Lecturer: Anupam Gupta Scribe: Bryan Lee,Albert Gu, Eugene Choi

1 The Mistake Bound Model

Suppose there are N experts who make predictions about a certain event every day – for example, whether it rains today or not. At the beginning of each time step t, the experts give their predictions in a vector $\mathcal{E}^t$. We then also make a prediction about the outcome. Finally, we see the actual outcome o^t. The goal is to minimize the number of times our prediction differs from the outcome.

Example 11.1. Suppose $\mathcal{E}^t = (0, 1, 0, 0, 0, 1, 1, 1, 0)$. We predict 0, but the actual outcome is $o^t = 1$, so this is a mistake.

Fact 11.2. *If there is a perfect expert, then there is an algorithm that makes at most* $\log_2 N$ *mistakes.*

Proof. The algorithm is given by Littlestone and Warmuth [1]. At each step, look at the all the experts who have made no mistakes so far. Predict what the majority of them predict. Note that everytime we make a mistake, the number of experts who have not been wrong yet is cut in half. Since there is at least one perfect expert, we can make at most $\log N$ mistakes. □

Fact 11.3. *If the best expert makes* m *mistakes, there is an algorithm that makes at most* $m(\log_2 N + 1) + \log_2 N$ *mistakes.*

Proof. Think of time as divided into "epochs". In each epoch, we proceed as in the perfect expert scenario, and keep track of all experts who have not made a mistake in that epoch. This set halves with every mistake the algorithm makes. When this set becomes empty, we have made at most $\log_2 N + 1$ mistakes, and every expert has made a mistake. This epoch is over and we start the next epoch.

Note that in each epoch, every expert makes a mistake. Therefore the number of completed epochs is at most m, and the number of mistakes our algorithm makes in these epochs is at most $m(\log_2 N + 1)$. In the last (current) epoch, the algorithm makes at most $\log_2 N$ mistakes as in Fact 11.2. □

2 The Weighted Majority Algorithm

- Assign a weight w_i to expert i. Let $w_i^{(t)}$ denote the weight of expert i at time t. Initially, all weights are 1 ($w_i^{(1)} = 1$).
- At each time t, predict according to the weighted majority of experts using their weights. In other words, choose the outcome that maximizes the sum of weights of experts that predicted it.
- When we see the outcome, set $w_i^{(t+1)} = w_i^{(t)} \cdot \begin{cases} 1 & \text{if } i \text{ was correct} \\ \frac{1}{2} & \text{if } i \text{ was incorrect} \end{cases}$.

Theorem 11.4. *For all times* t *and experts* i*, the number of mistakes the weighted majority algorithm (WM) makes is at most* $2.41(m_i + \log_2 N)$*, where* m_i *is the number of mistakes expert* i *makes.*

Proof. The proof uses a potential function argument. Let $\Phi^t = \sum_{i\in[N]} w_i^{(t)}$. Note that

- $\Phi_1 = N$
- $\Phi_{t+1} \leq \Phi_t$ for all t
- If WM makes a mistake at time t, then the sum of weights of the wrong experts is higher than the sum of the weights of the correct experts. Then

$$\begin{aligned}
\Phi^{t+1} &= \sum_{i \text{ wrong}} w_i^{(t+1)} + \sum_{i \text{ correct}} w_i^{(t+1)} \\
&= \frac{1}{2} \sum_{i \text{ wrong}} w_i^{(t)} + \sum_{i \text{ correct}} w_i^{(t)} \\
&= \Phi^t - \frac{1}{2} \sum_{i \text{ wrong}} w_i^{(t)} \\
&\leq \frac{3}{4}\Phi^t
\end{aligned}$$

Therefore if expert i makes m_i mistakes and WM makes M mistakes, then

$$\begin{aligned}
&\left(\frac{1}{2}\right)^{m_i} = w_i^{(t+1)} \leq \Phi^{t+1} \leq \Phi^1 \left(\frac{3}{4}\right)^M = N\left(\frac{3}{4}\right)^M \\
\implies & m_i \log_2\left(\frac{1}{2}\right) \leq \log_2 N + M \log_2 \frac{3}{4} \\
\iff & M \leq \frac{m_i + \log_2 N}{\log_2 \frac{4}{3}} \leq 2.41(m_i + \log_2 N)
\end{aligned}$$

□

Corollary 11.5. *By changing the re-weighting process to*

$$w_i^{(t+1)} = w_i^{(t)} \cdot \begin{cases} 1 & \text{if } i \text{ was correct} \\ 1-\varepsilon & \text{if } i \text{ was incorrect} \end{cases}$$

the bound in Theorem 11.4 is

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

Proof. Using similar analysis, we have

$$\begin{aligned}
(1-\varepsilon)^{m_i} &\leq N\left(1-\frac{\varepsilon}{2}\right)^M \\
&\leq N\left(e^{-\frac{\varepsilon M}{2}}\right) && \text{(Bernoulli's Inequality)} \\
m_i \log_2(1-\varepsilon) &\leq -\frac{\varepsilon M}{2} + \log_2 N \\
M &\leq \frac{\log_2 N - m_i \log(1-\varepsilon)}{\frac{\varepsilon}{2}} \\
&\leq 2\frac{\log_2 N}{\varepsilon} + 2\frac{m_i \log \frac{1}{1-\varepsilon}}{\varepsilon} \\
&\leq O\left(\frac{\log N}{\varepsilon}\right) + 2\frac{m_i\varepsilon + \varepsilon^2}{\varepsilon} \\
&\leq O\left(\frac{\log N}{\varepsilon}\right) + 2m_i(1+\varepsilon)
\end{aligned}$$

since $\log_2\left(\frac{1}{1-\varepsilon}\right) \approx \varepsilon + \frac{\varepsilon^2}{2} \leq \varepsilon + \varepsilon^2$ for $\varepsilon \in [0,1]$.

□

Remark 11.6. No deterministic algorithm $\mathcal{A}$ can do better than a factor of 2 compared to the best expert.

Proof. Then consider a scenario with 2 experts A, B, the first of whom always predicts 1 for each time step t, and the second of whom always predicts 0 for each time step t. Since $\mathcal{A}$ is deterministic, an adversary can fix all outcomes such that $\mathcal{A}$'s predictions are always wrong. Then at least 1 of A and B will have an error rate of ≤ 0.5, while $\mathcal{A}$'s error rate is 1. □

3 Randomized Weighted Majority

The randomized weighted majority algorithm (RMW) proceeds as MW, except the prediction is probabilistic based on the current weights of the experts.

- As before, set $w_i^{(1)} = 1$ for all i.
- After time t, predict

$$\begin{cases} 0 & \text{with probability } \dfrac{\sum\limits_{i:\mathcal{E}_i^t=0} w_i^{(t)}}{\sum\limits_i w_i^{(t)}} \\ 1 & \text{otherwise} \end{cases}$$

- After seeing o^t, set $w_i^{(t+1)} \leftarrow w_i^{(t)} \cdot \begin{cases} 1 & \text{if } i \text{ was correct} \\ 1-\varepsilon & \text{otherwise} \end{cases}$

Theorem 11.7. *Given any input sequence of $\mathcal{E}$ and o, for any prefix of length T, and for all i, if RWM makes M mistakes and expert i makes m_i mistakes then*

$$\mathbf{E}[M] \leq (1+\varepsilon)m_i + O\left(\frac{\log N}{\varepsilon}\right)$$

Remark 11.8. The $\varepsilon m_i + O\left(\frac{\log N}{\varepsilon}\right)$ gap from the best expert is called the *regret*.

Remark 11.9. When describing randomized algorithms, we must be careful in defining what adversaries can do.

Oblivious Adversary Plans entire sequence up front: The inputs $\mathcal{E}^1, o^1, \mathcal{E}^2, o^2, \cdots$ are pre-determined.

Adaptive Adversary Sees our prediction *before* producing output: In order, it creates $\mathcal{E}^1 \to$ sees prediction $\to o^t \to \mathcal{E} \cdots$

Semi-Adaptive Adversary Like the adaptive adversary, but our prediction happens in parallel with the actual outcome; neither depends on the other.

The adversaries are equivalent on deterministic algorithms, because it always outputs the same prediction and the oblivious adversary could have calculated that in advance when creating $\mathcal{E}^{t+1}$.

RWM works in the semi-adaptive model because predictions are not affected by the future. [1]

Proof of Theorem 11.7. Define the potential $\Phi^t = \sum_i w_i^{(t)}$ as before. Let

$$F^t = \frac{\sum_i \text{ incorrect } w_i^{(t)}}{\sum_i w_i^{(t)}}$$

be the fraction of weight on incorrect experts at time t.

Because we predict proportionally to the weights of the experts, the probably that RWM makes a mistake at time t is F_t. Therefore

$$\mathbf{E}[M] = \sum_{t \in [T]} F_t$$

By our re-weighting rules,

$$\Phi^{t+1} = \Phi^t \left((1 - F_t) + F_t(1 - \varepsilon)\right) = \Phi^t(1 - \varepsilon F_t)$$

Bounding the size of the potential after T steps,

$$\begin{aligned}
(1-\varepsilon)^{m_i} = w_i^{T+1} \le \Phi^{T+1} = \Phi^1 \prod_{t=1}^{T}(1 - \varepsilon F_t) \le N e^{-\varepsilon \sum F_t} = N e^{-\varepsilon \mathbf{E}[M]} \\
\implies m_i \ln(1-\varepsilon) \le \ln N - \varepsilon \mathbf{E}[M] \\
\implies \mathbf{E}[M] \le \frac{m_i \ln\left(\frac{1}{1-\varepsilon}\right) + \ln N}{\varepsilon}
\end{aligned}$$

where we used the inequality $1 + x \le e^x$ in the first line. Finally, note that $\ln\left(\frac{1}{1-\varepsilon}\right) \approx \varepsilon + \frac{\varepsilon^2}{2} \le \varepsilon + \varepsilon^2$ for $\varepsilon \in [0, 1]$. So we get the bound

$$\mathbf{E}[M] \le m_i(1+\varepsilon) + \frac{\ln N}{\varepsilon}$$

□

[1] In the literature, what we call the semi-adaptive adversary here is known as an adaptive adversary. There is no common name for what we call the adaptive adversary.

4 Extending the "game"

We can generalize the notion of experts who are right or wrong to arbitrary gain/loss functions. At each time t, suppose the algorithm produces a vector (of probabilities) $\bar{p}^{(t)} = (p_1^t p_2^t \cdots p_N^t) \in \Delta_N$ (Δ_N is the probability simplex so $p_i \geq 0$ and $\sum_i p_i^t = 1$). In parallel, the adversary produces loss vector $\bar{\ell}^{(t)} = (\ell_1^t \cdots \ell_N^t) \in [-1, 1]^N$. Our loss or cost is $\langle \bar{p}^{(t)}, \bar{\ell}^{(t)} \rangle$.

Theorem 11.10. *Consider a fixed $\varepsilon \leq 1$. For all sequences of loss vectors, for all times T, and for all indices $i \in [N]$, there exists a deterministic algorithm such that*

$$\sum_{i=1}^{T} \langle \bar{p}^{(t)}, \bar{\ell}^{(t)} \rangle \leq \sum_{t=1}^{T} \bar{\ell}_i^{(t)} + \varepsilon T + \frac{\ln N}{\varepsilon}$$

Corollary 11.11. *For all $T \geq \frac{4 \log N}{\varepsilon^2}$, the average loss is bounded by*

$$\frac{1}{T} \sum_t \langle \bar{p}^{(t)}, \bar{\ell}^{(t)} \rangle \leq \frac{1}{T} \sum_t \bar{\ell}_i^{(t)} + \varepsilon$$

Remark 11.12. In the scenario with experts, the loss vector can be thought of as a vector of 0 and 1 representing whether they are wrong. When the loss vector is non-negative, we can use the multiplicative weights algorithm to get a slightly stronger bound.

Proof of Theorem 11.10. The Hedge algorithm [2] proceeds as follows for a fixed ε.

As usual, consider weights $w_i^{(t)}$ initialized to $w_i^1 = 1$, and set $\Phi^t = \sum_i w_i^{(t)}$. Choose probabilities $p_i^{(t)} = \frac{w_i^t}{\Phi^t}$, so that $\sum_i p_i^{(t)} = 1$.

After each round, set $w_i^{(t+1)} \leftarrow w_i^{(t)} \cdot e^{-\varepsilon \ell_i^{(t)}}$.

Note that $\Phi^1 = N$, and

$$\begin{aligned}
\Phi^{t+1} &= \sum w_i^{(t+1)} = \sum_i w_i^{(t)} e^{-\varepsilon \bar{\ell}_i^{(t)}} \\
&\leq \sum_i w_i^{(t)} \left(1 - \varepsilon \bar{\ell}_i^t + \varepsilon^2 \left(\bar{\ell}_i^t\right)^2\right) \qquad \text{(using the inequality } e^x \leq 1 + x + x^2 \forall x \in [-1, 1]\text{)} \\
&\leq \sum w_i^{(t)} (1 + \varepsilon^2) - \varepsilon \sum w_i^{(t)} \bar{\ell}_i^{(t)} \qquad \text{(because } |\bar{\ell}_i^{(t)}| \leq 1\text{)} \\
&= \Phi^t (1 + \varepsilon^2) - \varepsilon \Phi^t \langle \bar{p}^{(t)}, \bar{\ell}_i^{(t)} \rangle \\
&= \Phi^t \left(1 + \varepsilon^2 - \varepsilon \langle \bar{p}^{(t)}, \bar{\ell}_i^{(t)} \rangle\right) \\
&\leq \Phi^t e^{\varepsilon^2 - \varepsilon \langle \bar{p}^{(t)}, \bar{\ell}_i^{(t)} \rangle} \qquad \text{(using } 1 + x \leq e^x\text{)}
\end{aligned}$$

So

$$\begin{aligned}
& e^{-\varepsilon \sum \bar{\ell}_i^{(t)}} = w_i^{(t+1)} \leq \Phi^{T+1} \leq \Phi^1 e^{\varepsilon^2 T - \varepsilon \sum \langle \bar{p}^{(t)}, \bar{\ell}_i^{(t)} \rangle} \\
\implies & -\varepsilon \sum \bar{\ell}_i^{(t)} \leq \ln N + \varepsilon^2 T - \varepsilon \sum_t \langle \bar{p}^{(t)}, \bar{\ell}_i^{(t)} \rangle \\
\implies & \sum_t \langle \bar{p}^{(t)}, \bar{\ell}_i^{(t)} \rangle \leq \sum \bar{\ell}_i^{(t)} + \varepsilon T + \frac{\ln N}{\varepsilon}
\end{aligned}$$

□

Note that if we choose $\epsilon = \sqrt{\frac{\ln N}{T}}$, then $\epsilon T + \frac{\ln N}{\epsilon} = 2\sqrt{T \ln N}$, so that the regret term is *sublinear* in time T. This indicates that the average regret $\leq 2\sqrt{\frac{\ln N}{T}}$ of Hedge(ϵ) converges towards the best expert, so that Hedge(ϵ) is in some sense "learning".

For future reference, we state the analogous result for gains g^t instead of losses ℓ^t, i.e., $g^t = -\ell^t$.

Theorem 11.13. *For every $0 < \epsilon \leq 1$, there exists an algorithm Hedge$_g(\epsilon)$ such that for all times $T > 0$, for every sequence of gain vectors $(g^1, \ldots, g^T)$, and for every $i \in \{1, \ldots, n\}$, at every time $t \leq T$, Hedge$_g(\epsilon)$ produces $p^t \in \Delta_N$ such that*

$$\sum_{t=1}^{T} \langle g^t, p^t \rangle \geq \sum_{t=1}^{T} \langle g^t, e_i \rangle - \epsilon T - \frac{\ln N}{\epsilon},$$

where e_i is the ith vector in the standard basis of $\mathbb{R}^N$. Note that the first term on the right hand side represents the gain of the ith expert, and the last two terms represents the regret of not having always chosen the ith expert.

We also state a corollary of 11.13 that we will use in a future lecture regarding zero-sum games.

Corollary 11.14. *Let $\rho \geq 1$. For every $0 < \epsilon \leq \frac{1}{2}$, for all times $T \geq \frac{4\rho^2 \ln N}{\epsilon^2}$, for all sequences of gain vectors $(g^1, \ldots, g^T)$ with each $g^t \in [-\rho, \rho]^N$, and for all $i \in \{1, \ldots, N\}$, at every time $t \leq T$, Hedge$_g(\epsilon)$ produces $p^t \in \Delta_N$ such that*

$$\frac{1}{T} \sum_{t=1}^{T} \langle g^t, p^t \rangle \geq \frac{1}{T} \sum_{t=1}^{T} \langle g^t, e_i \rangle - \epsilon.$$

5 Bandits

A further extension to this game occurs when one considers the scenario where the player, having made a prediction i_t randomly based on $\bar{p}^{(t)}$, is only given access to $l_i^{(t)}$ instead of the entire loss vector $\bar{l}^{(t)}$. An example of a realization of this problem comes from the analysis of slot machines, or as they are also affectionately known, "one-armed bandits".

Consider the following algorithm:

- We proceed as in the RWM algorithm in defining $\bar{p}^{(t)}$.
- Upon seeing $\bar{l}_i^{(t)}$ for choice i_t, we construct $\tilde{l}^{(t)}$ as follows:

$$\tilde{l}_j^{(t)} = \begin{cases} 0 & \text{if } j \neq i_t \\ \frac{l_j^{(t)}}{p_j^{(t)}} & \text{if } j = i_t \end{cases}$$

It turns out that this approximate algorithm achieves a fairly good estimate of the underlying loss vector. It can be shown that the following result holds:

Remark 11.15. $\mathbf{E}[\tilde{l}^{(t)}] = \bar{l}^{(t)}$.

References

[1] Nick Littlestone, Manfred K. Warmuth *The Weighted Majority Algorithm.* Information and Computation 108:212-261, 1994.
`https://users.soe.ucsc.edu/ manfred/pubs/J24.pdf` 1

[2] Yoav Freund, Robert E. Schapire *A Decision-Theoretic Generalization of On-line Learning and an Application to Boosting* Journal of Computer and System Sciences 55:119-139, 1997.

4