CS 170 HW 12

Due on 2019-04-22, at 11:59 pm

1 Study Group

List the names and SIDs of the members in your study group.

2 Experts Alternatives

Recall the experts problem. Every day you must take the advice of one of n experts. At the end of each day t, if you take advice from expert i, the advice costs you some c_i^t in $[0, 1]$. You want to minimize the regret R, defined as:

$$R = \frac{1}{T}\left(\sum_{t=1}^{T} c_{i(t)}^t - \min_{1 \le i \le n} \sum_{t=1}^{T} c_i^t\right)$$

where $i(t)$ is the expert you choose on day t. Your strategy will be probabilities where p_i^t denotes the probability with which you choose expert i on day t. Assume an all powerful adversary can look at your strategy ahead of time and decide the costs associated with each expert on each day. Give the maximum possible (expected) regret that the adversary can guarantee if your strategy is:

(a) Choose expert 1 at every step. That is, if $\forall t\ p_1^t = 1$ and C_i^t is the set of costs for all experts and all days, what is $\max_{C_i^t} R$?

(b) Any deterministic strategy. Note that a "deterministic strategy" can be thought of as a probability distribution that satisfies the following: $\forall t\ \exists i\ p_i^t = 1$.

(c) Always choose an expert according to some fixed probability distribution at every time step. That is, if for some $p_1 \dots p_n$, $\forall t, p_i^t = p_i$, what is $\max_{C_i^t}(\mathbb{E}[R])$?

   What distribution minimizes the regret of this strategy? In other words, what is $\mathrm{argmin}_{p_1 \dots p_n} \max_{C_i^t}(\mathbb{E}[R])$?

This analysis should conclude that a good strategy for the problem must necessarily be randomized and adaptive.

3 Follow the regularized leader

(a) **Follow the leader.** You are playing T rounds of the following game: At round t you pick one of n strategies; your payoff for picking strategy i is $A(t, i) \in [0, 1]$. You try the following algorithm: at each iteration pick the strategy which gave the highest average payoff so far (on the first iteration, you pick strategy 1).

   Give an example of payoffs for $T = 100$ and $n = 2$, where your algorithm obtains a payoff of 0, but sticking to either $i = 1$ or $i = 2$ would have given you a payoff of almost 50.

(b) **Follow the randomized leader.** The reason the algorithm above didn't do so well, is because when we deterministically jump from one strategy to another, an adversarially chosen set of strategies can be designed to thwart the algorithm.

To trick such adversaries, we want to use a *randomized* strategy; at time t we pick our strategy i at random from distribution D_t (These distributions are chosen upfront, i.e., before the algorithm is run and not chosen adaptively based on the loss functions). Let $p_t(i) \geq 0$ denote the probability that we assign to strategy i (i.e. $\sum_{i=1}^{n} p_t(i) = 1$).

The previous algorithm ("Follow the leader") corresponds to setting D_t that maximizes

$$\sum_{i=1}^{n} \left(p_t(i) \cdot \sum_{\tau \in \{1,\ldots,t-1\}} [A(\tau, i)] \right).$$

Why is this no better?.

(c) **Follow the regularized leader.** Instead, it is common to add a /regularized term that favors smoother distributions. A commonly used regularizer is the entropy function, i.e. we want to use pick i from the distribution that maximizes

$$\sum_{i=1}^{n} \left(p_t(i) \cdot \sum_{\tau \in \{1,\ldots,t-1\}} [A(\tau, i)] - \eta p_t(i) \ln p_t(i) \right). \tag{1}$$

(Here, $\eta > 0$ is a parameter that we can tweak to balance exploration and exploitation. Notice also that $\ln p_t(i) \leq 0$.)

In this exercise you will show that following the regularized leader with the entropy regularizer is the same as Multiplicative Weights Update!

Show that for any distribution p_t, (1) is at most

$$\eta \cdot \ln \left(\sum_{i=1}^{n} e^{\sum_{\tau \in \{1,\ldots,t-1\}} [A(\tau, i)]/\eta} \right) \tag{2}$$

(Hint: you may use the inequality $\sum_{i=1}^{n} p_t\left(i\right) \cdot \ln\left(y_i\right) \leq \ln\left(\sum_{i=1}^{n} p_t\left(i\right) \cdot y_i\right)$ for any vector $\vec{y}$.)

(d) Show that for some choice of ϵ (which depends on η), when computing p_t using Multiplicative Weight Update, (1) is equal to (2). What is the dependence of ϵ on η?

4 Linear Classifiers using Multiplicative Weights

In this problem, we will learn a basic algorithm in machine learning used to learn linear classifiers. The setup is as follows : We are given m labelled examples $(\mathbf{a}_1, y_1), \ldots, (\mathbf{a}_m, y_m)$ where $\mathbf{a}_j \in \mathbb{R}^n$ are m *data points* where each data point is a list of n *features* (we assume that all the co-ordinates/feature of all the data points $\mathbf{a}_j$ are either +1 or -1), and $y_j \in \{-1, +1\}$

are their labels. This can be modelled (**after some transformations to the problem which lets us get rid of the labels $\mathbf{y}_j$**) as the following LP which we will aim to solve:

$$\begin{aligned} \textsf{for all} \quad j = 1, \ldots, m: \quad & \langle \mathbf{a}_j, \mathbf{x} \rangle \geq 0 \\ & \langle \mathbf{1}, \mathbf{x} \rangle = 1 \\ \textsf{for all} \quad i = 1, \ldots, n: \quad & \mathbf{x}_i \geq 0 \end{aligned}$$

Here $\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=1}^{n} x[i] \cdot y[i]$ and $\mathbf{v}[j]$ refers to the j^{th} component of the vector.

Assume that there is a *large-margin* solution, i.e., there exists a $\eta > 0$ and a distribution $\mathbf{x}^*$ so that for all j, we have $\langle \mathbf{a}_j, \mathbf{x}^* \rangle \geq \eta$. Consider the following bound on the regret of the multiplicative weights algorithm (which is a modification of the one showed in class; See Section 3 in the notes for the Multiplicative Weights lecture for the general theorem).

THEOREM The multiplicative weights algorithm start with an iterate $\mathbf{x}_1 \in \mathbb{R}^n$ and then suffer a sequence of loss vectors $\ell_1, \ldots, \ell_T \in \mathbb{R}^n$ such that $\ell_t[i] \in [-1, 1]$ and $\varepsilon < 1/2$ and we produce a sequence of iterates $\mathbf{x}_2, \ldots, \mathbf{x}_T$ such that for any $i \in \{1, \ldots, n\}$, we have

$$\sum_{t=1}^{T} \langle \ell_t, x_t \rangle \leq \frac{\log(n)}{\varepsilon} + \min_{i=1}^{n} \sum_{t=1}^{T} (\ell_t[i] + \varepsilon |\ell_t[i]|)$$

(a) Using the bounds in the regret of the multiplicative weights algorithm mentioned above, prove that for any probability distribution $\mathbf{p}^*$,

$$\sum_{t=1}^{T} \langle \ell_t, x_t \rangle \leq \frac{\log(n)}{\varepsilon} + \sum_{i=1}^{n} \mathbf{p}^*[i] \Big(\sum_{t=1}^{T} (\ell_t[i] + \varepsilon |\ell_t[i]|) \Big)$$

(b) We want to use Multiplicative weights to find an $\mathbf{x}$ which satisfies the above LP. We will think of being an adversary who is providing the loss functions based on the constraints. What are the experts for running the algorithm?

(c) What are the loss functions for running the Multiplicative weights algorithm? Note that your loss functions should satisfy the conditions for applying the regret bound (assumptions required for the above theorem). (Hint : Consider what happens when there is some constraint not satisfied by your current iterate $\mathbf{x}_t$. Can you use that to find a loss vector to improve your solution?)

(d) Clearly specify the condition when you will terminate the algorithm?

(e) Combine parts (a) - (d) to devise a multiplicative weights algorithm to find a $\widetilde{\mathbf{x}}$ such that $\langle \mathbf{a}_j, \widetilde{\mathbf{x}} \rangle \geq 0$ for all j and $\widetilde{\mathbf{x}}$ is a probability distribution,i.e., we are looking for a solution $\widetilde{\mathbf{x}}$ to the above LP. Clearly specify what $\widetilde{x}$ should be. Also state the number of iterations T it takes to find this solution. You will have to take $\varepsilon = \eta/2$ for the algorithm. (You should think of what should $\mathbf{p}^*$ supposed to be and how should we plug in what you have derived in part (b)-(d) into the regret bound in part (a)).