IIIS 2017 Spring: ATCS - Selected Topics in Optimization
Lecture date: March 5th, 2018
Instructor: Jian Li **Scribe: Zihan Zhang**

1 Multiplicative Weight Algorithm

Coin Guess Problem (special case of expert problem): Given any sequence of binary value of length T (1 represents the face and 0 for the tail), at time step $t \leq T$, expert i gives prediction $c_i^t \in \{0, 1\}$. Then the algorithm give a distribution $p^t \in \Delta([n])$ based on the history and follow the prediction of expert i with probability of p_i^t. And then we observe the actual result c^t. The regret is define as:

$$Reg_T = \max_{i \in [n]} \sum_{t=1}^{T} I[c_i^t = c^t] - \sum_{t=1}^{T} \sum_{i=1}^{n} p_i^t I[c_i^t = c^t]$$

Remark: Here we choose to compare with the best expert because in the worst case, the adversary could choose c^t after we make the decision p^t such that the loss is at least one half.

This is a classical online optimization problem. Denote the loss $m_i^t = I[c_i^t \neq c^t]$ the MW algorithm is given as following:

Algorithm 1: MW for Coin Guess

Initially, let $w_i^1 = 0$ for all $i \in [n]$, $\epsilon < 1/2$;
for $= 1,2,\ldots,T$ **do**
  $w^t = \sum_{i \in [n]} w_i^t$;
  $p_i^t = w_i^t / w^t$;
  choose expert with probability of p_i^t;
  observe loss m_i^t;
  update $w_i^{t+1} = w_i^t (1 - \epsilon m_i^t)$

Theorem1.

$$\mathbb{E}[Reg_T] \leq \epsilon T + \frac{\log N}{\epsilon}$$

Proof.

Denote the loss of Algorithm 1 at time t as m_A^t

$$m_A^t = \sum_i p_i^t m_i^t$$

First, we observe that

$$\begin{aligned} w^{t+1} &= \sum_{i\in[n]} w_i^{t+1} = \sum_{i\in[n]} w_i^t(1-\epsilon m_i^t) \\ &= w^t - w^t \sum_{i\in[n]} \frac{w_i^t}{w^t}\epsilon m_i^t \\ &= w^t - w^t \sum_{i\in[n]} p_i^t \epsilon m_i^t \\ &= w^t(1-\epsilon E[m_A^t]) \end{aligned}$$

Then we have

$$w^{T+1} = w^1 \Pi_{t=1}^T (1-\epsilon E[m_A^t]) \geq w_i^{T+1} = \Pi_{t=1}^T (1-\epsilon m_i^t)$$

for any $i \in [n]$, and

$$\log N - \epsilon \sum_{t=1}^T m_A^t \geq \log N + \sum_{t=1}^T \log(1-\epsilon m_A^t) \geq \sum_{t=1}^T \log(1-\epsilon m_i^t) \geq -\sum_{t=1}^T \epsilon m_i^t + \epsilon^2 m_i^t$$

The following fact is used here. When $\epsilon \in [0, 1/2]$ and $m_i^t \in \{0,1\}$, we have inequality $-\epsilon m_A^t \geq \log(1-\epsilon m_A^t)$, $\log(1-\epsilon m_i^t) \geq -\epsilon m_i^t - \epsilon^2 m_i^t$.

We have

$$\sum_{t=1}^T m_A^T \leq \sum_{t=1}^T m_i^t + \log N/\epsilon + \epsilon \sum_{t=1}^T (m_i^t)^2 \leq \sum_{t=1}^T m_i^t + \log N/\epsilon + \epsilon T$$

For any $i \in [n]$. Let $\epsilon = \sqrt{\log N/T}$, we have

$$Reg_T = \sum_{t=1}^T m_A^T - \max_i \sum_{t=1}^T m_i^t \leq \sqrt{T \log N}$$

Hence, the regret is bounded by $O(\sqrt{TlogN})$.

2 Applications in Zero-Sum Game

Zero-sum game: Here we consider the problem $\min_x \max_y x^T My$ where $M \in R^{m\times n}$ (for simplicity, we assume that entries of M are either 0 or 1), $x \in \Delta^m = \{x \in R^m | x_i \geq 0, \forall i \in [m], \sum_{i=1}^m x_i = 1\}$ and $y \in \Delta^n$ (the capacity region is often omitted when clear). By Von Neumann's Minimax theorem, we have

$$\lambda^* = \min_x \max_y x^T My = \max_y \min_x x^T My$$

Now we want get a mixed δ-optimal strategy, i.e. find a x_δ and y_δ, such that

$$\max_y x_\delta^T My \leq \lambda^* + \delta$$

$$\min_x x^T My_\delta \geq \lambda^* - \delta$$

We only give an algorithm for the first inequality, i.e. find a δ-optimal strategy for the row player. Regarding each row as an expert, then for any adversarial sequence $\{j_t\}_{t=1}^T$, by theorem1 we could find a sequence of strategy $\{x_t\}_{t=1}^T$, such that

$$\sum_{t=1}^{T} x_t M e_{j_t} \leq \min_{i\in[m]} \sum_{t=1}^{T} e_i^T M e_{j_t} + 2\sqrt{\log(m)T} \leq T\lambda^* + 2\sqrt{\log(m)T}$$

By choosing e_{j_t} as the best responding strategy for x_t, we get

$$T\lambda^* \leq \sum_{t=1}^{T} x_t M e_{j_t} \leq T\lambda^* + 2\sqrt{\log(m)T}$$

$$\lambda^* \leq \frac{1}{T}\sum_{t=1}^{T} x_t M e_{j_t} \leq \lambda^* + 2\sqrt{\frac{\log(m)}{T}}$$

Let $\tau = arg\min_t x_t M e_{j_t}$, by choosing $T > \frac{4\log(m)}{\delta^2}$, then

$$x_\tau M e_{j_\tau} \leq \lambda^* + \delta$$

.

3 Online Gradient Descent

Let's now think about another expansion of experts problem. Suppose there is a convex set $K \subset \mathbb{R}^n$ and a family of convex functions $\mathcal{F}$ mapping K to $\mathbb{R}$. At time step t, the agent select $x_t \in K$, then the environment provides a convex function $f_t \in \mathcal{F}$. The regret is define as

$$Reg_T = \sum_{t=1}^{T} f_t(x_t) - \min_{x\in K} \sum_{t=1}^{T} f_t(x)$$

In this case, there are uncountable experts with each $x \in K$ corresponding one expert. However, convexity of f_t makes the problem tractable.

Algorithm 2: Gradient Descent

Initially, select $x_1 \in K$, $\eta_t = \sqrt{1/T}$;
for $t = 1,2,\dots,T$ **do**
  $x_{t+1} = Proj_K(x_t - \eta_t \nabla f_t(x_t))$
Return x_{T+1};

Theorem2 Assume $D(K) = \max_{x,y\in K} ||x-y|| = D$ and $||\nabla f(x)|| \leq N, \forall f \in \mathcal{F}$ and $x \in K$, we have following bounds for algorithm2

$$Reg_T \leq 2ND\sqrt{T}$$

Proof

For any $x \in K$, by convexity of f_t we have $f_t(x_t) - f_t(x) \le \nabla f_t(x_t)^T(x_t - x)$. Denote $g_t = \nabla f_t(x_t)$, $y_{t+1} = x_t - \eta_t g_t$, $x_{t+1} = Proj(y_{t+1})$.

$$\begin{aligned}
||x_{t+1} - x^*||^2 \le ||y_{t+1} - x^*||^2 &= (x_t - x^* - \eta_t g_t)^2 \\
&= ||x_t - x^*||^2 - 2\eta_t g_t(x_t - x^*) + \eta_t^2||g_t||^2
\end{aligned}$$

$$(x_t - x^*)g_t \le \frac{1}{2\eta_t}(||x_t - x^*||^2 - ||x_{t+1} - x^*||^2) + \frac{\eta_t^2}{2}||g_t||^2$$

$$\begin{aligned}
Reg_T &= \sum_{t=1}^{T}(x_t - x^*)g_t \\
&\le \frac{1}{2\eta_1}||x_1 - x^*||^2 - \frac{1}{2\eta_t}||x_T - x^*||^2 + \frac{1}{2}\sum_{t=2}^{T}(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}})||x_t - x^*||^2 + \frac{1}{2}\sum_{t=1}^{T}N^2\eta_t \\
&\le D^2(\frac{1}{2\eta_1} + \sum_{t=2}^{T}(\frac{1}{2\eta_t} - \frac{1}{2\eta_{t-1}})) + \frac{1}{2}\sum_{t=1}^{T}N^2\eta_t \\
&= \frac{D^2}{2\eta_T} + \frac{1}{2}N^2\sum_{t=1}^{T}\eta_t
\end{aligned}$$

When η_t is fixed and $\eta_t = \sqrt{\frac{D^2}{TN^2}}$, we have regret bound $Reg_T \le 2ND\sqrt{T}$.

When $\eta_t = \sqrt{\frac{D^2}{tN^2}}$, we have regret bound $Reg_T \le \sqrt{2}ND\sqrt{T}$.

4 Universal Portfolio Algorithm

In this section we'll talk about the stock investment problem. Suppose there are m stocks in the market, and in the t-th day, the return of stock i is $x_{it} > 0$ which we don't know at the start of the t-th day. Now we should make investment in the beginning of each day, the problem is: how to make profit as much as we can?

Constant rebalanced portfolio (CRP): The strategy is quite easy. Given $b \in \Delta_n$, at the start of each day, we buy stock i with b_i of all our wealth. Then at the end of T-th day, the total wealth is $S_T(b, x_{1:T}) = \Pi_{t=1}^{T}(\sum_{i=1}^{n} b_i x_{it})$. Denote μ_n be the uniform distribution on Δ_n, we have following algorithm:

Algorithm 3: Universal Portfolio (UP)

1 Initially, $S_0(b, x) = 1$ for all $b \in \Delta_n$ and x;
2 **for** $t = 1, 2, \ldots, T$ **do**
3   $\hat{b}_t = \frac{\int_{\Delta_n} bS_{t-1}(b, x_{1:t-1})d\mu(b)}{\int_{\Delta_n} S_{t-1}(b, x_{1:t-1})d\mu(b)}$;
4   make investment following $\hat{b}_t$

Performance of Universal Portfolio

Here the target is the best CRP strategy. We have following result:

Theorem3. Denote the total wealth of Universal Portfolio at the end of T-th day is $S_{UP}(T, x_{1:T})$, then for any $b^* \in \Delta_n$ and $x_{1:T}$,

$$\frac{S_{UP}(T, x_{1:T})}{S_T(b^*, x_{1:T})} \geq \frac{e^{-1}}{(T+1)^{n-1}}$$

Proof. Note that

$$\frac{S_{UP}(t, x_{1;t})}{S_{UP}(t-1, x_{1:t-1})} = \frac{\int_{\Delta_n} b^T x_t S_{t-1}(b, x_{1:t-1}) d\mu(b)}{\int_{\Delta_n} S_{t-1}(b, x_{1:t-1}) d\mu(b)} = \frac{\int_{\Delta_n} S_t(b, x_{1:t}) d\mu(b)}{\int_{\Delta_n} S_{t-1}(b, x_{1:t-1}) d\mu(b)}$$

by induction and $S_{UP}(0) = 1$, we have

$$S_{UP}(t, x_{1:t}) = \int_{\Delta_n} S_t(b, x_{1:t}) d\mu(b)$$

consider the region $B(b^*, \alpha) = (1-\alpha)b^* + \alpha\Delta_n = \{b | b = (1-\alpha)b^* + \alpha z, z \in \Delta_n\}$. It's clear that if $b_1 \geq b_2$, then $S_t(b_1, x_{1:t}) \geq S_t(b_2, x_{1:t})$ if we expand the definition of S into R_+^n, thus for any $b \in B(b^*, \alpha)$, $S_t(b, x_{1:t}) \geq (1-\alpha)^t S_t(b^*, x_{1:t})$. In the other side, it's easy to get $\mu(B(b^*, \alpha)) = \alpha^{n-1}$, so we have

$$S_{UP}(T, x_{1:T}) = \int_{\Delta_n} S_T(b, x_{1:T}) d\mu(b) \geq (1-\alpha)^T \alpha^{n-1} S_T(b^*, x_{1:T})$$

by setting $\alpha = \frac{1}{T+1}$, the conclusion follows.

References

[1] Arora S, Hazan E, Kale S. The Multiplicative Weights Update Method: a Meta-Algorithm and Applications[J]. Theory of Computing, 2012, 8(1): 121-164.

[2] Blum A, Kalai A. Universal portfolios with and without transaction costs[J]. Machine Learning, 1999, 35(3): 193-205.