Lecture 2: Greedy Algorithms

23rd January, 2015

1 Set Cover

The input to the set cover problem is a set system $(U, \mathcal{S})$ where U is a universe of n elements and $\mathcal{S}$ is a collection of m subsets of U. Each subset $S \in \mathcal{S}$ has a non-negative cost $c(S)$ associated with it. A set cover is a sub-collection $(S_1, \ldots, S_t)$ of $\mathcal{S}$ such that each element in U appears in **at least** one set S_i. The set cover problem is to find one with the minimum total cost.

The greedy algorithm is as follows.

- Initialize the set of **uncovered** elements $X = U$.
- While $X = \emptyset$:
  - Pick the set $S \in \mathcal{S}$ which **minimizes** $\frac{c(S)}{|S \cap X|}$.
  - $X = X \setminus S$.

A piece of notation: for any positive integer n define $H_n := 1 + 1/2 + 1/3 + \cdots + 1/n$. This is called the nth *harmonic number*. It is known that $\ln n \leq H_n < H_n + 1$. We now show that the above algorithm is a H_n-factor approximation algorithm. Let $S_1^*, \ldots, S_\ell^*$ be the optimal set cover. Let X_t be the set of uncovered elements jsut before the tth set was picked by the algorithm. We know that

$$c(S_t) \leq |S_t \cap X_t| \cdot \frac{c(S_i^*)}{|S_i^* \cap X_t|} \qquad \text{for all } 1 \leq i \leq \ell \text{ such that } S_i^* \cap X_t \neq \emptyset. \tag{1}$$

We now use the following fact: if $a_1, b_1, \ldots, a_\ell, b_\ell$ be a collection of 2ℓ positive reals. Then,

$$\min_{i=1,\cdots,\ell} \frac{a_i}{b_i} \leq \frac{\sum_{i=1}^{\ell} a_i}{\sum_{i=1}^{\ell} b_i}$$

We apply the above inequality to (1). We then make three observations: first, the sum of numerators is $\leq OPT$, second, the sum of the denominators is $\leq |X_t|$ since the S_i^*'s form a set cover, and lastly, just by definition, $|S_t \cap X_t| = |X_t| - |X_{t+1}|$. Together, we get

$$c(S_t) \leq OPT \cdot \frac{|X_t| - |X_{t+1}|}{|X_t|}$$

Now we add over all t to get $\sum_t c(S_t) \leq OPT \cdot H_n$. Why?

1.1 A Different Analysis

We now describe a different analysis of the algorithm. It will give a stronger result – it will prove that the above algorithm is an H_K-factor approximation algorithm where $K = \max_{S \in \mathcal{S}} |S|$, is the maximum size of a set in $\mathcal{S}$. This will follow via a "charging" argument; many weeks later we will come back to this charging in a slightly different context.

When the greedy algorithm picks a set S_t at time instant t, assign a charge $\alpha_j = c(S_t)/|S_t \cap X_t|$ to all elements j which are covered at this instant. That is, all elements in $S_t \cap X_t$. Recall X_t is the set of uncovered elements before S_t was picked. It should be clear that the cost of the sets picked by the greedy algorithm is precisely $\sum_{j \in U} \alpha_j$.

Now we upper bound this sum. Again, let $(S_1^*, \ldots, S_\ell^*)$ be the optimal set cover. Take any set S_i^* and *order* the elements in the order they are covered by the *greedy* algorithm breaking ties arbitrarily. Let $p = |S_i^*| \leq k$. Now consider an element j in the $q \leq p$th position in this order. We wish to upper bound α_j. Note that right before the instant t that the greedy algorithm covered j, there was exactly $(p-q+1)$ elements of S_i^* that were uncovered. In particular, $\alpha_j = c(S_t)/|S_t \cap X_t| \leq c(S_i^*)/(p-q+1)$. Thus, for every set S_i^* we have

$$\sum_{j \in S_i^*} \alpha_j \leq c(S_i^*) \sum_{q=1}^{p} \frac{1}{p-q+1} \leq c(S_i^*) H_K$$

Adding over all sets in the optimal set cover gives $\sum_{j \in U} \alpha_j \leq OPT \cdot H_K$.

1.2 Submodular Set Cover

A function f defined over subsets of a universe V is called submodular iff for all $A \subseteq B \subseteq V$ and $i \notin B$,

$$f(A \cup i) - f(A) \geq f(B \cup i) - f(B)$$

A set function is monotone if $f(A) \leq f(B)$ whenever $A \subseteq B$.

In the submodular set cover problem, we are given a universe V, oracle access to a *monotone* submodular function f, a cost function $c : V \to \mathbb{R}_{\geq 0}$, and a target parameter R. The goal is to find the minimum cost subset $W \subseteq V$ such that $f(W) \geq R$.

This problem generalizes set cover. But also captures many other problems. We looked at the following **source location** problem: given a directed graph G with all arcs having capacity say 1 unit, a sink t and a collection of possible sources $V = \{s_1, \ldots, s_k\}$ with costs $c_1, \ldots, c_k$. The goal is to find a minimum cost collection of sources S which together can send flow at least R units to the sink t.

The greedy algorithm for the submodular set cover is the following.

- Initialize $W = \emptyset$.
- While $f(W) < R$, pick $u \in V \setminus W$ which maximizes $\frac{c_u}{f(W \cup u) - f(W)}$.
  $W = W \cup u$.

In the exercises, you will be asked to analyze the above algorithm.

2 Greedy Maximization

Till now we have seen greedy algorithms for minimization problems. We now look at an algorithm for a maximization problem.

2.1 Constrained Submodular Maximization

Recall what a submodular function f is. We want to solve the following problem: given an integer k, find a set S with $|S| \leq k$ which maximizes $f(S)$. For this talk, we assume f is **monotone**, that is, for any $A \subseteq B$ we have $f(A) \leq f(B)$.

Here is the natural sounding greedy algorithm.

- Initialize $X = \emptyset$.
- Repeat k times: Pick element i which maximizes $f(X \cup i) - f(X)$. $X = X \cup i$.
- Return X.

Analysis. Let X_i be the set the algorithm maintains after step i. So the algorithm returns X_k. Let the optimal set be O. By the greedy property, we have

$$f(X_{i+1}) - f(X_i) \geq f(X_i \cup o) - f(X_i), \qquad \text{for all } i, \text{ for all } o \in O$$

If we average over all $o \in O$, we get

$$\begin{aligned} f(X_{i+1}) - f(X_i) &\geq \frac{1}{k} \sum_{o \in O} \left(f(X_i \cup o) - f(X_i) \right) & (2) \\ &\geq \frac{1}{k} \left(f(X_i \cup O) - f(X_i) \right) & (3) \\ &\geq \frac{1}{k} \left(f(OPT) - f(X_i) \right) \quad \text{because of monotonicity.} & (4) \end{aligned}$$

(3) is a result of submodularity. To see this, suppose $O = \{o_1, \ldots, o_k\}$ in any order. Because of submodularity, we have

$$f(X_i \cup o_j) - f(X_i) \geq f(X_i \cup \{o_1, \ldots, o_j\}) - f(X_i \cup \{o_1, \ldots, o_{j-1}\})$$

If we add the two sides for all j, the LHS is the RHS of (2) (without the scaling factor of k) while the RHS telescopes to the RHS of (3).

To rewrite (4), we get

$$f(X_{i+1}) \geq \frac{1}{k} f(OPT) + \left(1 - \frac{1}{k}\right) f(X_i)$$

A little bit of math shows that

$$\begin{aligned} f(X_k) &\geq f(OPT) \frac{1}{k} \left(1 + \left(1 - \frac{1}{k}\right) + \cdots + \left(1 - \frac{1}{k}\right)^{k-1} \right) + \left(1 - \frac{1}{k}\right)^k f(\emptyset) \\ &\geq f(OPT) \left(1 - (1 - 1/k)^k \right) \end{aligned}$$

since $f(\emptyset) \geq 0$. Using the fact that $(1 - 1/k)^k < 1/e$ where $e = 2.71...$, we get that the greedy algorithm is a $(1 - 1/e)$-factor appoximation algorithm.