Mathematical Foundations of Machine Learning(CS 4783/5783)

Lecture 10: Online Learning, Exponential Weights Algorithm

1 Mind Reading Machine

Most of you guys would have played games like Rock-Paper-Scissors and Matching-Pennies while growing up. The excitement of these games is in trying to predict the future — the next choice of the opponent. Of course, if opponent is random, there is no good strategy, and the game becomes boring. This boring strategy is in fact minimax optimal. However, it is the subtle cues from the other player and their past behavior that make the game interesting. Does the opponent tend to play "Rock" after losing with "Scissors"?, do they try to play more heads than tails?, does the opponent tend to stick with the same choice after winning a round? We try to notice such patterns in behavior to tip the balance in our favor.

Can we program a computer to beat humans at these games? sThis question was asked by Claude Shannon and David Hagelbarger in the 1950's. While at AT&T Bell Labs, they each built a machine—aptly called "mind reader"—to play the game of Matching-Pennies, According to various accounts, the machines were able to predict the sequence of heads/tails entered by an untrained human markedly better than random guess, picking up on a variety of patterns of the past play.

Figure 1: Shannon's Mind Reading Machine, MIT Museum. (Source: http://william-poundstone.com/blog/2015/7/30/how-i-beat-the-mind-reading-machine)

Deviating from the standard approach of time-series analysis, we will (typically) place no probabilistic assumptions on the mechanism generating the sequences. Then how does one make meaningful predictions, or claim guarantees on how well our strategy is doing. To this end, we will consider the objective called regret:

```
for *any* sequence,
       number of mistakes made by forecaster ≤ number of mistakes made by
a benchmark model.
```

For the Penny-Matching game, a simple benchmark can just be do as well as majority of heads vs tails, or more fine-grained statistics, such as predictability of the next outcome based on the last three outcomes. In fact, Shannon's mind reading machine was based on only 8 such states. Which benchmark can one choose? How to develop an efficient algorithm for a given benchmark?

We can contrast the "individual sequence" approach described above with an approach based on stochastic modeling. In the latter, `for any sequence` would be typically replaced with `for most sequences` (or, `with high probability`). However, "for most" is calculated according to the assumed probability model; if the assumption is violated, the result can become significantly weaker. On the other hand, the individual sequence statements are naturally robust to model misspecification. In the age of dynamic and streaming data with a large degree of intricate dependencies, the individual sequence approach appears to be desirable. On the downside, the approach presented in this paper is only focused on prediction rather than inference or estimation. Indeed, estimation requires the assumption that the estimand is there. Our prediction goal, however, is not based on a probabilistic model.

Let us continue with the example of penny matching game. What is a good strategy for the player to beat/or not to loose badly against any opponent. Can we beat Shannon's machine?

Does a strategy as simple as going with majority work, what about a randomized predictor that just uses the frequency of heads or tails so far? What do you think?

2 Experts/Exponential Weights Algorithm

More formally and more generally, we will consider the supervised online learning setting where on each round t from 1 to n, we are first provided the input x_t and are asked to predict the outcome. Our prediction for round t we will define as $f_t(x_t)$ which we make by choosing model $f_t \in \mathcal{F}$. Finally, at the end of the round the ture outcome y_t is revealed. We are interested in minimizing regret given by:

$$\mathrm{Reg}_n := \frac{1}{n}\sum_{t=1}^{n} \ell(f_t(x_t), y_t) - \frac{1}{n}\min_{f\in\mathcal{F}}\sum_{t=1}^{n} \ell(f(x_t), y_t)$$

For the penny matching game, x_t could simply just be equal to t, the time index, we are only trying to predict outcome $y_t \in \{\pm 1\}$. The loss $\ell(f_t(x_t), y_t) = \mathbb{1}_{\{f_t(t)\neq y_t\}}$. Finally, the class of models $\mathcal{F}$ could be as simple as $\mathcal{F} = \{(1, \ldots, 1), (-1, \ldots, -1)\}$, that is the model that either only goes with heads or only goes with tails. First, we claim that diminishing regret against even this simple class of models means that we are not going to loose too badly with any opponent. But the setting itself goes well beyond penny matching problem or even classification problems.

We will provide an algorithm called exponential weights algorithm for this problem below with bound on expected regret (expectation over randomness in the algorithm). The algorithm maintains a distribution over models and on every round picks a random model according to this distribution. This distribution of course favors model that has low cumulative loss so far.
Algorithm : $q_1(f) = 1/|\mathcal{F}|$. Further, each round we update the distribution over experts as,

$$q_{t+1}(f) \propto q_t(f) e^{-\eta \ell(f(x_t), y_t)}$$

Or in other words, $q_{t+1}(f) = \frac{e^{-\eta \sum_{i=1}^{t} \ell(f(x_t), y_t)}}{\sum_{f\in\mathcal{F}} e^{-\eta \sum_{i=1}^{t} \ell(f(x_t), y_t)}}$

Claim 1.

$$\mathbb{E}\left[\mathrm{Reg}_n\right] \leq \sqrt{\frac{2\log|\mathcal{F}|}{n}}$$

Proof. We use the notation $L_t(f) = \sum_{i=1}^{t} \ell(f(x_i), y_i)$. Define $W_0 = |F|$ and define $W_t = \sum_{f\in\mathcal{F}} e^{-\eta L_t(f)}$. Note that

$$\begin{aligned}
\log\left(\frac{W_n}{W_0}\right) &= \log\left(\sum_{f\in\mathcal{F}} e^{-\eta L_n(f)}\right) - \log|\mathcal{F}| \\
&\geq \log\left(\max_{f\in\mathcal{F}} e^{-\eta L_n(f)}\right) - \log|\mathcal{F}| \\
&= -\eta \min_{f\in\mathcal{F}} \sum_{t=1}^{n} \ell(f(x_t), y_t) - \log|\mathcal{F}|
\end{aligned}$$

On the other hand,

$$\begin{aligned}
\log\left(\frac{W_n}{W_0}\right) &= \sum_{t=1}^{n} \log\left(\frac{W_t}{W_{t-1}}\right) = \sum_{t=1}^{n} \log\left(\frac{\sum_{f\in\mathcal{F}} e^{-\eta L_t(f)}}{\sum_{f\in\mathcal{F}} e^{-\eta L_{t-1}(f)}}\right) \\
&= \sum_{t=1}^{n} \log\left(\sum_{f\in\mathcal{F}} \frac{e^{-\eta L_{t-1}(f)}}{\sum_{f\in\mathcal{F}} e^{-\eta L_{t-1}(f)}} e^{-\eta \ell(f(x_t),y_t)}\right) \\
&= \sum_{t=1}^{n} \log\left(\mathbb{E}_{f\sim q_t}\left[e^{-\eta \ell(f(x_t),y_t)}\right]\right) \\
&= \sum_{t=1}^{n} \log\left(\mathbb{E}_{f\sim q_t}\left[e^{-\eta\left(\ell(f(x_t),y_t) - \mathbb{E}_{f\sim q_t}[\ell(f(x_t),y_t)]\right) - \eta \mathbb{E}_{f\sim q_t}[\ell(f(x_t),y_t)]}\right]\right) \\
&= \sum_{t=1}^{n} \log\left(\mathbb{E}_{f\sim q_t}\left[e^{-\eta\left(\ell(f(x_t),y_t) - \mathbb{E}_{f\sim q_t}[\ell(f(x_t),y_t)]\right)}\right] \times e^{-\eta \mathbb{E}_{f\sim q_t}[\ell(f(x_t),y_t)]}\right) \\
&= \sum_{t=1}^{n} \log\left(\mathbb{E}_{f\sim q_t}\left[e^{-\eta\left(\ell(f(x_t),y_t) - \mathbb{E}_{f\sim q_t}[\ell(f(x_t),y_t)]\right)}\right]\right) - \eta \sum_{t=1}^{n} \mathbb{E}_{f\sim q_t}\left[\ell(f(x_t), y_t)\right]
\end{aligned}$$

Thus we conclude that

$$\sum_{t=1}^{n} \mathbb{E}_{f\sim q_t}\left[\ell(f(x_t), y_t)\right] - \min_{f\in\mathcal{F}} \sum_{t=1}^{n} \ell(f(x_t), y_t) \leq \frac{\log|\mathcal{F}|}{\eta} + \frac{1}{\eta} \sum_{t=1}^{n} \log\left(\mathbb{E}_{f\sim q_t}\left[e^{-\eta\left(\ell(f(x_t),y_t) - \mathbb{E}_{f\sim q_t}[\ell(f(x_t),y_t)]\right)}\right]\right)$$

Note that for any zero mean RV X in the range $[-1, 1]$, $\mathbb{E}\left[e^{-\eta X}\right] \leq e^{\eta^2/2}$. Hence,

$$\sum_{t=1}^{n} \mathbb{E}_{f\sim q_t}\left[\ell(f(x_t), y_t)\right] - \min_{f\in\mathcal{F}} \sum_{t=1}^{n} \ell(f(x_t), y_t) \leq \frac{\log|\mathcal{F}|}{\eta} + \frac{n\eta}{2}$$

Picking $\eta = \sqrt{2\log|\mathcal{F}|/n}$ concludes the statement. In fact, using concentration statement called Hoeffding Azuma inequality, we can even conclude that the regret bound even hold with high probability over randomization of the algorithm. □