CMU 15-451/651 lecture 11/14/13

An Algorithms-based Intro to Machine Learning, part II

Avrim Blum

Last time / today

Last time: looked at model where data is coming from some probability distribution.

- Take a sample S, find h with low $err_S(h)$.
- Ask: when can we be confident that $err_D(h)$ is low too? (Or more generally, that the gap $|err_D(h) - err_S(h)|$ is low.)
- Gives us confidence in our predictions.

Today: what if we don't assume the future looks like the past. What can we say then?

Online learning

- What if we don't want to make assumption that data is coming from some fixed distribution? Or any assumptions on data?
- Can no longer talk about past performance predicting future results.
- Can we … esting at all??

Idea: reg…

➢Show t… rly as well as best p… s.

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.,

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

Basic question: Is there a strategy that allows us to do nearly as well as best of these in hindsight?

["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 one.
- Can we find a strategy that makes no more than lg(n) mistakes?

Answer: sure. Just take majority vote over all experts that have been correct so far.

➢Each mistake cuts # available by factor of 2.

➢Note: this means ok for n to be very large.

What if no expert is perfect?

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

Weighted Majority Alg:

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

					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		

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 is at most $n(3/4)^M$.
- Weight of best expert is $(1/2)^m$. So,

$$\begin{aligned}(1/2)^m &\leq n(3/4)^M \\ (4/3)^M &\leq n2^m \\ M &\leq 2.4(m + \lg n)\end{aligned}$$

So, if m is small, then M is pretty small too.

Randomized Weighted Majority

2.4(m + lg n) 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 ½ to 1- ε.

Solves to: $M \leq \frac{-m\ln(1-\varepsilon)+\ln(n)}{\varepsilon} \approx (1+\varepsilon/2)m + \frac{1}{\varepsilon}\ln(n)$

M = expected #mistakes

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

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

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

Analysis

- Say at time t we have fraction F_t of weight on experts that made mistake.
- So, we have probability F_t of making a mistake, and we remove an εF_t fraction of the total weight.
  - $W_{final} = n(1-\varepsilon F_1)(1 - \varepsilon F_2)\ldots$
  - $\ln(W_{final}) = \ln(n) + \sum_t [\ln(1 - \varepsilon F_t)] \leq \ln(n) - \varepsilon \sum_t F_t$ (using $\ln(1-x) < -x$)

    $= \ln(n) - \varepsilon M.$ ($\sum F_t = E[\#\text{ mistakes}]$)
- If best expert makes **m** mistakes, then $\ln(W_{final}) > \ln((1-\varepsilon)^m)$.
- Now solve: $\ln(n) - \varepsilon M > m \ln(1-\varepsilon)$.

$$M \leq \frac{-m\ln(1-\varepsilon)+\ln(n)}{\varepsilon} \approx (1+\varepsilon/2)m + \frac{1}{\varepsilon}\log(n)$$

What can we use this for?

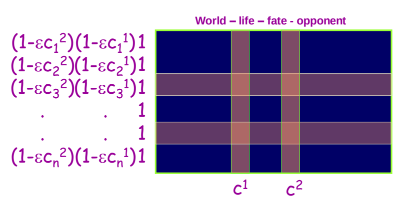
- Can use for repeated play of matrix game:
  - Consider cost matrix where all entries 0 or 1.
  - Rows are different experts. Start each with weight 1.
    - Notice that the RWM algorithm is equivalent to "pick an expert with prob $p_i = w_i/\sum_j w_j$, and go with it".
    - Can apply when experts are *actions* rather than *predictors*.
    - F_t = fraction of weight on rows that had "1" in adversary's column.
  - Analysis shows do nearly as well as best row in hindsight!

What can we use this for?

In fact, alg/analysis extends to costs in [0,1], not just {0,1}.

- We assign weights w_i, inducing probabilities $p_i = w_i/\sum_j w_j$.
- Adversary chooses column. Gives cost vector $\vec{c}$. We pay (expected cost) $\vec{p} \cdot \vec{c}$.
- Update: $w_i \leftarrow w_i(1 - \epsilon c_i)$.

RWM



scaling so costs in [0,1]

$E[\text{cost}] \leq (1+\epsilon)OPT + \left(\frac{1}{\epsilon}\right)\log(n)$

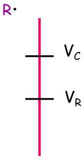
In T steps, $E[\text{cost}] \leq OPT + \epsilon T + \left(\frac{1}{\epsilon}\right)\log(n)$

RWM

In fact, gives a proof of the minimax theorem…

Nice proof of minimax thm (sketch)

- Suppose for contradiction it was false.
- This means some game G has $V_C > V_R$:
  - If Column player commits first, there exists a row that gets the Row player at least V_C.
  - But if Row player has to commit first, the Column player can make him get only V_R.
- Scale matrix so payoffs to row are in [-1,0]. Say $V_R = V_C - \delta$.



Proof sketch, contd

- Now, consider randomized weighted-majority alg, against Col who plays optimally against Row's distrib.
- In T steps,
  - Alg gets ≥ [best row in hindsight] $-\epsilon T - \log(n)/\varepsilon$
  - BRiH $\geq TV_C$ [Best against opponent's empirical distribution]
  - Alg $\leq TV_R$ [Each time, opponent knows your randomized strategy]
  - Gap is δT. Contradicts assumption if use $\varepsilon = \delta/2$, once $T > \log(n)/\epsilon^2$.