Transactions on Machine Learning
and Data Mining
Vol. 3, No. 2 (2010) 67-79
©ISSN:1865-6781 (Journal),
ISBN: 978-3-940501-19-6,
IBaI Publishing ISSN 1864-9734

ibai Publishing
www.ibai-publishing.org

Fast Algorithms for Constant Approximation k-Means Clustering

Mingjun Song and Sanguthevar Rajasekaran

Computer Science and Engineering
University of Connecticut
Storrs CT 06269, USA
{mjsong, rajasek}@engr.uconn.edu

Abstract. In this paper we study the k-means clustering problem. It is well-known that the general version of this problem is NP-hard. Numerous approximation algorithms have been proposed for this problem. In this paper, we propose three constant approximation algorithms for k-means clustering. The first algorithm runs in time $O((\frac{k}{\epsilon})^k nd)$, where k is the number of clusters, n is the size of input points, and d is dimension of attributes. The second algorithm runs in time $O(k^3 n^2 \log n)$. This is the first algorithm for k-means clustering that runs in time polynomial in n, k and d simultaneously. The run time of the third algorithm $\big(O(k^5 \log^3 kd)\big)$ is independent of n. Though an algorithm whose run time is independent of n is known for the k-median problem, ours is the first such algorithm for the k-means problem.

Keywords: k-means, clustering, approximation

1 Introduction

Among the different clustering techniques known, k-means is very popular. In this problem, given a set $P \subset \Re^d$ of n data points and a number k, we are required to partition P into k subsets (i.e., clusters). Each such cluster has a center defined by the centroid (i.e., mean) of the points in that cluster. The partitioning should minimize the following cost function:

$$\triangle^P(K) = \sum_{x \in P} \|x - K(x)\|^2,$$

Where $K(x)$ denotes the nearest centroid to x, and $\|x-y\|$ denotes the Euclidean distance between two points x and y.

One of the most popular heuristic algorithms for k-means is Lloyd's algorithm [1], which initially chooses k centers randomly. For each input point, the nearest center is identified. Points that choose the same center belong to a cluster. Now new centers are calculated for the clusters. Each input point identifies its nearest center; and so on. This process is repeated until no changes occur. The process of identifying the nearest center for each input point and recomputing centers is refered to as an *iteration*. This algorithm may converge to a local minimum with an arbitrarily bad distortion with respect to the optimal solution [2].

Researches have been conducted to find algorithms with bounded quality, either $(1+\epsilon)$-approximation or constant approximation. Matousek [3] has presented a $(1+\epsilon)$-approximation algorithm with a run time of $O(n \log^k n\epsilon^{-2k^2 d})$ for any fixed $\epsilon > 0$, k, and d using the approximate centroid set idea. The centroid set was constructed by recursively subdividing the 3-enlargement cube of the bounding box of the point set P. Then the algorithm generates all well-spread k-tuples and returns the k-tuple with the minimum cost.

Kanungo et al. [2] have given a $(9+\epsilon)$-approximation algorithm. This algorithm uses an ϵ-approximate centroid set generated from the algorithm of [3] as the candidate centers. The algorithm starts with k initial centers selected from the candidate centers, and iteratively removes p centers (for some appropriate value of p) and replaces them with another p centers from the candidate centers if the resulting cost decreases. The running time is $O(n \log n + n\epsilon^{-d} \log(1/\epsilon) + n^2 k^3 \log n)$.

The algorithm of Har-Peled and Mazumdar [6] takes time $O(n + k^{k+2}\epsilon^{-(2d+1)k} \log^{k+1} n \log^k \frac{1}{\epsilon})$ to find a $(1+\epsilon)$-approximate solution to the k-means problem. If k and d are fixed, the run time is $O(n)$. The algorithm constructed a corset by sampling in an exponential grid. The authors achieved the linear time solution by combining many other known algorithms.

Kumar et al. [7] propose a simple $(1+\epsilon)$-approximation algorithm with a run time of $O(2^{(\frac{k}{\epsilon})^{O(1)}} dn)$. The idea of the algorithm is to approximate the centroid of the largest cluster by trying all subsets of constant size from the sample, and doing the same on the smaller cluster by pruning points from the larger cluster.

A problem closely related to k-means clustering is the k-median clustering problem. In this problem the objective is to minimize the sum of the distances to the nearest median. Also, the cluster centers should form a subset of the input points. Finding optimal solutions to k-means and k-median problems are NP-hard. Jain et. al. [10] even showed that it is NP-hard to obtain an approximation within a factor of $1 + \frac{2}{e}$. Thus most of the research focusses on approximation algorithms. In this paper, we focus on constant approximations to the k-means problem. None of the previous ($O(1)$-approximation) algorithms for the k-means problem run in time polynomial on n, k and d at the same time. We present three algorithms in this paper. Run time of the first one is polynomial on n and d, of the second one is polynomial on n, k and d, of the third one is polynomial on k and d while being independent of n.

2 An Algorithm Polynomial on n and d

This algorithm is inspired by the following facts: the centroid of one cluster can be approximated by the centroid of a random sample from this cluster. Also the centroid of the sample can be approximated by the closest point to the centroid of the samples. Inaba et. al. [11] showed the first approximation by the following lemma.

Lemma 1. *[11] Let P be the set of input points, T be a random sample with size of $|T|$ from P, μ_P be the centroid of P, μ_T be the centroid of T, then with probability at least $1-\delta$ ($\delta > 0$),*

$$\sum_{x_i \in P} \|x_i - \mu_T\|^2 \le (1 + \frac{1}{\delta|T|}) \sum_{x_i \in P} \|x_i - \mu_P\|^2 .$$

Let $\delta = \frac{1}{4}$. Then if we choose $|T|$ to be $\frac{4}{\epsilon}$, with a probability at least $\frac{3}{4}$, the cost computed using the centroid of the sample is $1+\epsilon$ approximation to the real cost.

We show the second approximation by the following lemma.

Lemma 2. *Let C_T be the closest point within the sample to the centroid of the sample, then with probability greater than $\frac{1}{12}$,*

$$\sum_{x_i \in P} \|x_i - C_T\|^2 \le (5 + 2\epsilon) \sum_{x_i \in P} \|x_i - \mu_P\|^2 .$$

Proof. By the doubled triangle inequality,

$$\|x_i - C_T\|^2 \le 2(\|x_i - \mu_T\|^2 + \|C_T - \mu_T\|^2).$$

With respect to the second term on the right side of the above inequality,

$$\sum_{x_i \in P} \|C_T - \mu_T\|^2 = |P| \, \|C_T - \mu_T\|^2 \le \frac{|P|}{|T|} \sum_{x_i \in P} \|x_i - \mu_T\|^2 = |P| Var(T),$$

Where $Var(T)$ is the variance of the sample and is defined as $\frac{1}{|T|} \sum_{x_i \in P} \|x_i - \mu_T\|^2$.

Let $Var(P)$ denote the variance of P, then we have[9],

$$E(Var(T)) = \frac{|T|-1}{|T|} Var(P).$$

By Markov's inequality,

$$Pr[Var(T) \le 1.5Var(P)] \ge 1 - \frac{|T|-1}{1.5|T|} > \frac{1}{3}.$$

Thus, with a probability greater than $\frac{1}{3}$,

$$\sum_{x_i \in P} \|C_T - \mu_T\|^2 \le 1.5|P|Var(P) = 1.5 \sum_{x_i \in P} \|x_i - \mu_P\|^2 .$$

Let A represent this event, B be the event of satisfying the statement of Lemma 1 (with $\delta = \frac{1}{4}$), then $Pr(AB) = 1 - Pr(\bar{A} \bigcup \bar{B}) \geq 1 - (Pr(\bar{A}) + Pr(\bar{B})) = Pr(A) + Pr(B) - 1 > \frac{3}{4} + \frac{1}{3} - 1 = \frac{1}{12}$.

Therefore, with a probability greater than $\frac{1}{12}$,

$$\begin{aligned}\sum_{x_i \in P} \|x_i - C_T\|^2 &\leq 2 \sum_{x_i \in P} \|x_i - \mu_T\|^2 + 2 \sum_{x_i \in P} \|C_T - \mu_T\|^2 \\ &\leq 2(1+\epsilon) \sum_{x_i \in P} \|x_i - \mu_P\|^2 + 3 \sum_{x_i \in P} \|x_i - \mu_P\|^2 \\ &= (5 + 2\epsilon) \sum_{x_i \in P} \|x_i - \mu_P\|^2 . \Box\end{aligned}$$

Next, we will figure out the sample size $|T|$ such that the sample would include $\frac{4}{\epsilon}$ points for each cluster with high probability. Let n_s be the size of the smallest cluster, and assume $n_s = \alpha \frac{|P|}{k}$ (A similar assumption is found in [8]). By Chernoff Bounds, we have the following inequality with respect to the number of points (X_s) falling in the smallest cluster:

$$Pr[X_s \geq \beta |T| \frac{n_s}{|P|}] \geq 1 - exp(-\frac{(1-\beta)^2}{2} |T| \frac{n_s}{|P|}),$$

and hence

$$Pr[X_s \geq \beta |T| \frac{\alpha}{k}] \geq 1 - exp(-\frac{(1-\beta)^2}{2} |T| \frac{\alpha}{k}).$$

Let $\beta |T| \frac{\alpha}{k} = \frac{4}{\epsilon}$, and $\beta = \frac{1}{2}$, then $|T| = \frac{8}{\epsilon\alpha} k$. The above probability is greater than $1 - exp(-\frac{1}{\epsilon})$.

Therefore, we get algorithm1:

1) Draw a random sample of size $\frac{8}{\epsilon\alpha} k$, where $\alpha = \frac{n_s k}{n}$, n_s is the size of the smallest cluster, and $n = |P|$.

2) Using each k-subset of sample points as centers, calculate the cost of clustering with respect to all the original input points.

3) Retrieve the k-subset that results in the minimum cost.

Theorem 1. *The output of algorithm1 is a* $(5{+}2\epsilon)$*-approximation to the optimal clustering with a probability greater than* $\frac{1}{12}$*. Algorithm1 runs in time* $O((\frac{k}{\epsilon})^k nd)$*.*

Proof. The cost of clusters from algorithm1 is less than the cost of the following clustering: Each center of the cluster is the closest point within the sample to the centroid of the sample. By Lemma 2 and simple summation, $(5 + 2\epsilon)$-approximation holds. Obviously, the running time is $O((\frac{k}{\epsilon})^k nd)$. □

3 An Algorithm Polynomial on n, k and d

In this section we present an algorithm with a running time that is polynomial on n, k and d. Kanungo et al. [2]'s local search algorithm is polynomial on n and k, but exponential on d because they used the candidate centroid sets constructed by the algorithm of [3]. In our algorithm, we employ the local search algorithm, but we use all the input points as the candidate centers instead of just the candidate centroid sets. The algorithm is described as follows:

1) Initially select an arbitrary set of k centers (S) from the input points.

2) For some integer p, swap between any subset of p' ($p' \leq p$) centers from S and p' elements from the input points if the new centers decrease the cost significantly.

3) Repeat step 2 until there is no significant cost change after several swaps.

Theorem 2. *The local search algorithm using all the input points as candidate centers yields an $O(1)$-approximation to the optimal k-means clustering problem.*

To prove this theorem, we prove some related lemmas.

Lemma 3. *Let C_P be the closest input point to the mean μ_P of the input points P. Then,*

$$\sum_{x_i \in P} \|x_i - C_P\|^2 \leq 2 \sum_{x_i \in P} \|x_i - \mu_P\|^2 .$$

[4]

Proof.

$$\begin{aligned}\sum_{x_i \in P} \|x_i - C_P\|^2 &\leq \sum_{x_i \in P} ((x_i - \mu_P) + (\mu_P - C_P))^2 \\ &= \sum_{x_i \in P} \|x_i - \mu_P\|^2 + 2 \sum_{x_i \in P} ((x_i - \mu_P)(\mu_P - C_P)) \\ &\quad + \sum_{x_i \in P} \|C_P - \mu_P\|^2 \\ &\leq \sum_{x_i \in P} \|x_i - \mu_P\|^2 + 2(\mu_P - C_P) \sum_{x_i \in P} (x_i - \mu_P) \\ &\quad + \sum_{x_i \in P} \|x_i - \mu_P\|^2 \\ &= 2 \sum_{x_i \in P} \|x_i - \mu_P\|^2 . \square\end{aligned}$$

Lemma 4. *The algorithm that enumerates all sets of k points from the input, uses them as centers, computes the clustering cost for each such set, and identifies the best set yields a 2-approximation to the optimal k-means clustering problem.*

Proof. The cost of the algorithm described in the lemma is less than the cost of the following algorithm: The center of each cluster is taken to be the closest point to the centroid of this cluster. By Lemma 3, this lemma follows. □

Next, we prove theorem 2.

Proof. We use the same construction of the set of swap pairs as [2]. The readers are referred to [2] for details. Here, we redescribe the representation of some symbols. S is a local optimal set of k centers resulting from the local search algorithm, O is the optimal set of k centers from the input points. $\triangle(O)$ denotes the cost using the optimal centers O, $\triangle(S)$ denotes the cost using the heuristic centers S. For any optimal center $o \in O$, s_o represents the closest heuristic center in S to o, $N_O(o)$ represents the neighborhood of o. For any point $q \in P$, s_q denotes the closest heuristic center to q, o_q denotes the closest optimal center to q, s_{o_q} denotes the closest heuristic center to o_q. We use $d(x, y)$ to denote the Euclidean distance between two points x and y, i.e. $\|x - y\|$, and $\triangle(x, y)$ to denote $\|x - y\|^2$.

The following two lemmas adapted from [2] will be used.

Lemma 5.

$$0 \leq \triangle(O) - 3\triangle(S) + 2R,$$

where $R = \sum_{q \in P} \triangle(q, s_{o_q})$.

Lemma 6. *Let* $\alpha > 0$ *and* $\alpha^2 = \frac{\sum_i s_i^2}{\sum_i o_i^2}$ *for two sequences of reals* $< o_i >$ *and* $< s_i >$, *then*

$$\sum_{i=1}^{n} o_i s_i \leq \frac{1}{\alpha} \sum_{i=1}^{n} s_i^2.$$

First, consider the 1-swap case. By the triangle inequality and lemma 6, we have

$$\begin{aligned}
R &= \sum_{o \in O} \sum_{q \in N_O(o)} \triangle(q, s_o) \\
&= \sum_{o \in O} \sum_{q \in N_O(o)} (\triangle(q, o) + \triangle(o, s_o) + 2d(q, o)d(o, s_o)) \\
&\leq \sum_{o \in O} \sum_{q \in N_O(o)} (\triangle(q, o) + \triangle(o, s_q) + 2d(q, o)d(o, s_q)) \\
&= \sum_{q \in P} (\triangle(q, o_q) + \triangle(o_q, s_q) + 2d(q, o_q)d(o_q, s_q)) \\
&\leq \sum_{q \in P} (\triangle(q, o_q) + \sum_{q \in P} (d(o_q, q) + d(q, s_q))^2 + 2\sum_{q \in P} d(q, o_q)(d(o_q, q) + d(q, s_q)) \\
&= 4\sum_{q \in P} \triangle(q, o_q) + \sum_{q \in P} \triangle(q, s_q) + 4\sum_{q \in P} d(q, o_q)d(q, s_q) \\
&\leq 4\triangle(O) + \triangle(S) + \frac{4}{\alpha}\triangle(S) \\
&= 4\triangle(O) + (1 + \frac{4}{\alpha})\triangle(S).
\end{aligned}$$

By lemma 5, we have

$$0 \leq \triangle(O) - 3\triangle(S) + 2(4\triangle(O) + (1 + \frac{4}{\alpha})\triangle(S)),$$

$$0 \leq 9\triangle(O) - (1 - \frac{8}{\alpha})\triangle(S),$$

$$\frac{9}{1 - \frac{8}{\alpha}} \geq \frac{\triangle(S)}{\triangle(O)} = \alpha^2,$$

$$(\alpha + 1)(\alpha - 9) \leq 0.$$

We get $\alpha \leq 9$. Therefore, $\triangle(S) \leq 81\triangle(O)$.

Second, for p-swap case, by the replacement of $2R$ with $(1 + \frac{1}{p})$ in lemma 5, we have

$$\begin{aligned} 0 &\leq \triangle(O) - (2 + \frac{1}{p})\triangle(S) + (1 + \frac{1}{p})(4\triangle(O) + (1 + \frac{4}{\alpha}\triangle(S))) \\ &= (5 + \frac{4}{p})\triangle(O) - (1 - \frac{4}{\alpha}(1 + \frac{1}{p}))\triangle(S), \end{aligned}$$

$$\frac{5 + \frac{4}{p}}{1 - \frac{4}{\alpha}(1 + \frac{1}{p}} \geq \frac{\triangle(S)}{\triangle(O)} = \alpha^2,$$

$$(\alpha + 1)(\alpha - (5 + \frac{4}{p})) \leq 0.$$

We get

$$\alpha \leq 5 + \frac{4}{p}.$$

Therefore,

$$\triangle(S) \leq (5 + \frac{4}{p})^2\triangle(O).$$

As p increases, $\frac{\triangle(S)}{\triangle(O)}$ approaches 25. Further, using lemma 4, the output of algorithm2 is a 50-approximation to the optimal k-means clustering.

The number of swaps the algorithm takes is proportional to $\log(\frac{\triangle(S_0)}{\triangle(O)})$ [5], where S_0 is the initial solution. Because $\log\triangle(S_0)$ is polynomial in n [5], the algorithm terminates after $O(k\log n)$ swaps [6]. Each swap involves nk candidate sets of centers in the worst case. For each set, computing the cost of clusters requires $O(nk)$ time. Therefore, the running time of the algorithm is $O(k^3n^2\log nd)$.

4 An Algorithm Polynomial on k and d and Independent of n

In this algorithm, we apply the sampling approach of [8] for k-median clustering. These samples resulting from this approach are processed by algorithm2 to yield a solution. The algorithm has a run time that is polynomial in k and d while being independent of n. A description of the algorithm follows.

1) Draw a random sample T of size $\frac{512k}{\alpha}\log(32k)$, where $\alpha = \frac{n_s k}{n}$, n_s is the size of the smallest cluster.

2) Do k-means clustering on the sample using algorithm2.

3) Use the centers from step 2 as the centers for all the input points P.

Replacing n in the run time of algorithm2 with the sample size, we see that the run time of Algorithm3 is $O(k^5 \log^3 kd)$.

Theorem 3. *The output of Algorithm3 is an $O(1)$-approximation to the optimal k-means clustering with probability greater than $1/32$.*

We precede the proof of this theorem with the following lemma.

Lemma 7. *For any subset of k centers $K_T \subseteq T$, and any subset of k centers $K_P \subseteq P$,*

$$\triangle^T(K_T) \leq 4\triangle^T(K_P),$$

where $\triangle^T(K_T) = \sum_{x\in T} \|x - K_T(x)\|^2$, $\triangle^T(K_P) = \sum_{x\in T} \|x - K_P(x)\|^2$, $K_T(x)$ is the closest point in K_T to x, $K_P(x)$ is the closest point in K_P to x.

Proof. Let $q_T(K_P(x))$ denote the closest point in T to $K_P(x)$. For any $x \in T$,

$$\|x - K_T(x)\|^2 \leq \|x - q_T(K_P(x))\|^2 \leq 2(\|x - K_P(x)\|^2 + \|K_P(x) - q(K_P(x))\|^2).$$

By the definition of $q_T(K_P(x))$,

$$\|K_P(x) - q_T(K_P(x))\| \leq \|x - K_P(x)\|.$$

Thus we have,

$$\|x - K_T(x)\|^2 \leq 4\|x - K_P(x)\|^2.$$

Therefore,

$$\triangle^T(K_T) \leq 4\triangle^T(K_P).\Box$$

Next, we prove the theorem. We adapt the proof from [8] given for the k-median problem.

Proof. Let $K = K_1, ..., K_k$ denote the set of centers obtained by algorithm3, $K^* = K_1^*, ..., K_k^*$ denote the optimal centroid set, $K(K_i^*)$ denote the closest center in K to K_i^*, $N(K_i^*)$ denote the neighborhood of K_i^*, $n_i = |N(K_i^*)|$, and $n_i^T = |N(K_i^*) \cap T|$. For any $x \in P$, let $K(x)$ denote the closest center in K to x, and $K^*(x)$ denote the closest center in K^* to x. Let $Q_i = \sum_{x\in N(K_i^*)} \|x - K_i^*\|^2$, $R_i = \sum_{x\in N(K_i^*)} \|x - K(x)\|^2$, $Q_i^T = \sum_{x\in N(K_i^*)\cap T} \|x - K_i^*\|^2$, and $R_i^T =$

$\sum_{x\in N(K_i^*)\cap T}\|x-K(x)\|^2$. It follows that $\sum_{1\le i\le k} Q_i = \triangle^P(K^*)$, $\sum_{1\le i\le k} R_i = \triangle^P(K)$, $\sum_{1\le i\le k} Q_i^T = \triangle^T(K^*)$, and $\sum_{1\le i\le k} R_i^T = \triangle^T(K)$.

By the doubled triangle inequality,

$$\begin{aligned}\|K_i^* - K(K_i^*)\|^2 &\le min_{x\in N(K_i^*)\cap T}\|K_i^* - K(x)\|^2\\ &\le min_{x\in N(K_i^*)\cap T} 2(\|x-K_i^*\|^2 + \|x-K(x)\|^2\\ &\le \frac{2}{n_i^T}\sum_{x\in N(K_i^*)\cap T}(\|x-K_i^*\|^2+\|x-K(x)\|^2\\ &= \frac{2}{n_i^T}(Q_i^T+R_i^T).\end{aligned}$$

For $x\in N(K_i^*)$,

$$\begin{aligned}\|x-K(x)\|^2 &\le \|x-K(K_i^*)\|^2\\ &\le 2(\|x-K^*(x)\|^2+\|K^*(x)-K(K_i^*)\|^2).\end{aligned}$$

Therefore,

$$\begin{aligned}\triangle^P(K) &\le 2\sum_{x\in P}\|x-K^*(x)\|^2 + 2\sum_{i=1}^k\sum_{x\in N(K_i^*)}\frac{2}{n_i^T}(Q_i^T+R_i^T)\\ &= 2\sum_{x\in P}\|x-K^*(x)\|^2 + 4\sum_{i=1}^k\frac{n_i}{n_i^T}(Q_i^T+R_i^T).\end{aligned}$$

By Chernoff Bounds,

$$Pr[n_i^T < \beta n_i(s/n)] < exp(-sn_i(1-\beta)^2/(2n)).$$

Assume $n_i \ge \frac{\alpha n}{k}$, then if $s = \frac{2k}{(1-\beta)^2\alpha}\log(32k)$, $Pr[n_i^T < \beta n_i(s/n)] < 1/32$. Therefore,

$$\begin{aligned}\triangle^P(K) &\le 2\triangle^P(K^*) + \frac{4n}{s\beta}\sum_{i=1}^k(Q_i^T+R_i^T)\\ &= 2\triangle^P(K^*) + \frac{4n}{s\beta}(\triangle^T(K)+\triangle^T(K^*)).\end{aligned}$$

By Lemma 4,

$$\triangle^T(K) \le c_2\triangle^T(K_T^*),$$

where $c_2 = 50$, and K_T^* is the optimal clustering for sample T.

By Lemma 7,

$$\triangle^T(K_T^*) \le 4\triangle^T(K^*).$$

Thus

$$\triangle^T(K) \le 4c_2\triangle^T(K^*),$$

and

$$\triangle^P(K) \leq 2\triangle^P(K^*) + \frac{4n}{s\beta}(1+4c_2)\triangle^T(K^*).$$

It is known that

$$E(\triangle^T(K^*)) = \frac{s-1}{n}\triangle^P(K^*).$$

By Markov's inequality,

$$Pr[\triangle^T(K^*) \leq \frac{16s}{15n}\triangle^P(K^*)] > \frac{1}{16}.$$

Let $\beta = \frac{15}{16}$ (the corresponding sample size is $\frac{512k}{\alpha}\log(32k)$). We have

$$Pr[\triangle^P(K) \leq 2\triangle^P(K^*) + 4(1+4c_2)\triangle^P(K^*)] > 1 - \frac{1}{32} - \frac{15}{16} = \frac{1}{32}.$$

Therefore, with probability greater than 1/32,

$$\triangle^S(K^*) \leq (6+16c_2)\triangle^P(K^*).\square$$

5 Experiment

In theory, algorithm 2 proposes an algorithm that is polynomial on n, k and d simultaneously, and algorithm 3 proposes an algorithm whose run time is independent on n while polynomial on k and d based on algorithm 2. These algorithms, however, obtain large approximation ratios in the worst case, which raises a concern whether these algorithms can be practical. Keeping this in mind we have run some experiments to observe the experimental approximation ratio. Since algorithm 3 is based on algorithm 2 and requires large data size, we only ran algorithm 2. The implementation is based on Mount's k-means software downloaded from http://www.cs.umd.edu/mount/Projects/KMeans/. We produced a set of 2-dimensional random data with different sizes. In particular we used the following values for n: 100, 1000, 10000, and 100000. The following values of k have been employed: 4, 9, and 16. To compute the optimal cost easily, we randomly generated data points within known clusters in the way shown in figure 1: each cluster is within a square of area 1, the gap between two contiguous squares is 1. For each data size and a given number of clusters, we generated five random data sets and computed the average approximation ratio (the cost from the algorithm divided by the optimal cost). The results are shown in tables 1 through 3.

It is obvious from experimental results that the experimental approximation ratios (close to 1) are far smaller than the theoretical ones of the worst case, which indicates that the proposed algorithm can be practical.

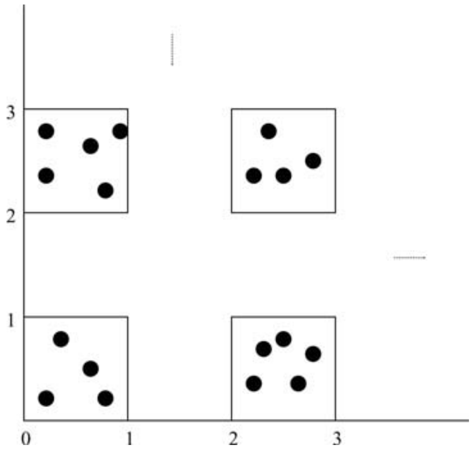

Fig. 1. Illustration for data generation

Table 1. Approximation ratios for k = 4

data size	approximation ratio
100	1.0668
1000	1.0098
10000	1.0010
100000	1.0006

Table 2. Approximation ratios for k = 9

data size	approximation ratio
100	1.1266
1000	1.0205
10000	1.0100
100000	1.0098

Table 3. Approximation ratios for k = 16

data size	approximation ratio
100	1.2405
1000	1.0629
10000	1.0564
100000	1.0579

6 Conclusion

In this paper we have proposed three $O(1)$-approximation algorithms for the k-means clustering problem. Algorithm2 is the first algorithm for the k-means problem whose run time is polynomial in n, k, and d simultaneously. There is a trade-off between the approximation ratio and the running time. Although the theoretical constant approximation ratio seems large, it is the bound in the worst case. In practice, especially when n is large, we could get better approximations. We obtained experimental approximation ratios that are close to 1. Also, the run time of Algorithm3 is independent of n. No prior algorithm for the k-means problem had this property.

Acknowledgements

The authors would like to give special thanks to the support from the grants NSF-0829916 and NSF-0326155.

References

1. Lloyd, S.P., Least squares quantization in PCM. IEEE Transactions on Information Theory 28, 129-137 (1982)
2. Kanungo, T., Mount, D.M., Netanyahu, N.S., Piatko, C.D., Silverman, R., Wu, A.Y.: A local search approximation algorithm for k-means clustering. In: Proceedings of the 18th Annual ACM Symposium on Computational Geometry, 10-18 (2002)
3. Matousek, J.: On approximate geometric k-clustering. Discrete and Computational Geometry 24, 61-84 (2000)
4. Jain, K., Vazirani, V.V.: Approximation algorithms for metric facility location and k-Median problems using the primal-dual schema and Lagrangian relaxation. Journal of the ACM 48, 272-296 (2001)
5. Arya, V., Garg, N., Khandekar, R., Pandit, V., Meyerson, A., Munagala, K.: Local search heuristics for k-median and facility location problems. In: Proceedings of the 33rd Annual Symposium on Theory of Computing, pp. 21-29, Crete, Greece (2001)
6. Har-Peled, S., Mazumdar, S.: Coresets for k-means and k-median clustering and their applications. In: Proceedings of the 36th Annual Symposium on Theory of Computing, 291-300 (2004)
7. Kumar, A., Sabharwal, Y., Sen, S.: A simple linear time $(1 + \epsilon)$-approximation algorithm for k-means clustering in any dimensions. In Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS '04), 454-462 (2004)
8. Meyerson, A., O'Callaghan, A., Plotkin, S.: A k-median algorithm with running time independent of data size. Machine Learning 56, 61-87 (2004)
9. Milton, J., Arnold, J.: Introduction to Probability and Statistics, 3rd edition, McGraw Hill (1994)
10. Jain, K., Mahdian, M., Saberi, A.:A new greedy approach for facility location problems. In: Proceedings of the 34th ACM Symposium on Theory of Computation, pp. 731-740 (2002)

11. Inaba, M., Katoh, N., Imai, H.: Applications of weighted Voronoi diagrams and randomization to variance-based k-clustering. In: Proceedings of the Tenth Annual ACM Symposium on Computational Geometry. pp. 332-339, Stony Brook, NY (1994)