

Lecture 23: Primal-dual algorithm for MST

Lecturer: *Sundar Vishwanathan*
COMPUTER SCIENCE & ENGINEERING

Scribe: *Keshri Nandan Nayak*
INDIAN INSTITUTE OF TECHNOLOGY, BOMBAY

1 Last lecture

In last class we derived relaxed LP formulation for the MST of a graph $G(V, E)$, the derived LP was:

$$\begin{aligned} \min \quad & \sum_e c_e x_e && \text{(ensures minimum cost)} \\ \text{s.t.} \quad & \text{for partition } \pi \\ & \sum_{e \text{ crosses } \pi} x_e \geq \#(\pi) - 1 && \text{(ensures graph is connected)} \\ & x_e \geq 0 \quad \forall e \in E \end{aligned}$$

where

- $c_e \geq 0$ is the cost of the edge $e \in E$
- π denotes a partition of the vertex set V and $\#(\pi)$ denotes the number of 'parts' in π .
- e crosses π means the end-points of e are in different parts of π .

Here the constraint x_e integer, $1 \geq x_e \geq 0$ was removed, and therefore LP was termed as relaxed. The dual of the above LP is given by

$$\begin{aligned} \max \quad & \sum_{\pi} y_{\pi} (\#(\pi) - 1) \\ \text{s.t.} \quad & \sum_{e \text{ crosses } \pi} y_{\pi} \leq c_e \quad \forall e \in E \\ & y_{\pi} \geq 0 \quad \forall \pi \end{aligned}$$

2 The primal-dual algorithm

As per the primal dual algo we start with any feasible dual solution. Thus the algorithm for finding MST of a graph is:

Initialize all $y_{\pi} := 0$

Set $i := 0, \pi_0 :=$ as the partition in which each vertex is a part by itself

Repeat:

Raise y_{π_i} until dual constraint corresponding to some edge e becomes tight i.e. $\sum y_{\pi} = c_e$.

Set $x_e := 1$

Set $\pi_{i+1} :=$ the partition got from π_i by merging end points of e

Set $i := i + 1$

Until: edges with $x_e = 1$ forms a spanning tree

In each iteration, we try and improve the dual solution. This we do while maintaining feasibility. In this case we will raise the value of the dual variables one by one.

Once a constraint is tight, for an edge we will not raise the dual variable for partitions which this edge crosses. Note that implicitly we are doing what problem 2 in Quiz 2 dictates.

After each iteration, we take the edges for which the constraints are equalities. If this set of edges yield a connected graph, we are done. A spanning tree will yield a primal feasible solution. Note the use of complementary slackness.

Here is a description of the algorithm.

We start with zero partition (i.e. an empty tree initially) and add edges as we go along until dual constraint corresponding to some edge becomes tight. We then include e into our solution by setting $x_e = 1$ and partition is updated by merging end points of e . The algorithm terminates with a spanning tree having edges with $x_e = 1$ of the input graph.

2.1 Proof of optimality:

We can prove this as optimum solution by proving that cost of primal and dual are same and they both are feasible.

Since cost of primal is given by:

$$\begin{aligned}
 & \sum_e c_e \quad \text{where } e \text{ is chosen by the algorithm} \\
 = & \sum_e \sum_{\substack{\pi \\ e \text{ crosses } \pi}} y_\pi \\
 = & \sum_\pi \sum_{\substack{e \\ e \text{ crosses } \pi}} y_\pi \\
 = & \sum_\pi y_\pi \sum_{\substack{e \\ e \text{ crosses } \pi}} 1 \\
 = & \sum_\pi y_\pi (\# \text{ of edges chosen which crosses } \pi) \\
 & \quad \quad \quad \downarrow \\
 = & \sum_\pi y_\pi (\#(\pi) - 1) \quad \text{Since when } y_\pi > 0 \text{ the } \# \text{ of edges chosen which} \\
 & \quad \quad \quad \text{crosses } \pi \text{ is exactly } (\#(\pi) - 1) \text{ for this algo.} \\
 = & \text{cost of dual}
 \end{aligned}$$

Thus there is a feasible primal and feasible dual and their cost are same. \Rightarrow Solution is optimum.

Note: In this LP, the constraint is exponential and so the dual has an exponential number of variables.

2.2 Exercises:

Run the primal-dual algorithm for following cases:

- π restricted to 2 parts and x_e are not integral, manipulate dual variables to get Kruskal's algo. We will get stuck in some point for this case while trying the above proof of optimality.
- $\forall n$, find an example where optimum(LP) < weight of MST.
- Add a new constraint $\sum_e X_e = n - 1$. And try the above exercise.