Recitation Notes: multiplicative weights update

1 Online Algorithms, Predictions, MWU

1. Definition of online algorithms, competitive ratio.
2. Showing lower bounds against online algorithms: build an adversary that takes the algorithm and constructs a bad input for it.
3. Rent or buy problem: optimal competitive ratio of 2 achieved by renting until enough money has been spent to buy, then buy.
4. Prediction with expert advice: predict a 0/1 sequence making mistakes comparable to the best expert.
5. Weighted majority for both pure or fractional/randomized predictions:
   (a) Start everyone off with weight 1.
   (b) Every time someone makes mistake, decrease their weight by factor of $\exp(-\beta)$.
   (c) Move by the weighted majority choice at each step, either as a pure or mixed (randomized) strategy.
6. Analyze multiplicative weights under the assumption of $\exp(\beta) \approx 1 + \beta$ when β is small.
7. **optional** Solving zero-sum games using MWUs. (this is only included to 'prove' duality, for this course we only require invocations of duality, not proving it)

   For finding the column player strategy

   $$\min_{x \text{ probability distribution}} \max Ax$$

   repeatedly set $x^{(t+1)}$ to be best response against the row player playing the mixed strategy

   $$\exp\left(\beta A\left(x^{(1)} + x^{(2)} + \ldots + x^{(t)}\right)\right).$$

   Main steps of analysis:
   (a) If there is probability distribution y such that $A^\top y > 1$ (entrywise), then weak duality implies it's not possible for such an x to exist where $Ax \leq 1$: $y^\top Ax > 1^\top x = 1$, but also $y^\top Ax \leq y^\top 1 = 1$, contradiction.

(b) So at any step, choosing the mixed row player strategy

$$y_i^{(t)} \leftarrow \frac{w_i^{(t)}}{\Phi(t)}$$

ensures that the best response $x^{(t+1)}$ satisfies

$$\sum_i w_i^{(t)} \left(Ax^{(t+1)}\right)_i \leq \Phi(t).$$

(c) But the change in potential function is approximately (using $\exp(\delta) \approx 1+\delta$):

$$\begin{aligned}
\Phi(t+1) &= \sum_i w_i^{(t)} \exp\left(\beta \left(Ax^{(t+1)}\right)_i\right) \\
&\approx \sum_i w_i^{(t)} \left(1+\beta \left(Ax^{(t+1)}\right)_i\right) \\
&= \Phi(t) + \beta \sum_i w_i^{(t)} \left(Ax^{(t+1)}\right)_i \\
&\leq \exp\left(\beta\right) \Phi(t).
\end{aligned}$$

so by induction we get $\Phi(t) \leq \exp(t\beta)n$.

(d) So we get $(A(x^{(1)} + x^{(2)} + \ldots x^{(t)}))_i \leq \beta^{-1} \ln(w_i^{(t)}) \leq t + \frac{\ln n}{\beta}$. Once $t > \ln/\beta$, we get $\max Ax$ gets arbitrarily close to 1.

2 Flows Using MWU

We show that just playing the MWU algorithm on the congestion game shown in oral homework 3, problem 2, leads to an algorithm that produces approximate min-congestion flows.

Pseudocode of the algorithm, due to Garg and Konemann (`https://ieeexplore.ieee.org/document/743463/`), for unit capacitated s-t max-flow, is as follows.

1. Initialize weights on edges to 1
2. Repeatedly:
   (a) Send 1 unit of flow along shortest $s \to t$ path (in terms of weights)
   (b) Double (or multiply by $\exp(\beta)$) all weights on the path found.

Lemma 1: MWU for flows

Show that MWU with shortest path as an oracle finds an $O(1)$-approximate s-t maxflow (congestion minimizing flow) in a unit capacitated graph with max-flow value F in $O(F \log n)$ iterations.

Note: n represents the number of edges; A congestion-minimizing flow is a flow f that satisfies the demands (in this case, sends F units from s to t) that minimizes $\max_e \frac{f_e}{\text{cap}_e}$.

Proof. Let w_e be the weight on edge e, so if an edge has f_e units of flow on it, $w_e = 2^{f_e}$.

The total potential is

$$\Phi = \sum_e w_e = \sum_e 2^{f_e}$$

we want to show that after $O(F \log n)$ iterations, $\Phi \le 2^{O(\log n)}$.

At any time, since the maximum s-t flow is F, there exist F edge-disjoint s-t paths. Thus, by averaging, there exists a path P whose total weight satisfies

$$\sum_{e \in P} w_e \le \frac{\Phi(t)}{F}.$$

In each iteration, the algorithm selects the shortest path (with respect to weights), so the chosen path has weight at most $\Phi(t)/F$. We then multiply the weight of every edge on this path by 2. Hence, the increase in potential is at most the total weight of the path, and we obtain

$$\Phi(t+1) \le \Phi(t) + \frac{\Phi(t)}{F} = \Phi(t)\left(1 + \frac{1}{F}\right).$$

Unrolling the recurrence and using $(1+1/F)^F \le e$, we get

$$\Phi(t) \le \Phi(0)\left(1 + \frac{1}{F}\right)^t \le n \cdot \exp\left(\frac{t}{F}\right),$$

since initially $\Phi(0) = \sum_e 1 \le n$.

For $t = O(F \log n)$, this gives

$$\Phi(t) \le \exp\big(O(\log n)\big) = 2^{O(\log n)}.$$

Finally, for any edge e, we have $w_e \le \Phi(t)$, hence

$$f_e = \log_2 w_e \le \log_2 \Phi(t) = O(\log n).$$

Since capacities are 1, this implies the congestion is $O(\log n)$. So rescaling this resulting flow down by a factor $O(\log n)$ leads to a flow that obeys the capacities. The amount sent here is $t/\log n = \Omega(F)$, so an $O(1)$-approximation to the maximum flow. $\square$