CMSC5706 Topics in Theoretical Computer Science

Week 10: Online Learning

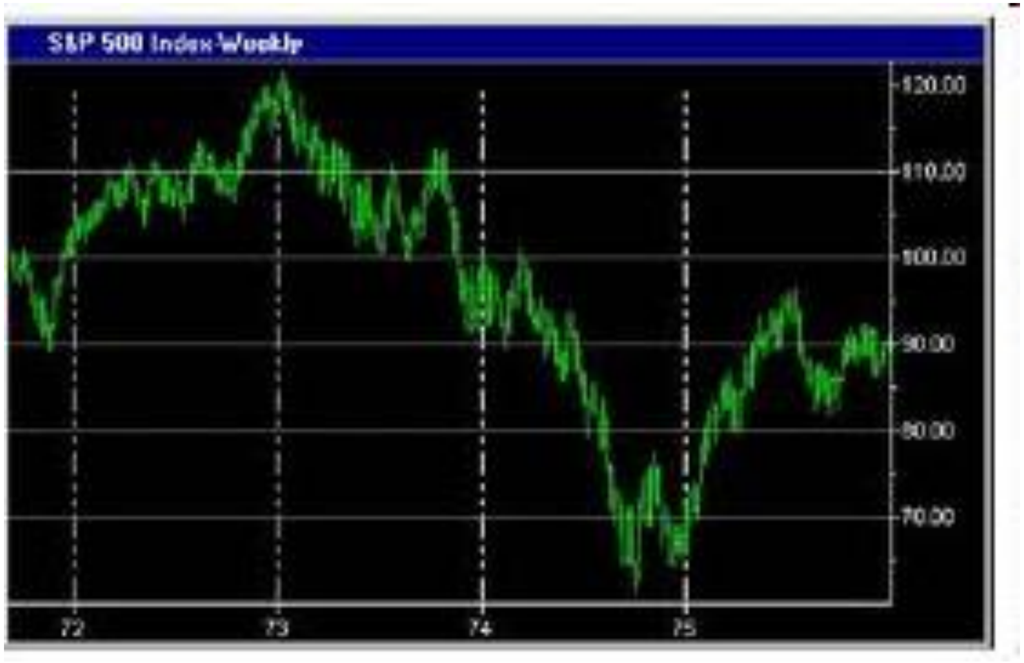
Instructor: Shengyu Zhang

Location change for the final 2 classes

- Nov 17: YIA 404 (Yasumoto International Academic Park 康本國際學術園)
- Nov 24: No class.
  - Conference leave.
- Dec 1: YIA 508 (Yasumoto International Academic Park 康本國際學術園)

Problem 1: Experts problem

Stock market



- Simplification: Only consider up or down.

Which expert to follow?

- Each day, stock market goes up or down.

- Each morning, n "experts" predict the market.
- How should we do? Whom to listen to? Or combine their advice in some way?

Which expert to follow?

- Each day, stock market goes up or down.

- At the end of the day, we'll see whether the market actually goes up or down.
- We lose 1 if our prediction was wrong.

- After a year, we'll see with hindsight that one expert is the best.
  - But, of course, we don't know who in advance.
- We'll think "If we had followed his advice…"
- Theorem: We have a method to perform close to the best expert!
  - We don't assume anything about the experts.
    - They may not know what they are talking about.
    - They may even collaborate in any bad manner.

Method and intuition

- Algorithm: *Randomized Weighted Majority*

- Use random choice: following expert i with probability p_i
- If an expert predicts wrongly: punish him by decreasing the probability of choosing him/her in next round.
  - If someone gives you wrong info, then you tend to trust him less in future.

Randomized Weighted Majority

$w_i^{(t)}$: weight of expert i at time t

$p_i^{(t)}$: probability of choosing expert i at time t

- **for** each $i \in [n]$

  $$w_i^{(1)} = 1, \;\; p_i^{(1)} = 1/n$$

- **for** each $t > 1$, $\forall i \in [n]$:
  - **if** expert i was wrong at step $t - 1$

    $$w_i^{(t)} = w_i^{(t-1)}(1 - \varepsilon)$$

    Decrease your weight!

    else

    $$w_i^{(t)} = w_i^{(t-1)}$$

  - $p_i^{(t)} = w_i^{(t)} / \sum_i w_i^{(t)}$

    Probability is proportional to weight

  - Choose i with prob. $p_i^{(t)}$, and follow expert i's advice.

Example (n=5, T=6, ε = 1/4)

	1	2	3	4	5	our	real
1	1, ↑	1, ↑	1, ↓	1, ↑	1, ↓	↑	↑
2	1, ↑	1, ↓	0.75, ↑	1, ↑	0.75, ↑	↑	↑
3	1, ↑	0.75, ↑	0.75, ↓	1, ↓	0.75, ↑	↓	↓
4	0.75, ↑	0.5625, ↑	0.75, ↓	0.75, ↓	0.5625, ↑	↑	↓
5	0.5625, ↓	0.4219, ↑	0.75, ↑	0.75, ↓	0.4219, ↓	↓	↑
6	0.4219, ↑	0.4219, ↑	0.75, ↓	0.5625, ↑	0.3164, ↑	↓	↓
loss	4	4	1	2	5	2	

- Numbers: weight
- Arrows: predications. Red: wrong.

- L_{RWM}: expected loss of our algorithm
- L_{min}: loss of the best expert

- Theorem. For $\epsilon < 1/2$, the loss on any sequence of $\{0,1\}$ in time T satisfies

$$L_{RWM} \leq (1+\epsilon)L_{min} + \ln(n)/\epsilon.$$

  - n: number of experts. (The more experts, the harder to catch the best one.)

Proof

- Key: Consider the total weight $W^{(t)}$ at time t.
- Fact: Any time our algorithm has significant expected loss, the total weight drops substantially.
- $l_i^{(t)}$: 1 if expert i is wrong at step t (and 0 otherwise)
- Let $F^{(t)} = (\sum_{i:l_i^{(t)}=1} w_i^{(t)})/W^{(t)}$. Two meanings:
  - The fraction of the weight on wrong experts
  - The expected loss of our algorithm at step t
- Note:$W^{(t+1)} = F^{(t)}W^{(t)}(1-\epsilon) + (1-F^{(t)})W^{(t)}$
  $= W^{(t)}(1-\epsilon F^{(t)})$

- Last slide: $W^{(t+1)} = W^{(t)}\left(1 - \epsilon F^{(t)}\right)$
- So $W^{(T+1)} = W^{(T)}(1 - \epsilon F^{(T)})$

$$= W^{(T-1)}\left(1 - \epsilon F^{(T-1)}\right)\left(1 - \epsilon F^{(T)}\right)$$

$$= \ \ldots$$

$$= W^{(1)}(1 - \epsilon F^{(1)}) \ \ldots (1 - \epsilon F^{(T)})$$

- On the other hand,

$$W^{(T+1)} \geq \max_i w_i^{(T+1)} = (1-\epsilon)^{L_{min}^{(T)}}$$

- So $(1-\epsilon)^{L_{min}^{(T)}} \leq W^{(1)}(1-\epsilon F^{(1)}) \ldots (1-\epsilon F^{(T)})$
- Note: $L_{min}^{(T)}$ is the loss of the best expert.

$$(1-\epsilon)^{L_{min}^{(T)}} \leq W^{(1)}(1-\epsilon F^{(1)}) \ldots (1-\epsilon F^{(T)})$$

- Note that $W^{(1)} = n$ since $w_i^{(1)} = 1, \forall i$
- Take log:

$$L_{min}^{(T)} \ln(1-\epsilon) \leq \ln(n) + \sum_{t=1,\ldots,T} \ln(1-\epsilon F^{(t)})$$

$$\leq \ln(n) - \sum_{t=1,\ldots,T} \epsilon F^{(t)} \quad \because \ln(1-z) \leq -z$$

$$= \ln(n) - \epsilon L_{RWM}^{(T)} \quad \because L_{RWM}^{(T)} = \sum_{t=1,\ldots,T} F^{(t)}$$

  - $L_{RWM}^{(T)}$ is the loss of our algorithm.

- Rearranging the inequality and using

$$-\ln(1-z) \leq z + z^2, \qquad 0 \leq z \leq 1/2$$

we get the inequality in the theorem.

$$L_{RWM} \leq (1+\epsilon) L_{min} + \ln(n)/\epsilon.$$

Extensions

- The case that T is unknown.
- The case that loss is in $[0,1]$ instead of $\{0,1\}$

- References:
  - **The Multiplicative Weights Update Method: a Meta-Algorithm and Applications**, Sanjeev Arora, Elad Hazan, and Satyen Kale, Theory of Computing, Volume 8, Article 6 pp. 121-164, 2012.
  - Chapter 4 of *Algorithmic Game Theory*, available at http://www.cs.cmu.edu/~avrim/Papers/regret-chapter.pdf

Problem 2: Multi-armed Bandit

One-armed bandit

- Bandit: a robber or outlaw belonging to a gang and typically operating in an isolated or lawless area.

- One-armed bandit:

Multi-armed bandit

- *Question: Which machine to play?*

Formal model

- k “arms”, each with a fixed but unknown distribution of reward.
  - Assume for simplicity that reward is in $[0,1]$.
- In particular, the expectation μ_i of machine i’s reward, is unknown.
  - If all μ_i’s are known, then the task is easy: just pick the $\max_i \mu_i$.
- Unfortunately the μ_i’s are unknown, thus we face the question of which arm to pull.

Operation, feedback and reward

- At each time step $t = 1,2, \ldots, T$:
  - each machine i has a random reward $X_{i,t}$.
    - $E[X_{i,t}] = \mu_i$, independent of the past.
  - we pick a machine I_t, and get reward $X_{I_t,t}$.
  - we don't see other machines' rewards.

Formal model

- Over the time period $t = 1,2, \ldots, T$, we get the total reward $\sum_{t=1}^{T} X_{I_t,t}$.
- If we had known all μ_i's, we would just have selected $\max_i \mu_i$ at each time t, which has expected total reward $T \cdot \max_i \mu_i$.
- Our "regret": $T \cdot \max_{i=1,\ldots,k} \mu_i \; - \; \sum_{t=1}^{T} X_{I_t,t}$.

  best machine's reward (in expectation)

  our reward

- *Question: How small can this regret be?*

Exploration vs. exploitation dilemma

- Exploration: to find the best.
  - Overhead: big loss when trying the bad arms.
- Exploitation: to exploit what we've discovered
  - weakness: there may be better ones that we haven't explored and identified.
- *Question: With the fixed budget, how to balance the exploration and exploitation, so that the total loss is small?*

Observations and ideas

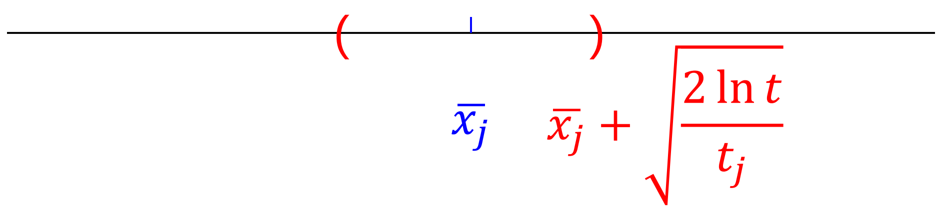
- Where does the loss come from?
- If μ_i is small, trying this arm too many times makes a big loss.
  - So we should try it less if we find the previous samples from it are bad.
- But how to know whether an arm is good?
- The more we try an arm i, the more information we get about its distribution.
  - In particular, the better estimate to its mean μ_i.

Observations and ideas

- So we want to estimate each μ_i precisely, and at the same time, don't try bad arms too often.
- These are two competing tasks.
  - Exploration vs. exploitation dilemma
- Rough idea: we try an arm if
  - either we haven't tried it often enough
  - or our estimate of μ_i so far looks good
- Next: an algorithm implementing this idea quantitatively.

Upper Confidence Bound (UCB)

- Pull each of the k arms once.
- **for** $t = k + 1, \dots, T$ **do**:
  - Pull arm j that maximizes $\bar{x}_j + \sqrt{\frac{2 \ln t}{T_j(t-1)}}$, where
  - $\bar{x}_j$: the average reward obtained from arm j so far,
  - $T_j(t-1)$: number of times arm j has been played in first $t-1$ rounds,



Performance

- Recall: Regret = $T \cdot \mu^* - \sum_{t=1}^{T} X_{I_t,t}$,
  - where $\mu^* = \max_{i=1,\dots,k} \mu_i$.
- Let $\Delta_i \stackrel{\text{def}}{=} \mu^* - \mu_i$,
  - the expected loss of pulling arm i once.
  - Independent of T (how long we play). Think of it as a constant.
- *Theorem. If each distribution of reward has support in* $[0,1]$, *then the regret of the UCB algorithm is at most*

$$O\left(\sum_{i:\mu_i<\mu^*} \frac{\ln T}{\Delta_i} + \sum_{j\in[k]} \Delta_j\right)$$

Performance

- *Theorem. If each distribution of reward has support in* [0,1], *then the regret of the UCB algorithm is at most*

$$O\left(\sum_{i:\mu_i<\mu^*}\frac{\ln T}{\Delta_i}+\sum_{j\in[k]}\Delta_j\right)$$

- The loss grows very slowly with T.
  - Only logarithmically.

Performance

- *Theorem. If each distribution of reward has support in* $[0,1]$, *then the regret of the UCB algorithm is at most*

$$O\left(\sum_{i:\mu_i<\mu^*}\frac{\ln T}{\Delta_i}+\sum_{j\in[k]}\Delta_j\right)$$

- We will show that for each suboptimal arm j, the expected number of times being pulled is $\frac{8}{\Delta_j^2}\ln T+O(1)$,
  - thus the overall loss is $O\left(\sum_{i:\mu_i<\mu^*}\frac{\ln T}{\Delta_i}+\sum_{j\in[k]}\Delta_j\right)$.

- Recall that $T_j(t)$ is the number of times arm j has been played by time t.
  - Thus $\sum_j T_j(t) = t$.
- The expected regret after time t is

$$\sum_{j:\mu_j<\mu^*} \mathbf{E}\big[T_j(t)\big]\Delta_j.$$

  - Recall that Δ_i is the one-time regret.
- So it's enough to bound $\mathbf{E}\big[T_j(t)\big]$.

- For an event A, we will use $\mathbb{I}[A]$ to denote the indicator function.
  - $\mathbb{I}[A] = \begin{cases} 1 & A\ happens \\ 0 & A\ doesn't\ happen \end{cases}$
- $T_i(T) = 1 + \sum_{t=k+1}^{T} \mathbb{I}[I_t = i]$
  - 1: we pulled each arm once at the beginning.
- For each ℓ (a parameter to be fixed later), considering whether $I_t \leq \ell$, we have

$$\mathbb{I}[I_t = i] \leq \ell + \mathbb{I}[I_t = i, T_i(n-1) \geq \ell]$$

- Note that in the algorithm, we pick whichever arm has the maximum $\bar{x}_j + \sqrt{\frac{2\ln t}{T_j(t-1)}}$.
- So if we pick i, then

$$\bar{X}_{i^*,T_{i^*}(t-1)} + c_{t-1,T_{i^*}(t-1)} \le \bar{X}_{i,T_i(t-1)} + c_{t-1,T_i(t-1)}$$

  - $X_{i,t}$: the random award arm i gives at time t
  - $\bar{X}_{i,n} = \frac{1}{n}\sum_{t=1}^{n} X_{i,t}$
    - The average award obtained from the first n samples of arm i.
  - $c_{t,s} \stackrel{\text{def}}{=} \sqrt{(2\ln t)/s}$.
- $\mathbb{I}[I_t = i, T_i(t-1) \ge \ell] \le \mathbb{I}\Big[\bar{X}_{i^*,T_{i^*}(t-1)} + c_{t-1,T_{i^*}(t-1)} \le \bar{X}_{i,T_i(t-1)} +$

- For the condition $\bar{X}_{i^*,T_{i^*}(t-1)} + c_{t-1,T_{i^*}(t-1)} \leq \bar{X}_{i,T_i(t-1)} + c_{t-1,T_i(t-1)}$, we don't know which is i^* and how many times i^* and i have been pulled.
- So let's use union bound: The above inequality implies that $\exists s \in [t-1]$ and $s_i \in [\ell, t]$, s.t. $\bar{X}_{i^*,s} + c_{t-1,s} \leq \bar{X}_{i,s_i} + c_{t-1,s_i}$
- Therefore, $\mathbb{I}\left[\bar{X}_{i^*,T_{i^*}(t-1)} + c_{t-1,T_{i^*}(t-1)} \leq\right.$

- In summary, we have (roughly) the following.
$T_i(T) \leq \ell + \sum_{t=K}^{T} \sum_{s=1}^{t-1} \sum_{s_i=1}^{t-1} \mathbb{I}\left[\bar{X}_{i^*,s} + c_{t,s} \leq \bar{X}_{i,s_i} + c_{t,s_i}\right]$
- Note that the event needs at least one of the following three to hold.
    - $\bar{X}_{i^*,s} \leq \mu^* - c_{t,s}$
    - $\bar{X}_{i,s_i} \geq \mu_i + c_{t,s_i}$
    - $\mu^* < \mu_i + 2c_{t,s_i}$
- Otherwise, we'd have

$$
\begin{aligned}
& \bar{X}_{i^*,s} + c_{t,s} > \mu^* && \text{(by 1)} \\
\geq\ & \mu_i + 2c_{t,s_i} && \text{(by 3)} \\
>\ & \bar{X}_{i,s_i} - c_{t,s_i} + 2c_{t,s_i} && \text{(by 2)} \\
=\ & \bar{X}_{i,s_i} + c_{t,s_i}
\end{aligned}
$$

The three conditions

- $\bar{X}_{i^*,s} \leq \mu^* - c_{t,s}$
  - The estimate of i^* is too small
- $\bar{X}_{i,s_i} \geq \mu_i + c_{t,s_i}$
  - The estimate of i is too large
- $\mu^* < \mu_i + 2c_{t,s_i}$
  - The two expectations μ^* and μ_i are very close.

The third one

- $\mu^* < \mu_i + 2c_{t,s_i}$
- Third one is simply false for $\ell = \frac{8 \ln T}{\Delta_i^2}$.
  - Indeed, $\mu^* - \mu_i - 2c_{t,s_i} = \mu^* - \mu_i - 2\sqrt{\frac{2 \ln t}{s_i}}$

$$\geq \mu^* - \mu_i - \Delta_i = 0$$

- Thus one of the first two must happen.

- But the first two events are very unlikely.
- Recall Chernoff-Hoeffding bound: $X_1, \dots, X_n$ are independent random variables in $[0,1]$ with the same expectation μ, let $S = X_1 + \cdots + X_n$. Then $\Pr[S \geq n\mu + a] \leq e^{-2a^2/n}$, and $\Pr[S \leq n\mu - a] \leq e^{-2a^2/n}$.
- Plugging the parameters in, we can see that both event happen with probability t^{-4}.

- Thus overall

$$\mathrm{E}[T_i(T)] \le \frac{8\ln T}{\Delta_i^2} + \sum_{t=K}^{T}\sum_{s=1}^{t-1}\sum_{s_i=1}^{t-1} 2t^{-4}$$
$$\le \frac{8\ln T}{\Delta_i^2} + \sum_{t=K}^{T} 2t^{-2}$$
$$\le \frac{8\ln T}{\Delta_i^2} + O(1)$$

- Recall that the total regret is $\sum_{i:\mu_i<\mu^*} \mathbf{E}\left[T_j(T)\right]\Delta_i$
- Putting the inequality in, we get

$O\left(\sum_{i:\mu_i<\mu^*} \frac{\ln T}{\Delta_i} + \sum_{j\in[k]} \Delta_j\right)$, as claimed.

- In retrospect, the UCB uses the principle of *optimism in face of uncertainty*.
  - We don’t have a good estimate $\hat{\mu}_i$ of μ_i before trying it many times.
  - We thus give a big confidence interval $[-c_i, c_i]$ (governed by Chernoff bound) for such i.
  - And select an i with maximum $\mu_i + c_i$.

- In retrospect, the UCB uses the principle of *optimism in face of uncertainty*.
  - If an arm hasn't been pulled many times, then the big confidence interval makes it still possible to be tried.
  - In face of uncertainty (of μ_i), we act optimistically by giving chances to those that haven't been pulled enough.

Summary

- In Expert problem, we achieved

$$L_{RWM} \leq (1+\epsilon)L_{min} + \ln(n)/\epsilon$$

- In (stochastic) Multi-Armed Bandit problem, we achieved total regret of

$$O\left(\sum_{i:\mu_i<\mu^*} \frac{\ln T}{\Delta_i} + \sum_{j\in[k]} \Delta_j\right)$$