College of Computer Science
Northeastern University
COM 3391: Advanced Algorithms

Lecture 9
24 November 1999
Scribe: Xiaowei Sun

Facility Location: the k-Median Problem

1 Introduction

In the class of facility location problems, we are given a set of n locations, each having a demand for a certain service. And the goal is to determine a subset of locations where to place facilities for the service. In this lecture, we study the *k-median problem.* We select a set of at most k locations where facilities are opened; these are called the *k medians.* Each location is served by its closest median, and the cost incurred is proportional to the distance between them. The goal is to select the k locations such that the total cost is minimized. We will focus on the special case in which the cost form a metric.

Definition 1 (Metric k-median Problem) *The* metric k-median problem *is defined as follows*

- **Input:**
  1. *n points(locations)*
  2. *point i has demand d_i for service*
  3. *cost of shipping unit demand from point i to j is c_{ij}, which forms a metric*

$$c_{ii} = 0, \textit{ reflexivity}$$

$$c_{ij} = c_{ji}, \textit{ symmetry}$$

$$c_{ij} \leq c_{ik} + c_{kj}, \textit{for } c_{ij} \geq 0, \textit{ triangle inequality}$$

  4. *k is the maximum number of locations where facilities can be opened*

- **Solution:** *A solution to k-median problem is specified by a set S of at most k locations where facilities are being opened.*
- **Value:** *The cost is given by:*

$$cost = \sum_j d_j \cdot c_{j\sigma(j,S)}.$$

The decision version of the k-median problem is proved to be *NP-complete* by a reduction from *Dominating Set.*

Theorem 1 *The k-median problem is NP-complete.*

Proof: Obviously the decision version of k-median problem is in NP. The certificate for a given yes-instance is simply the assignment of S. It's easy to count the number of medians and to calculate the total cost given the medians. To prove that it is also NP-hard, we use a reduction from the *Dominating Set(DS)* problem.

Given an undirected graph $G = \langle V, E \rangle$, a *dominating set* is a set V' such that for any $v \in V$, either $v \in V'$ or v is adjacent to a node in V'. The decision version of DS is the following: given a graph G and a number k, is there a dominating set of G of size at most k. The reduction from *DS* to *k-median problem* is as follows.

1. Let the points of k-median problem be the vertices V.
2. For each vertex i, let $d_i = 1$.
3. Let the cost c_{ij} between two points i and j be the shortest path from node i to node j.
4. Let the cost bound $C = |V| - k$.

We now prove that the k-median problem has a total cost of at most C if and only if there is a dominating set of size at most k of G.
$\Longrightarrow$: If the k-median has a cost of at most C, there is a DS of size at most k of G. Since the cost is at most $C = |V| - k$, the k medians form a DS and each of other $|V| - k$ nodes is adjacent to some node in the DS, that is, has a shortest path of length 1.
$\Longleftarrow$: If there is a dominating set of size at most k of G, let those points in the set be medians, if size of the set is less than k, just add some more nodes as medians to compensate the exact k. Since each non-median is one hop away from a median, the total cost is exactly C, which yields the desired result. ■

2 Integer Program for k-median

Now we express the k-median problem as an 0-1 integer linear program, and then we will use LP relaxation to solve the problem. For simplicity, we assume $d_j = 1$ for all j. We introduce the following variables:

$$y_j = \begin{cases} 1 & \text{if median location} \\ 0 & \text{otherwise} \end{cases}$$

$$x_{ij} = \begin{cases} 1 & \text{if } i \text{ is serviced by } j \\ 0 & \text{if } i \text{ is not serviced by } j \end{cases}$$

The k-median problem can be formulated as the following integer linear program:

$$min \sum_i \sum_j x_{ij} \cdot c_{ij}$$

subject to

$$\sum_j x_{ij} = 1, \text{ for all } i$$

$$x_{ij} \leq y_j, \text{ for all } i, j$$

$$\sum y_j \leq k$$
$$y_i \in \{0,\, 1\}, \text{ for all } i$$
$$x_{ij} \in \{0,\, 1\}, \text{ for all } i,\, j$$

If we replace the last two constraints by

$$0 \leq y_j \leq 1$$
$$0 \leq x_{ij} \leq 1,$$

then we have an LP relaxation for the integer program. The following algorithm, due to Lin and Vitter [LV92], is based on the LP relaxation.

3 Lin-Vitter Algorithm: An LP Relaxation Algorithm

Algorithm 1 *Lin-Vitter Algorithm*

1. Solve the LP relaxation optimally by an efficient linear programming algorithm. Denote the optimal fractional solution by x^*_{ij} and y^*_j.
2. For each i, compute $C_i = \sum_j c_{ij} \cdot x^*_{ij}$. The total cost of the LP relaxation is then
$$C^{LP} = \sum_i C_i.$$
Obviously the optimal cost of the k-median problem is at least C^{LP}. An observation is that to ensure that every location has at least one facility within a short distance, that is, $(1+\epsilon)C_i$, appears to be hard subject to constraint on the number of facilities. The idea is by opening a little bit more than k facilities, we will be able to get a lower cost bound.
3. Sort all points according to C_i in non-decreasing order.
4. Given $\epsilon > 0$, for each point i, the *neighborhood* V_i of point i consists of all points j such that $c_{ij} \leq (1+\epsilon)C_i$. Furthermore, a vertex j is in the *extended neighborhood* $\bar{V}_i$ of point i if and only if one of the following rules holds:
   (a) $c_{ij} \leq (1+\epsilon)C_i$, that is, j is in the neighborhood V_i of point i.
   (b) There exists a point j' such that $c_{ij'} \leq (1+\epsilon)C_i$ and $c_{jj'} \leq (1+\epsilon)C_j$, that is, $V_i \cap V_j \neq \emptyset$.
5. Go through the points in order. For current point i,
   (a) Add point i to S.
   (b) Remove all points j in the extended neighborhood of i, that is $j \in V_i$ or $V_i \cap V_j \neq \emptyset$.

One important property about the set S of locations determined by the algorithm is that if i and j are two distinct points in S, then $V_i \cap V_j = \emptyset$.

Claim 1 *The total cost of the Lin-Vitter Algorithm, denote by C, where C satisfies*

$$C \leq 2(1+\epsilon)C^{LP}.$$

Proof: Given a point i, let $\hat{C}_i$ be the cost associated with i in the solution obtained by the algorithm. That is, $\hat{C}_i$ is the distance to the closest median. We claim that $\hat{C}_i \leq 2(1+\epsilon)C_i$. There are two cases:

- case 1: $i \in S$, $\hat{C}_i = 0$.
- case 2: $i \notin S$, then i was removed when some j was selected, therefore i is in the extended neighborhood of j. Since we can assign the demand at i to be supplied by j,

$$\hat{C}_i \leq c_{ij} \leq (1+\epsilon)C_i + (1+\epsilon)C_j \leq 2(1+\epsilon)C_i.$$

Note that $C = \sum_i \hat{C}_i$, we therefore prove the desired claim. ■

Claim 2 *The number of facilities opened is at most $(1+\frac{1}{\epsilon})k$.*

Proof: Given a point i, an observation is that i gets at least $\frac{\epsilon}{1+\epsilon}$ of its demand from within its neighborhood V_i in the LP solution. If not, cost for j in LP solution will be greater than

$$(1-\frac{\epsilon}{1+\epsilon})\cdot(1+\epsilon)C_i = \frac{1}{1+\epsilon}\cdot(1+\epsilon)C_i = C_i.$$

This is a contradiction. Thus, for all $i \in S$,

$$\sum_{j\in V_i} y_j \quad \geq \quad \frac{\epsilon}{1+\epsilon}. \tag{1}$$

Since the neighborhood of the facility selected do not overlap, we have

$$\sum_{i\in S}\sum_{j\in V_i} y_j \leq k.$$

It follows from (1) that the number of medians in S is at most

$$\frac{k}{\frac{\epsilon}{1+\epsilon}} = k(1+\frac{1}{\epsilon}).$$

■

References

[LV92] J.-H. Lin and J. S. Vitter. Approximation algorithms for geometric median problems. *Information Processing Letters*, 44:245–249, 1992.