Weighted Approach to Projective Clustering

Przemyslaw Spurek, Jacek Tabor, Krzysztof Misztal

Weighted approach to projective clustering

Przemysław Spurek[1], Jacek Tabor[1], and Krzysztof Misztal[2]

[1] Jagiellonian University
Faculty of Mathematics and Computer Science
Łojasiewicza 6, 30-348 Kraków, Poland
przemyslaw.spurek@ii.uj.edu.pl
jacek.tabor@ii.uj.edu.pl
[2] AGH University of Science and Technology
Faculty of Physics and Applied Computer Science
al. A. Mickiewicza 30, 30-059 Kraków, Poland
Krzysztof.Misztal@fis.agh.edu.pl

Abstract. k-means is the basic method applied in many data clustering problems. As is known, its natural modification can be applied to projection clustering by changing the cost function from the squared-distance from the point to the squared distance from the affine subspace. However, to apply thus approach we need the beforehand knowledge of the dimension.
In this paper we show how to modify this approach to allow greater flexibility by using the weights over respective range of subspaces.

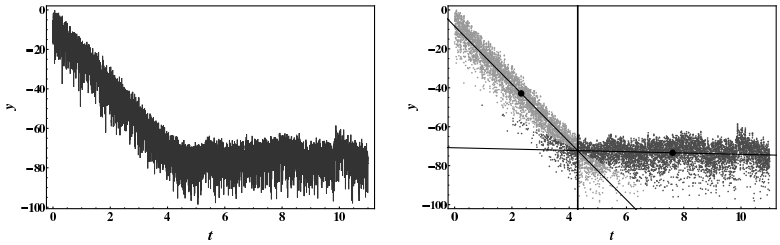
Keywords: Projective clustering, Karhunen-Loéve Transform, PCA, k-means

1 Introduction

Projective clustering is a part of the large subspace clustering family, which general aim lies in dividing the given high-dimensional data-set into clusters. For the survey, motivation and further references on subspace clustering we refer to [1, 8, 11, 12].

A typical application of the projective clustering concerns splitting of a given data-set S into clusters with respect to k affine subspaces. A typical illustrative motivation to create such an algorithm is given by a problem to determine linear components of the data obtained in acoustical experiments [6] (see Fig. 1(a)). In general, this kind of data contains two linear components: the first connected with sound (product in experiments) which linearly disappears being absorbed by the walls and the air, and the second consists of noise connected with empty space. To use statistical analysis, we have to extract both of them.

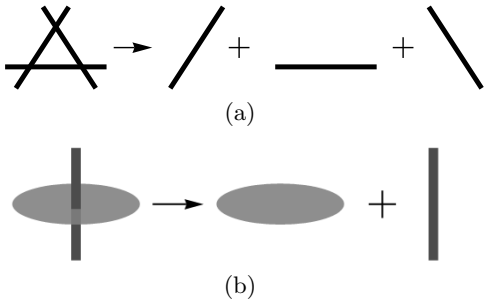
The simplest solution, see [4], lies in a natural modification of k-means: instead of finding k centers which best represent the data, we find k subspaces of given dimension. In other words we change the cost function from the squared-distance from the center of the cluster to the squared distance from the affine



(a) (b)

Fig. 1. Linear component in acoustical data structure. Fig. 1(a) – original data. Fig. 1(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).

space which best represents it and can be found by PCA [5]. To explain it graphically let us consider the following example. Fig. 2(a) represents three lines in the plane. The method given in [4] will split the data into three lines. However,



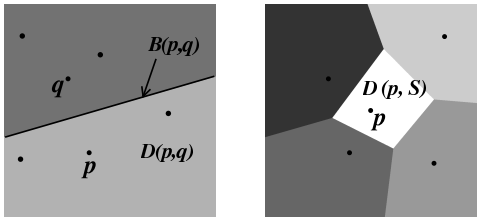
(a)

(b)

Fig. 2. Our goal is to create algorithm which will split Fig. 2(b) as two groups of one- and two-dimensional points.

it will not work sufficiently well on data given in 2(b), which we would like to split into a line and a circle.

In this paper we generalize the approach from [4] to allow the weights over subspaces of different dimensions, which gives a partial solution to the above mentioned problem. The resulted algorithm we call (ω, k)-means, where ω represents the weights, and k the number of clusters we are interested in. 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 $\omega = (1, 0, \ldots, 0)$ we obtain the k-means while for $k = 1$ we obtain the PCA).



(a) (b)

Fig. 3. Graphical presentation of $B(p,q)$, $D(p,q)$ and $D(p,S)$ in $\mathbb{R}^2$.

Detailed investigation of the results of (ω,k)-means allows to determine the optimal dimensions of subspaces, see algorithm in Section 4. The results of this approach we apply to image compression (Section 4, Example 3). In such a case we obtain almost twice the compression of the classical approach.

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 [10]. 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 [7]). For $N \in \mathbb{N}$ consider $\mathbb{R}^N$ with the standard Euclidean distance and let S be a finite subset of $\mathbb{R}^N$. For $p,q \in S$ such, that $p \neq q$, let

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

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

The hyperplane $B(p,q)$ divides $\mathbb{R}^N$ into two subsets, 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 3(a).

The set

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

of all points that are closer to p than to any other element of S is called the (open) *Voronoi region* [7] of p with respect to S. Set $D(p,S)$ for $N=2$ is the interior of a convex, possibly unbounded polygon (Figure 3(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. The union

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

of all region boundaries is called the Voronoi diagram [7] of S. The common boundary of two Voronoi regions is a *Voronoi edge*. Two edges meet at a Voronoi vertex.

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 $\mathbb{R}^N$. For $n \leq N$ let

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

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

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

We modify equations (1) and (2), by using distance between a point and affine subspace.

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

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

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

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

Remark 1. It is easy to notice, that DIST has the following properties:

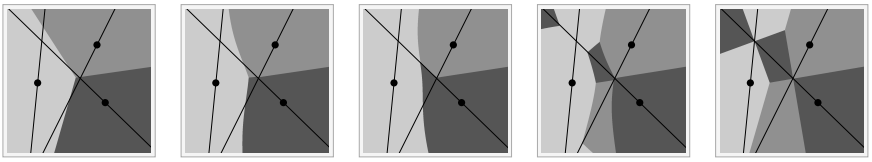
- for $\mathrm{v} \in \mathrm{E}_n(\mathbb{R}^N)$ and $\omega = (0, \ldots, 0, 1) \in [0,1]^{n+1}$ we obtain that $\mathrm{DIST}_\omega(x; \mathrm{v})$ is a distance between the point x and affine space $\mathrm{aff}(\mathrm{v})$;
- if $\mathrm{v}_0 = 0$ and $\omega = (0, \ldots, 0, 1)$ then DIST_ω is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

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

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

Corollary 1. *Formula (3) can be reformulated as follows*

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



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

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

To optimize calculations we define

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

and since $\sum \omega_j = 1$ *we can simplify our formula to*

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

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

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

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

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

Definition 2. *Let* $n \in \mathbb{N}$, $n < N$ *be fixed. Let* S *be a 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*

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

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.

Applying this definition we obtain a new type of Voronoi diagram. Figure 4 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 Fig. 5, where we show points with equal distance from two lines.

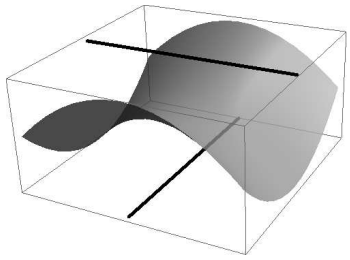


Fig. 5. Generalized Voronoi diagram for $\omega = (0, 1)$ and two lines.

3 Generalization of the k-means method

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

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 method.

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

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

which we interprets as an energy. We say that the point $\overline{r}$ best "describes" the set S if 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 barycenter (mean) of S minimizes the function $\mathrm{E}(S, \cdot)$ (for more information see [2, 3]). The above consideration can be precisely formulated as follows:

Theorem 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.

In this paper we consider generalization of k-means algorithm similar to that from the previous section concerning the Voronoi diagram. Instead of looking for 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 [5]. 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 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 with respect to the eigenvalues.*

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

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

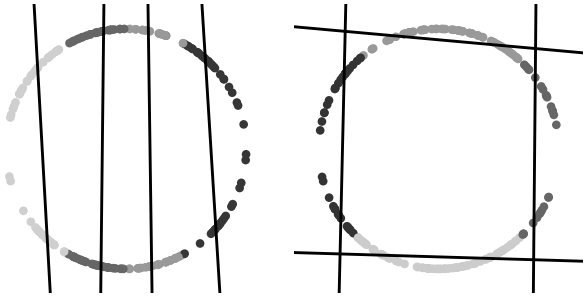
where $\mathcal{M}_k(S) := (v_0, \ldots, v_k)$.

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" 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 $\mathrm{E} - \mathrm{E}_\omega(S_1, \ldots, S_k) < \varepsilon$



(a) local minimum (b) global minimum

Fig. 6. 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. It is easy to notice that the above method has following properties:

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

Example 1. 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 6(a), shows clustering obtained by use (ω, k)-means algorithm. Of course it is a local minimum of E_ω, however as we see at Figure 6(b) it is far from being the global minimum.

4 Determining the dimensions of subspaces

In this section we present the method of determining the optimal dimensions subspaces which describe clusters. By using various value of parameter ω we are able to compress data by remembering different coordinates in each cluster.

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 $\{\mathrm{v}^1, \ldots, \mathrm{v}^k\} \subset \mathrm{E}_n(\mathbb{R}^N)$. 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 a maximal allowed error. We want to find the minimal number of parameters to "compress" the data with compression error below ε. Observe that if we approximate S_i by its projection onto subspaces of dimension n_i, the total number of parameters is given by

$$n_1 \cdot \mathrm{card}(S_1) + \ldots + n_k \cdot \mathrm{card}(S_k).$$

The procedure of determining respective dimensions $n_1, \ldots, n_k$ 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 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
$$\lambda_1^j \geq \ldots \geq \lambda_N^j \text{ for } j = 1, \ldots, k.$$
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

Lemma 1 ([5]). *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 $\mathrm{v}_k = \{v_0, v_1, \ldots, v_k\}$.

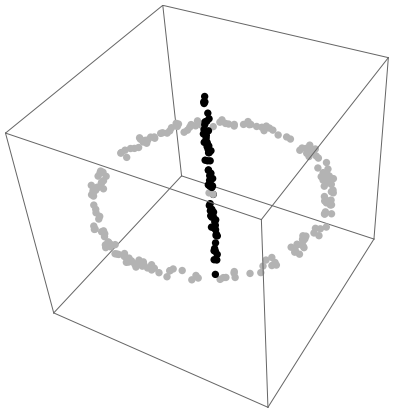


Fig. 7. Clustering with (ω, k)-means for $\omega = (0, \frac{1}{2}, \frac{1}{2})$.

Now by simple calculations we have

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

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

Example 2. Consider the dataset containing points group around a segment (200 points) and circle (200 points) – see Figure 7. In first step we fix $\varepsilon = 2.87$ (which gives 5.5% of total error[3]). Then to start our algorithm we have to choose the parameters k and ω. In our example we want to obtain two clusters, so we fix $k = 2$. Moreover the first cluster should represent the one–dimension data and the second two–dimension. So we put $\omega = (0, \frac{1}{2}, \frac{1}{2})$. Outcome obtained at the end of calculation is presented in Table 2. Cluster S_1 corresponds to points 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 presen our algorithm on the example of the classical Lena picture.

[3] 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.

	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 1. Outcome of the (ω, k)-means in the case of data from Example 2.

Example 3. Let us consider the classical Lena image. First, we interpret photo as a matrix. We do this by dividing a picture into disjoint squares (8 by 8) pixels, where each of them is described (in RGB) by using 3 parameters. Consequently we have vectors from $\mathbb{R}^{192}$. 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[4], we narrow our consideration to subspaces of dimension between 10–20 by choosing, according to Remark 1,

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

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 $\text{Comp\_err}(n_1, \ldots, n_5) < \varepsilon$ (4 first eigenvalues for each cluster) we obtain 16384 parameters – which is all most twice as much as in our method.

Sample implementation of (ω, k)-means algorithm prepared in Java programming language is available at [9].

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

References

1. Agarwal, P.K. and Mustafa, N.H.: k-Means projective clustering. Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, 155–165, 2004.
2. Ding, C. and He, X.: K-means clustering via principal component analysis. Proceedings of the twenty-first international conference on Machine learning, 29, 2004.
3. Fisher, W.: On grouping for maximum homogeneity. Journal of the American Statistical Association, 789–798, 1958.
4. Grim, J.: Multimodal discrete Karhunen-Loéve expansion. Institute of Information Theory and Automation AS CR Kybernetika, 329–330, 1986.
5. Jolliffe, I.: Principal component analysis. Encyclopedia of Statistics in Behavioral Science, 2002.
6. Kamisiński, T., Rubacha, J. and Pilch, A.: The Study of Sound Scattering Structures for the Purposes of Room Acoustic Enhancement. Acta Physica Polonica A, Polska Akademia Nauk. Instytut Fizyki, Warszawa, 83–86, 2010.
7. Klein, R.: Concrete and Abstract Voronoi Diagrams, Springer-Verlag, vol. 400, 1989.
8. Kriegel, H., Kröger, P. and Zimek, A.: Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. ACM Transactions on Knowledge Discovery from Data (TKDD), 3, (1), 1–58, 2009.
9. Misztal, K., Spurek, P. and Tabor, J.: Implementation of the (ω, k)-means algorithm. http://www2.im.uj.edu.pl/badania/preprinty/imuj2012/pr1201.zip, 2012.
10. Okabe, A.: Spatial tessellations: concepts and applications of Voronoi diagrams. John Wiley & Sons Inc, 2000.
11. Parsons, L., Haque, E. and Liu, H.: Subspace clustering for high dimensional data: a review. ACM SIGKDD Explorations Newsletter, 6, 1, 90–105, 2004.
12. Vidal, R.: Subspace clustering. Signal Processing Magazine, IEEE, 28, 2, 52–68, 2011.