Int. J. Appl. Math. Comput. Sci., 2020, Vol. 30, No. 4, 703–715
DOI: 10.34768/amcs-2020-0052

A FEASIBLE K–MEANS KERNEL TRICK UNDER NON–EUCLIDEAN FEATURE SPACE

Robert KŁOPOTEK [a], Mieczysław KŁOPOTEK [b,*], Sławomir WIERZCHOŃ [b]

[a]Faculty of Mathematics and Natural Sciences
Cardinal Stefan Wyszyński University in Warsaw
ul. Wóycickiego 1/3, 01-938 Warsaw, Poland
e-mail: r.klopotek@uksw.edu.pl

[b]Institute of Computer Science
Polish Academy of Sciences
ul. Jana Kazimierza 5, 01-248 Warsaw, Poland
email: {klopotek,stw}@ipipan.waw.pl

This paper poses the question of whether or not the usage of the kernel trick is justified. We investigate it for the special case of its usage in the kernel k-means algorithm. Kernel-k-means is a clustering algorithm, allowing clustering data in a similar way to k-means when an embedding of data points into Euclidean space is not provided and instead a matrix of "distances" (dissimilarities) or similarities is available. The kernel trick allows us to by-pass the need of finding an embedding into Euclidean space. We show that the algorithm returns wrong results if the embedding actually does not exist. This means that the embedding must be found prior to the usage of the algorithm. If it is found, then the kernel trick is pointless. If it is not found, the distance matrix needs to be repaired. But the reparation methods require the construction of an embedding, which first makes the kernel trick pointless, because it is not needed, and second, the kernel-k-means may return different clusterings prior to repairing and after repairing so that the value of the clustering is questioned. In the paper, we identify a distance repairing method that produces the same clustering prior to its application and afterwards and does not need to be performed explicitly, so that the embedding does not need to be constructed explicitly. This renders the kernel trick applicable for kernel-k-means.

Keywords: kernel methods, k-means, clustering, non-Euclidean feature space, Gower/Legendre theorem.

.

1. Introduction

A fundamental requirement for the evaluation of the quality of an algorithmic method is that it returns what it promises. This does not need to be the case with the quite common kernel trick method, widely used in conjunction with classification and clustering methods, including SVM (Shawe-Taylor and Cristianini, 2004, Chap.7) and kernel-k-means (Wierzchoń and Kłopotek, 2018). The so-called kernel methods have been invented to allow for the application of Euclidean embedding based algorithms to other data structure representations. There exist domains in which not an embedding, but rather a similarity matrix (Demontis *et al.*, 2017) or the distance matrix (which should be rather called the dissimilarity matrix) (Jacobs *et al.*, 2000; Jain and Zongker, 1998; Kleinberg, 2002) are available. In such cases, the dissimilarity matrix can be transformed to a similarity matrix (Demontis *et al.*, 2017; Gower, 1982; Cox and Cox, 2001; Kłopotek, 2019) and then the embedding can be found via eigendecomposition if it exists (that is, there are no negative eigenvalues (Pękalska and Duin, 2005)). If there is no such embedding, then the distance/similarity matrix can be repaired using various techniques (Demontis *et al.*, 2017; Gower and Legendre, 1986; Lingoes, 1971; Cailliez, 1983; Higham, 1988), which are also based on eigendecomposition. To by-pass the need for eigendecomposition, the so-called kernel-trick is applied which works directly

[*] Corresponding author

on the similarity matrix, as described in many popular publications; see, e.g., the works of Shawe-Taylor and Cristianini (2004) for kernel-k-means.

But some serious problems with this procedure emerge if embeddability is not granted (e.g., due to measurement or computational or typing errors (see Higham, 1988), due to a specific definition of distance, e.g., between time series (Marteau, 2019), or due to the treatment of missing values, or the nature of the distances themselves (Legendre and Legendre, 1998)) and other approaches need to be used (Schleif and Tino, 2015; Villmann *et al*., 2016; Higham, 1988; Marteau, 2019). In such a case, the kernel-k-means does not optimize the k-means cost function when applied to the dissimilarity matrix that is not Euclidean, or equivalently, to the similarity matrix that is not positive semidefinite (Section 3). This means that the eigendecomposition of the similarity matrix needs to be performed prior to the kernel trick application. This makes the kernel trick pointless. Furthermore, most of the aforementioned methods of repairing the distance/similarity matrix, besides relying on eigen decomposition, have the flaw that if the repair transformation is performed, the clustering resulting from the usage of kernel-k-means prior to the transformation may be different from that after the transformation so that we do not really know which one to trust (unless they are identical, see Section 8). Last but not least, the transformation method described by Gower and Legendre (1986) and reproduced in a number of later publications (e.g., Szekely and Rizzo, 2014; Qi, 2016) is actually inaccurate.[1]

Therefore, we contribute the following in this paper: (i) we show that a non-Euclidean distance matrix leads to wrong clustering by kernel-k-means (Section 3); (ii) we show that the distance matrix corrections proposed in (Gower and Legendre, 1986) (recalled in Section 4) do not repair the matrix to an Euclidean one (Sections 5 and 7); (iii) we show that the original theorems of Lingoes (1971) and Cailliez (1983), on which the method of Gower and Legendre (1986) was based, are correct (Sections 6 and 8); (iv) we show that the Cailliez (1983) correction is not suitable for the classical kernel-k-means as there are clustering discrepancies—it may be applied only for its variants rooted in ℓ_1 like the k-median algorithm (Bradley *et al*., 1996; Du *et al*., 2015; Kashima *et al*., 2008) (Section 6); and (v) we show under what assumptions the correction of Lingoes (1971) is suitable for usage with kernel-k-means as the clusterings before and after this transformation agree and hence the kernel trick can be validly applied, i.e., without the prior checking for embeddability of the distance matrix (Section 8).

2. Background

Apparently, k-means is a broadly used clustering algorithm. It works efficiently for data embedded into a fixed-dimensional Euclidean space. Its numerous properties have been widely studied. In spite of the high worst-case complexity, it is generally very quick and some good properties have been established, like its probabilistic k-richness (Ackerman *et al*., 2010), meaning that with high probability the intrinsic cluster structure may be recovered by k-means for favorable distances.

The k-means clustering algorithm seeks to split data points $\mathbf{x}_i$ into k clusters C_j, $j = 1, \dots, k$ by finding k points $\boldsymbol{\mu}_j$, $j = 1, \dots, k$, called cluster centers, or prototypes, in the data space such that the cost function

$$J(\{\boldsymbol{\mu}_j\}_{j=1}^{k}) = \sum_{i=1}^{m} \min_{1\le j\le k} \|\mathbf{x}_i - \boldsymbol{\mu}_j\|^2 \tag{1}$$

is minimized. The cluster C_j consists then of data points $\mathbf{x}_i$ such that $j = \arg\min_{1\le j\le k} \|\mathbf{x}_i - \boldsymbol{\mu}_j\|^2$. Note that in the optimal clustering,

$$\boldsymbol{\mu}_j = \frac{1}{m_j} \sum_{i\in C_j} \mathbf{x}_i \tag{2}$$

holds, due to properties of Euclidean spaces. The k-means algorithm has the very attractive property of being easy to implement, and there exist various variants of it like k-means++ possessing even closeness-to-optimum properties. A drawback of this algorithm is that it accepts numeric attributes only and requires an embedding in a Euclidean space. Embedding into other spaces were investigated, like hyperbolic space, but the computation of cluster centers which is vital and very easy in Euclidean spaces, is not so easy in the other spaces. However, real-world objects are frequently described by non-numeric attributes, or are not embedded in any space whatsoever and instead only similarity, dissimilarity or distance between objects is known. In such cases, the kernel-k-means clustering algorithm can be used which at least partially inherits the good properties of k-means. These applications usually rely on the kernel trick. Kernel-trick based k-means algorithms are applied in various areas (e.g., gene expression clustering (Handhayania and Hiryantob, 2015) or spectral clustering of graphs (Shawe-Taylor and Cristianini, 2004)).[2]

A kernel is understood as a function $\kappa : X \times X \to \mathbb{R}$ mapping the dot product of some representation

[1] Compare the original formulas derived by Cailliez (1983) and Lingoes (1971) and reproduced correctly later by Legendre and Legendre (1998) as well as Cox and Cox (2001). Note, however, that Legendre and Legendre (1998) refer on page 433 to the paper by Gower and Legendre (1986) notifying the reader that the form provided by Gower and Legendre (1986) is misprinted, which must have gone unnoticed by Szekely and Rizzo (2014) as well as (Qi, 2016).

[2] An overview of the kernel k-means algorithm may be found in, e.g., the works of Shawe-Taylor and Cristianini (2004) or Wierzchoń and Kłopotek (2018).

space X into a subset of real numbers, whereby satisfying, for all $x, x' \in X$, $\kappa(x, x') = \langle \Phi(x), \Phi(x') \rangle$, where Φ maps X into some dot product space H, sometimes called the feature space (Hofmann *et al.*, 2008). A dot product space is a vector space (with, e.g., real-valued or complex-valued coordinates) together with a dot product operator. The dot product operator $\langle \cdot, \cdot \rangle$ has to have, for all vectors $\mathbf{x}, \mathbf{y}, \mathbf{z}$ and all scalars a the properties: (i) conjugate symmetry, $\langle \mathbf{x}, \mathbf{y} \rangle = \langle \mathbf{y}, \mathbf{x} \rangle$; (ii) linearity in the first argument, $\langle a\mathbf{x}, \mathbf{y} \rangle = a\langle \mathbf{x}, \mathbf{y} \rangle$, $\langle \mathbf{x} + \mathbf{y}, \mathbf{z} \rangle = \langle \mathbf{x}, \mathbf{z} \rangle + \langle \mathbf{y}, \mathbf{z} \rangle$; (iii) positive definiteness, $\langle \mathbf{x}, \mathbf{x} \rangle > 0$, for any non-zero vector $\mathbf{x}$. The kernel approach exploits the function $\Phi : X \to \mathbb{R}^m$ mapping the original representation space X to a high-dimensional *Euclidean* space $\mathbb{R}^m$ (Shawe-Taylor and Cristianini, 2004), i.e., it operates on Mercer kernels. The (embedding) transformation Φ is chosen in such a way that classes or clusters not linearly separable in the original representation space will become linearly separable in the feature space. Hence, a number of data mining methods requiring linear data separation can be applied to non-linearly separated data sets.

Instead of the mapping Φ, the aforementioned kernel-function κ is used. In practice, however, we are not interested in the whole space X, but rather in a (finite) sample $S \subset X$, for which the function κ can be summarized by the kernel matrix K with the property that $K_{ij} = \kappa(\mathbf{i}, \mathbf{j})$ for any two data objects $\mathbf{i}, \mathbf{j} \in S$. The matrix K can be considered as a kind of similarity matrix between the data points. If κ is a Mercer kernel, K is positive semidefinite. To recover the embedding, the eigenproblem has to be solved for K. Though the solution has been shown to be of polynomial complexity (Pan and Chen, 1999), it may still be prohibitive if the matrix is huge.

For a number of algorithms, including kernel-k-means, the so-called *kernel trick* has been elaborated, allowing to sidestep the solving of the eigenproblem. The essence of the kernel trick is that the kernel matrix K is sufficient for performing the algorithm in the feature space, and one does not need to know the Φ function, even for data points from S. Section 3 explains the usage of the kernel trick for the k-means algorithm. Please pay attention to the difference between formulas (4) and (1).

Nonetheless, the very existence of the mapping Φ, and hence of the kernel function κ is of vital importance for the validity of the application of the k-means algorithm in the feature space. Φ transforms the data to points in an Euclidean space so that k-means can be applied at all. In other words, the matrix K needs to be positive semidefinite. In many cases, like Laplacians of graphs (Wierzchoń and Kłopotek, 2018) or density-based regression (Jaworski, 2018), one knows in advance that they can be deemed as kernels embedded into an Euclidean space, so that there are no obstacles to apply kernel-k-means clustering. However, the kernel matrix may not be a Mercer kernel matrix. The validity of the kernel trick underpins various improvements of kernel k-means clustering, like single pass clustering (Sarma *et al.*, 2013), global kernel k-means (Tzortzis and Likas, 2009), subsampling kernel k-means (Chitta *et al.*, 2011), robust kernel k-means (Yao and Chen, 2018) and other. Therefore, research is performed like the one reported by Li *et al.* (2013), Roth *et al.* (2003) and Marin *et al.* (2019) to transform a similarity matrix into the closest proper positive semidefinite kernel matrix.

This issue is closely connected to the other mentioned data representations. As mentioned, instead of the kernel matrix, the distance matrix D may be available. This matrix can be easily transformed into a kernel matrix for usage with kernel-k-means. The most general proposal of a distance-to-kernel-matrix transform seems to be that defined in Theorem 2 of Gower (1982):

$$K = K_{\mathcal{G}}(D) = \left(\mathbf{I} - \mathbf{1}\mathbf{s}^T\right)\left(-\frac{1}{2}D^{\circ 2}\right)\left(\mathbf{I} - \mathbf{s}\mathbf{1}^T\right) \quad (3)$$

where $\mathbf{s}$ is a vector such that $\mathbf{s}^T\mathbf{1} = 1$, and $D^{\circ 2}$ is a matrix containing squared distances from D as entries. If all elements of $\mathbf{s}$ are equal to $1/m$, then $\left(\mathbf{I} - \mathbf{1}\mathbf{s}^T\right)$ is called a centering matrix. It generalizes other proposals like that of Schoenberg (1938), recalled by Balaji and Bapat (2007) as well as Cox and Cox (2001). If the distance matrix D is Euclidean, then the kernel matrix K is positive semidefinite. But what if D is not Euclidean? As shown in Section 3, kernel-k-means applied to the respective kernel matrix K will not be able to reach the optimum (minimum) of the k-means cost function. Does it mean that we cannot legitimately apply kernel-k-means (with kernel trick) to distance matrix D without checking *a priori* whether or not it is Euclidean? But if it is so, then there is no point in applying the kernel trick at all. The advantage of the kernel trick was to save the costs of seeking eigenvalues and eigenvectors of K. But if we have to check the eigenvalues for non-negativity, then (i) this advantage is lost, and (ii) the distance matrix needs to be "repaired" to represent the Euclidean distance.

Theorem 7 of Gower and Legendre (1986) proposes two ways of "repairing" a non-Euclidean distance[3] matrix to become a Euclidean one (cf. Section 4). Other proposals come from Lingoes (1971), Roth *et al.* (2003), Cailliez (1983), Szekely and Rizzo (2014), Qi (2016), and others. For an overview of other approaches, see also the work of Gisbrecht and Schleif (2015).

In the context of transforming non-Euclidean distances to Euclidean ones, the paper by Gower and Legendre (1986) is cited, e.g., by Gonzalez and Munoz

[3] The non-Euclidean distance d for a dataset X should have the properties: $d(x, x) = 0$, $d(x, y) = d(y, x)$, $d(x, y) > 0$ for $x, y \in X$, $x \neq y$.

(2010), Marin *et al.* (2019), Cox and Cox (2001) and Pękalska *et al.* (2002). For example, Pękalska *et al.* (2002) cite the theorem on page 179, though, as we prove below, this theorem is inaccurate. Marin *et al.* (2019), in the proof of their Theorem 3, cite the Gower/Legendre theorem and the results of Lingoes (1971) as if they were equivalent, which is not true. Gonzalez and Munoz (2010) just assume the Gower/Legendre theorem works.

We will concentrate on Theorem 7 of Gower and Legendre (1986). Therefore we will call it simply the G/L-theorem. We will demonstrate that the G/L-theorem is misprinted (Sections 5 and 7), i.e., the two methods suggested there do not turn a non-Euclidean to a Euclidean distance matrix. We prove that, instead, the original proposals (Cailliez, 1983; Lingoes, 1971) from which the G/L-theorem was constructed yield such a matrix transformation (Sections 5 and 7). However, the proposal of Cailliez (1983) is unsatisfactory because kernel-k-means applied to the original dissimilarity matrix and to the euclidized one yield different results (Section 6). The same applies to the formulas suggested by Szekely and Rizzo (2014) as well as Qi (2016) for the very same reason that is a modification of the distances by the same constant. Some researchers, e.g., Choi and Choi (2005) or Bao and Kadobayashi (2008), recommend explicit usage of the Cailliez euclidization.

It is only in the case of the transformation proposed by Lingoes (1971) that kernel-k-means applied to the original dissimilarity matrix and to the euclidized one yield identical results (Section 8). The reason is that this transformation adds the same constant to the squared distances which corresponds to the nature of k-means cost function (Eqns. (4) and (1)). Hence no identification of eigenvalues is needed, and the advantage of the kernel trick is preserved. This is the main result of this paper, and it is of crucial importance for the applicability of the kernel trick for kernel-k-means.

Many applications of kernel-k-means do not care at all about whether the kernel matrices are embeddable in Euclidean spaces. A non-Euclidean space requires serious modification of k-means, accommodating to that fact that the gravity center of a cluster cannot serve any more as a cluster center (gradient descent methods are needed, for example (see Richter *et al.*, 2017, Section 6)).

There also exist other publications in other domains referring to the misprinted formulation of Theorem 7, like that by Dokmanic *et al.* (2015, Theorem 2).

These facts underpin the need to demonstrate that there exist such ways of transforming the non-Euclidean distance matrix to a Euclidean one such that the results of kernel k-means for the original and the repaired matrices agree, what we actually do in this paper.

3. k-Means under non-Euclidean kernels

The kernel-k-means algorithm consists in switching to a multidimensional feature space $\mathcal{F}$. The method relies upon searching for prototypes $\boldsymbol{\mu}_j^\Phi$ minimizing the error or cost function J, (see, e.g., Shawe-Taylor and Cristianini, 2004), Section 8.2 defined as

$$J(\{\boldsymbol{\mu}_j^\Phi\}_{j=1}^k) = \sum_{i=1}^m \min_{1\le j\le k} \|\Phi(i) - \boldsymbol{\mu}_j^\Phi\|^2 \tag{4}$$

over all possible choices of the set of cluster centers $\boldsymbol{\mu}_j^\Phi$, $j = 1, \dots, k$, on the analogy of (1). But the possible choices are limited. Here $\boldsymbol{\mu}_j^\Phi$ may only be equal to

$$\boldsymbol{\mu}_j^\Phi = \frac{1}{m_j} \sum_{i\in C_j} \Phi(i) \tag{5}$$

for some subset C_j of all the data points and no other vectors of cluster centers in the feature space are taken into account, by analogy to (2). If the feature space is Euclidean, it is guaranteed (Gower, 1982) that no other vector of cluster centers from the feature space will ever be considered as a cluster center, because the clustering will not be optimal. It is not so in the case of non-Euclidean feature spaces. Let us discuss the concerns for applying kernel-k-means in such situations and about the validity of the obtained clusters.

Kernel-k-means uses the so-called kernel-trick, eliminating the need to know the Φ function and the need to handle high-dimensional vectors that Φ may induce. The *kernel trick* relies on the equation

$$\begin{aligned}\|\Phi(i) - \boldsymbol{\mu}_j^\Phi\|^2 = k_{ii} - \frac{2}{m_j}\sum_{h\in C_j} k_{hi} \\ + \frac{1}{m_j^2}\sum_{r\in C_j}\sum_{s\in C_j} k_{rs},\end{aligned} \tag{6}$$

where $k_{ij} = \Phi(i)^T\Phi(j) = K(i,j)$. Hence

$$\begin{aligned} J(\{\boldsymbol{\mu}_j^\Phi\}_{j=1}^k) &= \sum_{i=1}^m \min_{1\le j\le k} \Big(k_{ii} - \frac{2}{m_j}\sum_{h\in C_j} k_{hi} \\ &\quad + \frac{1}{m_j^2}\sum_{r\in C_j}\sum_{s\in C_j} k_{rs}\Big) \\ &= \sum_{i=1}^m \min_{1\le j\le k} \Big(- \frac{2}{m_j}\sum_{h\in C_j} k_{hi} \\ &\quad + \frac{1}{m_j^2}\sum_{r\in C_j}\sum_{s\in C_j} k_{rs}\Big) + \sum_{i=1}^m k_{ii}, \end{aligned} \tag{7}$$

where $\sum_{i=1}^m k_{ii}$ is a constant in this optimization task. In this way, one can update the elements of clusters without explicitly determining the prototypes. A respective implementation of kernel-k-means is presented by Shawe-Taylor and Cristianini (2004, Algorithm 8.22, pp. 274–275). We use our own R implementation; see the package at the end of the paper.

Equation (5) and implicitly also (7) state that one uses only unweighted combinations of the Φ images of data points in the feature space to compute the candidate cluster centers under kernel-k-means. However, under the kernel mapping Φ, it is also possible to compute the squared distance of any data point i to other candidate prototypes, like those $\boldsymbol{\mu}^{\Phi}_{\mathbf{w}}(C)$ from the convex hull of C, defined by (8). Let $w_1, \dots, w_m$ be non-negative weights of data points $1, \dots, m$. Let C be a subset of $\{1, \dots, m\}$ such that $\sum_{i \in C} w_i \neq 0$. Define $\boldsymbol{\mu}^{\Phi}_{\mathbf{w}}(C)$ as a weighted center of the datapoints of C,

$$\boldsymbol{\mu}^{\Phi}_{\mathbf{w}}(C) = \frac{1}{\sum_{i \in C} w_i} \sum_{i \in C} w_i \Phi(i). \tag{8}$$

The distance from a data point in the feature space to such a candidate prototype amounts to

$$\begin{aligned}\|\Phi(i) - \boldsymbol{\mu}^{\Phi}_{\mathbf{w}}(C)\|^2 = k_{ii} - \frac{2}{\sum_{h \in C} w_h} \sum_{h \in C} w_h k_{hi} \\ + \frac{1}{(\sum_{h \in C} w_h)^2} \sum_{r \in C} \sum_{s \in C} w_r w_s k_{rs}.\end{aligned} \tag{9}$$

Consider the following example of a non-Euclidean distance matrix:

$${}_{nE}D = \begin{pmatrix} 0 & 12 & 24 & 24 & 48 & 48 \\ 12 & 0 & 48 & 48 & 24 & 48 \\ 24 & 48 & 0 & 48 & 48 & 24 \\ 24 & 48 & 48 & 0 & 24 & 12 \\ 48 & 24 & 48 & 24 & 0 & 48 \\ 48 & 48 & 24 & 12 & 48 & 0 \end{pmatrix},$$

and the corresponding kernel[4] matrix, computed according to (3) with $\mathbf{s} = 1/m$:

$${}_{nE}F = \begin{pmatrix} 384 & 456 & 276 & 96 & -588 & -624 \\ 456 & 672 & -444 & -624 & 420 & -480 \\ 276 & -444 & 744 & -588 & -408 & 420 \\ 96 & -624 & -588 & 384 & 276 & 456 \\ -588 & 420 & -408 & 276 & 744 & -444 \\ -624 & -480 & 420 & 456 & -444 & 672 \end{pmatrix}.$$

Kernel-k-means, with $k = 2$, produces a clustering $[1, 2, 1, 1, 2, 1]$ [5] with the total value of the cost function 1908, Eqn. (7). Other clusterings, around unweighted centers, would not be better.

Consider a different clustering, [1,1,1,2,2,2], where you choose weighted cluster centers of Eqn. (8) with weights [10,1,1,10,1,1], instead of the k-means cluster centers of Eqn. (5), and substitute them into Eqn. (4). Then the cost function will amount to 1692, which is below what kernel-k-means produces. Thus we have demonstrated by this example that the following results holds.

Theorem 1. *Kernel-k-means does not optimize the cost function $J(\boldsymbol{\mu}^{\Phi}_j; j = 1, \dots, k)$ of (4) for non-Euclidean kernel matrices, where the cluster assignment is driven by the condition of the closest cluster center.*

It is clearly a consequence of the non-suitability of k-means for non-Euclidean distances. Similar problems with SVM have been pointed out by Loosli *et al.* (2016). This theorem may seem not to be a dramatic discovery, but recall that many popular works do not warn the reader that kernel-k-means requires an (underlying) Euclidean distance matrix.[6]

4. Gower/Legendre formulation of the kernel-trick related theorem

Recall that a matrix $D \in \mathbb{R}^{m \times m}$ is a Euclidean distance matrix between points $1, \dots, m$ if and only if there exists a matrix $X \in \mathbb{R}^{m \times n}$ the rows of which $(\mathbf{x}_1^T, \dots, \mathbf{x}_m^T)$ are coordinate vectors of these points in the n-dimensional Euclidean space and

$$d_{ij} = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T (\mathbf{x}_i - \mathbf{x}_j)}. \tag{10}$$

Theorem 7 by Gower and Legendre seeks to transform a non-Euclidean distance matrix into a Euclidean one.

Theorem 2. (Gower and Legendre, 1986, Theorem 7)

Part (a) or the Lingoes (1971) part. *Any dissimilarity matrix D may be turned to a Euclidean distance matrix, by adding a constant σ to the squared distances: $d'(\mathfrak{z}, \mathfrak{y}) = \sqrt{d(\mathfrak{z}, \mathfrak{y})^2 + \sigma}$, where σ is such that $\sigma \geq -\lambda_m$, λ_m being the smallest eigenvalue of*

$$F_{\mathcal{L}}(D) = \Big(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{m}\Big)\Big(-\frac{1}{2} D^{\circ 2}\Big)\Big(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{m}\Big), \tag{11}$$

where $D^{\circ 2}$ is the matrix of squared values of elements of D, m is the number of rows/columns in D.

[4] The matrix ${}_{nE}F$ is actually not a Mercer kernel matrix, as it is not positive semidefinite. But we use this term by analogy to the usage of "non-Euclidean distances" for dissimilarities by, e.g., Gower (1985), in spite of the fact that they do not possess (Euclidean) distance properties.

[5] Clustering $[1, 2, 1, 1, 2, 1]$ means that the first, third, fourth and sixth elements belong to cluster 1, while second and fifth to cluster 2.

[6] Szalkai (2013) proceeds as if it were inessential that the space is not Euclidean.

Part (b) or the Cailliez part. *Define* $F_{\mathcal{C}}(D) = \left(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{m}\right)(-\frac{1}{2}D)\left(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{m}\right)$ *and*

$$M_{\mathcal{C}}(D) = \begin{pmatrix} 0 & 2F_{\mathcal{L}}(D) \\ -I & 2F_{\mathcal{C}}(D) \end{pmatrix}. \tag{12}$$

Let κ be any number greater than or equal to the largest eigenvalue of $M_{\mathcal{C}}(D)$. Then $d'(\mathfrak{z},\mathfrak{y}) = d(\mathfrak{z},\mathfrak{y}) + \kappa$ is Euclidean.

The significance of a theorem like this is the following: Each non-Euclidean distance matrix can be turned to an Euclidean one, though one has to compute the matrix eigenvalues, which is of polynomial complexity (Pan and Chen, 1999), but may be expensive for large matrices. But the question arises: What about the relationship between clusterings obtained via kernel-k-means from the original dissimilarity matrix and that after correction. If they are different, then the semantics of kernel-k-means results are questionable. If they are identical, then there is a tremendous advantage. First of all, the results of kernel-k-means on the original matrix mean those obtained via euclidization. Second, the euclidization itself does not need to be performed, because the results for the Euclidean distance are identical with those for the non-Euclidean one. Accordingly, the eigenvalues do not need to be determined, and we save computation while creating a semantics.

Regrettably, Gower and Legendre's Theorem 7 is actually *inaccurate* in both parts. We will demonstrate this by examples. As the old theorem of Gower and Legendre proved to be incompletely proven, the still older results of Lingoes and Cailliez need a careful review. In fact, a later publication by one co-author (Legendre and Legendre, 1998) points at misprints in the G/L-theorem, but some questions remain: (i) Are those misprints that bad? (ii) Why did some authors modify the formulas of the G/L-theorem, producing further inaccuracies? But we do not verify these results for the pure sake of reviewing, but rather to enable posing the question of whether or not they are suitable for the purposes of kernel-k-means.

5. Failure of the Gower/ Legendre euclidization theorem in the Cailliez part

Section 3 convinced us that the application of kernel-k-means to non-Euclidean distances may be disastrous. This fact strengthens the interest in the various ways of euclidizing a distance matrix.

Let us first investigate the Cailliez part of Theorem 2. The formulation by itself is not quite accurate, because the matrix $M_{\mathcal{C}}(D)$ defined by (12) has complex eigenvalues (it is not symmetric), so the concept of a maximum is ill-defined. But even when we take the largest real eigenvalue, the embeddability is not achieved.

Apparently, Gower and Legendre misrepresented Cailliez (1983). Let us note in passing that other authors also misrepresent this result, e.g., Qi (2016, page 2, $\hat{B}$ formula):

$$M_{\text{Qi}\mathcal{C}}(D) = \begin{pmatrix} 0 & 2F_{\mathcal{L}}(D) \\ -I & 4F_{\mathcal{C}}(D) \end{pmatrix}, \tag{13}$$

and Szekely and Rizzo (2014):

$$M_{\text{Szekely}\mathcal{C}}(D) = \begin{pmatrix} 0 & F_{\mathcal{L}}(D) \\ I & F_{\mathcal{C}}(D) \end{pmatrix}. \tag{14}$$

Let us illustrate this with an example. The matrix $F_{\mathcal{C}}({}_{nE}D) =_{nE} F_{\mathcal{C}}$ has the form

$$_{nE}F_{\mathcal{C}} = \\ \begin{pmatrix} 11.3 & 7.3 & 2.3 & -0.7 & -9.7 & -10.7 \\ 7.3 & 15.3 & -7.7 & -10.7 & 4.3 & -8.7 \\ 2.3 & -7.7 & 17.3 & -9.7 & -6.7 & 4.3 \\ -0.7 & -10.7 & -9.7 & 11.3 & 2.3 & 7.3 \\ -9.7 & 4.3 & -6.7 & 2.3 & 17.3 & -7.7 \\ -10.7 & -8.7 & 4.3 & 7.3 & -7.7 & 15.3 \end{pmatrix}.$$

The minimal κ from part (b) of Theorem 2 for $M_{\mathcal{C}}({}_{nE}D) =_{nE} M_{\mathcal{C}}$ amounts to $\kappa = 38.307$. Upon modifying the distance matrix according to part (b) of Theorem 2, we get the new kernel matrix which is not Euclidean, because its lowest eigenvalue is equal to -1076.146. Therefore, change $M_{\mathcal{C}}$ to the original notation by Cailliez (Cailliez, 1983):

$$M_{o\mathcal{C}}(D) = \begin{pmatrix} 0 & 2F_{\mathcal{L}}(D) \\ -I & -4F_{\mathcal{C}}(D) \end{pmatrix} \tag{15}$$

(factor -4 instead of 2 in the last row). Cailliez demonstrated that the transformation to an embeddable distance version may have the form

$$d'(\mathfrak{z},\mathfrak{y}) = d(\mathfrak{z},\mathfrak{y}) + \kappa$$

for the original $M_{o\mathcal{C}}$, where κ is a number greater than or equal to the largest eigenvalue of $M_{o\mathcal{C}}$ which is positive for non-Euclidean distances.

Only upon applying the original Cailliez formulation, Eqn. (15), yielding minimal $\kappa = 69.134$, we get a kernel matrix which is Euclidean, because its lowest eigenvalue is zero. *Regrettably, the kernel matrix $F_{\mathcal{C}}({}_{nE}D + \kappa(\mathbf{1}\mathbf{1}^T - \mathbf{I})) =_{nE} F_{c\mathcal{C}}$ implies a clustering [1, 1, 2, 2, 1, 2] with the total value of the cost function 19317.864, which is different from the original clustering,* $[1, 2, 1, 1, 2, 1]$, *obtained for the matrix ${}_{nE}F$* via kernel-k-means.

Qi (2016), though producing a Euclidean matrix, overshoots the distance correction, yielding the minimal number greater than or equal to the largest eigenvalue of (13) $\kappa = 118.309$, while Szekely's formula underestimates it, yielding the minimal number greater than or equal to the largest eigenvalue of (14) $\kappa = 67.948$, which results in a non-Euclidean distance matrix.

Though we have seen by example that the original Cailliez theorem works better, we still need a proof for the general case. It is sufficient to show that in fact the largest *real* eigenvalue of $M_{o\mathcal{C}}$ needs to be positive when the distance is non-Euclidean. Assume that λ is an eigenvalue of the matrix $M_{o\mathcal{C}}$ with the eigenvector $\mathbf{v}$ which can be decomposed into two parts of equal lengths $\mathbf{v} = (\mathbf{v}_L^T, \mathbf{v}_C^T)^T$. Therefore, due to (15),

$$\lambda \mathbf{v}_L = 2F_{\mathcal{L}}(D)\mathbf{v}_C 4$$

and

$$\lambda \mathbf{v}_C = -I\mathbf{v}_L - 4F_{\mathcal{C}}(D)\mathbf{v}_C.$$

Substitution of the former to the latter leads to

$$\lambda \mathbf{v}_C = -\lambda^{-1} 2F_{\mathcal{L}}(D)\mathbf{v}_C - 4F_{\mathcal{C}}(D)\mathbf{v}_C$$

(valid only if $\lambda \neq 0$). Upon multiplying with $-\lambda$, we get

$$-\lambda^2 \mathbf{v}_C = (2F_{\mathcal{L}}(D) + \lambda 4F_{\mathcal{C}}(D))\mathbf{v}_C. \tag{16}$$

Thus we may think of this equation as of an eigenvalue problem with some parameter β

$$-\lambda^2 \mathbf{v}_C = 2(F_{\mathcal{L}}(D) + \beta 2F_{\mathcal{C}}(D))\mathbf{v}_C$$

subject to the constraint $\beta = \lambda$. It can be rewritten as

$$\begin{aligned} -\lambda^2 \mathbf{v}_C &= \left(-\left(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{m}\right) D^{\circ 2} \left(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{m}\right) \right. \\ &\qquad \left. -2\beta \left(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{m}\right) D \left(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{m}\right) \right) \mathbf{v}_C, \\ -\lambda^2 \mathbf{v}_C &= \left(-\left(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{m}\right) (D^{\circ 2} + 2\beta D) \right. \\ &\qquad \left. \cdot \left(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{m}\right) \right) \mathbf{v}_C. \end{aligned} \tag{17}$$

If D is Euclidean, then $D' = D^{\circ \frac{1}{2}}$ (element-wise square root of D) has the property that it is also a matrix of Euclidean distances. If β (parameter) is positive, then $(D^{\circ 2} + 2\beta D'^{\circ 2})^{\circ \frac{1}{2}}$ is also a Euclidean distance matrix. The solution of the eigenvalue problem to the right will then generate non-negative eigenvalues. But apparently, $-\lambda^2$ is non-positive. So if D is Euclidean, then λ must be non-positive. Thus we have shown that, indeed, whenever the biggest eigenvalue of the matrix $M_{o\mathcal{C}}(D)$ is positive, D is non-Euclidean.

What remains to be shown is that the matrix D can be turned to Euclidean by adding a constant to each distance. For this purpose, let us return to (16). We claim that for any κ the following holds:

$$\begin{aligned} &-(\lambda - \kappa)^2 \mathbf{v}_C \\ &\quad = (2F_{\mathcal{L}}(D + \kappa\left(\mathbf{1}\mathbf{1}^T - \mathbf{I}\right)) \\ &\qquad + \lambda 4F_{\mathcal{C}}(D + \kappa\left(\mathbf{1}\mathbf{1}^T - \mathbf{I}\right)))\mathbf{v}_C, \end{aligned} \tag{18}$$

i.e., if we increase non-diagonal distances in D by κ, then the eigenproblem of Cailliez will have the same eigenvectors with associated eigenvalues $\lambda - \kappa$ for non-zero λ. This means that if we increase the distances by $\max(\lambda)$, then all the eigenvalues in the Cailliez eigenproblem will be non-positive, hence the distance matrix will be Euclidean, which completes the proof of the Cailliez claim (compare this with the original proof by Cailliez (1983)). Let us prove our claim. Recall (see Eqn.(17)) that

$$\begin{aligned} -\lambda^2 \mathbf{v}_C &= \left(-\left(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{m}\right) (D^{\circ 2} + 2\lambda D) \right. \\ &\qquad \left. \cdot \left(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{m}\right) \right) \mathbf{v}_C. \end{aligned} \tag{19}$$

After adding $(2\lambda\kappa - \kappa^2)\mathbf{v}_C$ to both the sides, we get

$$\begin{aligned} &-(\lambda - \kappa)^2 \mathbf{v}_C \\ &\quad = -\left(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{m}\right)\Big(D^{\circ 2} + 2\lambda D \\ &\qquad + (2\lambda\kappa - \kappa^2)\left(\mathbf{1}\mathbf{1}^T - \mathbf{I}\right)\Big)\left(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{m}\right)\mathbf{v}_C. \end{aligned}$$

Hence

$$\begin{aligned} &-(\lambda - \kappa)^2 \mathbf{v}_C \\ &\quad = -\left(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{m}\right) \Big(D^{\circ 2} + 2\kappa D + \kappa^2\left(\mathbf{1}\mathbf{1}^T - \mathbf{I}\right) \\ &\qquad + 2\lambda D - 2\kappa D + 2\lambda\kappa\left(\mathbf{1}\mathbf{1}^T - \mathbf{I}\right) \\ &\qquad - 2\kappa^2\left(\mathbf{1}\mathbf{1}^T - \mathbf{I}\right)\Big) \left(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{m}\right)\mathbf{v}_C. \end{aligned}$$

Obviously, $\left(D + \kappa\left(\mathbf{1}\mathbf{1}^T - \mathbf{I}\right)\right)^{\circ 2} = D^{\circ 2} + 2\kappa D + \kappa^2\left(\mathbf{1}\mathbf{1}^T - \mathbf{I}\right)$. Hence

$$\begin{aligned} &-(\lambda - \kappa)^2 \mathbf{v}_C \\ &\quad = \left(-\left(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{m}\right)\Big(\left(D + \kappa\left(\mathbf{1}\mathbf{1}^T - \mathbf{I}\right)\right)^{\circ 2} \right. \\ &\qquad + 2(\lambda - \kappa)\left(D + \kappa\left(\mathbf{1}\mathbf{1}^T - \mathbf{I}\right)\right)\Big) \\ &\qquad \left. \cdot \left(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{m}\right) \right)\mathbf{v}_C. \end{aligned}$$

But this is exactly (19), i.e., (16) after adding κ to off-diagonal elements of D and subtracting κ from λ. Thus we are done with the Cailliez theorem.

6. Deficiency of the Cailliez approach to euclidization

But, though the original theorem of Cailliez is correct, as we have seen with the numerical example, the transformation of the non-Euclidean matrix to a Euclidean one is useless from the point of view of kernel-k-means as the clustering is not preserved, i.e., the clustering obtained via kernel-k-means for the original matrix is not the same as the one produced after the transformation.

In the case of k-means, by adding a constant to the distance, and not to the squared distance, the contribution of each cluster element to the cost function will differ, and the "outlier points" (those more distant than the other ones) will contribute more, so that the structure of clusters may impact the cluster assignment after a distance change. If we add κ in a cluster C_j for an element i to all its distances, then its contribution will increase to

$$\begin{aligned}\frac{1}{2m_j}&\sum_{l\in C_j, l\neq i}(\|\Phi(i)-\Phi(l)\|+\kappa)^2\\ &=\kappa^2\frac{m_j-1}{2m_j}+2\kappa\frac{1}{2m_j}\sum_{l\in C_j, l\neq i}\|\Phi(i)-\Phi(l)\|\\ &\quad+\frac{1}{2m_j}\sum_{l\in C_j}\|\Phi(i)-\Phi(l)\|^2.\end{aligned}$$

This means an increase by

$$\kappa^2\frac{m_j-1}{2m_j}+\kappa\frac{1}{m_j}\sum_{l\in C_j, l\neq i}\|\Phi(i)-\Phi(l)\|$$

for a single element. For the whole cluster C_j, we get an increase of $\kappa^2(m_j-1)/2+\kappa\frac{1}{m_j}\sum_{i\in C_j}\sum_{l\in C_j, l\neq i}\|\Phi(i)-\Phi(l)\|$ which takes into account also the structure of the cluster in a different manner than in the kernel-k-means cost function (4). It is easily seen that, with an increase in κ, not squared distances, but linear distances will play a role in the clustering process. Hence the different outcome upon euclidization.

Theorem 3. *For kernel-k-means, adding a constant to dissimilarity measures of different elements is a clustering non-preserving operation.*

We conclude that even the original Cailliez transformation makes the application of the kernel-trick in the kernel k-means algorithm a questionable practice. We do not know how to interpret the outcome of the clustering with this method. It has to be stressed, however, that given a kernel clustering method with a cost function based not on ℓ_2, as in the case of traditional k-means, but rather based on ℓ_1 (Kashima *et al.*, 2008; Bradley *et al.*, 1996; Du *et al.*, 2015), the original Cailliez transformation would be the appropriate choice.

7. Failure of the Gower/ Legendre euclidization theorem in the Lingoes part

Turn our attention to the *Lingoes part* of the Gower and Legendre theorem. Note that Roth *et al.* (2003) in their Theorem 2 provide the correct formulation attributed also to Cox and Cox (2001). Continue the above example. Constant σ, implied by Part (a) of Theorem 2 , for ${}_{nE}F$ amounts to $\sigma=1090.376$. Modifying the distance matrix, as prescribed by Part (a) of Theorem 2, we get a new kernel matrix *which is again non-Euclidean, because its lowest eigenvalue is equal to* -545.188.

Let us investigate a "correction" of Gower/Legendre's "euclidization" theorem, that is, the result due to Lingoes (1971).

Theorem 4. (Lingoes, 1971) *Any dissimilarity matrix D may be turned to a Euclidean distance matrix, see their Theorem 7, by adding an appropriate constant (to non-diagonal elements), e.g., $d'(\mathfrak{z},\mathfrak{y})=\sqrt{d(\mathfrak{z},\mathfrak{y})^2+2\sigma}$, where σ is a constant such that $\sigma\geq-\lambda_m$, λ_m being the smallest eigenvalue of $(\mathbf{I}-\mathbf{1}\mathbf{1}^T/m)(-\frac{1}{2}D^{\circ 2})(\mathbf{I}-\mathbf{1}\mathbf{1}^T/m)$, $D^{\circ 2}$ is the matrix of squared values of elements of D, m is the number of rows/columns in D.*

Proof. It has been proven (Kłopotek, 2019, Eqns. (24) and (25)) that, given $\mathbf{s}^T\mathbf{1}=1, \mathbf{t}^T\mathbf{1}=1$, we have

$$(\mathbf{I}-\mathbf{1}\mathbf{t}^T)(\mathbf{I}-\mathbf{1}\mathbf{s}^T)=\mathbf{I}-\mathbf{1}\mathbf{t}^T. \tag{20}$$

Equation (20) allows us to conclude that given

$$F=-\frac{1}{2}\left(\mathbf{I}-\frac{\mathbf{1}\mathbf{1}^T}{m}\right)D^{\circ 2}\left(\mathbf{I}-\frac{\mathbf{1}\mathbf{1}^T}{m}\right)$$

for a dissimilarity matrix D, the following holds:

$$\begin{aligned}F&=\left(\mathbf{I}-\frac{\mathbf{1}\mathbf{1}^T}{m}\right)F\left(\mathbf{I}-\frac{\mathbf{1}\mathbf{1}^T}{m}\right)\\ &=F\left(\mathbf{I}-\frac{\mathbf{1}\mathbf{1}^T}{m}\right)=\left(\mathbf{I}-\frac{\mathbf{1}\mathbf{1}^T}{m}\right)F.\end{aligned}$$

Let $\mathbf{v}$ be an eigenvector of F for a non-zero eigenvalue λ. Therefore

$$\begin{aligned}\lambda\left(\mathbf{I}-\frac{\mathbf{1}\mathbf{1}^T}{m}\right)\mathbf{v}&=\left(\mathbf{I}-\frac{\mathbf{1}\mathbf{1}^T}{m}\right)F\mathbf{v}\\ &=F\mathbf{v}=F\left(\mathbf{I}-\frac{\mathbf{1}\mathbf{1}^T}{m}\right)\mathbf{v}.\end{aligned}$$

Assuming that $\mathbf{v}'=(\mathbf{I}-\frac{\mathbf{1}\mathbf{1}^T}{m})\mathbf{v}$, we get

$$\lambda\mathbf{v}'=F\mathbf{v}',$$

which means that $\mathbf{v}'$ is also an eigenvector of F for the same eigenvalue. Notably, the sum of components of $\mathbf{v}'$ is equal to zero. Consider the following expression for some number σ:

$$\begin{aligned} F' &= \left(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{m}\right)\left(-\frac{1}{2}D^{\circ 2} - \sigma\left(\mathbf{1}\mathbf{1}^T - \mathbf{I}\right)\right) \\ &\quad \cdot \left(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{m}\right) \\ &= \left(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{m}\right)\left(-\frac{1}{2}D^{\circ 2}\right)\left(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{m}\right) \\ &\quad + \sigma\left(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{m}\right). \end{aligned}$$

Consider an eigenvector $\mathbf{v}'$ of F' for a non-zero eigenvalue λ', such that the sum of its components equals zero. For each λ, such a vector always exists. We see immediately that

$$F'\mathbf{v}' = \left(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{m}\right)\left(-\frac{1}{2}D^{\circ 2}\right)\left(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{m}\right)\mathbf{v}' + \sigma\mathbf{v}',$$

$$\lambda'\mathbf{v}' = F\mathbf{v}' + \sigma\mathbf{v}',$$

$$(\lambda' - \sigma)\mathbf{v}' = F\mathbf{v}',$$

i.e., that $(\lambda' - \sigma)$ is an eigenvalue of F with eigenvector $\mathbf{v}'$.

This means that, by subtracting σ from non-diagonal elements of $-\frac{1}{2}D^{\circ 2}$ in the computation of F, we can increase its eigenvalues of eigenvectors with zero-sum components by σ. But subtracting σ from non-diagonal elements of $-\frac{1}{2}D^{\circ 2}$ means adding σ to non-diagonal elements of $\frac{1}{2}D^{\circ 2}$, or adding 2σ to non-diagonal elements of $D^{\circ 2}$, or just replacing non-diagonal elements d_{ij} of D with $\sqrt{d_{ij}^2 + 2\sigma}$, when $i \neq j$. If we add at least the negative of the lowest eigenvalue of non-Euclidean F to all its eigenvalues, then, of course, it turns to a Euclidean one, given that all eigenvectors with non-zero eigenvalues have zero sums of components.

How can we tell if all such eigenvectors have zero-sums? In case all eigenvalues are different, this is simple. As shown, each eigenvalue has a zero-sum eigenvector, and this is the only one up to a scaling factor.

The details of handling special cases (of identical eigenvalues) follow now. Consider the set of all eigenvectors related to multiple eigenvalues. The whole set can be represented as a linear combination of some number of orthogonal vectors from this set with the number equal to the multiplicity of the eigenvalue.

Let $\mathbf{v}$ be one of these orthogonal vectors. Then any linear combination of all the other orthogonal vectors is orthogonal to $\mathbf{v}$. Let $\mathbf{v}$” be an example from this combination. Then clearly $\mathbf{v}\text{”}^T\mathbf{v} = 0$. But also $\mathbf{v}\text{”}^T(F\mathbf{v}) = \lambda\mathbf{v}\text{”}^T\mathbf{v} = 0$. Hence $\mathbf{v}\text{”}^T(F(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{m})\mathbf{v}) = \mathbf{v}\text{”}^T\lambda\mathbf{v}' = 0$. So $\mathbf{v}' = (\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{m})\mathbf{v}$ is orthogonal to $\mathbf{v}$”. As the latter represents any vector orthogonal to $\mathbf{v}$ of the subspace co-spanned by $\mathbf{v}$, so $\mathbf{v}'$ must be identical to $\mathbf{v}$ up to scaling factor. Hence the subspace of eigenvectors can be spanned by a set of orthogonal vectors with components summing up to zero. Thus all the eigenvectors of F have this property and hence adding the respective constant adds it to all the eigenvalues of the matrix F. ■

Let us illustrate Theorem 4 by continuing the previous example. The euclidization of the kernel ${}_{nE}F$, according to Theorem 4, will lead to a kernel matrix ${}_{E}F$ which is now Euclidean, because its lowest eigenvalue is zero. The kernel matrix ${}_{E}F$ implies a clustering [2, 1, 2, 2, 1, 2] with the total value of the cost function 6269.502, cf. (7). Other clusterings would not do better. Check, e.g., that the clustering [1,1,1,2,2,2] produces the cost function amounting to 6377.502, which is higher than what kernel-k-means produces. Consider a different clustering, [1,1,1,2,2,2], where you choose weighted cluster centers with weights [10,1,1,10,1,1], instead of the k-means cluster centers. Then the cost function will amount to 8506.847, which is again higher than what kernel-k-means produces. In a Euclidean space, kernel-k-means produces appropriate results. The clustering obtained is identical with the clustering delivered by kernel-k-means from the original kernel matrix ${}_{nE}F$. But what about the case when we have the kernel matrix $K_{\mathcal{G}}(D)$ of Gower's general form (3) instead of the distance matrix D? In this case

$$\begin{aligned} K^* &= \left(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{m}\right)K_{\mathcal{G}}(D)\left(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{m}\right) \\ &= \left(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{m}\right)(-\frac{1}{2}D^{\circ 2})\left(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{m}\right) \\ &= F_{\mathcal{L}}(D) \end{aligned} \tag{21}$$

see (11). Hence, we can make Euclidean (Mercer's) any non-Euclidean kernel matrix K by identifying $\sigma \geq -\lambda_m$ with λ_m being the lowest (negative) eigenvalue of $\left(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{m}\right)K\left(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{m}\right)$ and then modifying K directly to

$$K' = K - \sigma\left(\mathbf{I} - \mathbf{1}\mathbf{s}^T\right)\left(\mathbf{1}\mathbf{1}^T - \mathbf{I}\right)\left(\mathbf{I} - \mathbf{s}\mathbf{1}^T\right) \tag{22}$$

because

$$\begin{aligned} &\left(\mathbf{I} - \mathbf{1}\mathbf{s}^T\right)\left(-\frac{1}{2}\left(D^{\circ 2} + 2\sigma\left(\mathbf{1}\mathbf{1}^T - \mathbf{I}\right)\right)\right)\left(\mathbf{I} - \mathbf{s}\mathbf{1}^T\right) \\ &\qquad = K - \sigma\left(\mathbf{I} - \mathbf{1}\mathbf{s}^T\right)\left(\mathbf{1}\mathbf{1}^T - \mathbf{I}\right)\left(\mathbf{I} - \mathbf{s}\mathbf{1}^T\right). \end{aligned}$$

Consider once again

$$F = -\frac{1}{2}\left(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{m}\right)D^{\circ 2}\left(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{m}\right)$$

and its eigenvalue λ and eigenvector $\mathbf{v} = \left(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{m}\right)\mathbf{v}$. Thus $\lambda\mathbf{v} = F\mathbf{v}$ and therefore $\lambda\left(\mathbf{I} - \mathbf{1}\mathbf{s}^T\right)\mathbf{v} =$

$\left(\mathbf{I}-\mathbf{1}\mathbf{s}^T\right) F\mathbf{v}$. Applying (20), we obtain

$$\begin{aligned}&\lambda\left(\mathbf{I}-\mathbf{1}\mathbf{s}^T\right)\mathbf{v}\\ &\qquad=\left(\mathbf{I}-\mathbf{1}\mathbf{s}^T\right)F\left(\mathbf{I}-\mathbf{s}\mathbf{1}^T\right)\left(\mathbf{I}-\frac{\mathbf{1}\mathbf{1}^T}{m}\right)\mathbf{v},\end{aligned}$$

that is

$$\lambda\left(\mathbf{I}-\mathbf{1}\mathbf{s}^T\right)\mathbf{v}=K\mathbf{v}. \tag{23}$$

Therefore, λ is a solution to the generalized eigenproblem (23). Thus we can generalize the Lingoes euclidization Theorem 4.

Theorem 5. *Any dissimilarity matrix D may be turned to a Euclidean distance matrix by adding constant σ to the squared distances: $d'(\mathfrak{z},\mathfrak{y})=\sqrt{d(\mathfrak{z},\mathfrak{y})^2+2\sigma}$, where σ is such that $\sigma\geq-\lambda_m$, λ_m being the smallest eigenvalue of the generalized eigenproblem*

$$\lambda\left(\mathbf{I}-\mathbf{1}\mathbf{s}^T\right)\mathbf{v}=K_{\mathcal{G}}(D)\mathbf{v},$$

where $K_{\mathcal{G}}(D)=\left(\mathbf{I}-\mathbf{1}\mathbf{s}^T\right)(-\frac{1}{2}D^{\circ 2})\left(\mathbf{I}-\mathbf{s}\mathbf{1}^T\right)$ being Gower's general form (3).

It follows that we are not restricted to the Lingoes kernelizations $F_{\mathcal{L}}(D)$ of (11) and we can apply the general Gower kernelization $K_{\mathcal{G}}(D)$ of (3) when we are discussing euclidizations for the purposes of kernel-k-means application, with a kernel trick. But this happens at the expense that we cannot solve the simple eigenproblem of the matrix $F_{\mathcal{L}}(D)$. We have to solve a more general eigenproblem (23) and need to know *a priori* the vector $\mathbf{s}$.

Note that, in order to apply (22) to the kernel matrix K, we need to know the vector $\mathbf{s}$. But this is not necessary if we instead use the kernel matrix

$$K^*=\left(\mathbf{I}-\frac{\mathbf{1}\mathbf{1}^T}{m}\right)K\left(\mathbf{I}-\frac{\mathbf{1}\mathbf{1}^T}{m}\right).$$

Then the euclidization is performed as

$$\begin{aligned}K^{*\prime}&=K^*-\sigma\left(\mathbf{I}-\frac{\mathbf{1}\mathbf{1}^T}{m}\right)\left(\mathbf{1}\mathbf{1}^T-\mathbf{I}\right)\left(\mathbf{I}-\frac{\mathbf{1}\mathbf{1}^T}{m}\right)\\ &=K^*-\sigma(-\left(\mathbf{I}-\frac{\mathbf{1}\mathbf{1}^T}{m}\right))\left(\mathbf{I}-\frac{\mathbf{1}\mathbf{1}^T}{m}\right)\\ &=K^*+\sigma\left(\mathbf{I}-\frac{\mathbf{1}\mathbf{1}^T}{m}\right),\end{aligned}$$

where σ is derived from solving the simple eigenproblem of the matrix K^*, because of (21) and Theorem 4.

8. Advantage of the Lingoes approach to euclidization

Let us investigate this phenomenon more generally.

Theorem 6. *If we pursue the kernel-k-means clustering when seeking an optimum among cluster center sets being a subset of the set of $\boldsymbol{\mu}_j^\Phi$ that may only be equal to*

$$\boldsymbol{\mu}_j^\Phi=\frac{1}{m_j}\sum_{i\in C_j}\Phi(i) \tag{24}$$

for some subset C_j of all the data points and no other vectors in the feature space are taken into account, then, after adding a constant 2σ to the distance matrix $d'(\mathfrak{z},\mathfrak{y})=\sqrt{d(\mathfrak{z},\mathfrak{y})^2+2\sigma}$, the optimal clustering will remain the same.

Proof. Note that the cost function of kernel-k-means from (4) may be reformulated as follows: Assume that we have the set of k cluster centers $\{\boldsymbol{\mu}_j^\Phi; j=1,\dots,k\}$, inducing clusters $\mathcal{C}=\{C_1,\dots,C_k\}$ (consisting of points for which the given cluster center is the closest one), where cluster $C_j\in\mathcal{C}$ is of cardinality m_j. From (4) it follows that

$$\begin{aligned}&J(\boldsymbol{\mu}_j^\Phi; j=1,\dots,k)\\ &\quad=\sum_{j=1}^k\sum_{i\in C_j\in\mathcal{C}}\|\Phi(i)-\boldsymbol{\mu}_j^\Phi\|^2\\ &\quad=\sum_{j=1}^k\frac{1}{2m_j}\sum_{i\in C_j\in\mathcal{C}}\sum_{l\in C_j\in\mathcal{C}}\|\Phi(i)-\Phi(l)\|^2.\end{aligned}$$

Let us investigate any such set of cluster centers $\{\boldsymbol{\mu}_j^\Phi; j=1,\dots,k\}$ inducing the clustering $\mathcal{C}$. Consider only such clusterings where the cluster centers are at the same time their gravity centers. Under this latter condition, we can attribute to each data point i from cluster C_j the contribution to the entire cost function being

$$\frac{1}{2m_j}\sum_{l\in C_j}\|\Phi(i)-\Phi(l)\|^2.$$

If we add a value 2σ to all squared distances of an element i of a cluster C_j, then its contribution will increase to

$$\begin{aligned}&\frac{1}{2m_j}\sum_{l\in C_j,l\neq i}(\|\Phi(i)-\Phi(l)\|^2+2\sigma)\\ &\qquad=\sigma\frac{m_j-1}{m_j}+\frac{1}{2m_j}\sum_{l\in C_j}\|\Phi(i)-\Phi(l)\|^2\end{aligned}$$

(increase by $\sigma\frac{m_j-1}{m_j}$) because $d(i,i)=0$ is unchanged. This increase is just cluster size-dependent and not cluster structure-dependent. Consequently, in the whole cluster C_j contribution to the cost function will increase by

$$\sigma\frac{m_j-1}{m_j}m_j=\sigma\cdot(m_j-1).$$

Thus, the overall cost function of all k clusters will increase by $\sigma\cdot(m-k)$. That is, it is independent of the clustering $\mathcal{C}$, given of course that the cluster centers are their gravity centers, which is what kernel-k-means produces.

Hence the optimum clustering of k-means, achievable by kernel-k-means will remain unchanged after this addition. ■

Theorem 6 and its proof imply the following result.

Theorem 7. *For kernel-k-means, adding a constant to squared dissimilarity measures of non-identical elements is a clustering preserving and embeddability improving operation.*

Note that the transformation mentioned above (i) increases all distances, (ii) the absolute increase in distances is the largest for the smallest distances, and the smallest for the largest, and (iii) no new clustering structures occur under this transformation. In this way, we define a new axiom/property of k-means, i.e., in that we require that the clustering algorithm yields the same result under the mentioned distance change/transformation. The idea behind is that in the permissible domain for k-means (Euclidean), the optimum is unchanged if we add a constant to the squared distances between different elements. By means of this conceptual extension, we can carry on this assumption backward into non-Euclidean distances. Then we need to define under what regime we compute the permissible optimum of k-means, because it is not true in the whole space itself. Only if we limit the permissible space in a reasonable way, we can still assume that we are computing k-means optimum. In consequence, if we agree that the kernel function $\Phi(\cdot)$ for kernel k-means is deemed to transmit the data points into the Euclidean space under the above-mentioned invariance transformation, then it is permissible to apply kernel-k-means without checking for embeddability.[7]

9. Concluding remarks

In this paper, we resolved the issue of applicability of kernel-k-means for non-embeddable kernel matrices. First, we showed that kernel k-means produces wrong results when applied to non-Euclidean kernel matrices. We demonstrated that, under some types of euclidization, the kernel-k-means will produce different results before and after euclidization. We identified that the *Lingoes* transformation is the one free of this effect and so the usage of the kernel trick for non-Euclidean spaces is justified.

Though other researchers, like Roth *et al.* (2003), were interested in euclidizations providing the same results as kernel-k-means for non-Euclidean spaces, but they did not realize that applying kernel-k-means for non-Euclidean spaces produces essentially wrong results and hence neither sought nor provided ways of resolving this issue.

Additionally, we provided alternative proofs of the correctness of euclidizations of Lingos and Cailliez, which give new insights into their results. In particular, we paved the way for considering the general type of kernelization proposed by Gower (1982), recalled here as Eqn. (3), instead of the double-centering one. As we have shown, the Lingoes theorem applies with a slight modification not only to dissimilarity matrices, but also to kernel matrices of this generalized type. An open question is whether or not the generalized Gower kernelization covers all conceivable (Euclidean and non-Euclidean) kernel matrices matching a given dissimilarity matrix. In such a case we would not need to turn the kernel matrix to a dissimilarity matrix in order to verify if it is Euclidean and to correct the modified kernel matrix $\left(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{m}\right) K \left(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{m}\right)$ directly. Otherwise, there is a question of whether or not the cases not covered here are easily reducible to the Gower's kernel matrices so that the same would apply.

Software. Please feel free to experiment with our R package (source code) implementing the kernel-k-means functionality: `https://home.ipipan.waw.pl/m.klopotek/ipi_archiv/kernelKmeansAndPlusPlusDemo_1.0.tar.gz`.

References

Ackerman, M., Ben-David, S. and Loker, D. (2010). Towards property-based classification of clustering paradigms, *in* J. Lafferty *et al.* (Eds), *Advances in Neural Information Processing Systems 23*, Curran Associates, Red Hook, NY, pp. 10–18.

Balaji, R. and Bapat, R. (2007). On Euclidean distance matrices, *Linear Algebra and Its Applications* **424**(1): 108–117.

Bao, T. and Kadobayashi, Y. (2008). On tighter inequalities for efficient similarity search in metric spaces, *IAENG International Journal of Computer Science* **35**(3): IJCS_35_3_17.

[7] As pointed out by Higham (1988), we can consider this operation as a search for a matrix K', which is positive semi-definite, close (preferably the closest) to the matrix K. The closeness can be expressed as a norm $\|K - K'\|$, whereby for the matrix A we may use the 2-norm $\|A\|_2$, which is the maximal absolute value of the eigenvalues of A or the Frobenius norm $\|A\|_2 = \sqrt{\sum_i \sum_j a_{ij}^2}$. Therefore, $\|K^{*\prime} - K^*\|_2 = \|\sigma\left(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{m}\right)\|_2 = |\sigma| \|\left(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{m}\right)\|_2 = |\sigma|$. Note that Higham (1988) proposes a different way of making K positive semidefinite by seeking the closest matrix (see their Theorem 3.1 in particular), but the resulting matrix does not fit our criterion of clustering before and after transformation being identical. The reason is that their formula (3.2) modifies the matrix K by adding a constant to the diagonal. As implied by Gower's formula (3), this leads to a possible variety of modifications of squared distances between data points. The approach proposed here, based on the Lingoes formula, is optimal in some other sense than Higham's. That is, the smallest constant is found such that adding it to all distances (between distinct point) leads to a positive semidefinite matrix K', as implied by our proof.

Bradley, P.S., Mangasarian, O.L. and Street, W.N. (1996). Clustering via concave minimization, *Proceedings of the 9th International Conference on Neural Information Processing Systems, NIPS'96, Denver, CO, USA*, pp. 368–374.

Cailliez, F. (1983). The analytical solution of the additive constant problem, *Psychometrika* **48**(2): 305–308.

Chitta, R., Jin, R., Havens, T. and Jain, A. (2011). Approximate kernel k-means: Solution to large scale kernel clustering, *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'11, San Diego, CA, USA*, pp. 895–903.

Choi, H. and Choi, S. (2005). Kernel Isomap on noisy manifold, *4th International Conference on Development and Learning, Osaka, Japan*, pp. 208 – 213.

Cox, T.F. and Cox, M.A.A. (2001). *Multidimensional Scaling, 2nd Edn*, Chapman & Hall, London.

Demontis, A., Melis, M., Biggio, B., Fumera, G. and Roli, F. (2017). Super-sparse learning in similarity spaces, *CoRR*: abs/1712.06131.

Dokmanic, I., Parhizkar, R., Ranieri, J. and Vetterli, M. (2015). Euclidean distance matrices: A short walk through theory, algorithms and applications, *CoRR*: abs/1502.07541.

Du, L., Zhou, P., Shi, L., Wang, H., Fan, M., Wang, W. and Shen, Y. (2015). Robust multiple kernel k-means using l_{21}-norm, *Proceedings of the 24th International Conference on Artificial Intelligence, IJCAI'15, Buenos Aries, Argentina*, pp. 3476–3482.

Gisbrecht, A. and Schleif, F.-M. (2015). Metric and non-metric proximity transformations at linear costs, *Neurocomputing* **167**(1): 643–657.

Gonzalez, J. and Munoz, A. (2010). Representing functional data in reproducing kernel Hilbert spaces with applications to clustering and classification, *Statistics and Econometrics Series* **013**, Working paper 10-27.

Gower, J.C. (1982). Euclidean distance geometry, *The Mathematical Scientist* **7**: 1–14.

Gower, J.C. (1985). Properties of Euclidean and non-Euclidean distance matrices, *Linear Algebra and Its Applications* **67**: 81–97.

Gower, J. and Legendre, P. (1986). Metric and Euclidean properties of dissimilarity coefficients, *Journal of Classification* **3**(1): 5–48.

Handhayania, T. and Hiryantob, L. (2015). Intelligent kernel k-means for clustering gene expression, *International Conference on Computer Science and Computational Intelligence (ICCSCI 2015), Jakarta, Indonesia*, Vol. 59, pp. 171–177.

Higham, N.J. (1988). Computing a nearest symmetric positive semidefinite matrix, *Linear Algebra and Its Applications* **103**: 103–118.

Hofmann, T., Schölkopf, B. and Smola, A.J. (2008). Kernel methods in machine learning, *Annals of Statistics* **36**(3): 1171–1220.

Jacobs, D., Weinshall, D. and Gdalyahu, Y. (2000). Classification with nonmetric distances: Image retrieval and class representation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(6): 583–600.

Jain, A. and Zongker, D. (1998). Representation and recognition of handwritten digits using deformable templates, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **19**(12): 1386–1390.

Jaworski, M. (2018). Regression function and noise variance tracking methods for data streams with concept drift, *International Journal of Applied Mathematics and Computer Science* **28**(3): 559–567, DOI: 10.2478/amcs-2018-0043.

Kashima, H., Hu, J., Ray, B. and Singh, M. (2008). K-means clustering of proportional data using l1 distance, *Proceedings of the 19th International Conference on Pattern Recognition, Tampa, FL, USA*.

Kleinberg, J. (2002). An impossibility theorem for clustering, *Proceedings of the 16th Neural Information Processing Systems Conference, NIPS 2002, Vancouver, BC, Canada*, pp. 446–453.

Kłopotek, M. (2019). On the existence of kernel function for kernel-trick of k-means in the light of Gower theorem, *Fundamenta Informaticae* **168**(1): 25–43.

Legendre, P. and Legendre, L. (1998). *Numerical Ecology, 2nd Edn*, Elsevier, Amsterdam.

Li, C., Georgiopoulos, M. and Anagnostopoulos, G.C. (2013). Kernel-based distance metric learning in the output space, *International Joint Conference on Neural Networks (IJCNN), Dallas, TX, USA*, pp. 1–8.

Lingoes, J. (1971). Some boundary conditions for a monotone analysis of symmetric matrices, *Psychometrika* **36**(2): 195–203.

Loosli, G., Canu, S. and Ong, C. (2016). Learning SVM in Kreĭn spaces, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **38**(6): 1204–1216.

Marin, D., Tang, M., Ayed, I.B. and Boykov, Y. (2019). Kernel clustering: Density biases and solutions, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **41**(1): 136–147.

Marteau, P.-F. (2019). Times series averaging and denoising from a probabilistic perspective on time-elastic kernels, *International Journal of Applied Mathematics and Computer Science* **29**(2): 375–392, DOI: 10.2478/amcs-2019-0028.

Pan, V.Y. and Chen, Z.Q. (1999). The complexity of the matrix eigenproblem, *Proceedings of the 31st Annual ACM Symposium on Theory of Computing, STOC'99, Atlanta, GA, USA*, pp. 507–516.

Pękalska, E. and Duin, R. (2005). *The Dissimilarity Representation for Pattern Recognition: Foundations and Applications (Machine Perception and Artificial Intelligence)*, World Scientific, River Edge, NJ.

Pękalska, E., Paclík, P. and Duin, R. (2002). A generalized kernel approach to dissimilarity-based classification, *Journal of Machine Learning Research* **2**: 175–211.

Qi, H. (2016). A convex matrix optimization for constant problem in multidimensional scaling with application to LLE, *SIAM Journal on Optimization* **26**(4): 2564–2590.

Richter, R., Kyprianidis, J., Springborn, B. and Alexa, M. (2017). Constrained modelling of 3-valent meshes using a hyperbolic deformation metric, *Computer Graphics Forum* **36**(6): 62–75.

Roth, V., Laub, J., Kawanabe, M. and Buhmann, J. (2003). Optimal cluster preserving embedding of nonmetric proximity data, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **25**(12): 1540–1551.

Sarma, T., Vishwanath, P. and Reddy, B. (2013). Single pass kernel k-means clustering method, *Sadhana* **38**(3): 407–419.

Schleif, F. and Tino, P. (2015). Indefinite proximity learning: A review, *Neural Computation* **27**(10): 2039–2096.

Schoenberg, I.J. (1938). Metric spaces and positive definite functions, *Transactions of the American Mathematical Society* **44**(3): 522–536.

Shawe-Taylor, J. and Cristianini, N. (2004). *Kernel Methods for Pattern Analysis*, Cambridge University Press, Cambridge.

Szalkai, B. (2013). An implementation of the relational k-means algorithm, *CoRR*: abs/1304.6899.

Szekely, G. and Rizzo, M. (2014). Partial distance correlation with methods for dissimilarities, *CoRR*: abs/1310.2926.

Tzortzis, G. and Likas, A.C. (2009). The global kernel k-means algorithm for clustering in feature space, *IEEE Transactions on Neural Networks* **7**(20): 1181–94.

Villmann, T., Kaden, M., Nebel, D. and Bohnsack, A. (2016). Similarities, dissimilarities and types of inner products for data analysis in the context of machine learning, *in* L. Rutkowski *et al.* (Eds), *Artificial Intelligence and Soft Computing: 15th International Conference, ICAISC 2016*, Springer, Cham, pp. 125–133.

Wierzchoń, S. and Kłopotek, M. (2018). *Modern Clustering Algorithms*, Springer, Cham.

Yao, Y. and Chen, H. (2018). Multiple kernel k-means clustering by selecting representative kernels, *CoRR*: abs/1811.00264.

Robert A. Kłopotek, PhD, is a lecturer at the Faculty of Mathematics and Natural Sciences of Cardinal Stefan Wyszyński University in Warsaw and the deputy director of the Institute of Computer Science at that university. He also works in an international company as a data scientist. His professional interests are social network analysis, data mining, data analysis, application of GPGPU in selected machine learning algorithms, and theoretical foundations for machine learning algorithms.

Mieczysław A. Kłopotek received his MSc and PhD degrees in computer science from the Dresden University of Technology. He then worked at the Semiconductor Research and Production Center in Warsaw. Thereafter he joined the Institute of Computer Science of the Polish Academy of Sciences, where he received his DSc degree in 1999. In 2009 he was granted a professorial title by the President of Poland. He has also worked at IBM Warsaw on parallel statistical software for the Netezza appliance, co-authoring five patents. He has led a research group developing the first large-scale Polish semantic search engine. His current research encompasses data, text and web mining and machine learning.

Sławomir T. Wierzchoń received his MSc and PhD degrees in computer science from the Warsaw University of Technology, Poland. He holds a DSc degree from the Polish Academy of Sciences. In 2003 he received a professorial title from the President of Poland. Currently he is a full professor at the Institute of Computer Science of the Polish Academy of Sciences. He has published over 150 peer reviewed papers in international journals and international conferences, and 11 books in the field of machine learning. He has cooperated with medical centers in the area of statistical analysis, and has participated in research projects concerning various topics of machine learning.

Received: 17 February 2020
Revised: 27 June 2020
Accepted: 17 July 2020