15-859(B) Machine Learning Theory

Lecture 01/21/02

Online learning contd: predicting from expert advice

Avrim Blum

Online learning recap

- See examples one at a time.
- Need to make predictions online.
- Last time: mistake-bound model. Assume target in some class $\mathcal{C}$. Goal: minimize worst-case number of mistakes.

Plan for today:

- Introduce "Expert Advice" problem.
- Some interesting algorithms
- Some applications.

Using "expert" advice

Say we want to predict the stock market.

- We solicit n "experts" for their advice. (Will the market go up or down?)
- We then want to use their advice somehow to make our prediction.

E.g. $(n = 4)$:

Expt 1	Expt 2	Expt 3	neighbor's dog	truth
down	up	up	up	up
down	up	up	down	down
...	...	...	...	...

Rough question: What's a good strategy for combining their opinions, given that in advance we don't know which is best?

["expert" ≡ someone with an opinion. Not necessarily someone who knows anything.]

Simpler question

- We have n "experts".
- One of these is perfect (never makes a mistake). We just don't know which.
- Can we find a strategy that makes no more than $\log_2 n$ mistakes?

Answer: Yes. Just take majority vote of all experts that have been correct so far. Each mistake allows us to cross off at least half.

Expt 1	Expt 2	Expt 3	your dog	truth
down	up	up	up	up
down	up	up	down	down
...	...	...	...	...

This is the "halving algorithm".

Think of "n" as "$|C|$" from last lecture, and "expert" as "function in C".

Back to "expert" advice

What if no expert is perfect? Our goal is just to do nearly as well as the best one in hindsight.

Strategy #1: iterated halving algorithm. Same as before, but once we've crossed off all the experts, restart from the beginning.

- Makes at most $\log(n) \cdot OPT$ mistakes, where OPT is # mistakes of the best expert in hindsight.

Seems wasteful. Constantly forgetting what we've "learned". Can we do better? Yes.

The Weighted Majority Alg. [LW]

Intuition: Making a mistake doesn't completely disqualify an expert. So, instead of crossing off, just lower its weight.

Simple algorithm: **Weighted Majority**

- Start with all experts having weight 1.
- Predict based on weighted majority vote.
- Penalize mistakes by cutting weight in half.

Example:

					prediction	correct
weights	1	1	1	1		
predictions	Y	Y	Y	N	Y	Y
weights	1	1	1	.5		
predictions	Y	N	N	Y	N	Y
weights	1	.5	.5	.5		
predictions	Y	N	N	N	N	N
weights	.5	.5	.5	.5		
predictions	N	Y	N	Y	either	N
weights	.5	.25	.5	.25		

Analysis: do nearly as well as best expert in hindsight

- M = # mistakes we've made so far.
- m = # mistakes best expert has made so far.
- W = total weight (starts at n).

- After each mistake, W drops by at least 25%.

  So, after M mistakes, $W \leq n(3/4)^M$.

- Weight of best expert is $(1/2)^m$. So,

$$(1/2)^m \leq n(3/4)^M$$

$$(4/3)^M \leq n2^m$$

$$M \leq 2.4(m + \lg n)$$

Randomized Weighted Majority

Previous bound not so good if the best expert makes a mistake 20% of the time. Can we do better? Yes.

- Instead of taking majority vote, use weights as probabilities. (e.g., if 70% on up, 30% on down, then pick 70:30)

  Idea: smooth out the worst case.

- Also, generalize 1/2 to $1 - \varepsilon$.

Solves to: $M \leq \dfrac{-m \ln(1-\varepsilon) + \ln(n)}{\varepsilon}$

$$M \leq 1.39m + 2 \ln n \quad \leftarrow \varepsilon = 1/2$$

$$M \leq 1.15m + 4 \ln n \quad \leftarrow \varepsilon = 1/4$$

$$M \leq 1.07m + 8 \ln n \quad \leftarrow \varepsilon = 1/8$$

As $\varepsilon \to 0$, this goes to $(1 + \varepsilon/2)m + \frac{1}{\varepsilon} \ln n$.

Here, M is our expected number of mistakes.

A simpler warmup

Before analyzing, let's go back to simpler problem where we assumed a perfect expert.

- Instead of taking majority, use votes as probabilities.
- Claim: *expected* number of mistakes is at most $\ln n$. (this is better than $\log_2 n$)

Analysis #1: (will analyze in 3 ways)

- Say at time t, the fraction of "alive" experts that make a mistake is F_t.
- So, our expected # mistakes $M = \sum F_t$.
- Also, we know that $\prod(1 - F_t) \geq 1/n$.
- For a fixed product, we minimize the sum when the numbers are all equal [e.g., (12,1) vs (6,2) vs (4,3) vs (3.46,3.46)].
- Solving $(1 - F)^T = 1/n$ gives $F \leq (\ln n)/T$.

⇒ So, $M \leq \ln n$.

A simpler warmup (# 2)

- Instead of taking majority, use votes as probabilities.
- Claim: *expected* number of mistakes is at most $\ln n$. (this is better than $\log_2 n$)

Analysis #2:

- Say at time t, the fraction of "alive" experts that make a mistake is F_t.
- So, our expected # mistakes $M = \sum F_t$.
- Also, we know that $\prod(1 - F_t) \geq 1/n$.
- Take logs: $\sum \ln(1 - F_t) \geq -\ln n$.
- Use the inequality $\ln(1 - x) < -x$:

⇒ $-\sum F_t \geq -\ln n$ ⇒ $M \leq \ln n$.

A simpler warmup (# 3)

- Instead of taking majority, use votes as probabilities.
- Claim: *expected* number of mistakes is at most $\ln n$. (this is better than $\log_2 n$)

Analysis #3:

- Alg can be viewed as "pick expert at random and follow it until it makes a mistake. Then pick a new (alive) expert at random, and continue."

  (Same as Marking Algorithm [BLS])
- Let's number experts in the order they make their first mistake (break ties arbitrarily).
- The chance we were following expert 1 when it made its first mistake is $1/n$. For expert 2 it's at most $1/(n-1)$, etc.
- So, our total expected number of mistakes is at most $\frac{1}{n} + \frac{1}{n-1} + \ldots + \frac{1}{2} + 1 \approx \ln n$.

Randomized Weighted Majority

Analyze like in #2.

- Say at time t we have fraction F_t of weight on experts that made mistake.
- So, our expected # mistakes $M = \sum_t F_t$.

$W_{final} = n(1 - \varepsilon F_1)(1 - \varepsilon F_2) \ldots.$

$$\begin{aligned} \ln(W_{final}) &= \ln(n) + \sum_t [\ln(1 - \varepsilon F_t)] \\ &\leq \ln(n) - \varepsilon \sum_t F_t \\ &\qquad \text{(using } \ln(1 - x) < -x\text{)} \\ &= \ln(n) - \varepsilon M \end{aligned}$$

- If best expert makes m mistakes, then

$$\ln(W_{final}) \geq \ln((1 - \varepsilon)^m).$$

- Now solve....

$$\begin{aligned} M &\geq \frac{-m \ln(1 - \varepsilon) + \ln(n)}{\varepsilon} \\ &\approx (1 + \varepsilon/2)m + \frac{1}{\varepsilon}\log(n) \end{aligned}$$

$\{$Ap,Im$\}$plications

- If you have several strategies and don't know which to use,
- and if you have the computational resources to run them all,

⇒ then can use WM to do nearly as well as the best of them in hindsight.

From theoretical perspective: if we don't care about running time, we can now do nearly as well as *best* $f \in C$.

Things to note

Note 1: Get same bounds even if losses in $[0, 1]$ rather than just $\{0, 1\}$. (Just replace F_t with $\vec{P} \cdot \vec{L_t}$)

Note 2: RWM is in a sense more general than WM because can apply to settings where experts can't be combined. (e.g., viewing as in analysis # 3).

Here is an example....

A fun application

- n buckets. (Think of as startup companies.)
- You are standing in one of them.
- At each time step, a ball falls into one of the buckets. If it's your bucket, you get $1.
- Then you can choose to move if you want.
- Game ends when fullest bucket has d balls.

In a sense this is hopeless if an opponent is tossing the balls based on knowing where you are(n't). E.g., what if you choose to always stand in bucket with most money so far?

But, suppose sequence of balls is predetermined (but unknown). Randomized WM will guarantee you an expected gain of at least $d - \sqrt{2d \log n}$.

[Multiply weight by $1 + \varepsilon$ whenever ball falls in.]

..more..

Let's keep milking this for all we can.

Application to adaptive game playing and proof of min-max theorem. [Freund & Schapire]

2-player zero-sum games

E.g., Rock-Paper-Scissors.

Payoff to row player:

	R	P	S
R	0	-1	1
P	1	0	-1
S	-1	1	0

- Minimax optimal strategy: (randomized) strategy with best worst-case guarantee.

  What is minimax optimal for RPS?

- What about the game below:

  Payoff to row player:

	N	D
N	-5	5
D	10	-10

  Optimal strategy for row player?

  Column player?

The min-max theorem

	N	D
N	-5	5
D	10	-10

- Suppose that for any (randomized) strategy of your opponent, there exists a deterministic counter-strategy for you that guarantees you an expected gain $\geq V$.

  Then, there exists a randomized strategy for you such that for any counter-strategy of the opponent, you get an expected gain $\geq V$.

- Equivalently:

$$\max_{S_{row}} \min_{S_{col}} \mathbf{E}[\text{payoff}] = \min_{S_{col}} \max_{S_{row}} \mathbf{E}[\text{payoff}]$$

  I.e., suppose that for all S_{col} there exists S_{row} such that expected gain is $\geq V$. Then there exists a fixed S_{row} such that for all S_{col} the expected gain is $\geq V$ too.

  (strategy $\equiv$ randomized strategy)

Using RWM for online play

- rows are "experts". Pick row j with probability w_j/W. (This time we *will* pick independent from the past.)
- Penalize rows/experts based on outcome.

  (Technically, let's scale matrix entries to be in the range $[0, 1]$)

	R	P	S
R	1/2	0	1
P	1	1/2	0
S	0	1	1/2

- For any sequence of games,

$$\text{Our expected gain} \geq d\left(1 - \frac{\epsilon}{2}\right) - \frac{\ln n}{\epsilon},$$

  where d is gain of best fixed strategy in hindsight (which may or may not be the minimax optimal).

We've actually just *proven* the Min-max theorem.

Why?

What would it mean for min-max to be false?

→ If we know opponents randomized strategy, we can get expected gain $\geq V$, but if we have to choose our randomized strategy first, then opponent can force us to get $\leq V - \delta$.

This contradicts our bound if we use $\epsilon = \delta$. Our gain per game is approaching $\text{OPT}(1 - \epsilon/2)$.

In other words: If there was a gap (V versus $V - \delta$), then for *any* randomized strategy we choose, opponent who knows our strategy should be able to force us to get no more than $V - \delta$ on average per play.

But, we are doing better.

Shifting bounds

This is nice but not fully satisfying. If opponent plays R 100 times, alg will win a lot but...

- Ends up with $w_P = 1, w_R = 2^{-50}, w_S = 2^{-100}$.
- So, if opponent switches to S, alg will lose 50 times in a row.

How to fix?

- Lower-bound weights (need to be a careful)
- Weight-sharing: if remove X from total weight, give back αX, distributed evenly.

For all t, our expected gain is at least (approx):

$$d_t(1-\varepsilon) - (t+1)\frac{\ln n}{\epsilon}$$

where d_t is the gain of optimal strategy in hindsight that switches between experts t times.