of growth curves $\mathcal{H}$, number k of clusters.

• Construct similarity matrix $S \in \mathbb{R}^{n \times n}$ based on a specific similarity measure for comparison of growth curves.
• Construct the similarity graph.
• Construct weighted adjacency matrix W of the constructed similarity graph.
• Compute matrix MOD which is one of the matrices M, M_D.
(↓ spectral clustering)
• Compute k eigenvectors $u_1, \ldots, u_k$ of MOD which correspond to the k largest
eigenvalues and place them as columns in matrix $U \in \mathbb{R}^{n \times k}$.
(For M_D we need an extra step which is a normalization of rows of U to a length of 1.)
• Cluster the points $(y_i)_{i=1,\ldots,n}$ in $\mathbb{R}^k$, which are vectors corresponding to the rows of U, with the algorithm for finding a nearly global optimal partition, proposed in Scitovski & Scitovski (2013), into clusters $C_1, \ldots, C_k$.
OUTPUT: Clusters $C_1, \ldots, C_k$.

4 Numerical experiments

4.1 Synthetic data

In this section, we generate synthetic data to illustrate the possibilities of spectral clustering methods that use five types of matrices described in Section 2.2.

For this purpose we consider $2 \leq k \leq 10$ curve templates

$$f(t; A_j, b_j, c_j) = \frac{A_j}{1 + e^{-b_j(t-c_j)}}, \quad j = 1, 2, \ldots, k$$

with arbitrarily chosen parameters (A_j, b_j, c_j) given in Table 1.

Table 1: Curve template parameters

k	2	3	4	5	6
(A, b, c)	(100,0.8,8.1)	(90,0.3,7.4)	(91,0.52,6.4)	(95,0.5,5.4)	(94,0.5,5.1)
	(101,1.1,8.3)	(88,0.6,9.2)	(92,0.7,9)	(95,0.6,5)	(96,0.7,5.8)
		(91,0.5,8)	(90,0.8,8)	(94,0.7,6.2)	(97,0.7,6.1)
			(92,0.5,7.8)	(96,0.4,6.6)	(94,0.6,6.2)
				(96,0.5,6.7)	(94,1,7)
					(95,0.55,5.9)
k	7	8	9	10	
(A, b, c)	(97,0.5,5.1)	(99,0.5,5.1)	(101,0.6,5)	(104,0.7,4)	
	(99,1,5.8)	(97,1,6)	(94,0.7,6)	(96,0.5,7)	
	(96,3.7,6.1)	(98,2,6)	(98,1.5,8)	(98,1,5)	
	(97,2.1,4)	(102,0.4,5.5)	(102,2,5.6)	(102,2,5.6)	
	(105,2.5,7)	(104,1,7)	(104,1,8)	(104,1,7)	
	(100,3,5.9)	(100,0.8,5.1)	(100,0.8,5.1)	(100,0.8,5.1)	
	(102,4.1,6.9)	(108,0.7,6.9)	(108,0.7,6.9	(108,0.7,6.9	
		(102,1.1,6.9)	(102,1.1,6.9)	(102,1.1,6.9)	
			(108,1,4)	(108,1,4)	
				(105,0.5,4)	

For each template we generate 10 data sets

$$\mathcal{A}_{jl} = \{(t_i, A_j/(1+e^{-b_j(t-c_j)})+\varepsilon_i),\ i = 1, \ldots, 100\},\ j = 1, 2, \ldots, k,\ l = 1, \ldots, 10,$$

where $t_i = \frac{20}{100} i$ and $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$. We estimate parameters $(A^\star_{jl}, b^\star_{jl}, c^\star_{jl})$ by minimizing the least squares objective function

$$F_{jl}(A, b, c) = \sum_{(t_i, y_i^{(jl)}) \in \mathcal{A}_{jl}} \left(A/(1 + e^{-b(t-c)}) - y_i^{(jl)} \right)^2.$$

In this way, we generate the partition $\Pi^\star = \{\pi^\star_1, \pi^\star_2, \ldots, \pi^\star_k\}$ which consists of k clusters of curves

$$\pi^\star_j = \{A^\star_{jl}/(1 + e^{-b^\star_{jl}(t-c^\star_{jl})}),\ l = 1, \ldots, 10\}, \quad j = 1, \ldots, k.$$

Let $\mathcal{H} = \pi^\star_1 \cup \pi^\star_2 \cup \ldots \cup \pi^\star_k$. By using spectral partitioning algorithms we partition $\mathcal{H}$ into k clusters and obtain the partition $\hat{\Pi}$. Then, we use the adjusted Rand index (ARI) to measure the similarity between partitions $\hat{\Pi}$ and $\Pi^\star$. Details on the adjusted Rand index are given in Appendix B.

Figures 1 (a), (b) and (c) show data sets of growth curves generated from logistic curve templates with parameters $(90, 0.3, 7.4)$ (red), $(91, 0.5, 8)$ (green)

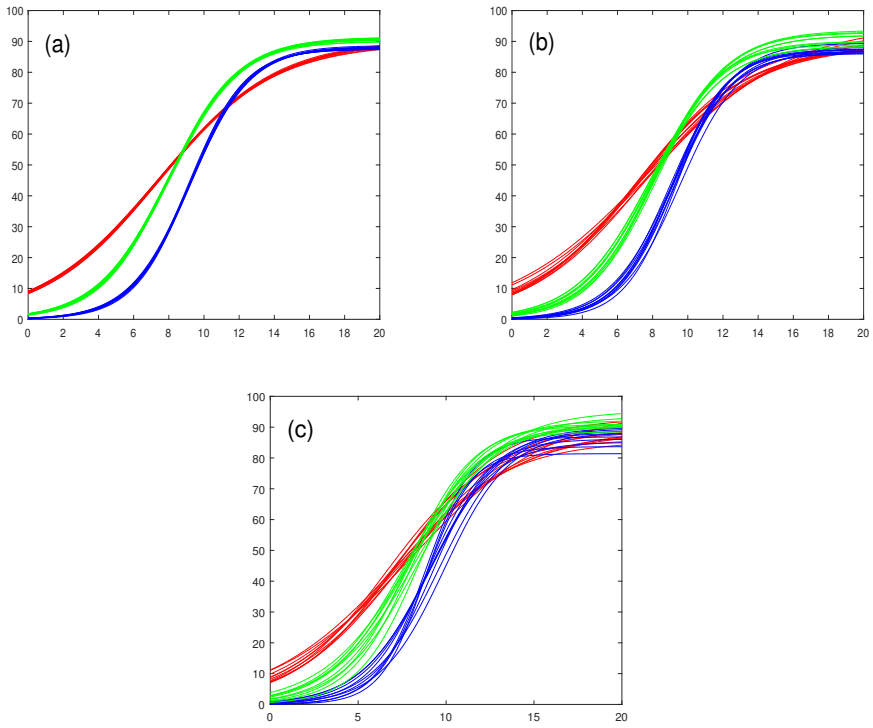

Fig. 1: Generated sets of growth curves with variance (a) $\sigma^2 = 10$, (b) $\sigma^2 = 150$ and (c) $\sigma^2 = 500$.

and $(88, 0.6, 9.2)$ (blue) on the interval $[0, 20]$. Errors are all independent and normally distributed with mean 0 and variances 10, 150 and 500. When growth curves are more spread out from the corresponding curve templates, that is, when variance is large, the clustering structure of the data set is less obvious.

All calculations were performed on the MatLab platform. For each clustering value k we considered 7 different values of variance σ^2: $10, 50, 100, 150, 250, 300$ and 500. Then, for each k and each σ^2 we generated 30 data sets of curves to which we applied the spectral clustering method and calculated the ARI indices. This enabled us to make statistical results concerning the ARI index: its mean value μ and its standard deviation sd.
The algorithm for searching for a nearly global optimal partition was run up to 150000 times for each of the five matrices considered.

Table 2 shows mean values and standard deviations of ARI indices for the spectral growth curve clustering method which uses cosine similarity of curves defined with (3.2). The method is tested for all five matrices described in Section 2.2. Results indicate that the mean value of the ARI decreases as the variance grows. Standard deviation is usually low, which means that ARI

indices tend to be close to the mean value. The spectral method is most effective when symmetric normalized matrix $\mathbf{L}_{sym}$ is used. This is justified by the fact that matrix $\mathbf{L}_{sym}$ appears in a relaxed version of the normalized min-cut problem in which the objective function tries to balance clusters with respect to the sum of their edge weights, i.e. similarities between their vertices. The objective function related to random walk Laplacian $\mathbf{L}_{rw}$ is the same, but there is no additional step in which vectors corresponding to a new representation of data are normalized to norm 1 (which happens in the case of $\mathbf{L}_{sym}$). The objective function that uses Laplace matrix $\mathbf{L}$ balances clusters with respect to their sizes. This approach has deficiencies because sizes of clusters are not necessarily related to within–cluster similarity. Within–cluster similarity depends on the edges and not on the vertices in the cluster. A similar comparison can be done for a modularity matrix $\mathbf{M}$ and a normalized modularity matrix $\mathbf{M}_D$: the objective function that uses $\mathbf{M}$ considers only the sizes of clusters, while the one that uses $\mathbf{M}_D$ balances clusters with respect to their edge weights. This explains why $\mathbf{M}_D$ gives better results than $\mathbf{M}$.

To further explore the efficiency of spectral methods, we used generated sets of growth curves to test the spectral clustering method described in Section 3. For a set of $N = 2001$ fixed time points we calculated growth function values. In this way, each growth curve is represented by a $2001-$dimensional vector. To measure similarity between these vectors we used the Gaussian similarity function in which the value of parameter γ as well as the choice of norm $||\cdot||$ depends on the particular set of generated growth curves. Mean values and standard deviations of the ARI for this method are given in Table 3. Results for Laplace matrix $\mathbf{L}$ were excluded due to some calculation errors.
Although mean values and standard deviations of the ARI are much higher than the ones obtained using cosine similarity, their behaviour is similar. Mean values tend to decrease as variance grows, while standard deviation is low. Again, matrix $\mathbf{L}_{sym}$ gives the best results, but the efficiency of the spectral method which uses other matrices is quite close to the one that uses $\mathbf{L}_{sym}$.

Table 4 is crucial to proving that spectral-based clustering techniques are more efficient for growth curve clustering than some other well-known approaches. For this purpose, we studied a family of algorithms that are implemented in the so–called Curve Clustering Toolbox, a Matlab toolbox that implements a family of two-stage clustering algorithms combining a mixture of Gaussian models with spline or polynomial basis approximation (see Gaffney 2004). Considering the nature of our data set of curves, an appropriate choice was the linear regression mixture model. We compared this method with the spectral method that uses cosine similarity between curves and concluded that the spectral method gives much better results.
For the alternative approach, we compared mean values and standard deviations of the ARI obtained from spectral methods using the Gaussian similarity function on the set of $2001-$dimensional vectors with the ordinary $k-$means algorithm. We just inserted these vectors into the $k-$means algorithm as row data. The conclusion in this case is that the efficiency of $k-$means is comparable to our spectral approach, but the spectral approach still gives better

results. We should also mention that spectral methods are much faster than $k-$means.

Table 2: Mean value μ and standard deviation sd of ARI indices for the spectral growth curve clustering method that uses cosine similarity.

	L	L_{rw}	L_{sym}	M	M_D
$\sigma^2 = 10$	μ; sd	μ; sd	μ; sd	μ; sd	μ; sd
$k = 2$	1; 0	1; 0	1; 0	-0.389; 0.015	0.354; 0.230
$k = 3$	0.507; 0.006	1; 0	1; 0	-0.042; 0.016	0.548; 0.175
$k = 4$	0.507; 0.006	1; 0	1; 0	-0.042; 0.016	0.548; 0.175
$k = 5$	0.324; 0.140	0.662; 0.073	0.886; 0.069	0.389; 0.253	0.516; 0.085
$k = 6$	0.069; 0.340	0.652; 0.011	0.771; 0.024	0.527; 0.146	0.412; 0.095
$k = 7$	0.048; 0.019	0.837; 0	0.971; 0.013	0.889; 0.136	0.634; 0.019
$k = 8$	0.023; 0.010	0.645; 0	0.827; 0.001	0.372; 0.059	0.687; 0.014
$k = 9$	0.041; 0.027	0.467; 0.019	0.843; 0.006	0.316; 0.050	0.695; 0.026
$k = 10$	0.199; 0.010	0.552; 0.010	0.848; 0.025	0.310; 0.039	0.542; 0.012
$\sigma^2 = 50$					
$k = 2$	1; 0	1; 0	1; 0	-0.032; 0.020	0.351; 0.242
$k = 3$	0.513; 0.017	1; 0	1; 0	-0.036; 0.024	0.523; 0.167
$k = 4$	0.253; 0.045	0.761; 0.106	0.910; 0.060	0.808; 0.254	0.477; 0.580
$k = 5$	0.292; 0.174	0.628; 0.058	0.841; 0.054	0.268; 0.151	0.454; 0.100
$k = 6$	0.069; 0.340	0.652; 0.011	0.771; 0.024	0.527; 0.146	0.412; 0.095
$k = 7$	0.047; 0.019	0.842; 0.029	0.976; 0.016	0.482; 0.026	0.629; 0.018
$k = 8$	0.065; 0.083	0.643; 0.006	0.822; 0.010	0.230; 0.059	0.687; 0.012
$k = 9$	0.047; 0.027	0.472; 0.024	0.830; 0.016	0.305; 0.072	0.648; 0.035
$k = 10$	0.202; 0.039	0.546; 0.036	0.823; 0.090	0.321; 0.039	0.547; 0.039
$\sigma^2 = 100$					
$k = 2$	0.630; 0.455	0.973; 0.068	0.980; 0.060	-0.032; 0.020	0.357; 0.247
$k = 3$	0.517; 0.019	0.993; 0.025	0.993; 0.025	0.058; 0.174	0.446; 0.177
$k = 4$	0.232; 0.088	0.759; 0.114	0.907; 0.074	0.954; 0.112	0.476; 0.760
$k = 5$	0.174; 0.148	0.625; 0.053	0.776; 0.092	0.223; 0.179	0.453; 0.101
$k = 6$	0.028; 0.023	0.561; 0.047	0.606; 0.072	0.429; 0.074	0.325; 0.074
$k = 7$	0.038; 0.021	0.893; 0.090	0.972; 0.028	0.437; 0.077	0.616; 0.018
$k = 8$	0.128; 0.109	0.639; 0.009	0.805; 0.022	0.262; 0.046	0.676; 0.017
$k = 9$	0.038; 0.028	0.464; 0.028	0.800; 0.032	0.300; 0.070	0.599; 0.051
$k = 10$	0.196; 0.016	0.546; 0.011	0.801; 0.035	0.314; 0.039	0.567; 0.055
$\sigma^2 = 150$					
$k = 2$	0.497; 0.465	0.929; 0.127	0.929; 0.127	-0.030; 0.026	0.269; 0.230
$k = 3$	0.504; 0.095	0.971; 0.060	0.970; 0.062	0.057; 0.169	0.376; 0.168
$k = 4$	0.232; 0.910	0.800; 0.122	0.871; 0.100	0.893; 0.097	0.471; 0.092
$k = 5$	0.122; 0.145	0.606; 0.063	0.741; 0.087	0.185; 0.129	0.437; 0.085
$k = 6$	0.032; 0.023	0.509; 0.046	0.545; 0.050	0.341; 0.050	0.279; 0.063
$k = 7$	0.045; 0.037	0.901; 0.088	0.960; 0.031	0.345; 0.089	0.600; 0.021
$k = 8$	0.129; 0.107	0.627; 0.017	0.796; 0.037	0.238; 0.046	0.664; 0.014
$k = 9$	0.040 0.028	0.465; 0.048	0.759; 0.042	0.246; 0.066	0.557; 0.050
$k = 10$	0.183; 0.036	0.534; 0.018	0.763; 0.050	0.307; 0.047	0.546; 0.043
$\sigma^2 = 250$					
$k = 2$	0.181; 0.281	0.764; 0.195	0.813; 0.149	-0.026; 0.026	0.167; 0.201
$k = 3$	0.452; 0.169	0.875; 0.111	0.889; 0.097	0.091; 0.205	0.350; 0.198
$k = 4$	0.229; 0.088	0.805; 0.102	0.840; 0.097	0.770; 0.131	0.390; 0.061
$k = 5$	0.084; 0.116	0.542; 0.070	0.628; 0.090	0.155; 0.088	0.369; 0.095
$k = 6$	0.021; 0.014	0.455; 0.042	0.487; 0.048	0.326; 0.067	0.239; 0.050
$k = 7$	0.059; 0.063	0.863; 0.071	0.897; 0.055	0.312; 0.080	0.575; 0.024
$k = 8$	0.171; 0.096	0.594; 0.027	0.726; 0.046	0.255; 0.054	0.606; 0.032
$k = 9$	0.026; 0.020	0.451; 0.057	0.684; 0.050	0.241; 0.048	0.511; 0.055
$k = 10$	0.172; 0.045	0.511; 0.018	0.682; 0.050	0.283; 0.036	0.519; 0.040
$\sigma^2 = 300$					
$k = 2$	0.311; 0.400	0.772; 0.166	0.767; 0.180	-0.030; 0.022	0.245; 0.221
$k = 3$	0.429; 0.196	0.890; 0.104	0.898; 0.093	0.064; 0.118	0.293; 0.208
$k = 4$	0.160; 0.129	0.779; 0.098	0.789; 0.104	0.724; 0.149	0.385; 0.079
$k = 5$	0.059; 0.110	0.541; 0.073	0.607; 0.077	0.187; 0.108	0.384; 0.081
$k = 6$	0.026; 0.021	0.420; 0.051	0.456; 0.047	0.302; 0.060	0.226; 0.046
$k = 7$	0.035; 0.021	0.844; 0.081	0.884; 0.063	0.357; 0.089	0.570; 0.033
$k = 8$	0.151; 0.104	0.583; 0.030	0.690; 0.055	0.210; 0.054	0.582; 0.047
$k = 9$	0.028; 0.023	0.430; 0.038	0.668; 0.065	0.220; 0.055	0.494; 0.053
$k = 10$	0.170; 0.045	0.490; 0.019	0.660; 0.055	0.277; 0.035	0.512; 0.036
$\sigma^2 = 500$					
$k = 2$	0.057; 0.141	0.534; 0.260	0.570; 0.238	-0.019; 0.038	0.146; 0.189
$k = 3$	0.244; 0.238	0.697; 0.129	0.721; 0.136	0.112; 0.113	0.241; 0.122
$k = 4$	0.108; 0.115	0.601; 0.115	0.615; 0.120	0.524; 0.170	0.332; 0.059
$k = 5$	0.042; 0.073	0.418; 0.085	0.447; 0.077	0.183; 0.135	0.296; 0.081
$k = 6$	0.027; 0.020	0.351; 0.062	0.374; 0.056	0.230; 0.056	0.177; 0.046
$k = 7$	0.078; 0.079	0.709; 0.087	0.773; 0.057	0.410; 0.080	0.542; 0.039
$k = 8$	0.133; 0.101	0.526; 0.050	0.599; 0.059	0.196; 0.046	0.509; 0.042
$k = 9$	0.019; 0.015	0.398; 0.053	0.579; 0.056	0.202; 0.046	0.434; 0.058
$k = 10$	0.160; 0.052	0.458; 0.033	0.575; 0.042	0.247; 0.051	0.448; 0.040

Table 3: Mean value μ and standard deviation sd of the ARI index for the spectral clustering method that uses the Gaussian similarity function.

	L_{rw}	L_{sym}	M	M_D
$\sigma^2 = 10$	μ; sd	μ; sd	μ; sd	μ; sd
$k = 2$	1; 0	1; 0	1; 0	1; 0
$k = 3$	1; 0	1; 0	1; 0	1; 0
$k = 4$	1; 0	1; 0	1; 0	1; 0
$k = 5$	0.998; 0.090	0.998; 0.009	0.998; 0.009	1; 0
$k = 6$	1; 0	1; 0	1; 0	1; 0
$k = 7$	1; 0	1; 0	1; 0	1; 0
$k = 8$	1; 0	1; 0	0.985; 0.040	1; 0
$k = 9$	1; 0	1; 0	1; 0	1; 0
$k = 10$	1; 0	1; 0	0.990; 0.029	0.998; 0.006
$\sigma^2 = 50$				
$k = 2$	1; 0	1; 0	1; 0	1; 0
$k = 3$	1; 0	1; 0	1; 0	1; 0
$k = 4$	1; 0	1; 0	1; 0	1; 0
$k = 5$	1; 0	1; 0	1; 0	1; 0
$k = 6$	0.962; 0.004	0.967; 0.043	0.895; 0.120	0.960; 0.038
$k = 7$	1; 0	1; 0	1; 0	0.998; 0.006
$k = 8$	0.952; 0.030	0.956; 0.035	0.850; 0.029	0.907; 0.066
$k = 9$	1; 0	1; 0	1; 0	1; 0
$k = 10$	0.957; 0.034	0.963; 0.036	0.875; 0.033	0.897; 0.049
$\sigma^2 = 100$				
$k = 2$	0.987; 0.050	0.980; 0.060	0.980; 0.060	0.973; 0.068
$k = 3$	1; 0	1; 0	1; 0	1; 0
$k = 4$	1; 0	1; 0	1; 0	1; 0
$k = 5$	0.970; 0.040	0.977; 0.040	0.966; 0.050	0.966; 0.051
$k = 6$	0.876; 0.068	0.883; 0.068	0.686; 0.084	0.868; 0.095
$k = 7$	0.997; 0.010	0.999; 0.006	0.989; 0.023	0.983; 0.023
$k = 8$	0.905; 0.050	0.913; 0.040	0.831; 0.043	0.831; 0.046
$k = 9$	0.998; 0.006	0.998; 0.006	0.996; 0.011	0.997; 0.008
$k = 10$	0.906; 0.040	0.910; 0.042	0.857; 0.031	0.848; 0.044
$\sigma^2 = 150$				
$k = 2$	0.974; 0.082	0.993; 0.036	0.967; 0.075	0.980; 0.060
$k = 3$	1; 0	1; 0	1; 0	1; 0
$k = 4$	0.998; 0.120	0.998; 0.120	0.998; 0.120	0.998; 0.120
$k = 5$	0.925; 0.072	0.940 0.054	0.904; 0.077	0.890; 0.088
$k = 6$	0.788; 0.085	0.790 0.085	0.645; 0.064	0.728; 0.115
$k = 7$	0.989; 0.018	0.989; 0.018	0.958; 0.031	0.954; 0.044
$k = 8$	0.882; 0.038	0.895; 0.038	0.830; 0.036	0.805; 0.053
$k = 9$	0.984; 0.023	0.986; 0.019	0.982; 0.022	0.962; 0.054
$k = 10$	0.855; 0.042	0.866; 0.034	0.820; 0.041	0.794; 0.043
$\sigma^2 = 250$				
$k = 2$	0.862; 0.124	0.87; 0.135	0.838; 0.149	0.866; 0.162
$k = 3$	1; 0	1; 0	1; 0	1; 0
$k = 4$	0.980; 0.390	0.984; 0.290	0.977; 0.030	0.982; 0.300
$k = 5$	0.789; 0.090	0.820; 0.090	0.741; 0.118	0.742; 0.120
$k = 6$	0.639; 0.089	0.656; 0.080	0.578; 0.083	0.586; 0.086
$k = 7$	0.942; 0.038	0.952; 0.031	0.891; 0.079	0.897; 0.053
$k = 8$	0.781; 0.054	0.790; 0.052	0.763; 0.053	0.758; 0.048
$k = 9$	0.958; 0.026	0.960; 0.025	0.935; 0.036	0.888; 0.068
$k = 10$	0.781; 0.042	0.802; 0.043	0.765; 0.042	0.731; 0.046
$\sigma^2 = 300$				
$k = 2$	0.845; 0.149	0.864; 0.147	0.751; 0.207	0.810; 0.221
$k = 3$	1; 0	1; 0	1; 0	1; 0
$k = 4$	0.978; 0.040	0.984; 0.330	0.975; 0.037	0.977; 0.360
$k = 5$	0.741; 0.105	0.764; 0.091	0.725; 0.114	0.722; 0.131
$k = 6$	0.577; 0.086	0.701; 0.087	0.531; 0.052	0.526; 0.062
$k = 7$	0.929; 0.051	0.941; 0.049	0.886; 0.064	0.878; 0.065
$k = 8$	0.742; 0.037	0.759; 0.038	0.723; 0.062	0.735; 0.047
$k = 9$	0.942; 0.032	0.951; 0.032	0.924; 0.054	0.878; 0.074
$k = 10$	0.759; 0.041	0.776; 0.048	0.733; 0.047	0.698; 0.048
$\sigma^2 = 500$				
$k = 2$	0.577; 0.213	0.588; 0.213	0.542; 0.236	0.544; 0.282
$k = 3$	0.980; 0.047	0.983; 0.045	0.977; 0.054	0.980; 0.047
$k = 4$	0.860; 0.070	0.875; 0.790	0.865; 0.076	0.878; 0.760
$k = 5$	0.519; 0.100	0.539; 0.103	0.512; 0.117	0.528; 0.114
$k = 6$	0.460; 0.069	0.469; 0.062	0.458; 0.065	0.446; 0.066
$k = 7$	0.855; 0.046	0.860; 0.050	0.795; 0.049	0.786; 0.052
$k = 8$	0.613; 0.057	0.624; 0.054	0.631; 0.043	0.637; 0.046
$k = 9$	0.880; 0.043	0.886; 0.042	0.835; 0.051	0.779; 0.058
$k = 10$	0.678; 0.056	0.684; 0.051	0.664; 0.038	0.633; 0.031

Table 4: Comparison of mean values and standard deviations of the ARI for the SCCC (spectral curve clustering method with cosine similarity) and the LRM (linear regression mixture model for curve clustering) method. For the alternative approach, mean values and standard deviations of the ARI for the SCG (spectral clustering method with Gaussian similarity) and the $K-$MEANS method are compared.

	SCCC	LRM	SCG	K-MEANS
$\sigma^2 = 10$	μ; sd	μ; sd	μ; sd	μ; sd
$k = 2$	1; 0	1; 0	1; 0	1; 0
$k = 3$	1; 0	0.526; 0.037	1; 0	1; 0
$k = 4$	1; 0	0.664; 0.057	1; 0	1; 0
$k = 5$	0.886; 0.069	0.805; 0.134	1; 0	0.992; 0.042
$k = 6$	0.771; 0.024	0.455; 0.047	1; 0	1; 0
$k = 7$	0.971; 0.013	0.680; 0.094	1; 0	1; 0
$k = 8$	0.827; 0.001	0.694; 0.096	1; 0	1; 0
$k = 9$	0.843; 0.006	0.633; 0.044	1; 0	1; 0
$k = 10$	0.848; 0.025	0; 0	1; 0	1; 0
$\sigma^2 = 50$				
$k = 2$	1; 0	0.967; 0.088	1; 0	1; 0
$k = 3$	1; 0	0.313; 0.242	1; 0	1; 0
$k = 4$	0.910; 0.060	0.483; 0.086	1; 0	1; 0
$k = 5$	0.841; 0.054	0.700; 0.168	1; 0	1; 0
$k = 6$	0.771; 0.024	0.258; 0.061	0.967; 0.043	0.974; 0.035
$k = 7$	0.976; 0.016	0.673; 0.096	1; 0	1; 0
$k = 8$	0.822; 0.010	0.637; 0.071	0.956; 0.035	0.950; 0.035
$k = 9$	0.830; 0.016	0.559; 0.042	1; 0	1; 0
$k = 10$	0.823; 0.090	0; 0	0.963; 0.036	0.961; 0.026
$\sigma^2 = 100$				
$k = 2$	0.980; 0.060	0.791; 0.207	0.987; 0.050	0.980; 0.060
$k = 3$	0.993; 0.025	0.213; 0.232	1; 0	1; 0
$k = 4$	0.954; 0.112	0.370; 0.065	1; 0	1; 0
$k = 5$	0.776; 0.092	0.262; 0.285	0.977; 0.040	0.976; 0.036
$k = 6$	0.606; 0.072;	0.168; 0.035	0.883; 0.068	0.881; 0.071
$k = 7$	0.972; 0.028	0.655; 0.094	0.999; 0.006	0.997; 0.010
$k = 8$	0.805; 0.022	0.521; 0.112	0.913; 0.040	0.912; 0.041
$k = 9$	0.800; 0.032	0.509; 0.043	0.998; 0.006	0.998; 0.006
$k = 10$	0.801; 0.035	0; 0	0.910; 0.042	0.905; 0.038
$\sigma^2 = 150$				
$k = 2$	0.929; 0.127	0.741; 0.171	0.993; 0.036	0.974; 0.082
$k = 3$	0.971; 0.060	0.024; 0.090	1; 0	1; 0
$k = 4$	0.893; 0.097	0.342; 0.078	0.998; 0.120	0.995; 0.170
$k = 5$	0.741; 0.087	0.198; 0.248	0.940; 0.054	0.937; 0.045
$k = 6$	0.545; 0.050	0.104; 0.035	0.790; 0.085	0.793; 0.088
$k = 7$	0.960; 0.031	0.646; 0.087	0.989; 0.018	0.989; 0.018
$k = 8$	0.796; 0.037	0.392; 0.178	0.895; 0.038	0.893; 0.050
$k = 9$	0.759; 0.042	0.470; 0.042	0.986; 0.019	0.863; 0.040
$k = 10$	0.763; 0.050	0; 0	0.866; 0.034	0.865; 0.048
$\sigma^2 = 250$				
$k = 2$	0.813; 0.149	0.482; 0.250	0.870; 0.135	0.820; 0.160
$k = 3$	0.889; 0.097	0.020; 0.097	1; 0	1; 0
$k = 4$	0.840; 0.097	0.214; 0.059	0.984; 0.290	0.975; 0.040
$k = 5$	0.628; 0.090	0.059; 0.155	0.820; 0.090	0.811; 0.079
$k = 6$	0.487; 0.048	0.070; 0.025	0.656; 0.080	0.656; 0.091
$k = 7$	0.897; 0.055	0.595; 0.036	0.952; 0.031	0.948; 0.031
$k = 8$	0.726; 0.046	0.217; 0.195	0.790; 0.052	0.809; 0.049
$k = 9$	0.684; 0.050	0.399; 0.040	0.960; 0.025	0.956; 0.026
$k = 10$	0.682; 0.050	0; 0	0.802; 0.043	0.792; 0.035
$\sigma^2 = 300$				
$k = 2$	0.767; 0.180	0.414; 0.230	0.864; 0.147	0.851; 0.151
$k = 3$	0.898; 0.093	0.021; 0.060	1; 0	1; 0
$k = 4$	0.789; 0.104	0.234; 0.079	0.984; 0.330	0.982; 0.034
$k = 5$	0.607; 0.077	0.074; 0.154	0.764; 0.091	0.756; 0.073
$k = 6$	0.456; 0.047	0.066; 0.032	0.701; 0.087	0.611; 0.084
$k = 7$	0.884; 0.063	0.501; 0.178	0.941; 0.049	0.929; 0.051
$k = 8$	0.690; 0.055	0.143; 0.177	0.759; 0.038	0.767; 0.048
$k = 9$	0.668; 0.065	0.342; 0.099	0.951; 0.032	0.947; 0.035
$k = 10$	0.660; 0.055	0; 0	0.776; 0.048	0.771; 0.004
$\sigma^2 = 500$				
$k = 2$	0.570; 0.238	0.328; 0.192	0.588; 0.213	0.565; 0.236
$k = 3$	0.721; 0.136	0.008 0.041	0.983; 0.045	0.974; 0.061
$k = 4$	0.615; 0.120	0.141; 0.094	0.878; 0.760	0.868; 0.077
$k = 5$	0.447; 0.072	0.008; 0.041	0.539; 0.103	0.569; 0.092
$k = 6$	0.374; 0.056	0.029; 0.028	0.469; 0.062	0.471; 0.067
$k = 7$	0.773; 0.057	0.309; 0.224	0.860; 0.050	0.852; 0.047
$k = 8$	0.599; 0.059	0.089; 0.128	0.637; 0.046	0.671; 0.053
$k = 9$	0.579; 0.056	0.282; 0.084	0.886; 0.042	0.885; 0.040
$k = 10$	0.575; 0.042	0; 0	0.684; 0.051	0.677; 0.047

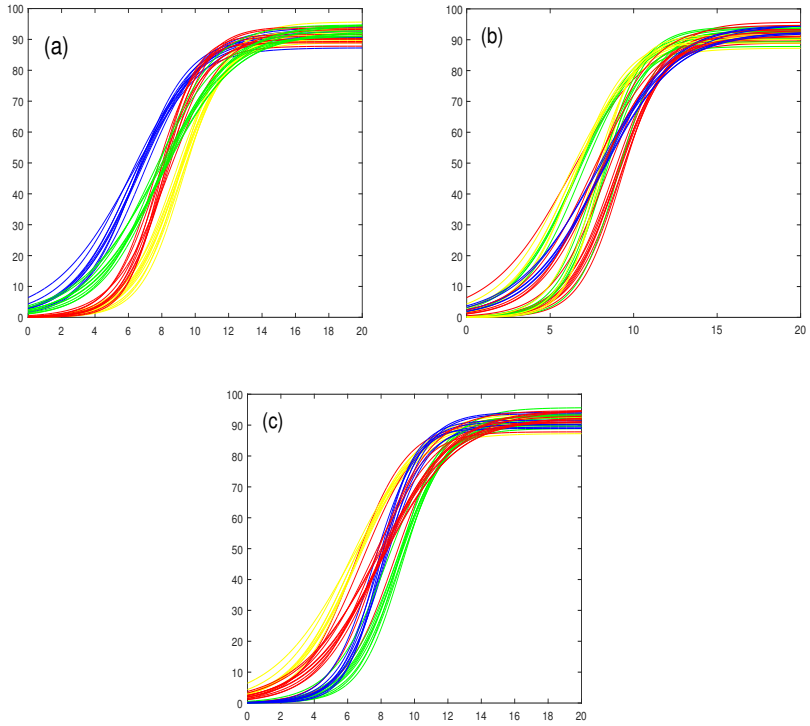

Fig. 2: Partition of sets of curves generated from the four logistic curve templates with mean 0 and variance 300. Figure (a) shows the ideal partition obtained with the spectral method that uses $\mathbf{L}_{sym}$ and the Gaussian similarity function with the Manhattan norm and $\gamma = 100$. The same ideal partition is obtained by using the $k-$means algorithm. For both methods, curves are represented as 2001-dimensional vectors. Figure (b) shows a curve partition obtained by using linear regression mixtures, while Figure (c) shows a curve partition obtained by the spectral method that uses cosine similarity. One color is associated to each cluster.

4.2 A real-world problem

A set of 60 generalized logistic growth curves (Fig. 3a) is generated on the basis of measurement of weights for 60 pigs (30 barrows and 30 gilts) in the interval between the age of 49 and 215 days (see Vincek et al. 2012). Since the weight of pigs grows differently for barrows and gilts, it is reasonable to expect the existance of two clusters in the set of these growth curves, one for barrows and one for gilts. The weight of barrows increases faster than the weight of gilts. Therefore, in order to find a proper $2-$partition of the set of growth curves, we apply spectral clustering methods that use cosine similarity for all five matrices

Table 5: The adjusted Rand index for all types of considered matrices for the set of growth curves.

	L	L_{rw}	L_{sym}	M	M_D
ARI	0.006801	0.6892	0.746819	0.0386	0.633821

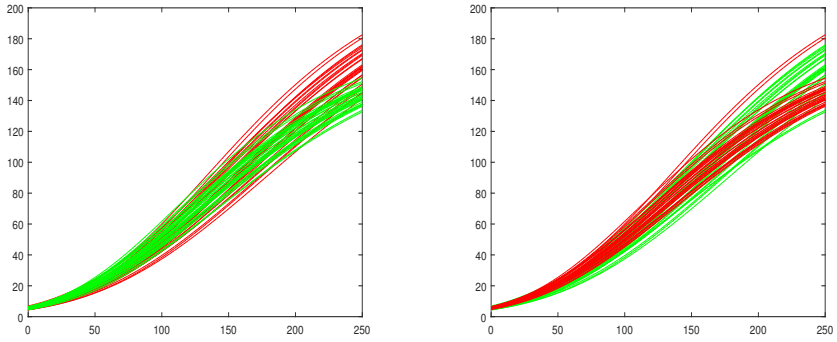

Fig. 3: 2-clustering of growth curves. On the left, the ideal partition is presented where red and green curves correspond to growth curves of barrows and growth curves of gilts, respectively. On the right, the partition obtained by using spectral methods with cosine similarity and $\mathbf{L}_{sym}$ is presented. One color is associated to each cluster.

described in Section 2.2. The values of the Adjusted Rand Index between the estimated partition and a physical classification (barrows and gilts) are given in Table 5. Figure 3 shows the ideal 2−partition of growth curves on the left, and a 2−partition of growth curves obtained by using matrix $\mathbf{L}_{sym}$. As in synthetic data, matrix $\mathbf{L}_{sym}$ gives the best results. However, curves are very close to each other so it is not surprising that the ARI index is not very high.

5 Conclusion

In this paper, we study a problem of functional data clustering where functions are growth curves. On a given set of growth curves we apply spectral clustering techniques which require a construction of a similarity graph and solving a relaxed version of the minimum cut problem or the Newman-Girvan modularity maximization problem. These methods are nonparametric and they consist of choosing a new point-based representation of curves that allows the usage of some standard clustering algorithms such as $k-$means.

To demonstrate the performance of the spectral approach to growth curve clustering, we generate synthetic data sets and compare our results with other curve clustering methods such as polynomial regression mixtures or the usual

$k-$means method. Results indicate that spectral methods show better performance than other methods, i.e. they are more accurate and faster.

We firmly believe that the spectral approach to functional data clustering deserves further work, for example, considering other types of curves, developing other measures to calculate similarity between curves and using other types of similarity graphs.

Appendix A Adjusted Rand Index

Let S be the set of n data items, $S = \{x_1, x_2, \ldots, x_n\}$, and let $U = \{U_1, U_2, \ldots, U_k\}$ and $V = \{V_1, V_2, \ldots, V_r\}$ be two partitions of S. The measure of similarity between partitions U and V is based on counting the pairs of data items that are in the same/different partition sets in U and V. Each pair (x_i, x_j) of data items is classified into one of four groups based on their comembership in U and V, which results in the pair-counts derived using the contingency table. The contingency table is a $k \times r$ matrix of all possible overlaps between each pair of clusters U and V, where its ijth element shows the intersection of cluster U_i and V_j, that is, $n_{ij} = |U_i \cap V_j|$.

	V_1	V_2	$\ldots$	V_r	marginal sums
U_1	n_{11}	n_{12}	$\ldots$	n_{1r}	$n_{1.}$
U_2	n_{21}	n_{22}	$\ldots$	n_{2r}	$n_{2.}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
U_k	n_{k1}	n_{k2}	$\ldots$	n_{kr}	$n_{k.}$
marginal sums	$n_{.1}$	$n_{.2}$	$\ldots$	$n_{.r}$	n

In the case of disjoint clusters we have $n_{i.} = |U_i|$ and $n_{.j} = |V_j|$. Let us define the following numbers:
a - the number of pairs of elements in S that are in the same set in U and in the same set in V;
b - the number of pairs of elements in S that are in different sets in U and in different sets in V;
c - the number of pairs of elements in S that are in the same set in U and in different sets in V;
d - the number of pairs of elements in S that are in different sets in U and in the same set in V.

The Rand index is a measure of similarity between U and V defined as

$$RI = \frac{a+b}{a+b+c+d} = \frac{a+b}{\binom{n}{2}}.$$

The numbers a, b, c, d can be easily obtained by using simple combinatorial methods.
The Rand index has a fixed range of $[0, 1]$. A problem with this type of similarity measure is that the expected value of this index of two random partitions

does not take a constant value (say zero). The adjusted Rand index ARI is proposed in Hubert & Arabie (1985) assuming that the contingency table is constructed randomly when the size of the clusters in U and V is fixed. The ARI is calculated based on an upper bound 1 on the RI and its expected value

$$ARI = \frac{\sum_{i=1}^{k}\sum_{j=1}^{r}\binom{n_{ij}}{2} - \sum_{i=1}^{k}\binom{n_{i.}}{2}\sum_{j=1}^{r}\binom{n_{.j}}{2}/\binom{n}{2}}{\frac{1}{2}\left[\sum_{i=1}^{k}\binom{n_{i.}}{2} + \sum_{j=1}^{r}\binom{n_{.j}}{2}\right] - \sum_{i=1}^{k}\binom{n_{i.}}{2}\sum_{j=1}^{r}\binom{n_{.j}}{2}/\binom{n}{2}}.$$

It is the normalized difference of the Rand Index and its expected value under the null hypothesis (Wagner & Wagner 2007).

References

Bezdek JC (1981) Pattern Recognition with Fuzzy Objective Function Algorithms. Kluwer Academic Publishers Norwell

Bolla M (2011) Penalized versions of the Newman-Girvan modularity and their relation to normalized cuts and k-means clustering. Physical Review E 84:016108

Chamroukhi F (2016) Piecewise Regression Mixture for Simultaneous Functional Data Clustering and Optimal Segmentation. Journal of Classification 33:374–411

Ćwik M, Józefczyk J (2017) Heuristic algorithms for the minmax regret flow-shop problem with interval processing times. Central European Journal of Operations Research 1–24

Diestel R (2000) Graph theory. Electronic Edition 2000, Springer - Verlag, New York

Duda RO, Hart PE, Stork DG (2001) Pattern Classification, (2nd ed.). Wiley

Gaffney S, Probabilistic Curve-Aligned Clustering and Prediction with Mixture Models. PhD thesis, Department of Computer Science, University of California, Irvine, USA (2004)

Gan G, Wu J, Ma C, Data Clustering: Theory, Algorithms, and Applications, SIAM, Philadelphia (2007)

Golub GH, Van Loan CF, Matrix Computations, Baltimore: Johns Hopkins University Press, p. 320 (1996)

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

Jacques J, Preda C (2014) Functional data clustering: a survey. Advances in Data Analysis and Classification 8:231–255

Jain AK (2010) Data clustering: 50 years beyond k-means. Pattern Recognition Letters 31:651–666

Janković M, Leko A, Šuvak N (2016) Application of lactation models on dairy cow farms. Croatian Operational Research Review 7:217–227

Jukić D, Scitovski R (2003) Solution of the least-squares problem for logistic function. Journal of Computational and Applied Mathematics 156:159–177

Kogan J (2007) Introduction to Clustering Large and High-Dimensional Data. Cambridge University Press

Luxburg U (2007) A tutorial on spectral clustering. Statistics and Computing 17:395–416

Majstorović S, Stevanović D (2014) A note on graphs whose largest eigenvalues of the modularity matrix equals zero. Electronic Journal of Linear Algebra 27:611–618

Marošević T, Sabo K, Taler P (2013) A mathematical model for uniform distribution of voters per constituencies. Croatian Operational Research Review 4:63–64

Newman, MEJ, Girvan M (2004) Finding and evaluating community structure in networks. Physical Review E 69:026113

Ng AY, Jordan MI, Weiss Y (2001) On Spectral Clustering: Analysis and an Algorithm. Advances in Neural Information Processing Systems MIT Press 14:849–856

Park J, Ahn J (2017) Clustering multivariate functional data with phase variation. Biometrics 73:324–333

Ratkowsky DA (1990) Handbook of nonlinear regression models. Marcel Dekker, New York
Sabo K (2014) Center-based l_1-clustering method. International Journal of Applied Mathematics and Computer Science 24:151–163.
Sangalli LM, Secchi P, Vantini S, Veneziani A (2009) A Case Study in Exploratory Functional Data Analysis: Geometrical Features of the Internal Carotid Artery. Journal of the American Statistical Association 104:37–48
Sangalli LM, Secchi P, Vantini S, Vitelli V (2010) K-mean alignment for curve clustering. Computational Statistics and Data Analysis 54: 1219–1233
Scitovski R, Scitovski S (2013) A fast partitioning algorithm and its application to earthquake investigation. Computers & Geosciences 59:124–131
Su Z, Kogan J, Nicholas C (2010) Constrained clustering with k-means type algorithms. In M. W. Berry, J. Kogan (Eds.), Text Mining Applications and Theory, pp 81-103, Wiley, Chichester
Tarpey T, Kinateder KKJ (2003) Clustering Functional Data. Journal of Classification 20:93–114
Turkalj Ž, Markulak D, Singer S, Scitovski R (2016) Research project grouping and ranking by using adaptive Mahalanobis clustering. Croatian Operational Research Review 7:81–96
Vincek D, Kralik G, Kušec G, Sabo K, Scitovski R (2012) Application of growth functions in the prediction of live weight of domestic animals. Central European Journal of Operations Research 20:719–733
Vincek D, Sabo K, Kušec G, Kralik G, Djurkin I, Scitovski R (2012) Modeling of pig growth by S-function - least absolute deviation approach for parameter estimation. Archiv für Tierzucht 55:364–374
Wagner S, Wagner D, Comparing Clusterings. http://i11www.iti.uni-karlsruhe.de/extra/publications/ww-cco-06.pdf. Accessed 2007
Zhang Z, Pati D, Srivastava A (2015) Bayesian clustering of shapes of curves. Journal of Statistical Planning and Inference 166:171–186