These lecture notes have not undergone rigorous peer-review. Please email quanquan.liu@yale.edu if you see any errors.

1 Introduction

Today we'll begin our discussion of the *multiplicative update method (MWU)* which is a fundamental tool for obtain many scalable graph algorithms. We'll begin our discussion with the weighted majority algorithm for the online prediction with expert advice problem. Suppose we use the classic example of the stock market where on each day, we want to predict whether the stock goes up or down. We have N experts who give advice on each day on whether the stock goes up or down on that day. You may consult the experts in any way you'd like to give you prediction for each day. If you predict, *correctly* then you receive nothing for that day; however, if you predict *incorrectly*, then you lose $1 for that day. (I know, it's a tough world out there in stock market land.) Your goal is to minimize your loss using expert advice.

If there exists one expert who always answers correctly, then there exists a relatively straightforward algorithm for this problem that always achieves loss upper bounded by $O(\log N)$. The algorithm in this case always takes the majority opinion and eliminate experts that are wrong each day. Thus, on each day t on which you predict wrongly, you eliminate at least $N_t/2$ experts where N_t is the number of experts you started with on day t. Thus, by the end of $\log_2(N)$ days in which you are wrong, you have only the correct expert remaining.

1.1 Weighted Majority

Now, suppose as is likely in the real world, no expert is correct all of the time. However, some experts are correct more often than other experts. You want to perform as well or almost as well as the *best* expert, hence, making as few mistakes as the best expert. Our weighted majority algorithm is given in Algorithm 1. The weighted majority algorithm maintains a weight of the experts so that experts who are correct for more of the time are given higher weights and experts which are incorrect are given lower weights. The algorithm initially sets all weights of experts to 1; then, based on the weights of the experts and their advice, we determine the weighted majority opinion. We predict the value UP or DOWN depending on which option has more of the vote. Then, for each expert which predicted the wrong value, we decrease their weight by a factor of $1/(1+\eta)$ for the next day.

Algorithm 1: Weighted Majority Algorithm

Input: A set of experts $i \in [N]$, number of rounds T
Output: Predicted directions UP or DOWN

for $i = 1$ **to** N **do**
  Initialize $w_i^1 \leftarrow 1$
end
for $t = 1$ **to** T **do**
  foreach *expert* $i \in [N]$ **do**
    Expert i advises UP or DOWN
  end
  Predict UP or DOWN based on the weighted majority using the weights w_i^t
  foreach *expert* i *that predicted incorrectly* **do**
    Update weight $w_i^{t+1} \leftarrow w_i^t/(1+\eta)$
  end
end

1.2 Upper Bounding the Number of Mistakes

We now upper bound the number of mistakes made by our weighted majority algorithm which we will denote as WM. We denote the number of mistakes made by WM by M_{WM}. Specifically, we prove the following theorem.

Theorem 1.1 (Weighted Majority Utility)**.** *For any sequence of observed outcomes, after T days and given expert i, it holds that the number of mistakes made by WM is upper bounded by:*

$$2(1+\eta) \cdot M_i + O\left(\frac{\log N}{\varepsilon}\right)$$

for any $\eta \in [0, 1)$ and where M_i is the number of total mistakes made by expert i.

Proof. We define the following potential function:

$$\Pi^t = \sum_{i \in [N]} w_i^t. \tag{1.1}$$

We first note the following characteristics:

1. $\Pi^1 = N$ since all experts have weight 1 on day 1.
2. $\left(\frac{1}{1+\eta}\right)^{M_i} \leq \Pi^{T+1}$ since Π^{T+1} is the sum of all of the weight of all experts after the T-th day and these weights are multiplied by a factor of $1/(1+\eta)$ for each day the expert is wrong.
3. On any day τ when WM is wrong, at least half of the weights of all of the experts gets multiplied by $1/(1+\eta)$. So, on day $\tau \in [T]$, $\Pi^{\tau+1} \leq \left(1 - \frac{1}{2(1+\eta)}\right)\Pi^\tau$. This means that $\Pi^{T+1} \leq \left(1 - \frac{1}{2(1+\eta)}\right)^{M_{WM}} \cdot \Pi^1 = \left(1 - \frac{1}{2(1+\eta)}\right)^{M_{WM}} \cdot N$.

Now we combine all of the above characteristics. We first know by 2 and 3 that

$$\left(\frac{1}{1+\eta}\right)^{M_i} \leq \left(1 - \frac{1}{2(1+\eta)}\right)^{M_{WM}} \cdot N.$$

We then take the logarithm of both size of the inequality above to obtain

$$\begin{aligned}
-M_i &\leq \log_{(1+\eta)}(N) + M_{WM} \cdot \log_{(1+\eta)}\left(\frac{1+2\eta}{2(1+\eta)}\right) \\
-M_i &\leq \log_{(1+\eta)}(N) - M_{WM} \cdot \log_{(1+\eta)}\left(\frac{2(1+\eta)}{1+2\eta}\right) \\
M_{WM} \cdot \log_{(1+\eta)}\left(\frac{2(1+\eta)}{1+2\eta}\right) &\leq M_i + \log_{(1+\eta)}(N) \\
M_{WM} &\leq \frac{1}{\log_{(1+\eta)}\left(\frac{2(1+\eta)}{1+2\eta}\right)} \cdot \left(M_i + \log_{(1+\eta)}(N)\right) \\
M_{WM} &\leq 2(1+\eta) \cdot (M_i + O(\log(N)/\eta)),
\end{aligned}$$

for any $\eta \in [0, 1)$. □

These lecture notes use materials from [Gup]

References

[Gup] Anupam Gupta. Multiplicative Weight Updates. https://www.cs.cmu.edu/afs/cs.cmu.edu/academic/class/15859-f11/www/notes/lecture16.pdf.