CSC2420 - Fall 2010 - Lecture 5

Lecturer: Prof. Allan Borodin
Scribe: Gokul Sridharan

13-October-2010

In the previous lecture we developed an LP formulation for the f-frequency weighted set cover problem and further noted that the weighted vertex cover problem is a case of 2-frequency covering problem. We also considered a greedy algorithm for the set cover problem and claimed that it has an approximation ratio of H_d. This claim is proved in this class using the dual fitting analysis.

1 Weighted Set Cover Problem

In this section we formally state the set cover problem and its dual. We then discuss the primal-dual method in this setting.

Let U be the set of all elements $\{e_1,\ e_2,\ \dots, e_m\}$. Assume a collection of sets $C = \{S_1,\ S_2,\ \dots, S_n\}$. Let the weight associated with each set be $w_i = w(S_i)$ for all i and let $d = \max\ |S_i|$. We would like to find a sub-collection $C' \subseteq C$ such that $\bigcup_{i:S_i \in C'} S_i = U$, while minimizing $\sum_{i:S_i \in C'} w_i$. Let x_i denote the decision to include/not include the set S_i in C'. After relaxing the integer constraint on x_i, the primal problem in such s setting can be written as

$$\text{Minimize} \sum_{i=1}^{n} w_i x_i \tag{1}$$

$$\text{Subject to} \sum_{j:e_i \in S_j} x_j \geq 1 \text{ for } i = 1, 2, \dots, n \tag{2}$$

$$X_j \geq 0 \text{ for all } j. \tag{3}$$

Note that an explicit constraint restricting x_is to less than 1 is not necessary as it is inherently captured in equation (2). The dual to the above problem, with $y_1,\ y_2, \dots\ y_m$ representing the dual variables is given by

$$\text{Maximize} \sum_{i=1}^{m} y_i \tag{4}$$

$$\text{Subject to} \sum_{i:e_i \in S_j} y_i \leq w_j\ \forall S_j. \tag{5}$$

While the dual variables do not always have an intuitive meaning, in this problem it is useful to think of y_i's as the price paid to cover the element e_i. The constraint in equation (5) above essentially states that we wouldn't want to pay anything more than w_j.

Having formally stated the primal and the dual problem, we now introduce the primal-dual method to solve the optimization problem.

1.1 The Primal Dual Method

The primal-dual method to solve the above problem can be broken down to the following steps

1. Start with an infeasible primal solution and a feasible dual solution. In this case, we set all X_is to 0 and all y_is to 0.

2. Increase some unfrozen y_i until some constraint j gets tight. Freeze all y_i's involved in the constraint.

3. Set the primal variable x_j to 1, i.e., include S_j in the cover. (If multiple constraints get tight, do the same for all such constraints.)

4. Repeat step 2 until all primal constraints are met, i.e., all e_i's are covered.

At the end of this procedure, we will have

1. A set of y_i's that are dual feasible.

2. A set of x_i's that are primal feasible and integral.

3. Primal slackness is satisfied, i.e., $x_j \neq 0 \Rightarrow j^{th}$ dual variable is tight.

4. f-approx dual slackness is satisfied, i.e., $y_i \neq 0 \Rightarrow \sum_{j:e_i \in S_j} X_j \leq f \quad \forall i$.

This algorithm is conceptually simple and unlike LP rounding does not need the "heavy machinery" of solving an LP.

1.2 Dual Fitting Analysis for Greedy Algorithm

We now analyze the greedy algorithm that was used to solve the set cover problem in the previous lecture.

The basic algorithm can be informally stated as: while there exists some e_k that is not covered, pick S_j so as to minimize $w_j/|\hat{S}_j|$. Here $|\hat{S}_j|$ represents the number of elements in S_j that are yet to be covered. Let the price paid to cover the i^{th} element be $z_i = w_j/|\hat{S}_j|$, where e_i gets covered when S_j is included to the collection C' and $|\hat{S}_j|$ is the number of uncovered elements in S_j when it is included to the collection. Now, if we were to set the dual variable y_i to the value z_i/H_d, we claim that this is a feasible solution to the dual problem. This implies that the solution obtained at the end of the greedy algorithm obtains

The proof of the above claim is as follows. Let the set $S = \{e_1,\ e_2,\ \ldots, e_l\}$ with $l \leq d$. Assume that the elements in S are listed according to the order in which they have been covered. In case more two or more element have been covered in the same iteration, the tie is broken arbitrarily. We need to show that $\sum_{i=1}^{l} y_i \leq w(S)$.

Say e_i was covered in the k^{th} iteration. If e_i was covered by S, the cost would be at most $w(S)/(k-i+1)$. Since the greedy algorithm chooses the most cost effective set in each iteration, the price z_i paid by e_i is at most $w(S)/(k-i+1)$. Hence when we sum the price paid by each element in S, we have:

$$\begin{aligned}
\sum_{i=1}^{l} y_i &= \frac{1}{H_d} \sum_{i=1}^{l} z_i && (6)\\
&\leq \frac{1}{H_d} w(S) \left(\frac{1}{l} + \frac{1}{l-1} + \ldots + \frac{1}{l} \right) && (7)\\
&= \frac{H_l}{H_d} w(S) && (8)\\
&\leq w(S) && (9)
\end{aligned}$$

This proves that the H_d-approx. dual slackness condition is satisfied and validates the claim made earlier.

2 Weighted Set Packing Problem

Consider a collection of sets $C = \{S_1,\ S_2,\ \ldots, S_n\}$. Let $U = \{e_1,\ e_2,\ \ldots, e_m\}$ and $w_i = w(S_i)$. We would like to find a disjoint sub-collection C' such that for $S_i,\ S_j \in C' \Rightarrow S_i \bigcap S_j = \phi$. We would like to choose C' so as to maximize $\sum_{i:S_i \in C'} w_i$. When the size of the sets S_i is restricted, i.e., $|S_i| \leq k$ we call this the k-set packing problem.

The graph theoretic interpretation of the above problem is as follows. Let graph $G = (V, E)$ with $V = S_1,\ S_2,\ \ldots S_n$ and $E = \{(S_i, S_j) | S_i \bigcap S_j = \phi\}$. In any graph $G = (V, E)$, a set of vertices V' is said to be an independent set if $v_i, v_j \in V' \Rightarrow (v_i, v_j) \notin E$. The weighted maximum independent set (WMIS) problem is to find an independent set of maximum weight. The unweighted case is called the maximum independent set (MIS) problem. In arbitrary graphs MIS is hard to approximate within a factor of $n^{1-\epsilon}$ for any $\epsilon > 0$ (assuming ZPP$\neq$ NP), while it trivial to get an approximation factor of n.

Given an MIS problem, interpreting it as a set packing problem is also quite straightforward. The set of elements $U = e_1,\ e_2,\ \ldots, e_m$ consist of the edges of the graph $G = (V, E)$ and the collection of sets S_i for $1 \leq i \leq |V| (= n)$ is given by the adjacency list of the vertices. Note that in this case, $m \leq n^2$.

It is easy to see that a natural greedy algorithm obtains an approximation ratio of $\min(n, m)$ in the case of the weighted set packing problem.

We now take a closer look at the k-set packing problem and the graphs induced by it. In the graph induced by the set packing problem, the neighborhood of a set S_i is given by $N(S_i) = \{S_j | (S_i, S_j) \in E\}$ and since every set contains at most k elements, the neighborhood is vertex covered by at most k cliques. Such graphs are called locally-VCC_k (vertex clique cover) graphs. These are not the same as VCC_k graphs where the entire graph that can be vertex covered by at most k cliques. Locally-VCC_k graphs belong to a much broader class of graphs called $k+1$ clawfree graphs or locally-IS_k (independent set) graphs. A graph is said to be $k+1$ clawfree if for any vertex v in the graph, $N(v)$ has at most k independent vertices.

Such graphs occur in various other scenarios as well. Intersection graphs of unit discs can be shown to be 6-clawfree (i.e. locally-IS_5). Intersection graphs of axis parallel unit squares are locally-IS_4 graphs.

Greedy algorithms are relatively efficient on clawfree graphs. In fact, the natural greedy Algorithm is a k-approximation algorithm for WMIS problem on any $(k+1)$ clawfree (i.e. locally IS_k) graph. The proof is as follows.

Let C_{opt} represent the optimal set of vertices and let C_{gre} represent the the set of vertices obtained using the natural greedy algorithm. Let h be a mapping from C_{opt} to C_{gre}. For $\nu \in C_{opt}$, define $h(\nu)$ as follows:

$$h(\nu) = \arg \max_{v' \in C_{gre}:(v,v') \in E} w(v') \tag{10}$$

We assume that the ties are broken lexicographically. Now, the number of elements $v \in C_{opt}$ that can get mapped to the same $v' \in E$ is limited by the clawfree nature of the graph. In particular, it cannot happen that $k+1$ elements in C_{opt} get mapped to the same element in C_{gre} since this would either imply the existence of a $k+1$ claw or that the $k+1$ vertices in C_{opt} are not independent, both of which cannot be true. This proves that the natural greedy algorithm is a k-approximation algorithm for the WMIS problem.

A variant on the natural greedy algorithm is to sort the sets according to the ratio $w(S)/|S|$. This approach can also be shown to result in a k-approximation algorithm.

2.1 Another Greedy Algorithm for Weighted Set Packing Problem

For the set packing problem, the natural greedy algorithm has an approximation ratio of $\min(n, m)$ which in practice is a very poor approximation ratio. In order to improve upon this ratio, we consider a variant on the natural greedy algorithm. Consider a greedy algorithm where sets are sorted according to the ratio $w(S)/\sqrt{|S|}$. This variant of the algorithm can be shown to have an approximation ratio of $min(n, \sqrt{m})$. The proof is as follows.

Let $r_i = w(S_i)/\sqrt{S_i}$. The sum of weights of the sets chosen in the case of the greedy algorithm is given by

$$\begin{aligned} \sum_{i:S_i \in C_{gre}} w(S_i) &\geq \sqrt{\sum_{i:S_i \in C_{gre}} (w(S_i))^2} && (11) \\ &\geq \sqrt{\sum_{i:S_i \in C_{gre}} r_i^2 |S_i|} && (12) \end{aligned}$$

The sum of weights of the optimal choice of sets is given by

$$\begin{aligned} \sum_{i:S_i \in C_{opt}} w(S_i) &\leq \sqrt{(\sum_{i:S_i \in C_{opt}} r_i^2)} \sqrt{(\sum_{i:S_i \in C_{opt}} |S_i|)} && (13) \\ &\leq \sqrt{(\sum_{i:S_i \in C_{opt}} r_i^2)} \sqrt{m}, && (14) \end{aligned}$$

where the second inequality follows from the fact that all the sets in C_{opt} are independent and that there a total of m elements in U. It now suffices to show that

$$\sum_{i:S_i \in C_{opt}} r_i^2 \leq \sum_{i:S_i \in C_{gre}} r_i^2 |S_i| \tag{15}$$

Since $|S_i| \geq 1$, it suffices to show that

$$\sum_{i:S_i \in C_{opt}, S_i \notin C_{gre}} r_i^2 \leq \sum_{i:S_i \in C_{gre}, S_i \notin C_{opt}} r_i^2 |S_i| \tag{16}$$

To prove this, we argue as follows. Suppose S_i is a set in C_{opt} but not in C_{gre}, it means that when S_i was considered for inclusion in the greedy algorithm, there was already another set S_j of higher weight ($r_j \geq r_i$) in C_{gre} such that $S_i \bigcap S_j \neq \phi$. While a number of sets such as S_i can be associated with the same set S_j, this number is limited by the number of elements in S_j (Note that all sets in C_{opt} are independent). Hence, if we denote the colection of sets in $C_{opt} \backslash C_{gre}$ ($C_{opt} \bigcap C_{gre}^c$) that intersect with a set S_j in $C_{gre} \backslash C_{opt}$ as C_{opt-S_j}, we have

$$\sum_{i:S_i \in C_{opt-S_j}} r_i^2 \leq r_j^2 |S_j|, \tag{17}$$

and the required inequality follows immediately.

3 Interval graphs and the greedy algorithm

Another class of problems where the greedy algorithm is optimal or gives good approximation bounds are problems involving interval graphs. An interval graph is the intersection graph of a collection of intervals on the real line. Every interval in the collection is represented as a vertex and two vertices are joined by an edge if the intervals they represent overlap. While interval graphs in general are not IS_k for any k, unit interval graphs are easily seen to be be IS_2 (assuming we do not have duplicates of the same interval).

The natural greedy algorithm (sorting by non-increasing weight) does not obtain a constant approximation for any constant. However, there is a greedy algorithm that can optimally solve the unweighted MIS problem for interval graphs. In fact, this greedy algorithm can be extended to an optimal algorithm for a much larger class of graphs called chordal graphs. Chordal graphs, their properties and the reason why the greedy algorithm is optimal for such graphs are discussed in the next lecture.