Lecture #18

<u>Last Time</u>: Multiplicative Weights & Zero Sum Games

(compete with best expert, find minimax strategies)

<u>Today</u>: Applications of MWU

Part #1 Adaboost

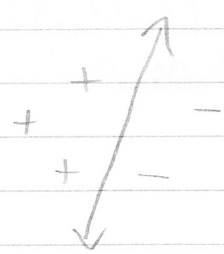
First we will need basic notions from <u>PAC learning</u>, introduced by Valiant

PAC = <u>P</u>robably <u>A</u>pproximately <u>C</u>orrect

Concept class $\mathcal{H}$ (set of functions $f: X \to \{\pm 1\}$)

domain

e.g. linear classifier



We are given $(x_1, f(x_1)), (x_2, f(x_2)) \ldots (x_m, f(x_m))$ sampled from some distribution D on X, and $f \in \mathcal{H}$

<u>Goal</u> (strong learning) Output a hypothesis $h: X \to \{\pm 1\}$ that with probability $\geq 1-\delta$ satisfies

$$\underbrace{\Pr_{x \in D}[h(x) \neq f(x)]}_{err(h, D)} \leq \varepsilon$$

Probably := with probability $\geq 1-\delta$

Approximately Correct := error$(h, D) \leq \varepsilon$

In many settings, much easier to find a <u>weak classifier</u>:

$$err(h, D) \le \frac{1}{2} - \eta$$ ← advantage over random guessing

Today we will use ideas from MWU to prove

weak classifiers ⇒ strong classifiers

<u>First Goal</u>: Combine weak classifiers that have $\le \frac{1}{2} - \eta$ empirical error into a strong classifier

<u>Adaboost</u> [Freund, Schapire]

Given labelled examples $(x_1, h(x_1)) \dots (x_m, h(x_m))$, set $D_1 \equiv$ uniform distribution on them

For $t = 1$ to T

Find weak learner h_t on D_t with error ε_t

Set $\alpha_t = \frac{1}{2} \ln \frac{1-\varepsilon_t}{\varepsilon_t}$

Set $D_{t+1}(x) = \frac{D_t(x) e^{-h_t(x) f(x) \alpha_t}}{Z_t}$ ← normalizing constant

Output $h(x) = \text{sgn}\Big(\underbrace{\sum_{t=1}^{T} \alpha_t h_t(x)}_{\text{weighted majority of weak classifiers}}\Big)$

<u>Note</u>: Each D_t only defined on $x_1, x_2 \dots x_m$

Theorem: Let η_t be such that $\varepsilon_t = \frac{1}{2} - \eta_t$. Then

$$\mathrm{err}(h, D_1) \le e^{-2\sum_{t=1}^{T} \eta_t^2}$$

Proof: Expanding D_t we have

$$D_{T+1}(x_i) = \underbrace{\left(\frac{1}{m}\right)}_{D_1(x)} \left(\frac{e^{-\alpha_1 h_1(x_i) f(x_i)}}{Z_1}\right) \cdots \left(\frac{e^{-\alpha_T h_T(x_i) f(x_i)}}{Z_T}\right)$$

Now we bound the final error

$$\mathrm{err}(h, D_1) = \frac{1}{m} \sum_i \mathbb{1}_{f(x_i) \neq h(x_i)}$$

$$\le \frac{1}{m} \sum_{i=1}^{m} e^{-f(x_i) \sum_{t=1}^{T} \alpha_t h_t(x_i)}$$

$$= \sum_{i=1}^{m} D_{T+1}(x_i) \prod_{t=1}^{T} Z_t$$

All that remains is to bound Z_t.

Claim: $Z_t \le e^{-2\eta_t^2}$

Proof: $Z_t = \sum_{i=1}^{m} D_t(x_i) e^{-\alpha_t h_t(x_i) f(x_i)}$

$$= e^{-\alpha_t} \underbrace{\sum_{\text{correct } x_i} D_t(x_i)}_{1-\varepsilon_t} + e^{\alpha_t} \underbrace{\sum_{\text{incorrect } x_i} D_t(x_i)}_{\varepsilon_t}$$

Hence $Z_t = e^{-\alpha_t}(1-\varepsilon_t) + e^{\alpha_t}\varepsilon_t$

Recall $\alpha_t = \frac{1}{2}\ln\frac{1-\varepsilon_t}{\varepsilon_t} \Rightarrow$

$$Z_t = \left(\sqrt{\frac{\varepsilon_t}{1-\varepsilon_t}}\right)(1-\varepsilon_t) + \left(\sqrt{\frac{1-\varepsilon_t}{\varepsilon_t}}\right)\varepsilon_t$$

$$= 2\sqrt{\varepsilon_t(1-\varepsilon_t)} = 2\sqrt{(\tfrac{1}{2}-\eta_t)(\tfrac{1}{2}+\eta_t)} = \sqrt{1-4\eta_t^2}$$

$$\overset{1-x\le e^{-x}}{\le} e^{-2\eta_t^2} \quad \boxtimes$$

Since D_{t+1} is a distribution, this complets proof $\boxtimes$

What about $err(h, D)$? Freund-Schapire prove

$$err(h, D) \le \underbrace{err(h, D_1)}_{\text{training error}} + \tilde{O}\left(\sqrt{\frac{Td}{m}}\right)$$

VC-dimension of weak classifiers

Intuition: Not too many rounds means $h(x)$ does not get too complicated, hence

low training error $\Rightarrow$ low true error.

This is another example of how to use MWU to build complex algorithms from simpler ones part #2 flows

Next we give another application of MWU

Returning to max flow

$$(P) \quad \max \sum_{p \in \mathcal{P}_{s,t}} x(p) \qquad\qquad (D) \quad \min \sum_{e} \ell(e)$$

$$\text{s.t.} \quad \sum_{p \ni e} x(p) \leq 1 \quad \forall e \qquad\qquad \text{s.t.} \quad \sum_{e \in p} \ell(e) \geq 1 \quad \forall p \in \mathcal{P}_{s,t}$$

$$x(p) \geq 0 \quad \forall p \qquad\qquad \ell(e) \geq 0 \quad \forall e$$

Let γ denote the optimal values. We will consider a zero-sum game

P-player: choose s-t path p

D-player: choose edge e

$$\text{payoff (to D)} = \begin{cases} 1 & \text{if } e \in p \\ 0 & \text{else} \end{cases}$$

Let ν be the optimal value of the game (for D)

Claim: $\nu = \frac{1}{\gamma}$

Proof: From an optimal solution to (D), can construct a randomized strategy:

Choose e with probability $\frac{\ell(e)}{\sum_e \ell(e)} = \frac{\ell(e)}{\gamma}$

By construction, each path $p \in \mathcal{P}_{s,t}$ has expected value of at least $\frac{1}{\gamma}$. Hence $\nu \geq \frac{1}{\gamma}$

Conversely can scale down by γ an optimal solution to (P)

$$\frac{\text{flow}}{\gamma} \Big\} \text{ flow of value } 1 \equiv \text{distribution on paths}$$

By construction each edge has flow at most $\frac{1}{\gamma}$, and its flow is its probability of being on chosen path. Hence $\omega \le \frac{1}{\gamma}$. $\blacksquare$

Now we can apply MWU:

For $t=1$ to $T \left(= \frac{4\gamma^2 \ln m}{\varepsilon^2}\right)$

Use MWU to choose distribution w^t on edges

Let P^t be optimal response from R-player — i.e.

$$P^t \in \operatorname*{argmin}_{P \in \mathcal{P}_{s,t}} \sum_{e \in P} w^t(e)$$

feed in

Set reward vector to be $r^t(e) = \begin{cases} 1 & \text{if } e \in P^t \\ 0 & \text{else} \end{cases}$

Now let f be the flow that routes $\frac{\gamma}{T}$ units on each path $P^1, P^2 \ldots P^T$.

<u>Lemma</u>: f routes at most $1+\varepsilon$ units on each edge

<u>Proof</u>: Suppose not, for some edge e. Then more than $\frac{(1+\varepsilon)T}{\gamma}$ paths use edge e. But then if

the D-player chooses e, in hindsight he would get

$$> \frac{(1+\varepsilon)}{\gamma} \text{ in average}$$

because the P-player is best responding

But how well he actually does is $\leq \frac{1}{\gamma}$, this violates the regret guarantee of MWU by setting T appropriately:

$$\text{we need } 2\sqrt{\frac{\ln m}{T}} \leq \frac{\varepsilon}{\gamma} \iff \frac{4\gamma^2 \ln m}{\varepsilon^2} \leq T \quad \boxtimes$$

When we are given k-source sink pairs $(s_1, t_1) \ldots (s_k, t_k)$ and our goal is to maximize total flow between them, we get Garg-Konneman

In particular, can solve problems like

$$\max \sum_{P \in \bigcup_{i=1}^{k} \mathcal{P}_{s_i, t_i}} x(P)$$

$$\text{s.t.} \sum_{P \ni e} x(P) \leq 1 \quad \forall e$$

$$x(P) \geq 0 \quad \forall P$$

by repeatedly finding shortest path, augmenting and updating distances