Lecture Week 8

Enoch Lau

October 1, 2006

1 Set cover

1.1 Problem

Given:

- A set U of n elements
- A list $S_1, S_2, \ldots, S_m$, subsets of U
- Each set S_i has a weight w_i.

1.2 Definition

A **set cover** is a collection of sets whose union covers U.

1.3 Objective

Find a set cover C which minimises:

$$\sum_{S_i \in C} w_i$$

1.4 Greedy set cover algorithm

1. Start with $R = U$ and no sets selected (R is like the remainder).
2. While $R \neq \emptyset$:
   (a) Select set S_i which minimises:
   $$\frac{w_i}{|S_i \cap R|}$$
   (b) Define:
   $$c_i = \frac{w_i}{|S_i \cap R|}$$
   $\forall s \in S_i \cap R$.
   (c) Delete set S_i from R.
3. Return selected sets.

1.5 Some notation

Harmonic series:

$$\sum_{i=1}^{n} \frac{1}{i} = H(n) = \Omega(\log n)$$

1.6 Analysis

- If C is the set cover obtained by the greedy set cover algorithm, then:

$$\sum_{S_i \in C} w_i = \sum_{s \in U} c_s$$

- Claim: For every set S_k, $\sum_{s \in S_k} c_s$ is at most $H(|S_k|)w_k$.
- Proof:
  - Assume the elements of S_k are the first d elements of U: $S_k = \{s_1, s_2, \ldots, s_d\}$.
  - Assume that the elements are labelled in the order in which they are assigned c_{s_j} by the algorithm.
  - Consider the iteration in which s_j, $j \leq d$ is picked up.
  - At the start of the iteration, $s_j, s_{j+1}, \ldots, s_d \in R$.
  - $|S_k \cap R| \geq d - j + 1$
  - If it picks S_i (greedy algorithm comes into play here):

$$c_{s_j} = \frac{w_i}{|S_i \cap R|} \leq \frac{w_k}{|S_k \cap R|} \leq \frac{w_k}{d - j + 1}$$

$$\sum_{s \in S_k} c_s = \sum_{j=1}^{d} c_{s_j} \leq \sum_{j=1}^{d} \frac{w_k}{d - j + 1} = \frac{w_k}{d} + \frac{w_k}{d-1} + \cdots + \frac{w_k}{1}$$

  - $w_k H(d) = w_k \left(\frac{1}{d} + \frac{1}{d-1} + \cdots + 1 \right)$
- Claim: The set cover C selected by the greedy algorithm has weight at most $H(d^*)w^*$, where w^* is the optimal weight.