THE APPROXIMATION ALGORITHM BASED ON SEEDING METHOD FOR FUNCTIONAL $k$-MEANS PROBLEM†

MIN LI
School of Mathematics and Statistics
Shandong Normal University, Jinan 250014, China

YISHUI WANG
Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences
1068 Xueyuan Avenue, Shenzhen University Town, Shenzhen 518055, China

DACHUAN XU
Department of Operations Research and Information Engineering
Beijing University of Technology, Beijing 100124, China

DONGMEI ZHANG*
School of Computer Science and Technology
Shandong Jianzhu University, Jinan 250101, China

(Communicated by Wenxun Xing)

ABSTRACT. Different from the classical $k$-means problem, the functional $k$-means problem involves a kind of dynamic data, which is generated by continuous processes. In this paper, we mainly design an $O(\ln k)$-approximation algorithm based on the seeding method for functional $k$-means problem. Moreover, the numerical experiment presented shows that this algorithm is more efficient than the functional $k$-means clustering algorithm.

1. Introduction. The $k$-means problem aims to separate a given set with $n$ data into $k (k \leq n)$ parts in order to minimize the sum of squared distances. This is one of the classic problems in computational geometry and machine learning. Since the $k$-means problem is NP-hard [3], there are several algorithms presented in the literature based on different techniques. So far, the best approximation factor for $k$-means problem is $6.357 + \epsilon$, which is obtained by using primal-dual algorithm [2]. A more popular algorithm due to Lloyd [14,16] is called Lloyd’s algorithm or $k$-means, which performs very well in practice but belongs to heuristic algorithms. Moreover, it also belongs to Top 10 algorithms in data mining [23]. In order to improve the performance guarantee, an $O(\ln k)$-approximation algorithms based on Lloyd’s algorithm is designed, called $k$-means++, where the first initial $k$ centers are chosen with a specific probability [4]. By applying the technique of bi-criteria, Wei [22] designs a new seeding algorithm with a constant factor. These two methods have

2020 Mathematics Subject Classification. Primary: 90C27; Secondary: 68W25.

Key words and phrases. functional $k$-means, $k$-means, seeding algorithm, approximation algorithm.

* Corresponding author: Dongmei Zhang.
† A preliminary version of this article appeared in Proceedings of COCOON 2019, pp. 387-396.
been successfully applied to the variants of \( k \)-means problem \([10,12,13]\). There is also another clustering, which is based on polyhedral conic functions algorithm \([17]\).

The given data in the standard \( k \)-means problem usually are real vectors, so the distance between two points can be expressed by Euclidean distance and the triangle inequality can be used. In fact, there are a special kind of data generated by continuous processes, which are also called dynamic data or functional data. The observation point in functional \( k \)-means problem belongs to above special data \([7,18]\). By using the property of the continuous functions, several kinds of clustering methods have been introduced for functional \( k \)-means problem, such as two-stage methods \([1,11,19]\), non-parametric clustering methods \([5,21]\), model-based clustering methods \([6]\), and so on. One can refer to \([9]\) for more information.

In \([15]\), by involving the derivative information in the distance between two functional data, Meng et al. define a new distance and apply the Lloyd’s algorithm of \( k \)-means problem to solve this functional \( k \)-means problem. The numerical experiment shows this method is very efficient. However, there is lack of theoretical analysis to explain how good or bad this algorithm could get. That is, the algorithm presented by Meng et al. belongs to heuristic algorithms. In this paper, we will design an approximation algorithm for functional \( k \)-means problem, by using the new distance defined in \([15]\) and applying the seeding algorithm for \( k \)-means problem. Moreover, the approximation ratio is obtained as \( 8(\ln k + 2) \).

The rest of this paper is organized as follows. In Section 2, the functional \( k \)-means problem and some basic notations are presented. In Section 3, We mainly introduce the approximation algorithm based on seeding method, as well as the main result for functional \( k \)-means problem. The proof to show the correctness of the algorithm is given in Section 4. In Section 5, the numerical experiment about the seeding algorithm for functional \( k \)-means problem is presented. And the final remarks are concluded in Section 6.

2. Preliminaries. In this section, we mainly present the definition of the functional \( k \)-means problem, as well as some symbols and notations used in the following part. For the convenience of distinction these notations, we also explain their meanings in Table 1.

In general, given two real numbers \( R_1, R_2 \) such that \( R_1 \leq R_2 \), we use \( R \) to denote an interval from \( R_1 \) to \( R_2 \), i.e., \( R = [R_1, R_2] \). For any \( t \in R \), the function \( x(t) : R \rightarrow \mathbb{R} \) is called a functional curve, which is one real valued function and assumed to be continuous. If \( x_1(t), x_2(t), \ldots, x_d(t) \) are the functional curves with the same ground set \( R \), then the functional sample \( X(t) = (x_1(t), x_2(t), \ldots, x_d(t))^T \) can be given. And we denote \( \mathfrak{F}^d(t) \) as the set of all \( d \)-dimensional functional samples, whose functional curves have the same ground set. Therefore, given two functional samples \( X^i(t) = (x^i_1(t), x^i_2(t), \ldots, x^i_d(t))^T \in \mathfrak{F}^d(t) \), \( X^j(t) = (x^j_1(t), x^j_2(t), \ldots, x^j_d(t))^T \in \mathfrak{F}^d(t) \), their similarity metric can be defined in the following way,

\[
d(X^i(t), X^j(t)) = \sqrt{\sum_{p=1}^{d} \left[ \int_{R} (x^i_p(t) - x^j_p(t))^2 dt + \int_{R} (\nabla x^i_p(t) - \nabla x^j_p(t))^2 dt \right]},
\]

where \( \nabla x^i_p(t) \) denotes the first order derivative of \( x^i_p(t) \), which is the \( p \)-th functional curve in the \( i \)-th functional sample \( X^i(t) \). Meng et al. have shown that this metric is a distance metric in \([15]\). Given a functional sample \( X(t) \in \mathfrak{F}^d(t) \) and a set of functional samples \( \Gamma(t) \subseteq \mathfrak{F}^d(t) \), if \( X(t)_{\Gamma(t)} \) is the nearest functional sample to \( X(t) \)
Table 1. Notation Index

| Notations          | Meaning of symbols                                      |
|--------------------|---------------------------------------------------------|
| $x_i(t)$ or $x^*_j(t)$ | functional curve                                      |
| $X(t)$ or $X^i(t)$ or $C^i(t)$ | $d$-dimensional functional sample with functional curves as its components |
| $\mathfrak{F}^d(t)$ | set of $d$-dimensional functional samples              |
| $\Gamma(t)$ or $\Delta(t)$ or $C(t)$ | subset of $\mathfrak{F}^d(t)$                      |
| $\mu(\Gamma(t))$ | center of mass of functional samples of $\Gamma(t)$ |
| $d(X^i(t), X^j(t))$ | distance between the functional samples $X^i(t)$ and $X^j(t)$ |
| $d(X^i(t), \Gamma(t))$ | distance from the functional sample $X^i(t)$ to the subset of functional samples $\Gamma(t)$ |
| $X(t|_{\Gamma(t)})$ | the closest functional sample in $\Gamma(t)$ to the functional sample $X(t)$ |
| $\Phi(\Gamma(t), C(t))$ | potential function of $\Gamma(t)$ over $C(t)$ |
| $\Gamma^*_C(t)$ | the $i$-th cluster of $\Gamma(t)$ with respect to $C(t)$ |

in $\Gamma(t)$, i.e.,

$$X(t|_{\Gamma(t)}) \in \arg \min_{Y(t) \in \Gamma(t)} d(X(t), Y(t)),$$

we can give the distance of $X(t)$ to $\Gamma(t)$ as follows:

$$d(X(t), \Gamma(t)) = d(X(t), X(t|_{\Gamma(t)})).$$

Assume that $\Gamma(t)$ is a set with $n$ functional samples in $\mathfrak{F}^d(t)$ and $C(t)$ is a set with $k$ functional samples in $\mathfrak{F}^d(t)$ (also called clustering of functional samples or clustering for short), we define the loss function or potential function of $\Gamma(t)$ over $C(t)$ as the sum of the squared similarity distances over each functional sample of $\Gamma(t)$ to the clustering $C(t)$, i.e.,

$$\Phi(\Gamma(t), C(t)) = \sum_{X(t) \in \Gamma(t)} d^2(X(t), C(t)).$$

The functional $k$-means problem is to find one clustering $C(t|_{\Gamma(t)})^*$ minimizing the potential function. That is,

$$C(t|_{\Gamma(t)})^* \in \arg \min_{|C(t)| = k, C(t) \subseteq \mathfrak{F}^d(t)} \left( \sum_{X(t) \in \Gamma(t)} d^2(X(t), C(t)) \right).$$

Given an optimal clustering set $C(t|_{\Gamma(t)})^*$ of $\Gamma(t)$, $\Phi^*(\Gamma(t)) = \Phi(\Gamma(t), C(t|_{\Gamma(t)})^*)$ denotes the corresponding optimal cost for short. Specially, when $k = 1$, the clustering of $\Gamma(t)$ denoted by $\mu(\Gamma(t))$ can be given by $\mu(\Gamma(t)) = \frac{1}{n} \sum_{X(t) \in \Gamma(t)} X(t)$ in [15], which is also called the center of mass of functional samples $\Gamma(t)$. We will present the corresponding result in Lemma 2.1 as well as a simpler proof.
Given a clustering \( C(t) = \{C^1(t), C^2(t), \ldots, C^k(t)\} \) with \( k \) functional samples, \( \Gamma(t) \) can be partitioned into \( k \) parts according to the closest distance of functional samples in \( \Gamma(t) \) to the clustering \( C(t) \). In fact, for any \( C^i(t) \in C(t) \), we use \( \Gamma_{C(t)}^i(t) \) to denote a cluster of the functional samples, consisting of those points which are closer to \( C^i(t) \) than other centers,

\[
\Gamma_{C(t)}^i(t) = \{X(t) \in \Gamma(t) : d(X(t), C^i(t)) \leq d(X(t), C^j(t)), \forall C^j(t) \in C(t), j \neq i\}.
\]

Therefore, the loss function can be presented as follows.

\[
\Phi(\Gamma(t), C(t)) = \sum_{i=1}^{k} \sum_{X(t) \in \Gamma_{C(t)}^i(t)} d^2(X(t), C^i(t)).
\]

Specially, if \( C(t)_{\Gamma(t)}^* \) is an optimal functional sample clustering, it also can decide \( k \) clusters of \( \Gamma(t) \). These clusters will be denoted by \( \Gamma^1(t)^*, \Gamma^2(t)^*, \ldots, \Gamma^k(t)^* \). Moreover, for any \( \Delta(t) \subseteq \Gamma(t) \), we use \( \Phi(\Delta(t), C(t)) = \sum_{X(t) \in \Delta(t)} d^2(X(t), C(t)) \) to denote the contribution of \( \Delta(t) \) to the loss function. Specially, if \( \Delta(t) = \{X(t)\} \) with only one functional sample, we can denote \( \Phi(\Delta(t), C(t)) \) by \( \Phi(X(t), C(t)) \) for short.

Now, we present a novel proof of the special case for functional \( k \)-mean problem with \( k = 1 \), which is simpler than the one given in \[15\].

**Lemma 2.1.** Given any functional sample set \( \Gamma(t) = \{X^1(t), X^2(t), \ldots, X^n(t)\} \) with \( n \) functional samples in \( \mathcal{F}(t) \), then for any \( X(t) \in \mathcal{F}(t) \), we have

\[
\sum_{i=1}^{n} d^2(X^i(t), X(t)) = \sum_{i=1}^{n} d^2(X^i(t), \mu(\Gamma(t))) + nd^2(\mu(\Gamma(t)), X(t)),
\]

where \( \mu(\Gamma(t)) = \frac{1}{n} \sum_{X(t) \in \Gamma(t)} X(t) \).
Theorem. First, applying the definition of the metric between functional samples, we have

\[ \sum_{i=1}^{n} d^2(X^i(t), X(t)) \]

\[ = \sum_{i=1}^{n} \sum_{l=1}^{d} \left[ \int_{R} (x^i_l(t) - x_l(t))^2 dt + \int_{R} (\nabla x^i_l(t) - \nabla x_l(t))^2 dt \right] \]

\[ = \sum_{i=1}^{n} \sum_{l=1}^{d} \left[ \int_{R} (x^i_l(t) - \mu_l(\Gamma(t)) + \mu_l(\Gamma(t)) - x_l(t))^2 dt \right. \]

\[ + \left. \int_{R} (\nabla x^i_l(t) - \nabla \mu_l(\Gamma(t)) + \nabla \mu_l(\Gamma(t)) - \nabla x_l(t))^2 dt \right] \]

\[ = \sum_{i=1}^{n} \sum_{l=1}^{d} \left[ \int_{R} (x^i_l(t) - \mu_l(\Gamma(t)))^2 dt + \int_{R} (\nabla x^i_l(t) - \nabla \mu_l(\Gamma(t)))^2 dt \right] \]

\[ + \sum_{i=1}^{n} \sum_{l=1}^{d} \left[ \int_{R} (\mu_l(\Gamma(t)) - x_l(t))^2 dt + \int_{R} (\nabla \mu_l(\Gamma(t)) - \nabla x_l(t))^2 dt \right] \]

\[ + 2 \sum_{i=1}^{n} \sum_{l=1}^{d} \left[ \int_{R} (x^i_l(t) - \mu_l(\Gamma(t))(\mu_l(\Gamma(t))) - x_l(t)) dt \right. \]

\[ + \left. \int_{R} (\nabla x^i_l(t) - \nabla \mu_l(\Gamma(t))(\nabla \mu_l(\Gamma(t)) - \nabla x_l(t)) dt \right] \]

\[ = \sum_{i=1}^{n} n^2 d^2(X^i(t), X(t)) + n d^2(\mu(\Gamma(t)), X(t)) \]

\[ + 2 \sum_{i=1}^{n} \sum_{l=1}^{d} \left[ \int_{R} (x^i_l(t) - \mu_l(\Gamma(t))(\mu_l(\Gamma(t))) - x_l(t)) dt \right. \]

\[ + \left. \int_{R} (\nabla x^i_l(t) - \nabla \mu_l(\Gamma(t))(\nabla \mu_l(\Gamma(t)) - \nabla x_l(t)) dt \right]. \tag{2} \]

Then, according to the definition of \( \mu(\Gamma(t)) \), we can obtain the following result: for any \( l \in \{1, 2, \ldots, d\} \),

\[ \mu_l(\Gamma(t)) = \frac{1}{n} \sum_{j=1}^{n} x^j_l(t). \]

Thus, we have

\[ \sum_{i=1}^{n} (x^i_l(t) - \mu_l(\Gamma(t))) = \sum_{i=1}^{n} (x^i_l(t) - \frac{1}{n} \sum_{j=1}^{n} x^j_l(t)) = 0, \]

and

\[ \sum_{i=1}^{n} \nabla (x^i_l(t) - \mu_l(\Gamma(t))) = \nabla \sum_{i=1}^{n} (x^i_l(t) - \mu_l(\Gamma(t))) = 0. \]
Therefore, by the commutation property of summation and integrals, we get that the last two terms of the right hand in (2) are zero, i.e.,
\[
\sum_{i=1}^{n} \sum_{l=1}^{d} \int_{\mathbb{R}} (x^l_i(t) - \mu_l(\Gamma(t))(\mu_l(\Gamma(t)) - x^l_i(t))dt \right.
\]
\[
\left. + \int_{\mathbb{R}} (\nabla x^l_i(t) - \nabla \mu_l(\Gamma(t))(\nabla \mu_l(\Gamma(t)) - \nabla x^l_i(t))dt \right] = 0.
\]

At last, we present the following property, which can be easily verified.

**Property 2.2.** Given any functional sample set $\Gamma(t)$ in $\mathbb{R}^d(t)$, and two clusterings $C(t)$ and $C(t)'$ in $\mathbb{R}^d(t)$ such that $C(t) \subseteq C(t)'$, then
\[
\Phi(\Gamma(t), C(t)) \geq \Phi(\Gamma(t), C(t)')).
\]

When we add new elements to a clustering, the value of potential function of $\Gamma(t)$ to the new clustering will not be increased.

3. **The approximation algorithm based on seeding method and our main result.** In this section, we will mainly introduce an approximation algorithm based on seeding method for the functional $k$-means problem. The motivation comes from the following two algorithms: The $k$-means algorithm solving functional $k$-means problem and the seeding algorithm for $k$-means problem. From Step 1 to Step 5, we choose the initial functional sample clustering $C(t)$ consisting of $k$ functional sample centers from the functional sample set $\Gamma(t)$ with very specific probabilities. The idea is that the distance between any centers should be as big as possible. Then, the set of observation functional samples $\Gamma(t)$ is partitioned into $k$ clusters according to the nearest similarity metric distance from Step 6 to Step 8. In the following processes from Step 9 to Step 11, we update the clustering by the result given in Lemma 2.1. In fact, the renewed clustering cannot increase the cost of potential function, which also has been explained by the Lloyd’s method for $k$-means problem [8]. Therefore, we can improve the clustering in each iteration, and the algorithm will stop when the clustering is not changed. Therefore, if we can bound the cost function of the initial clustering which is returned at Step 5, the return clustering of this algorithm can be found with this property, too.

Now, we show our main result in the following theorem.

**Theorem 3.1.** Given $\Gamma(t) \subseteq \mathbb{R}^d$, assume that $C(t)$ is the clustering constructed in Algorithm 1 for $\Gamma(t)$, then the corresponding potential function satisfies
\[
E[\Phi(\Gamma(t), C(t))]] \leq 8(ln \ k + 2)\Phi^*(\Gamma(t)).
\]

4. **Proof of correctness.** In this section, we mainly present the proof that the clustering $C(t)$ returned by Algorithm 1 is an $8(ln \ k + 2)$-approximate functional clustering for functional $k$-means problem, which mainly follows the proof of $k$-means++ for $k$-means problem. In fact, if we can prove that the approximation ratio is correct for the set returned from seeding part (from Step 1 to Step 5), we can show that the result of Theorem 3.1 is also correct. The reason is that the value of loss function cannot be increased in the following iterations according to the high-level of Algorithm 1. Therefore, we can just pay attention to the first $k$ sampled functional centers in the following discussion. Since the group of clusters
Algorithm 1 The approximation algorithm based on seeding method for functional k-means problem.

**Input:** A set $\Gamma(t) \subseteq \mathbb{R}^d$ with $n$ functional samples, an integer $k$, and $C(t) := \emptyset$.

**Output:** An approximate functional $k$-means $C(t)$ of $\Gamma(t)$.

1: Choose the first functional sample center $C^1(t)$ uniformly at random from $\Gamma(t)$, then set $C(t) := C(t) \cup \{C^1(t)\}$;
2: for $i$ from 2 to $k$ do
3: Choose the functional sample center $C^i(t)$ from $\Gamma(t)$ with probability
4: $d^2(C^i(t), C(t))$ such that $\sum_{X(t) \in \Gamma(t)} d^2(X(t), C(t))$;
5: Set $C(t) := C(t) \cup \{C^i(t)\}$;
6: end for
7: for $i$ from 1 to $k$ do
8: Set the cluster $\Gamma_{C(t)}(i)(t) := \{X(t) \in \Gamma(t) : d^2(X(t), C(t)) = d^2(X(t), C^i(t))\}$;
9: end for
10: for $i$ from 1 to $k$ do
11: Update the functional sample clustering $C(t)$ by setting $C^i(t) := \mu(\Gamma_{C(t)}(i)(t))$, which is the center functional sample of $\Gamma_{C(t)}(i)(t)$;
12: end for
13: Repeat Step 6 to Step 11 until $C(t)$ no longer change;
14: Return $C(t)$.

$\Gamma^1(t)^*, \Gamma^2(t)^*, \ldots, \Gamma^k(t)^*$ is a partition of $\Gamma(t)$, and the first center is chosen uniformly from $\Gamma(t)$, this chosen center should belong to one of these clusters. That is, there must exist one cluster which is a joint cluster to $C(t)$. For this special joint cluster, if we cluster its functional samples to the first chosen center in Algorithm 1, rather than its center of mass, we can show that it doubles the value of loss function in the following lemma.

**Lemma 4.1.** Take any functional sample cluster $\Delta(t)$ of $\Gamma(t)$ with respect to the optimal clusters $C^*(t)$, and assume that $C(t)$ is a clustering with only one center; which is sampled uniformly from $\Delta(t)$, then

$$E(\Phi(\Delta(t), C(t))) = 2 \Phi(\Delta(t), C(t)^*_\Gamma(t)).$$

**Proof.** From Lemma 2.1, we have

$$E(\Phi(\Delta(t), C(t))) = \sum_{Q(t) \in \Delta(t)} \Pr(Q(t) \text{ is chosen}) \cdot \Phi(\Delta(t), Q(t))$$

$$= \sum_{Q(t) \in \Delta(t)} \frac{1}{|\Delta(t)|} \sum_{Q'(t) \in \Delta(t)} d^2(Q'(t), Q(t))$$

$$= \sum_{Q(t) \in \Delta(t)} \frac{1}{|\Delta(t)|} \left[ \sum_{Q'(t) \in \Delta(t)} d^2(Q'(t), \mu(\Delta(t))) + |\Delta(t)| d^2(\mu(\Delta(t), Q(t))) \right]$$

$$= 2 \sum_{Q(t) \in \Delta(t)} d^2(\mu(\Delta(t), Q(t)))$$

$$= 2 \Phi(\Delta(t), C(t)^*_\Gamma(t)).$$
Take any functional sample cluster $\Delta(t)$ of $C(t)$ with respect to an optimal solution $C(t)_{\Gamma(t)}^*$. Suppose that $C(t)$ is any clustering of $\Gamma(t)$ with less than $k$ centers. If we sample one center $Q(t)$ from $\Delta(t)$ with probability
\[
\sum_{X(t) \in \Gamma(t)} d^2(Q(t), C(t)),
\]
and add $Q(t)$ to $C(t)$, we have
\[
E[\Phi(\Delta(t), C(t)')] \leq 8 \Phi(\Delta(t), C(t)_{\Gamma(t)}^*),
\]
where $C(t)' = C(t) \cup \{Q(t)\}$.

Proof. By the rule of clustering, the functional samples in $\Delta(t)$ will be clustered to the functional sample in $C(t)$ or $Q(t)$ after adding $Q(t)$ to $C(t)$. Therefore, the contribution of $\Delta(t)$ to $C(t)'$ can be expressed by
\[
\Phi(\Delta(t), C(t)') = \sum_{X(t) \in \Delta(t)} \min\{d^2(X(t), C(t)), d^2(X(t), Q(t))\}.
\]

Then, we obtain the following result by the mathematical expectation of discrete variables,
\[
E[\Phi(\Delta(t), C(t)')] = \sum_{Q(t) \in \Delta(t)} \frac{d^2(Q(t), C(t))}{\sum_{X(t) \in \Delta(t)} \min\{d^2(X(t), C(t)), d^2(X(t), Q(t))\}}. \quad (3)
\]

Recall the definition of $X(t)_{C(t)}$, which denotes one closest functional sample in $C(t)$ to $X(t)$. Then the part $d^2(Q(t), C(t))$ can be bounded by the triangle inequality of the similarity metric as follows,
\[
d^2(Q(t), C(t)) \leq d^2(Q(t), X(t)_{C(t)}) \\
\leq 2d^2(X(t), X(t)_{C(t)}) + 2d^2(Q(t), X(t)) \\
\leq 2d^2(X(t), C(t)) + 2d^2(Q(t), X(t)).
\]

Summing up all $X(t) \in \Delta(t)$, we have
\[
d^2(Q(t), C(t)) \leq \frac{2}{|\Delta(t)|} \sum_{X(t) \in \Delta(t)} d^2(X(t), C(t)) + \frac{2}{|\Delta(t)|} \sum_{X(t) \in \Delta(t)} d^2(Q(t), X(t)).
\]

Thus, (3) can be relaxed as follows.
\[
E[\Phi(\Delta(t), C(t)')] \leq \sum_{Q(t) \in \Delta(t)} \left( \frac{2}{|\Delta(t)|} \sum_{X(t) \in \Delta(t)} d^2(X(t), C(t)) \right) \\
\cdot \sum_{X(t) \in \Delta(t)} \min\{d^2(X(t), C(t)), d^2(X(t), Q(t))\} \\
+ \sum_{Q(t) \in \Delta(t)} \left( \frac{2}{|\Delta(t)|} \sum_{X(t) \in \Delta(t)} d^2(Q(t), X(t)) \right) \\
\cdot \sum_{X(t) \in \Delta(t)} \min\{d^2(X(t), C(t)), d^2(X(t), Q(t))\}. \quad (4)
\]
The first term in the right hand of inequality (4) can be relaxed to

\[
\sum_{Q(t) \in \Delta(t)} \frac{2}{|\Delta(t)|} \sum_{X(t) \in \Delta(t)} \min\{d^2(X(t), C(t)), d^2(X(t), Q(t))\},
\]

which is less or equal to

\[
\sum_{Q(t) \in \Delta(t)} \frac{2}{|\Delta(t)|} \sum_{X(t) \in \Delta(t)} d^2(X(t), Q(t)).
\]

The second term in the right hand of inequality (4) also can be relaxed to

\[
\sum_{Q(t) \in \Delta(t)} \frac{2 \sum_{X(t) \in \Delta(t)} d^2(Q(t), X(t))}{|\Delta(t)|},
\]

because of \(\min\{d^2(X(t), C(t)), d^2(X(t), Q(t))\} \leq d^2(X(t), C(t))\).

Therefore, (3) can be relaxed in the following way,

\[
E(\Phi(\Delta(t), C(t)')) \leq \frac{4}{|\Delta(t)|} \sum_{Q(t) \in \Delta(t)} \sum_{X(t) \in \Delta(t)} d^2(X(t), Q(t)) = 8\Phi(\Delta(t), C(t)_{\Gamma(t)}').
\]

From above Lemmas, we find that the approximation guarantee is a constant if the \(k\) centers are chosen from different clusters, which is a partition of \(\Gamma(t)\) with respect to an optimal solution \(C(t)_{\Gamma(t)}'\). In following lemma, we will consider the case that more than one centers are sampled in the same cluster. Assume \(\Gamma^1(t)^*, \Gamma^2(t)^*, \ldots, \Gamma^k(t)^*\) are the \(k\) optimal functional sample clusters with respect to the optimal solution \(C(t)_{\Gamma(t)}'\). Now we will distinguish them into joint and disjoint clusters by the relationship between them and the current clustering \(C(t)\) returned by Algorithm 1. For any \(i \in \{1, 2, \ldots, k\}\), if \(\Gamma^i(t)^* \cap C(t) = \emptyset\), \(\Gamma^i(t)^*\) is called a disjoint cluster with \(C(t)\). Otherwise, it is called a joint cluster of \(C(t)\). And \(q\) is used to denote the number of the disjoint functional sample clusters of \(C(t)\), \(N_{C(t)} = \{X(t) \in \Gamma^i(t)^* : \Gamma^i(t)^* \cap C(t) \neq \emptyset\}\) means all the functional samples in the joint clusters, and \(D_{C(t)} = \Gamma(t) \setminus N_{C(t)}\) denotes the functional samples in the joint clusters.

**Lemma 4.3.** Let \(C(t)\) be any functional sample clustering of \(\Gamma(t)\). Now we add \(m (\leq q)\) functional samples \(Q^1(t), Q^2(t), \ldots, Q^m(t)\) from \(\Gamma(t)\) to \(C(t)\), where the chosen probability of \(Q^i(t)\) \((i = 1, 2, \ldots, m)\) is

\[
\frac{d^2(Q^i(t), C(t))}{\sum_{X(t) \in \Gamma(t)} d^2(X(t), C(t))}.
\]

If we set \(C(t)' = C(t) \cup \{Q^1(t), Q^2(t), \ldots, Q^m(t)\}\), we have

\[
E(\Phi(\Gamma(t), C(t)')) \leq \left[\Phi(D_{C(t)}, C(t)) + 8\Phi(N_{C(t)}, C(t)_{\Gamma(t)}')\right] \cdot (1 + H_m) + \frac{q - m}{q} \Phi(N_{C(t)}, C(t)),
\]

where \(H_m = 1 + \frac{1}{2} + \ldots + \frac{1}{m}\).
Proof. The proof of this lemma will be given by induction. That is, assume that this lemma holds for \((q, m - 1)\) and \((q - 1, m - 1)\), we will show that it is still correct for \((q, m)\). First, we check two special cases. One is that there is no functional samples chosen (i.e., \(m = 0\)) when \(q > 0\), the other is \((q, m) = (1, 1)\). Since the former special case can be obtained trivially by Property 2.2, we only need to consider the case \(q = m = 1\), that is, there is only one disjoint (optimal) cluster to \(C(t)\) and we will chose one functional sample, denoted by \(Q(t)\) and added to \(C(t)\). Therefore, we need to prove that the following inequality holds,

\[
E(\Gamma(t), C(t) \cup \{Q(t)\}) \leq 2[\Phi(D_{C(t)}, C(t)) + 8\Phi(N_{C(t)}, C(t)_{\Gamma(t)})].
\]

We can give the proof by discussing that the chosen functional sample \(Q(t)\) belongs to \(N_{C(t)}\) or \(D_{C(t)}\). That is,

\[
E[\Phi(\Gamma(t), C(t)'\})] = \frac{\Phi(N_{C(t)}, C(t))}{\Phi(\Gamma(t), C(t))}E[\Phi(\Gamma(t), C(t)')|Q(t) \in N_{C(t)}] + \frac{\Phi(D_{C(t)}, C(t))}{\Phi(\Gamma(t), C(t))}E[\Phi(\Gamma(t), C(t)')|Q(t) \in D_{C(t)}]
\]

\[
\leq E[\Phi(\Gamma(t), C(t)')|Q(t) \in N_{C(t)}] + \Phi(D_{C(t)}, C(t)) + E[\Phi(N_{C(t)}, C(t)')|Q(t) \in N_{C(t)}] + \Phi(D_{C(t)}, C(t))
\]

\[
\leq 2\Phi(D_{C(t)}, C(t)) + 8\Phi(N_{C(t)}, C(t)_{\Gamma(t)}).
\]

Thus, (5) can be obtained.

From now on, we will show the inductive step. The main idea is similar to the one of \((q, m) = (1, 1)\), where we will discuss the first functional sample chosen from disjoint cluster or joint cluster. And without loss of generality, we assume the functional samples added to \(C(t)\) is ordered as \(\{Q^1(t), Q^2(t), \ldots, Q^m(t)\}\), and denote \(C^i(t) = C(t) \cup \{Q^1(t), Q^2(t), \ldots, Q^i(t)\}\). Then

\[
E[\Phi(\Gamma(t), C(t)')] = \frac{\Phi(N_{C(t)}, C(t))}{\Phi(\Gamma(t), C(t))}E[\Phi(\Gamma(t), C(t)')|Q^1(t) \in N_{C(t)}] + \frac{\Phi(D_{C(t)}, C(t))}{\Phi(\Gamma(t), C(t))}E[\Phi(\Gamma(t), C(t)')|Q^1(t) \in D_{C(t)}].
\]

In the following part, we will give the bounds to \(E[\Phi(\Gamma(t), C(t)')|Q^1(t) \in N_{C(t)}]\) and \(E[\Phi(\Gamma(t), C(t)')|Q^1(t) \in D_{C(t)}]\) in two cases, respectively.
Case 1. If $Q^1(t) \in N_{C(t)}$, we have
\[
E[\Phi(\Gamma(t), C(t)')] \leq [\Phi(D_{C(t)}, C(t)) + 8\Phi(N_{C(t)}, C(t)'_{\Gamma(t)})] \cdot (1 + H_{m-1}) + \frac{q-m}{q} \Phi(N_{C(t)}, C(t))]
\]

Since $Q^1(t) \in N_{C(t)}$, both the joint and disjoint clusters will be changed when the functional sample clustering is changed from $C(t)$ to $C^1(t)$. If we assume that the (optimal) cluster containing $Q^1(t)$ is $\Delta(t)$, then $D_{C^1(t)} = D_{C(t)} \cup \Delta(t)$, $N_{C^1(t)} = N_{C(t)} \setminus \Delta(t)$, and $\Pr(Q^1(t) \in \Delta(t)) = \frac{\Phi(\Delta(t), C(t))}{\Phi(\Gamma(t), C(t))}$. Here, we should pay attention to that the probability of the event that $Q^1(t)$ is chosen as the first functional sample added to $C(t)$ is $\frac{\Phi(\Delta(t), C(t))}{\Phi(\Gamma(t), C(t))}$, denoted by $Pr_Q(t)$. By Lemma 3, we can easily obtain the following result,
\[
E[\Phi(\Delta(t)), C^1(t)] \leq 8E[\Phi(\Delta(t)), C(t)'_{\Gamma(t)}].
\]

Therefore, starting from $C^1(t)$, there is $q-1$ disjoint clusters and it should be added $m-1$ functional samples to up to $C(t)'$. Then, we can have some easy results as follows,
\[
\begin{align*}
\Phi(D_{C^1(t)}, C^1(t)) &= \Phi(D_{C(t)}, C^1(t)) + \Phi(\Delta(t), C^1(t)) \\
&\leq \Phi(D_{C(t)}, C(t)) + \Phi(\Delta(t), C^1(t)), \quad (8) \\
\Phi(N_{C^1(t)}, C^1(t)) &= \Phi(N_{C(t)}, C(t)) \\
&\leq \Phi(N_{C(t)}, C(t)) - \Phi(\Delta(t), C(t)). \quad (9)
\end{align*}
\]

Without loss of generality, in the discussion of (10), we still use $\Delta(t)$ to denote any disjoint cluster contained in $N_{C(t)}$, and $Q^1(t)$ to denote any functional sample in $\Delta(t)$,
\[
\begin{align*}
& \frac{\Phi(N_{C(t)}, C(t))}{\Phi(\Gamma(t), C(t))}E[\Phi(\Gamma(t), C(t)')|Q^1(t) \in N_{C(t)}] \\
= & \sum_{\Delta(t) \subseteq N_{C(t)}} \frac{\Phi(\Delta(t), C(t))}{\Phi(\Gamma(t), C(t))}E[\Phi(\Gamma(t), C(t)')|Q^1(t) \in \Delta(t)] \\
= & \sum_{\Delta(t) \subseteq N_{C(t)}} \frac{\Phi(\Delta(t), C(t))}{\Phi(\Gamma(t), C(t))}E[\Phi(\Gamma(t), C(t) \cup \{Q^1(t), Q^2(t), \ldots, Q^m(t)\})|Q^1(t) \in \Delta(t)] \\
= & \sum_{\Delta(t) \subseteq N_{C(t)}} \frac{\Phi(\Delta(t), C(t))}{\Phi(\Gamma(t), C(t))} \cdot \sum_{Q^1(t) \in \Delta(t)} \frac{\Phi(Q^1(t), C(t))}{\Phi(\Gamma(t), C(t))}E[\Phi(\Gamma(t), C^1(t) \cup \{Q^2(t), Q^3(t), \ldots, Q^m(t)\})].
\end{align*}
\]

(10)
Now, we will mainly discuss the expectation part in (10) by induction,

\[
E[\Phi(\Gamma(t), C^1(t) \cup \{Q^2(t), \ldots, Q^m(t)\})] \\
\leq \left[ \Phi(D_{C^1(t)}, C^1(t)) + 8\Phi(N_{C^1(t)}, C(t)^*_\Gamma(t)) \right] \cdot (1 + H_{m-1}) \\
+ \frac{q - m}{q - 1} \Phi(N_{C^1(t)}, C^1(t)) \\
\tag{8),(9}
\]

By applying (7), we know that

\[
\sum_{Q^1(t) \in \Delta(t)} \frac{\Phi(Q^1(t), C(t))}{\Phi(\Gamma(t), C(t))} [\Phi(\Delta(t), C^1(t)) - 8\Phi(N_{\Delta(t)}, C(t)^*_\Gamma(t))]
\]

\[
= \sum_{Q^1(t) \in \Delta(t)} \frac{\Phi(Q^1(t), C(t))}{\Phi(\Gamma(t), C(t))} \Phi(\Delta(t), C^1(t)) - 8\Phi(N_{\Delta(t)}, C(t)^*_\Gamma(t)) \\
\leq 8\Phi(N_{\Delta(t)}, C(t)^*_\Gamma(t)) - 8\Phi(N_{\Delta(t)}, C(t)^*_\Gamma(t)) = 0.
\]

Therefore, we obtain that

\[
\frac{\Phi(N_{C(t)}, C(t))}{\Phi(\Gamma(t), C(t))} E[\Phi(\Gamma(t), C(t)' | Q^1(t) \in N_{C(t)}] \\
\leq \sum_{\Delta(t) \subseteq N_{C(t)}} \frac{\Phi(\Delta(t), C(t))}{\Phi(\Gamma(t), C(t))} \sum_{Q^1(t) \in \Delta(t)} \frac{\Phi(Q^1(t), C(t))}{\Phi(\Gamma(t), C(t))} \left\{ \left[ \Phi(D_{C(t)}, C(t)) \\
+ 8\Phi(N_{C(t)}, C(t)^*_\Gamma(t)) \right] \cdot (1 + H_{m-1}) \\
+ \frac{q - m}{q - 1} \Phi(N_{C(t)}, C(t)) - \Phi(\Delta(t), C(t)) \right\] \\
\leq \sum_{\Delta(t) \subseteq N_{C(t)}} \frac{\Phi(\Delta(t), C(t))}{\Phi(\Gamma(t), C(t))} \left\{ \left[ \Phi(D_{C(t)}, C(t)) + 8\Phi(N_{C(t)}, C(t)^*_\Gamma(t)) \right] \\
\cdot (1 + H_{m-1}) \\
+ \frac{q - m}{q - 1} \Phi(N_{C(t)}, C(t)) - \Phi(\Delta(t), C(t)) \right\] \\
= \frac{\Phi(N_{C(t)}, C(t))}{\Phi(\Gamma(t), C(t))} [\Phi(D_{C(t)}, C(t)) + 8\Phi(N_{C(t)}, C(t)^*_\Gamma(t)) \cdot (1 + H_{m-1}) \\
+ \frac{1}{\Phi(\Gamma(t), C(t))} \cdot \frac{q - m}{q - 1} [\Phi^2(N_{C(t)}, C(t)) - \sum_{\Delta(t) \subseteq N_{C(t)}} \Phi^2(\Delta(t), C(t))]].
\]

Then, the proof of Case 1 can be finished by using

\[
\sum_{\Delta(t) \subseteq N_{C(t)}} \Phi^2(\Delta(t), C(t)) \geq \frac{1}{q} \Phi^2(N_{C(t)}, C(t)).
\]
Case 2. If $Q^1(t) \in D_{C(t)}$, we have
\[
E[\phi(\Gamma(t), C(t))] \leq \left[\phi(D_{C(t)}, C(t)) + 8\phi(N_{C(t)}, C(t)_{\Gamma(t)})\right] \cdot (1 + H_{m-1}) \\
+ \frac{q - m + 1}{q} \phi(N_{C(t)}, C(t)).
\]

Since $Q^1(t) \in D_{C(t)}$, the clustering $C^1(t)$ does not change the classification of joint and disjoint clusters. That is, $D_{C^1(t)} = D_{C(t)}$ and $N_{C^1(t)} = N_{C(t)}$. Moreover, after adding $Q^1(t)$ to $C(t)$, we can obtain $C(t)'$ just by adding $m - 1$ functional samples to $C(t)'$, and there are still $q$ disjoint clusters to $C(t)'$. Therefore, by induction on $(q, m - 1)$ and Property 2.2, we easily get the following results,

\[
E[\phi(\Gamma(t), C(t)'') | Q^1(t) \in D_{C(t)}] = E[\phi(\Gamma(t), C(t) \cup \{Q^1(t), Q^2(t), \ldots, Q^m(t)\}) | Q^1(t) \in D_{C(t)}] \\
\leq [\phi(D_{C^1(t)}, C(t)) + 8\phi(N_{C^1(t)}, C(t)_{\Gamma(t)})] \cdot (1 + H_{m-1}) \\
+ \frac{q - m + 1}{q} \phi(N_{C^1(t)}, C^1(t))
\]

Property 2.2

\[
\leq [\phi(D_{C^1(t)}, C(t)) + 8\phi(N_{C^1(t)}, C(t)_{\Gamma(t)})] \cdot (1 + H_{m-1}) \\
+ \frac{q - m + 1}{q} \phi(N_{C^1(t)}, C(t)).
\]

The proof of Case 2 is finished.

Then, by taking (6), Case 1, and Case 2 together, we have
\[
E[\phi(\Gamma(t), C(t)')] \leq \frac{\phi(N_{C(t)}, C(t))}{\phi(\Gamma(t), C(t))} \left[\phi(D_{C(t)}, C(t)) + 8\phi(N_{C(t)}, C(t)_{\Gamma(t)})\right] \\
\cdot (1 + H_{m-1}) + \frac{q - m}{q} \phi(N_{C(t)}, C(t)) \\
+ \frac{\phi(D_{C(t)}, C(t))}{\phi(\Gamma(t), C(t))} \left[\phi(D_{C(t)}, C(t)) + 8\phi(N_{C(t)}, C(t))_{\Gamma(t)}\right] \\
\cdot (1 + H_{m-1}) + \frac{q - m + 1}{q} \phi(N_{C(t)}, C(t))
\]

\[
\leq [\phi(D_{C(t)}, C(t)) + 8\phi(N_{C(t)}, C(t))_{\Gamma(t)}] \cdot (1 + H_{m-1} + \frac{1}{q}) \\
+ \frac{q - m}{q} \phi(N_{C(t)}, C(t)).
\]

Therefore,
\[
E[\phi(\Gamma(t), C(t)')] \leq [\phi(D_{C(t)}, C(t)) + 8\phi(N_{C(t)}, C(t)_{\Gamma(t)})] \cdot (1 + H_m) \\
+ \frac{q - m}{q} \phi(N_{C(t)}, C(t)).
\]
Now, we can present the proof of Theorem 3.1 by using above lemmas.

**Proof of Theorem 3.1.** We denote $C(t)$ as the functional sample clustering returned by Algorithm 1, whose elements are sampled from Step 1 to Step 5. We also suppose that the current functional clustering is turned by Algorithm 1, whose elements are sampled from Step 1 to Step 5. We following result, $N$ functional sample center $Q(t)$, then compare it to the standard functional $k$-means clustering algorithm based on seeding method (Algorithm 1) on the data sets $Sdata$ [20] and $Sdata$ [15], then compare it to the standard functional $k$-means clustering algorithm presented in [15]. We first introduce the two data sets as follows.

- **Simudata**
  There are two clusters in the data set. And the two clusters are decided by the functions $X_1(t) = u h_1(t) + (1 - u) h_2(t) + \epsilon(t)$ and $X_2(t) = u h_1(t) + (1 - u) h_3(t) + \epsilon(t)$, where $h_1(t) = \max(6 - |t - 11|, 0)$, $h_2(t) = h_1(t - 4)$, $h_3(t) = h_1(t + 4)$, $u$ is a random number drawn uniformly from $[0, 1]$, and $\epsilon(t)$ is a white noise with expectation $E[\epsilon(t)] = 0$ and variance $\text{Var}[\epsilon(t)] = 1$. There are 100 functional curves in each cluster, and they are generated by observation points $t = 1, 1.2, 1.4, \cdots, 21$.

- **Sdata**
  There are three clusters in this data set. The clusters are with respect to functions $X_1(t) = \cos(1.5 \pi t) + \epsilon(t)$, $X_2(t) = \sin(1.5 \pi t) + \epsilon(t)$, and $X_3(t) = \sin(\pi t) + \epsilon(t)$, where $\epsilon(t)$ is a white noise with expectation $E[\epsilon(t)] = 0$ and variance $\text{Var}[\epsilon(t)] = 1$. Each cluster also includes 100 functional curves on $[0, 1]$.

Since we need to calculate the derivatives in our distance, so we first smooth the functional data by using the technology of the third order polynomial fit. Then we use two measures to evaluate the effectiveness of the two algorithms. These two measures are adjusted rand index (ARI) and Davies Bouldin index (DBI). ARI is an external clustering validation index, which has a value between 0 and 1. A larger ARI indicates higher consistence between the clustering result and the real class labels, especially, the value 1 of ARI indicates the clustering result is the same to the real class labels. DBI is an internal clustering validation index. If DBI is smaller, the between-cluster similarity will be less. We also present the costs of the initial and returned solutions, as well as the running times of the algorithms. Since
the two algorithms are both random, we run them for 100 times and compute the average values for both measures. Results are shown in Table 2 where the notations SeedAlg and FuncAlg denote the Algorithm 1 and the functional $k$-means algorithm in [15] respectively.

From these results, we can find that the clustering accuracies of the two algorithms are very close, since the relative differences of the ARIs, DBIs, and returned costs of these two algorithms are less than 1.4%. Nevertheless, SeedAlg can produce a better initial solution (the value is 38.9% less than that of FuncAlg for Simudata, and 32.9% for Sdata) and consumes less time (5.2% less than that of FuncAlg for Simudata, and 11.3% for Sdata), implying that SeedAlg is competitive in efficiency.

6. Conclusions. At beginning, we introduce a novel proof of the property of functional $k$-mean problem when $k$ is 1, by using a new distance with derivative information. Then we design an $O(\log k)$-approximation algorithm based on seeding method for functional $k$-means problem. Moreover, we present the numerical experiment showing the effectiveness of this new algorithm, comparing to the standard functional $k$-means with the first clustering chosen at random. The result of numerical experiment shows that our new algorithm can produce a better initial solution and has a faster runtime. In future, we will plan to study the parallel seeding algorithm, bi-criteria, as well as local search method for functional $k$-means problem.

Acknowledgments. The first author is supported by Higher Educational Science and Technology Program of Shandong Province (No. J17KA171) and Natural Science Foundation of Shandong Province (No. ZR2019MA032) of China. The second author is supported by Shenzhen research grant (KQJSCX20180330170311901, JCYJ20180305180840138). The third and fourth authors are supported by National Natural Science Foundation of China (No. 11871081) and the third author is also supported by Beijing Natural Science Foundation Project No. Z200002.

REFERENCES
[1] C. Abraham, P. A. Cornillon, E. Matzner-Lober and N. Molinari, Unsupervised curve clustering using B-splines, Scandinavian Journal of Statistics, 30 (2003), 581–595.
[2] S. Ahmadian, A. Norouzi-Fard, O. Svensson and J. Ward, Better guarantees for $k$-means and Euclidean $k$-median by primal-dual algorithms, SIAM Journal on Computing, (2019), FOCS17-97–FOCS17-156.
[3] D. Aloise, A. Deshpande, P. Hansen and P. Popat, NP-hardness of Euclidean sum-of-squares clustering, Machine Learning, 75 (2009), 245–248.
[4] D. Arthur and S. Vassilvitskii, $K$-means++: The advantages of careful seeding, Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, (2007), 1027–1035.
[5] M. Boullé, Functional data clustering via piecewise constant nonparametric density estimation, *Pattern Recognition*, 45 (2012), 4389–4401.
[6] C. Bouveyron and C. Brunet-Saumard, Model-based clustering of high-dimensional data: A review, *Computational Statistics & Data Analysis*, 71 (2014), 52–78.
[7] R. Gamasaee and M. Zarandi, A new dirichlet process for mining dynamic patterns in functional data, *Information Sciences*, 405 (2017), 55–80.
[8] S. Har-Peled and B. Sadri, How fast is the $k$-means method?, *Algorithmica*, 41 (2005), 185–202.
[9] J. Jacques and C. Preda, Functional data clustering: A survey, *Advances in Data Analysis and Classification*, 8 (2014), 231–255.
[10] S. Ji, D. Xu, L. Guo, M. Li and D. Zhang, The seeding algorithm for spherical $k$-means clustering with penalties, *Journal of Combinatorial Optimization*, 2020.
[11] M. Kayano, K. Dozono and S. Konishi, Functional cluster analysis via orthonormalized Gaussian basis expansions and its application, *Journal of Classification*, 27 (2010), 211–230.
[12] M. Li, The bi-criteria seeding algorithms for two variants of $k$-means problem, *Journal of Combinatorial Optimization*, 2020.
[13] M. Li, D. Xu, J. Yue, D. Zhang and P. Zhang, The seeding algorithm for $k$-means problem with penalties, *Journal of Combinatorial Optimization*, 39 (2020), 15–32.
[14] S. Lloyd, Least squares quantization in PCM, *IEEE Transactions on Information Theory*, 28 (1982), 129–137.
[15] Y. Meng, J. Liang, F. Cao and Y. He, A new distance with derivative information for functional $k$-means clustering algorithm, *Information Sciences*, 463/464 (2018), 166–185.
[16] R. Ostrovsky, Y. Rabani, L. Schulman and C. Swamy, The effectiveness of Lloyd-type methods for the $k$-means problem, *Journal of the ACM*, 59 (2012), 28:1–28:22.
[17] G. Ozturk, M. Ciftci, Clustering based polyhedral conic functions algorithm in classification, *Journal of Industrial & Management Optimization*, 11 (2015), 921–932.
[18] J. Park and J. Ahn, Clustering multivariate functional data with phase variation, *Biometrics*, 73 (2017), 324–333.
[19] J. Peng and H. G. Müller, Distance-based clustering of sparsely observed stochastic processes, with applications to online auctions, *The Annals of Applied Statistics*, 2 (2008), 1056–1077.
[20] C. Preda, G. Saporta and C. Lévéder, PLS classification of functional data, *Computational Statistics*, 22 (2007), 223–235.
[21] T. Tarpey and K. K. Kinateder, Clustering functional data, *Journal of Classification*, 20 (2003), 93–114.
[22] Y. Wu, V. Kumar, J. Quinlan, J. Ross Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P.S. Yu, Z. H. Zhou, M. Steinbach, D. J. Hand, D. Steinberg, Top 10 algorithms in data mining, *Knowledge and Information Systems*, 14 (2008), 1–37.