15-854: Approximations Algorithms **Lecturer:** R. Ravi
Topic: Primal-Dual Algorithms **Date:** Nov. 21, 2005
Scribe: Daniel Golovin

21.1 Primal-Dual Algorithms

So far, we have seen many algorithms based on linear program (LP) relaxations, typically involving rounding a given fractional LP solution to an integral solution of approximately the same objective value. In this lecture, we will look at another approach to LP relaxations, in which we will construct a feasible integral solution to the LP from scratch, using a related LP to guide our decisions. Our LP will be called the *Primal LP*, and the guiding LP will be called the *Dual LP*.

As we shall see, the PD method is quite powerful. Often, we can use the Primal-Dual (PD) method to obtain a good approximation algorithm, and then extract a good combinatorial algorithm from it. Conversely, sometimes we can use the PD method to prove good performance for combinatorial algorithms, simply by reinterpreting them as PD algorithms. So without further ado...

21.2 Every Primal has a Dual

We begin with a generic covering LP, and illustrate the ideas later with Vertex Cover as an example. Let $[k] := \{1, 2, \ldots, k\}$. Suppose we have matrix $A \in \mathbb{R}^{m \times n}$ and vectors $c \in \mathbb{R}^n, b \in \mathbb{R}^m$. We can represent the primal LP as

$$\begin{array}{ll} \min \sum_j c_j x_j & \text{subject to} \\ \sum_j a_{ij} x_j \geq b_i & \forall i \in [m] \\ x_j \geq 0 & \forall j \in [n] \end{array} \qquad (\textbf{Primal})$$

Now suppose we want to develop a lower bound on the optimal value of this LP. One way to do this is to find constraints that "look like" $\sum_j c_j x_j \geq Z$, for some Z, using the constraints in the LP. To do this, note that any convex combination of constraints from the LP is also a valid constraint. Therefore, if we have non-negative multipliers y_i on the constraints, we get a new constraint which is satisfied by all feasible solutions to the primal LP. That is, if for all i, $\sum_j a_{ij} x_j \geq b_i$, then

$$\sum_i y_i \left(\sum_j a_{ij} x_j \right) \geq \sum_i y_i b_i \qquad (21.2.1)$$

Note that we require the y_i's to be non-negative, because multiplying an inequality (in this case $\sum_j a_{ij} x_j \geq b_i$) by a negative number switches the sign of the inequality. (If a constraint has the form $\sum_j a_{ij} x_j = b_i$, then its multiplier y_i can be any real number.) Consider equation 21.2.1. If we ensure $\sum_i y_i \left(\sum_j a_{ij} x_j \right) \leq \sum_j c_j x_j$, we will obtain a lower bound of $\sum_i y_i b_i$ on the optimal value of the primal LP. Switching the order of summation, we get $\sum_i y_i \left(\sum_j a_{ij} x_j \right) = \sum_j \left(\sum_i y_i a_{ij} \right) x_j$, and can ensure this sum is at most $\sum_j c_j x_j$ by requiring the y_i's to satisfy

$$\sum_i y_i a_{ij} \le c_j \quad \forall j \in [n] \tag{21.2.2}$$

(Note that in the previous step we rely on the fact that the x_j's are non-negative.)

Putting it all together, if the y_i's are non-negative and satisfy constraint 21.2.2, then

$$\sum_i y_i b_i \quad \le \quad \sum_i y_i \left(\sum_j a_{ij} x_j \right) \quad = \quad \sum_j \left(\sum_i y_i a_{ij} \right) x_j \quad \le \quad \sum_j c_j x_j \tag{21.2.3}$$

Note that the constraints on the y_i's are linear, as is the lower bound we obtain. Thus we can write down an LP to find the y_i's giving the best lower bound. This is the *dual LP*.

$$\begin{array}{ll} \max \sum_i y_i b_i & \text{subject to} \\ \sum_i y_i a_{ij} \le c_j & \forall j \in [n] \\ y_i \ge 0 & \forall i \in [m] \end{array} \qquad (\textbf{Dual})$$

In more compact notation:

Primal	**Dual**
$\min c^\top x$ subject to $Ax \ge b$ $x \ge 0$	$\max y^\top b$ subject to $y^\top A \le c^\top$ $y \ge 0$

Fact 21.2.1 (LP Monogamy[1]) *Given a linear program P, if D is the dual LP of P, then P is the dual LP of D.*

Theorem 21.2.2 (Weak Duality) *For any feasible Primal-Dual solution pair (x, y), $y^\top b \le c^\top x$*

Proof: This follows immediately from our remarks above, specifically equation 21.2.3. ∎

Theorem 21.2.3 (Strong Duality) *If either the Primal or Dual have bounded optimal solution, then both of them do. Moreover, their objective values are equal. That is, if x is optimal for the primal, and y is optimal for the dual, then $y^\top b = c^\top x$.*

We won't prove the Strong Duality theorem. However, one way to prove it is to look at equation 21.2.3 and note that the inequalities must be tight if $y^\top b = c^\top x$. One can argue that if $y^\top b < c^\top x$, there must be some improvement we can make to either x or y.

Strong Duality gives additional constraints on optimal (x, y) pairs, namely the following equalities, which are derived by combining equation 21.2.3 with $y^\top b = c^\top x$.

$$\sum_i y_i b_i \quad = \quad \sum_i y_i \left(\sum_j a_{ij} x_j \right)$$

[1]Disclaimer: I invented this name. It's not standard terminology.

$$\sum_j \left(\sum_i y_i a_{ij} \right) x_j \quad = \quad \sum_j c_j x_j$$

Combining these equations with the constraints in the primal and dual LPs, we can derive the *complementary slackness* conditions.

Theorem 21.2.4 (Complementary Slackness) *Let (x, y) be solutions to a primal-dual pair of LPs with bounded optima. Then x and y are both optimal iff all of the following hold*

- $\forall j \in [n]$, *either* $x_j = 0$ *or* $\sum_i y_i a_{ij} = c_j$ *(Primal Complementary Slackness Conditions)*
- $\forall i \in [m]$, *either* $y_i = 0$ *or* $\sum_j a_{ij} x_j = b_j$ *(Dual Complementary Slackness Conditions)*

21.3 Constructing the Dual: An Example

By now we have covered everything you need to know to mechanically find the dual of any LP. However, here is some intuition to help with the process.

We start with a covering linear program, P, like the primal above. First, assign a variable y_i to each constraint in P (excluding the $x_j \geq 0$ constraints). Writing down the objective, $\max \sum_i y_i b_i$, is easy. The only tricky part is the dual constraints, $y^\top A \leq c^\top$. For now, let us fix a coordinate of c, say j, and figure out the constraint of the form $\sum (\ldots) \leq c_j$ in the dual. Note $y^\top A$ is an row vector whose j^{th} coordinate is the dot product of y and the j^{th} column of A, which we will denote by a^c_j. This column contains the coefficients for variable x_j. Thus we get the constraint $y^\top a^c_j \leq c_j$. Lastly, force the y_i's to be non-negative.

To make this concrete, consider the "natural" LP for Vertex Cover. Here, we are given an undirected graph $G = (V, E)$ whose edges are sets of vertices, each of size two. We associate a variable x_v with each vertex, and interpret $x_v = 1$ as including v in the solution.

$$\begin{array}{lll} \min \sum_v c_v x_v & \text{subject to} & \\ \sum_{v \in e} x_v \geq 1 & \forall e \in E & \qquad (\mathbf{VC-LP}) \\ x_v \geq 0 & \forall v \in V & \end{array}$$

We assign dual variable y_e to the constraint $\sum_{v \in e} x_v \geq 1$. Since $b_e = 1$ for all e, the dual objective is $\max \sum_e y_e \cdot 1$. Now consider the dual constraint corresponding to v, which is $y^\top \cdot a^c_v \leq c_v$. We can write this as $\sum_e y_e a_{e,v} \leq c_v$. Since $a_{e,v} = 1$ if e is incident on v, and $a_{e,v} = 0$ otherwise, we can write this constraint as $\sum_{e \in \delta(v)} y_e \leq c_v$. The final result is

$$\begin{array}{lll} \max \sum_e y_e & \text{subject to} & \\ \sum_{e \in \delta(v)} y_e \leq c_v & \forall v \in V & \qquad (\mathbf{VC-Dual-LP}) \\ y_e \geq 0 & \forall e \in E & \end{array}$$

21.4 Duality and Max-Min Relations

Before we get to primal-dual algorithms, observe that strong duality is useful as a min-max relation. In fact many min-max relations can be proven from it relatively easily. For example, Von Neumann's minimax theorem follows easily from it. The max-flow/min-cut theorem also falls right out of LP duality, if you realize that the natural max-flow LP is dual to the natural min-cut LP and the min-cut LP has integral basic feasible solutions.

We will prove another min-max relation using the Weak Duality Theorem and the dual LP for Vertex Cover given above. Suppose we are given an unweighted vertex cover instance, so that $c_v = 1$ for all v. Then integral solutions to VC-Dual-LP correspond exactly to matchings in the input graph. Thus, by weak duality, we conclude that the minimum vertex cover in an unweighted instance is at least the size of the maximum cardinality matching in the input graph. That is, in non-decreasing order of cost, we have the maximum cardinality matching (equal to VC-Dual-IP-OPT), VC-Dual-LP-OPT, VC-LP-OPT, and finally VC-IP-OPT (equal to the Vertex Cover OPT).

21.5 The Primal-Dual Method

Consider vertex cover. If we could bound the cost of some vertex cover S by $\rho \cdot \sum_e y_e$ for some dual feasible y, then we immediately obtain a ρ approximation by weak duality

$$\sum_{v \in S} c_v \quad \leq \quad \rho \cdot \sum_e y_e \quad \leq \quad \rho \cdot \text{VC-LP-OPT} \quad \leq \quad \rho \cdot \text{VC-IP-OPT}$$

So we consider dual variables as providing money, specifically $\sum_e y_e$ dollars, and allow the algorithm to spend up to $\rho \sum_e y_e$ dollars to buy a vertex cover. As a simple example, consider we have an unweighted vertex cover instance G. We find a maximum cardinality matching M, and set y to be the characteristic vector of M – that is, $y_e = 1$ if $e \in M$, and $y_e = 0$ otherwise. Note that y is dual feasible. Now, we use y to determine what vertices to buy for our vertex cover as follows: if v is incident on some vertex of M, buy it, otherwise do not. Let S be the output set of vertices. Then S is a vertex cover, since if it were not, some edge e could be added to M, contradicting the fact that it is of maximum cardinality. Now, each edge $e = \{u, v\} \in M$ has $y_e = 1$, so if we charge the cost of u and v to $e = \{u, v\}$, we spend $2y_e$ dollars. It follows that $\sum_{v \in S} c_v \leq 2 \cdot \sum_e y_e$, and we obtain a 2-approximation.

The Primal-Dual Schema We typically devise algorithms (for minimization problems) using the PD-schema in the following way:

1. Write down an LP relaxation of the problem, and find its dual. Try to find some intuitive meaning for the dual variables.
2. Start with vectors $x = 0, y = 0$, which will be dual feasible, but primal infeasible.
3. Until the primal is feasible,

(a) increase[2] the dual values y_i in some controlled fashion until some dual constraint(s) goes tight (i.e. until $\sum_i y_i a_{ij} = c_j$ for some j), while always maintaining the dual feasibility of y.

(b) Select some subset of the tight dual constraints, and increase the primal variable corresponding to them by an integral amount.

4. For the analysis, prove that the output pair of vectors (x, y) satisfies $c^\top x \leq \rho \cdot y^\top b$ for as small a value of ρ as possible. Keep this goal in mind when deciding how to raise the dual and primal variables.

Sometimes the output x buys too much, and we can decrease some of its coordinates[3] to get a cheaper (feasible) solution x'.

We will apply the primal-dual schema to give a 2-approximation for vertex cover with weighted vertex costs.

Primal-Dual Algorithm for Vertex Cover

Input: undirected graph $G = (V, E)$ and vertex costs c.
Initialize $x = 0$, $y = 0$
While $E \neq \emptyset$
  Select nonempty $E' \subseteq E$ arbitrarily.
  Raise y_e for each $e \in E'$ uniformly until some dual constraint goes tight.
  Let S be the set of vertices corresponding to dual constraints that just went tight.
  Set $x_v = 1$ for each $v \in S$, and delete all edges incident on vertices in S from E.
Output (x, y) and buy vertex set $A = \{v \mid x_v = 1\}$

Claim 21.5.1 *Let x, y be vectors output by the algorithm above. Then x is primal feasible, and y is dual feasible.*

Proof: Each edge deleted from E is incident on some vertex v such that $x_v = 1$. The algorithm only terminates when every edge has been deleted. Thus $\sum_{v \in e} x_v \geq 1$ for all $e \in E$, and x is feasible. As for y, no constraint is violated, since once a constraint goes tight, the edges in that constraint are deleted and thus their y_e values are not raised any further. ■

Claim 21.5.2 *Let x, y be vectors output by the algorithm above. Then $c^\top x \leq 2 \cdot (y^\top \cdot \vec{1})$.*

Proof: Let $A = \{v \mid x_v = 1\}$. Then

[2] Some sophisticated algorithms may sometimes decrease some dual variables, but that is for another day.
[3] See e.g. the reverse delete step in the algorithm of Agrawal, Klein, & Ravi for the Steiner Forest Problem.

$$
\begin{aligned}
c^\top x &= \sum_{v\in A} c_v \\
&= \sum_{v\in A}\left(\sum_{e\in\delta(v)} y_e\right) \\
&= \sum_{e\in E}\left(\sum_{v\in A\cap e} 1\right) y_e \\
&\le 2\cdot\sum_{e} y_e
\end{aligned}
$$

The second line follows from the fact that we set $x_v = 1$ only for vertices v corresponding to tight dual constraints. That is, $v \in A$ implies $\sum_{e\in\delta(v)} y_e = c_v$. The third line is simply switching the order of summation, and the last line follows from the fact that $|e| = 2$ for all e. ■

Using weak duality, we immediately obtain the following

Corollary 21.5.3 *The algorithm above is a 2-approximation for Weighted Vertex Cover.*

Observe that if we select E' to be a single edge, then we obtain the local-ratio algorithm for vertex cover. Note how much simpler the proof of the approximation gaurantee is in this case – no induction was needed. However, we could use an inductive proof to show that at all times, the cost we have paid is no more than twice the money we have "collected" from the dual, i.e. $c^\top x \le 2\cdot(y^\top \cdot \vec{1})$ at all times during the execution, not just at the end. This idea is useful when analyzing some primal-dual algorithms.

21.6 Further Reading

The primal dual algorithm for vertex cover presented above is due to Bar-Yehuda and Even [1]. For further reading, there is a book chapter [2] available online. For a more advanced (and more recent) treatment, a published survey on primal dual algorithms [3], is also available online.

References

[1] R. Bar-Yehuda and S. Even. A linear time approximation algorithm for the weighted vertex cover problem. *Journal of Algorithms*, 2:198–203, 1981.

[2] Michel X. Goemans and David P. Williamson. The primal-dual method for approximation algorithms and its application to network design problems. In Dorit Hochbaum, editor, *Approximation algorithms for NP-hard problems*, chapter 4, pages 144–191. PWS Publishing Co., Boston, MA, USA, 1997.

[3] D. Williamson. The primal-dual method for approximation algorithms. *Mathematical Programming*, Series B, 91(3):447–478, 2002.