MIT 6.854/18.415: Advanced Algorithms	Spring 2016
Lecture 18 – April 11, 2016	
Prof. Ankur Moitra	*Scribe: Yang Liu, Lara Araujo, Steven Hao*

1 Last Time

Last class we talked about multiplicative weights and their application to zero-sum games.

2 Basic Notions from Learning

PAC learning, which was introduced by Valiant [2], stands for *probably approximately correct* learning. To define this we first must define a *concept class*.

Definition 1. *A concept class $\mathcal{H}$ is a set X along with a set of functions $f : X \to \{\pm 1\}$.*

An easy example of a concept class is the set of emails and functions mapping emails to spam or not spam. Another example is the set of points and a line ℓ, and a single function determining which side of the line ℓ the points are on.

PAC learning is the following problem. We are given a concept class $\mathcal{H}$, a function $f \in \mathcal{H}$, and a hidden distribution D on X. The algorithm is allowed to get m labeled examples $(x_i, f(x_i))$ for $1 \le i \le m$, by drawing each $x_i \in X$ according to the distribution D.

We want that for any constants ϵ, δ, after getting these m examples, with probability $1 - \delta$, the algorithm should ensure that the error on future examples drawn from the distribution D is $\le \epsilon$. The δ denotes the *probably*, and the ϵ denotes the *approximately* in the name PAC learning.

Our final definition here will be a *weak learner*.

Definition 2. *A weak learner is one that has error at most $\frac{1}{2} - \eta$ for some $\eta > 0$.*

3 Adaboost

Adaboost is an algorithm introduced by Freund and Schapire [1] that in some sense can take many weak learners and turn them into a strong learner. The precise algorithm follows.

We will construct distributions $D_1, D_2, \ldots, D_{T+1}$ on the m example objects.

1. Start with some examples $(x_i, f(x_i))$ for $1 \le i \le m$.
2. Set D_1 to be the uniform distribution on the examples.
3. Loop from $t = 1$ to T.

4. Find a weak learner h_t on D_t, with error ϵ_t.
5. Set $\alpha_t = \frac{1}{2}\log\frac{1-\epsilon_t}{\epsilon_t}$. Afterwards, set $D_{t+1}(x) = D_t(x)\exp(-h_t(x)f(x)\alpha_t)$. The term $h_t(x)f(x)$ simply denotes whether h_t and f agree on x or not. Afterwards, normalize D_{t+1}. Let this normalization factor be Z_t.
6. After looping all the way through, output

$$h(x) = \operatorname{sgn}\left(\sum_{t=1}^{T}\alpha_t h_t(x)\right),$$

where sgn is a function returning ± 1 denoting whether the input is positive or negative.

Theorem 3. *Let $err(h, D_1)$ denote the error of h on D_1. If we let $\eta_t = \frac{1}{2} - \epsilon_t$, then*

$$err(h, D_1) \leq \exp\left(-2\sum \eta_t^2\right).$$

Proof. Expanding D_{t+1} we get that

$$D_{T+1}(x_i) = \frac{1}{m}\frac{\exp(-\alpha_1 h_1(x_i)f(x_i))}{Z_1}\cdots\frac{\exp(-\alpha_T h_T(x_i)f(x_i))}{Z_T}$$

Then, we bound the final error

$$\begin{aligned} err(h, D_1) &= \frac{1}{m}\sum_{i=1}^{m}\mathbb{1}_{f(x_i)\neq h(x_i)} \\ &\leq \frac{1}{m}\sum_{i=1}^{m}\exp(-f(x_i)\sum_{t=1}^{T}\alpha_t h_t(x_i)) \\ &\leq \sum_{i=1}^{m}D_{T+1}(x_i)\prod_{t=1}^{T}Z_t \end{aligned}$$

Now, if we can bound Z_t we will complete the proof, since D_{T+1} is a distribution.

Claim 4. $Z_t \leq \exp(-2\eta_t^2)$

$$\begin{aligned} Z_t &\leq \sum_{i=1}^{m}D_t(x_i)\exp(-\alpha_t h_t(x_i)f(x_i)) \\ &= \sum_{\text{correct } x_i} D_t(x_i)\exp(-\alpha_t) + \sum_{\text{incorrect } x_i} D_t(x_i)\exp(\alpha_t) \end{aligned}$$

Recall that $\alpha_t = \frac{1}{2} \log \frac{1-\epsilon_t}{\epsilon_t}$

$$\begin{aligned} Z_t &= 2\sqrt{\epsilon_t(1-\epsilon_t)} \\ &= 2\sqrt{\left(\frac{1}{2} - \eta_t\right)\left(\frac{1}{2} + \eta_t\right)} \\ &= 2\sqrt{\left(\frac{1}{4} - \eta_t^2\right)} \\ &\leq \exp(-2\eta_t^2) \end{aligned}$$

□

But what about $err(h, D)$?

Intuition: if we do not have too many rounds of boosting, this means that $h(x)$ does not get too complicated, so low training error $\rightarrow$ low true error.

Freunde-Shapire proved the following, where d is the VC-dimension of the weak classifiers.

$$err(h, D) \leq err(h, D_1) + \tilde{O}\left(\sqrt{\frac{Td}{m}}\right)$$

4 Approximating Max Flow

Consider an (unweighted) instance of max flow:

- (P): $\max \sum_{P \in \mathcal{P}_{s,t}} x(P)$ such that $\sum_{P \ni e} x(P) \leq 1$ and $x(P) \geq 0$.
- (D): $\min \sum_e l(e)$ such that $\sum_{e \in P} l(e) \geq 1$ and $l(e) \geq 0$.

Let γ denote the optimal flow. Consider the following Zero-Sum Game. We have two players: P and D, for primal and dual.

- The P-player chooses some s-t path P.
- The D-player chooses edge e.

The payoff for D is 1 if $e \in P$, and 0 otherwise. Note that, for larger min cuts, the game is harder for D.

Lemma 5. *Let ν be the optimal value for D. Then, $\nu = \frac{1}{\gamma}$.*

Proof. Given an optimal solution for the (fractional) min-cut, then we choose e with probability $\frac{l(e)}{\sum_e l(e)} = \frac{l(e)}{\gamma}$. By construction, for all paths, $\sum_{e \in P} \mathbb{P}(D \text{ chooses } e) \geq \frac{1}{\gamma}$.

Conversely, given an optimal solution to P, we choose paths according to $x(P)$. Since fro each path p, $\sum_{P \ni e} x(p) \leq 1$, the chance that dual player catches an edge in the selected path by primal player is at most $\frac{1}{\gamma}$.

Thus, this primal-dual pair corresponds precisely to this zero-sum game. $\square$

Now, if we run multiplicative weights on this zero-sum game, we can find a good solution to max flow.

For each $t = 1 \ldots T$, use MWU to choose distribution w_t on edges for the D-player. Let P^t be the best response to w_t, which corresponds to the shortest path. Set the reward vector as $r^t(e) = \mathbb{I}_{e \in P^t}$.

Let f be the flow that routes $\frac{\gamma}{T}$ units of flow on each $P^1 \ldots P^T$.

Lemma 6. *f routes at most $1 + \epsilon$ units on each edge, for $T = \frac{4\gamma^2 \ln m}{\epsilon^2}$. Essentially, scaling f down slightly gives a valid flow.*

Proof. Suppose for contradiction that there exists some e such that f routs more than $(1 + \epsilon)$ on e. In other words, more than $\frac{(1+\epsilon)T}{\gamma}$ of the paths $P^1, \cdots, P^T$ use e.

Then, if the D-player plays this edge in hindsight, he would get larger than $\frac{1+\epsilon}{\gamma}$ in average payoff.

However, each step, he gets at most $\frac{1}{\gamma}$ in expectation, as P^t is a best-response. Then, if we set T sufficiently large, we get a contradiction with MWU. $\square$

Thus, we have a way of solving flow (approximately) with MWU.

References

[1] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

[2] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.