IOE 691: Approximation Algorithms Date: 01/25/2017

Lecture Notes: K-median (Local Search)

Instructor: Viswanath Nagarajan Scribe: Haofan Zhang

1 K-median

Definition 1.1 *The input is a set V of vertices, distance function d: $V \times V \to \mathbb{R}_+$ (symmetric and satisfies triangle inequality) and a bound k. The goal is to choose a set $S \subseteq V$, $|S| = k$, of cluster centers so as to minimize the sum of distances of each vertex to its closest center in S.*

$$\min_{S \subseteq V, |S|=k} \quad \sum_{j \in V} \min_{u \in S} d(j,u)$$

For any $S \subseteq V$ we will call $Med(S) = \sum_{j \in V} \min_{u \in S} d(j,u) = \sum_{j \in V} d(j,S)$.

Algorithm 1 Local Search Algorithm for k-median problem

Data: Metric (V,d), bound k.
start with any $S \subseteq V$ with $|S| = k$
while there is any pair $a \in V \setminus S, r \in S$ such that
/* local move */
$Med(S - r + a) < Med(S)$ **do**
$S = S - r + a$
end while

We use $(S - r + a)$ as an abbreviation for $(S \setminus \{r\}) \cup \{a\}$.

We need some definitions for the analysis. Let $L = \{l_1, l_2, \cdots, l_k\}$ be the algorithm's solution and $Q = \{q_1, q_2, \cdots, q_k\}$ be the optimal solution. Let $\pi : Q \to L$ be the map defined as $\pi(q_i) =$ closest vertex in L to q_i. Let D_j be the distance between any vertex $j \in V$ to its closest center in L; and D_j^* be the distance between vertex $j \in V$ to its closest center in Q. Let $\{L_1, L_2, \cdots, L_k\}$ denote the partition of V in solution L where each L_i denotes the vertices that are closest to center l_i. Similarly, let $\{Q_1, Q_2, \cdots, Q_k\}$ denote the partition of V in solution q where each Q_i denotes the vertices that are closest to center q_i. See Figure 1. Note that

$$OPT = \sum_{j \in V} D_j^*, \qquad ALG = \sum_{j \in V} D_j.$$

1.1 Special case: π is a bijection

We first bound the approximation ratio when π is a bijection. Without loss of generality, in the following discussion we will have $\pi(q_i) = l_i$ for each $i \in [k]$. See Figure 2

Theorem 1.1 *If π is a bijection, the local search algorithm is a 3-approximation algorithm.*

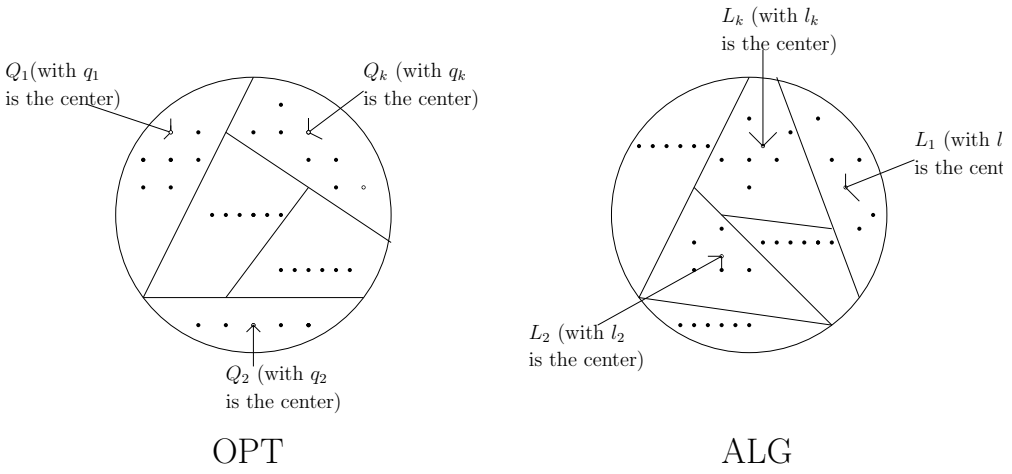

Figure 1: The partition for OPT and ALG.

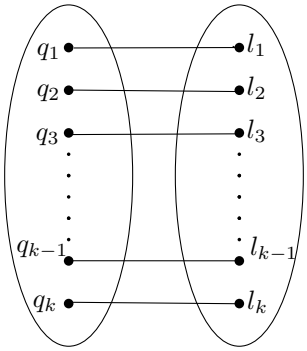

Figure 2: π is a bijection

The following lemma is the key step.

Lemma 1.1 *$Med(L - l_i + q_i) - Med(L) \leq \sum_{j \in Q_i} D_j^* - \sum_{j \in Q_i} D_j + 2 \sum_{j \in L_i} D_j^*$, for any $i \in [k]$.*

Proof: Let $L^{'} = L - l_i + q_i$. We have $Med(L') = \sum_{j \in V} d(j, L^{'})$. We now bound this term by term. The vertices $j \in V$ are divided into three categories:

(i) When $j \in Q_i$, we connect j to q_i in L', i.e. j switches its center from l_i to q_i (see Figure 3). So $d(j, L^{'}) \leq D_j^*$.

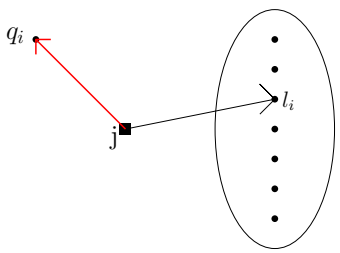

Figure 3: j changes from l_i to q_i (from black arrow to red arrow).

(ii) When $j \in L_i \backslash Q_i$, then j is assigned to $\pi(q_r)$ where q_r represents the center that j is assigned to OPT. See Figure 4. Crucially, $\pi(q_r) \neq l_i$ and so $\pi(q_r) \in L'$. Then we have:

$$\begin{aligned} d(j, \pi(q_r)) &\leq d(j, q_r) + d(q_r, \pi(q_r)) \\ &= D_j^* + d(q_r, \pi(q_r)) \\ &\leq D_j^* + d(q_r, l_i) \qquad (by\ the\ definition\ of\ \pi) \\ &\leq D_j^* + d(q_r, j) + d(j, l_i) \\ &= D_j^* + D_j^* + D_j \end{aligned}$$

So we get that $d(j, L^{'}) \leq d(j, \pi(q_r)) \leq 2D_j^* + D_j$

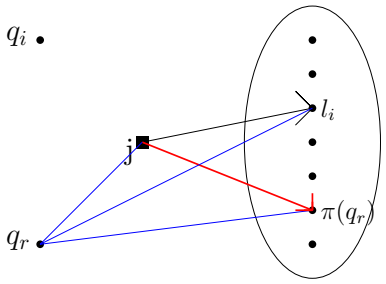

Figure 4: j changes from l_i to $\pi(q_r)$(from black arrow to red arrow).

(iii) When $j \in V \backslash (L_i \cup Q_i)$, it does not change its center, so $d(j, L^{'}) \leq D_j$.

As $-Med(L) = -\sum_{j \in V} D_j$, combining the above three situations together, we get:

$$Med(L - l_i + q_i) - Med(L) \leq \sum_{j \in Q_i} D_j^* - \sum_{j \in Q_i} D_j + 2 \sum_{j \in L_i} D_j^*$$

So we finished proving the lemma. ■

Now we will prove Theorem 1.1. By adding lemma 1.1 for all $i \in [k]$, the LHS will be $\sum_{i \in [k]} Med(L - l_i + q_i) - Med(L)$. By the definition of Local Search Algorithm, we know that $Med(L - l_i + q_i) - Med(L) \geq 0$ for all $i \in [k]$ as L is locally optimal. This means $\sum_{i \in [k]} Med(L - l_i + q_i) - Med(L) \geq 0$. For the RHS, it will become

$$\begin{aligned} \sum_{i \in [k]} (\sum_{j \in Q_i} D_j^* - \sum_{j \in Q_i} D_j + 2 \sum_{j \in L_i} D_j^*) &= \sum_{i \in [k]} \sum_{j \in Q_i} D_j^* - \sum_{i \in [k]} \sum_{j \in Q_i} D_j + 2 \sum_{i \in [k]} \sum_{j \in L_i} D_j^* \\ &= OPT - ALG - 2OPT \end{aligned}$$

So we have $0 \leq OPT - ALG + 2OPT$, which implies that $ALG \leq 3OPT$.

1.2 General π

Now we turn to general case when π is not necessarily a bijection.

Theorem 1.2 *The 1-swap local search algorithm for k-median is a 5-approximation algorithm.*

First of all, we will define the candidate swaps. Recall that $\pi : Q \to L$. Define $U \subseteq Q$ to consist of those vertices $q \in Q$ which are not uniquely mapped to $\pi(q)$, i.e.

$$U = \{q \in Q : \exists q' \in Q \setminus q \text{ with } \pi(q) = \pi(q')\}.$$

Define $Z \subseteq L$ to consist of those $l \in L$ that are not in the image of π, i.e.

$$L = \{l \in L : |\pi^{-1}(l)| = 0\}.$$

As $|U|$ needs to be at least $2(|U| - |Z|)$ by its definition, we can easily get that $|Z| \geq \frac{|U|}{2}$. Then we introduce the following candidate swaps (see Figure 5). There is one swap for each $q \in Q$:

(1) If $\pi(q)$ is unique to q (i.e. $q \in Q \setminus U$), then let the swap add q and remove $\pi(q)$. This is identical to the bijection special case.

(2) For $q \in U$, pair them arbitrary with elements in Z such that:

   (i) each $q \in U$ is added in one pair;

   (ii) each $l \in Z$ is removed in at most two pairs.

   Note that this is possible as $|U| \leq 2|Z|$.

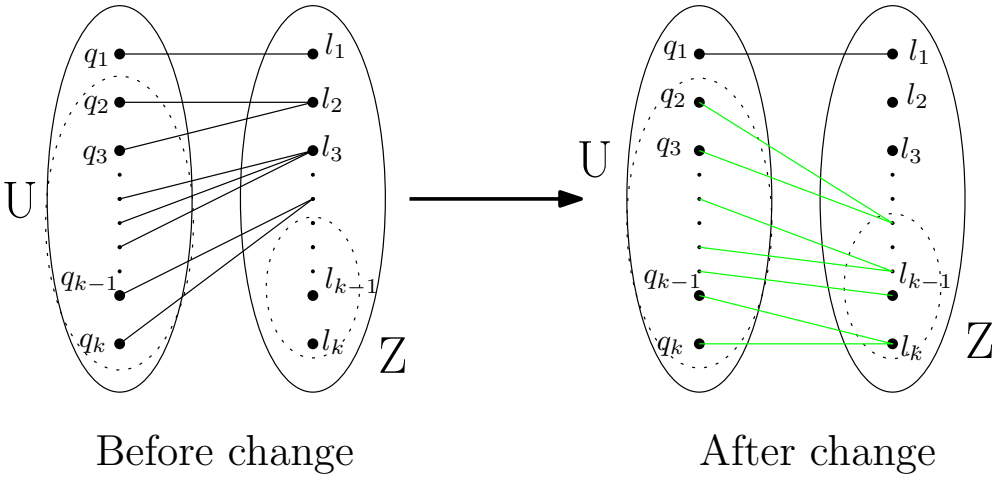

Figure 5: candidate swap

Let Y denote the set of pairs above; note that each is of the form (l, q) where $l \in L$ is removed and $q \in Q$ is added.

Lemma 1.2 *For any $(l, q) \in Y$, $Med(L - l + q) - Med(L) \leq \sum_{j \in Q_0} D_j^* - \sum_{j \in Q_0} D_j + 2 \sum_{j \in L_0} D_j^*$ where Q_0 and L_0 are the parts in $\{Q_i\}_{i=1}^k$ and $\{L_i\}_{i=1}^k$ corresponding to q and l.*

Proof: The proof is almost identical to Lemma 1.1. Let $L' = L - l + q$. For $j \in Q_0$, the situation doesn't change, so we also have $d(j, L') \leq D_j^*$. For $j \in L_0 \setminus Q_0$, we still can pick $\pi(q_r)$ as the new center for j: (i) if $q \in Q \setminus U$ then this is identical to the case in Lemma 1.1, and (ii) if $q \in U$ then we know that $l \in Z$ (i.e. $\pi^{-1}(l) = \emptyset$), so $\pi(q_r) \neq l$ and we can indeed assign j to $\pi(q_r) \in L'$. So $d(j, L') \leq d(j, \pi(q_r) \leq 2D_j^* + D_j$ still holds for $j \in L_0 \setminus Q_0$. For $j \in V \setminus (L_0 \cup Q_0)$, it is the same as

before, so we have $d(j, L^{'}) \leq D_j$. As $-Med(L) = -\sum_{j \in V} D_j$, combining the above three situations together, we get:

$$Med(L - l + q) - Med(L) \leq \sum_{j \in Q_0} D_j^* - \sum_{j \in Q_0} D_j + 2 \sum_{j \in L_0} D_j^*$$

So we finished proving the lemma. ■

Now we will finish the proof of Theorem 1.2. Use the same way as we have used in proving Theorem 1.1. By adding Lemma 1.2 for all $i \in [k]$, the LHS will be $\sum_{(l,q) \in Y} Med(L - l + q) - Med(L) \geq 0$. For the RHS, the first two items will still be $OPT - ALG$. But the third item will be different. As each center $l_i \in L$ occurs in 0,1 or 2 swaps, we can give an upper bound by double the summation $2 \sum_{i \in [k]} (\sum_{j \in L_i} D_j^*) \leq 4 \sum_{j \in V} D_j^* = 4OPT$. So we have $0 \leq OPT - ALG + 4OPT$, i.e. $ALG \leq 5OPT$.

1.3 Polynomial-time algorithm

The running time of the above local-search may depend polynomially on the distances in d, which is not adequate for a polynomial-time algorithm. Recall that the input size is polynomial in n and $\log D$ where D is the maximum distance in d (assuming all integer distances). We can ensure a polynomial runtime by making a local move only when Med decreases by an ϵ-factor. So each swap will satisfy $Med(S^{'}) - Med(S) < -\epsilon Med(S)$, where S is the original set and $S^{'}$ is the set after swap. If we start with solution S_0 and end with solution S_k after k swaps, then

$$1 \leq Med(S_k) \leq (1 - \epsilon)^k \cdot Med(S_0) \leq (1 - \epsilon)^k \cdot nD.$$

This bounds the number of iterations by $O(\frac{1}{\epsilon} \log(nD))$ which is polynomial. This also does not affect the approximation ratio much. For any L' obtained from a swap of L, we know $-\epsilon \cdot Med(L) \leq Med(L') - Med(L)$. As we only added n such swaps in the analysis, we will get a $5/(1 - n\epsilon)$ approximation ratio. Setting $\epsilon = \frac{1}{n^2}$ gives: approximation algorithm.

Theorem 1.3 *The local search algorithm for the k-median problem that uses bigger improving swaps yields a* $5 + o(1)$ *approximation algorithm.*