Lecture 6: Primal Dual Method in Linear Programming

Rajat Mittal

IIT Kanpur

The focus of this lecture note is to learn primal dual method to solve linear programming problems. We will take the example of max-flow/min-cut problem. In the beginning, we will see the definition of this problem and the formulation as a linear program. Later, we will introduce the idea of primal dual method. Finally, we will solve the max-flow/min-cut problem using primal dual approach.

1 Maximum flow problem

Look at a water distribution network in any country, say India. A water distribution network consists of water pipes of different capacities connecting different nodes. Here, nodes can be source of water, cities needing water or just some transfer nodes. In general, the problem is to transfer maximum amount of water to different cities using this network.

For the sake of simplicity, let us assume that there is only one source (of water) and only one sink (a city requiring water). The goal is to find out the maximum amount of water which can be sent from source to sink given the capacities of pipes.

The water sent from source to sink can be thought of as: amount of water sent between every pair of nodes (flow between the pair) with the following properties.

1. The amount of flow in every pipe is less than its capacity (capacity constraints).
2. At every node, the incoming flow is same as the outgoing flow, except source and sink (flow conservation constraints).
3. The outgoing flow at the source is same as the incoming flow at the sink, the amount of water transferred (the quantity which needs to be maximized).

The problem can be formulated in terms of a graph. We are given a directed graph $G(V, E)$ with:

- A node called source (s),
- A node called sink (t),
- Capacities $c_{u,v}$ for every edge $(u, v) \in E$ (edge represents a connection by a pipe).

The problem is to find the maximum flow possible in the graph G from s to t. A flow is a positive value $f_{u,v}$ for every edge $(u, v) \in E$.

Exercise 1. Convince yourself that the graph problem is same as the simple water distribution problem. What are the constraints on the flow?

Exercise 2. Can you think of an algorithm for this problem? Don't worry about the time taken by the algorithm. What if you are give the guarantee that the flow will only take integer values?

1.1 Linear program for max-flow

As you might have suspected, this problem can be converted into a linear program. Suppose, $f_{u,v}$ denotes the flow from vertex u to v. Taking into consideration the conditions for a valid flow, the linear program looks like:

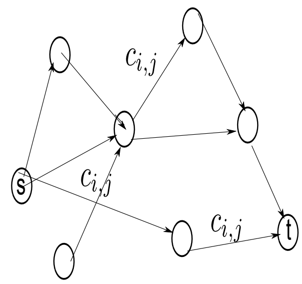

Fig. 1. Max flow problem: Capacities, source and sink.

$$
\begin{aligned}
& \max \sum_{\{s,u\}} f_{s,u} \\
\text{s.t.} \quad \sum_{u:\{u,v\}\in E} f_{u,v} = & \sum_{u:\{v,u\}\in E} f_{v,u} \;\forall v \neq s,t \quad \text{(flow conservation)} \\
& f_{u,v} \leq c_{u,v} \qquad \text{(capacity constraint)} \\
& f_{u,v} \geq 0
\end{aligned}
$$

Note 1. There is also a condition that the outgoing flow at source is same as incoming flow at the sink. But this condition is redundant given other flow conservation constraints.

Exercise 3. Convince yourself that the linear program above captures the max flow problem. Can you think of an algorithm now to solve max flow?

The flow conservation is present on every vertex except at source and the sink. This can be made uniform by putting an edge from t to s and it can be given ∞ capacity. If that does not make you feel comfortable, any upper bound on the maximum flow will work.

Exercise 4. Give an upper bound on maximum flow using capacities.

Then the linear program becomes,

$$
\begin{aligned}
& \max f_{t,s} \\
\text{s.t.} \;\forall v \quad \sum_{u:\{u,v\}\in E} f_{u,v} \leq & \sum_{u:\{v,u\}\in E} f_{v,u} \quad \text{(flow conservation)} \\
& f_{u,v} \leq c_{u,v} \quad \text{(capacity constraint)} \\
& f_{u,v} \geq 0
\end{aligned}
$$

Notice that we have exchanged equality in flow conservation with inequality. You will show in the assignment that this change does not affect the LP.

1.2 Dual for the linear program

Let's construct the dual of the above mentioned linear program for max flow. The dual will have variable $d_{u,v}$ for every edge and p_u for every vertex. Using these variables, the dual can be written as:

$$
\begin{aligned}
\min \sum_{u,v} c_{u,v} d_{u,v} \\
\text{s.t. } d_{u,v} \geq p_u - p_v \ \forall (u,v) \in E \\
p_s - p_t \geq 1 \\
d_{u,v} \geq 0 \ \forall (u,v) \in E \\
p_u \geq 0 \quad \forall u \in V
\end{aligned}
$$

Note 2. There is no constraint corresponding to the edge (t,s) (and no variable $d_{t,s}$). Also, the quantities p_u are translation invariant, i.e., all of them can have a constant subtracted or added if that keeps them positive.

2 Minimum cut problem

Let's look at the problem from other perspective. Suppose you need to cut the water supply from the source to the sink. The amount of time taken to cut a pipe is proportional to its capacity. How can you minimize your effort in cutting off the water supply.

Say the set of nodes still connected to s after cutting off some pipes is S. If the water supply is cut then there is no edge going from a vertex in S to a vertex in $\bar{S}$. Here $\bar{S}$ denotes the remaining vertices. In graph theoretic framework, this is called a cut between s and t (any subset S containing s). The size of the cut is the total capacity of edges going from S to $\bar{S}$. So, our problem is to find the minimum cut in the given graph.

It is interesting to note that a cut specifies the solution to the dual of max flow above. In other words, the capacity of any cut is an upper bound on the maximum flow possible between s and t. Actually this statement is easy to verify by looking at the cut picture.

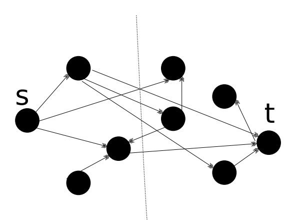

Fig. 2. Min cut problem: Capacities, source and sink.

How does the cut specify a dual solution? Take p_u to be 1 if $u \in S$ otherwise $p_u = 0$. The $d_{u,v}$ is 1 only for the edges in the cut and 0 otherwise.

Exercise 5. Show that the solution mentioned above is feasible for the dual linear program.

3 Max flow and min cut theorem

Let's take a look at the optimal solutions for the primal and dual formulation of max flow. Since $f_{u,v} = 0$ for all edges is a feasible solution for primal and also there is an upper bound on the maximum flow, the primal and dual both are feasible and the optimal solution exists.

Say $f^*_{u,v}$ and $d^*_{u,v}, p^*_u$ are the optimal solutions for the primal and dual. We can assume that s and t are connected and hence flow is positive. By complementary slackness,

1. $f^*_{u,v} > 0 \Rightarrow d^*_{u,v} = p^*_u - p^*_v$
2. $d^*_{u,v} > 0 \Rightarrow f^*_{u,v} = c_{u,v}$
3. $p^*_s - p^*_t = 1$
4. $p^*_v > 0 \Rightarrow \sum_{u:\{u,v\}\in E} f^*_{u,v} = \sum_{u:\{v,u\}\in E} f^*_{v,u}$

We know that the last condition is trivially satisfied, so let's worry about the first three. If we can show a dual solution which satisfies first three conditions then it is the optimal solution for dual (by complementary slackness). We will now create the solution for dual using the optimal solution $f^*_{u,v}$ with the same value.

Consider all the edges (u, v) where $f^*_{u,v} < c_{u,v}$ and all edges (v, u) where $f^*_{u,v} > 0$. Notice that the second set of edges are reversed. The graph with these edges is called the *residual graph* for the flow $f^*_{u,v}$. Since the flow is optimum, vertices s and t are not connected in the residual graph.

Suppose, S is the set of vertices connected to s in the residual graph. The dual solution corresponds to the cut $S, \bar{S}$. So, if $u \in S$ then $p_u = 1$ otherwise 0. The edges (u, v) which go from S to $\bar{S}$ are assigned $d_{u,v} = 1$ and others 0.

Exercise 6. Check that this dual solution satisfies the complementary slackness conditions given above. Most importantly check the first condition.

Hence for the optimal primal solution we get a dual solution which satisfies the complementary slackness conditions and hence is optimal. This implies that the value of min cut is not just an upper bound on the value of max flow but exactly equal. Let's write out the consequences.

- Max flow = min cut: In a graph, the value of max flow is equal to value of min cut.
- The dual LP for max flow is the LP for min cut.
- If the capacities are all integer then min cut is integer and hence max flow is integer too. Actually if capacities are integer then all flows in the graph are integer, this is called the *integrality theorem* in networks. We will not go through the proof here.

4 Overview of the primal dual approach

Now, we will look at one of the approaches to solve the linear program for maximum flow (that will solve the min cut problem too). The approach, called *primal dual approach*, is quite general and can be applied to any linear program.

This section will give a brief overview of this primal dual approach. In the next section, we will apply the approach to max flow problem. The general approach will assume that we have a linear programming solver (say simplex/interior method).

Suppose the LP's are given in the standard form. We will assume that $b \geq 0$.

$$\begin{aligned} &\max c^T x \\ \text{s.t.}\ \ &Ax - b \leq 0. \end{aligned}$$

$$\begin{aligned} &\min b^T y \\ \text{s.t.}\ \ &A^T y - c = 0 \\ &y \geq 0 \end{aligned}$$

The algorithm starts with a feasible primal solution x^* ($x = 0$ is feasible since $b \geq 0$). The idea is to use complementary slackness and the feasible primal solution to simplify the dual problem. The new dual problem is what we call the *restricted dual LP*.

Suppose, S is the index set of all the constraints in primal which are strictly followed by x^*. Then, by complementary slackness we know that only the corresponding variables in dual can be non-zero. The restricted dual LP and its dual (restricted primal) are,

$$\begin{aligned} \min \sum_i z_i \\ \text{s.t. } A_i^T y + z_i = c_i \quad \forall i \in [n] \\ y_j \geq 0 \quad \forall j \in S \\ y_j = 0 \quad \forall j \notin S \\ z_i \geq 0 \quad \forall i \in [n] \end{aligned}$$

$$\begin{aligned} \max c^T x \\ \text{s.t. } a_j^T x \leq 0 \quad \forall j \in S \\ x_i \leq 1 \quad \forall i \in [n] \end{aligned}$$

Here A_j are columns of A and a_i are rows of A. z_j's are called the *artificial variables*.

If x^* was optimal, the restricted dual will be feasible with optimal value 0. Otherwise, the value will be greater than 0 and the restricted primal solution will help us in finding a better primal solution.

Say x' is the restricted primal solution. You will show in the assignment that $x^* + \theta x'$ for a carefully chosen value of θ is feasible for primal with a better objective value.

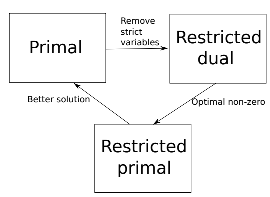

Fig. 3. Primal dual algorithm

Exercise 7. What conditions should θ satisfy? Why positive θ increases the objective value?

This process will continue till we get an optimal solution. The biggest question here is, we wanted to solve LP and now we need to solve restricted LP, what is the advantage? It turns out that in some cases restricted LP is much simpler to solve as compared to the original LP; the restricted LP can potentially be solved directly (without using LP solvers). We will see one such application in the next section.

5 Application to max flow

We discussed the LP for max flow before for graph $G(V, E)$. Let A be the edge incidence matrix ($|V| \times |E|$). So, a_{ie} is $1(-1)$ if i is the tail (head) respectively of the edge e, otherwise $a_{ie} = 0$. Suppose the first vertex in A is s and the second is t.

The LP for max flow is,

$$
\begin{aligned}
\max \quad & v \\
\text{s.t.} \quad Af + dv \leq 0 \quad d = (+1, -1, 0, \cdots, 0)^T \\
f \leq c \\
f \geq 0
\end{aligned}
$$

Here c is the capacity vector.

Exercise 8. Show that the LP defined at the start of this lecture is equivalent to this one.

Then the restricted primal is (with f_e^* as the primal feasible LP),

$$
\begin{aligned}
\max \quad & v \\
\text{s.t.} \quad Af + dv \leq 0 \quad d = (+1, -1, 0, \cdots, 0)^T \\
f_e \leq 0 \quad \forall e \in E: \; f_e^* = c_e \\
f_e \geq 0 \quad \forall e \in E: \; f_e^* = 0 \\
f \leq 1, v \leq 1
\end{aligned}
$$

Exercise 9. Show that this is the restricted primal.

An *undirected* path P from s to t in G, s.t.,

- If $(i, j) \in E$ is a forward edge in path then $f_{i,j} < c_{i,j}$,
- If $(i, j) \in E$ is a backward edge (travelled in reverse direction) in path then $f_{i,j} > 0$,

is called an *augmenting path*. Existence of augmenting path tells us that the value of the flow can be increased till some forward edge gets saturated or backward edge gets empty.

Exercise 10. Show that if there is an augmenting path then the value of the flow can be increased.

You will show in the assignment that the restricted primal LP above has value 1 if there is an augmenting path otherwise 0.

Every augmenting path is a solution to the LP is a simple exercise. For showing that the optimal solution of LP (if optimal value is greater than zero) can be converted into an augmenting path. Start with the least flow value and construct a path from s to t with that flow value. That will be the augmenting path and LP value will be 1.

So, if the restricted primal has optimal value 0 then complementary slackness conditions are satisfied and the flow is optimal. If not, then we can increase the flow using the augmenting path.

We will show below that we don't need to solve the LP above (through simplex or some other method) and directly find an augmenting path if flow is not optimal. The algorithm to find an augmenting path is known as the *labelling algorithm*.

5.1 Algorithm to find augmenting path

This algorithm runs on the undirected version of the given directed graph.

The idea of the algorithm is to label vertices starting from s. Every vertex $x \in V(G)$ is assigned label $(s(x), f(x))$. First entry tell us how to reach x (from which vertex) and second tell us how much flow can be put into it. If we reach t then we have an augmenting path.

Let us label s by (s, ∞). At every step, algorithm will pick a labelled vertex and try to label its neighbours (called scanning from that vertex). It can label a neighbour in these ways:

- Suppose x is labelled and $(x, y) \in E$ is an edge, s.t., $f_{x,y} < c_{x,y}$, then $s(y) = x$ and $f(y) = \min(s(x), c_{x,y} - f_{x,y})$.
- Suppose x is labelled and $(y, x) \in E$ is an edge, s.t., $f_{y,x} > 0$, then $s(y) = -x$ and $f(y) = \min(s(x), f_{y,x})$.

To make sure that we don't scan a vertex more than once, we will keep a list of labelled but unscaned vertices. It will initially contain only s. At every step, labelling algorithm picks an element from this list, scans it and deletes it from the list.

The algorithm can terminate in two ways, either we will label t or the list of labelled but unscanned vertices become empty.

Exercise 11. Show that this algorithm will terminate in finite steps.

If t is labelled then we have the augmenting path. In the other case, we can construct a cut with labelled vertices and unlabeled vertices. This cut satisfies the complementary slackness conditions with the given flow. Hence, both the flow as well as cut are optimal.

Exercise 12. Show that the cut obtained between labelled and unlabelled vertices satisfies complementary slackness conditions.

The complete algorithm to find max flow is known as Ford-Fulkerson algorithm. It terminates in finite iterations if all capacities are integer. It is known that in some cases (when capacities are irrational) the algorithm can run for infinite iterations.

6 Assignment

Exercise 13. Show that exchanging equality in flow conservation with inequality does not change the LP.

Exercise 14. Show that if the flow is optimal then s and t are not connected in the *residual graph*.

Exercise 15. Show that if x' is the solution for restricted primal (with x^* as the feasible primal), then $x^* + \theta x'$ will be feasible for primal with a better objective value (for a carefully chosen θ).

Exercise 16. Show that the restricted primal LP for max-flow has value 1 if there is an augmenting path otherwise 0.

Exercise 17. Write the Ford-Fulkerson algorithm in steps.