Spectral Graph Theory and Algorithms, Summer 2017	29 June 2017

Lecture 9

Lecturer: David P. Williamson

1 Multiplicative Weights for Packing Problems

Recall from last time our setup of the multiplicative weight update algorithm. We assume at each time step, there are N choices for a decision to be made. At time t, we will gain an unknown value $v_t(i) \in [0,1]$ for making decision i, and the values of all decisions will be revealed after the decision is made. In the multiplicative weights update algorithm (Algorithm 1), a weight is maintained for each decision i. If we let $w_t(i)$ be weight for decision i at start of time step t and

$$W_t = \sum_{i=1}^{N} w_t(i),$$

then decision i is chosen with probability proportional to $w_t(i)$, so the probability we make decision i at time t is $p_t(i) = w_t(i)/W_t$. After making the decision, the weights will be updated so as to be larger for decisions that get larger values. When the algorithm terminates, the expected value gained by the algorithm is

$$\sum_{t=1}^{T} \sum_{i=1}^{N} p_t(i) v_t(i).$$

Algorithm 1: Multiplicative Weights

$w_1(i) \leftarrow 1,\ \forall i = 1, \ldots, N$
for $t \leftarrow 1$ **to** T **do**
  Pick i with probability $p_t(i)$, get value $v_t(i)$
  $w_{t+1}(i) \leftarrow (1 + \epsilon v_t(i)) w_t(i),\ \forall i = 1, \ldots, N$

Recall that we proved the following.

Theorem 1 *Assume $\epsilon \leq 1/2$, then for any j,*

$$\sum_{t=1}^{T} \sum_{i=1}^{N} p_t(i) v_t(i) \geq (1-\epsilon) \sum_{t=1}^{T} v_t(j) - \frac{1}{\epsilon} \ln N.$$

[0]This lecture is based in part by a survey of Arora, Hazan, and Kale 2012 `http://theoryofcomputing.org/articles/v008a006/v008a006.pdf` from Christiano, Kelner, Madry, Spielman, and Teng 2010, `https://arxiv.org/pdf/1010.2921v2.pdf`, and also partly based on Lau's 2015 Lecture 15, `https://cs.uwaterloo.ca/~lapchi/cs798/notes/L15.pdf`. Scribes for these lectures were Yingjie Bi and Venus Lo.

Today we apply the multiplicative weights method to find feasible solutions to the system:

$$Ax \le e, \quad x \in Q. \tag{1}$$

Here $A \in \mathbb{R}^{m \times n}$, $e \in \mathbb{R}^m$ is the vector of all ones, and $Q \subseteq \mathbb{R}^n$ is a convex set. Assume $Ax \ge 0$ for each $x \in Q$.

We also assume that it is easy to optimize over Q, i.e., we have an *oracle* which can find $x \in Q$ such that $p^T Ax \le p^T e$ if such an x exists given any nonnegative vector $p \in \mathbb{R}^m$. If no such $x \in Q$ exists, then we can conclude that the system (1) is infeasible. Since $p^T Ax$ is a linear function in x, we have an oracle as long as we can optimize linear functions over Q.

The goal is to find approximate solution $x \in Q$ to (1) such that $Ax \le (1+\epsilon)x$, which can be solved by Algorithm 2 based on the multiplicative weights method. For convenience, define the *width* ρ of the oracle to be

$$\rho = \max_{i=1,\dots,m} \max_{\substack{x \in Q \\ \text{returned} \\ \text{by oracle}}} (Ax)(i).$$

The idea in Algorithm 2 is to run multiplicative weights algorithm in which each decision corresponds to a row of A and its value is $(Ax_t)(i)/\rho \in [0,1]$, where x_t is a vector returned by the oracle.

Algorithm 2: Finding Feasible Solution to (1)

$w_1(i) \leftarrow 1, \ \forall i = 1, \dots, m$
for $t \leftarrow 1$ **to** T **do**
  $W_t \leftarrow \sum_{i=1}^m w_t(i), \quad p_t(i) \leftarrow w_t(i)/W_t$
  Run oracle to find $x_t \in Q$ such that $p_t^T Ax_t \le p_t^T e$
  $v_t(i) \leftarrow (Ax_t)(i)/\rho$ (observe the value is in $[0,1]$ by the definition of width ρ)
  $w_{t+1}(i) \leftarrow (1 + \epsilon v_t(i)) w_t(i), \ \forall i = 1, \dots, m$
return $\bar{x} = \sum_{t=1}^T x_t / T$

The running time is $O(Tm)$ time plus $O(T)$ oracle calls and additional matrix-vector multiplications. The intuition in Algorithm 2 is to increase the weights most on the most violated inequalities, so in later iterations the oracle will work harder to find x_t satisfying these constraints.

The returned value $\bar{x}$ is always in Q by the convexity of Q. What remains to do is to bound $A\bar{x}$. Observe that

$$\sum_{i=1}^m p_t(i) v_t(i) = \frac{1}{\rho} p_t^T Ax_t \le \frac{1}{\rho} p_t^T e = \frac{1}{\rho}.$$

By Theorem 1, for any j,

$$\begin{aligned}
\frac{T}{\rho} \ge \sum_{t=1}^T \sum_{i=1}^N p_t(i) v_t(i) &\ge (1-\epsilon) \sum_{t=1}^T v_t(j) - \frac{1}{\epsilon} \ln m \\
&= (1-\epsilon) \sum_{t=1}^T \frac{1}{\rho} (Ax_t)(j) - \frac{1}{\epsilon} \ln m \\
&= (1-\epsilon) \frac{T}{\rho} (A\bar{x})(j) - \frac{1}{\epsilon} \ln m.
\end{aligned}$$

Hence

$$(1-\epsilon)\frac{T}{\rho}(A\bar{x})(j) \leq \frac{T}{\rho} + \frac{1}{\epsilon}\ln m.$$

Set $T = \rho \ln m/\epsilon^2$,

$$(A\bar{x})(j) \leq \frac{1}{1-\epsilon}\left(1 + \frac{\rho \ln m}{\epsilon T}\right) = \frac{1+\epsilon}{1-\epsilon} \leq 1 + 4\epsilon$$

for $\epsilon \leq 1/3$, which gives $A\bar{x} \leq (1+4\epsilon)e$. The running time is

$$O\left(\frac{m\rho}{\epsilon^2}\ln m\right) + O\left(\frac{\rho}{\epsilon^2}\ln m\right) \text{ oracle calls.}$$

2 Maximum flow in undirected graphs

We now show that we can apply electrical flows as the oracle in the feasibility problem solved by multiplicative weights above to a standard problem in combinatorial optimization known as the maximum flow problem. In this problem we are given as input an undirected graph $G = (V, E)$; it will be convenient to think of the undirected edges as being directed arcs, so we let $\vec{E}$ be an arbitrary orientation of edges in E. We also have as input a source vertex $s \in V$ and a sink vertex $t \in V$. Normally we also have as input a capacity for each edge in E, but for simplicity in presentation, we assume that the capacity of every edge in 1. A feasible flow f on $\vec{E}$ is one such that flow conservation is obeyed, namely,

$$\sum_{j:(i,j)\in\vec{E}} f(i,j) = \sum_{j:(j,i)\in\vec{E}} f(j,i) \text{ for all } i \neq s, t,$$

and capacity constraints are obeyed, namely,

$$-1 \leq f(i,j) \leq 1 \text{ for all } (i,j) \in \vec{E};$$

notice that we assume that if $f(i,j)$ is negative this means that $|f(i,j)|$ units of flow are going from j to i. The goal of the problem is to find a flow f that maximizes the net flow out of the source, namely

$$\sum_{j:(s,j)\in\vec{E}} f(s,j) - \sum_{j:(j,s)\in\vec{E}} f(j,s).$$

As an extension of what we discussed above, we can use a multiplicative weights algorithm to find approximately feasible solutions to the system $x \in Q, |Ax| \leq e$ for convex Q and $A \in \mathbb{R}^{m\times n}, x \in \mathbb{R}^n$. To do this, we require an oracle such that for any $p \in \mathbb{R}^m$, $p \geq 0$, the oracle is able to find $x \in Q$ such that $p^T|Ax| \leq p^T e$ or determine that no such x exists; note that if no such x exists, then there is no feasible solution to the system. We let

$$\rho = \max_i \max_{\substack{x\in Q \\ \text{ret. by oracle}}} |Ax|(i).$$

Following the proof above, we can show that $|A\bar{x}| \leq (1+4\epsilon)e$, $\bar{x} \in Q$, and that for $T = \frac{1}{\epsilon^2}\rho \ln m$, we get running times of $O(\frac{1}{\epsilon^2}m\rho \ln m) + O(\frac{1}{\epsilon^2}\rho \ln m)$ oracle calls.

3 Using Electrical Flows to Find a Max Flow

Let's start with a simple question: Suppose we can the multiplicative weights algorithm to determine if there is a feasible flow of value k. How can we use this to find the max flow? The solution would be to do a binary search on the possible flow values. In fact, since flows are integral and max flow = min cut, we only need to search on the integers in $[1, m]$.

The algorithm that follows determine if there exists a flow of value $k/\sqrt{1+\epsilon}$, and find such a flow if it exists. We can run it $O(\log m)$ times to get an approximate max flow.

3.1 Finding flow f_t

We will use an electrical flow computation as our oracle. We want to put commonalities between max flow and electrical flows into our description of Q, and the remainder into A. Specifically:

- The capacity constraint goes into A: Let $A = I_m$ ($m \times m$ identity matrix) to represent $|f(i,j)| \le 1$.
- The flow conservation constraints and flow value constraint go into Q, so that

$$
\begin{aligned}
Q = \{f \in \mathbb{R}^{|E|} : &\sum_{j:(i,j)\in\vec{E}} f(i,j) = \sum_{j:(j,i)\in\vec{E}} f(j,i) \text{ for all } i \ne s,t, \\
&\sum_{j:(s,j)\in\vec{E}} f(s,j) - \sum_{j:(j,s)\in\vec{E}} f(j,s) = k\}.
\end{aligned}
$$

Notice that we have one row in A per $(i,j) \in \vec{E}$, so weights w_t, values v_t, and probabilities p_t are all indexed by (i,j).

Idea: Use s-t electrical flows as an oracle to find a flow f_t with $b = k \cdot (e_s - e_t)$ (vectors with 1 in position of s, t respectively) and resistance $r_t(i,j) = w_t(i,j) + \frac{\epsilon}{m} W_t$ (Note in this discussion we are using t both to represent the sink vertex and a time step, but hopefully it is clear from context which is which). In order for electrical flows to work as an oracle, we have to be able to return an $x \in Q$ such that $p^T|Ax| \le p^T e$ (and certify that the system is infeasible if there is no such x) which in this context means we need to find f_t such that

$$
\sum_{(i,j)\in\vec{E}} p_t(i,j)|f_t(i,j)| \le \sum_{(i,j)\in\vec{E}} p_t(i,j) \tag{2}
$$

$$
\Leftrightarrow \sum_{(i,j)\in\vec{E}} w_t(i,j)|f_t(i,j)| \le \sum_{(i,j)\in\vec{E}} w_t(i,j) = W_t \tag{3}
$$

The second row is obtained by multiplying both sides by W_t.

Recall from our discussion of electrical flows that f_t minimizes total energy out of all feasible flows obeying the flow conservation constraints with supply vector b. Hence we can bound the energy of f_t based on the total weight. Let f_* be a flow of value k if one exists; if the max flow has value k, then such a flow exists. Note that

$|f_*(i,j)| \leq 1$ since it obeys the capacity constraints. We now observe that the energy of f_t must be at most the energy of f_* so that

$$\begin{aligned}
\mathcal{E}(f_t) &= \sum_{(i,j)\in\vec{E}} f_t(i,j)^2 \cdot r_t(i,j) \\
&\leq \sum_{(i,j)\in\vec{E}} f_*(i,j)^2 \cdot r_t(i,j) \\
&\leq \sum_{(i,j)\in\vec{E}} 1 \cdot r_t(i,j) \\
&= \sum_{(i,j)\in\vec{E}} \left(w_t(i,j) + \frac{\epsilon}{m} W_t\right) \\
&= (1+\epsilon) W_t
\end{aligned}$$

We will use this inequality again later; note again that it is conditional on the existence of the flow f_* of value k. Now we use this inequality to show that the electrical flow returned works for our oracle (see (3) above). We want to bound $\sum_{(i,j)\in\vec{E}} w_t(i,j) \cdot |f_t(i,j)|$ using Cauchy-Schwartz:

$$\begin{aligned}
\left(\sum_{(i,j)\in\vec{E}} w_t(i,j) \cdot |f_t(i,j)|\right)^2 &\leq \left(\sum_{(i,j)\in\vec{E}} f_t(i,j)^2 \cdot w_t(i,j)\right)\left(\sum_{(i,j)\in\vec{E}} w_t(i,j)\right) \\
&\leq \left(\sum_{(i,j)\in\vec{E}} f_t(i,j)^2 \cdot w_t(i,j)\right) W_t \\
&\leq \left(\sum_{(i,j)\in\vec{E}} f_t(i,j)^2 \cdot r_t(i,j)\right) W_t \\
&\leq (1+\epsilon) W_t^2
\end{aligned}$$

so that

$$\sum_{(i,j)\in\vec{E}} w_t(i,j) \cdot |f_t(i,j)| \leq \sqrt{(1+\epsilon)} W_t.$$

So to satisfy (3), we scale the returned flow f_t down by $\sqrt{1+\epsilon}$.

3.2 Bounding the Width of our Oracle: ρ

For our particular application, the width ρ is

$$\rho = \max_{(i,j)\in\vec{E}} |f_t(i,j)|$$

for f_t returned by the oracle. First we can bound the energy on an edge by the energy of the flow:

$$f_t(i,j)^2 r_t(i,j) \leq \mathcal{E}(f_t) \leq (1+\epsilon) W_t$$

We can also bound the resistance of each edge:

$$f_t(i,j)^2 \cdot r_t(i,j) \geq f_t(i,j)^2 \cdot \frac{\epsilon}{m} W_t$$

Combining the above gives us:

$$f_t(i,j)^2 \cdot \frac{\epsilon}{m} W_t \leq (1+\epsilon) W_t$$
$$|f_t(i,j)| \leq \sqrt{\frac{1+\epsilon}{\epsilon} m}$$

which implies that

$$\rho = O\left(\sqrt{\frac{m}{\epsilon}}\right).$$

3.3 Running time

Using the version of the multiplicative weight algorithm given above, we can check whether there exists a flow of value $\geq k/\sqrt{1+\epsilon}$ such that $|f(i,j)| \leq 1+4\epsilon$ in $O(\frac{1}{\epsilon^2}\rho \ln m)$ oracle calls. Substituting in ρ, we get that we can find an approximate flow in $O\left(\frac{1}{\epsilon^2}\sqrt{\frac{m}{\epsilon}} \ln m\right)$ oracle calls, which implies a running time of $\tilde{O}\left(\frac{m^{1.5}}{\epsilon^{2.5}}\right)$.

But why do we care about this when we can compute max flow on undirected unit-capacity graphs exactly in $O(m^{1.5})$ time (rather than approximately in this algorithm)? Can we go faster?

4 A Faster Algorithm: Deleting Edges

Since there can be flows such that $f_t(i,j) = \Theta(\sqrt{m})$, we need at least $\Omega(\sqrt{m})$ more oracle calls so that the average of the flow on (i,j) approximately obeys the capacity constraint; thus we cannot get a better running time than $\Omega(m^{1.5})$ this way. We want to decrease the upper-bound on $f_t(i,j)$ and ρ.

Idea: Here's an idea that rarely works, but does in this case: we just enforce that what we want to be true is true. Whenever the oracle finds an edge (i,j) of with $|f_t(i,j)| > \hat{\rho}$ for some $\hat{\rho}$, we will delete the edge for the remainder of the algorithm and recompute the flow; the oracle will do this until all edges have flow $\leq \hat{\rho}$. This forces the width of the oracle to be $\hat{\rho}$.

Let H be the set of all deleted edges. We will use $\hat{\rho} = \frac{4}{\epsilon} \cdot (m \ln m)^{1/3}$. In the next subsection, we will show that

- $|H| \leq (m \ln m)^{1/3}$
- $|H| \leq \frac{1}{8}\epsilon k$, so that the flow value decreases by factor of at most $\left(1 - \frac{1}{8}\epsilon\right)$ by deleting H.

Since there are at most $|H|$ extra flow computations, and we have a smaller oracle width, there are $O\left(\frac{1}{\epsilon^2} \cdot \hat{\rho} \ln m\right) + |H|$ oracle calls, or $O\left(\frac{1}{\epsilon^3} \cdot m^{1/3} (\ln m)^{4/3}\right)$ oracle calls, for a running time of $\tilde{O}\left(\frac{m^{4/3}}{\epsilon^3}\right)$.

4.1 Analysis of faster algorithm

Here are the things we will need to show to prove the result.

Claim 2 *Energy never decreases, and is always at most* $(1+\epsilon)W_{T+1}$ *where* $W_{T+1} \leq m \exp\left(\frac{1}{\epsilon}\ln m\right)$.

Claim 3 *The initial energy is at least* $(1/m)^2$.

Lemma 4 *The energy increases by at least a factor of* $\left(1 + \frac{\epsilon\hat{\rho}^2}{2m}\right)$ *for each deletion of an edge.*

Once we have these claims and the lemma, we see that

$$\left(1 + \frac{\epsilon\hat{\rho}^2}{2m}\right)^{|H|} \leq \frac{\text{final energy}}{\text{initial energy}} \leq \frac{(1+\epsilon)\, m \exp\left(\frac{1}{\epsilon}\ \ln m\right)}{\left(\frac{1}{m}\right)^2}.$$

Taking the log of both sides, we have

$$\begin{aligned}
|H| &\leq \frac{\ln(1+\epsilon) + 3\ln m + \frac{1}{\epsilon}\ln m}{\ln\left(1 + \frac{\epsilon\hat{\rho}^2}{2m}\right)} \\
&\leq \frac{\frac{2}{\epsilon}\ln m \cdot \left(1 + \frac{\epsilon\hat{\rho}^2}{2m}\right)}{\frac{\epsilon\hat{\rho}^2}{2m}} \qquad \text{using} \ln(1+x) \geq \frac{x}{1+x} \\
&\leq \frac{4m\ln m}{\epsilon^2\hat{\rho}^2} + \frac{2}{\epsilon}\ln m \\
&\leq \frac{6m\ln m}{\epsilon^2\hat{\rho}^2}
\end{aligned}$$

Thus for $\hat{\rho} = \frac{4}{\epsilon}(m\ln m)^{1/3}$, we have that $|H| \leq \frac{6}{16}(m\ln m)^{1/3}$. We also note that on any deleted edge (i,j), $\hat{\rho} < |f(i,j)| \leq k$, so we can multiply the last line by $k/\hat{\rho} \geq 1$ to get

$$|H| \leq \frac{6mk\ln m}{\epsilon^2\hat{\rho}^3} = \frac{6}{64}\epsilon k \leq \frac{1}{8}\epsilon k$$

We will now proceed to the proofs of the claims/lemmas in the outline.

Proof of Claim 2: The resistances sent by the multiplicative weight algorithm to the oracle only increase. By Rayleigh monotonicity (as shown in our problem set), this implies that energy does not decrease. Even when we remove an edge, this is equivalent to increasing the resistance to ∞.

To bound W_{T+1}:

$$\begin{aligned}
W_{T+1} &\leq m \cdot \exp\left(\epsilon \sum_{t=1}^{T}\sum_{i=1}^{n} v_t(i)p_t(i)\right) \\
&= m \cdot \exp\left(\epsilon\frac{T}{\hat{\rho}}\right) \qquad T = \frac{\hat{\rho}\ln m}{\epsilon^2} \\
&= m \cdot \exp\left(\frac{1}{\epsilon}\ln m\right).
\end{aligned}$$

$\square$

Proof of Claim 3: There must be some (i,j) such that $f(i,j) \geq 1/m$ on a flow of $k \geq 1$ units, and $r_1(i,j) > w_1(i,j) = 1$. Hence the energy is at least $f(i,j)^2 r_1(i,j) \geq 1/m^2$. $\square$

Proof of Lemma 4: Let $\mathcal{E}$ be the energy before deleting an edge $e = (i,j)$ and $\mathcal{E}'$ be the energy afterwards. Let p be the potentials associated with the initial energy $\mathcal{E}$. Then we have showed in a previous lecture that any potentials can be used to give a lower bound on the current energy $\mathcal{E}'$ with

$$\begin{aligned}
\mathcal{E}' &\geq 2b^T p - p^T L_{G-e} p \\
&= 2b^T p - p^T L_G p + \frac{(p(i) - p(j))^2}{r(i,j)} \\
&= \mathcal{E} + f(i,j)^2 \cdot r(i,j) \\
&\geq \mathcal{E} + \hat{\rho}^2 \left(\frac{\epsilon}{m} W_t \right) \\
&\geq \mathcal{E} + \hat{\rho}^2 \left(\frac{\epsilon}{m} \right) \left(\frac{1}{1+\epsilon} \cdot \mathcal{E} \right) \\
&= \left(1 + \frac{\epsilon \hat{\rho}^2}{(1+\epsilon)m} \right) \mathcal{E} \\
&\geq \left(1 + \frac{\epsilon \hat{\rho}^2}{2m} \right) \mathcal{E}.
\end{aligned}$$

The second inequality is true because we delete (i,j) when $|f(i,j)| > \hat{\rho}$; the third inequality is true because whenever there is a flow of value k, then $\mathcal{E} \leq (1+\epsilon) W_t$. $\square$

Peng (2016) shows that it is possible to find an $(1-\epsilon)$-approximate flow in undirected graphs in time

$$O(m \log^{32} n (\log \log n)^2 \max(\log^9 n, 1/\epsilon^3)).$$