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 all??
  - Show that our algorithm does nearly as well as best predictor in some large class.

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 \( \log(n) \) mistakes?

    Idea: reward experts according to their prediction accuracy.
    - Show that we can do nearly as well as best of these in hindsight.

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.

    | weights | predictions | prediction | correct |
    |--------|-------------|------------|---------|
    | 1 1 1 1 | Y Y Y Y     | Y          | Y       |
    | 1 1 1 .5| Y Y N Y     | Y          | N       |
    | 1 .5 .5 | Y Y Y Y     | Y          | Y       |
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, 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% up, 30% on down, then pick 70:30)
  Idea: smooth out the worst case.
- Also, generalize $\frac{1}{2}$ to $1 - \epsilon$.

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

$M = \text{expected # mistakes}$

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.
  - $F_i =$ 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 $\tilde{c}$.
  We pay (expected cost) $\tilde{p} \cdot \tilde{c}$.
- Update: $w_i \leftarrow w_i (1 - \epsilon c_i)$.

$$E[\text{cost}] \leq (1 + \epsilon)OPT + \frac{\epsilon}{2} \log(n)$$

In $T$ steps, $E[\text{cost}] \leq OPT + \epsilon T + \frac{\epsilon}{2} \log(n)$
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,
  - $\text{Alg gets } \geq \text{ [best row in hindsight] } - eT - \log(n)/\epsilon$
  - BRiH $\geq TV_C$ [Best against opponent's empirical distribution]
  - $\text{Alg } \leq TV_R$ [Each time, opponent knows your randomized strategy]
  - Gap is $\delta T$. Contradicts assumption if use $\epsilon = \delta/2$, once $T > \log(n)/\epsilon^2$. 