Control and Cybernetics

vol. **40** (2011) No. 2

k-centroids clustering for asymmetric dissimilarities*

by

Dominik Olszewski

Warsaw University of Technology
ul. Koszykowa 75, 00-662 Warsaw, Poland
e-mail: olszewsd@ee.pw.edu.pl

Abstract: In this paper, an asymmetric version of the k-centroids clustering algorithm is proposed. The asymmetry arises from the use of the asymmetric dissimilarities in the k-centroids algorithm. Application of the asymmetric measures of dissimilarity is motivated by the basic nature of the k-centroids algorithm, which uses dissimilarities in the asymmetric manner. It finds the minimal dissimilarity between an object being currently allocated, and one of the clusters centroids. Clusters centroids are treated as the dominant points governing the asymmetric relationships in the entire cluster analysis. The results of the experimental study on real and simulated data have shown the superiority of the asymmetric dissimilarities employed for the k-centroids method over their symmetric counterparts.

Keywords: k-centroids clustering, asymmetric dissimilarity, sound recognition, heart rhythm recognition, feature extraction.

1. Introduction

We are used to consider symmetry as a preferable property in many aspects of our life. We use to associate it with the state of balance and harmony, especially in science. However, there might be some cases, when our expectations or even requirements may be quite opposite, and an asymmetric view on the problem might be more advantageous.

Thus, for instance, we are used to demand the symmetry for dissimilarities as one of their most important properties. And, when it comes to deal with an asymmetric dissimilarity we often attempt to symmetrize it someway. This paper describes a problem, which as we claim, presents the circumstances suggesting employing the asymmetric measure of dissimilarity, as more desirable.

The k-centroids clustering algorithm (Chaturvedi et al., 1997; Czekala and Kuziak, 1999; Leisch, 2006) is a data analysis tool used to form arbitrary number of clusters in the analyzed data set. The algorithm aims to separate clusters

*Submitted: July 2009; Accepted: May 2011.

of possibly most similar objects. The similarity between two objects needs to be expressed with a certain chosen measure of similarity, or proximity, or in other cases dissimilarity, in the specific space. Each object, in turn, needs to be represented by a vector of reasonably selected features. Object represented as a vector of d features can be interpreted as a point in d-dimensional space. Hence, according to its geometric nature, the k-centroids clustering algorithm can be formulated as follows: given n points in d-dimensional space and the number of desired clusters k, the algorithm seeks a set of k clusters so as to minimize the sum of dissimilarities between each point and its cluster centroid. The cluster centroid is a point being possibly the best representation of the whole cluster.

One particular implementation of the k-centroids algorithm became very famous and is widely used for clustering, mainly because of its computational efficiency. It is called "k-means" after MacQueen's publication (MacQueen, 1967), where the name "k-means" was used for the first time. However, the algorithm itself was known much earlier. It was introduced by Steinhaus (1956). The k-means algorithm is a well-known cluster analysis tool, and, over the years, it has been extensively studied (Kanungo et al., 2002; Olszewski et al., 2010; Xiong et al., 2009).

Application of the asymmetric proximities in data analysis has been extensively studied by Okada and Imaizumi (Okada, 2000; Okada and Imaizumi, 1997, 2007). They have concentrated in their work on the multidimensional scaling for analyzing one-mode two-way (object $\times$ object) and two-mode three-way (object $\times$ object $\times$ source) asymmetric proximities. They have introduced the dominant point governing asymmetry in the proximity relationships among objects, represented as points in the multidimensional Euclidean space. They claim that ignoring or neglecting the asymmetry in proximity analysis discards potentially valuable information. An asymmetric version of the k-means algorithm is presented in Olszewski (2011).

Our method can be regarded as an extension of these solutions for the k-centroids clustering algorithm, where centroids of clusters are treated as the dominant points governing the multiple allocations of objects, and consequently, governing the whole clustering process. Therefore, the distinction between a centroid and a single object is that the centroid is a privileged entity acting as an attractor of objects in the analyzed data set. Our solution can be also interpreted as a generalization of Okada's and Imaizumi's idea for the multidimensional non-Euclidean spaces, associated with the non-standard asymmetric dissimilarity measures, like, e.g., the Kullback-Leibler divergence. Finally, we wanted to confirm their assertion that the property of asymmetry does not have to be considered as an inhibiting shortcoming, but, quite the contrary, in certain areas of research, it can be even significantly beneficial.

The k-centroids algorithm forms clusters on the basis of multiple allocations of objects to the nearest clusters. The nearest cluster is the one with the minimal dissimilarity between its centroid and the object considered. Hence, the principal behavior of the discussed algorithm is based on evaluating the dissimilarity

measure between two distinct entities (object vs. cluster centroid). Therefore, the choice of the dissimilarity measure for this algorithm may have significant influence on its performance. When asymmetry arises, symmetric dissimilarities produce big values for most pairs of data points, and do not reflect accurately the object relationships (Martín-Merino and Muñoz, 2005). We propose employing the asymmetric dissimilarities as dissimilarity measures in the k-centroids algorithm, since we claim that it is more consistent with the fundamental nature of this algorithm, i.e., properly reflects the asymmetric relationship between a single object and a cluster centroid.

The relationship between a cluster centroid and a single object can be considered to be asymmetric, because of the hierarchical association between these two entities. The hierarchical associations in data are closely related to the asymmetry. This relation has been noticed in Muñoz et al. (2003). The dissimilarity from a more general entity to a more specific one should be greater than in the opposite direction. As stated in Martín-Merino and Muñoz (2005), asymmetry can be interpreted as a particular type of hierarchy. A cluster centroid is a one-point representation of the entire cluster, and it is computed on the basis of all current members of the cluster. Therefore, it reflects the properties of all objects in the cluster. On the other hand, a single object is an elementary portion of data in the analyzed set. Consequently, it can be regarded as a sub-element with respect to centroid. Hence, the dissimilarity from a centroid to an object should be greater than from an object to a centroid.

In other words, we can assert that following the Okada's and Imaizumi's work, we treat centroids of the clusters as the dominant points responsible for the asymmetry in the cluster analysis. Therefore, they act as privileged entities "attracting" the objects in the analyzed data set.

Our method maintains the advantages of the symmetric k-centroids algorithm, i.e., efficiency, flexibility, and computational simplicity for large data sets with both numerical and categorical attributes.

2. k-centroids clustering algorithm

The k-centroids clustering algorithm can be formulated in two different versions: the batch version and the online version. The difference between these two variants refers to the execution of Step 1 of the algorithm (see below). In case of the batch version, the algorithm, in Step 1, iterates over the entire data set, and assigns each object to the nearest cluster. On the other hand, in case of the online version, Step 1 consists of only one assignment of a single object to the nearest cluster. Consequently, in case of the batch variant, the clusters centroids are recalculated after allocation of each object from the entire data set, while, in case of the online variant, the clusters centroids are moved after each allocation of a single object to the nearest cluster. We focus on the batch version, and this kind of approach will be considered throughout this paper.

PROCEDURE 1 *The k-centroids algorithm starts from a random choice of k samples from the entire data space. Then, the algorithm consists of two alternating steps:*

Step 1. Forming of the clusters: The algorithm iterates over the entire data set, and allocates each sample to the cluster represented by the centroid – nearest to this sample. The nearest centroid is determined with use of a chosen dissimilarity measure.

Step 2. Finding centroids for the clusters: For each cluster, a centroid is determined on the basis of samples belonging to this cluster. The algorithm calculates centroids of the clusters so as to minimize a formal objective function, the error distortion:

$$e\left(X_j\right) \equiv \sum_{i=1}^{n_j} d\left(x_i, c_j\right), \tag{1}$$

where X_j, $j = 1, \ldots, k$ is the j-th cluster, x_i, $i = 1, \ldots, n_j$ are the data samples in the j-th cluster, n_j, $j = 1, \ldots, k$, is the number of data samples in the j-th cluster, c_j, $j = 1, \ldots, k$, is the centroid of the j-th cluster, k is the number of clusters, and $d\left(a, b\right)$ is a chosen dissimilarity measure.

Both these steps must be carried out with the same dissimilarity measure, in order to guarantee the monotone property of the k-centroids algorithm.

Steps 1 and 2 have to be repeated until the termination condition is met. The termination condition might be either reaching convergence of the iterative application of the objective function (2), or reaching the pre-defined number of cycles.

After each cycle (Step 1 and 2), the value of the following objective function needs to be computed, in order to track the convergence of the whole clustering process:

$$e\left(X\right) \equiv \sum_{j=1}^{k} \sum_{i=1}^{n_j} d\left(x_i, c_j\right), \tag{2}$$

where X is the analyzed set of data samples, and the rest of notation is described in (1).

The most commonly used implementation of the k-centroids clustering algorithm – the k-means algorithm employs the Euclidean distance. In our paper, we propose application of the asymmetric dissimilarities in the k-centroids algorithm, since we claim they are more advisable for this algorithm than popular symmetric quantities. We claim that they are consistent with operating of the k-centroids in its both steps.

A serious problem concerning the k-centroids algorithm is that the clustering process may not converge to an optimal or near-optimal configuration. The

algorithm can assure only local optimality, which depends on the initial locations of samples. An exhaustive study of asymptotic behavior of the k-means algorithm was conducted by MacQueen (1967), where the convergence of the clustering process is proved under certain assumptions. However, MacQueen indicates that the process is not convergent in general. Also, Selim and Ismail (1984) give a rigorous proof of the finite convergence of the k-means-type algorithm, noting that under certain conditions the algorithm may fail to converge to a local optimum, and that it converges under differentiability conditions to the Kuhn-Tucker point.

3. Symmetric and asymmetric dissimilarities

In this section, we present six dissimilarity measures. Three of them are symmetric (the Hellinger distance, the total variation distance, and the Euclidean distance), one is asymmetric (the Kullback-Leibler divergence), and two are either symmetric, or asymmetric, depending on the values of their parameters (the Chernoff distance, and the Lissack-Fu distance). Some of these measures are metrics (satisfy all metric conditions) and some are not, but they still present interesting properties.

3.1. Notation

Throughout this section, we will use the following notation. Let $\mathbb{P}$ and $\mathbb{Q}$ denote two probability measures on a measurable space Ω with σ-algebra $\mathcal{F}$. Let λ be a measure on $(\Omega, \mathcal{F})$ such that $\mathbb{P}$ and $\mathbb{Q}$ are absolutely continuous with respect to λ, with corresponding probability density functions p and q (e.g., λ can be taken to be $(\mathbb{P}+\mathbb{Q})/2$, or can be the Lebesgue measure). For a countable space Ω, measures $\mathbb{P}$ and $\mathbb{Q}$ on $(\Omega, \mathcal{F})$ are N- tuples $(p_1, p_2, \ldots, p_N)$ and $(q_1, q_2, \ldots, q_N)$, respectively (also called as the probability mass functions), satisfying the following conditions: $p_i \geq 0$, $q_i \geq 0$, $\sum_i p_i = 1$, and $\sum_i q_i = 1$. All of the definitions presented in this section do not depend on the choice of the measure λ.

3.2. Symmetric dissimilarities

3.2.1. Hellinger distance

Definition 1 (Gibbs and Su, 2002) *The Hellinger distance between $\mathbb{P}$ and $\mathbb{Q}$ on a continuous measurable space $(\Omega, \mathcal{F})$ is defined as*

$$\begin{aligned} d_{\mathrm{H}}(\mathbb{P}, \mathbb{Q}) &\equiv \left[\frac{1}{2}\int_{\Omega}\left(\sqrt{\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\lambda}} - \sqrt{\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\lambda}}\right)^2 \mathrm{d}\lambda\right]^{1/2} \\ &= \left[\frac{1}{2}\int_{\Omega}\left(\sqrt{p} - \sqrt{q}\right)^2 \mathrm{d}\lambda\right]^{1/2}. \end{aligned} \tag{3}$$

For a countable space Ω, the definition is formulated as follows:

DEFINITION 2 (Gibbs and Su, 2002) *The Hellinger distance between* $\mathbb{P}$ *and* $\mathbb{Q}$ *on a discrete measurable space* $(\Omega, \mathcal{F})$ *is defined as*

$$d_{\mathrm{H}}(\mathbb{P}, \mathbb{Q}) \equiv \left[\frac{1}{2}\sum_{i\in\Omega}\left(\sqrt{p_i} - \sqrt{q_i}\right)^2\right]^{1/2}. \tag{4}$$

3.2.2. Total variation distance

DEFINITION 3 *The total variation distance between* $\mathbb{P}$ *and* $\mathbb{Q}$ *on a continuous measurable space* $(\Omega, \mathcal{F})$ *is defined as*

$$\|\mathbb{P} - \mathbb{Q}\|_1 \equiv 2\sup_{A\subset\Omega}|\mathbb{P}(A) - \mathbb{Q}(A)| = \max_{|h|\leq 1}\left|\int_\Omega h\,\mathrm{d}\mathbb{P} - \int_\Omega h\,\mathrm{d}\mathbb{Q}\right|, \tag{5}$$

where $h : \Omega \to \mathbb{R}$ *satisfies* $|h(x)| \leq 1$. *Total variation distance is a metric, which assumes values in the interval* $[0, 2]$. *Total variation distance is often called the* L^1*-norm of* $\mathbb{P} - \mathbb{Q}$, *and is denoted by* $\|\mathbb{P} - \mathbb{Q}\|_1$.

For a countable space Ω, the definition above becomes:

DEFINITION 4 (Gibbs and Su, 2002) *The total variation distance between* $\mathbb{P}$ *and* $\mathbb{Q}$ *on a discrete measurable space* $(\Omega, \mathcal{F})$ *is defined as*

$$\|\mathbb{P} - \mathbb{Q}\|_1 \equiv \sum_{i\in\Omega}|p_i - q_i|. \tag{6}$$

3.2.3. Euclidean distance

This measure is used to determine the distance between two points in the Euclidean space.

DEFINITION 5 *The Euclidean distance between points* p *and* q *in the* N*-dimensional Euclidean space is defined as*

$$d_{\mathrm{E}}(p, q) \equiv \sqrt{\sum_{i\in\Omega}(p_i - q_i)^2}. \tag{7}$$

The Euclidean distance is a metric, which takes values from the interval $[0, \infty]$. It can be interpreted as a generalization of distance between two points in a plane, i.e., in the 2-dimensional Euclidean space, which can be derived from the Pythagorean theorem.

3.3. Asymmetric dissimilarity

3.3.1. Kullback-Leibler divergence (relative entropy)

DEFINITION 6 (Kullback and Leibler, 1951) *The Kullback-Leibler divergence between* $\mathbb{P}$ *and* $\mathbb{Q}$ *on a continuous measurable space* $(\Omega, \mathcal{F})$ *is defined as*

$$d_{\mathrm{KL}}(\mathbb{P}, \mathbb{Q}) \equiv \int_{\mathrm{S}(\mathbb{P})} p \log_2 \left(\frac{p}{q}\right) \mathrm{d}\lambda, \tag{8}$$

where $\mathrm{S}(\mathbb{P})$ *is the support of* $\mathbb{P}$ *on* Ω. *According to the convention, the value of* $0 \log \frac{0}{q}$ *is assumed as* 0 *for all real* q, *and the value of* $p \log \frac{p}{0}$ *is assumed as* ∞ *for all real non-zero* p. *Therefore, relative entropy takes values from the interval* $[0, \infty]$. *The Kullback-Leibler divergence is not a metric, since it is not symmetric and it does not satisfy the triangle inequality.*

For a countable space Ω the definition is formulated as follows:

DEFINITION 7 (Gibbs and Su, 2002) *The Kullback-Leibler divergence between* $\mathbb{P}$ *and* $\mathbb{Q}$ *on a discrete measurable space* $(\Omega, \mathcal{F})$ *is defined as*

$$d_{\mathrm{KL}}(\mathbb{P}, \mathbb{Q}) \equiv \sum_{i \in \mathrm{S}(\mathbb{P})} p_i \log_2 \left(\frac{p_i}{q_i}\right). \tag{9}$$

3.4. Parameterized dissimilarities

In this subsection, we present two dissimilarities, whose definitions involve parameters. Depending on the parameters values, these dissimilarities can be either symmetric, or asymmetric. This is a very convenient property for the purpose of this paper, since it allows for investigating the influence of symmetrizing and asymmetrizing the same dissimilarity on the final results of clustering.

3.4.1. Chernoff distance

DEFINITION 8 (Chernoff, 1952) *The Chernoff distance between* $\mathbb{P}$ *and* $\mathbb{Q}$ *on a continuous measurable space* $(\Omega, \mathcal{F})$ *is defined as*

$$d_{\mathrm{CH}}(\mathbb{P}, \mathbb{Q}) \equiv -\log_2 \left(\int_{\Omega} p^{\alpha} q^{1-\alpha} \mathrm{d}\lambda\right), \tag{10}$$

where $0 < \alpha < 1$.

Depending on the choice of the parameter α, Chernoff distance can be either symmetric or asymmetric. For $\alpha = 0.5$, it is symmetric, and for all other values of this parameter, it is not. We have chosen $\alpha = 0.1$ and $\alpha = 0.9$, in order to obtain the asymmetric dissimilarity, while $\alpha = 0.5$ resulted in the symmetric dissimilarity.

For a countable space Ω the definition has the following form:

DEFINITION 9 *The Chernoff distance between* $\mathbb{P}$ *and* $\mathbb{Q}$ *on a discrete measurable space* $(\Omega, \mathcal{F})$ *is defined as*

$$d_{\mathrm{CH}}(\mathbb{P}, \mathbb{Q}) \equiv -\log_2 \left(\sum_{i \in \Omega} p_i^{\alpha} q_i^{1-\alpha} \right). \tag{11}$$

3.4.2. Lissack-Fu distance

DEFINITION 10 (Lissack and Fu, 1976) *The Lissack-Fu distance between* $\mathbb{P}$ *and* $\mathbb{Q}$ *on a continuous measurable space* $(\Omega, \mathcal{F})$ *is defined as*

$$d_{\mathrm{LF}}(\mathbb{P}, \mathbb{Q}) \equiv \int_{\Omega} \frac{|p\,P_{\mathrm{a}} - q\,P_{\mathrm{b}}|^{\alpha}}{|p\,P_{\mathrm{a}} + q\,P_{\mathrm{b}}|^{\alpha-1}} \mathrm{d}\lambda, \tag{12}$$

where $0 \leq \alpha \leq \infty$.

Changing the values of the parameters P_{a} and P_{b} enables to obtain either symmetric, or asymmetric dissimilarity. For $P_{\mathrm{a}} = P_{\mathrm{b}}$, one has a symmetric measure, and for $P_{\mathrm{a}} \neq P_{\mathrm{b}}$, the measure is asymmetric. The value of α does not affect the symmetry property of the dissimilarity. Therefore, in our experiments, we have fixed $\alpha = 0.5$.

For a countable space Ω the definition is formulated as follows:

DEFINITION 11 *The Lissack-Fu distance between* $\mathbb{P}$ *and* $\mathbb{Q}$ *on a discrete measurable space* $(\Omega, \mathcal{F})$ *is defined as*

$$d_{\mathrm{LF}}(\mathbb{P}, \mathbb{Q}) \equiv \sum_{i \in \Omega} \frac{|p_i\,P_{\mathrm{a}} - q_i\,P_{\mathrm{b}}|^{\alpha}}{|p_i\,P_{\mathrm{a}} + q_i\,P_{\mathrm{b}}|^{\alpha-1}}. \tag{13}$$

4. k-centroids algorithm with asymmetric dissimilarities

The enhancement to the k-centroids clustering algorithm, proposed in this paper, consists in utilization of the asymmetric measure of dissimilarity in this algorithm. We consider the batch variant of the algorithm. The starting phase of our algorithm is exactly the same as in case of the symmetric k-centroids method. As stated in Section 2, the k-centroids algorithm, essentially, consists of two alternating steps, i.e., Step 1 and Step 2 of Procedure 1 in Section 2.

Step 1. Forming of the clusters: The algorithm iterates over the entire set of objects and assigns each object to the cluster with the nearest centroid. The nearest centroid is determined by the minimal asymmetric dissimilarity between a given object and one of the centroids. Therefore, for each object, the following has to be found:

$$\min_i \; d_{\mathrm{ASYM}}(\mathrm{FE}_{\mathrm{new}}, \mathrm{FE}_{c_i}), \tag{14}$$

where d_{ASYM} is the asymmetric dissimilarity, $\mathrm{FE}_{\mathrm{new}}$ is the vector of features of a given object in the analyzed data set, and FE_{c_i} is the vector of features of the i-th cluster centroid, $i = 1, \ldots, k$.

We apply the definitions of the asymmetric dissimilarities in their discrete forms, since the vectors of features are considered in (14). Moreover, the N-tuples in these definitions (Section 3) are of finite length, since the feature-vectors in (14) are of finite length, as well.

This process can be presented with the following pseudocode:

```
1: for x ∈ X do
2:   min ← MAX_VALUE
3:   for c ∈ centroids do
4:     if min > d_ASYM(x, c) then
5:       min ← d_ASYM(x, c)
6:       x temporarily belongs to cluster cluster(c)
7:     end if
8:   end for
9: end for
```

After the execution of this pseudocode, each object x, from the entire data set X, is allocated to the cluster represented by the centroid – nearest to this object. The *centroids* variable stores the set of all current centroids, $cluster\,(c)$ denotes the cluster with centroid c, min is an auxiliary variable, while the MAX_VALUE is the maximal value of the min variable.

Step 2. Finding centroids for the clusters: Cluster centroids are computed on the basis of minimization of the objective function (1), where a chosen dissimilarity measure $d\,(a, b)$ should be the asymmetric dissimilarity measure $d_{\mathrm{ASYM}}\,(a, b)$. As for optimization technique, we do not focus on this problem, i.e., any minimization technique is allowed. Minimization algorithm used in our experimental study was the complete search. For the numerical simplicity and speed, we have limited the variables space to the points corresponding to the current members of the specific cluster. This means that the search process was carried out in the set of current members of the considered cluster. This kind of approach is sometimes referred to as the k-medoids algorithm.

These two steps are repeated until the termination condition is met. The termination condition is either reaching the convergence of the whole clustering process, i.e., the convergence of the iterative application of the objective function (2) with the asymmetric dissimilarity inserted as the dissimilarity, or reaching the pre-defined number of iterations (Step 1 and 2). The value of the objective function (2) should be computed after each iteration (execution of Steps 1 and 2), in order to track the convergence of the cluster analysis. This conver-

gence, generally, is not guaranteed by the algorithm, as stated in Section 2. Therefore, the second termination condition (maximal number of iterations) secures the algorithm from the infinite execution.

The objective function (2), with the asymmetric dissimilarity, is the criterion optimized by our algorithm.

Application of different asymmetric dissimilarities is possible, however, what needs to be ensured is that both steps (Step 1 and 2) are implemented with the same asymmetric dissimilarity, in order to guarantee the monotone property of the algorithm, and consequently, a reasonable solution returned by the algorithm. In our experiments, we have tested several kinds of asymmetric dissimilarities, in order to evaluate their usefulness in supporting our k-centroids method improvement.

All of the drawbacks of the symmetric k-centroids algorithm, mentioned in Section 2, still hold. Thus, a random choice of the initial centroids of the clusters does not necessarily lead to the proper solution. Furthermore, there is a slight chance that a single run of the algorithm will return a satisfactory result. Therefore, the whole process needs to be replicated, i.e., multistarted with random starts.

Notice that the asymmetric k-centroids algorithm maintains the simplicity of the original algorithm, and does not add computational burden (discussed in more details in Subsection 4.1).

4.1. Computational complexity

Computational complexity of the asymmetric k-centroids algorithm is exactly the same as in case of the symmetric version of the algorithm. Hence, the estimated complexity of Step 1 is $\mathcal{O}(knd)$, where k is the number of clusters, n is the number of objects, and d is the dimensionality (the number of features of each object in the analyzed data set). The complexity of Step 2 depends on minimization technique employed for the clusters centroids computation. Minimization technique, we have employed is based on simple complete search, therefore, it is not computationally efficient, however, the matter of optimization technique is not the subject of our paper, and a simple and numerically - easy method was chosen to do not distract the reader's attention from the main issues of this work. Consequently, in case of our minimization method the estimated computational complexity of Step 2 is $\mathcal{O}\left(\frac{n^2 d}{k}\right)$. Therefore, the complexity of Steps 1 and 2 is $\mathcal{O}\left(\left(k + \frac{n}{k}\right) nd\right)$. And, considering the number of iterations t, the entire estimated computational complexity of the chosen implementation of our algorithm is $\mathcal{O}\left(t\left(k + \frac{n}{k}\right) nd\right)$. However, use of an efficient optimization technique in Step 2 will significantly reduce its complexity, and generally, will lead to computational complexity, of the entire proposed algorithm, of order $\mathcal{O}(tknd)$.

5. Experiments

We have tested the performance of the discussed asymmetric k-centroids clustering algorithm by carrying out the experiments on real data: in the field of signal recognition, i.e., piano music composer clustering, and human heart rhythm clustering. Human heart rhythms are represented with the ECG recordings derived from the MIT-BIH ECG Databases. This clustering process leads to cardiac arrhythmia detection and recognition. Asymmetric k-centroids algorithm was forming clusters representing normal sinusoidal heart rhythm and two types of arrhythmia. Performance of the considered clustering algorithm is evaluated on the basis of the clustering accuracy. We have employed different symmetric, asymmetric, and parametrized dissimilarities presented in Section 3, in order to evaluate their effectiveness in cooperating with the discussed k-centroids clustering algorithm. Consequently, we verify the main assertion of this paper, which is the proposal of applying the asymmetric dissimilarities as more recommended for the k-centroids algorithm.

We report also the experimental results obtained with use of the mixture-model-based clustering method and two classical methods, i.e., the traditional k-means algorithm (Ward's criterion) and the agglomerative hierarchical clustering algorithm, for the purpose of comparing them with the results obtained with use of our asymmetric method. The mixture-model-based clustering method estimates the parameters of statistical mixture models, and forms the clusters corresponding to components of mixtures. In our research, this clustering was carried out using the `MCLUST` Version 3 package (Fraley and Raftery, 2006) for the `R` programming language (`R` Development Core Team, 2010). We have used the Gaussian Mixture Models (GMMs) with parameters estimated via the EM algorithm. Specific covariance structure and the number of components of the mixture were selected using the Bayesian Information Criterion (BIC).

The Ward's criterion refers to the k-means algorithm with centroids of the clusters computed as the arithmetic averages. The agglomerative hierarchical clustering was performed using the single linkage criterion (the nearest distance method) and the Euclidean distance for creating the hierarchical cluster tree.

Finally, we have carried out the experiments on simulated 2-dimensional data, where the convergence of the objective function (2) was considered providing the insight into the algorithm.

In each part of our experiments, the "correct" number of clusters was assumed (i.e., three clusters), except for the GMM-based clustering, where also the BIC criterion was employed for choosing the number of clusters.

5.1. Piano music composer clustering

5.1.1. Experimental setup

In this part of our experiments, we have tested the asymmetric k-centroids algorithm forming three clusters representing three piano music composers: Johann

Sebastian Bach, Ludwig van Beethoven, and Fryderyk Chopin. The numbers of music pieces belonging to each of these composers are given in Table 1. Each music piece was represented with a 20-seconds sound signal sampled with the 44100 Hz frequency. The entire data set was composed of 32 sound signals. The feature extraction process was carried out according to the Digital Fourier Transform based method.

5.1.2. Experimental results

The results of this part of our experiments are shown in Table 1, presenting the accuracy degree of clustering with our k-centroids algorithm cooperating with different symmetric, asymmetric, and parameterized dissimilarities. Table 1 shows, also, clustering results obtained with use of the GMM-based method and two classical methods, i.e., the traditional k-means algorithm (Ward's criterion) and the agglomerative hierarchical clustering algorithm. The numbers 1 and 2 given with each asymmetric dissimilarity denote this dissimilarity computed in two different directions, i.e., $d_{\mathrm{ASYM}}1 = d_{\mathrm{ASYM}}(p,q)$ (i.e. computed in such way that the dissimilarity from centroid to object is greater than from object to centroid) and $d_{\mathrm{ASYM}}2 = d_{\mathrm{ASYM}}(q,p)$. The asymmetric Chernoff distance was obtained by applying parameter $\alpha = 0.9$, while the symmetric Chernoff distance was obtained with $\alpha = 0.5$. The asymmetric Lissack-Fu distance, in turn, was obtained by applying parameters $P_{\mathrm{a}} = 0.5$ and $P_{\mathrm{b}} = 1.0$, while the symmetric form of this quantity was obtained with the $P_{\mathrm{a}} = 1.0$ and $P_{\mathrm{b}} = 1.0$. Fig. 1 presents the BIC values for different covariance structures and numbers of components of GMM. For details on the covariance structures see Fraley and Raftery (2006).

The accuracies were calculated on the basis of the following accuracy degree:

$$a_i \equiv \frac{x^i_{\max}}{N_i}, \tag{15}$$

where a_i, $i = 1,2,3$, is the accuracy degree for the i-th composer, $x^i_{\max}$, $i = 1,2,3$, is the maximal number of music pieces of the i-th composer in any of the clusters, N_i, $i = 1,2,3$, is the total number of music pieces of the i-th composer.

Once the accuracy degree for the i-th composer is calculated, the corresponding cluster is not considered in calculations of the accuracy degrees for the remaining composers.

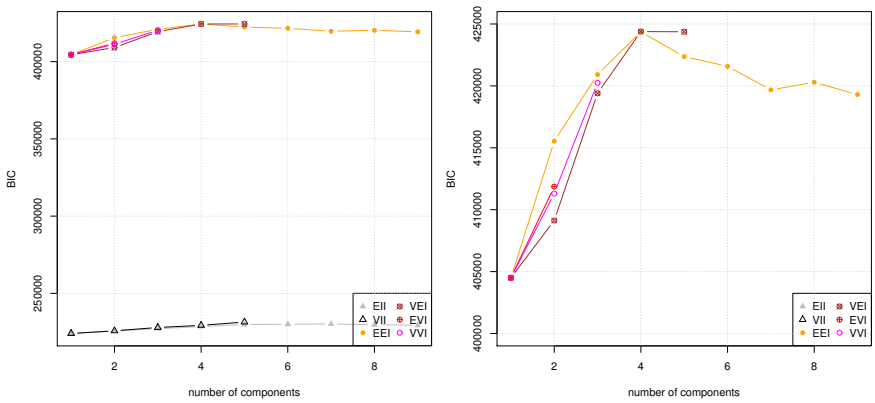
Each row with the accuracy entries ends with the average accuracy degree, estimating the quality of each clustering approach. It is the arithmetic average of all three accuracy degrees associated with all three composers:

$$a_{\mathrm{average}} \equiv \frac{a_1 + a_2 + a_3}{3}. \tag{16}$$

This average accuracy degree is used as the basis of comparison between investigated approaches.

Table 1. Accuracies of Piano Music Composer Clustering

	Bach	Beethoven	Chopin	Average Accuracy
Number of Signals	11	12	9	
Kullback-Leibler Divergence 1	0.818	0.750	0.778	**0.781**
Kullback-Leibler Divergence 2	0.727	0.667	0.778	**0.719**
Asymmetric Chernoff Distance 1	0.818	0.750	0.778	**0.781**
Symmetric Chernoff Distance	0.727	0.750	0.778	**0.750**
Asymmetric Chernoff Distance 2	0.727	0.667	0.778	**0.719**
Asymmetric Lissack-Fu Distance 1	0.818	0.750	0.778	**0.781**
Symmetric Lissack-Fu Distance	0.727	0.750	0.778	**0.750**
Asymmetric Lissack-Fu Distance 2	0.727	0.667	0.778	**0.719**
Hellinger Distance	0.727	0.750	0.778	**0.750**
Total Variation Distance	0.636	0.750	0.778	**0.719**
Euclidean Distance	0.818	0.583	0.556	**0.656**
GMM-based Clustering	0.636	1.000	0.778	**0.805**
Traditional K-Means	0.909	0.250	0.778	**0.625**
Hierarchical Clustering	0.636	0.500	0.778	**0.638**



(a) BIC for all models (b) BIC for the best models

Figure 1. The BIC values for various covariance structures and numbers of components of GMM, where EII is the spherical-distributional equal-volume equal-shape covariance structure, VII is the spherical-distributional variable-volume equal-shape covariance structure, EEI is the diagonal-distributional equal-volume equal-shape covariance structure, VEI is the diagonal-distributional variable-volume equal-shape covariance structure, EVI is the diagonal-distributional equal-volume variable-shape covariance structure, and VVI is the diagonal-distributional variable-volume variable-shape covariance structure

5.1.3. Discussion

Table 1 shows that clustering with the asymmetric k-centroids algorithm allowed for obtaining superior results than with the original version of the algorithm. What is worth noting, is the fact that the clustering performance strongly depends on the direction of asymmetry in case of the asymmetric dissimilarities, i.e., whether we consider $d_{\mathrm{ASYM}}(p, q)$ or $d_{\mathrm{ASYM}}(q, p)$. This is consistent with the justification of our proposal, according to which the dissimilarity from centroid to object should be greater than in the opposite direction. And, in this case, our method appeared superior over the symmetric one. We have checked the other "incorrect" direction of asymmetry only to confirm the justification of our method. Moreover, our method outperformed two investigated classical clustering methods: k-means and the hierarchical clustering method. Note that the hierarchical clustering method uses hierarchy only as a technique for generating the clustering tree, and does not regard the hierarchical asymmetric relations in data, when a symmetric dissimilarity (e.g., the Euclidean distance) is applied. Therefore, it should not be surprising that it produces the inferior results. An open problem is application of the asymmetric dissimilarities in hierarchical clustering algorithms.

On the other hand, the GMM-based method with the enforced number of components (i.e., number of clusters) to three, and the VEI model providing the highest clustering accuracy among all GMMs available in `MCLUST`, returned the result superior over our method. The GMM-based method for the non-spherical covariance structures is more advanced and sophisticated than the k-centroids-type approaches, therefore, such result should not be surprising. However, each non-spherical-distributional model leads to higher computational complexity than the spherical ones (which lead to the k-means criterion). Estimating of more complex covariance structures is, naturally, more computationally expansive. Specifically, the three-component VEI model (with the covariance matrix $\Sigma_k = \lambda_k A$) requires k times greater complexity than our method, see Subsection 4.1 and Fraley and Raftery, 2006).

The experiments with automatic model selection showed that according to the BIC-based model selection, the best model is the VEI model with four components (Fig. 1). Hence, the choice of the components' number was slightly missed in context of our knowledge about the data.

5.2. Human heart rhythm clustering

5.2.1. Experimental setup

In this part of our experiments, we have investigated our algorithm forming three clusters representing three types of human heart rhythms: normal sinus rhythm, atrial arrhythmia, and ventricular arrhythmia. This kind of clustering can be recognized as cardiac arrhythmia detection and recognition based on the ECG recordings. In general, the cardiac arrhythmia disease may be classified

either by rate (tachycardias – the heart beat is too fast, and bradycardias – the heart beat is too slow), or by site of origin (atrial arrhythmias – they begin in the atria, and ventricular arrhythmias – they begin in the ventricles). Our clustering recognizes the normal rhythm, and also, recognizes arrhythmias originating in the atria, and in the ventricles. We have analyzed 20-minutes ECG holter recordings sampled with the 250 Hz frequency. The entire data set was composed of 63 ECG signals. The numbers of recordings belonging to each rhythm type are given in Table 2. The feature extraction process was carried out in the same way as in case of the piano music composer clustering.

5.2.2. Experimental results

The results of this part of our experiments are shown in Table 2, constructed in the same way as Table 1, and Fig. 2 presents the BIC values for different covariance structures and numbers of components of GMM.

5.2.3. Discussion

Table 2 shows results very similar to the results of the previous part of our experiments. The same effect can be observed due to the direction of asymmetry, in case of the asymmetric dissimilarities. In one of the directions of asymmetry (the "correct" direction), the asymmetric dissimilarities outperform symmetric ones, while in the other direction (the "incorrect" direction), they provide lower clustering performance. This behavior was observed in case of both parts of our experiments on real data. Table 2 reports also the superiority of our approach over two tested classical clustering methods, just like in the previous subsection. Again, the VEI mixture model, with the number of components set to three, outperformed our method, and according to the BIC criterion the best model is the VEI model with four components (Fig. 2). Naturally, our expectation was three components.

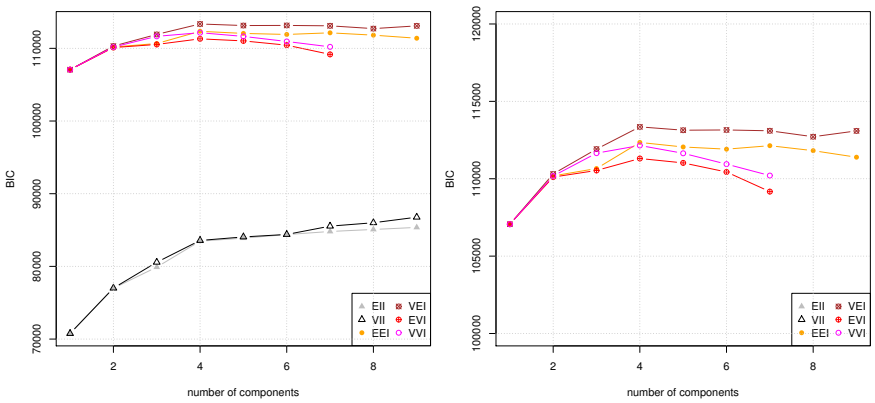
5.3. Simulated 2-dimensional data clustering

5.3.1. Experimental setup

In the last part of our experiments, we have carried out clustering of the simulated points in the 2-dimensional (x, y) space. The entire analyzed set consisted of 60 points. The points are shown in Figs. 3(a)-10(a). Their locations were chosen randomly from the uniform distribution with the spread ± 20 units in both dimensions around the chosen central points $((15, 90), (80, 20), (150, 150))$, and then the coordinates were normalized. In this way, we intended to gather the points around the chosen central points, in order to form certain divisions in the analyzed data set. However, after normalization, the divisions were not very clear so as to enable verification of effectiveness of investigated algorithms. We have normalized the features, just like in the experiments on real data in the previous subsections. Therefore, all points in our data set lay on the axis

Table 2. Accuracies of Human Heart Rhythm Clustering

	Normal Rhythm	Atrial Arrhythmia	Ventricular Arrhythmia	Average Accuracy
Number of Signals	18	23	22	
Kullback-Leibler Divergence 1	0.944	0.783	0.773	**0.825**
Kullback-Leibler Divergence 2	0.944	0.826	0.636	**0.794**
Asymmetric Chernoff Distance 1	0.944	0.826	0.773	**0.841**
Symmetric Chernoff Distance	0.944	0.826	0.727	**0.825**
Asymmetric Chernoff Distance 2	0.944	0.826	0.636	**0.794**
Asymmetric Lissack-Fu Distance 1	0.944	0.826	0.773	**0.841**
Symmetric Lissack-Fu Distance	0.944	0.826	0.727	**0.825**
Asymmetric Lissack-Fu Distance 2	1.000	0.826	0.636	**0.810**
Hellinger Distance	0.944	0.783	0.727	**0.810**
Total Variation Distance	0.944	0.826	0.682	**0.810**
Euclidean Distance	0.833	0.739	0.636	**0.730**
GMM-based Clustering	0.636	1.000	1.000	**0.879**
Traditional K-Means	0.889	0.435	0.773	**0.730**
Hierarchical Clustering	0.636	0.583	0.889	**0.703**



(a) BIC for all models (b) BIC for the best models

Figure 2. The BIC values for various covariance structures and numbers of components of GMM, where EII is the spherical-distributional equal-volume equal-shape covariance structure, VII is the spherical-distributional variable-volume equal-shape covariance structure, EEI is the diagonal-distributional equal-volume equal-shape covariance structure, VEI is the diagonal-distributional variable-volume equal-shape covariance structure, EVI is the diagonal-distributional equal-volume variable-shape covariance structure, and VVI is the diagonal-distributional variable-volume variable-shape covariance structure

$x+y-1=0$. Additionally, we have analyzed the values of the objective function (2) computed in ten iterations of the single k-centroids cycle (Steps 1 and 2, described in Section 4).

5.3.2. Experimental results

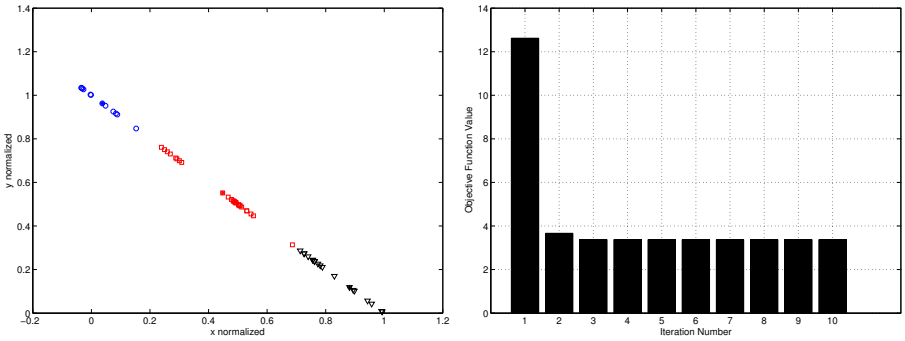
The results of this part of our experiments are demonstrated with Figs. 3-10. Figures 3(a)-10(a) present the generated clusters, while Figs. 3(b)-10(b) show the corresponding values of the objective function (2) for ten iterations of the k-centroids cycle. The clustering results are presented graphically, where the clusters obtained with the investigated algorithms are shown with different shapes (squares, circles, and triangles). Clusters centroids are marked with the asterisk character.

The quality of clustering with symmetric and asymmetric dissimilarities was evaluated on the basis of the objective function (2) final values, i.e. when the function (2) converged. The objective function (2) is the criterion minimized by the k-centroids algorithm, therefore, it can be used for our clustering performance evaluation. Additionally, the values of (2) computed in ten iterations of the single k-centroids cycle allow for tracking of the convergence speed of (2), providing the insight into the algorithm. Just like in the previous experiments, the numbers 1 and 2 given with the asymmetric dissimilarities denote the specific dissimilarity computed in two different directions.

5.3.3. Discussion

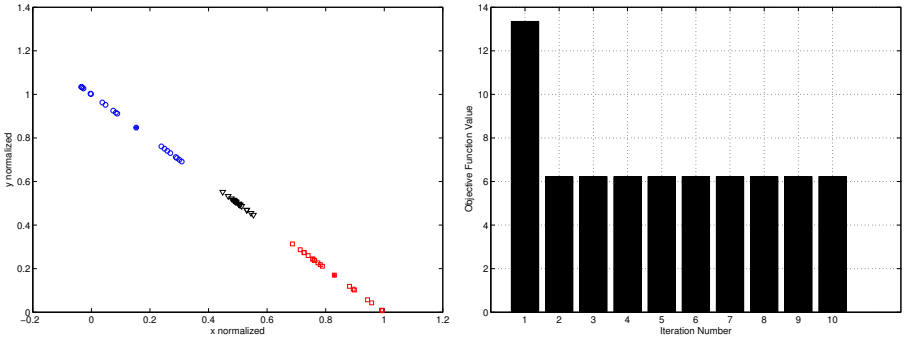
The experiments on simulated data confirmed our observations and conclusions referring to the previous parts of our empirical study on real data. The asymmetric dissimilarities outperform their symmetric counterparts. Analysis of the objective function (2) values in ten iterations of the k-centroids cycle aims to determine the clustering quality of each of tested methods, and verify the convergence of (2). The clustering quality can be assessed on the basis of the final value of (2). The lower is that value, the better is the clustering result.

The objective function (2) converges within ten iterations of the k-centroids cycle in all tested cases. Fig. 6(b) shows the slowing down of the convergence of (2), however, sometimes the problem can be more serious, and, the convergence of (2) may be disrupted, i.e., the objective function may not converge at all to an optimal or near-optimal solution (discussed in Sections 2 and 4). Nevertheless, according to our results, the asymptotic behavior of (2) does not differ for the asymmetric and symmetric dissimilarities. Therefore, regarding the fact of higher clustering accuracy, we conclude that the k-centroids clustering with asymmetric dissimilarities can be considered as superior over the same algorithm with symmetric counterparts.



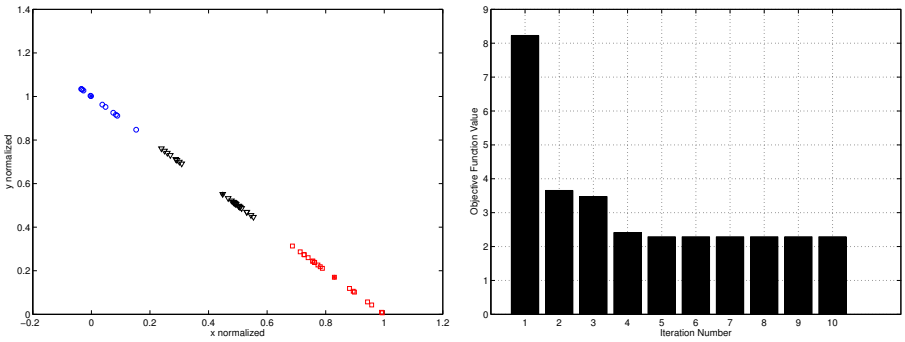
(a) Clusters (b) Objective function

Figure 3. Clustering results for the Kullback-Leibler divergence 1



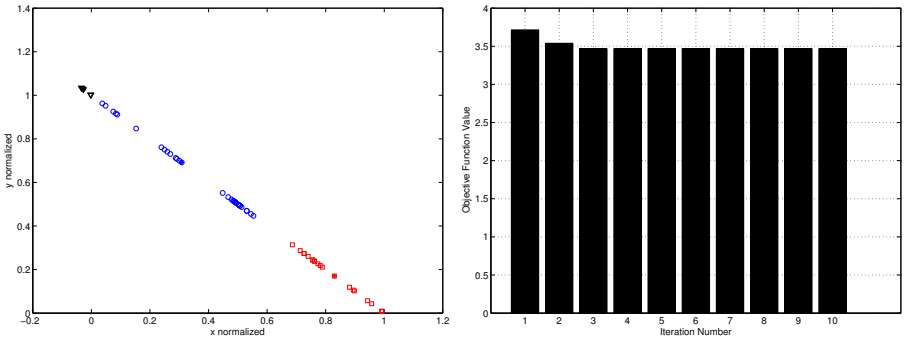
(a) Clusters (b) Objective function

Figure 4. Clustering results for the Kullback-Leibler divergence 2



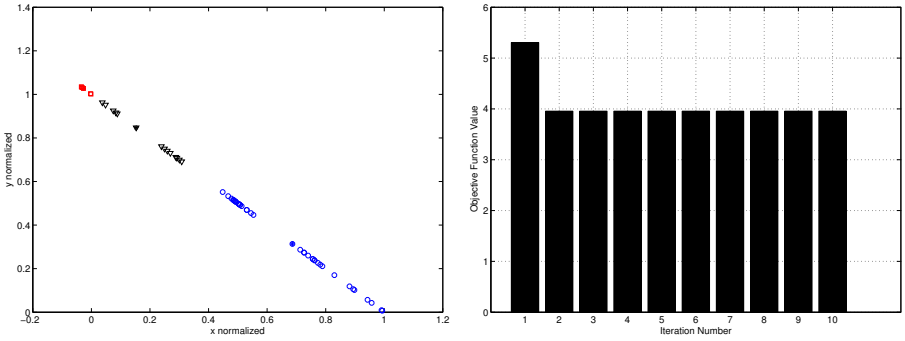
(a) Clusters (b) Objective function

Figure 5. Clustering results for the asymmetric Lissack-Fu distance 1



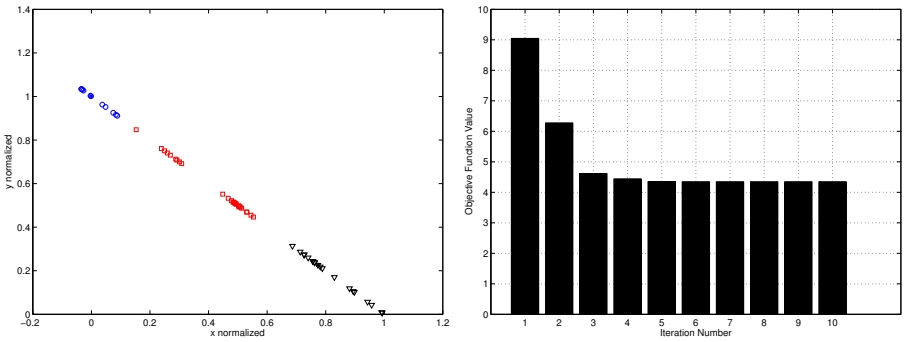
(a) Clusters (b) Objective function

Figure 6. Clustering results for the symmetric Lissack-Fu distance



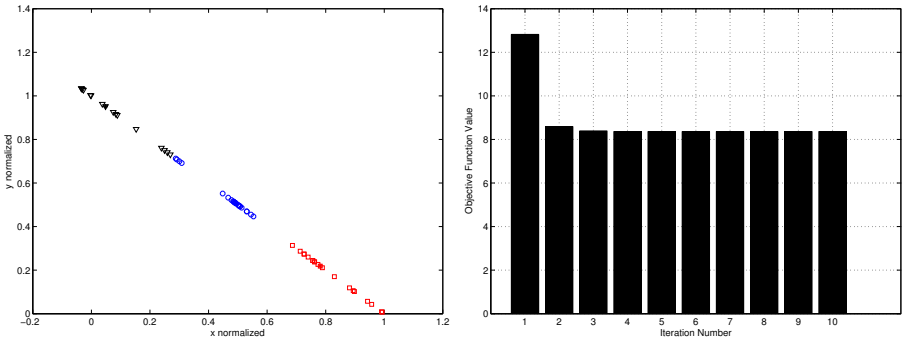
(a) Clusters (b) Objective function

Figure 7. Clustering results for the asymmetric Lissack-Fu distance 2



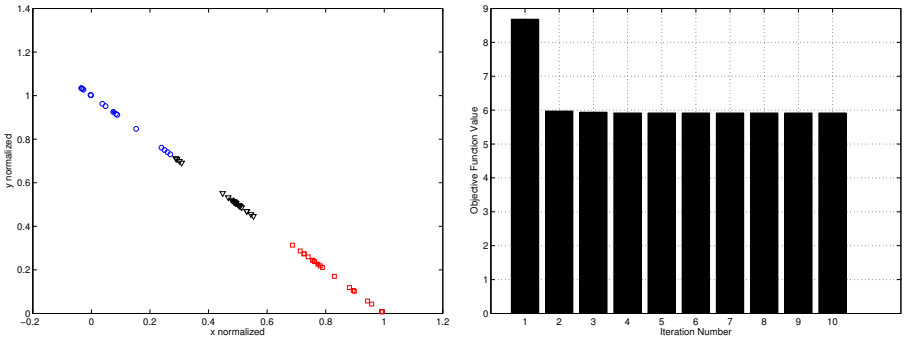
(a) Clusters (b) Objective function

Figure 8. Clustering results for the Hellinger distance



(a) Clusters (b) Objective function

Figure 9. Clustering results for the total variation distance



(a) Clusters (b) Objective function

Figure 10. Clustering results for the Euclidean distance

6. Summary and concluding remarks

This paper presented an improvement to the k-centroids clustering algorithm. We proposed application of the asymmetric dissimilarities, in this algorithm, as more consistent with the behavior of the algorithm than the most commonly employed symmetric dissimilarities, e.g., the Euclidean distance. We claim that asymmetric measures are more suitable for the k-centroids technique, because this clustering method evaluates the dissimilarity between two distinct entities (object vs. cluster centroid). The k-centroids algorithm consists of two alternating steps (Section 2), and, in both these steps, the dissimilarity is used between the entities of different nature: clusters centroids, being the privileged dominant points, and single objects "attracted" by the centroids. In other words, the k-centroids method is based on asymmetric relationships. Therefore, employing of the asymmetric measures tends to fit this property. Consequently, we

wanted to assert that asymmetric dissimilarities, in certain areas of research, can be regarded as superior over their symmetric counterparts, on the contrary to the frequent opinion, considering them as the mathematically inconvenient quantities.

Acknowledgments. The author wishes to thank the anonymous reviewers for their helpful and constructive suggestions, which have importantly improved the quality of this paper.

References

Chaturvedi, A., Carroll, J.D., Green, P.E., Rotondo, J.A. (1997) A Feature-based Approach to Market Segmentation via Overlapping K-Centroids Clustering. *Journal of Marketing Research* **34** (3), 370–377.

Chernoff, H. (1952) A Measure of Asymptotic Efficiency for Tests of a Hypothesis Based on the Sum of Observations. *The Annals of Mathematical Statistics* **23** (4), 493–507.

Czekala, M. and Kuziak, K. (1999) Clustering of Stocks in Risk Context. In: *Classification in the Information Age. Studies in Classification, Data Analysis, and Knowledge Organization*. Springer-Verlag, Berlin, 447–452.

Fraley, C. and Raftery, A.E. (2006) `MCLUST` version 3 for `R`: Normal Mixture Modeling and Model-Based Clustering. Tech. Rep. No. 504. Department of Statistics, University of Washington.

Gibbs, A.L. and Su, F.E. (2002) On Choosing and Bounding Probability Metrics. *International Statistical Review* **70** (3), 419–435.

Kanungo, T., Mount, D.M., Netanyahu, N.S. and Piatko, C.D., Silverman, R., Wu, A.Y. (2002) An Efficient k-Means Clustering Algorithm: Analysis and Implemetation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24** (7), 881–892.

Kullback, S. and Leibler, R.A. (1951) On Information and Sufficiency. *The Annals of Mathematical Statistics* **22** (1), 79–86.

Leisch, F. (2006) A Toolbox for K-Centroids Cluster Analysis. *Computational Statistics & Data Analysis* **51**(2), 526–544.

Lissack, T. and Fu, K.S. (1976) Error Estimation in Pattern Recognition via L^{α}- Distance Between Posterior Density Functions. *IEEE Transactions on Information Theory* **IT-22** (1), 34–45.

MacQueen, J. (1967) Some Methods for Classification and Analysis of Multivariate Observations. In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*. **1**, 281–297.

Martín-Merino, M. and Muñoz, A. (2005) Visualizing Asymmetric Proximities with SOM and MDS Models. *Neurocomputing* **63**, 171–192.

Muñoz, A., Martin, I. and Moguerza, J.M. (2003) Support Vector Machine Classifiers for Asymmetric Proximities. In: *Artificial Neural Networks and Neural Information Processing - ICANN/ICONIP 2003, Joint*

International Conference ICANN/ICONIP 2003. **LNCS 2714**, Springer-Verlag, 217–224.

Okada, A. (2000) An Asymmetric Cluster Analysis Study of Car Switching Data. In: *Data Analysis. Studies in Classification, Data Analysis, and Knowledge Organization*. Springer, Berlin-Heidelberg.

Okada, A. and Imaizumi, T. (1997) Asymmetric Multidimensional Scaling of Two-Mode Three-Way Proximities. *Journal of Classification* **14** (2), 195–224.

Okada, A. and Imaizumi, T. (2007) Multidimensional Scaling of Asymmetric Proximities with a Dominance Point. In: *Advances in Data Analysis. Studies in Classification, Data Analysis, and Knowledge Organization*. Springer, Berlin-Heidelberg, 307–318.

Olszewski, D. (2011) Asymmetric k-Means Algorithm. In: *Adaptive and Natural Computing Algorithms - 10th International Conference, ICANNGA 2011*. **LNCS 6594**, Springer, 1–10.

Olszewski, D., Kolodziej, M. and Twardy, M. (2010) A Probabilistic Component for K-Means Algorithm and its Application to Sound Recognition. *Przeglad Elektrotechniczny* **86** (6), 185–190.

R Development Core Team (2010) R: *A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Selim, S.Z. and Ismail, M.A. (1984) K-Means-Type Algorithms: A Generalized Convergence Theorem and Characterization of Local Optimality. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-6** (1), 81–87.

Steinhaus, H. (1956) Sur la Division des Corps Matériels en Parties. *Bulletin de l'Académie Polonaise des Sciences, C1. III* **4** (12), 801–804.

Xiong, H., Wu, J. and Chen, J. (2009) K-Means Clustering Versus Validation Measure: A Data-Distribution Perspective. *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics* **39** (2), 318–331.