17-740 Spring 2025
Algorithmic Foundations of Interactive Learning

Lecture 4: Maximum Entropy & FTRL

Lecturer : Drew Bagnell & Steven Wu Scribe: Anupam Nayak and Steven Man

4.1 Recap: Entropy maximization under expectation constraint

We revisit the problem of maximizing entropy under a constraint, as discussed in the last class. The goal is to determine the probability distribution p_i for $i \in \{1, 2, \dots, 6\}$ representing a biased die. The distribution must satisfy the expectation constraint that the expected value of the top face is 4.5 while maximizing the entropy. The mathematical formulation involves finding p_i that maximizes the entropy:

$$\begin{aligned} \text{Maximize:} \quad & H(\mathbf{p}) = -\sum_{i=1}^{6} p_i \log p_i, \\ \text{Subject to:} \quad & \sum_{i=1}^{6} p_i = 1 \quad \text{(normalization constraint)}, \\ & \mathbb{E}[f] = c \quad \text{(expectation constraint)}. \end{aligned}$$

Where $f(i) = i$ is the value on the top face. In order to solve the problem we use the method of Lagrangian multipliers.

$$\mathcal{L}(\mathbf{p}, \gamma, \lambda) = \min_{\gamma, \lambda} \max_{\mathbf{p}} H(\mathbf{p}) - \gamma \left(\sum_{i=1}^{6} p_i - 1 \right) - \lambda \left(\mathbb{E}_{\mathbf{p}}[f] - c \right)$$

Setting $\frac{\partial \mathcal{L}}{\partial p_i} = -\gamma - \lambda f(i) - \log(p_i) - 1 = 0$ we obtain

$$p_i = \frac{e^{-\lambda f(i)}}{e^{\gamma'}} = \frac{e^{-\lambda f(i)}}{Z(\lambda)}$$

where $\gamma' = \gamma + 1 = \log \left(\sum_{i=1}^{6} e^{-\lambda f(i)} \right) = \log Z(\lambda)$. Now we are left with a variable λ. Note that setting $\lambda = 0$ we get the uniform distribution and setting $\lambda = -\infty/\infty$ we get a distribution that has all the probability mass concentrated at $i = 6/1$ respectively. The function $Z(\lambda)$ is usually called the partition function and λ is called the temperature. Substituting

p_i back into the Lagrangian we obtain the expression

$$
\begin{aligned}
&-\mathbb{E}_{\mathbf{p}}[-\lambda f(i) - \log Z(\lambda)] - \lambda\left(\mathbb{E}_{\mathbf{p}}[f] - c\right) && (4.1)\\
&= \mathbb{E}_{\mathbf{p}}[\log Z(\lambda) + \lambda c] && (4.2)
\end{aligned}
$$

Taking the gradient of the above expression wrt λ we obtain

$$\frac{-\sum f e^{-\lambda f}}{Z(\lambda)} + c = -\mathbb{E}_{\mathbf{p}}[f] + c \tag{4.3}$$

This is also called the Max-Ent gradient. The given distribution $p_i = \frac{e^{-\lambda f(i)}}{Z(\lambda)}$ is a member of the exponential family because it can be written in the general form $p(x|\theta) = h(x)\exp(\eta(\theta)\cdot T(x) - A(\theta))$, where $h(x) = 1$, $\eta = -\lambda$, $T(i) = f(i)$, and $A(\theta) = \log Z(\lambda)$[1].

To obtain the value of λ set the Max-Ent gradient to 0 and solve for $\mathbb{E}_{\mathbf{p}}[f] = c$. One can also note the fact that the expected value $\mathbf{E}_{\mathbf{p}}[f]$ is a monotonically decreasing function of λ which allows the use of root finding techniques for efficiently computing λ.

The principle is also widely used in statistical physics. For example, the probability of finding a molecule at height h in the Earth's atmosphere is given by $p(h) = \frac{e^{-\lambda mgh}}{Z}$, where $\lambda = 1/kT$, m is the mass of the molecule, g is the gravitational acceleration, and Z is the partition function. The expectation value of the potential energy, $\mathbb{E}[mgh]$, is constrained to a constant C, which reflects equilibrium conditions.

For a 1D ideal gas, the mean velocity is $\mathbb{E}[v] = 0$, assuming a symmetric velocity distribution. The mean kinetic energy is $\mathbb{E}[v^2] = \frac{2C}{m}$, where C relates to the temperature or average energy. The velocity distribution follows the Maxwell-Boltzmann form: $p(v) = \frac{e^{-\lambda \frac{1}{2}mv^2}}{Z}$, derived using the maximum entropy principle under the constraint of fixed mean energy.

4.2 Online Learning with Expert Advice

4.2.1 Setup

- N experts: $i = 1, \ldots, N$
- For round $t = 1, \ldots, T$:
  1. Algorithm chooses $\mathbf{p}^t = (p_1^t, \ldots, p_N^t)$
  2. Adversary chooses $\mathbf{l}^t = (l_1^t, \ldots, l_N^t)$ after observing $\mathbf{p}^t$
  3. Algorithm incurs loss: $\langle \mathbf{p}^t, \mathbf{l}^t\rangle = \sum_{i=1}^N p_i^t l_i^t$

Assumption 1 (Bounded losses) *The losses l_i^t are upper bounded for all $i \in [N], t \in [T]$*

Follow-the-Leader (FTL):

A natural choice of algorithm here would be the Follow-the-Leader (FTL) algorithm which chooses the best performing expert based on losses observed until time t.

$$\mathbf{p}^t = \arg\min_{\mathbf{p}} \sum_{\tau=1}^{t-1} \langle \mathbf{p}, \mathbf{l}^\tau \rangle.$$

This strategy effectively reduces the problem to selecting a single expert, which can lead to poor performance in adversarial setups. Specifically, the adversary can design a sequence of losses that results in constant regret for the algorithm. Consider the two expert case if the adversary chooses a sequence of losses $[1,0],[0,1],[0,1],[1,0],[1,0],[0,1],[0,1],[1,0],[1,0]\cdots$ alternating between the sequences, FTL switch periodically between experts and will incur a loss that scales linearly in T.

4.2.2 Weighted Majority / Multiplicative Weights

The weighted majority algorithm, also referred to as the multiplicative weights algorithm, is a framework for decision-making with experts. At the start, the algorithm initializes the probability distribution $\mathbf{p}^1$, typically as a uniform distribution across all experts. Over the course of rounds $t = 1, 2, \ldots$, the probabilities are updated iteratively using the rule:

$$p_i^{t+1} \propto p_i^t (1 - \eta l_i^t),$$

where $\eta > 0$ is a learning rate, and l_i^t denotes the loss incurred by expert i during round t. This rule reduces the weight of experts that perform poorly while maintaining higher weights for better-performing ones. Recall that the halving algorithm discussed in lecture 2 is a special case of this algorithm under the assumption that there exists a perfect expert. Any expert that incurs a loss has its weight multiplied by 0 at each time step. $p_i^{t+1} \propto p_i^t(1 - \eta l_i^t)$, can be seen as a first-order Taylor series approximation of the standard exponential weighting update rule:

$$p_i^{t+1} \propto p_i^t \exp(-\eta l_i^t).$$

To address the issue of linear regret in FTL, we introduce the "Follow the Regularized Leader" method. This approach incorporates entropy regularization into the objective as a result of which we get a multiplicative weights update, which discourages overly confident (highly skewed) distributions and promotes diversity. The updated objective function becomes:

$$\mathbf{p}^t = \arg\min_{\mathbf{p}} \sum_{\tau=1}^{t-1} \langle \mathbf{p}, \mathbf{l}^\tau \rangle - \frac{1}{\eta} H(\mathbf{p}),$$

where $H(\mathbf{p}) = \sum_i p_i \log p_i$ is the entropy term. This regularization ensures that the algorithm maintains a spread across multiple experts, mitigating the effects of adversarial losses. Consequently, the final update rule for the weights is given by: $p_i^t \propto \exp\left(-\eta \sum_{\tau=1}^{t-1} l_i^\tau\right)$. This formulation balances exploration (assigning nonzero probability to all experts) and exploitation (favoring experts with lower cumulative losses), achieving robustness and low regret across rounds. The optimization oracle is defined as:

$$\mathcal{O}(\ell^{1:t}) = \arg\min_{\mathbf{p}} \sum_{\tau=1}^{t} \ell^\tau(\mathbf{p}),$$

where $\ell^\tau(\mathbf{p})$ represents the loss function at time τ for a decision vector $\mathbf{p}$.

4.2.3 Analysis

To analyze the Follow-The-Regularized-Leader (FTRL) algorithm, we first introduce an auxiliary algorithm called Be-The-Leader (BTL). We begin by demonstrating that the cumulative loss incurred by the best expert in hindsight is lower-bounded by the loss of the Be-The-Leader algorithm in lemma 2. Next, we incorporate entropy regularization, denoted as l_0, which represents the loss incurred by the algorithm at time step 0. We define BTL in relation to this newly introduced cumulative loss. Finally, we establish an upper bound on the difference between the losses incurred by FTRL and BTL, thereby completing the proof.

Under the Be-The-Leader (BTL) algorithm, at each time step t, we determine the probability vector p_t by selecting the vector that minimizes the cumulative loss observed up to and including time step t.

$$p_t = \mathcal{O}(\ell^{1:t})$$

However, it is important to note that this approach is not a practical algorithm, as it relies on prior knowledge of the loss that will be incurred at time t.

Lemma 2 (Be the Leader)

$$\sum_{t=1}^{T} \underbrace{\langle \mathcal{O}(\ell^{1:t}), \ell^t \rangle}_{\texttt{Be the leader}} \leq \underbrace{\qquad \langle \mathcal{O}(\ell^{1:T}), \sum_{t=1}^{T} \ell^t \rangle \qquad}_{\textit{Loss of the best expert in hindsight}}.$$

Here, $\mathcal{O}(\ell^{1:t})$ corresponds to the decisions made incrementally up to time t, while $\mathcal{O}(\ell^{1:T})$ represents the expert in hindsight, based on all cumulative losses.

Proof: At each time t, `Be the leader` selects the best expert based on the cumulative losses up to time $t+1$. It is important to note that the algorithm also makes its decision using ℓ^t, even though this value has not yet been observed. The proof follows by induction For the base case $t=2$, we have:

$$\langle \mathcal{O}(\ell^1), \ell^1 \rangle + \langle \mathcal{O}(\ell^{1:2}), \ell^2 \rangle \leq \langle \mathcal{O}(\ell^{1:2}), \ell^1 \rangle + \langle \mathcal{O}(\ell^{1:2}), \ell^2 \rangle$$

Now, assume that the inequality holds for all rounds up to $t-1$. That is,

$$\sum_{\tau=1}^{t} \langle \mathcal{O}(\ell^{1:\tau}), \ell^\tau \rangle \leq \left\langle \mathcal{O}(\ell^{1:t}), \sum_{\tau=1}^{t} \ell^\tau \right\rangle.$$

Now for $t+1$ we have

$$\begin{aligned}
\sum_{\tau=1}^{t+1} \langle \mathcal{O}(\ell^{1:\tau}), \ell^\tau \rangle &= \sum_{\tau=1}^{t} \langle \mathcal{O}(\ell^{1:\tau}), \ell^\tau \rangle + \langle \mathcal{O}(\ell^{1:t+1}), \ell^{t+1} \rangle \\
&\leq \left\langle \mathcal{O}(\ell^{1:t}), \sum_{\tau=1}^{t} \ell^\tau \right\rangle + \langle \mathcal{O}(\ell^{1:t+1}), \ell^{t+1} \rangle \\
&\leq \left\langle \mathcal{O}(\ell^{1:t+1}), \sum_{\tau=1}^{t} \ell^\tau \right\rangle + \langle \mathcal{O}(\ell^{1:t+1}), \ell^{t+1} \rangle = \left\langle \mathcal{O}(\ell^{1:t+1}), \sum_{\tau=1}^{t+1} \ell^{t+1} \right\rangle
\end{aligned}$$

Thus, by induction, the inequality holds for all t, this completes the proof ■

Let l_0 represent the loss at step zero, defined as $l_0 = -\frac{1}{\eta} H(\mathbf{p})$, where $H(\mathbf{p})$ denotes the entropy of the probability distribution $\mathbf{p}$. Let $l(\mathbf{p})$ denote the expected loss incurred while using the distribution $\mathbf{p}$. Using lemma 2,the cumulative regret is bounded as follows:

$$\begin{aligned}
\sum_{t=0}^{T} \big(l_t(\mathbf{p}^t) - l_t(\mathbf{p}^*) \big) &\leq \sum_{t=0}^{T} l_t(\mathbf{p}^t) - l_t(\mathcal{O}(\ell^{0:t})) \\
&= \sum_{t=0}^{T} l_t(\mathbf{p}^t) - l_t(\mathbf{p}^{t+1})
\end{aligned}$$

where $\mathcal{O}(\ell^{0:t})$ refers to the leader with entropy regularization. Now, incorporating the loss at step zero:

$$\text{Regret}(\mathbf{p}^{1:T}) + l_0(\mathbf{p}^0) - l_0(\mathbf{p}^*) \leq l_0(\mathbf{p}^0) - l_0(\mathbf{p}^1) + \sum_{t=1}^{T} l_t(\mathbf{p}^t) - l_t(\mathbf{p}^{t+1}).$$

Thus, the regret becomes:

$$\text{Regret}(\mathbf{p}^{1:T}) \leq \underbrace{l_0(\mathbf{p}^*) - l_0(\mathbf{p}^1)}_{\leq \frac{\log(N)}{N}} + \underbrace{\sum_{t=1}^{T} l_t(\mathbf{p}^t) - l_t(\mathbf{p}^{t+1})}_{\text{Stability term}}. \tag{4.4}$$

Lemma 3 (Stability)

$$\sum_{t=1}^{T} l_t(\mathbf{p}^t) - l_t(\mathbf{p}^{t+1}) \leq 2\eta \sum_{t=1}^{T} l_t(\mathbf{p}^t) \leq 2\eta T.$$

Proof: We have

$$\begin{aligned} l_t(\mathbf{p}^t) - l_t(\mathbf{p}^{t+1}) &= \left\langle \mathbf{l}^t, \mathbf{p}^t - \mathbf{p}^{t+1} \right\rangle && (4.5) \\ &= \sum_{i=1}^{N} l_i^t \left(p_i^t - p_i^t \frac{e^{-\eta l_i^t}}{\sum_{j=1}^{N} p_j^t e^{-\eta l_j^t}} \right) && (4.6) \\ &\leq \sum_{i=1}^{N} l_i^t p_i^t \left(1 - e^{-\eta l_i^t} \right) && (4.7) \\ &\leq \eta \sum_{i=1}^{N} l_i^t p_i^t = \eta \langle \mathbf{l}^t, \mathbf{p}^t \rangle \leq \eta && (4.8) \end{aligned}$$

■

Here equation (4.7) follows from the non-negativity of the loss function and equation (4.8) follows from assumption 1. By setting $\eta = \sqrt{\frac{\log(N)}{T}}$ in (4.4) and using lemma 3, we achieve a regret bound of:

$$\text{Regret} = \mathcal{O}(\sqrt{T \log(N)}).$$

This result highlights the efficiency of the algorithm, demonstrating that the regret grows sublinearly with the number of rounds T, while also scaling logarithmically with the number of experts N.

References

[1] Exponential family — wikipedia, the free encyclopedia, 2025.