CMPUT 675: Approximation Algorithms **Fall 2014**

Lecture 5 (Sept 12): The k-Centre Problem

Lecturer: Zachary Friggstad *Scribe: Leah Hackman*

5.1 The k-Centre Problem

In these notes, we look at the k-Centre problem, introduce a 2-approximation algorithm for this problem, and prove that if we can find a c-approximation approximation for this problem with $c < 2$, then P = NP. Finally, we introduce the Set Cover problem and present an approximation algorithm for this problem. The analysis of this algorithm will appear next class.

We begin by defining the k-Centre problem.

Definition 1 *Given a metric graph $G = (V, E)$ with distances $d(i, j)$ and integer k such that $1 \leq k \leq |V|$, find a set $F \subseteq V$ with $|F| = k$ which minimizes $\max_{j \in V} d(j, F)$.*

Here we let $d(j, S) = \min_{i \in S} d(i, j)$ for any $j \in V$ and any nonempty $S \subseteq V$.

For example, suppose all our vertices are client locations and we would like to select k client sites to build facilities to service all our clients. To make our clients happy, we do not want them to have to drive very far to get to our facility, so we would like to minimize the maximum amount of driving any client has to do. Each client is expected to drive to their nearest facility (for each client j, they drive to i such that $d(i, j) = d(j, F)$). Given clients select their facility this way, then the furthest any client has to drive is $\max_{j \in V} d(j, F)$. If we minimize this, we ensure no client is driving unnecessarily far.

We now present a greedy algorithm for approximating the k-Centre problem.

Algorithm 1 A greedy 2-approximation for the k-Centre problem

Input: A metric graph $G = (V, E)$ with distances $d(i, j)$, and an integer k such that $1 \leq k \leq |V|$
Output: A set $F \subseteq V$ with $|F| = k$

```
F ← {i} for an arbitrary i ∈ V
while |F| < k do
    i ← farthest location from F (select i such that d(i, F) = max_{i' ∈ V} d(i', F))
    F ← F ∪ {i}
end while
return F
```

Theorem 1 *Algorithm 1 is a 2-approximation for the k-Centre problem.*

Proof.

Let F^* be an optimum solution with cost OPT.

For $i \in F^*$, let $C(i)$ be the locations which "go to i" (more formally: $C(i) = \{j \in V : d(i, j) = d(j, F^*)\}$).

Lemma 1 *For each $i \in F^*$ and any two $j, \bar{j} \in C(i), d(j, \bar{j}) \leq 2 \cdot OPT$.*

Proof. First note $d(j, \bar{j}) \leq d(j, i) + d(i, \bar{j})$ by the triangle inequality. Because $j, \bar{j} \in C(i)$, then $d(j, i) + d(i, \bar{j}) \leq 2 \cdot OPT$, which gives us $d(j, \bar{j}) \leq 2 \cdot OPT$ ■

We break the analysis of Algorithm 1 into two cases:

Case 1: $\forall i \in F^*$, $F \cap C(i) \neq \emptyset$

That is, each $i \in F$ is taken from a different facility/client clusters in the optimal solution F^*.

Consider any $j \in V$ and say $i \in F^*$ is such that $j \in C(i)$. In this case, we also have some $i' \in F \cap C(i)$. We know that $d(j, F) \leq d(j, i')$ since $i' \in F$. And we know that $d(j, i') \leq 2 \cdot OPT$ from Lemma 1 since $j, i' \in C(i)$. Thus $d(j, F) \leq d(j, i') \leq 2 \cdot OPT$. This proves that if we are in Case 1, the cost to all client vertices is less that $2 \cdot OPT$.

Case 2: $\exists i \in F^*$ such that $F \cap C(i) = \emptyset$

Less formally, F contains none of the elements of one of the facility/client clusters in the optimal solution F^*.

By the pigeon hole principle, this means there must be an optimal cluster from which F has selected at least two facilities. Specifically: $|F \cap C(\bar{i})| \geq 2$ for some $\bar{i} \in F^*$.

Say $i_1, i_2 \in F \cap C(\bar{i})$ and, without loss of generality, the algorithm added i_1 to F before adding i_2. Let us also say that $\bar{F}$ is the set F in the algorithm just before i_2 is added.

We know that $\forall j \in V, d(j, F) \leq d(j, \bar{F})$ since $\bar{F} \subseteq F$. We also know that $d(j, \bar{F}) \leq d(i_2, \bar{F})$ since the algorithm chose to add i_2 to $\bar{F}$.

Since $i_1 \in \bar{F}$, we know that $d(i_2, \bar{F}) \leq d(i_1, i_2)$. Finally, because $i_1, i_2 \in C(\bar{i})$, from Lemma 1 we know that $d(i_1, i_2) \leq 2 \cdot OPT$.

Chaining these inequalities together shows

$$d(j, F) \leq d(j, \bar{F}) \leq d(i_2, \bar{F}) \leq d(i_1, i_2) \leq 2 \cdot OPT.$$

Thus in both cases, our solution has cost at most $2 \cdot OPT$. ■

Theorem 2 *If there is a c-approximation for the k-Centre problem for some $c < 2$, then $\mathrm{P} = \mathrm{NP}$.*

Proof. Given a graph $G = (V, E)$ with no isolated vertices, and integer k such that $1 \leq k \leq n$, deciding if there is a vertex cover of size k is NP-complete.

Define a metric over $V \cup V'$ where $V' = \{v_e, e \in E\}$ is a new set of vertices (one for each edge $e \in E$). The distances $d(u, v)$ in this metric are defined as follows. For every $u, v \in V$ with $(u, v) \in E$, set $d(u, v) = 1$. For every edge $e = (u, v) \in E$ set $d(u, v_e) = d(v, v_e) = 1$. Every other pair of nodes $(u, v) \in V \cup V'$ has $d(u, v) = 2$.

Lemma 2 *G has a vertex cover of size k if and only if the k-Centre instance has optimal value 1.*

Proof.

Proof for G has a vertex cover of size k $\implies$ the k-Centre optimal value is 1:

Let $S \subseteq V$ be a vertex cover of size k in G.

Consider any $v \in (V - S) \cup V'$:
Case 1: $v \in V - S$.
Since v is not an isolated vertex in G, there is some edge $(u, v) \in E$. Since S is a vertex cover in G and $v \notin S$, then $u \in S$. Then $d(v, S) = d(u, v) = 1$.
Case 2: $v \in V'$ (say $v = v_e$).
Since S covers e in $G, d(v, S) = 1$.

So we have $d(v, S) \leq 1$ for all $v \in V \cup V'$.

Proof for k-Centre optimal value is 1 $\implies$ G has a vertex cover of size k:

Let F be a k-Centre solution of cost 1. If $\exists v_e \in V' \cap F$, say $e = (u, w)$, then replace F with $(F - \{v_e\}) \cup \{u\}$. Notice that $|F| \leq k$ and that $d(v, F) = 1$ for all $v \in V \cup V'$. Repeat this until there are no more vertices from V' in F. For every $e \in E$, since $d(v_e, F) = 1$ then F must cover e in G. So, F is a vertex cover in G of size at most k. ■

By Lemma 2, we can see that if a $c < 2$ approximation for k-Centre existed then it could be used to distinguish between $OPT = 1$ and $OPT = 2$ cases in the above reduction. This would allow us to decide the Vertex Cover problem in polynomial time. ■

The first 2-approximation for k-Centre was presented in [G85] and the lower bound in Theorem 2 was proven in [HN79].

5.2 Set Cover Problem

Definition 2 *Given:*

- *set X of items*
- *a collection $\mathcal{S}$ of subsets of X*
- *costs $c(S) \geq 0, \forall S \in \mathcal{S}$*

Find the cheapest $\mathcal{C} \subseteq \mathcal{S}$ that covers X (i.e. $X = \bigcup_{S \in \mathcal{C}} S$)

Algorithm 2 A greedy approximation for the Set Cover Problem

Input: A set X of items, collection $\mathcal{S}$ of subsets of X, and costs $c(S) \geq 0, \forall S \in \mathcal{S}$.
Output: A collection $\mathcal{C} \in \mathcal{S}$ that covers X

```
C ← ∅ (sets we choose)
Y ← ∅ (items covered by C)
while Y ≠ X do
    select any S ∈ S which minimizes c(S)/|S−Y|
    C ← C ∪ {S}
    Y ← Y ∪ S
end while
return C
```

Theorem 3 *Algorithm 2 is an H_k-approximation of the Set Cover problem, where:*

- $k = \max_{S \in \mathcal{S}} |S|$
- $H_k = \sum_{d=1}^{k} \frac{1}{d} = \ln(k) + \mathcal{O}(1)$

Theorem 4 *Unless* P = NP*, there is no* $c \cdot \log n$*-approximation for* Set Cover *for any constant* $c < 1$*.*

Theorem 3 was first proven in [C79] and Theorem 4 in [DS14], though the same hardness lower bound under the stronger assumption that NP $\not\subseteq$ DTIME($2^{O(\log \log n)}$) was earlier proven in [F98]. The proof of Theorem 3 will be presented next class. We will not prove Theorem 4, but we will prove a slightly weaker lower bound near the end of the term.

References

C79 V. Chvátal, A greedy heuristic for the set-covering problem, *Mathematics of Operations Research*, 4:233–235, 1979.

DS14 I. Dinur and D. Steurer, An analytic approach to parallel repetition, *in proceedings of ACM Symposium on the Theory of Computing*, 2014.

F98 U. Feige, A threshold of $\ln n$ for set cover, *Journal of the ACM*, 45:634–652, 1998.

G85 T. F. Gonzales, Clustering to minimize the maximum intercluster distance, *Theoretical Computer Science*, 38:293–306, 1985.

HN79 W.-L. Hsu and G. L. Nemhauser, Easy and hard bottleneck problems, *Discrete Applied Mathematics*, 1:209–215, 1979.