CS880: Approximation and Online Algorithms	**Scribe:** Rojin Rezvan
Lecture 9: Dual Fitting, Primal Dual Algorithms	**Date:** 4 October

9.1 Properties of Primal-Dual Linear Programming

Primal		Dual	
min $c^T x$ s.t.		max $b^T y$ s.t.	
$Ax \geq b$	$\iff$	$y \geq 0$	m constraints
$x \geq 0$	$\iff$	$A^T y \geq 0$	n constraints

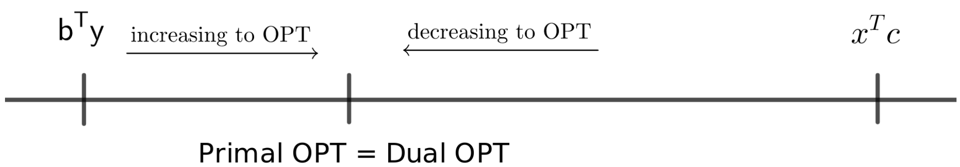

Figure 9.1.1: Primal-Dual Relationship

Theorem 9.1.1 ***Weak Duality***

For every feasible x and y:

$$c^T x \geq (A^T y)^T x = x^T A^T y = (Ax)^T y \geq b^T y$$

Theorem 9.1.2 ***Strong Duality***

For optimal solutions x^* and y^*:

$$x^{*T} c = b^T y^*$$

Theorem 9.1.3 ***Complementary Slackness***

x and y are optimal iff:

(a) $\forall i$, either $x_i = 0$ or $(A^T y)_i = c_i$.

(b) $\forall j$, either $y_j = 0$ or $(Ax)_j = b_j$.

Theorem 9.1.4 ***Approximate version of complementary slackness***

Given $\alpha, \beta > 1$:

(a) $\forall i$,, either $x_i = 0$ or $(A^T y)_i \geq \frac{1}{\beta} c_i$.

(b) $\forall j$, either $y_j = 0$ or $(Ax)_j \leq \alpha b_j$

Theorem 9.1.5 *For all feasible x and y which satisfy conditions of $\alpha-\beta$ complementary slackness, x and y are $(\alpha\beta)-$approximately optimal.*

Proof:

$$c^T x \leq \beta(A^T y)^T x = \beta(Ax)^T y \leq (\beta\alpha) b^T y$$

■

9.2 Set Cover

We write a primal-dual program for set cover problem: Consider i as indexes of sets and e as index of elements. Then we have:

Primal	Dual
$\min \Sigma_i c_i x_i$ s.t.	$\max \Sigma_e y_e$ s.t.
$\forall e \Sigma_{S_i \ni e} x_i \geq 1$	$\forall e y_e \geq 0$
$\forall i x_i \geq 0$	$\forall i \Sigma_{e \in S_i} y_e \leq c_i$
covering constraints	packing constraints

Fact 9.2.1 *The dual presents variables which their values represent how much should be paid for each edge to be covered.*

Consider the greedy algorithm below to solve set cover problem.[1]

Algorithm 1 Greedy Algorithm

Initialize $C \leftarrow \emptyset$
while $C \neq$ all elements **do**
  Let $i = \arg\min \frac{c_i}{|S_i \backslash C|}$
  Include S_i in solution
  For all $e \in S_i \backslash C$ set $y_e = \frac{c_i}{|S_i \backslash C|}$
  $C \leftarrow C \cup S_i$ [i.e. set $x_i = 1$]
end while

Claim 9.2.2 *x is feasible.*

Proof: The algorithm works until all elements are covered and sets 1 for the corresponding variables. So the condition in primal linear program will hold in the end. ■

Claim 9.2.3 $\Sigma_{i \in solution} c_i = \Sigma_i c_i x_i = \Sigma_e y_e$.

Proof: We only set x variables 1 which were included in the solution. This concludes the first equality.

Moreover, at each iteration which we found a new set to include in the solution, we set exactly $|S_i \backslash C|$ y variables as $\frac{c_i}{|S_i \backslash C|}$. This concludes the second equality. ■

Claim 9.2.4 *If n is the number of elements, $\forall i : \Sigma_{e \in S_i} y_e \leq H_n c_i$.*

Proof: Order elements in S_i in order of coverage. Consider j^{th} element.

For each element, cost of S_i when j was covered is less than or equal to $\frac{c_i}{|S_i| - j + 1}$. Therefore $y_i \leq \frac{c_i}{|S_i| - j + 1}$.

$$\Sigma_{e \in S_i} y_e = \Sigma_j \frac{c_i}{|S_i| - j + 1} = c_i \left[\frac{1}{|S_i|} + \frac{1}{|S_i| - 1} + \ldots + 1 \right] \leq c_i H_n \qquad \blacksquare$$

Implication 9.2.5 *$\hat{y} = \frac{y}{H_n}$ is a feasible dual solution and $\Sigma_e \hat{y}_e \geq \frac{1}{H_n} \Sigma_i c_i x_i$. This means we have a H_n approximation factor in which $H_n = \Sigma_{i=1}^{n} \frac{1}{i} = \theta(\log n)$.*

9.3 Primal Dual Algorithms

Basic Steps in a Primal-Dual Algorithm (in a minimization problem) is as follows:

Algorithm 2 Primal Dual Algorithm

1: **Initialize** Start with $x = 0, y = 0$. Always maintain dual feasibility.
2: **while** Primal is not feasible **do**
3: Raise some y's until some dual constraint goes tight.
4: Raise corresponding primal variables; freeze dual variables in the tight constraint. (Raise it enough to satisfy all constraints it is in.)
5: **end while**

Primal-Dual for set cover is as follows:

Algorithm 3 Primal Dual Algorithm for Set Cover Problem

1: **Initialize** Start with $x = 0, y = 0$.
2: Pick some uncovered e. Raise y_e until for some i, $\Sigma_{e \in S_i} y_e = c_i$.
3: Set $x_i = 1$. For all $e \in S_i$, freeze y_e.
4: If there exists any uncovered e, go to Step 2.

Claim 9.3.1 *x and y are feasible.*

Claim 9.3.2 *If $x_i > 0$ then $\Sigma_{e \in S_i} y_e = c_i$.*

Claim 9.3.3 *$\Sigma_{i:S \ni e} x_i \leq F$ where F is the maximum frequency. (i.e. $F = \max_{\mathbf{e}} |\{i \in S_i \ni e\}|$) This concludes a F-approximation factor for the problem.*

9.4 Feedback Vertex Set

Assume $G = (V, E)$ is a graph with costs c_v on vertices.

Goal: Remove the min cost set of vertices to make the graph acyclic.

This problem is NP-hard due to Karp. [2]

Primal	Dual
min $\Sigma_v c_v x_v$ s.t.	max $\Sigma_C y_C$ s.t.
$\forall C\ \Sigma_{v \in C} x_v \geq 1$	$\forall C\ y_C \geq 0$
$\forall v\ x_v \geq 0$	$\forall v\ \Sigma_{C: v \in C} y_C \leq \text{cost}(C)$

Here by C we mean any cycle in the graph. Primal-Dual algorithm for FVS is as follows:[3]

Algorithm 4 Primal Dual Algorithm for Set Cover Problem

1: **Initialize** Start with $x = 0, y = 0$.
2: Pick some uncovered cycle C with fewest vertices of degree greater than or equal to 3. Raise y_C until for some v, $\Sigma_{C \ni v} y_C = c_v$.
3: Set $x_v = 1$. For all $C \ni v$, freeze y_C.
4: If there exists any uncovered C, go to Step 2.

Claim 9.4.1 *x and y are feasible.*

Claim 9.4.2 *If for some v, $x_v > 0$, then $\Sigma_{C \ni v}\ y_C = c_x$.*

Claim 9.4.3 *In each iteration, we can find an uncovered edge with less than or equal to $2 \log n$ vertices of degree greater than or equal to 3.*

Claim 9.4.4 *For all cycles C in which $y_C > 0$,*

$$\Sigma_{v \in C} x_v \leq 4 \log n$$

Proof: Before C is covered, $\Sigma_{v \in C} x_v = 0$.

C contains less than or equal to $2 \log n$ vertices of degree greater than or equal to 3 and between them there at less than or equal to $2 \log n$ segments that are paths over degree 2 vertices. (one per segment)

Therefore, we pick at most $2 \log n + 2 \log n = 4 \log n$ vertices during this algorithm. ■

Corollary 9.4.5 *The above claim concludes a $4 \log n$-approximation factor for FVS.*

Fact 9.4.6 *We would not get a bounded result if we had chosen cycles arbitrarily during step 2 of primal-dual algorithm.*

Remark 9.4.7 *This analysis is nearly tight. The best approximation ratio for this problem is 2.*

References

[1] V. Chvatal. A greedy heuristic for the set-covering problem. *Mathematics of Operations Research*, 4(3):233–235, 1979.

[2] Richard M. Karp. *Reducibility among Combinatorial Problems*, pages 85–103. Springer US, Boston, MA, 1972.

[3] David P. Williamson and David B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, New York, NY, USA, 1st edition, 2011.