MW and Experts

Note Title 4/14/2020

(1) the Mistake Bounded Model

- difference from CRatio
  - measure the difference not ratio
  - compare not to the best sequence of adap. decisions the algo could have made

    but to the best fixed "expert" in hindsight.

So fixed set of decision profiles could follow (N Aplans) which is best?

Eg. have several paging algos can use, which one should we use?

Get: Bonds of the form

$$\#\text{mistakes of algo} \leq \#\text{mistakes of best expert} \times \underbrace{\quad}_{\text{mult. loss}} + \underbrace{\quad}_{\text{additive loss.}}$$

Ideally: get mult loss $= (1+\varepsilon)$

and additive loss $\frac{\log N}{\varepsilon}$

$\Rightarrow$ almost as good as best expert "in the long run"

Start slow:

(1) Show can get

$$A_g \leq O(\log N)(\text{Best expert} + 1)$$

(a) What if best expert makes no errors. ?
(b) — — — — m^* errors. ?

So both additive and mult terms are $O(\log N)$! ☹

(2) Then introduce the multiplicative weights (MW) or Weighted Majority (WM) algo.

$$A_g \leq \underbrace{2.4}\, m^* + \underbrace{O(\log N)} \;\; !!$$

So much better now!

However: it is a deterministic algorithm, and no deterministic algo can do

$$A_{lg} \leq (2.-\varepsilon)\, m^*$$

☹

③ Randomized Weighted Majority

In the bad example, should have hedged bets.

Suppose make random choices of experts.

Then $E[\text{Algo mistakes}] \leq (1+\varepsilon)\, m^* + O\left(\frac{\log N}{\varepsilon}\right)$

(m^*: best expert.)

This is the holy grail, — cannot do better. ☺

BTW: in all these arguments, show that $\forall$ each expert i

$E[\text{Algo}]$ or Algo $\leq \alpha\, m_i + \beta$

(m_i: expert i's mistakes)

which implies statement for best expert.

See the notes for proof / algo.

assume losses are in $[0,1]$

Slight generalization / restatement of model.

Done simultaneously

- Each time expert gives a loss value ℓ_i^t

  $\Rightarrow$ put them in vector $(\ell_1^t \ \ell_2^t \ \dots \ \ell_N^t) = \vec{\ell}(t)$

- We choose probability of picking $1, 2 \dots N$ as

  $(p_1^t \ p_2^t \ \dots \ p_N^t) = \vec{p}(t)$

$\sum_i p_i^t = 1$, $p_i^t \geq 0$

Then $E[\text{loss for us}] = \langle \vec{p}(t), \vec{\ell}(t) \rangle$ ← inner product

$= \sum_i p_i^t \ell_i^t$

So imagine:

Each time the expert panel / Nature / adversary / world chooses a loss vector ℓ^t
- can depend on $p^1 \ p^2 \dots p^{t-1}$, not on p^t

We choose a prob. vector p^t
- which can depend on the past losses $\ell^1 \ \ell^2 \dots \ell^{t-1}$ but not this time
- also can depend on our choices if we want. (but these are implied by losses)

And then our actual (or realized) loss at time t is dot product

$\langle p^t, \ell^t \rangle$. ← (can think as "expected" loss)

We get that for all experts i

$$\sum_t \langle p^t, \ell^t \rangle \leq \sum_t \ell_i^t \times (1+\varepsilon) + \frac{\log N}{\varepsilon}.$$

$\sum_t \langle p^t, \ell^t \rangle$: our "expected" loss

$\sum_t \ell_i^t$: loss of expert i

$(1+\varepsilon)$: $\times$

$\frac{\log N}{\varepsilon}$: $+$

This view is useful :-

Nature chooses a loss vector ℓ^t in $[0,1]^N$.

We choose an "action" p^t in the "probability simplex"

$$\Delta_N = \{p \in [0,1]^N : \sum_i p_i = 1\}$$

Our loss is $\langle p^t, \ell^t \rangle$

Want to compare with the best coordinate

$$\min_i \ell_i^t = \min_{q^* \in \Delta_N} \langle \ell^t, q^* \rangle$$

make sure you believe this (its simple)

Good: Why is this worth studying?

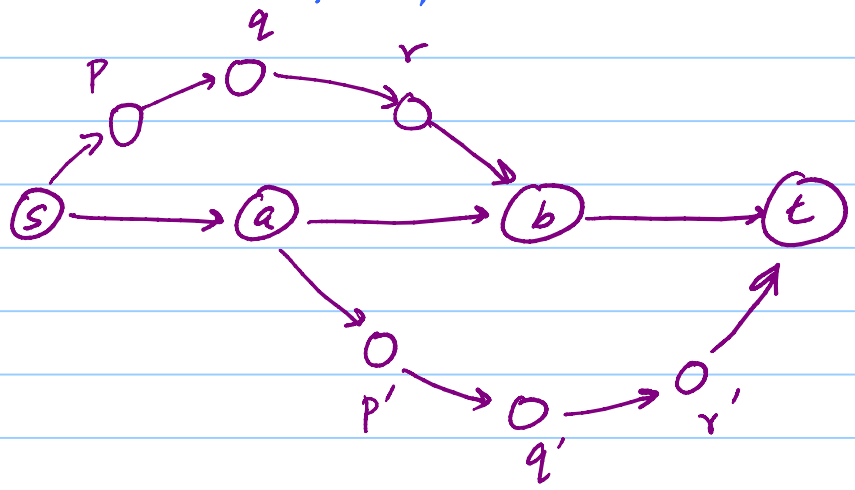
① Choose best of multiple options online

(really choosing the best of N "experts")

↑ maybe self-styled,

② Algorithmic idea of multiplicative weights is very pervasive

- e.g. here's a max-flow problem.

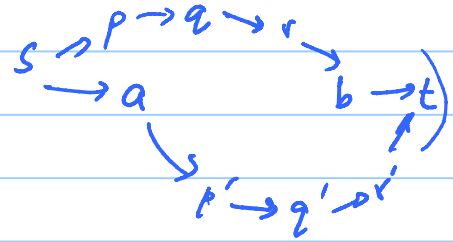


all edges have capacity 1 (say).

say send flow on $s \to a \to b \to t$ (unit amount)

then saturate those edges.

But not a max flow (since $F^* = 2$ ⇒



How to proceed. ① Either use residual graph, etc.

(2) Or use MW.

may depend on capacities, for now assume unit cap.

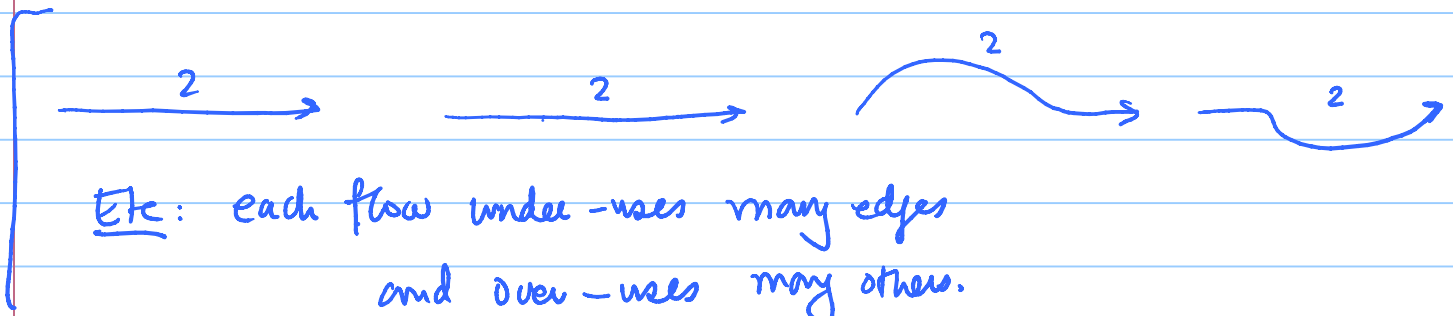
Algo: put lengths 1 on all edges.

→ push F^* flow on shortest path.

Now set lengths to be $(1+\varepsilon)^{\text{flow on edge}}$

repeat $\frac{F^* \log n}{\varepsilon}$ times

Take average of all these flows



Etc: each flow under-uses many edges and over-uses many others.

But averaging them ensures that each edge gets about the right amount of flow.

Another Example: Boosting in Learning. (HW?)

Suppose you have a weak classifier

Given any weights on data points, correctly classifies 51% of data.

Then can use a "majority of weak classifiers" to get a strong classifier

that gets 99% of data right.