15-451/651 Algorithms, Fall 2017
Recitation #12 Worksheet

Multiplicative Weights

1. In lecture we saw that the simple procedure that multiplied the weight of each expert by $\frac{1}{2}$ whenever the expert made a mistake, resulted in

$$m = \text{ \#mistakes of algorithm } \leq 2.41(M + \log_2 n),$$

where M = #mistakes made by the best expert and n = # of experts. If we change the weight by 2/3 at each time, how does this analysis change?

Solution: Again, potential is total weight. Every time we make a mistake, total weight goes down by $5/6$. So final weight is $n(5/6)^m$. And every time best expert makes a mistake its weight drops by $2/3$. So $(2/3)^M \geq n(5/6)^m$, and hence

$$m \leq \frac{1}{\log_2(6/5)}(M \log_2(3/2) + \log_2 n).$$

2. In the lecture: in order to get a better mistake bound of $(1+\epsilon)M + O(\frac{\ln n}{\epsilon})$, we used randomization. Let us now show that you cannot get better than a factor of 2 if you don't use randomness.

There are two experts. One always predicts 0*. The other always predicts* 1*. Fix any deterministic algorithm* A *for prediction. Here is one sequence of days: each day, the actual outcome is the opposite of what the algorithm predicts.*

After T days, the algorithm would have made T mistakes. Show that the better of the two experts makes at most $T/2$ mistakes. Hence infer that $m \geq 2M$.

Solution: Each day exactly one of the two experts is correct. So by the pigeonhole principle, one of them makes $\leq T/2$ mistakes.

Finding misplaced keys (Online Algorithms)

You are at the origin of $\mathbb{Z}$ and have misplaced your keys at some integer n, the value of n is unknown to you. If you are at position i, in a single step you can move to $i+1$ or $i-1$. To check if your keys are at some position i, you must visit i. Your goal is to find your keys while minimizing the number of steps you take. If you knew what n was, you would simply walk there in n steps, but since you don't know the value of n, you are satisfied with taking cn steps, where c is some small constant.

1. A natural strategy to try is to visit nodes in the following order: $1, -1, 2, -2, 3, -3, \ldots$ Show that this is not a good strategy, that is, it has a competitive ratio of $\Omega(n)$.

   Solution: The competitive ratio is unbounded. Note that if the keys are at $X > 0$, we will travel at least $1 + 2 + \ldots + X = \Omega(X^2)$, so the competitive ratio is $\frac{\Omega(X^2)}{X} = \Omega(X)$. And by making X large, we can make the competitive ratio as large as we want.

2. Consider this alternate strategy. You visit nodes in the order $1, -1, 2, -2, 4, -4, 8 - 8, \ldots$ Show that this strategy has a competitive ratio of at most 13.

   Solution: Suppose $X > 0$. Then let $X = 2^i + y$, where $y < 2^i$. Now the algorithm can be viewed as starting from 0, going to 1 and returning to 0, going to -1 and returning to 0, going to 2 and back, -2 and back, …, 2^i and back, -2^i and back, and then finally reaching X. So the total distance traveled by the algorithm is

   $$2(1 + 1 + 2 + 2 + 4 + 4 + \ldots + 2^i + 2^i) + X = 4(2^{i+1} - 1) + X.$$

   If $X = -(2^i + y) < 0$, then we get an additional trip to 2^{i+1} and back on the positive side before reaching the (negative) X, so the total travel is $4(2^{i+1} - 1) + 2 \cdot 2^{i+1} + |X| = 12 \cdot 2^i + |X| - 4$. To get an upper bound on the comp.ratio, we want the algorithm to travel the most for the same value of $|X|$, so the worst-case input will choose a negative X. This gives us a comp.ratio of $12\frac{2^i}{|X|} + 1 - \frac{4}{|X|}$. And to maximize this, we should make $y = |X| - 2^i$ as small as possible, so set $y = 1$, and hence we get the comp.ratio $13 - o(1)$.

3. As a slight optimization over part (b), visit nodes in order $1, -2, 4, -8, 16, -32, \ldots$ Show that this strategy has a competitive ratio of at most 9.

   Solution: Again, suppose $X = 2^i + y > 0$. This time we can think of us traveling to

   $$1, 0, -2, 0, 4, 0, \ldots, 2^i, 0, -2^{i+1}, 0, X$$

   So the total distance is

   $$2(1 + 2 + \ldots + 2^{i+1}) + X = 2(2^{i+2} - 1) + X = 8 \cdot 2^i + X - 2.$$

   And if you do the calculation for negative Xs, you get a similar distance. So the competitive ratio this time is $\frac{8 \cdot 2^i + |X| - 2}{|X|} = 9 - o(1)$.