Lecture 1: Greedy Approximation Algorithms

1 Set Cover

Given a universe U of n elements, and a collection of sets $\{S_1, \ldots, S_m\}$, each $S_i \subseteq U$ and having cost $c(S_i)$. The set cover problem is to pick a minimum cost collection of the sets which cover all elements.

Procedure Greedy-SC

X denotes the set of uncovered elements and $\mathcal{F}$ denotes the set cover picked by the algorithm.
Initialize $X \to U$ and $\mathcal{F} \to \emptyset$.
while X is not $\emptyset$ **do**
  Let S be the set which minimizes $\frac{c(S)}{|S \cap X|}$.
  $\mathcal{F} = \mathcal{F} \cup S$. $X = X \setminus S$.
end while

Analysis. Let $\mathcal{F} := \{S_1, \ldots, S_r\}$ be the sets cover picked by the algorithm, of total cost `alg`. Let $\{O_1, \ldots, O_\ell\}$ be the the optimal set cover of cost `opt`. Also, let us denote the set of uncovered elements just before iteration i to be X_i. Thus, $X_1 = U$ and $X_{r+1} = \emptyset$. Note that $S_i \cap X_i$, the set of elements covered at iteration i, is precisely $X_i \setminus X_{i+1}$.

Greedy choice tells us for all $i \in [r]$, we have

$$\forall j \in [\ell] : \quad \frac{c(S_i)}{|S_i \cap X_i|} \leq \frac{c(O_j)}{|O_j \cap X_i|} \tag{1}$$

$$\Rightarrow \quad \frac{c(S_i)}{|S_i \cap X_i|} \leq \frac{\sum_{j=1}^{\ell} c(O_j)}{\sum_{j=1}^{\ell} |O_j \cap X_i|} \leq \frac{\texttt{opt}}{|X_i|} \tag{2}$$

The last inequality follows since O_j's form a cover and thus, $\bigcup_{j=1}^{\ell} O_j \cap X_i = X_i$. Adding over all i we get

$$\begin{aligned}
\texttt{alg} = \sum_{i=1}^{r} c(S_i) &\leq \texttt{opt} \cdot \sum_{i=1}^{r} \frac{|S_i \cap X_i|}{|X_i|} \\
&\leq \texttt{opt} \cdot \sum_{i=1}^{r} \frac{|X_i| - |X_{i+1}|}{|X_i|} \\
&\leq \texttt{opt} \cdot \left(\frac{1}{|U|} + \frac{1}{|U| - 1} + \cdots + 1 \right) \\
&\leq \texttt{opt} \cdot H_n
\end{aligned}$$

Theorem 1. *Procedure* GREEDY-SC *is a H_n-approximation algorithm.*

Can we do a better analysis? We now show a slightly different way of analyzing giving us a better factor. Let $k := \max |S_i|$ be the size of the largest cardinality set in the collection. We argue now that the factor can be improved to H_k. To do this we introduce the "charging trick".

Once again let $\{S_1, \ldots, S_r\}$ be the sets picked by our algorithm. Recall that we pick set S_i in iteration i because it minimized $\alpha := \frac{c(S_i)}{|S_i \cap X_i|}$. For each element $j \in S_i \cap X_i$, that is, each new element covered by S_i, assign a charge $\alpha_j = \alpha$. Do this for every set picked. Observe the following things: each element gets charged once, and $\texttt{alg} = \sum_{j \in U} \alpha_j$.

Now pick a set O_i in the optimal set cover. Order the elements of O_i in the order in which they got covered by the algorithm. What do we know about α_j? When this element j was being covered by our algorithm, we had the choice of picking O_i. Furthermore, none of the elements $j, j+1, \ldots$ were covered. So, it must be that $\alpha_j \leq \frac{c(O_i)}{|O_i|-j+1}$. Thus, $\sum_{j \in O_i} \alpha_j \leq c(O_i) \cdot H_{|O_i|} \leq c(O_i) \cdot H_k$. Summing over all O_i's, we get

$$\texttt{alg} = \sum_{j \in U} \alpha_j \leq \sum_{i=1}^{\ell} \sum_{j \in O_i} \alpha_j \leq \texttt{opt} \cdot H_k.$$

Theorem 2. *Procedure* GREEDY-SC *is a H_k-approximation algorithm, where k is the cardinality of the maximum cardinality set.*

Consider now the vertex cover problem. This is a special case of set cover where $k = \Delta$, the max-degree. Thus, the greedy algorithm which picks the maximum degree vertex, deletes it, and iterates till all edges are covered is a H_Δ-approximation.

2 Metric Facility Location

In the facility location problem, we are given a set of facilities F, a set of clients C. Facility $i \in F$ has a cost f_i of opening. It costs $c(i,j)$ to connect client j to facility i. Clients can only be connected to open facilities. The objective is to find a set of facilities to open and connect clients to open facilities minimizing the total cost. This problem is normally called the *uncapacitated* facility location problem or simply UFL, so as to distinguish it from the capacitated facility location problem where each facility has a capacity which bounds the number of clients it can serve. This is a harder problem. If the connection costs form a metric, that is, $c(i,j) \leq c(i,j') + c(j',i') + c(i',j)$ for all $i, i' \in F$ and $j, j' \in C$, then the problem is called the metric UFL. We now give a constant factor approximation for the metric UFL problem.

Procedure Greedy-UFL

X denotes the set of facilities opened and D denotes the set of assigned clients. Each client in D will be assigned a facility in X, and we will maintain this assignment as $\sigma : D \to X$.
Initialize $X, D \to \emptyset$.
while D is not C **do**
    Given a facility i, let $D' \subseteq D$ be the set of clients who are closer to i than their currently assigned facility. Let $\delta(D, i) := \sum_{j \in D'} (c(\sigma(j), j) - c(i, j))$ denote the reduction in connection costs if i is opened.
    Pick a facility i and a set of unassigned clients $Y \subseteq C \setminus D$ so as to minimize

$$\frac{\mathbf{0}_{i \in X} \cdot f_i + \sum_{j \in Y} c(i, j) - \delta(D, i)}{|Y|}$$

    where $\mathbf{0}_{i \in X}$ is 0 if $i \in X$, 1 otherwise. {*Note if $i \in X$, then $\delta(D, i) = 0$.*}
    $X = X \cup i$. $D = D \cup Y$. Assign all $j \in Y \cup D'$ to i.
end while

Let's fix some notation. Let X^* be the set of facilities opened by the optimal algorithm. Let σ^* be the assignment of clients to X^* of the optimal solution. Given a client j, let $c_j^* := c(\sigma^*(j), j)$, and let $c_j := c(\sigma(j), j)$. We let $F^* = \sum_{i \in X^*} f_i$ and $C^* = \sum_{j \in C} c_j^*$. Note that $\texttt{opt} = F^* + C^*$. Similarly, let $F_{\texttt{alg}} = \sum_{i \in X} f_i$ and $C_{\texttt{alg}} = \sum_{j \in C} c_j$. $\texttt{alg} = F_{\texttt{alg}} + C_{\texttt{alg}}$. We introduce another bit of notation: Γ_i and Γ_i^* will respectively denote the set of clients assigned to facility i by our and the optimal algorithm.

We applying the charging idea. Whenever a client j is assigned to a facility for the *first time*, we let $\alpha_j := \frac{\mathbf{0}_{i \in X} \cdot f_i + \sum_{j \in Y} c(i,j) - \delta(D,i)}{|Y|}$, where Y is the set of clients being assigned for the first time in that iteration. Note that j could be re-assigned later on, but we do not modify α_j. Observe that, $\texttt{alg} = \sum_{j \in C} \alpha_j$.

Pick a facility i^* in X^* and let $k := |\Gamma_{i^*}^*|$. Order the clients in $\Gamma_{i^*}^*$ in the order they arrive in D. Consider the iteration at which the jth client is being added. A facility i is chosen along with a set of clients Y containing j. Let σ' be the assignment at the beginning of this iteration.

$$\forall \ell < j : \quad \alpha_j \quad \leq \quad c(\sigma'(\ell), \ell) + c_\ell^* + c_j^* \tag{3}$$

$$\alpha_j \quad \leq \quad \frac{f_{i^*} + c^*(\Gamma_{i^*}^*) - \sum_{\ell < j} c(\sigma'(\ell), \ell)}{(k - j + 1)} \tag{4}$$

When j is being added, one possible choice of the algorithm is to connect the singleton j to $\sigma'(\ell)$ for some $\ell < j$. Thus, $\alpha_j \leq c(\sigma'(\ell), j) \leq c(\sigma'(\ell), \ell) + c(i^*, \ell) + c(i^*, j)$, by metricity (finally used!). This implies (3) since both $j, \ell \in \Gamma_{i^*}^*$. Another possible choice of the algorithm is to add the facility i^* and the set $Y := \{\ell : \ell \geq j\}$. Furthermore, all the clients $\ell < j$, could be re-assigned to i^*. (Note that this might be suboptimal, but it is erring in the correct direction). So, $\alpha_j \leq \frac{f_{i^*} + \sum_{\ell \geq j} c(i^*, \ell) - \sum_{\ell < j} (c(\sigma'(\ell), \ell) - c(i^*, \ell))}{(k - j + 1)}$ which on rearrangement gives (4).

Adding (3) for all $\ell < j$, and (4) gives us $k\alpha_j \leq \sum_{\ell < j} c_\ell^* + (j-1)c_j^* + f_{i^*} + c^*(\Gamma_{i^*}^*)$. Adding over all $j \in \Gamma^*_{i^*}$ gives,

$$
\begin{aligned}
k\alpha(\Gamma^*_{i^*}) &\leq \sum_{j=1}^{k}\sum_{\ell<j} c^*_\ell + \sum_{j=1}^{k}(j-1)c^*_j + k(f_{i^*} + c^*(\Gamma^*_{i^*}) \\
&= kf_{i^*} + (2k-1)c^*(\Gamma^*_{i^*})
\end{aligned}
$$

Dividing by k and adding over all $i^* \in X^*$, we get $\texttt{alg} = \alpha(C) \leq \sum_{i^* \in X^*} \alpha(\Gamma^*_{i^*}) \leq F^* + 2C^*$.

Theorem 3. *Procedure* Greedy-UFL *is a 2-approximation.*

2.1 Improving the factor with greedy augmentation

This was not covered in the class. An algorithm is a (λ, μ) approximation to UFL if $\texttt{alg} \leq \lambda F^* + \mu C^*$. Note that the above is a $(1, 2)$ approximation. The following theorem shows how to "balance" the two out to get a better factor.

1. **Input:** Algorithm $\mathcal{A}$ which is a (λ, μ) approximation; parameter $\alpha \geq 1$.
2. Scale up all facility opening costs by a factor of α.
3. Run $\mathcal{A}$ to open a set of facilities X_0. σ_0 be the assignment of clients to nearest facility in X_0. Scale down facility costs back to original. Initialize X to X_0.
4. While there exists facility $i \in F \setminus X$ such that $f_i \leq \delta(C, i)$

   Add facility i which minimizes $\frac{f_i}{\delta(C,i)}$, to X.
5. Return X as the set of opened facilities. Connect every client to the nearest open facility.

Theorem 4. *The above algorithm is a* $(\lambda + \ln(\alpha), 1 + \frac{\mu-1}{\alpha})$ *approximation to UFL.*

Proof. Let F_0 and C_0 be the facility opening and connection costs of X_0 and σ_0, and let $F_{\texttt{alg}}$ and $C_{\texttt{alg}}$ be the same for X and σ. Note that

$$\alpha F_0 + C_0 \leq \lambda\alpha F^* + \mu C^* \tag{5}$$

Let the new facilities picked be $\{1, \ldots, t\}$, and let $X_i := X \cup \{1, \ldots, i\}$. Thus, $X = X_t$. σ_i be the assignment of clients to the nearest facility in X_i. Let C_i be the connection costs of σ_i.

Observe the following things. C_i's are decreasing and $C_{i-1} - C_i = \delta(C, i)$ for $1 \leq i \leq t$. Also, for each $1 \leq i \leq t$ and for each $i^* \in X^* \setminus X$, $\frac{f_i}{\delta(C,i)} \leq \frac{f_{i^*}}{\delta(C,i^*)}$. Thus,

$$\frac{f_i}{\delta(C,i)} \leq \frac{\sum_{i^* \in X^* \setminus X} f_{i^*}}{\sum_{i^* \in X^* \setminus X} \delta(C, i^*)} \leq \frac{F^*}{C_{i-1} - C^*}.$$

Let's do a technical gimmick here: since we pick facilities from 1 to t, only if the total cost of the algorithm decreases (since $f_i \leq \delta(C, i)$), the cost of our algorithm when we open $X = X_t$ is no more than the cost of the algorithm when we open X_ℓ, for $\ell \leq t$. Let ℓ be the smallest iteration at which $C_\ell \leq F^* + C^*$. That is, $C_{\ell-1} > F^* + C^*$. Henceforth we analyze the cost of the algorithm which opens only X_ℓ.

The above inequality gives us

$$\sum_{i=1}^{\ell} f_i \le F^* \sum_{i=1}^{\ell} \frac{C_{i-1} - C_i}{C_{i-1} - C^*}$$

The summation on the RHS is familiar – we saw it in our first analysis of set cover. Here's a slightly different way of bounding the expression on the right. First we break the summation in the RHS as follows:

$$\sum_{i=1}^{\ell} \frac{C_{i-1} - C_i}{C_{i-1} - C^*} = \sum_{i=1}^{\ell-1} \frac{C_{i-1} - C_i}{C_{i-1} - C^*} + \frac{C_{\ell-1} - (F^* + C^*)}{C_{\ell-1} - C^*} + \frac{(F^* + C^*) - C_\ell}{C_{\ell-1} - C^*}$$

Now the two summands on the above RHS can be upper bounded by the integration

$$\sum_{i=1}^{\ell-1} \frac{C_{i-1} - C_i}{C_{i-1} - C^*} + \frac{C_{\ell-1} - (F^* + C^*)}{C_{\ell-1} - C^*} \qquad \le \qquad \int_{F^*+C^*}^{C_0 - C^*} \frac{dx}{(x - C^*)}$$

Putting everything together and using $C_{\ell-1} > F^* + C^*$,

$$\sum_{i=1}^{\ell} f_i \le F^* \ln\left(\frac{C_0 - C^*}{F^*}\right) + (F^* + C^* - C_\ell)$$

Since $F_{\texttt{alg}} = F_0 + \sum_{i=1}^{\ell} f_i$ and $C_{\texttt{alg}} = C_\ell$, we get

$$\texttt{alg} \le F_0 + F^* \ln\left(\frac{C_0 - C^*}{F^*}\right) + (F^* + C^*) \tag{6}$$

Let $\beta = \frac{C_0 - C^*}{F^*}$. Using (5), we get that $F_0 = F^*\left(\lambda - \frac{\beta}{\alpha}\right) + C^*\left(\frac{\mu-1}{\alpha}\right)$. So,

$$\texttt{alg} \le F^*\left(\lambda + 1 + \ln(\beta) - \frac{\beta}{\alpha}\right) + C^*\left(1 + \frac{\mu - 1}{\alpha}\right)$$

The proof completes by noting the maximum value of the coefficient of F^* is obtained when $\beta = \alpha$. $\square$

Now using the $(1, 2)$ approximation described above, we get the following

Corollary 1. *There is a* 1.57*-approximation for metric UFL.*