Online Learning: Sufficient Statistics and the Burkholder Method

Dylan Foster, Alexander Rakhlin, and Karthik Sridharan

djfoster@cs.cornell.edu, rakhlin@mit.edu, sridharan@cs.cornell.edu

Cornell University

Overview

Online supervised learning

Protocol:
For $t = 1$ to n:

Receive input instance $x_t \in \mathcal{X}$.

Learner picks (possibly randomized) prediction $\hat{y}_t \in \mathbb{R}$.

Receive outcome $y_t \in \mathcal{Y} = \{\pm 1\}$.

Learner suffers loss $\ell(\hat{y}_t, y_t) = -\hat{y}_t \cdot y_t$
(in general, convex and 1-Lipschitz wrt $\hat{y}$).

End

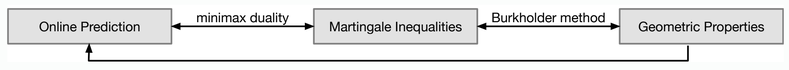
Goal:

$$\forall x_{1:n}, y_{1:n} \quad \sum_{t=1}^{n} \ell(\hat{y}_t, y_t) \le \underbrace{\phi(x_1, y_1, \ldots, x_n, y_n)}_{\text{Small if data is "nice"}}$$

Typically, ϕ based on **hypothesis class** $\mathcal{F} \subset [-1, +1]^{\mathcal{X}}$:

$$\sum_{t=1}^{n} \ell(\hat{y}_t, y_t) \le \underbrace{\inf_{f \in \mathcal{F}} \sum_{t=1}^{n} \ell(f(x_t), y_t)}_{\text{relationship btw. } x/y} + \underbrace{\mathcal{C}_n(\mathcal{F}; x_1, \ldots, x_n)}_{\text{complexity of } \mathcal{F} \text{ on } x_{1:n}}$$

Contributions



1) Alg. design tool: The Burkholder method

- Generic technique to go from desired regret inequality to concrete algorithm.
- Based on connection to martingale inequalities.

2) Sufficient statistics for online learning

- If regret can be (approximately) expressed in terms of certain "sufficient statistics", Burkholder algorithm only needs to store sufficient statistics in memory.

Martingale Inequalities and Online Learning

- Let $\epsilon_1, \ldots, \epsilon_n$ be coin flips and let $\boldsymbol{x}_t = \boldsymbol{x}_t(\epsilon_1, \ldots, \epsilon_{t-1})$ be an arbitrary **predictable process**.
- $(\epsilon_t \boldsymbol{x}_t(\epsilon))_{t \le n}$ is a **martingale difference sequence**.

Suppose algorithm $(\hat{y}_t)_{t \le n}$ guarantees

$$\forall x_{1:n}, y_{1:n} \quad \sum_{t=1}^{n} -\hat{y}_t \cdot y_t \le \phi(x_1, y_1, \ldots, x_n, y_n).$$

In particular, for any draw of ϵ,

$$\sum_{t=1}^{n} -\hat{y}_t \cdot \epsilon_t \le \phi(\boldsymbol{x}_1, \epsilon_1, \ldots, \boldsymbol{x}_n, \epsilon_n)$$

Taking expectation gives **martingale inequality**:

$$\mathbb{E}_\epsilon[\phi(\boldsymbol{x}_1, \epsilon_1, \ldots, \boldsymbol{x}_n, \epsilon_n)] \ge 0.$$

$\implies$ Inequality is necessary for existence of algorithm.
(also sufficient via minimax theorem)

Burkholder Method for Martingale Inequalities

Generic martingale inequality:

$$\forall \boldsymbol{x}, n \quad \mathbb{E}_\epsilon[V(\epsilon_1 \boldsymbol{x}_1, \ldots, \epsilon_n \boldsymbol{x}_n)] \le 0.$$

Theorem (Burkholder)

Inequality holds if and only if there exists Burkholder function $\boldsymbol{U} : \bigcup_n \mathcal{X}^n \to \mathbb{R}$ such that

1^o $V(x_1, \ldots, x_n) \le \boldsymbol{U}(x_1, \ldots, x_n) \quad \forall n$

2^o $\boldsymbol{U}(0) \le 0.$

3^o $\mathbb{E}_\epsilon \boldsymbol{U}(x_1, \ldots, x_n, \epsilon x_{n+1}) \le \boldsymbol{U}(x_1, \ldots, x_n) \quad \forall n, x_1, \ldots, x_n, x_{n+1},$ restricted concavity

$\boldsymbol{U}$: "Extremal function" for martingale inequality.

Proof

Burkholder function $\implies$ martingale inequality

$$\begin{aligned}\mathbb{E}_\epsilon[V(\epsilon_1 \boldsymbol{x}_1, \ldots, \epsilon_n \boldsymbol{x}_n)] &\le \mathbb{E}_\epsilon[\boldsymbol{U}(\epsilon_1 \boldsymbol{x}_1, \ldots, \epsilon_n \boldsymbol{x}_n)] \\ &\le \mathbb{E}_\epsilon[\boldsymbol{U}(\epsilon_1 \boldsymbol{x}_1, \ldots, \epsilon_{n-1} \boldsymbol{x}_{n-1})] \\ &\vdots \\ &\le \boldsymbol{U}(0).\end{aligned}$$

martingale inequality $\implies$ Burkholder function
Define

$$\boldsymbol{U}^\star(x_1, \ldots, x_t) = \sup_{n \ge t, \boldsymbol{x}_{t+1:n}} \mathbb{E}_\epsilon[V(x_1, \ldots, x_t, \epsilon_{t+1} \boldsymbol{x}_{t+1}, \ldots, \epsilon_n \boldsymbol{x}_n)].$$

1^o and 2^o are trivial. For 3^o, fix any x_{t+1}.

$$\begin{aligned}&\boldsymbol{U}^\star(x_1, \ldots, x_t) \\ &\ge \sup_{n \ge t+2, \boldsymbol{x}_{t+1:n}} \mathbb{E}_\epsilon[V(x_1, \ldots, x_t, \epsilon_{t+1} x_{t+1}, \epsilon_{t+2} \boldsymbol{x}_{t+2}, \ldots, \epsilon_n \boldsymbol{x}_n)] \\ &= \mathbb{E}_\epsilon[\boldsymbol{U}^\star(x_1, \ldots, x_t, \epsilon_{t+1} x_{t+1})].\end{aligned}$$

($\boldsymbol{U}^\star$ is the smallest Burkholder function for V)

Burkholder Method and Online Learning

Back to online learning.
Recall We want $\sum_{t=1}^{n} -\hat{y}_t \cdot y_t \le \phi(x_1, y_1, \ldots, x_n, y_n)$.

- Let $V(x_1 y_1, \ldots, x_n y_n) =: -\phi(x_1, y_1, \ldots, x_n, y_n)$
- Then $\mathbb{E}_\epsilon[V(\epsilon_1 \boldsymbol{x}_1, \ldots, \epsilon_n \boldsymbol{x}_n)] \le 0$ necessary and sufficient for existence of online prediction algorithm.

The Burkholder Algorithm

- Find Burkholder function $\boldsymbol{U}$ for V.
- At round t, play

$$\hat{y}_t = \frac{\boldsymbol{U}(x_1 y_1, \ldots, +x_t) - \boldsymbol{U}(x_1 y_1, \ldots, -x_t)}{2}.$$

Theorem (Burkholder prediction guarantee)

The Burkholder algorithm guarantees

$$\forall x_{1:n}, y_{1:n} \quad \sum_{t=1}^{n} -\hat{y}_t \cdot y_t \le \phi(x_1, y_1, \ldots, x_n, y_n).$$

Proof

Suffices to show

$$\sum_{t=1}^{n} -\hat{y}_t \cdot y_t - \phi(x_1, y_1, \ldots, x_n, y_n) \le 0.$$

Using property 1^o, upper bounded by

$$\sum_{t=1}^{n} -\hat{y}_t \cdot y_t + \boldsymbol{U}(x_1 y_1, \ldots, x_n y_n).$$

At round n, our goal is to solve

$$\min_{\hat{y}_n} \max_{y_n \in \{\pm 1\}} \{-\hat{y}_n \cdot y_n + \boldsymbol{U}(x_1 y_1, \ldots, x_n y_n)\}.$$

Equivalently,

$$\min_{\hat{y}_n} \{-\hat{y}_n + \boldsymbol{U}(x_1 y_1, \ldots, +x_n), \hat{y}_n + \boldsymbol{U}(x_1 y_1, \ldots, -x_n)\}.$$

Choosing $\hat{y}_n$ to equalize quantities gives

$$\hat{y}_n = \frac{\boldsymbol{U}(x_1 y_1, \ldots, +x_n) - \boldsymbol{U}(x_1 y_1, \ldots, -x_n)}{2}.$$

Resulting value is

$$\mathbb{E}_\epsilon \boldsymbol{U}(x_1 y_1, \ldots, \epsilon x_n) \underbrace{\le}_{\text{Using property } 3^o.} \boldsymbol{U}(x_1 y_1, \ldots, x_{n-1} y_{n-1}).$$

Repeating for each round, we get

$$\sum_{t=1}^{n} -\hat{y}_t \cdot y_t - \phi(x_1, y_1, \ldots, x_n, y_n) \le \boldsymbol{U}(0) \underbrace{\le}_{\text{Using property } 2^o.} 0.$$

Sufficient Statistics for Online Learning

What data should an online learner keep in memory?

- Mirror Descent/FTRL: sum of gradients
- Online Newton Step: ... + sum of outer products $\sum x_t x_t^\top$
- Adaptive gradient descent: ... + sum of norms $\sum \|x_t\|^2$

Theorem

Suppose prediction guarantee takes form

$$\forall x_{1:n}, y_{1:n} \quad \sum_{t=1}^{n} -\hat{y}_t \cdot y_t \le \phi\left(\sum_{t=1}^{n} \boldsymbol{T}(x_t, y_t)\right),$$

where sufficient statistic $\boldsymbol{T}$ maps $\mathcal{X} \times \mathcal{Y}$ to some vector space.

Then the Burkholder Algorithm only keeps $\sum_{t=1}^{n} \boldsymbol{T}(x_t, y_t)$ in memory.

Update structure:

$$\hat{y}_t = \frac{\boldsymbol{U}\left(\sum_{\tau=1}^{t-1} \boldsymbol{T}(x_\tau, y_\tau) + \boldsymbol{T}(x_t, 1)\right) - \boldsymbol{U}\left(\sum_{\tau=1}^{t-1} \boldsymbol{T}(x_\tau, y_\tau) + \boldsymbol{T}(x_t, -1)\right)}{2}.$$

Martingale Inequality Examples

Rademacher mgf For any real-valued $\boldsymbol{x}$,

$$\mathbb{E}_\epsilon \exp\left(\sum_{t=1}^{n} \epsilon_t \boldsymbol{x}_t - \frac{\boldsymbol{x}_t^2}{2}\right) \le 1.$$

Corresponding V:

$$V(x_1, \ldots, x_n) = \exp\left(\sum_{t=1}^{n} \epsilon_t x_t - \frac{x_t^2}{2}\right) - 1.$$

Nemirovski's Inequality / Martingale Type For any Banach space-valued $\boldsymbol{x}$ with smooth norm $\|\cdot\|$, there is $C > 0$ such that

$$\mathbb{E}_\epsilon \left\|\sum_{t=1}^{n} \epsilon_t \boldsymbol{x}_t\right\| \le C \sqrt{\sum_{t=1}^{n} \|\boldsymbol{x}_t\|^2}.$$

Corresponding V:

$$V(x_1, \ldots, x_n) = \left\|\sum_{t=1}^{n} x_t\right\| - C \sqrt{\sum_{t=1}^{n} \|x_t\|^2}.$$

Application: Matrix Prediction

Setup:

- $x_t \in \mathbb{R}^{d \times d}$ (e.g. user-movie pairs)
- Regret inequality:

$$\phi(x_1, y_1, \ldots, x_n, y_n) = \inf_{w : \|w\|_{\text{nuc}} \le \tau} \sum_{t=1}^{n} -y_t \langle w, x_t \rangle + C_n(x_1, \ldots, x_n).$$

What to choose for C_n?

- Want to capture all rank-r matrices $\implies \tau \approx \sqrt{rd^2}$.
- Martingale inequality:

$$\tau \cdot \mathbb{E}_\epsilon \left\|\sum_{t=1}^{n} \epsilon_t \boldsymbol{x}_t\right\|_{\text{op}} \le \mathbb{E}_\epsilon[C_n(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)].$$

✗ Uniform C_n must be $\Omega(\sqrt{rd^2 n})$.

✓ Matrix Concentration (Tropp '11):
$C_n(x_1, \ldots, x_n) = C\tau \sqrt{\left\|\sum_{t=1}^{n} x_t x_t^\top\right\|_{\text{op}} \vee \left\|\sum_{t=1}^{n} x_t^\top x_t\right\|_{\text{op}}}.$

- Suppose $x_t = e_{i_t} e_{j_t}^\top$, so $\sum_{t=1}^{n} x_t x_t^\top$ is histogram of row occurrences on diagonal.
- Let $N_{\text{row}} = \#$row occurrences, $N_{\text{col}} = \#$col. occurrences.
- Then $C_n(x_1, \ldots, x_n) = C\tau \sqrt{N_{\text{row}} \vee N_{\text{col}}}$.

Applying the Burkholder Method:

First guess:

$$V(x_1 y_1, \ldots, x_n y_n) = \tau \left\|\sum_{t=1}^{n} x_t y_t\right\|_{\text{op}} - C\tau \sqrt{\left\|\sum_{t=1}^{n} x_t x_t^\top\right\|_{\text{op}} \vee \left\|\sum_{t=1}^{n} x_t^\top x_t\right\|_{\text{op}}}$$

Instead, we will use

$$V(x_1 y_1, \ldots, x_n y_n) = \tau \left\|\sum_{t=1}^{n} \mathcal{H}(x_t) y_t\right\|_{\text{op}} - \frac{\eta\tau}{2} \left\|\sum_{t=1}^{n} \mathcal{H}(x_t)^2\right\|_{\text{op}} - \frac{\tau \log(2d)}{\eta},$$

with $\mathcal{H}(x) = \begin{pmatrix} 0 & x \\ x^\top & 0 \end{pmatrix}$ (introduces additional parameter η)

Sufficient statistic: $\sum_{t=1}^{n} \mathcal{H}(x_t) y_t$, $\sum_{t=1}^{n} \mathcal{H}(x_t)^2$.

Burkholder function

$$\begin{aligned}&\boldsymbol{U}(x_1 y_1, \ldots, x_n y_n) \\ &= \frac{\tau}{\eta} \log \operatorname{tr} \exp\left(\sum_{t=1}^{n} \eta y_t \mathcal{H}(x_t) - \frac{\eta^2}{2} \mathcal{H}(x_t)^2\right) - \frac{\tau \log(2d)}{2}.\end{aligned}$$

Proof sketch

For property 1^o, upper bound V by U.

$$\begin{aligned}&\tau \left\|\sum_{t=1}^{n} \mathcal{H}(x_t) y_t\right\|_{\text{op}} - \frac{\eta\tau}{2} \left\|\sum_{t=1}^{n} \mathcal{H}(x_t)^2\right\|_{\text{op}} \\ &\le \tau \cdot \lambda_{\max}\left(\sum_{t=1}^{n} \mathcal{H}(x_t) y_t - \frac{\eta}{2} \sum_{t=1}^{n} \mathcal{H}(x_t)^2\right) \\ &\le \frac{\tau}{\eta} \log \operatorname{tr} \exp\left(\sum_{t=1}^{n} \eta y_t \mathcal{H}(x_t) - \frac{\eta^2}{2} \mathcal{H}(x_t)^2\right).\end{aligned}$$

For property 3^o, fix n and let $R = \sum_{t=1}^{n-1} \eta y_t \mathcal{H}(x_t) - \frac{\eta^2}{2} \mathcal{H}(x_t)^2$.
Need to show

$$\mathbb{E}_\epsilon \log \operatorname{tr} \exp\left(R + \eta\epsilon \mathcal{H}(x_n) - \frac{\eta^2}{2} \mathcal{H}(x_n)^2\right) \le \log \operatorname{tr} \exp(R).$$

Use Lieb's Concavity Theorem:
$X \mapsto \operatorname{tr} \exp(A + \log X)$ is concave over PD cone for Hermitian A.

$$\begin{aligned}&\mathbb{E}_\epsilon \log \operatorname{tr} \exp\left(R + \eta\epsilon \mathcal{H}(x_n) - \frac{\eta^2}{2} \mathcal{H}(x_n)^2\right) \\ &= \mathbb{E}_\epsilon \log \operatorname{tr} \exp\left(R + \log(\exp(\eta\epsilon \mathcal{H}(x_n))) - \frac{\eta^2}{2} \mathcal{H}(x_n)^2\right) \\ &\le \log \operatorname{tr} \exp\left(R + \log(\mathbb{E}_\epsilon \exp(\eta\epsilon \mathcal{H}(x_n))) - \frac{\eta^2}{2} \mathcal{H}(x_n)^2\right).\end{aligned}$$

Result follows from Rademacher matrix mgf bound: $\log \mathbb{E}_\epsilon \exp(\epsilon\eta \mathcal{H}(x_t)) \preceq \frac{\eta^2}{2} \mathcal{H}(x_t)^2$.

Corollary: Martingale matrix Khintchine inequality: For all $\boldsymbol{x}$,

$$\mathbb{E}_\epsilon \left\|\sum_{t=1}^{n} \epsilon_t \boldsymbol{x}_t(\epsilon)\right\|_{\text{op}} \le \sqrt{2\, \mathbb{E}_\epsilon \left\|\sum_{t=1}^{n} \boldsymbol{x}_t(\epsilon) \boldsymbol{x}_t(\epsilon)^\top\right\|_{\text{op}} \vee \left\|\sum_{t=1}^{n} \boldsymbol{x}_t(\epsilon)^\top \boldsymbol{x}_t(\epsilon)\right\|_{\text{op}} \log(2d)}.$$

Further results

General losses

- Can handle general convex / Lipschitz losses and even get fast rates.
- Caveat: Not for arbitrary ϕ—Need benchmark + regret structure.

See paper for:

- General losses (including fast rates for square loss).
- Parameter-free online learning algorithms.
- AdaGrad-type algorithms with sharper constants.
- Further martingale inequalities.