On-line Algorithms in Machine Learning

Avrim Blum

Carnegie Mellon University

[Taken from presentation given at 1998 School and Workshop on On-line Algorithms, Udine, Italy, and from CS Colloquium at Cornel]

On-line Algorithms

- Algorithms for settings where inputs/data arriving over time.

- Need to make decisions based on info so far, without knowing what will happen in the future.

  (e.g., like investing and unlike problems such as sorting where have all inputs at the start)

- Can't hope to do as well as if you knew the future, but can hope to bound your "regret" — relative performance in hindsight. (Also called "competitiveness")

- Settings are ubiquitous. It's life.

Machine Learning

Machine Learning is often expressed as improving performance at a task based on experience. Many similarities to On-line algorithms:

- Adapt behavior according to inputs received.
- Need to make decisions based on the past, without knowing future.
- Competitive-style minimax criteria useful as performance guarantees.

Plan for today:

- Introduce "Expert Advice" problem.
- Some interesting algorithms
- Relations to game theory and MTS problem.
- Talk about portfolio selection if there's time.

Predicting from "expert" advice

- Each day we get advice from n experts about whether it will rain.
- Then we make a prediction.
- Then we are penalized 1 unit if wrong.

Example:

Day 1	predictions	Y	Y	Y	N	
	correct					Y
Day 2	predictions	Y	N	N	Y	
	correct					N

etc.

Intuitively: want to go with the majority and with the ones with best track record.

The Weighted Majority Algorithm

[Littlestone & Warmuth] [Cesa-Bianchi et al.] [Freund & Schapire]

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, $W \leftarrow cW$ for $c \leq 3/4$.

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

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

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

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

A better bound

- Instead of taking majority vote, use weights as probabilities.
  (e.g., if 70% on Y, 30% on N, then pick 70:30)
- Also, generalize $1/2$ to β.
- If expert has fractional loss ℓ, multiply by β^{ℓ}.

Solves to: $M \leq \frac{m \ln(1/\beta) + \ln(n)}{1-\beta}$.

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

$$M \leq 1.15m + 4\ln n \quad \leftarrow \beta = 3/4$$

$$M \leq 1.07m + 8\ln n \quad \leftarrow \beta = 7/8$$

where M is our expected number of mistakes (loss), m is # mistakes (loss) of best expert.

Nice property: Can apply when voting doesn't make sense.

Analysis

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

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

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

- If best expert makes m mistakes, then

$$\ln(W_{final}) \geq \ln(\beta^m).$$

- Now solve....

$$M \geq \frac{m\ln(1/\beta) + \ln(n)}{1-\beta}$$

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

E.g., nickel/dime game.

Payoff to row player:

	N	D
N	-5	5
D	10	-10

- Minimax optimal strategy: (randomized) strategy with best worst-case guarantee.
- Min-max theorem: if, knowing opponent's (randomized) strategy, we could always guarantee **E**[payoff] $\geq V$, then we must have a fixed (randomized) strategy that also guarantees **E**[payoff] $\geq V$.
- But what if we want to take advantage of opponent's frailties?

Applying Randomized Wtd Majority

- rows are "experts". Pr[pick row i] $= w_i / \sum w_j$.

- Penalize rows/experts based on outcome.

  (Technically we need to scale matrix entries to be in the range $[-1, 0]$)

	R	P	S
R	$-1/2$	-1	0
P	0	$-1/2$	-1
S	-1	0	$-1/2$

- So, in T games, we do nearly as well as best fixed strategy in hindsight (which may be better than the minimax optimal).

This is actually a proof of the Min-Max theorem....

[Freund & Schapire] give tight analysis

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.

Now, Alg does nearly as well as best strategy in hindsight that was allowed to switch k times:

$$\text{Alg's loss } \leq (1+\varepsilon)L_{opt} + \frac{\ln(n)}{\varepsilon}(k+1).$$

[Herbster & Warmuth] [Blum & Burch]

Relation to MTS problem

MTS recap:

- Have a machine that can be in one of n states.

- Can move machine from state i to state j at cost d_{ij}, where d is a distance metric.

- We are given, on-line, a sequence of tasks $\sigma = T_1, T_2, T_3, \ldots$. Each task has a cost vector, e.g., $T_1 = (2, 9, \ldots, 10)$, that gives the cost of processing the task in each of the n states of the machine.

$\rightarrow$ Given a new task T, we can stay in the current state i, paying $T[i]$, or move to a new state j and perform it there, paying $d_{ij} + T[j]$.

Relation to MTS problem, cont.

Expert advice	MTS problem
n Experts	n States
loss vector $(\ell_1, \ldots, \ell_n)$	cost vector $(c_1, \ldots, c_n)$
Movement is free	Costs d_{ij} to move
0-step lookahead	1-step lookahead

Another difference: performance measure.

- Competitive ratio: compete against best in hindsight by aribtrary algorithm.
- Best-expert: compete against best single state/ expert in hindsight.
- Shifting bounds: parametrize the ratio as a function of "complexity" of off-line competitor.

MTS under Uniform Metric

- "Marking Alg" achieves E[online cost] $\leq$ $O(\log n)$ $\cdot$(optimal cost in hindsight).

- WM-sharing achieves

  $(1+\epsilon)\cdot$(local cost)$+O(\frac{1}{\epsilon}\log n)\cdot$(movement cost)

  of arbitrary strategy in hindsight.

- Alg with similar bounds at core of [BBBT] $polylog(n)$ guarantee for arbitrary metric spaces.

$$p_i = \frac{1}{n} + \frac{1}{n}\sum_{j=1}^{n}\left(\frac{\mathrm{OPT}_j - \mathrm{OPT}_i}{r}\right)^t$$