Approximation Bound for K-Means clustering of Binary Data

Nikolaj Tatti [ab]

[a] *HIIT, Department of Information and Computer Science, Helsinki University of Technology, Finland*
[b] *ADReM, Mathematics and Computer Science Department, University of Antwerp, Belgium*

Abstract

We prove that a p-swap search algorithm for the K-means clustering problem has an approximation bound $3 + \frac{2}{p}$, assuming a binary data set and Euclidean distance. This is tighter than the general bound $\left(3 + \frac{2}{p}\right)^2$. We also present an example resulting in a cost ratio of $3 - \epsilon$. Thus, our bound is almost sharp for the p-swap algorithm.

1 Introduction

Clustering, grouping similar data in groups, is perhaps the most widely used application in data mining. In K-means we are asked to find k clusters such that the L_2 cost is minimised. A popular variant of K-means is K-median where the L_1 cost is used instead. Both problems are known to be **NP**-hard for higher dimensions [4].

A popular choice for approximating the K-means problem is Lloyd's algorithm. However, this algorithm can produce arbitrarily bad approximations [2]. In an alternative approach we translate the problem by finding a (large) candidate set U such that an almost optimal solution is a subset of U [3]. This subset is searched in a hill-climbing fashion by making swaps of at most p elements. The search is stopped when a local minimum is reached. It has been proved that the ratio of any local minimum and the global minimum is $3 + \frac{2}{p}$ for K-median [1] and $\left(3 + \frac{2}{p}\right)^2$ for K-means [2].

The reason for the larger ratio in K-means is that the triangle inequality does not hold for squared distances. In this paper we will show that we can use the triangle inequality if our data set is binary and we are using Euclidean distance. This result leads to a tighter ratio $3 + \frac{2}{p}$ for K-means. We also provide an almost tight example showing that the ratio cannot be improved.

2 Inequality Lemma

In this section we will introduce the inequality lemma. This crucial lemma is a triangle inequality for squared distance. We should point out that this lemma does not hold in general (real) case. For example,

$$4 = |1 - (-1)|^2 > |1 - 0|^2 + |0 - 1|^2 = 2.$$

However, if the underlying data is binary and we are using Euclidean distance, the following lemma holds

Lemma 1. *Let d be Euclidean distance, z a binary vector, and x and y real vectors inside the binary hyper-cube. Then it holds that*

$$d^2(x, y) \leq d^2(x, z) + d^2(z, y).$$

Proof. We can safely assume that z is a zero vector. If it is not, we can rotate the vectors such that the conditions still hold and the distances remain unchanged.

We can now write

$$\begin{aligned} d^2(x,y) &= \langle x-y, x-y\rangle \\ &= \langle x,x\rangle + \langle y,y\rangle - 2\langle x,y\rangle \\ &\leq \langle x,x\rangle + \langle y,y\rangle \\ &= d^2(x,z) + d^2(z,y). \end{aligned}$$

The inequality holds because x and y have only positive coordinates. This proves the lemma. □

In our proof for the bound, x and y will be candidates for centroids, meaning that the conditions imposed by the lemma hold.

Note that the lemma holds for Euclidean distance, but for instance does not hold for Manhattan distance.

3 Bound for Clustering

In this section we will state and prove our main theorem, that is, the approximation bound for the the K-means clustering problem. Our proof is essentially the same than in [2], except that we are able to apply Lemma 1.

Let us first introduce some notation. Given a set of centroids S and a data set D, a neighbourhood $N_S(s)$ for $s \in S$ is the subset of D having s as the closest centroid among S. The cost of S is defined to be

$$C(S) = \sum_{s\in S}\sum_{t\in N_S(s)} d^2(s,t).$$

Given a set of centroids S, a subset $S' \in S$, and a set O' disjoint with S such that $|S'| = |O'|$, a swap (S', O') is a procedure where we replace S' from S by O'. A set is called p-stable if its cost cannot be decreased by a swap of at most p elements. Given two sets, say S and O, of centroids. We say that $s \in S$ *captures* $o \in O$ if s is the closest point to o among S.

Theorem 1. *Assume binary data and Euclidean distance. Let S be a p-stable set and O be the optimal set. The cost $C(S)$ is bounded by $\left(3 + \frac{2}{p}\right) C(O)$.*

To prove the result we need the following technical lemmas. The proofs of these lemmas can be found in [2].

Lemma 2 ([2])**.** *Given a candidate set U and two subsets S and O having k elements, there is a set of swaps $\{(S_i, O_i)\}$ and a set of weights $\{w_i\}$ such that*

1. *For each $o \in O$, $\sum_{O_i \ni o} w_i = 1$.*
2. *For each $s \in S$, $\sum_{S_i \ni s} w_i \leq 1 + \frac{1}{p}$.*
3. *S_i does not capture elements outside O_i.*

Lemma 3 ([2])**.** *If v is a centroid for a set V, then for any w*

$$\sum_{t\in V} d^2(t,w) = \sum_{t\in V} \left(d^2(t,v) + d^2(v,w)\right).$$

Proof of Theorem 1. Let us consider a single swap (S_i, O_i). Select $s \in S_i$ and $o \in O_i$. Let $s_t \in S$ be the closest centroid for a data point t, also let $o_t \in O$ be the closest centroid among O. During the swap we need to reassign the data points. Assign the points inside $N_O(o)$ to o. This changes the cost by

$$\sum_{t\in N_O(o)} d^2(t,o) - d^2(t,s_t) = A_o.$$

The points $t \in N_S(s) - N_O(O_i)$ need to be reassigned. Let $o_t \in O$ be such that $t \in N_O(o')$. Let $s_{o_t} \in S$ be the closest centroid to o_t. Note that $o_t \notin O_i$ so, according to Lemma 2, s_{o_t} is not swapped out. Assign t to s_{o_t}. The cost change is

$$\sum_{t\in N_S(s)-N_O(O_i)} d^2(t,s_{o_t}) - d^2(t,s_t).$$

We can bound this term by

$$\sum_{t \in N_S(s)} d^2\left(t, s_{o_t}\right) - d^2\left(t, s_t\right) = B_s$$

because t is the closer to s_t than to s_{o_t}.

Since S is p-stable, by weighting with w_i and summing up we get

$$\begin{aligned}
0 &\leq \sum_i w_i \left(\sum_{o \in O_i} A_o + \sum_{s \in S_i} B_s \right) \\
&\leq \sum_t d^2\left(t, o_t\right) - \sum_t d^2\left(t, s_t\right) + \left(1 + \frac{1}{p}\right) \sum_t d^2\left(t, s_{o_t}\right) - d^2\left(t, s_t\right) \\
&= \sum_t d^2\left(t, o_t\right) - \left(2 + \frac{1}{p}\right) \sum_t d^2\left(t, s_t\right) + \left(1 + \frac{1}{p}\right) \sum_t d^2\left(t, s_{o_t}\right) \\
&= C(O) - \left(2 + \frac{1}{p}\right) C(S) + \left(1 + \frac{1}{p}\right) \sum_t d^2\left(t, s_{o_t}\right).
\end{aligned}$$

By applying Lemma 3, the last term can be written as

$$\sum_t d^2\left(t, s_{o_t}\right) = \sum_{o \in O} \sum_{t \in N_O(o)} d^2\left(t, s_o\right) = \sum_t d^2\left(t, o_t\right) + d^2\left(o_t, s_{o_t}\right).$$

Since s_{o_t} is the closest to o_t, we have

$$\sum_t d^2\left(t, o_t\right) + d^2\left(o_t, s_{o_t}\right) \leq \sum_t d^2\left(t, o_t\right) + d^2\left(o_t, s_t\right)$$

Here we can improve on the proof in [2] by applying Lemma 1

$$\sum_t d^2\left(t, o_t\right) + d^2\left(o_t, s_t\right) \leq \sum_t 2d^2\left(t, o_t\right) + d^2\left(t, s_t\right) = 2C(O) + C(S).$$

This gives us

$$\begin{aligned}
0 &\leq C(O) - \left(2 + \frac{1}{p}\right) C(S) + \left(1 + \frac{1}{p}\right) (2C(O) + C(S)) \\
&= \left(3 + \frac{2}{p}\right) C(O) - C(S).
\end{aligned}$$

□

4 A tight example

In this section we will provide an almost tight example. That is, given a parameter p and $\epsilon > 0$, we construct a binary data set D, a candidate set, and a p-stable set S such that the cost of S is at least $3 - \epsilon$ times as large as the cost of the optimal set.

To ease the notation, we define a clone operator $c_M(x)$ taking a vector $x = (x_1, \ldots, x_N)$ and resulting in a vector of length MN such that each element x_i is copied M times.

Let d, n, and m be integers to be specified later. Let Ω be the set of binary vectors having length d and only one element equal to 1. We define the data set D to be

$$D = \{(c_n(x), c_m(y)) \mid x, y \in \Omega\}.$$

Define two sets of centroids

$$O = \left\{\left(c_n(x), c_{md}(d^{-1})\right) \mid x \in \Omega\right\}$$

and

$$S = \left\{\left(c_{nd}(d^{-1}), c_m(y)\right) \mid y \in \Omega\right\}.$$

It is clear that the neighbourhoods of the centroids are

$$N_O(o) = \{(c_n(x), c_m(y)) \in D \mid c_n(x) = (o_1, \ldots, o_{nd})\}.$$

and

$$N_S(s) = \left\{(c_n(x), c_m(y)) \in D \mid c_m(y) = \left(o_{nd+1}, \ldots, o_{(n+m)d}\right)\right\}.$$

Our candidate set is $O \cup S$.

The squared distances of a data point t to its closest centroids s_t and o_t are

$$C_o = d^2(t, o_t) = m\left((d-1)d^{-2} + \left(1 - d^{-1}\right)^2\right) = mR$$

and

$$C_s = d^2(t, s_t) = nR.$$

The cost ratio is now

$$\frac{C(S)}{C(O)} = \frac{C_s}{C_o} = \frac{n}{m}.$$

Next, we will demonstrate how to choose n and m such that S is a p-stable set and the ratio n/m is at least $3 - \epsilon$. From now on, we assume that $n > m$.

Consider a p-swap (S', O'). For each $o \in O'$, reassigning the data points $N_{O'}(o)$ to o changes the cost by

$$d(C_o - C_s) = d(n - m)R.$$

Swapping $s \in S'$ out leaves at least $d - p$ points without a centroid. Let t be such a point. If t is reassigned to the closest centroid in $S - S'$, then the cost change is $2m$. It t is reassigned to a centroid in O', then the cost change is $C_o - C_s + 2n$. Since R approaches 1 as d grows, we have

$$C_o - C_s + 2n = (n - m)R + 2n \to n - m + 2n > 2m.$$

We can assume d is large enough so that the cost change for t is at least $2m$.

For S to be a p-stable set it suffices to have

$$pd(C_o - C_s) + p(d - p)2m \geq 0$$

and thus

$$\frac{n}{m} \leq \frac{dR + 2d - 2p}{dR}. \tag{1}$$

Note that R approaches 1 as d grows. Hence we have for sufficiently large d

$$\frac{dR + 2d - 2p}{dR} \geq (3 - \epsilon/2).$$

Find n and m such that

$$(3 - \epsilon) \leq \frac{n}{m} \leq (3 - \epsilon/2).$$

This satisfies the condition in Eq. 1 and thus making S a p-stable set.

References

[1] Vijay Arya, Naveen Garg, Rohit Khandekar, Kamesh Munagala, and Vinayaka Pandit. Local search heuristic for k-median and facility location problems. In *ACM Symposium on Theory of Computing*, pages 21–29, 2001.

[2] T. Kanungo, D. M. Mount, N. Netanyahu, C. Piatko, R. Silverman, and A. Y. Wu. A local search approximation algorithm for k-means clustering. *Computational Geometry: Theory and Applications*, 28:89–112, 2004.

[3] J. Matoušek. On approximate geometric k -clustering. *Discrete and Computational Geometry*, 24(1):61–84, May 2000.

[4] N. Megiddo and K. J. Supowit. On the complexity of some common geometric location problems. *SIAM Journal on Computing*, 13(1):182–196, 1984.