COMPSCI 632: Approximation Algorithms September 20, 2017

Lecture #8

Lecturer: Debmalya Panigrahi *Scribe: Mark Nemecek*

1 Overview

In the previous lecture, we introduced *primal-dual methods* which make use of *weak duality* in order to use the structure of a linear program and its dual to both find an approximate solution and prove an approximation factor. We formulated the *vertex cover*, *set cover*, and *feedback vertex set* problems as LPs and analyzed the primal-dual solutions.

In this lecture, we continue our study of primal-dual methods by considering the *metric facility location* problem.

2 Metric Facility Location

The *metric facility location* problem is defined as follows:

Definition 1. *Metric Facility Location (MFL): given a set of clients C, a set of possible facility locations F, the cost for opening a facility at each location $\forall i \in F, f_i$, and a metric $d(\cdot,\cdot)$, find a set of facilities to open which minimizes the total cost to open the facilities plus the cost to service all clients, where the cost to service a client is the distance from the client to the closest open facility as given by d.*

2.1 LP Formulation

We can formulate this problem as an LP as follows:

$$\begin{array}{lll} \text{minimize} & \sum_{j\in C}\sum_{i\in F} d_{ij}x_{ij} + \sum_{i\in F} f_i y_i & \\ \text{subject to} & \sum_{i\in F} x_{ij} \geq 1 & \forall j \in C \\ & y_i - x_{ij} \geq 0 & \forall i \in F, j \in C \\ & x_{ij}, y_i \geq 0 & \forall i \in F, j \in C \end{array}$$

By associating a variable α_j with each constraint in the first set and β_{ij} with each constraint in the second set, we can then formulate the dual of this LP as:

$$\begin{array}{lll} \text{maximize} & \sum_{j\in C} \alpha_j & \\ \text{subject to} & \alpha_j \leq \beta_{ij} + d_{ij} & \forall i \in F, j \in C \\ & \sum_{j\in C} \beta_{ij} \leq f_i & \forall i \in F \\ & \alpha_j, \beta_{ij} \geq 0 & \forall i \in F, j \in C \end{array}$$

For ease of reference, We will call the first set of constraints ”client” constraints and the second set ”facility” constraints.

We are then reminded that weak duality tells us that when the primal objective is a minimization, the value of the primal objective function for any primal feasible solution is no less than the value of the dual objective function for any dual feasible solution. Since we do not know the value of the optimal solution, we can use this theorem and the constructed dual solution to bound the error, subject to some caveats:

1. Sometimes we must be careful about which dual variables we increase at a given step
2. Sometimes we need to do some post-processing

2.2 Primal-Dual Algorithm

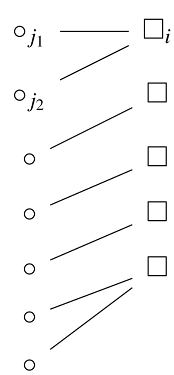
In this primal-dual method, we start with all α and β variables set to 0. We then raise the α variables uniformly, which we can visualize as an expanding ball around each client. Once an α_j crosses some d_{ij}, i.e., the ball around client j reaches facility i, we say that j is "putting weight on" i and note that a given client can put weight on multiple facilities and a facility can receive weight from multiple clients. As α_j has reached this threshold, we cannot increase it further without violating a client constraint.

Therefore, we start raising the corresponding β_{ij} while continuing to raise the α's uniformly, adding more β's to our set of increasing variables as more clients start putting weight on more facilities. However, once a facility constraint goes tight, we can no longer increase the β's for that facility and thus can no longer increase the α's for clients which are putting weight on the facility. In this case, we freeze these α's and add this facility, i, to the solution by setting the primal variable for i, y_i, to 1. We note that we do not need to specify the primal x variables as they are determined simply by which facility is closest to a client.

We must then consider when other α_j's can stop increasing - when α_j reaches a facility which already has a tight constraint, i.e., $\alpha_j = d_{ij}$ where $f_i = \sum_j \beta_{ij}$. In this case, however, why was α_j not already frozen? This is because the facility constraint became tight *before* α_j reached the facility, so it was not one of the variables frozen at that time (see the diagram below). We note that not all clients which put weight on a facility are necessarily growing, as they may have frozen due to reaching another facility. The algorithm stops when no duals can grow further without resulting in violated constraints.

2.2.1 Is this a good algorithm?

Here we make an argument for this being a good algorithm by noting that every client connection to the closest facility is at least as good as a connection to any facility. We consider the following diagram which shows the map of each client to its constraining facility:



Given this map, we claim that the following equations hold based on the dual constraints:

$$\begin{aligned} f_i &\geq \beta_{ij_1} + \beta_{ij_2} \\ \alpha_{j_1} &= \beta_{ij_1} + d_{ij_1} \\ \alpha_{j_2} &= \beta_{ij_2} + d_{ij_2} \end{aligned}$$

And thus, the total contribution to the primal objective is equal to $\alpha_{j_1} + \alpha_{j_2}$.

2.2.2 Why is this incorrect?

While this might seem like a good argument, it is incorrect because a given client can put weight on other facilities which are not preventing its growth. The connection costs are fine, but the sum is less than the total dual cost. Each α_j contributes to many f_i's in an effect we call *fan out*.

2.2.3 Fixing the problem

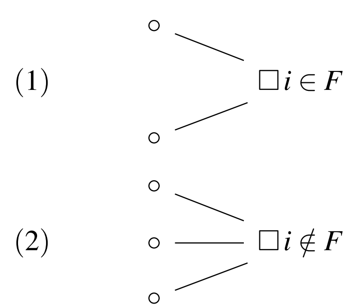
In order to fix the this problem, we want to try to bound the number of duals to which are contributed. If F is the set of facilities in our solution and C is the set of clients,

$$\sum_{i \in F} f_i + \sum_{j \in C} \min_{i \in F} d_{ij} \leq \sum_{i \in F} \sum_{j \in C} \beta_{ij} + \sum_{j \in C} \alpha_j \tag{1}$$

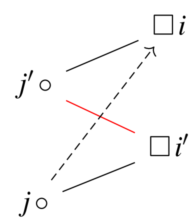
$$\leq \sum_{i \in F} \sum_{j \in C} \alpha_j + \sum_{j \in C} \alpha_j \tag{2}$$

Equation 1 follows as the cost for serving a client (the minimization term) is no more than the dual on that client because there must be a facility in the dual radius preventing growth. We then arrive at Equation 2, but note that that the double sum of α variables involves double counting some of them since we don't know how many α's contribute to a given facility.

In order to work around this, we introduce a post-processing step in which we arbitrarily pick a facility $i \in F$ and remove all of its 2-hop neighbors, i.e., those which are two steps away in a graph where an edge between a client and a facility indicate that the client is putting weight on the facility, from F. This results in two client types:



For case (1), we do not need to do anything more. For case (2), we map the clients served by i instead to the facility which caused the removal of i from F in our post-processing step. In the following diagram, j' was placing weight on both i and i', but was frozen by i (solid black line). Facility i was selected in post-processing and caused i' to be removed. Thus, j, which was frozen by i', is remapped to i (dashed line). We then repeat this process until we have dealt with all of the facilities.



Previously, we got stuck analyzing the facility costs, but we now claim that after this post-processing, it is easy to analyze the facility costs, but difficult to analyze the connection costs. We are no longer double counting because a client cannot be putting weight on multiple facilities. Where $F_j = \{i \in F \mid \beta_{ij} > 0\}$,

$$\sum_{i \in F} f_i = \sum_{i \in F} \sum_{j \in C} \beta_{ij} \tag{3}$$
$$= \sum_{j \in C} |F_j| \beta_{ij} \tag{4}$$
$$\leq \sum_{j \in C} \beta_{ij} \tag{5}$$
$$\leq \sum_{j \in C} \alpha_j \tag{6}$$

And (4) follows because there can be only one facility per client, which means $|F_j| \leq 1$, giving us (5). For client type (1), we can still bound the service cost with α_j as nothing has changed for such clients. However, for client type (2), the $\min_{i \in F} d_{ij}$ is not bound by α_j because j was not putting weight on i'. By applying the triangle inequality multiple times, we can determine that

$$d_{ij} \leq d_{ij'} + d_{i'j'} + d_{i'j} \tag{7}$$
$$\leq \alpha_{j'} + \alpha_{j'} + \alpha_j \tag{8}$$

(8) follows from (7) because j' was putting weight on both facilities and j was frozen by i'. We note that j' must have gotten tight at i earlier than j getting tight at i', so $\alpha_{j'} < \alpha_j$. Therefore, $d_{ij} \leq 3\alpha_j$.

All together, this gives us a 3-approximation because Type 1 clients pay 2 times the dual and Type 2 clients pay 3 times the dual.

Lemma 1. *The following holds true:* $3 \sum_{i \in F} f_i + \sum_{j \in C} \min_{i \in F} d_{ij} \leq 3 \sum_{j \in C} \alpha_j$ *where the factor of 3 on the first sum does not change the inequality because the cost of facilities is bounded by OPT, while the cost of the clients is the slack one.*

Proof. Let C_1 be the set of clients of Type 1. We can split the facility costs amongst the clients in C_1, so

$$\sum_{i \in F} f_i \leq \sum_{i \in F} \min_{j \in C_1} \beta_{ij} = \sum_{i \in F} \min_{j \in C} (\alpha_j - d_{ij}) \tag{9}$$

Thus it follows that

$$3 \sum_{i \in F} \min_{j \in C_1} \beta_{ij} = 3 \sum_{i \in F} \min_{j \in C} (\alpha_j - d_{ij}) \tag{10}$$

Which implies that

$$3\sum_{i\in F} f_i + \sum_{j\in C_1} \min_{i\in F} d_{ij} \leq 3\sum_{j\in C_1} \alpha_j - 2\sum d_i j \tag{11}$$

$$\leq 3\sum_{j\in C_1} \alpha_j \tag{12}$$

As $\sum_{j\in C_2} \min_{i\in F} d_{ij} \leq 3\sum_{j\in C_2} \alpha_j$, we can add this inequality with (12) to arrive at our claim. □

3 Lagrangian Approximation Preservers

If all of the facility costs f_i in the facility location problem were 0, then we could just open all facilities. If we are restricted to opening k facilities, then this problem is equivalent to k-medians. Recall the LP formulation for k-medians:

$$\begin{array}{lll} \text{minimize} & \sum_{i,j} d_{ij}x_{ij} & \\ \text{subject to} & y_i - x_{ij} \geq 0 & \forall i,j \\ & \sum_j x_{ij} \geq 1 & \forall i \\ & \sum_i y_i \leq k & \\ & x_{ij}, y_i \geq 0 & \forall i,j \end{array}$$

Where the constraint involving k is a new constraint added to restrict us to k medians. In what we call a *Lagrangian relaxation*, we can remove these constraints but penalize their violation in the objective function. The new primal and its dual are as follows:

$$\begin{array}{lll} \text{minimize} & \sum_{i,j} d_{ij}x_{ij} + \lambda(\sum_i y_i - k) & \\ \text{subject to} & y_i - x_{ij} \geq 0 & \forall i,j \\ & \sum_j x_{ij} \geq 1 & \forall i \\ & x_{ij}, y_i \geq 0 & \forall i,j \end{array}$$

$$\begin{array}{lll} \text{maximize} & \sum_j \alpha_j - \lambda k & \\ \text{subject to} & \alpha_j \leq \beta_{ij} + dij & \forall i,j \\ & \sum_j \beta_{ij} \leq \lambda & \forall i,j \\ & \alpha_j, \beta_{ij} \geq 0 & \forall i,j \end{array}$$

We note that if $\lambda = 0$, we would open all facilities and if we smoothly increased λ, we would open fewer and fewer facilities. If we chose a value for λ such that we happen to choose exactly k facilities, then this would be a feasible solution to k-medians, but how good would it be? Based on Lemma 1,

$$3\lambda k + \sum_j \min_i d_{ij} \leq 3\sum_j \alpha_j \tag{13}$$

$$\sum_j \min_i d_{ij} \leq 3\sum_j \alpha_j - 3\lambda k \tag{14}$$

$$= 3(\sum_j \alpha_j - \lambda k) \tag{15}$$

Therefore, the cost of our primal solution is no more than 3 times the cost of our dual solution, so we have a 3-approximation. If we use Lagrangian relaxation, we preserve the approximation factor - thus we call them Lagrangian approximation preservers.