Following the theme of Wouter's talk, the topic will be "easy data" in online learning. We will focus on online algorithms that obtain near optimal guarantees both in the statistical setting and the adversarial setting, and whose performance more generally interpolates between these extremes. We achieve this by exploiting a new connection to the theory of decoupling inequalities for martingales in Banach spaces. This connection will be explained at length.

This is based on joint work with Sasha Rakhlin and Karthik Sridharan.

Paper link: `http://dylanfoster.net/papers/zigzag_draft.pdf`

Here's the full abstract:

We develop a new family of algorithms for the online learning setting with regret against any data sequence bounded by the empirical Rademacher complexity of that sequence. To develop a general theory of when this type of adaptive regret bound is achievable we establish a connection to the theory of decoupling inequalities for martingales in Banach spaces. When the hypothesis class is a set of linear functions bounded in some norm, such a regret bound is achievable if and only if the norm satisfies certain decoupling inequalities for martingales. Donald Burkholders celebrated geometric characterization of decoupling inequalities (1984) states that such an inequality holds if and only if there exists a special function called a Burkholder (or Bellman) function satisfying certain restricted concavity properties. Our online learning algorithms are efficient in terms of queries to this function.

We realize our general theory by giving new and efficient algorithms for classes including lp norms, Schatten p-norms, group norms, operator norms, and reproducing kernel Hilbert spaces. The empirical Rademacher complexity regret bound implies—when used in the i.i.d. setting—a data-dependent complexity bound for excess risk after online-to-batch conversion. To obtain such adaptive methods, we introduce novel machinery, and the resulting algorithms are not based on the standard tools of online convex optimization.

1 Intro

At a high level, I'll talk about performance guarantees, give the algorithm (missing one key piece), and talk about how to fill in the key piece.

We work in the online supervised learning setting. The protocol is as follows.

1. For $t = 1, \dots, n$,
   - nature plays $x_t \in X$ (covariates).
   - Learner picks $\widehat{y}_t \in \mathbb{R}$.
   - Nature plays $y_t \in Y$.
   - Observe loss $\ell(\widehat{y}_t, y_t)$.

   Here nature is adversarial. (cf. ciontextual bandits)

Fix a hypothesis class F of $f : X \to \mathbb{R}$; look at the regret against this class.

$$\sum_{t=1}^{n} \ell(\widehat{y}_t, y_t) - \inf_{f \in F} \sum_{t=1}^{n} \ell(f(x_t), y_t).$$

Assume the loss is 1-Lipschitz.

To motivate our problem, let's look at several settings.

One setting is where the data is drawn from an iid source, $(x_t, y_t) \sim D$ Rates are the same as in batch statistical setting and we want bound on the excess risk. Online-to-batch (O2B) turns all the predictions our algorithm makes into a single prediction that gives an excess risk bound

$$ER := \mathop{\mathbb{E}}_{(x,y)\sim D} \ell(\widehat{y}(x), y) - \inf_{f\in F} \mathop{\mathbb{E}}_{(x,y)\sim D} \ell(f(x), y) \le \frac{B}{n}.$$

Instead of using O2B, we could directly use empirical risk minimization. Here the excess risk can be bounded in terms of empirical Rademacher complexity (sample n coin flips, take the function most correlated with those coin flips, and see how well we do on average),

$$\widehat{\text{Rad}}_F(x_1, \dots, x_n) = \mathop{\mathbb{E}}_{\varepsilon_1,\dots,\varepsilon_n\in\{\pm 1\}^n} \sup_{f\in F} \sum_{t=1}^n \varepsilon_t f(x_t)$$

This is appealing: for hinge loss, absolute loss, etc. that don't have curvature, you can still get lower bound.

ERM gives

$$ER \le \frac{1}{n} \mathop{\mathbb{E}}_{x_1,\dots,x_n\sim D} \widehat{\text{Rad}}_F(x_1, \dots, x_n).$$

A bound on VC dimension or covering number gives an upper bound on $\widehat{\text{Rad}}$.

When can we get this as upper bound on regret in fully online setting? We want a strategy $(\widehat{y}_t)$ that guarantees

$$\sum_{t=1}^n \ell(\widehat{y}_t, y_t) - \inf_{f\in F} \sum_{t=1}^n (f(x_t), y_t) \le D_F \widehat{\text{Rad}}_F(x_1, \dots, x_t).$$

If my data is drawn iid I get $\widehat{\mathbb{E}\text{Rad}}$.

Can I get this on arbitrary (not necessarily iid) sequence? No. The extent which I can do this depends on the hypothesis class.

Why pick $\widehat{\text{Rad}}$ in particular? Maybe there's some other function $B(x_1, \dots, x_n)$ measuring hardness. $\widehat{\text{Rad}}_F$ is in some sense the best measure.

Lemma 1.1. *Suppose $B(x_1, \dots, x_n)$ such that*

$$Reg(x_1, \dots, x_n) = \sum_{t=1}^n \ell(\widehat{y}_t, y_t) - \inf_{f\in F} \sum_{t=1}^n (f(x_t), y_t) \le B(x_1, \dots, x_n).$$

Then

$$\widehat{\text{Rad}}(x_1, \dots, x_n) \le B(x_1, \dots, x_n).$$

This works for hinge loss, linear loss, 0-1 loss with binary predictions, $|\widehat{y}_t| < 1, |y_t| < 1, |f(x_t)| < 1$. We can consider prediction or regression. Square loss is a fairly rich problem we are still investigating.

If you know your data is iid to begin with, getting this regret is pretty straightforward: follow-the-leader. This doesn't give a bound when your data is not iid. We recover the bound when the data is iid, still have guarantees in the adversarial setting, interpolates between these two cases.

Proof. Let's focus on hinge loss.

$$\ell_{\text{hinge}}(\widehat{y}, y) = \max(0, 1 - \widehat{y}y) = 1 - \widehat{y} \cdot y$$

(Assume arguments bounded by 1.)

We can rewrite this regret bound as

$$\sum_{t=1}^{n} 1 - \widehat{y}_t \cdot y_t - \inf_{f \in F} \sum_{t=1}^{n} (1 - f(x_t) \cdot y_t \leq B(x_1, \ldots, x_n) \tag{1}$$

$$-\sum_{t=1}^{n} \widehat{y}_t \cdot y_t - \inf_{f \in F} - \sum_{t=1}^{n} f(x_t) \cdot y_t \leq B(x_1, \ldots, x_n). \tag{2}$$

Sample $\varepsilon_1, \ldots, \varepsilon_n \in \{\pm 1\}$ uniformly. Play $y_t = \varepsilon_t$. Then

$$\mathop{\mathbb{E}}_{\varepsilon} \left[-\sum_{t=1}^{n} \widehat{y}_t \cdot \varepsilon_t - \inf_{f \in F} f(x_t) \cdot y_t \right] \leq B(x_1, \ldots, x_n). \tag{3}$$

The first term is mean 0 because the learner chooses $\widehat{y}_t$ before seeing the loss.

$$\mathop{\mathbb{E}}_{\varepsilon}[- \inf_{f \in F} - \sum_{t=1}^{n} f(x_t) \cdot \varepsilon_t] \leq B(x_1, \ldots, x_n) \tag{4}$$

$$\mathop{\mathbb{E}}_{\varepsilon} \sup_{f \in F} \sum_{t=1}^{n} f(x_t) \cdot \varepsilon_t \leq B(x_1, \ldots, x_n). \tag{5}$$

□

2 Algorithm

Consider a special case. Let $X \subseteq (B, \|\cdot\|)$ be a subset of Banach space. Our hypotheses live in the dual

$$\{x \mapsto \langle w, x \rangle : \|w\|_* \leq 1\}.$$

In this case $\widehat{\text{Rad}}$ has a very nice form.

$$\widehat{\text{Rad}} = \mathop{\mathbb{E}}_{\varepsilon_{1:n}} \sup_{w : \|w\|_* \leq 1} \sum_{t=1}^{n} \varepsilon_t \langle w, x_t \rangle = \mathop{\mathbb{E}}_{\varepsilon_{1:n}} \left\| \sum_{t=1}^{n} \varepsilon_t x_t \right\|.$$

Example: ℓ_2. Both norms are ℓ_2. We have

$$\widehat{\text{Rad}}(x_1, \ldots, x_n) \approx \sqrt{\sum_{t=1}^{n} \|x_t\|_2^2} \leq \sqrt{n}.$$

This bound can be achieved using gradient descent with learning rate tuning. Let

$$w_{t+1} = w_t - \eta x_t, \qquad \eta_t = \frac{1}{\sqrt{\sum_{s=1}^{t} \|x_s\|_2^2}} \tag{6}$$

$$\widehat{y}_{t+1} = \langle w_{t+1}, x_{t+1} \rangle. \tag{7}$$

Before this work this was the only case where you get this bound.

Goal:

$$\sum_{t=1}^n \ell(\widehat{y}_t, y_t) - \inf_f \sum_{t=1}^n \ell(f(x_t), y_t) - D\widehat{\text{Rad}}(x_1, \ldots, x_t). \tag{8}$$

(Think of this as value of game against adversary.) Assume loss is convex (so that $\ell(\widehat{y}, y) - \ell(f, y) \le \ell'(\widehat{y}, y)(\widehat{y} - f)$ and linearize.

$$\begin{aligned} &\le \sum_{t=1}^n \widehat{y}_t \cdot \ell'(\widehat{y}_t, y_t) - \inf_{f\in F} \sum_{t=1}^n f(x_t)\ell'(\widehat{y}_t, y_t) - D\widehat{\text{Rad}} & (9)\\ &= \sum_{t=1}^n \widehat{y}_t cdpt \ell'_t + \left\|\sum_{t=1}^n \ell'_t \cdot x_t\right\| - D \mathop{\mathbb{E}}_{\varepsilon} \left\|\sum_{t=1}^n \varepsilon_t \ell'_t \cdot x_t\right\|. & (10) \end{aligned}$$

We need to control this difference of norms.

Pick $\widehat{y}$ so I can upper bound the first two terms by something that looks like the third term.

(Actually, we use a slightly different $\widehat{\text{Rad}}$ that is slightly smaller, and easier to handle using calculations.)

$$\widehat{\text{Rad}}(x_1, \ldots, x_n, \ell'_1, \ldots, \ell'_n) = \mathop{\mathbb{E}}_{\varepsilon_1, \ldots, \varepsilon_n} \left\|\sum_{t=1}^n \varepsilon_t \ell'_t \cdot x_t\right\|.$$

We want to analyze difference of two norms, and see what happens when adversary changes. $Z \mapsto \|A + Z\| - D\|B + Z\|$ is hard to analyze because this is neither convex nor concave. Move to surrogate that has a weak concavity property.

Proposition 2.1: Suppose $U : X \times X \to \mathbb{R}$,

1. $U(x, x') \ge \|x\| - D\|x'\|$.
2. (Concave along the diagonals) $Z \mapsto U(X + Z, X' + \varepsilon Z)$ is concave for all $x, x', \varepsilon \in \{\pm 1\}$.
3. $U(0, 0) \le 0$.

Then there is an algorithm, efficient in queries to U, giving $(\widehat{y}_t)$ such that

$$\text{Reg} \le D\widehat{\text{Rad}}(x_1, \ldots, x_n),$$

Continuing,

$$\begin{aligned} &\le \sum_{t=1}^n \widehat{y}_t \cdot \ell'_t + \mathop{\mathbb{E}}_{\varepsilon} U(\ell'_t x_t, \textstyle\sum \varepsilon_t \ell'_t x_t) & (11)\\ &= \sum_{t=1}^{n-1} \widehat{y}_t \ell'_t + \widehat{y}_n \cdot \ell'_n + \mathop{\mathbb{E}}_{\varepsilon} U(\cdot, \cdot) & (12) \end{aligned}$$

$$\sup_{x_n} \inf_{\widehat{y}_n} \sup_{\ell'_n} \left[\widehat{y}_n \cdot \ell'_n + \mathop{\mathbb{E}}_{\varepsilon_1, \ldots, \varepsilon_n} U(\sum_{t=1}^n \ell'_t \cdot x_t, \sum_{t=1}^n \varepsilon_t \ell'_t \cdot x_t)\right] \le \mathop{\mathbb{E}}_{\varepsilon_1, \ldots, \varepsilon_n} U(\sum_{t=1}^{n-1}, \sum_{t=1}^{n-1}) \tag{13}$$

To see this, let $G_n(z) = \mathbb{E}_{\varepsilon_1,\ldots,\varepsilon_n} U(\sum_{t=1}^{n-1} \ell'_t x_t + z \cdot x_t, \sum_{t=1}^{n-1} \varepsilon_t \ell'_t x_t + \varepsilon_n Z x_n)$. Now

$$\begin{aligned} \sup\inf\sup &= \sup_{x_n} \inf_{\widehat{y}_n} \sup_{\ell'_n} (\widehat{y}_n \cdot \ell'_n + G_n(\ell'_n)) \qquad \widehat{y}_n = -G'_n(0) && (14)\\ &\le \sup_{x_n} \sup_{\ell'_n} [-G'_n(0)\ell'_n + G_n(\ell'_n)] && (15)\\ &\le G_n(0) && (16)\end{aligned}$$

using zigzag concavity (2). Use this recursively.

We show TFAE:

1. $\exists U$.
2. $D\widehat{\text{Rad}}$ achievable
3. UMD (unconditional martingale difference)

Martingale difference property for $(Z_1, \ldots, Z_n)$, $Z_t \in B$ is

$$\mathbb{E}[Z_t | Z_1, \ldots, Z_{t=1}] = 0.$$

In a lot of ways this behaves like a sequence of random variables, UMD quantifies the way which it doesn't.

Definition 2.2 (Unconditional martingale difference)**:** UMD_p means: for all $(z_t)_{t\le n}$ and $(\varepsilon_t)_{t\le n}$,

$$\begin{aligned} \mathop{\mathbb{E}}_{z} \left\| \sum_{t=1}^n Z_t \right\|^p &\le C_p^p \mathop{\mathbb{E}}_{z} \left\| \sum_{t=1}^n \varepsilon_t Z_t \right\|^p && (17)\\ \mathop{\mathbb{E}}_{z} \left\| \sum_{t=1}^n \varepsilon_t Z_t \right\|^p &\le C_p^p \mathop{\mathbb{E}}_{Z} \left\| \sum Z_t \right\|^p . && (18)\end{aligned}$$

The proposition still holds if we replace (1) with

$$U(x, x') \ge \|x\|^p - D_p^p \|x'\|^p .$$

Theorem 2.3 (Burkholder 1984)**.** *UMD_p holds with constant C_p iff there exists U satisfying (1) with constant C_p.*

You can use the UMD_p to known information-theoretically whether the bound is achievable.

We can leverage tools people have used in probability and analysis to create U functions. People are interested in U functions because it gives a way to certify that UMD_p property holds.

Burkholder's construction for the scalar case:

$$U_p^{\mathbb{R}}(x, x') = \alpha_p(|x| - \beta_p|x'|)(|x| + |x'|)^{p-1}$$

where $\beta_p = \max\{p, \frac{p}{p-1}\} - 1$,

$$U(x, x') \geq |x|^p - \beta_p |x'|^p.$$

This is the tightest possible constant for which this inequality holds. This blows up when $p = 1$, so is motivation for considering other powers. Take $p = 1 + \frac{1}{\ln n}$ which gives what you want up to log factors. Martingale inequalities often don't hold for $p = 1$.

For $p = 2, \beta_p = 1$,

$$U(x + z, x' + z) = (x + z)^2 - (x' + z)^2 = x^2 - (x')^2 + 2xz - 2x'z.$$

This spits out gradient descent with parameter tuning.

For ℓ_p, sum componentwise,

$$U_{\mathbb{P}}^{\ell_p}(x, x') = \sum_i U_p^{\mathbb{R}}(x_i, x_i'),$$

cf. AdaGrad.

What would have surprised someone in the field? Often in online/statistical learning, the worst-case and best-case rates match. You can imagine an algorithm in online setting that has same worst-case setting as ERM setting. Having it match best-case is much stronger. This interpolates between statistical and adversarial.

For a wide class of properties, this reduces to crisp geometric property. U reduces problem of algorithm design to purely geometric problem.

All the U's we know are either U's in closed form, and U's not efficiently computable. For future work, what can we gain in terms of computing U functions from CS perspective. People in probability aren't thinking of computing U, ex. as optimization problem. If you don't care about exact constants, it should be easier.

Burkholder reduced to PDE and solved it, a tour-de-force. Ex. with β_p^4 it's much easier, doable in a few lines.

Ex. trace norm bounded matrices: $(\log d)^4$. Difference between worst and best case is $\sqrt{d}$.

High-level future work directions:

- OLO/OCO and regression.
- Bandit.

Constraint sets? When losses are curved, can you get better rates?

When function class is not linear, you can still write down generalization for UMD. What the right U function is, is not clear. UMD property holds for function class, does it hold if you compose or transform them? What are the algorithms? Our understanding of U is limited. Ex. go from a p to U for different p. Algorithmically we know how to make this work.

More broadly, I'm curious about other places where this idea of taking a property of dependent random process and reducing to a geometric property helps. You can do this beyond online learning.

UMD is the first property with a geometric form, but people have come up with U functions for different inequalities, like Doob's martingale inequality.