COL758: Advanced Algorithms	**Spring 2019**

Lecture 10: February 11

Lecturer: Naveen Garg — *Scribes: Ambreen Bashir*

Note: *LaTeX template courtesy of UC Berkeley EECS dept.*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications. They may be distributed outside this class only with the permission of the Instructor.*

10.1 Introduction

Given a graph G, each edge e has an associated resistance r_e. The energy of a $s-t$ flow f is given by $E(f) = \sum_e r_e f^2(e)$. The electrical flow of value F from source node s to a sink node t is the flow of value F from s to t that minimises the energy. The electrical flow can be computed in linear time.We can use this idea to compute maximum s-t flow in a graph with each edge having capacity C_e. In other words, our intent is to use electrical machinery to determine max flow. For this, we propose a **width - dependent** multiplicative update algorithm.

10.2 Simultaneous Minimization Problem

Given some convex functions $h_1, h_2, \ldots, h_m$, with each $h_i \colon Q \to \mathbb{R}^+$, over a convex set Q. We want to find a point in Q for which all the convex functions have a small value.

$$\forall i \quad h_i(x) \leq 1$$

To compute the point, we will use an oracle which on receiving a multiplier y_i will output a point x where the convex combination of all the functions will be atmost 1.

$$\frac{\sum_i y_i h_i(x)}{\sum_i y_i} \leq 1 \qquad \forall y_i \geq 0$$

For this point, different functions will have different values but the convex combination of all functions will be atmost 1.

10.3 Algorithm

We propose an iterative algorithm as follows:

In every round, r, we give an associated y_i^r which is initially taken as 1 to the oracle. Using y_i^r, oracle takes the convex combination of h_is and outputs a point x^r where the value of the convex combination of all functions will be small ($\sum_i \underline{y_i^r} h_i(x) \leq 1$). The individual functions can have higher values. For the functions having high values, we define $w^r = \frac{1}{\max_i h_i(x^r)}$ to ensure $w_r h_i(x)$

Algorithm 1 Width dependent multiplicative update algorithm

procedure WIDTHDEPENDENT($h_i(x)$)
  Initially, $y_i \leftarrow 1$
  while $\sum_r w^r \geq \frac{\ln m}{\epsilon^2}$ **do**
    Given y_i^r find $x^r \in Q$ for which $\sum_i \underline{y_i^r} h_i(x) \leq 1$
    Step size: $w_r = \frac{1}{\max_i h_i(x^r)} \implies w_r h_i(x) \leq 1 \quad \forall i$
    Update $y_i^{r+1} \leftarrow y_i^r e^{\epsilon w^r h_i x^r}$
  return $w^r, x^r \quad \forall r$ ▷ where w^r is the step size and x^r is the point picked at every step

is at most 1 for all functions. In the next step, y_i is updated for the next round. For the functions having high value, y_i is kept high and y_i is small for functions with low value. This is done inorder to lower the value of functions when next point is picked. But a small change $(1+\epsilon)$ is made to keep it controlled. This is achieved by $w_r h_i(x) \leq 1$. These steps are repeated for some r rounds until the desired value of sum of w^r is obtained which is discussed later.

The final solution $\overline{x}$ is defined as the convex combination of all the points taken at each round.

$$\overline{x} = \frac{\sum_r w^r x^r}{\sum_r w^r}$$

10.3.1 Analysis

Claim 1:

$$\sum_r w^r \sum_i \underline{y_i^r} h_i(x^r) \geq \frac{\max_i \sum r w^r h_i(x^r)}{e^\epsilon} - \frac{\ln m}{\epsilon e^\epsilon}$$

Proof: Change in the value of y_i at each step is given by:

$$\begin{aligned}
y_i^{r+1} - y_i^r &= (y_i^r e^{\epsilon w^r h_i(x^r)}) - y_i^r \\
&= y_i^r (e^{\epsilon w^r h_i(x^r)} - 1) \\
&\leq y_i^r \epsilon w^r h_i(x^r) e^{\epsilon w_r h_i(x^r)} && (e^x - 1 \leq x e^x \text{ —Taylors series}) \\
&\leq y_i^r \epsilon w^r h_i(x^r) e^\epsilon && (w^r h_i(x) \leq 1 \quad \text{always}) \\
&\leq \epsilon e^\epsilon w^r h_i(x^r) y_i^r
\end{aligned}$$

Thus summing over all the functions;

$$\sum_i y_i^{r+1} - \sum_i y_i^r \leq \epsilon e^\epsilon w^r \sum_i h_i(x^r) y_i^r$$

Summing over the rounds:

$$\sum_r \sum_i y_i^{r+1} - \sum_i y_i^r \leq \epsilon e^\epsilon \sum_r w^r \sum_i h_i(x^r) y_i^r$$

$$\begin{aligned}
\sum_r w^r \sum_i \underline{y_i^r} h_i(x^r) &\geq \frac{1}{\epsilon e^\epsilon} \sum_r \frac{\sum_i y_i^{r+1} - \sum_i y_i^r}{\sum_i y_i^r} && (\text{where } \underline{y_i} = \tfrac{y_i}{\sum_i y_i^r}) \\
&\geq \frac{1}{\epsilon e^\epsilon} \int_{\sum_i y_i^1}^{\sum_i y_i^N} \frac{1}{x} dx && (\text{where N is the total number of rounds}) \\
&\geq \frac{1}{\epsilon e^\epsilon} \ln \frac{\sum_i y_i^N}{\sum_i y_i^1} \\
&\geq \frac{1}{\epsilon e^\epsilon} (\ln \sum_i y_i^N - \ln \sum_i y_i^1) \\
&\geq \frac{\ln \sum_i y_i^N}{\epsilon e^\epsilon} - \frac{\ln m}{\epsilon e^\epsilon} \\
&\geq \frac{\ln \max_i y_i^N}{\epsilon e^\epsilon} - \frac{\ln m}{\epsilon e^\epsilon} && (y_i \text{ is the sum of exponentials}) \\
&\geq \frac{\ln \max_i e^{\epsilon w^r h_i(x^r)}}{\epsilon e^\epsilon} - \frac{\ln m}{\epsilon e^\epsilon} \\
&\geq \frac{\ln \max_i e^{\epsilon w^r h_i(x^r)}}{\epsilon e^\epsilon} - \frac{\ln m}{\epsilon e^\epsilon} \\
&\geq \max_i \frac{\epsilon w^r h_i(x^r)}{\epsilon e^\epsilon} - \frac{\ln m}{\epsilon e^\epsilon} \\
&\geq \max_i \frac{\sum_r w^r h_i(x^r)}{e^\epsilon} - \frac{\ln m}{\epsilon e^\epsilon} && (1)
\end{aligned}$$

Hence we have proved claim 1.

Stopping criteria:
Since value returned by the oracle will always be at most 1

$$\begin{aligned}
\sum_r w^r &\geq \sum_r w^r \sum_i \underline{y_i^r} h_i(x^r) \\
\sum_r w^r &\geq \max_i \frac{\sum_i r w^r h_i(x^r)}{e^\epsilon} - \frac{\ln m}{\epsilon e^\epsilon} && (\text{from } (1)) \\
e^\epsilon \sum_r w^r &\geq \max_i \sum_r w^r h_i(x^r) - \frac{\ln m}{\epsilon} \\
e^\epsilon &\geq \max_i \frac{\sum_r w^r h_i(x^r)}{\sum_r w^r} - \frac{\ln m}{\epsilon \sum_r w^r} \\
e^\epsilon &\geq \max_i h_i(\overline{x}) - \frac{\ln m}{\epsilon \sum_r w^r} && (2)
\end{aligned}$$

$\overline{x}$ is the point where the final solution is given. The value of the functions at point $\overline{x}$, $h_i(\overline{x})$ is less than any other $h_i(x)$

$$h_i(\overline{x}) = h_i\left(\frac{\sum_r w^r x^r}{\sum_r w^r}\right) \leq \sum_r \frac{w^r h_i(x^r)}{\sum_r w^r}$$

Thus,

$$\max_i h_i(\overline{x}) \leq (1+\epsilon) + \frac{\ln m}{\epsilon \sum_r w^r} \qquad \text{(from (2))}$$

The stopping criteria for the algorithm will be given by the fact that it will be run till the term $\frac{\ln m}{\epsilon \sum_r w^r} \leq \epsilon$. Therefore,

$$\sum_r w^r \geq \frac{\ln m}{\epsilon^2}$$

And

$$\max_i h_i(\overline{x}) \leq (1+\epsilon) + \epsilon = (1+2\epsilon)$$

Thus we have found a point $\overline{x}$ such that all the functions are less than 1+2ϵ ■

10.4 Width

The maximum value any function here can have value is at most ρ and not infinite. We try to keep the value of function $h_i(x)$ controlled. They should not have very high values but there should be a bound ρ which is the maximum value of any function at the point given by oracle. The ρ is known as **width** of the problem.

If $\max_r \max_i h_i(x^r) = \rho$ then,

$$w^r \geq \frac{1}{\rho} \qquad \forall r \qquad \text{(since } w^r = \frac{1}{\max_i h_i(x^r)}\text{)}$$

This implies that the number of rounds, N is atmost $\rho \frac{\ln m}{\epsilon^2}$ and the algorithm will get completed in N rounds.

If we have an oracle and a width ρ to control, then we can get (1+2ϵ) approximation algorithm. The running time will be atmost $\rho \frac{\ln m}{\epsilon^2}$. For low values of ρ, running time will be low.

10.5 Max Flow Algorithms

The algorithm discussed above can be used to compute approximate max flows in a graph. But there are two assumptions to be followed:

- Every edge has a capacity $C_e = 1$.
- The value of the flow, F is known

The domain Q contains the set of all possible valid flows i.e., set of all flows of value F from s to t. The flows obey the conservation constraint but violate the capacity constraint.

The oracle required will find a flow of value F where value of $\sum_e y_e h_e(x)$ is minimum for any given y_e. We need to find $x \in Q$ such that $\forall e, \quad h_e(x) \leq 1$ (i.e. they should meet the capacity constraints).