Tight Bounds on Minimax Regret under Logarithmic Loss via Self-Concordance

Blair Bilodeau [1,2,3] Dylan J. Foster [4] Daniel M. Roy [1,2,3]

VECTOR INSTITUTE

[1] University of Toronto [2] Vector Institute [3] Institute for Advanced Study [4] Massachusetts Institute of Technology

Contribution Summary

- **Tight upper bounds** on minimax regret under log loss for all equivalence classes of experts up to sequential entropy.
- **Matching lower bound** for 1-Lipshitz experts on $[0,1]^p$.
- Minimax regret under log loss **cannot be resolved entirely by the sequential entropy** of the expert class, unlike square loss.
- First **truncation-free argument** which improves on previous best results, and leads to a **chaining-free** upper bound.

Online Learning and Minimax Regret

Traditional statistical learning analyzes data in a *batch* to produce a prediction function, which is used on future observations assumed to be generated i.i.d. from the training distribution.

Online learning is a framework for predicting future observations without any assumptions about the data generating process.

For rounds $t = 1, \dots, n$:
- Environment supplies *context* $x_t \in \mathcal{X}$, using the history;
- Player *predicts* $\hat{p}_t \in [0,1]$, a distribution on binary observations;
- Adversary generates an *observation* $y_t \in \{0,1\}$;
- Player incurs *log loss* $\ell(\hat{p}_t, y_t) = -y_t \log(\hat{p}_t) - (1-y_t)\log(1-\hat{p}_t)$.

Observe that the log loss corresponds to the *negative log-likelihood* of the observation under the predicted distribution.

In general, the player's cumulative loss grows super-linearly in n.

Performance is measured with respect to an *expert class* $\mathcal{F} \subseteq [0,1]^{\mathcal{X}}$. The player's goal is to **compete against the best expert in hindsight**, which characterizes their *regret*:

$$\mathcal{R}_n(\mathcal{F}; \hat{\boldsymbol{p}}, \boldsymbol{x}, \boldsymbol{y}) = \sum_{t=1}^{n} \ell(\hat{p}_t, y_t) - \inf_{f \in \mathcal{F}} \sum_{t=1}^{n} \ell(f(x_t), y_t).$$

The *minimax regret* is an **algorithm-free concept** that measures how difficult an expert class is to learn over worst-case observations.

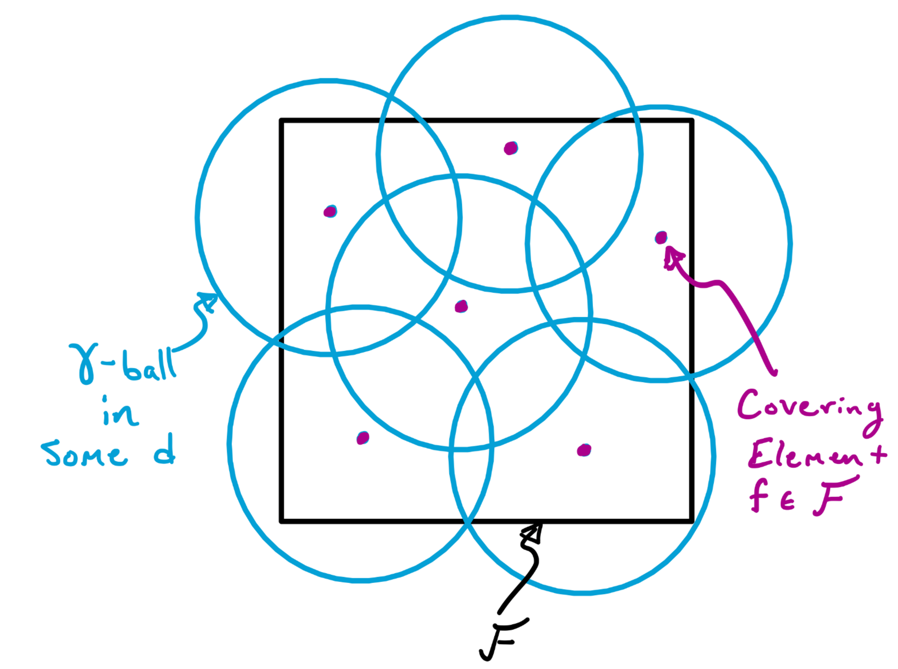
$$\mathcal{R}_n(\mathcal{F}) = \sup_{x_1} \inf_{\hat{p}_1} \sup_{y_1} \cdots \sup_{x_n} \inf_{\hat{p}_n} \sup_{y_n} \mathcal{R}_n(\mathcal{F}; \hat{\boldsymbol{p}}, \boldsymbol{x}, \boldsymbol{y}).$$

Goal: Bound the minimax regret for arbitrary expert classes.
Difficulty: Log loss is neither bounded nor Lipschitz.

Sequential Covering and Entropy

We control the minimax regret by:
i) Bounding regret against a *finite cover* of $\mathcal{F}$, and
ii) Bounding the *approximation error* of this cover.

A cover is determined by the *notion of distance* (d). Cesa-Bianchi & Lugosi (1999) used a uniform covering of $\mathcal{F}$ on all of $\mathcal{X}$, which is too coarse for many expert classes.



An *empirical cover* only **covers $\mathcal{F}$ on the observed contexts**, but we also need to consider the sequential dependency structure.
We use *sequential covering*, introduced by Rakhlin & Sridharan (2014).

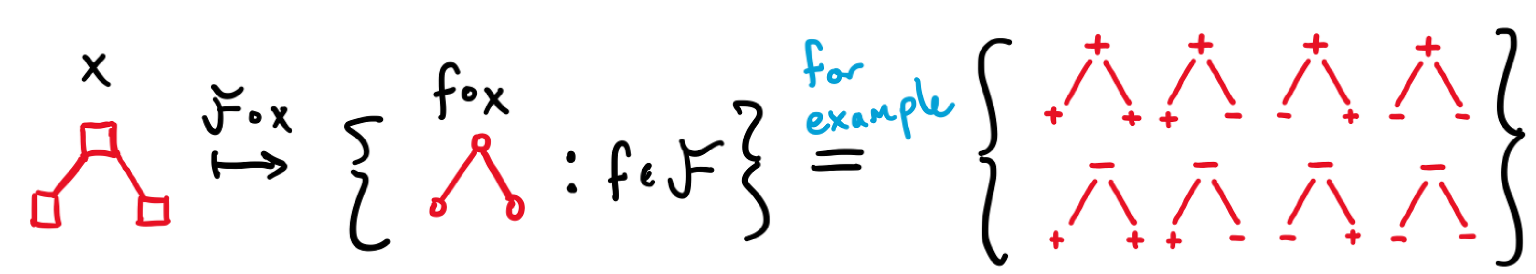
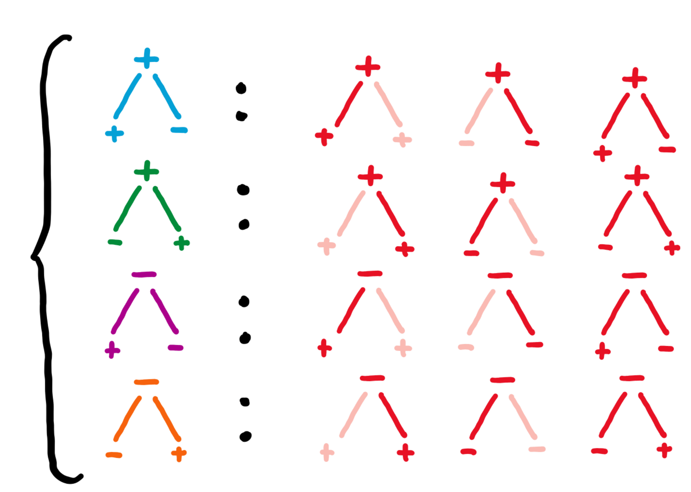


Fig: Composition of context tree with experts illustrated for binary experts.

An exact sequential cover of the binary experts example requires only 4 trees rather than the 8 needed for an empirical cover, since **a new covering element can be chosen for each path** rather than each tree of $\mathcal{F} \circ \boldsymbol{x}$.



We denote the *sequential γ-covering number* by $\mathcal{N}_\infty(\mathcal{F} \circ \boldsymbol{x}, \gamma)$.
The *sequential entropy* for trees of depth n is defined by

$$\mathcal{H}_\infty(\mathcal{F}, \gamma, n) = \sup_{\boldsymbol{x}} \log \mathcal{N}_\infty(\mathcal{F} \circ \boldsymbol{x}, \gamma).$$

Upper Bound

For any context space $\mathcal{X}$ and class of experts $\mathcal{F} \subseteq [0,1]^{\mathcal{X}}$,

$$\mathcal{R}_n(\mathcal{F}) \le \mathcal{O}\Big(\inf_{\gamma > 0} \Big\{ n\gamma + \mathcal{H}_\infty(\mathcal{F}, \gamma, n) \Big\} \Big).$$

In particular, if $\mathcal{H}_\infty(\mathcal{F}, \gamma, n) \le \mathcal{O}(\gamma^{-p})$, then $\mathcal{R}_n(\mathcal{F}) \le \mathcal{O}(n^{\frac{p}{p+1}})$.

Applications

Sequential Rademacher Complexity

Using $\mathfrak{R}_n(\mathcal{F}) = \sup_{\boldsymbol{x}} \mathbb{E}_{\varepsilon \sim \{\pm 1\}^n} \sup_{f \in \mathcal{F}} \sum_{t=1}^{n} \varepsilon_t f(x_t(\varepsilon))$, Rakhlin et al. (2015) showed that $\mathcal{H}_\infty(\mathcal{F}, \gamma, n) \le \tilde{\mathcal{O}}(\mathfrak{R}_n^2(\mathcal{F})/(n\gamma^2))$. So, for all $\mathcal{F}$,

$$\mathcal{R}_n(\mathcal{F}) \le \tilde{\mathcal{O}}\Big(\mathfrak{R}_n^{2/3}(\mathcal{F}) \cdot n^{1/3} \Big).$$

Neural Networks

$\mathcal{F} = \{$neural nets | Lipschitz activations and ℓ_1-bounded weights$\}$
Rakhlin et al. (2015) also showed $\mathfrak{R}_n(\mathcal{F}) \le \tilde{\mathcal{O}}(\sqrt{n})$, so we have

$$\mathcal{R}_n(\mathcal{F}) \le \tilde{\mathcal{O}}(n^{2/3}).$$

Linear Predictors

For $\mathcal{F} = \{f(x) = \frac{1}{2}[1 + \langle w, x\rangle] \mid \|w\| \le 1\}$, $\mathcal{H}_\infty(\mathcal{F}, \gamma, n) = \tilde{\mathcal{O}}(1/\gamma^2)$, so

$$\mathcal{R}_n(\mathcal{F}) \le \tilde{\mathcal{O}}(n^{2/3}).$$

However, Rakhlin & Sridharan (2015) have an algorithm specifically for linear predictors that gives $\mathcal{R}_n(\mathcal{F}) \le \tilde{\mathcal{O}}(\sqrt{n})$.

Lower Bound

For any $p \in \mathbb{N}$, let $\mathcal{F} = \{f : [0,1]^p \to [0,1] \mid f \text{ is 1-Lipschitz}\}$.
Then, $\mathcal{H}_\infty(\mathcal{F}, \gamma, n) = \Theta(\gamma^{-p})$ and $\mathcal{R}_n(\mathcal{F}) = \Theta(n^{\frac{p}{p+1}})$.

Implications

1) **Our upper bound is tight** if only sequential entropy is used.
2) Using the linear predictors example, minimax regret under log loss **cannot be resolved entirely by sequential entropy**.

Ask me about how this differs from other losses.

Self-Concordance

Our proof technique exploits the *self-concordance* of logarithms.
A function $F : \mathbb{R} \to \mathbb{R}$ is self-concordant if for all $x \in \mathbb{R}$,

$$|F'''(x)| \le 2F''(x)^{3/2}.$$

Ask me about how this leads to a truncation-free argument.

NSERC CRSNG