Jagiellonian University
Faculty of Mathematics and Computer Science

k-means Approach to the Karhunen-Loéve Transform

IM UJ preprint 2012/01

K. Misztal, P. Spurek, J. Tabor

k-means Approach to the Karhunen-Loéve Transform

K. Misztal

Faculty of Mathematics and Computer Science, Jagiellonian University, Łojasiewicza 6, 30-348 Kraków, Poland

P. Spurek

Faculty of Mathematics and Computer Science, Jagiellonian University, Łojasiewicza 6, 30-348 Kraków, Poland

J. Tabor

Faculty of Mathematics and Computer Science, Jagiellonian University, Łojasiewicza 6, 30-348 Kraków, Poland

Abstract

We present a simultaneous generalization of the well-known Karhunen-Loéve (PCA) and k-means algorithms. The basic idea lies in approximating the data with k affine subspaces of a given dimension n. In the case $n = 0$ we obtain the classical k-means, while for $k = 1$ we obtain PCA algorithm. Moreover, by our approach we can obtain clusters with different dimensionality which describe the structure of date.

We show that for some data exploration problems this method gives better result then either of the classical approaches.

Keywords:
Karhunen-Loéve Transform, PCA, k-Means, optimization, compression, data compression, image compression.

1. Introduction

Our general problem concerns splitting of a given data-set W into clusters with respect to their intrinsic dimensionality. The motivation to create such an algorithm is a desire to cluster a high dimensional data. More precisely, we present possible approach to dealing with *the curse of dimensionality* [1, 2, 3] by finding affine subspaces $S_1, \ldots, S_k$ such that every element of W belongs (with certain error) to one of the spaces $S_1, \ldots, S_k$. To explain it graphically, let us consider following example. Figure 1(a) represents three lines in the plane, while Figure 1(b) a circle and an orthogonal line in the space. Our goal is to construct an algorithm that will split them into three lines and into a line and a circle.

In recent years various type of algorithms, which try to deal with this problem, were presented. The most important are Subspace Clustering, Pattern-Based Clustering, and Correlation Clustering [4, 5]. All of them try to simultaneously deal with two important problems. The first is the search for the relevant subspaces and the second focuses on the detection of the final clusters. The second group of algorithms, which try to solve the problem of *the curse of dimensionality*, are kernel and spectral methods [6, 7, 8, 9, 10]. This algorithms, by use of the spectrum of the similarity matrix[1] of the data, try to reduce the dimensionality of clusters.

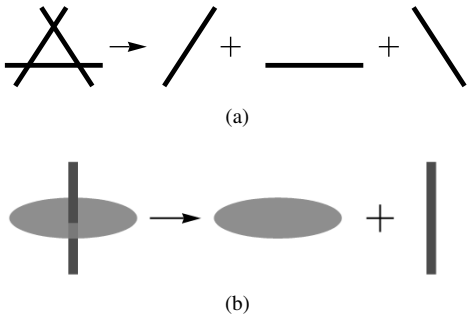
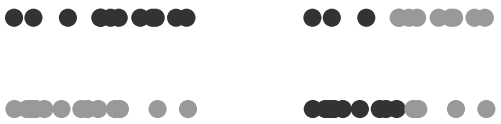


Figure 1: Our goal is to create algorithm which will interpret Fig. 1(a) as three groups of one-dimensional points and Fig. 1(b) as two groups of one- and two-dimensional points.

In this paper we present the method of clustering which is able to extract groups of points which represent affine subspaces of different dimension. We have constructed a simultaneous generalization of the k-means method [11] and the Karhunen-Loéve transform (called also PCA – Principle Component Analysis) [12] – we call it (ω, k)-means. Instead of finding k centers which best represent the data as in the classical k-means, we find k nearest subspaces. Thanks to weight parameter $\omega = (\omega_1, \ldots, \omega_n)$ we are able to restrict our algorithm to some class of affine subspaces (the role of ω, is described in next sections, see Remark 2.5, Example 5.2, Example 5.3). In analogy to the case of k-means, we obtain a version of the Voronoi diagram (see next section). In the simplest form our algorithm needs the number of clusters k and the weight parameter ω (for

Email addresses: krzysztof.misztal@ii.uj.edu.pl (K. Misztal), przemyslaw.spurek@ii.uj.edu.pl (P. Spurek), jacek.tabor@ii.uj.edu.pl (J. Tabor)

[1] For given a dataset W, the similarity matrix may be defined as a matrix S, where S_{ij} represents a measure of the similarity between points of W



(a) (b)

Figure 2: Example of clustering: Fig. 2(a) for $k = 2$ clusters which are 1–dimensional; Fig. 2(b) for classical k-means with $k = 2$ clusters.

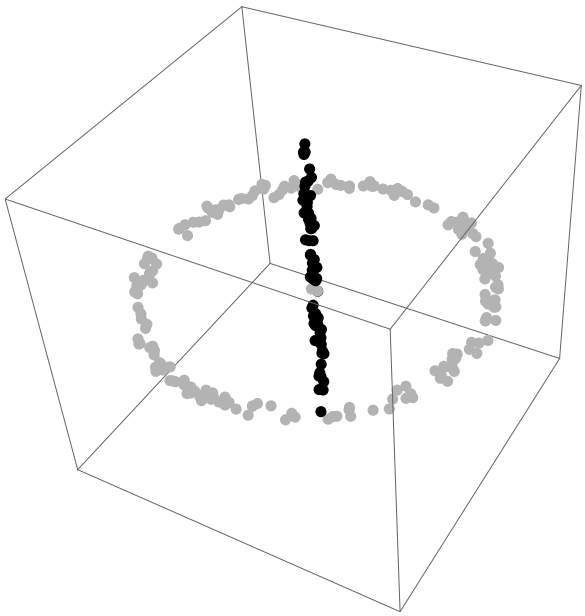


Figure 3: Clustering with (ω, k)-means for $\omega = (0, 0, 1)$.

$\omega = (1, 0, \ldots, 0)$ we obtain the k-means while for $k = 1$ we obtain the PCA).

To present our method, consider the points grouped along two parallel lines – Figure 2 shows the result of our program on the clustering of this set. Another example is given by the dataset consisting of two groups of points along circle and interval, see Figure 3. As an outcome of our algorithm, we obtain two clusters. The first contains the points group along the internal, which represents one dimensional data (in compression we need only one parameter for each point) and the second one, which represents two dimensional data (in this time we need two parameters to describe each point). Further discussion of this example we present in Section 5.

The approach can be clearly used in most standard applications of either the k-means or the Karhunen-Loéve transform. In particular, (since it is a generalization of the Karhunen-Loéve transform [13]) one of the possible natural applications of our method lies in the image compression. Figure 4 presents error[2] in image reconstruction of a classical Lena photo (508×508 pixels) as a function of k. Observe that just by modifying the number of clusters from 1 to 3, which makes the minimal increase in the necessary memory, we decrease twice the level of error in the compression.

[2]By error in image comparison we understand pixel by pixel image compare using standard Euclidean norm.

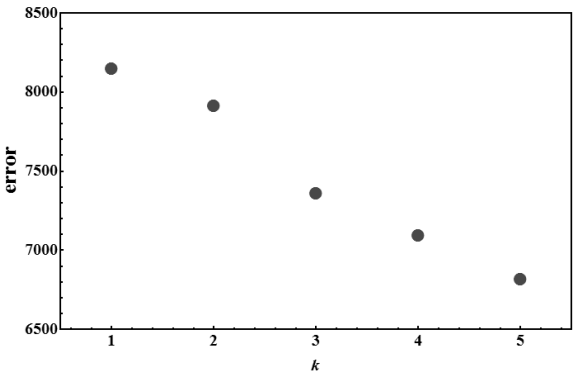


Figure 4: Error in image decompression as a function of number of clusters k for $n = 5$.

Except for image compression our method can by applied in various situations where the classical k-means or PCA where used, for example in:

- data mining – we can detect important coordinates and subsets with similar properties;
- clustering – our modification of k-means can detect different, high dimensional relation in data;
- image compression and image segmentation;
- pattern recognition – thanks to detection of relation in data we can use it to assign data to defined before classes.

2. Generalized Voronoi Diagram

The Voronoi diagram is one of the most useful data structures in computational geometry, with applications in many areas of science [14]. For the convenience of the reader and to establish the notation we shortly describe the classical version of the Voronoi diagram (for more details see [15]). For $N \in \mathbb{N}$ consider $\mathbb{R}^N$ with the standard Euclidean distance and let S be a finite set of $\mathbb{R}^N$. For $p, q \in S$ such, that $p \neq q$, let

$$B(p, q) = \{z \in \mathbb{R}^N : \|p - z\| = \|q - z\|\}, \tag{1}$$

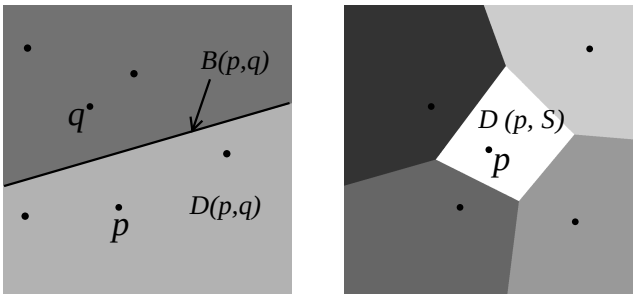
$$D(p, q) = \{z \in \mathbb{R}^N : \|p - z\| < \|q - z\|\}. \tag{2}$$

Hyperplane $B(p, q)$ divides $\mathbb{R}^N$ into two sets, one containing points which are closer to point p then q ($D(p, q)$), and the second one containing points which are closer to point q then p ($D(q, p)$) – see Figure 5(a).

Definition 2.1 ([15])**.** *The set*

$$D(p, S) := \bigcap_{q \in S \,:\, q \neq p} D(p, q)$$

of all points that are closer to p than to any other element of S is called the (open) Voronoi region of p with respect to S.



(a) (b)

Figure 5: Graphical presentation of $B(p,q)$, $D(p,q)$ and $D(p,S)$ in $\mathbb{R}^2$.

For $N = 2$ set $D(p,S)$ is the interior of a convex, possibly unbounded polygon (Figure 5(b)).

The points on the contour of $D(p,S)$ are those that have more than one nearest neighbour in S, one of which is p.

Definition 2.2 ([15])**.** *The union*

$$V(S) := \bigcup \partial D(p,S)$$

of all region boundaries is called the Voronoi diagram of S.

The common boundary of two Voronoi regions is a Voronoi edge. Two edges meet at a Voronoi vertex such a point has three or more nearest neighbours in the set S.

Now we proceed to the description of our modification of the Voronoi diagram. We divide the space $\mathbb{R}^N$ with respect to affine subspaces of $\mathbb{R}^N$.

Definition 2.3. *For* $n \leq N$ *let*

$$\mathrm{E}_n(\mathbb{R}^N) := \{(v_0,\dots,v_n) \in (\mathbb{R}^N)^{n+1} \text{ such that } v_i, v_j \text{ are orthonormal for } i,j > 0, i \neq j\}.$$

Thus v_0 denotes a center of affine space we consider, while $v_1,\dots,v_n$ is the orthonormal base of its "vector part". From the geometrical point of view the element $\mathrm{v} = (v_0, v_1, \dots, v_n) \in \mathrm{E}_n(\mathbb{R}^N)$ represents the affine space

$$v_0 + \mathrm{lin}(v_1,\dots,v_n) = \mathrm{aff}(v_0, v_1, \dots, v_n)$$

or more precisely coordinate system of the affine subspace. We modify equations (1) and (2), by using distance between a point and affine subspace generated by linear independence vector instead of distance between points.

Definition 2.4. *Let* $n < N$ *and let* $\mathrm{v} \in \mathrm{E}_n(\mathbb{R}^N)$, $\omega = (\omega_0,\dots,\omega_n) \in [0,1]^{n+1}$ *such that* $\sum_{j=0}^{n} \omega_j = 1$ *be given. For* $x \in \mathbb{R}^N$ *let*

$$\mathrm{DIST}_\omega(x;\mathrm{v}) := \left(\sum_{j=0}^{n} \omega_j \,\mathrm{dist}(x;\mathrm{aff}(v_0,\dots,v_j))^2\right)^{1/2}, \quad (3)$$

where $\mathrm{dist}(x;V)$ *denotes the distance of the point* x *from the space* V.

In formula 3, $\omega = (\omega_0,\dots,\omega_n)$ is interpreted as vector of weights, where ω_k denotes the weight of the affine subspace of dimension k.

Remark 2.5. *It is easy to notice, that* DIST *has following properties:*

- *for* $\mathrm{v} \in \mathrm{E}_n(\mathbb{R}^N)$ *and* $\omega = (0,\dots,0,1) \in [0,1]^{n+1}$ *we obtain that* $\mathrm{DIST}_\omega(x;\mathrm{v})$ *is a distance between the point* x *and affine space* $\mathrm{aff}(\mathrm{v})$*;*
- *if* $\mathrm{v}_0 = 0$ *and* $\omega = (0,\dots,0,1)$ *then* DIST_ω *is a distance between point and linear space generated by* $(v_1,\dots,v_n)$*;*
- *if* $\omega = (1,0,\dots,0)$ *then* DIST_ω *is the classical distance between* x *and* v_0*:*

$$\mathrm{DIST}_\omega(x;\mathrm{v}) = \|x - v_0\|.$$

- *if* $\omega = \left(\underbrace{0,\dots,0}_{k}, \underbrace{\frac{1}{l-k},\dots,\frac{1}{l-k}}_{k-l}, 0,\dots,0\right)$ *for* $k < l$*, then* DIST_ω *describes the mean distance between* x *and subspaces of dimension from* k *to* l*.*

Remark 2.6. *Formula 3 can be computed as follows*

$$\begin{aligned}(\mathrm{DIST}_\omega(x;\mathrm{v}))^2 &= \sum_{j=0}^{n} \omega_j \left(\|x - v_0\|^2 - \sum_{i=1}^{j} \langle x - v_0; v_i\rangle^2\right)\\ &= \sum_{j=0}^{n} \omega_j \|x - v_0\|^2 - \sum_{j=0}^{n} \omega_j \sum_{i=1}^{j} \langle x - v_0; v_i\rangle^2.\end{aligned}$$

To optimize calculations we define

$$\begin{aligned}\bar{v}_1 &= \langle x - v_0; v_1\rangle^2,\\ \bar{v}_j &= \bar{v}_{j-1} + \langle x - v_0; v_j\rangle^2,\end{aligned}$$

and since $\sum \omega_j = 1$ *thus we simplify our computation to*

$$(\mathrm{DIST}_\omega(x;\mathrm{v}))^2 = \|x - v_0\|^2 - \sum_{j=0}^{n} \omega_j \bar{v}_j.$$

Now we are ready to define our generalization of the Voronoi diagram. Let S be a finite subset of $\mathrm{E}_n(\mathbb{R}^N)$ and $\omega \in [0,1]^{n+1}$, $\sum \omega_j = 1$, where $n \leq N$. For $\mathrm{p},\mathrm{q} \in S$ such, that $\mathrm{p} \neq \mathrm{q}$, let

$$B_\omega(\mathrm{p},\mathrm{q}) := \{z \in \mathbb{R}^N : \mathrm{DIST}_\omega(z;\mathrm{p}) = \mathrm{DIST}_\omega(z;\mathrm{q})\},$$

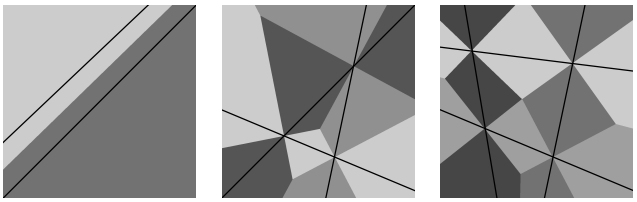
$$D_\omega(\mathrm{p},\mathrm{q}) := \{z \in \mathbb{R}^N : \mathrm{DIST}_\omega(z;\mathrm{p}) < \mathrm{DIST}_\omega(z;\mathrm{q})\}.$$

The set $B_\omega(\mathrm{p},\mathrm{q})$ divides the space $\mathbb{R}^N$ into two sets, first containing points which are closer to p then to q ($D_\omega(\mathrm{p},\mathrm{q})$) and second contain points which are closer to q then p ($D_\omega(\mathrm{q},\mathrm{p})$).

Definition 2.7. *Let* $n \in \mathbb{N}$, $n < N$ *be fixed. Let* S *be finite subset of* $\mathrm{E}_n(\mathbb{R}^N)$ *and* $\omega \in [0,1]^{n+1}$, $\sum_{j=0}^{n} \omega_j = 1$ *be given. For* $\mathrm{p} \in S$ *the set*

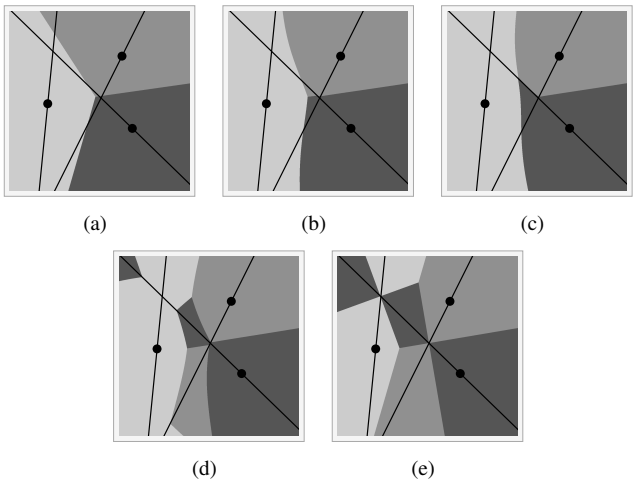
$$D_\omega(\mathrm{p},S) := \bigcap_{\mathrm{q}\in S\,:\,\mathrm{q}\neq\mathrm{p}} D_\omega(\mathrm{p},\mathrm{q})$$

of all points that are closer to p *than to any other element of* S *is called the (open) generalized Voronoi region of* p *with respect to* S.



(a) two lines (b) three lines (c) four lines

Figure 6: Generalized Voronoi diagram for $\omega = (0, 1)$ and two, three and four lines on plane.



(a) (b) (c) (d) (e)

Figure 7: Generalized Voronoi diagram for clustering of 3 clusters for different weight vectors Fig. 7(a), $\omega = (1, 0)$; Fig. 7(b), $\omega = (\frac{3}{4}, \frac{1}{4})$; Fig. 7(c), $\omega = (\frac{1}{2}, \frac{1}{2})$; Fig. 7(d), $\omega = (\frac{1}{4}, \frac{3}{4})$; Fig. 7(e), $\omega = (0, 1)$.

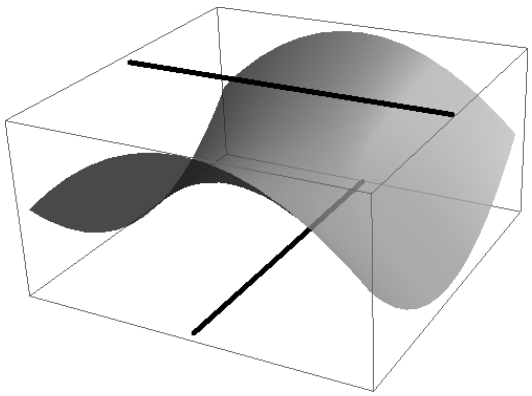


Figure 8: Generalized Voronoi diagram for $\omega = (0, 1)$ and two lines.

Applying this definition we obtain a new type of Voronoi diagram. As we can see in Figure 6, if $\omega = (0, 1)$ we divide the plane into (not necessarily convex) polygons (similar situation to the classical Voronoi diagram). Figure 7 presents a generalized diagram on the plane for different weights changing from $\omega = (1, 0)$ to $\omega = (0, 1)$. In general we obtain that the boundary sets usually are not polygons but zeros of quadratic polynomials. The same happens in $\mathbb{R}^3$ even for $\omega = (0, 1)$ see the Figure 8, where we show points with equal distance from two lines.

3. Generalization of the *k*-means method

Clustering is a classical problem of the division of the set $S \subset \mathbb{R}^N$ into separate clusters, or in other words, into sets showing given type of behaviour.

3.1. k-means

One of the most popular and basic method of clustering is the k-means algorithm. By this approach we want to divide S into k clusters $S_1, \ldots, S_k$ with minimal energy. For convenience of the reader and to establish the notation we shortly present the k-means algorithm.

For a cluster S and $r \in \mathbb{R}^N$ we define

$$E(S, r) := \sum_{s \in S} \|s - r\|^2.$$

The function $\mathrm{E}(S, r)$ is often interpreted as an energy. We say that the point $\overline{r}$ best "describe" the set S if the energy is minimal, more precisely, if

$$\mathrm{E}(S, \overline{r}) = \inf_{r \in \mathbb{R}^N} \{\mathrm{E}(S, r)\}.$$

It is easy to show that barycentre (mean) of S minimizes the function $\mathrm{E}(S, \cdot)$ (for more information see [16, 17]). The above consideration can be precisely formulated as follows:

Theorem 3.1 (k-means)**.** *Let S be a finite subset of $\mathbb{R}^N$. We have*

$$\mathrm{E}(S, \mu(S)) = \inf_{r \in \mathbb{R}^N} \{\mathrm{E}(S, r)\}$$

where $\mu(S) := \frac{1}{\mathrm{card} S} \sum_{s \in S} s$ denotes the barycentre of S.

Thus in the k-means the goal is to find such clustering $S = S_1 \cup \ldots \cup S_k$ that the function

$$\mathrm{E}(S_1, \ldots, S_k) = \sum_{j=1}^{k} \mathrm{E}(S_j, \mu(S_j))$$

is minimal. k-means algorithm for the set S after S. Lloyd [18, 19, 20] proceeds as follows:

stop condition
  choose $\varepsilon > 0$
initial conditions
  choose randomly points $\{\overline{s}_1, \ldots, \overline{s}_k\} \subset S$
  obtain first clustering $(S_1, \ldots, S_k)$ by matching each of the point $s \in S$ to the cluster S_j specified by $\overline{s}_j$ such that $\|s - \overline{s}_j\|^2$ is minimal
repeat
    let $\mathrm{E} = \mathrm{E}(S_1, \ldots, S_k)$
    compute new points $\overline{s}_1, \ldots, \overline{s}_k$ which best "describe" the clusters ($\overline{s}_j = \mu(S_j)$ for $j = 1, \ldots, k$)
    obtain new clustering $(S_1, \ldots, S_k)$ by adding each of the points $s \in S$ to the cluster such that $\|s - \overline{s}_j\|^2$ is minimal
until $\mathrm{E} - \mathrm{E}(S_1, \ldots, S_k) < \varepsilon$

Lloyd's method guarantees a decrease in each iteration but does not guarantee that the result will be optimal.

3.2. (ω, k)-means

In this chapter we consider generalization of k-means similar to that from the previous section concerning the Voronoi diagram. Instead of looking for the points which best "describe" clusters we seek n dimensional subspaces of $\mathbb{R}^N$.

Let $S \subset \mathbb{R}^N$ and $\omega \in [0,1]^{n+1}$, $\sum \omega_j = 1$ be fixed. For $\mathrm{v} \in \mathrm{E}_n(\mathbb{R}^N)$ let

$$\mathrm{E}_\omega(S, \mathrm{v}) := \sum_{s \in S} \mathrm{DIST}^2_\omega(s, \mathrm{v}).$$

We interpret the function $\mathrm{E}_\omega(S, \mathrm{v})$ as an energy of the set S respectively to the subspace generated by v. If the energy is zero, the set S is subset of affine space generated by v. We say that $\overline{\mathrm{v}}$ best "describes" the set S if the energy is minimal, more precisely if

$$\mathrm{E}_\omega(S, \overline{\mathrm{v}}) = \inf_{\mathrm{v} \in \mathrm{E}_n(\mathbb{R}^N)} \{\mathrm{E}_\omega(S, \mathrm{v})\}.$$

To obtain an optimal base we use a classical Karhunen-Loeve transform (called also Principal Component Analysis, shortly PCA), see [12]. The basic idea behind the PCA is to find the coordinate system in which the first few coordinates give us a "largest" possible information about our data.

Theorem 3.2 (PCA). *Let $S = \{s_1, \ldots, s_m\}$ be a finite subset of $\mathbb{R}^N$. Let*

$$\mathcal{M}(S) := (v_0, \ldots, v_N) \in \mathrm{E}_N(\mathbb{R}^N)$$

be such that

- $v_0 = \mu(S)$;
- $v_1, \ldots, v_N$ *are pairwise orthogonal eigenvectors of* $[s_1 - v_0, \ldots, s_m - v_0] \cdot [s_1 - v_0, \ldots, s_m - v_0]^T$ *arranged in descending order (according to the eigenvalues)*[3].

For every $n < N$ and $\omega \in [0,1]^{n+1}$ we have

$$\mathrm{E}_\omega(S, \mathcal{M}_k(S)) = \inf_{\mathrm{v} \in \mathrm{E}_n(\mathbb{R}^N)} \{\mathrm{E}_\omega(S, \mathrm{v})\},$$

where $\mathcal{M}_k(S) := (v_0, \ldots, v_k)$.

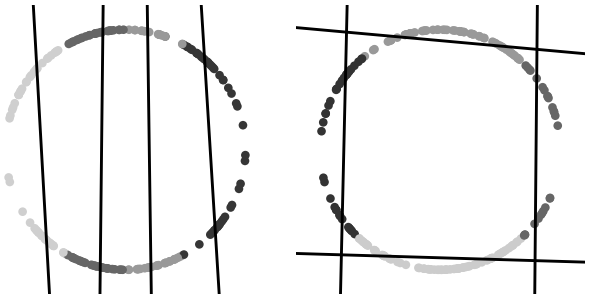
Thus given $\omega \in [0,1]^{n+1}$, $\sum \omega_j = 1$, in (w, k)-means our goal is to find such clustering $S = S_1 \cup \ldots \cup S_k$ that the function

$$\mathrm{E}_\omega(S_1, \ldots, S_k) := \sum_{j=1}^{k} \mathrm{E}_\omega(S_j, \mathcal{M}_n(S)) \tag{4}$$

is minimal. Consequently (ω, k)-means algorithm can be described as follows:

stop condition
choose $\varepsilon > 0$
initial conditions
choose randomly points $\{\overline{s}_1, \ldots, \overline{s}_k\} \subset S$
obtain first clustering $(S_1, \ldots, S_k)$ by matching each of the points $s \in S$ to the cluster such that $\|s - \overline{s}_j\|^2$ is minimal
repeat
let $\mathrm{E} = \mathrm{E}_\omega(S_1, \ldots, S_k)$
compute vectors $\mathrm{v}^1, \ldots, \mathrm{v}^k$, which best "describe" the clusters, by the PCA method $(\mathrm{v}_j = \mathcal{M}_n(S_j))$
obtain new clustering $(S_1, \ldots, S_k)$ by adding each of the point $s \in S$ to the cluster such that $\mathrm{DIST}_\omega(s, \mathrm{v}_j)$ is minimal
until $\mathrm{E} - \mathrm{E}_\omega(S_1, \ldots, S_k) < \varepsilon$

[3] $[s_1 - v_0, \ldots, s_m - v_0]$ is a matrix with columns $s_j - v_0$, for $j = 1, \ldots, m$.



(a) local minimum (b) global minimum

Figure 9: Circle clustering in $\mathbb{R}^2$ for 4 clusters with $\omega = (0, 1)$. (ω, k)-means method strongly dependents on initial conditions.

As is the case in the classical k-means, our algorithm guarantees a decrease in each iteration but does not guarantee that the result will be optimal (cf. Example 3.3).

Example 3.3. *As already mentioned in Section 2, the k-means do not find a global minimum and strongly depends on initial selection of clusters. In our case, this effect can be even more visible. Consider the case of circle C in $\mathbb{R}^2$ with 4 clusters and $\omega = (0, 1)$. The picture, see Figure 9(a), shows clustering obtained by use (ω, k)-means algorithm. Of course it is a local minimum of E^C_ω, however as we see at Figure 9(b) it is far from being the global minimum.*

Initial cluster selection in our algorithm is the same as in k-mean algorithm, but it is possible to consider others ways:

- k-means++ algorithm [21];
- starting from a given division (not from random distribution);
- repeating the initial choice of clusters many times.

Each of above approaches usually solves the problem described in Example 3.3.

Remark 3.4. *Let $S \subset \mathbb{R}^N$ and $\mathrm{v} \in \mathrm{E}_n(\mathbb{R}^N)$. It is easy to notice that the above method has following properties:*

- *for $\omega = (1, 0, \ldots, 0)$ we obtain the classical k-means,*
- *for $n = 1$ we get Karhunen-Loéve transform.*

As an algorithm's outcome we get:

- division of the data into clusters $\{S_1, \ldots, S_k\}$;

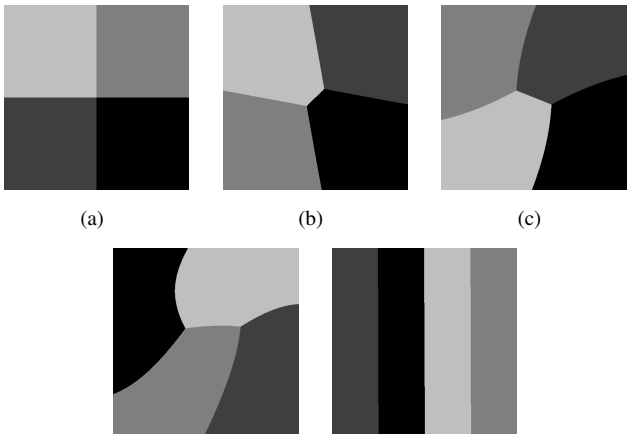


Figure 10: (ω, k)-means method for clustering into 4 clusters of set $\{\frac{0}{1000}, \frac{1}{1000}, \ldots, \frac{1000}{1000}\} \times \{\frac{0}{1000}, \frac{1}{1000}, \ldots, \frac{1000}{1000}\}$ for different weight vectors: Fig. 7(e), $\omega = (1, 0)$; Fig. 7(d), $\omega = (\frac{3}{4}, \frac{1}{4})$; Fig. 7(c), $\omega = (\frac{1}{2}, \frac{1}{2})$; Fig. 7(b), $\omega = (\frac{1}{4}, \frac{3}{4})$; Fig. 7(a), $\omega = (0, 1)$.

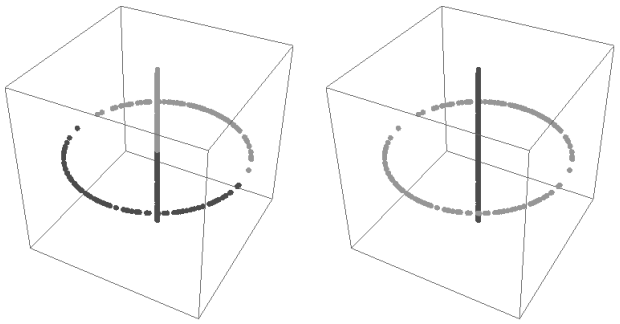


Figure 11: Clustering with: Fig. 11(a) – k-means; Fig. 11(b) – (ω, k)-means.

- for each cluster an affine space of dimension n obtained by the Karhunen-Loéve method which best represents the given cluster.

Example 3.5. *If we apply our algorithm for regular plane subset (ex. square) we obtain generalized Voronoi diagram (cf. Fig. 7) – Figure 10 present clustering for different weight vector changing from* $\omega = (1, 0)$ *to* $\omega = (0, 1)$.

4. Applications

4.1. Clustering

Clustering, by (ω, k)-means algorithm, gives a better description of the internal geometry of a set, in particular it found a reasonable splitting into connected components of consider the points grouped along two parallel sections (see Figure 2). Similar effect we can see in next example, when we consider the points grouped along circle and interval, see Figure 11.

Concluding, in many cases the (ω, k)-means method can be very useful in seeking n-dimensional (connected) components of given data sets.

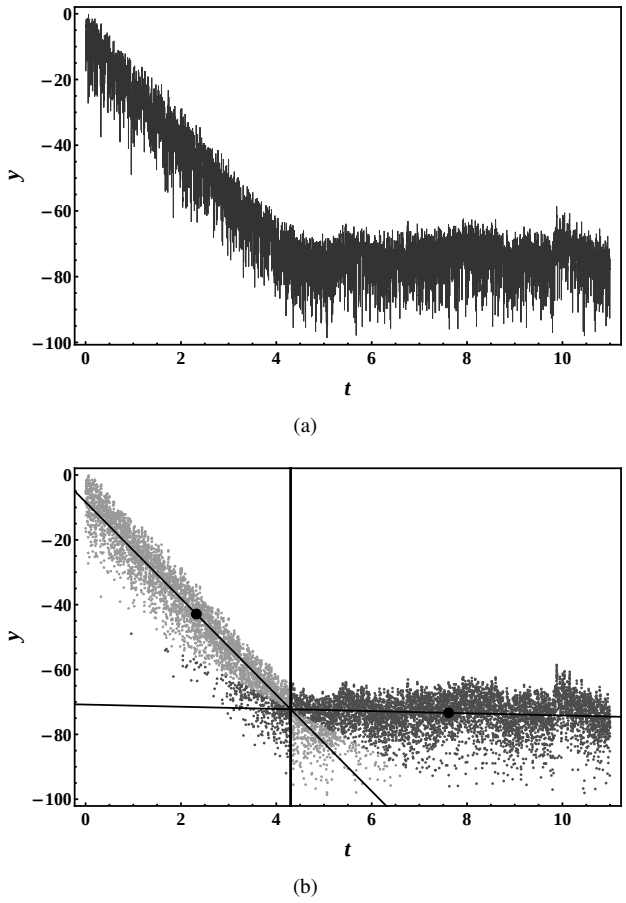


Figure 12: Linear component of the data structure. Fig. 12(a) – decay curve (original data). Fig. 12(b) – outcome from (ω, k)-means algorithm for $k = 2$, $\omega = (0, 1)$ we extract two linear components in data (black dots match clusters centers with the corresponding lines describing those clusters, vertical line separate sound and background noise – after 4.3 s).

4.2. Analysis of Functions

In this subsection we consider real data from acoustics. Acoustical engineers [22] study reverberation which is observed when a sound is produced in an enclosed space causing a large number of echoes to build up and then slowly decay as the sound is absorbed by the walls and the air. Reverberation time is crucial for describing the acoustic quality of a room or space. It is the most important parameter for describing sound levels, speech intelligibility and the perception of music and is used to correct or normalize building acoustics and sound power measurements.

We analyse the decay curve (see Figure 12(a)) which presents measurement of sound level in time and describe way in which sound impulse vanishing into background noise. Based on this we want to recover reverberation time. In particular we know that we have two linear component: first connected with sound absorption by the space and second – background noise. To use statistical analysis, we have to extract both of them, so we fix $k = 2$ (the number of cluster). Moreover, we are looking for two dimensional cluster so we choose $\omega = (0, 1)$ (see Remark 2.5 and Figure 12(b)).

Results obtained using our algorithm are comparable with

(a) k-means: $k = 5, n = 0$

(b) PCA: $k = 1, n = 1$

(c) (ω, k)-means: $k = 5, n = 1$

(d) (ω, k)-means: $k = 5, n = 5$

Figure 13: Compressed version of Lena picture. Subimage compare: Fig. 13(a) – classical k-means; Fig. 13(b) – Karhunen-Loéve Transform; Fig. 13(c) and Fig. 13(d) – (ω, k)-means algorithm.

those obtained by classical methods and give more opportunities for further research.

4.3. Image compression

Our algorithm can be used to compress image. First, we interpret photo as a matrix. We do this by dividing it into 8 by 8 pixels, where each pixel is described (in RGB) by using 3 parameters. Each of the pieces is presented as a vector from $\mathbb{R}^{192}$. By this operation we obtain dataset from $\mathbb{R}^{192}$.

Taking into consideration the classical Lena picture (508×508 pixels), let us present its compressed version with the use of k-means method (Figure 13(a), $k = 5$, $n = 0$), Karhunen-Loéve Transform (Figure 13(b), $k = 1, n = 1$) and (ω, k)-means algorithm (Figures 13(c) and 13(d)). As we can see the algorithm allows to reconstruct with great accuracy compressed images while reducing the amount of needed information to save (in our example we remember ex. only 5 coordinates in 192-dimensional space).

Table 4.3 presents error in image reconstruction for Lena picture. We run (ω, k)-means algorithm 16 times and each run improve clustering quality 50 times.

5. Memory compression

In this section we present a method of compression based on our algorithm. Let k (the number of clusters) and ω (weight parameter) be fixed. As a result of the (ω, k)-means algorithm for the dataset S we obtain k clusters $\{S_1, \ldots, S_k\}$ and k coordinate systems $\{\mathrm{v}^1, \ldots, \mathrm{v}^k\} \subset \mathrm{E}_n(\mathbb{R}^N)$.

	n					
k	0	1	2	3	4	5
1	40328	19499	16358	12452	10160	8149
2	27502	17193	13031	10382	9082	7913
3	23261	15437	11631	9612	8350	7358
4	20990	14454	11004	9192	7922	7095
5	20150	13740	10602	8867	7745	6814

Table 1: Error in image decompression for certain k and n.

For $\mathrm{v} \in \mathrm{E}_n(\mathbb{R}^N)$ and $n_0 \leq n$ we define sub–base of dimension n_0 by

$$\mathrm{v}_{n_0} = (v_0, \ldots, v_{n_0}).$$

We choose $n_1, \ldots, n_k \in \{1, \ldots, N\}$ and we compress the data of S by replacing each element $s \in S_i$ by its orthogonal projection on a suitable subspace spanned on v_{n_i}.

Let $S_1, \ldots, S_k$ and $\{\mathrm{v}^1, \ldots, \mathrm{v}^k\} \subset \mathrm{E}_n(\mathbb{R}^N)$ be a result of the (ω, k)-mean algorithm. For parameters $n_1, \ldots, n_k$ we consider the compression error

$$\mathrm{Comp\_err}(n_1, \ldots, n_k) := \left(\sum_{i=1}^{k} \sum_{s \in S_i} \mathrm{dist}^2(s; \mathrm{v}^i_{n_i}) \right)^{1/2}.$$

Let $\varepsilon > 0$ be given. The procedure of determining $n_1, \ldots, n_k$ of clusters, such that

$$\mathrm{Comp\_err}(n_1, \ldots, n_k) < \varepsilon$$

can be formulated as follows:

1. Apply the (ω, k)–means algorithm with arbitrary given k (in general this parameter should be chosen respectively to data structure) and ω (which describe possible dimensions of clusters.
2. In each cluster $S_1, \ldots, S_k$ determinate eigenvalue of covariance matrix
$$\lambda_1^j, \ldots, \lambda_N^j \text{ for } j = 1, \ldots, k.$$
3. Put
$$\Lambda := \left\{ \lambda_1^{l_1}, \ldots, \lambda_{kN}^{l_{kN}} \right\}.$$
4. Sort the eigenvalues increasingly
$$\Lambda_{(\cdot)} = \left\{ \lambda_{(1)}^{l_{(1)}}, \ldots, \lambda_{(kn)}^{l_{(kn)}} \right\}.$$
5. Let
$$\bar{n} := \sup \left\{ n \colon \sum_{i=1}^{n} \lambda_{(i)}^{l_{(i)}} \cdot m_{l_{(i)}} \leq \varepsilon \right\},$$
where $m_i = \mathrm{card}(S_i)$, for $i = \{1, \ldots, k\}$.
6. We define $n_1, \ldots, n_k$ by
$$n_j = \mathrm{card} \left\{ \lambda_{(i)}^{l(i)} \colon \text{ such that } l_{(i)} = j \text{ and } (i) > \bar{n} \right\}.$$

Before we show that this algorithm gives good accuracy we present following theorem

	S_1	S_2
$\mu(S_i)$	(-0.048, -0.027, 0.0)	(-0.004, 0.002, -0.012)
eigenvector	$\begin{bmatrix} 0.0 & 0.534 & -0.845 \\ 0.0 & -0.845 & -0.534 \\ 1.0 & 0.0 & 0.0 \end{bmatrix}$	$\begin{bmatrix} 0.692 & -0.722 & 0.001 \\ 0.722 & 0.692 & 0.002 \\ 0.003 & 0.0 & -1.0 \end{bmatrix}$
eigenvalue	(9.241, 0.040, 0.001)	(2.124,1.885,0.001)

Table 2: Outcome

Lemma 5.1 ([12]). *Let $S = \{x_1, \ldots, x_n\}$ be subset of $\mathbb{R}^N$. By $\{\lambda_1, \ldots, \lambda_n\}$ we denote eigenvalues corresponding to eigenvectors $\{v_1, \ldots, v_n\}$ of matrix* $\mathrm{cov}([x_1, \ldots, x_n])$.

Then

$$\sum_{x \in S} \mathrm{dist}^2(x, \mathrm{v}_k) = \sum_{i=k+1}^{n} \lambda_i \cdot n,$$

where $\mathrm{v}_k = \{v_0, v_1, \ldots, v_k\}$.

Now by simple calculations we have

$$\mathrm{Comp\_err}(n_1, \ldots, n_k) = \left(\sum_{i=1}^{k} \sum_{s \in S_i} \mathrm{dist}(s; \mathrm{v}_{n_i})^2 \right)^{1/2}$$

$$= \left(\sum_{i}^{k} \sum_{j=n_i+1}^{m_i} \lambda_j^i \cdot m_i \right)^{1/2} = \left(\sum_{i=1}^{\bar{n}} \lambda_{(i)}^{l_{(i)}} \cdot m_{l_{(i)}} \right)^{1/2} < \varepsilon.$$

Lets us consider extension of example presented in Subsection 4.1 (which was also mentioned in the Introduction to this paper).

Example 5.2. *Consider the dataset from Subsection 4.1 modified by adding same noise (ex. white Gaussian noise) – see Figure 3. In first step we fix epsilon $\varepsilon = 2.87$ (which gives 5.5% of total error[4]). Then to start our algorithm, we have to choose the parameters k and ω. In our example we want to obtain two cluster, so we fix $k = 2$. Moreover the first cluster should represent the one–dimension data and the second two–dimension. So we put not zero elements at ω_1, ω_2[5] for example $\omega = (0, \frac{1}{2}, \frac{1}{2})$. Outcome obtained at the end of calculation is presented in Table 5.2. Cluster S_1 corresponds to point grouped along interval, and the S_2 – along circle.*

Now by steps 3–6 we have

$$\Lambda_{(\cdot)} = \{0.001, 0.001, 0.040, 1.885, 2.124, 9.241\},$$

$$\bar{n} = 3,$$

$$n_1 = 1, \quad n_2 = 2.$$

Consequently, we get $2 \cdot 199$ parameters for S_2 and $1 \cdot 201$ for S_1.

At the end of this section we back to example of the Lena picture.

Example 5.3. *Consider the Lena image from Subsection 4.3. Let $\varepsilon = 427$ (which gives 1% of total error) be fixed. We use $k = 5$. Then we have to choose ω. If we do not have any intuition about possible dimension of cluster we can put*

$$\omega = \left(\frac{1}{192}, \ldots, \frac{1}{192} \right).$$

Since in picture compression we expect the data to have lower dimensional structure[6], we narrow our consideration to subspaces of dimension between 10–20 by choosing, according to Remark 2.5,

$$\omega = \left(\underbrace{0, \ldots, 0}_{1-10}, \underbrace{\frac{1}{10}, \ldots, \frac{1}{10}}_{11-20}, \underbrace{0, \ldots, 0}_{21-192} \right).$$

By applying points 3–6 we obtain:

- $1 \cdot 2375$ *parameters for the first cluster,*
- $2 \cdot 151$ *parameters for the second cluster,*
- $5 \cdot 880$ *parameters for the third cluster,*
- $4 \cdot 229$ *parameters for the fourth cluster,*
- $3 \cdot 461$ *parameters for the fifth cluster.*

As we see by use our method we have to remember 9376 *parameters. If we fix $n_1 = \ldots = n_5$ such that* $\mathrm{Comp\_err}(n_1, \ldots, n_5) < \varepsilon$ *(*4 *first eigenvalues for each cluster) we obtain* 16384 *parameters – which is all most twice as much as in our method.*

6. Implementation

Sample implementation of (ω, k)-means algorithm prepared in Java programming language is available at [23].

References

[1] G. Zervas, S. Ruger, *The curse of dimensionality and document clustering*, in: IEEE Seminar Digests.

[2] A. Hinneburg, D. Keim, *Optimal grid-clustering: Towards breaking the curse of dimensionality in high-dimensional clustering*, Citeseer, 1999.

[3] F. Murtagh, J. Starck, M. Berry, *Overcoming the curse of dimensionality in clustering by means of the wavelet transform*, The Computer Journal (2000) 107–120.

[4] H. Kriegel, P. Kröger, A. Zimek, *Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering*, ACM Transactions on Knowledge Discovery from Data (TKDD) (2009) 1–58.

[4] By total error we understand error obtained in the worst case of compression, when we replace each element in each cluster by barycenter of all data.

[5] Indexes of parameters ω are shifted respectively to elements of $\mathrm{v} \in \mathrm{E}_n(\mathbb{R}^N)$ so ω_1 corresponds with zero-dimension subspace (barycenter of cluster v_0).

[6] That why the compression based on Karhunen–Loéve transform or JPG format gives good results.

[5] L. Parsons, E. Haque, H. Liu, *Subspace clustering for high dimensional data: a review*, ACM SIGKDD Explorations Newsletter (2004) 90–105.
[6] A. Zimek, *Correlation clustering*, SIGKDD Explor. Newsl. (2009) 53–54.
[7] M. Filippone, F. Camastra, F. Masulli, S. Rovetta, *A survey of kernel and spectral methods for clustering*, Pattern recognition (2008) 176–190.
[8] I. Dhillon, Y. Guan, B. Kulis, *Kernel k-means: spectral clustering and normalized cuts*, in: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, pp. 551–556.
[9] U. Von Luxburg, *A tutorial on spectral clustering*, Statistics and Computing 17 (2007) 395–416.
[10] B. Zhao, J. Kwok, C. Zhang, *Multiple kernel clustering*, in: Proceedings of the 9th SIAM International Conference on Data Mining, pp. 638–649.
[11] A. Jain, R. Dubes, *Algorithms for clustering data*, Prentice Hall, 1988.
[12] I. Jolliffe, *Principal component analysis*, Encyclopedia of Statistics in Behavioral Science (2002).
[13] D. Salomon, *Data compression: the complete reference*, Springer-Verlag New York Inc, 2007.
[14] A. Okabe, *Spatial tessellations: concepts and applications of Voronoi diagrams*, John Wiley & Sons Inc, 2000.
[15] R. Klein, *Concrete and Abstract Voronoi Diagrams*, volume 400, 1989.
[16] C. Ding, X. He, *K-means clustering via principal component analysis*, in: Proceedings of the twenty-first international conference on Machine learning, p. 29.
[17] W. Fisher, *On grouping for maximum homogeneity*, Journal of the American Statistical Association (1958) 789–798.
[18] E. Forgy, *Cluster analysis of multivariate data: Efficiency vs. interpretability of classification*, Biometrics 21 (1965) 768–769.
[19] S. Lloyd, *Least squares quantization in PC*, IEEE Transactions on Information Theory (2002) 129–137.
[20] J. MacQueen, et al., *Some methods for classification and analysis of multivariate observation*, in: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, p. 14.
[21] D. Arthur, S. Vassilvitskii, *k-means++: The advantages of careful seeding*, in: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, pp. 1027–1035.
[22] T. Kamisiński, J. Rubacha, A. Pilch, *The Study of Sound Scattering Structures for the Purposes of Room Acoustic Enhancement*, Acta Physica Polonica A, Polska Akademia Nauk. Instytut Fizyki, Warszawa (2010) 83–86.
[23] K. Misztal, P. Spurek, J. Tabor, *Implementation of the* (ω, k)*-means algorithm*, `http://www2.im.uj.edu.pl/badania/preprinty/imuj2012/pr1201.zip`, 2012.