Interpretation and optimization of the k-means algorithm

Kristian Sabo
Department of Mathematics, University of Osijek
Trg Lj. Gaja 6, HR – 31 000 Osijek, Croatia
e-mail: ksabo@mathos.hr

Rudolf Scitovski[1]
Department of Mathematics, University of Osijek
Trg Lj. Gaja 6, HR – 31 000 Osijek, Croatia
e-mail: scitowsk@mathos.hr

Abstract. The paper gives a new interpretation and a possible optimization of the well-known k-means algorithm for searching for the locally optimal partition of the set $\mathcal{A} = \{a_i \in \mathbb{R}^n : i = 1, \ldots, m\} \subset \mathbb{R}^n$ which consists of k disjoint nonempty subsets $\pi_1, \ldots, \pi_k$, $1 \leq k \leq m$. For this purpose, a new *Divided k-means Algorithm* was constructed as a limit case of the well-known Smoothed k-means Algorithm. It is shown that the algorithm constructed in such way coincides with the k-means algorithm if during the iterative procedure no data points appear in the Voronoi diagram. If in the partition obtained by applying the *Divided k-means Algorithm* there are data points lying in the Voronoi diagram, it is shown that the obtained result can be improved further.

Key words: Clustering; Data mining; k-means; Voronoi diagram

MSC2010: 68T10, 62H30, 91C20, 90C26

1 Introduction

Clustering or grouping a data set into conceptually meaningful clusters is a well-studied problem in recent literature, and it has practical importance in a wide variety of applications [4, 6, 11–13, 19, 27].

Let $I = \{1, \ldots, m\}$ and $J = \{1, \ldots, k\}$, $1 \leq k \leq m$ be the set of natural numbers. A partition of the set $\mathcal{A} = \{a_i \in \mathbb{R}^n : i = 1, \ldots, m\} \subset \mathbb{R}^n$ into k disjoint subsets $\pi_1, \ldots, \pi_k$, $1 \leq k \leq m$, such that

$$\bigcup_{i=1}^{k} \pi_i = \mathcal{A}, \qquad \pi_i \cap \pi_j = \emptyset, \quad i \neq j, \qquad |\pi_j| \geq 1, \quad j = 1, \ldots, k, \tag{1}$$

will be denoted by $\Pi(\mathcal{A}) = \{\pi_1, \ldots, \pi_k\}$, and the elements $\pi_1, \ldots, \pi_k$ of such partition are called *clusters in* $\mathbb{R}^n$.

[1]Corresponding author: Rudolf Scitovski, e-mail: scitowsk@mathos.hr, telephone number: ++385-224-800, fax number: ++385-224-801

If $d\colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_+$, $\mathbb{R}_+ = [0, +\infty\rangle$ is some distance-like function (see e.g. [11, 13, 22, 25]), then to each cluster $\pi_j \in \Pi$ we can associate its center c_j defined by

$$c_j = c(\pi_j) := \operatorname*{argmin}_{x \in \operatorname{conv}(\pi_j)} \sum_{a_i \in \pi_j} d(x, a_i). \tag{2}$$

In the sequel, a special and well-known *least square distance-like function* given by $d(x, y) = \|x - y\|_2^2$, $x, y \in \mathbb{R}^n$ will be used as a distance-like function.

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

$$\mathcal{F}(\Pi) = \sum_{j=1}^{k} \sum_{i=1}^{m} d(c_j, a_i),$$

then we can say that $\Pi^\star$ is an optimal k-partition if

$$\mathcal{F}(\Pi^\star) = \min_{\Pi \in \mathcal{P}(\mathcal{A}, k)} \mathcal{F}(\Pi). \tag{3}$$

Conversely, for a given set of different points $z_1, \ldots, z_k \in \mathbb{R}^n$, applying the minimal distance condition, we can define the partition $\Pi = \{\pi_1, \ldots, \pi_k\}$ of the set $\mathcal{A}$ in the following way:

$$\pi_j = \{a \in \mathcal{A} : d(z_j, a) \leq d(z_s, a),\ \forall s \in J\}, \qquad j \in J, \tag{4}$$

where one has to take care that every element of the set $\mathcal{A}$ occurs in one and only one cluster. Therefore, the problem of finding an optimal partition of the set $\mathcal{A}$ can be reduced to the following optimization problem

$$\min_{z_1, \ldots, z_k \in \mathbb{R}^n} F(z_1, \ldots, z_k), \qquad F(z_1, \ldots, z_k) = \sum_{i=1}^{m} \min_{1 \leq j \leq k} d(z_j, a_i) = \sum_{i=1}^{m} \sum_{j=1}^{k} w_i^{(j)} d(z_j, a_i), \tag{5}$$

where $F\colon \mathbb{R}^{kn} \to \mathbb{R}_+$, and

$$w_i^{(j)} = \begin{cases} 1, & a_i \in \pi(z_j); \\ 0, & a_i \notin \pi(z_j) \end{cases}, \quad j \in J, \tag{6}$$

and for all $i \in I$ it holds

$$\sum_{j=1}^{k} w_i^{(j)} = 1. \tag{7}$$

Optimization problems (3) and optimization problem (5) are equivalent [1, 20]. A global optimization problem (5) can also be found in literature as a *center-based clustering problem* or *k-means/k-median problem* [7, 14, 18, 22]. Thereby the objective function F can have a great number of independent variables (the number of clusters in the partition multiplied by the dimension of data points $(k \cdot n)$), it does not have to be either convex or differentiable and generally it may have several local minima. Therefore, this becomes a complex global optimization problem [5, 9].

In this paper, we will show a new interpretation of the well-known k-means algorithm. It is also demonstrated how the obtained result can be improved further.

The paper is organized as follows. In the next section, two well-known algorithms for searching for the locally optimal partition, i.e. the k-means algorithm and Smoothed k-means Algorithm (`smoka`), are briefly described and a new Divided k-means Algorithm (DKM) is proposed. In Section 3, some properties of the DKM algorithm and connection with the k-means algorithm is shown. Finally, some conclusions are given in Section 4.

2 Algorithms for searching for the locally optimal partition

In this section, we will briefly show two well-known algorithms for searching for the locally optimal partition, i.e. the k-means algorithm and `smoka` (see e.g. [11, 12, 20]), and propose a new DKM algorithm.

2.1 k-means algorithm

There are various notation variants of this well-known algorithm (see e.g. [11, 14, 15]). For further usage in this paper, the algorithm will be written in the following way.

Algorithm 1. (k-means algorithm)

Step 0: Input $1 \le k \le m$; $I = \{1, \dots, m\}$; $J = \{1, \dots, k\}$; $A = \{a_i \in \mathbb{R}^n : i \in I\}$. Choose mutually different points $z_1 \dots, z_k \in \operatorname{conv}(\mathcal{A})$.

Step 1: (Assignment step) Define clusters

$$\pi(z_j) = \{a_i \in \mathcal{A} : d(z_j, a_i) \le d(z_s, a_i),\ \forall s \in J\}, \quad j \in J,$$

where one has to take care that every element of the set $\mathcal{A}$ occurs in one and only one cluster. Define weights $w_i^{(j)}$ according to (6) and (7);
Calculate $F_0 = \sum\limits_{j=1}^{k} \left(\sum\limits_{i=1}^{m} w_i^{(j)} d(z_j, a_i) \right)$.

Step 2: (Update step) Determine

$$c_j = \operatorname*{argmin}_{x \in \mathbb{R}^n} \sum_{i=1}^{m} w_i^{(j)} d(x, a_i) = \frac{1}{\sum\limits_{l=1}^{m} w_l^{(j)}} \sum_{i=1}^{m} w_i^{(j)} a_i, \quad j \in J; \tag{8}$$

$$\pi(c_j) = \{a_i \in \mathcal{A} : d(c_j, a) \le d(c_s, a),\ \forall s \in J\}, \quad j \in J;$$

Define new weights

$$w_i^{(j)} = \begin{cases} 1, & a_i \in \pi(c_j); \\ 0, & a_i \notin \pi(c_j) \end{cases}, \quad j \in J, \quad \text{such that } \sum_{j=1}^{k} w_i^{(j)} = 1;$$

Calculate $F_1 = \sum\limits_{j=1}^{k} \left(\sum\limits_{i=1}^{m} w_i^{(j)} d(c_j, a_i) \right)$.

Step 3: If $F_1 < F_0$, set $F_0 = F_1$ and go to *Step 2*.
Else set $c_j^\star = c_j$, $\forall j \in J$ and STOP.

Points $z_1 \ldots, z_k$ from *Step 1* and points $c_1, \ldots, c_k$ from *Step 2* are called *assignment points* and *centroids* of the clusters, respectively. Centroids in *Step 2* become assignment points on the basis of which we define new clusters.

Algorithm 1 is finite and in every step it reduces the value of the objective function. Centroids $(c_1^\star, \ldots, c_k^\star)$ obtained by applying Algorithm 1 are called *locally optimal centroids*, and the corresponding partition $\{\pi_1, \ldots, \pi_k\}$ is called a *locally optimal partition.*

In addition to that, it may happen that one of clusters becomes an empty set [11]. In relation to that, [21] gives a sufficient condition under which functional (5) attains its local minimum at the point $(c_1^\star, \ldots, c_k^\star)$. A partition determined by this point is called a *stable partition* [11, 22, 27]. Also, in accordance with [21], a stable partition does not contain empty clusters.

2.2 `smoka`

The `smoka` algorithm has appeared relatively recently in literature as a natural generalization of the well-known Weiszfeld algorithm for the Fermat–Weber location problem (see e.g. [2, 8]). In the sequel, we will briefly describe this algorithm and give its most important properties. Consider the optimization problem

$$\min_{z_1,\ldots,z_k \in \mathbb{R}^n} F(z_1, \ldots, z_k), \qquad F(z_1, \ldots, z_k) = \sum_{i=1}^{m} \min_{1 \le j \le k} d(z_j, a_i). \tag{9}$$

Since for every vector $r = (r_1, \ldots, r_n) \in \mathbb{R}^n$ holds (see e.g. [11])

$$\max_{1 \le j \le k} r_j = \lim_{\epsilon \to 0+} \epsilon \ln \sum_{j=1}^{n} \exp\left(\frac{r_j}{\epsilon}\right),$$

and $\min\limits_{1 \le j \le k} r_j = -\max\limits_{1 \le j \le k} (-r_j)$, functional (9) can be approximated by

$$F_\epsilon(z_1, \ldots, z_k) = -\sum_{i=1}^{m} \epsilon \ln \sum_{j=1}^{k} e^{-\frac{d(z_j, a_i)}{\epsilon}}, \tag{10}$$

and instead of solving the non-differentiable optimization problem (9), we can solve the following differentiable optimization problem (see [11, 22])

$$\min_{z_1,\ldots,z_k \in \mathbb{R}^n} F_\epsilon(z_1, \ldots, z_k). \tag{11}$$

Let us note that $\hat{\theta} := (\hat{c}_1, \ldots, \hat{c}_k) \in \mathbb{R}^{nk}$ is a stationary point of the functional F_ϵ if and only if for every $j \in J$ holds

$$\hat{c}_j = \frac{1}{\sum_{l=1}^{m} \omega_l^{(j)}(\epsilon)} \sum_{i=1}^{m} \omega_i^{(j)}(\epsilon) a_i, \quad \text{where} \quad \omega_i^{(j)}(\epsilon) = \frac{e^{-\frac{d(\hat{c}_j, a_i)}{\epsilon}}}{\sum\limits_{s=1}^{k} e^{-\frac{d(\hat{c}_s, a_i)}{\epsilon}}}, \quad i \in I,\ j \in J. \tag{12}$$

Therefore, the stationary point $\hat{\theta} := (\hat{c}_1, \ldots, \hat{c}_k) \in \mathbb{R}^{nk}$ of the functional F_ϵ can be searched for by the following iterative procedure

$$c_j^{(t+1)} = \frac{1}{\sum\limits_{l=1}^{m} \omega_l^{(j)}(\epsilon)} \sum_{i=1}^{m} \omega_i^{(j)}(\epsilon) a_i, \quad \text{where} \quad \omega_i^{(j)}(\epsilon) = \frac{e^{-\frac{d(c_j^{(t)}, a_i)}{\epsilon}}}{\sum\limits_{s=1}^{k} e^{-\frac{d(c_s^{(t)}, a_i)}{\epsilon}}}, \quad t = 0, 1, \ldots, \tag{13}$$

whereby $\theta^{(0)} = (c_1^{(0)}, \ldots, c_k^{(0)}) \in \mathbb{R}^{nk}$ is some initial approximation – initial assignment points. In every step, iterative procedure (13) determines the next approximation of the j-th component of vectors of centers θ as a weighted arithmetic mean of data $a_i \in \mathcal{A}$ with weights $\omega_i^{(j)}(\epsilon)$. In literature, this iterative procedure is called the `smoka` algorithm [11, 12].

From the construction it can be seen that this algorithm is numerically very demanding and practically it cannot compete with the k-means algorithm.

The properties of iterative procedure (13) are given in [11, 22], and sufficient conditions under which functional F_ϵ in the stationary point attains its local minimum are given in [21]. Specially, in [18], this problem is considered for an l_1-metric function.

2.3 Divided k-means Algorithm

In this section, we will analyze properties of weighted functions $\epsilon \mapsto \omega_i^{(j)}(\epsilon)$, $i \in I$, $j \in J$ used in iterative procedure (13) and define a new algorithm for searching for the locally optimal partition.

Suppose we are given a set of data $\mathcal{A}$ and a set of mutually different assignment points $z_1, \ldots, z_k$. As already mentioned in Section 2.2, the `smoka` algorithm is determined by iterative procedure (13), which in every step of the given assignment points defines new centers as weighted arithmetical means of data $a_i \in \mathcal{A}$ with weights $\omega_i^{(j)}(\epsilon)$ given by

$$\omega_i^{(j)}(\epsilon) = \frac{e^{-\frac{d(z_j, a_i)}{\epsilon}}}{\sum\limits_{s=1}^{k} e^{-\frac{d(z_s, a_i)}{\epsilon}}}, \quad i \in I,\ j \in J. \tag{14}$$

Note that weights (14) satisfy the following simple properties:

$$0 < \omega_i^{(j)}(\epsilon) < 1, \tag{15}$$

$$\sum_{j=1}^{k} \omega_i^{(j)}(\epsilon) = 1. \tag{16}$$

Specially, if $k = |J| = 1$, then $\omega_i^{(1)}(\epsilon) = 1$ for every $i \in I$.

Furthermore, for every $a_i \in \mathcal{A}$ define a set of indexes of the nearest assignment points

$$U_i = \{j \in J \colon d(z_j, a_i) \leq d(z_s, a_i),\ \forall s \in J\}. \tag{17}$$

Note that the set U_i is unempty, and that it can be a single member set (if $a_i \notin V[z_1, \ldots, z_k]$) or a multi-member set (if $a_i \in V[z_1, \ldots, z_k]$). If for every $a_i \in \mathcal{A}$ the set U_i is a single member set,

then a corresponding partition $\Pi = \{\pi(z_1), \ldots, \pi(z_k)\}$ is said to be a *well-separated partition*, i.e. the partition Π is said to be a well-separated partition if and only if the following holds

$$(\forall a_i \in \mathcal{A})(\exists j \in J) \quad d(z_j, a_i) < d(z_s, a_i), \quad \forall\, s \in J \setminus \{j\}. \tag{18}$$

Remark 1. Let ϵ_M be the machine epsilon number (see e.g. [3]). The set U_i defined by (17) can be determined as follows

$$\begin{aligned}
&U_i = \emptyset; d_{min} := \min_{s \in J} d(z_s, a_i);\\
&\text{For } j = 1, \ldots, k\\
&\quad \Delta_j := (d(z_j, a_i) - d_{min});\\
&\quad \text{If } \Delta_j < \phi(\epsilon_M),\\
&\qquad U_i = U_i \cup \{j\};
\end{aligned}$$

where $\phi(\epsilon_M)$ is a calculation error due to machine precision.

Lemma 1. *Let $A = \{a_i \colon i \in I\}$ be a set of data points, and $z_1, \ldots, z_k$, $k > 1$, a set of assignment points. Let U_i, $|U_i| = \mu_i \leq k$ be the set of indices associated to element $a_i \in \mathcal{A}$ by* (17).

(*i*) *If $\mu_i < k$, for functions given by* (14) *for every $i \in I$ holds*

$$\begin{aligned}
v_i^{(j)} := \lim_{\epsilon \to 0^+} \omega_i^{(j)}(\epsilon) &= \begin{cases} \frac{1}{\mu_i}, & \text{if } j \in U_i \\ 0, & \text{if } j \in J \setminus U_i, \end{cases} \\
\sum_{j \in U_i} v_i^{(j)} &= \sum_{j \in U_i} \tfrac{1}{\mu_i} = 1,
\end{aligned} \tag{19}$$

whereby functions $\epsilon \mapsto \omega_i^{(r)}(\epsilon)$, $r \in U_i$ are strictly monotonically decreasing on the interval $\langle 0, +\infty \rangle$;

(*ii*) *If $\mu_i = k$, then for every $j \in J$ and every $\epsilon \in \langle 0, +\infty \rangle$ functions $\epsilon \mapsto \omega_i^{(j)}(\epsilon) = \frac{1}{k}$ are constants.*

Proof. (*i*) Let us choose $r \in U_i$ and denote function $\epsilon \mapsto \omega_i^{(j)}(\epsilon)$ given by (14) as

$$\omega_i^{(j)}(\epsilon) = \begin{cases} \dfrac{1}{\mu_i + \sum\limits_{s \in J \setminus U_i} e^{-\frac{1}{\epsilon}\left(d(z_s, a_i) - d(z_r, a_i)\right)}} & \text{if } j \in U_i, \\ \dfrac{e^{-\frac{1}{\epsilon}\left(d(z_j, a_i) - d(z_r, a_i)\right)}}{\mu_i + \sum\limits_{s \in J \setminus U_i} e^{-\frac{1}{\epsilon}\left(d(z_s, a_i) - d(z_r, a_i)\right)}} & \text{if } j \in J \setminus U_i. \end{cases} \tag{20}$$

Since $1 < k < \mu_i$, it holds that $J \setminus U_i \neq \emptyset$. Hence, in accordance with definition (17) of the set U_i, for every $r \in U_i$ and every $s \in J \setminus U_i$ it holds that $d(z_s, a_i) > d(z_r, a_i)$. Therefore, (19) follows directly from (20).

Further, for $r \in U_i$, the derivative of function $\epsilon \mapsto \omega_i^{(r)}(\epsilon)$ given by (14) can be written as

$$\frac{d}{d\epsilon}\left(\omega_i^{(r)}(\epsilon)\right) = -\frac{1}{\epsilon^2} e^{-\frac{d(z_r,a_i)}{\epsilon}} \left(\sum_{s=1}^{k} e^{-\frac{d(z_s,a_i)}{\epsilon}}\right)^{-2} \sum_{s=1}^{k} e^{-\frac{d(z_s,a_i)}{\epsilon}} \left(d(z_s,a_i) - d(z_r,a_i)\right). \tag{21}$$

Since $d(z_s,a_i) > d(z_r,a_i)$ holds for every $s \in J \setminus U_i$, from (21) it follows $\frac{d}{d\epsilon}\left(\omega_i^{(r)}(\epsilon)\right) < 0$. Hence, functions $\epsilon \mapsto \omega_i^{(r)}(\epsilon)$, $\forall r \in U_i$ are strictly monotonically decreasing on the interval $\langle 0, +\infty\rangle$.

(ii) If $\mu_i = k$, then the data a_i is situated on the border of all clusters so that $\omega_i^{(r)}(\epsilon) = \frac{1}{k}$ holds for every $\epsilon \in \langle 0, +\infty\rangle$, from where follows the assertion. □

Note that weights $v_i^{(j)}$ defined by (19) in this way retain property (7), whereas property $w_i^{(j)} \in \{0,1\}$, $i \in I$, $j \in J$, relaxes into a more general form $v_i^{(j)} \in \{0, 1, \frac{1}{2}, \ldots, \frac{1}{k}\} \subset [0,1]$, $i \in I$, $j \in J$.

By modifying the k-means algorithm (Algorithm 1) such that weights $w_i^{(j)}$ are redefined according to (19), we obtain a new algorithm that will be called the *Divided k-means Algorithm* (DKM). In this way, the effect will be such as if the data $a_i \in \mathcal{A}$ that appeared in the Voronoi diagram $V[z_1, \ldots, z_k]$ was evenly distributed to all clusters on whose borders it is located. If in every step of the k-means algorithm no data $a_i \in \mathcal{A}$ appear in the Voronoi diagram, then the DKM algorithm becomes a common k-means algorithm. Similarly to the k-means algorithm, such algorithm is finite and in every step it reduces the objective function value.

Algorithm 2. (Divided k-means Algorithm – DKM)

Step 0: Input $1 \le k \le m$; $I = \{1, \ldots, m\}$; $J = \{1, \ldots, k\}$; $A = \{a_i \in \mathbb{R}^n : i \in I\}$.
Choose mutually different points $z_1 \ldots, z_k \in \operatorname{conv}(\mathcal{A})$.

Step 1: (Assignment step)
For each $j \in J$ define clusters $\pi(z_j) = \{a_i \in \mathcal{A} : d(z_j,a_i) \le d(z_s,a_i),\ \forall s \in J\}$.
According to Remark 1, determine sets U_i, $i \in I$ and according to (19) corresponding new weights $v_i^{(j)}$.
Calculate $F_0 = \sum_{j=1}^{k} \left(\sum_{i=1}^{m} v_i^{(j)} d(z_j,a_i)\right)$.

Step 2: (Update step) Determine centers of clusters

$$c_j = \operatorname*{argmin}_{x \in \mathbb{R}^n} \sum_{i=1}^{m} v_i^{(j)} d(x,a) = \frac{1}{\sum_{l=1}^{m} v_l^{(j)}} \sum_{i=1}^{m} v_i^{(j)} a_i, \quad j \in J. \tag{22}$$

Define new clusters $\pi(c_j) = \{a_i \in \mathcal{A} : d(c_j,a_i) \le d(c_s,a_i),\ \forall s \in J\}$, $j \in J$.
According to Remark 1, determine the sets U_i, $i \in I$ and according to (19) corresponding new weights $v_i^{(j)}$
Calculate $F_1 = \sum_{j=1}^{k} \left(\sum_{i=1}^{m} v_i^{(j)} d(c_j,a_i)\right)$;

Step 3: If $F_1 < F_0$, set $F_0 = F_1$ and go to *Step 2*.
Else set $c_j^\star = c_j$, $\forall j \in J$ and STOP.

It is obvious that $\sum_{j=1}^{k} v_i^{(j)} = 1$ holds for every $i \in I$ in *Step 1* and *Step 2*.

In contrast to the common k-means algorithm, by stopping the DKM algorithm it is possible to get a partition such that some elements lie on the border between two clusters, i.e. in the Voronoi diagram.

Example 1. *Given are the data points* $\mathcal{A} = \{a_1, \ldots, a_8\} \subset \mathbb{R}^2$, *where*

$$\mathcal{A} = \{\left(\tfrac{57}{10}, \tfrac{57}{10}\right), (3,6), \left(\tfrac{133}{30}, \tfrac{43}{30}\right), (7,3), (9,5), \left(\tfrac{280}{30}, \tfrac{203}{30}\right), (4,8), \left(\tfrac{173}{30}, \tfrac{263}{30}\right)\}$$

and initial assignment points (see Fig. 1a),

$$c_1^{(0)} = (4,4), \quad c_2^{(0)} = (8,5), \quad c_3^{(0)} = (5,8).$$

According to (19), *we associate the weights* $v_i^{(1)}, v_i^{(2)}, v_i^{(3)}$ *to each data point* $a_i \in \mathcal{A}$ *in the following way (see Fig. 1a)*

$j \backslash i$	1	2	3	4	5	6	7	8
1	1/3	1	1	0	0	0	0	0
2	1/3	0	0	1	1	1	0	0
3	1/3	0	0	0	0	0	1	1

(a) Initial approximation (b) Solution

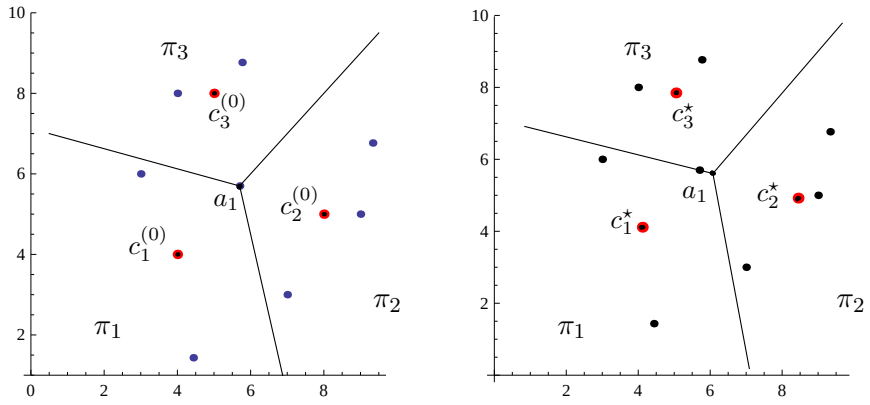

Figure 1: Divided k-means Algorithm

After two iterations of the DKM algorithm we obtain locally optimal centroids (see Fig. 1b). The corresponding clusters will be denoted as pairs by elements of the set $\mathcal{A}$ *with corresponding weights*

$$\pi_1 = \{(a_1, \tfrac{1}{2}), (a_2, 1), (a_3, 1)\}, \quad \pi_2 = \{(a_4, 1), (a_5, 1), (a_6, 1)\}, \quad \pi_3 = \{(a_1, \tfrac{1}{2}), (a_7, 1), (a_8, 1)\}.$$

Note that the element a_1 *takes place in the Voronoi diagram of an optimal partition. The flow of the iterative procedure is shown in Table 1.*

	$c_1^{(t)}$	$c_2^{(t)}$	$c_3^{(t)}$	$F(c_1^{(t)}, c_2^{(t)}, c_3^{(t)})$
t=0	(4,4)	(8,5)	(5,8)	30.6300
t=1	(4,4)	(8.17,5)	(5,8)	30.5337
t=2	(4.1133, 4.1133)	(8.4444, 4.9222)	(5.0467, 7.8467)	29.8908

Table 1: Iterative procedure

3 Properties of the DKM algorithm and connection with the k-means algorithm

Suppose that by applying the DKM algorithm we obtained centroids $c_1^\star, \ldots, c_k^\star$, whereby there exists element $a_{i_0} \in \mathcal{A}$ lying in the Voronoi diagram $V[c_1^\star, \ldots, c_k^\star]$, like e.g. in Example 1. Let us show that then the objective function value can be reduced such that by using the minimal distance principle we define a partition $\hat{\Pi} = \{\hat{\pi}_1, \ldots, \hat{\pi}_k\}$ by which element a_{i_0} is completely associated to only one of the clusters on whose edge that element lies.

Theorem 1. *Let $\mathcal{A} = \{a_i \in \mathbb{R}^n \colon i \in I\}$ be a set of data points, and let $c_1^\star, \ldots, c_k^\star \in \mathbb{R}^n$ be the centroids obtained by the DKM algorithm. Let U_i, $|U_i| = \mu_i \leq k$ be the set of indices associated to element $a_i \in \mathcal{A}$ by (17).*

If there exists $i_0 \in I$, such that $|U_{i_0}| > 1$, then there exist $\hat{c}_1, \ldots, \hat{c}_k \in \mathbb{R}^n$ such that

$$F(\hat{c}_1, \ldots, \hat{c}_k) := \sum_{i=1}^{m} \min_{1 \leq j \leq k} d(\hat{c}_j, a_i) \leq F(c_1^\star, \ldots, c_k^\star). \tag{23}$$

Proof. Let us notice that for given $c_1^\star, \ldots, c_k^\star \in \mathbb{R}^n$ and $v_i^{(j)} \in [0,1]$ given by (19), there always exists $w_i^{(j)} \in \{0,1\}$, $\sum_{j=1}^k w_i^{(j)} = 1$, such that

$$\begin{aligned} F(c_1^\star, \ldots, c_k^\star) = \sum_{i=1}^{m} \sum_{j=1}^{k} v_i^{(j)} d(c_j^\star, a_i) &\geq \sum_{i=1}^{m} \min_{1 \leq j \leq k} d(c_j^\star, a_i) \\ &= \sum_{i=1}^{m} \sum_{j=1}^{k} w_i^{(j)} d(c_j^\star, a_i) \\ &\geq \sum_{j=1}^{k} \left(\min_{x \in \mathbb{R}^n} \sum_{i=1}^{m} w_i^{(j)} d(x, a_i) \right) \\ &= \sum_{i=1}^{m} \sum_{j=1}^{k} w_i^{(j)} d(\hat{c}_j, a_i) = \hat{F}(\hat{c}_1, \ldots, \hat{c}_k), \end{aligned}$$

whereby

$$\hat{c}_j = \operatorname*{argmin}_{x \in \mathbb{R}^n} \sum_{i=1}^{m} w_i^{(j)} d(x, a_i),\ j \in J.$$

□

The following example shows how a better locally optimal partition can be found by means of an improved DKM algorithm based upon Theorem 1 in relation to a locally optimal partition obtained by the k-means algorithm.

Example 2. *By applying the DKM algorithm to the data from Example 1 we gave a locally optimal partition*

$$\pi_1 = \{(a_1, \tfrac{1}{2}), (a_2, 1), (a_3, 1)\}, \quad \pi_2 = \{(a_4, 1), (a_5, 1), (a_6, 1)\}, \quad \pi_3 = \{(a_1, \tfrac{1}{2}), (a_7, 1), (a_8, 1)\},$$

whereby element a_1, that appears in the Voronoi diagram, is divided into clusters π_1 and π_3 (see Fig. 1b) attaining in this way the objective function value $F^\star = 29.8908$.

	Centroids			Objective function value
DKM	$(4.1133, 4.1133)$	$(8.4444, 4.9222)$	$(5.0467, 7.8467)$	29.8908
Correction 1	$(4.3778, 4.3778)$	$(8.4444, 4.9222)$	$(4.8833, 8.3833)$	28.8419
Correction 2	$(3.7167, 3.7167)$	$(8.4444, 4.9222)$	$(5.1556, 7.4889)$	28.8419
k-means	$(3.7167, 3.7167)$	$(7.7583, 5.1167)$	$(4.8833, 8.3833)$	29.6997

Table 2: Iterative procedure

If the element a_1 is associated to the cluster π_1, we obtain new centers $\hat{c}_i$ and a smaller objective function value of 28.8419*. If the same element a_1 is associated to the cluster π_3, we obtain new centroids $\tilde{c}_i$ and the same smaller objective function value* 28.8419*, as can be seen in Table 2 and Fig. 2. Results obtained in such way are compared with results obtained by the k-means algorithm. By the same initial assignment points $c_i^{(0)}$, the k-means algorithm gives a weaker locally optimal partition (see Table 2)*

$$\pi_1 = \{a_2, a_3\}, \quad \pi_2 = \{a_1, a_4, a_5, a_6\}, \quad \pi_3 = \{a_7, a_8\},$$

with centroids $\bar{c}_i$ (see Fig. 2c). Hence, application of the DKM algorithm, with corrections according to Theorem 1, can give better results in comparison with the k-means algorithm.

Association of the data point a_1 to the cluster π_1 or π_3, yields lower, but mutually equal objective function values. The following sample example shows that the objective function value can differ depending on the choice of a cluster to which the data point from the Voronoi diagram is associated.

Example 3. *Given are the data points $\mathcal{A} = \{1, 2, 6, 11.4\} \subset \mathbb{R}$. Partition*

$$\Pi = \{\pi_1, \pi_2\}, \qquad \pi_1 = \{(1, 1), (2, 1), (6, \tfrac{1}{2})\}, \quad \pi_2 = \{(6, \tfrac{1}{2}), (11.4, 1)\},$$

is locally optimal in terms of the DKM algorithm, whereby the corresponding locally optimal centroids are $c_1^* = 2.4$ and $c_2^* = 8.6$, and the objective function value is $F^* = 18.32$. If the data point 6 *is associated entirely to the cluster π_1, we obtain new centroids $\hat{c}_1 = 3$ and $\hat{c}_2 = 11.4$ and the objective function value $\hat{F} = 14$. On the other hand, if the data point* 6 *is associated entirely to the cluster π_2, we obtain new centroids $\tilde{c}_1 = 1.5$ and $\tilde{c}_2 = 8.7$ and a higher objective function value $\tilde{F} = 15.08$.*

(a) Correction 1 ($a_1 \in \pi_1$) (b) Correction 2 ($a_1 \in \pi_3$) (c) Locally optimal k-means partition

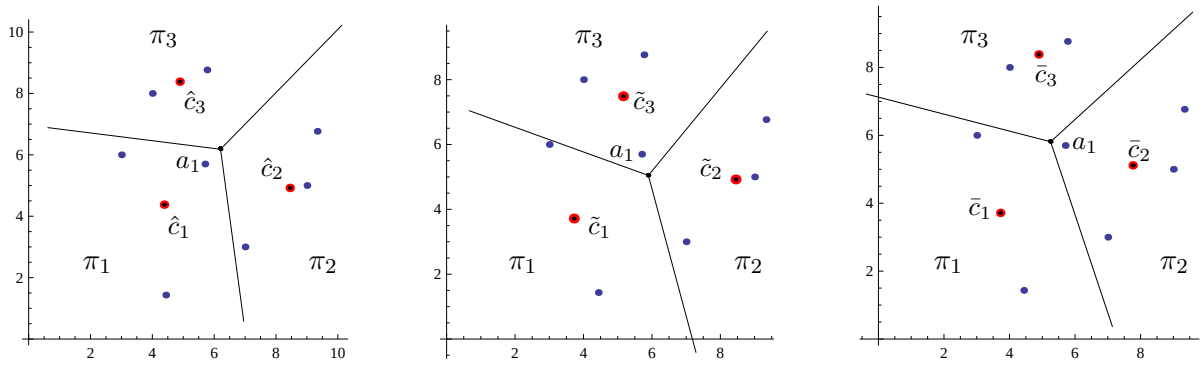

Figure 2: Locally optimal partitions

The DKM Algorithm could be modified and improved in that sense.

4 Conclusions

In this paper, we would like to point out the mathematical background of the well-known k-means algorithm for searching for the locally optimal partition of the set $\mathcal{A} = \{a_i \in \mathbb{R}^n : i = 1, \ldots, m\} \subset \mathbb{R}^n$. It has been shown that the k-means algorithm is directly connected with the limit case of another well-known algorithm for searching for the locally optimal partition, i.e. `smoka`. In this sense, a new DKM algorithm is constructed as a limit case of `smoka`, which differs from the k-means algorithm only in case if during the iterative process some data points appear in the Voronoi diagram. It has been shown that in this case the results can still be improved. In this way, the DKM algorithm gives an improvement of the well-known k-means algorithm.

Taking into account that the `smoka` algorithm came into existence as a natural generalization of the well-known Weiszfeld algorithm for solving the Fermat–Weber location problem [2, 8] for the case of applying least squares distance-like functions, cases when some other distance-like functions are applied could be treated in a similar way [13].

Acknowledgments. This work is supported by the Ministry of Science, Education and Sports, Republic of Croatia, through research grants `235-2352818-1034` and `165-0361621-2000`.

References

[1] F. Aurenhammer, R. Klein, *Voronoi diagrams*, In: J. Sack, G. Urrutia, editors, *Handbook of Computational Geometry, Chapter V*. Elsevier Science Publishing, 2000, 201–290.

[2] A. Ben-Israel, C. Iyigun, *Probabilistic d-clustering*, Journal of Classification, **25**(2008) 5–26, URL `http://dx.doi.org/10.1007/s00357-008-9002-z`, 10.1007/s00357-008-9002-z.

[3] J. J. Dennis, R. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, SIAM, Philadelphia, 1996.

[4] B. S. Everitt, S. Landau, M. Leese, *Cluster analysis*, Wiley, London, 2001.

[5] C. A. Floudas, C. E. Gounaris, *A review of recent advances in global optimization*, Journal of Global Optimization, **45**(2009) 3–38.

[6] G. Gan, C. Ma, J. Wu, *Data clustering: theory, algorithms, and applications*, SIAM, Philadelphia, 2007.

[7] C. Iyigun, *Probabilistic distance clustering*, Ph.D. thesis, Graduate School – New Brunswick, Rutgers, 2007.

[8] C. Iyigun, A. Ben-Israel, *A generalized weiszfeld method for the multi-facility location problem*, Operations Research Letters, **38**(2010) 207–214.

[9] A. K. Jain, *Data clustering: 50 years beyond k-means*, Pattern Recognition Letters, **31**(2010) 651–666.

[10] L. Kaufman, P. J. Rousseeuw, *Finding groups in data: An introduction to cluster analysis*, Jonh Wiley & Sons, Hoboken, 2005.

[11] J. Kogan, *Introduction to clustering large and high-dimensional data*, Cambridge University Press, 2007.

[12] J. Kogan, C. Nicholas, M. Wiacek, *Hybrid clustering of large high dimensional data*, In: M. Castellanos, M. W. Berry, editors, *Proceedings of the Workshop on Text Mining, SIAM*, 2007.

[13] J. Kogan, M. Teboulle, *Scaling clustering algorithms with bregman distances*, In: M. W. Berry, M. Castellanos, editors, *Proceedings of the Workshop on Text Mining at the Sixth SIAM International Conference on Data Mining*, 2006.

[14] F. Leisch, *A toolbox for k-centroids cluster analysis*, Computational Statistics & Data Analysis, **51**(2006) 526–544.

[15] M. Ng, *A note on constrained k-means algorithms*, Pattern Recognition, **33**(2000) 525–519.

[16] A. Okabe, B. Boots, K. Sugihara, *Spatial tessellations: Concepts and applications of Voronoi diagrams*, John Wiley & Sons, Chichester, UK, 2000.

[17] K. Rizman-Žalik, *An efficient k'-means clustering algorithm*, Pattern Recognition Letters, **29**(2008) 1385–1391.

[18] K. Sabo, R. Scitovski, I. Vazler, *One-dimensional center-based l_1-clustering method*, Optimization Letters, (in press) DOI: 10.1007/s11590–011–0389–9.

[19] K. Sabo, R. Scitovski, I. Vazler, M. Zekić-Sušac, *Mathematical models of natural gas consumption*, Energy Conversion and Management, **52**(2011) 1721–1727.

[20] H. Späth, *Cluster-Formation und Analyse*, R. Oldenburg Verlag, München, 1983.

[21] Z. Su, J. Kogan, *Second order conditions for k-means clustering: Partitions vs. centroids*, In: *Text Mininig 2008 Workshop (held in conjuction with the 8th Siam International Conference on Data Minining), Apr 26, 2008, Atlanta, Ga*, 2008.

[22] M. Teboulle, *A unified continuous optimization framework for center-based clustering methods*, Journal of Machine Learning Research, **8**(2007) 65–102.

[23] I. Vazler, K. Sabo, R. Scitovski, *Weighted median of the data in solving least absolute deviations problems*, Communications in Statistics - Theory and Methods, **41:8**(2012) 1455–1465.

[24] V. Volkovich, J. Kogan, C. Nicholas, *Building initial partitions through sampling techniques*, European Journal of Operational Research, **183**(2007) 1097–1105.

[25] N. Wu, J. Zhang, *Factor-analysis based anomaly detection and clustering*, Decision Support Systems, **42**(2006) 375– 389.

[26] X. S. Yang, *Firefly algorithms for multimodal optimization*, In: *Proceedings of the 5th international conference on Stochastic algorithms: foundations and applications*, 2009, 169–178.

[27] B. Yin, *Hierarchical stability based model selection for clustering algorithms*, Master's thesis, (Advisor: G.J.Hamerly), Department of Computer Science, Baylor University, 2009.