Online Convex Optimization

Presented by: Sharan Vaswani

Introduction

- **Online learning:** Make a sequence of accurate predictions given knowledge of the correct answer to previous prediction tasks and possibly additional available information.

- **Applications:** Online advertisement placement, web ranking, spam filtering, online shortest paths, portfolio selection, recommender systems

Notation

$\mathbf{x}_t$ Decision/point chosen on timestamp t

$\mathcal{K}$ Bounded convex, decision set

$f_t \in \mathcal{F} : \mathcal{K} \mapsto \mathbb{R}$ Bounded convex function available at timestamp t

T Number of iterations

$\mathcal{A}$ Online algorithm

Protocol

For t = 1:T

1. Learner chooses $\mathbf{x}_t$
2. Environment / Adversary chooses f_t
3. Learner suffers loss $f_t(\mathbf{x}_t)$

Aim: To minimize cumulative loss across rounds measured using regret

$$\mathrm{regret}_T(\mathcal{A}) = \sup_{\{f_1,\ldots,f_t\}\subseteq\mathcal{F}} \left\{ \sum_{t=1}^{T} f_t(\mathbf{x}_t) - \min_{\mathbf{x}\in\mathcal{K}} \sum_{t=1}^{T} f_t(\mathbf{x}) \right\}$$

Chosen by our online algorithm

Chosen by the offline algorithm

Learning from expert advice

- Need to make a decision (eg: whether to invest or not), helped by N experts who make predictions
- Decide which expert to follow at each step
- Receive feedback for each expert

Learning from expert advice

Algorithm 1 Hedge

Initialize: $\forall i \in [N],\ W_1(i) = 1$
for $t = 1$ to T **do**
    Pick $i_t \sim_R W_t$, i.e., $i_t = i$ with probability $p_t(i) = \frac{W_t(i)}{\sum_j W_t(j)}$
    Observe loss $\ell_t(i_t)$.
    Update weights $W_{t+1}(i) = W_t(i) e^{-\varepsilon \ell_t(i)}$
end for

Theorem 1.5. Let ℓ_t^2 denote the N-dimensional vector of square losses, i.e., $\ell_t^2(i) = \ell_t(i)^2$, and let $\varepsilon < \frac{1}{2}$. The Hedge algorithm satisfies for any expert $i^* \in [N]$:

$$\sum_{t=1}^{T} p_t^\top \ell_t \leq \sum_{t=1}^{T} f_t(i^*) + \varepsilon \sum_t p_t^\top \ell_t^2 + \frac{\ln N}{\varepsilon}$$

Asymptotic Regret Bounds

Theorem 3.2. Any algorithm for online convex optimization incurs $\Omega(DG\sqrt{T})$ regret in the worst case. This is true even if the cost functions are generated from a fixed stationary distribution.

- Achieved by first order methods like online gradient descent and second order methods like online newton

α-strongly convex	β-smooth
$\frac{1}{\alpha}\log T$ [1]	$\sqrt{T}$ [2]

- Smoothness doesn't help (unlike in offline gradient descent), strong convexity does

1 - Hazan, 2007; 2 - Zinkevich, 2003

Follow the leader

Idea: Use loss information in previous iterations to choose point in current iteration.

Follow-The-Leader (FTL)

$$\forall t, \quad \mathbf{w}_t = \operatorname*{argmin}_{\mathbf{w} \in S} \sum_{i=1}^{t-1} f_i(\mathbf{w}) \quad \text{(break ties arbitrarily)}$$

Special case: Obtains log(T) regret with quadratic functions (shooting game)
Problem: Predictions are not stable and may fluctuate drastically

Example 2.2 (Failure of FTL). Let $S = [-1, 1] \subset \mathbb{R}$ and consider the sequence of linear functions such that $f_t(w) = z_t w$ where

$$z_t = \begin{cases} -0.5 & \text{if } t = 1 \\ 1 & \text{if } t \text{ is even} \\ -1 & \text{if } t > 1 \ \wedge \ t \text{ is odd} \end{cases}$$

Then, the predictions of FTL will be to set $w_t = 1$ for t odd and $w_t = -1$ for t even. The cumulative loss of the FTL algorithm will therefore be T while the cumulative loss of the fixed solution $u = 0 \in S$ is 0. Thus, the regret of FTL is T !

Follow the regularized leader

- **Idea:** Add regularization to make prediction more stable.

Algorithm 10 Regularized Follow The Leader

Input: $\eta > 0$, strongly convex regularization function R, and a convex compact set $\mathcal{K}$.
Let $\mathbf{x}_1 = \arg\min_{\mathbf{x}\in\mathcal{K}} \{R(\mathbf{x})\}$.
for $t = 1$ to T **do**
  Predict $\mathbf{x}_t$.
  Observe the payoff function f_t and let $\nabla_t = \nabla f_t(\mathbf{x}_t)$.
  Update

$$\mathbf{x}_{t+1} = \arg\min_{\mathbf{x}\in\mathcal{K}} \underbrace{\left\{\eta \sum_{s=1}^{t} \nabla_s^\top \mathbf{x} + R(\mathbf{x})\right\}}_{\Phi_t(\mathbf{x})} \tag{5.1}$$

end for

Follow the regularized leader

Theorem 5.1. The RFTL algorithm 10 attains for every $\mathbf{u} \in \mathcal{K}$ the following bound on the regret:

$$\text{regret}_T \leq \sum_{t=1}^{T} \nabla_t^\top (\mathbf{x}_t - \mathbf{u}) \leq 2\eta \sum_t \|\nabla_t\|_t^{*2} + \frac{1}{\eta} D_R$$

Local norm of gradient at t
Can be upper bounded

Diameter of the decision set relative to R

Special Cases:

- **Online Gradient Descent** with L2 regularization i.e. $R(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|_2^2$
- **Hedge** with negative entropy regularization i.e. $R(\mathbf{x}) = \mathbf{x} \log \mathbf{x}$

Online gradient descent

Algorithm 6 ONLINE GRADIENT DESCENT

Input: convex set $\mathcal{K}$, T, $\mathbf{x}_1 \in \mathcal{K}$, step sizes $\{\eta_t\}$
for $t = 1$ to T **do**
  Play $\mathbf{x}_t$ and observe cost $f_t(\mathbf{x}_t)$.
  Update and project:

$$\mathbf{y}_{t+1} = \mathbf{x}_t - \eta_t \nabla f_t(\mathbf{x}_t)$$
$$\mathbf{x}_{t+1} = \prod_{\mathcal{K}} [\mathbf{y}_{t+1}]$$

end for

Decrease as 1 / sqrt(t)

Projection

Online gradient descent

Theorem 3.1. ONLINE GRADIENT DESCENT with step sizes $\{\eta_t = \frac{D}{G\sqrt{t}}, t = 1, ..., T\}$ guarantees the following for all $T \geq 1$.

$$\text{regret}_T = \sum_{t=1}^{T} f_t(\mathbf{x}_t) - \min_{\mathbf{x}^* \in \mathcal{K}} \sum_{t=1}^{T} f_t(\mathbf{x}^*) \leq 3GD\sqrt{T}$$

Theorem 3.3. For α-strongly convex loss functions, ONLINE GRADIENT DESCENT with step sizes $\eta_t = \frac{1}{\alpha t}$ achieves the following guarantee, for all $T \geq 1$

$$\text{regret}_T \leq \frac{G^2}{2\alpha}(1 + \log T).$$

Online to Batch reduction

Suppose we have a batch of data, can we OGD to look one point at a time and get a rate as good as SGD ?

Algorithm 7 Stochastic Gradient Descent

Input: f,$\mathcal{K}$, T, $\mathbf{x}_1 \in \mathcal{K}$, step sizes $\{\eta_t\}$
for $t = 1$ to T **do**
    Let $\tilde{\nabla}_t \leftarrow \mathcal{O}(\mathbf{x}_t)$ and define: $f_t(x) \triangleq \langle \tilde{\nabla}_t, x \rangle$
    Update and project:

$$\mathbf{y}_{t+1} = \mathbf{x}_t - \eta_t \tilde{\nabla}_t$$

$$\mathbf{x}_{t+1} = \prod_{\mathcal{K}}[\mathbf{y}_{t+1}]$$

end for
return $\bar{\mathbf{x}}_T \triangleq \frac{1}{T}\sum_{t=1}^{t} \mathbf{x}_t$

Use linearized functions with noisy gradient as input and return the average point over the iterations

Online to Batch reduction

Regret bound:

$$\mathbf{E}[f(\bar{\mathbf{x}}_T)] \leq \min_{\mathbf{x}^* \in \mathcal{K}} f(\mathbf{x}^*) + \frac{3GD}{\sqrt{T}}$$

- For general convex function, achieves 1/sqrt(T) rate
- For strongly convex functions, online to batch conversion can give a log(T)/T rate later improved to 1/T (Hazan’14)

ADAGRAD

- Different regularizations in RTFL lead to different algorithms. Can we learn an optimal regularization to use ?

Algorithm 16 AdaGrad

1: Input: parameters $\eta, \delta > 0, \mathbf{x}_1 \in \mathcal{K}$.
2: Initialize: $\mathbf{S}_0 = \mathbf{G}_0 = \delta I$,
3: **for** $t = 1$ to T **do**
4: Predict $\mathbf{x}_t$, suffer loss $f_t(\mathbf{x}_t)$.
5: Update: $\mathbf{S}_t \leftarrow \mathbf{S}_{t-1} + \nabla_t \nabla_t^\top$, $\mathbf{G}_t = {\mathbf{S}_t}^{1/2}$

$$\mathbf{y}_{t+1} \leftarrow \mathbf{x}_t - \eta \mathbf{G}_t^{-1} \nabla_t$$

$$\mathbf{x}_{t+1} \leftarrow \arg\min_{\mathbf{x} \in \mathcal{K}} \|\mathbf{y}_{t+1} - \mathbf{x}\|_{\mathbf{G}_t}^2$$

6: **end for**

ADAGRAD

$$\forall \mathbf{x} \in \mathcal{K} \ . \ \nabla^2 R(\mathbf{x}) = \nabla^2 \in \mathcal{H} \triangleq \{X \in \mathbb{R}^{d \times d} \ , \ \mathbf{Tr}(X) \leq 1 \ , \ X \succeq 0\}$$

Theorem 5.9. Let $\{\mathbf{x}_t\}$ be defined by Algorithm 16 with parameters $\delta = 1, \eta = \frac{1}{D}$, where

$$D = \max_{\mathbf{u} \in \mathcal{K}} \max_{H \in \mathcal{H}} \|\mathbf{u} - \mathbf{x}_1\|_H$$

Then for for any $\mathbf{x}^* \in \mathcal{K}$,

$$\text{regret}_T(\text{AdaGrad}) \leq 2D \cdot \sqrt{\min_{H \in \mathcal{H}} \sum_t \|\nabla_t\|_H^{*2}} \tag{5.6}$$

$$\text{regret}(\text{AdaGrad}) \leq 2D\mathbf{Tr}(\mathbf{G}_T)$$

Variants

- Follow the perturbed leader (FPL) – form of randomized regularization
- Bandit feedback (observe loss only for the selected point) – can be analyzed using OCO by estimating the gradient at each point
- Use Frank Wolfe when projection is expensive