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

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

Presented at the 2020 International Conference on Machine Learning

[1]Department of Statistical Sciences, University of Toronto
[2]Vector Institute
[3]Institute for Foundations of Data Science, Massachusetts Institute of Technology

Contextual Online Learning with Log Loss

Example: Image Identification

For rounds $t = 1, \ldots, n$:

Contextual Online Learning with Log Loss

Example: Image Identification

For rounds $t = 1, \ldots, n$:

- Receive an image.

Contextual Online Learning with Log Loss

Example: Image Identification

For rounds $t = 1, \ldots, n$:

- Receive an image.
- Assign a probability to whether the image is adversarially generated.

Contextual Online Learning with Log Loss

Example: Image Identification

For rounds $t = 1, \ldots, n$:

- Receive an image.
- Assign a probability to whether the image is adversarially generated.
- Observe the true label.

Contextual Online Learning with Log Loss

Example: Image Identification

For rounds $t = 1, \ldots, n$:

- Receive an image.
- Assign a probability to whether the image is adversarially generated.
- Observe the true label.
- Incur penalty based on prediction and observation.

Contextual Online Learning with Log Loss

Example: Image Identification

For rounds $t = 1, \ldots, n$:

- Receive an image. Context $x_t \in \mathcal{X}$
- Assign a probability to whether the image is adversarially generated.
- Observe the true label.
- Incur penalty based on prediction and observation.

Contextual Online Learning with Log Loss

Example: Image Identification

For rounds $t = 1, \ldots, n$:

- Receive an image. Context $x_t \in \mathcal{X}$
- Assign a probability. Prediction $\hat{p}_t \in [0, 1]$
- Observe the true label.
- Incur penalty based on prediction and observation.

Contextual Online Learning with Log Loss

Example: Image Identification

For rounds $t = 1, \ldots, n$:

- Receive an image. Context $x_t \in \mathcal{X}$
- Assign a probability. Prediction $\hat{p}_t \in [0, 1]$
- Observe the true label. Observation $y_t \in \{0, 1\}$
- Incur penalty based on prediction and observation.

Contextual Online Learning with Log Loss

Example: Image Identification

For rounds $t = 1, \ldots, n$:

- Receive an image. Context $x_t \in \mathcal{X}$
- Assign a probability. Prediction $\hat{p}_t \in [0, 1]$
- Observe the true label. Observation $y_t \in \{0, 1\}$
- Incur penalty. Loss $\ell_{\log}(\hat{p}_t, y_t) = -y_t \log(\hat{p}_t) - (1 - y_t) \log(\hat{p}_t)$

Contextual Online Learning with Log Loss

Example: Image Identification

For rounds $t = 1, \ldots, n$:

- Receive an image. Context $x_t \in \mathcal{X}$
- Assign a probability. Prediction $\hat{p}_t \in [0, 1]$
- Observe the true label. Observation $y_t \in \{0, 1\}$
- Incur penalty. Loss $\ell_{\log}(\hat{p}_t, y_t) = -y_t \log(\hat{p}_t) - (1 - y_t) \log(\hat{p}_t)$

Notice that $\ell_{\log}$ equals the negative log likelihood of y_t under the model $\hat{p}_t$.

Contextual Online Learning with Log Loss

Example: Image Identification

For rounds $t = 1, \ldots, n$:

- Receive an image. Context $x_t \in \mathcal{X}$
- Assign a probability. Prediction $\hat{p}_t \in [0, 1]$
- Observe the true label. Observation $y_t \in \{0, 1\}$
- Incur penalty. Loss $\ell_{\log}(\hat{p}_t, y_t) = -y_t \log(\hat{p}_t) - (1 - y_t) \log(\hat{p}_t)$

Notice that $\ell_{\log}$ equals the negative log likelihood of y_t under the model $\hat{p}_t$.

Challenges

Contextual Online Learning with Log Loss

Example: Image Identification

For rounds $t = 1, \ldots, n$:

- Receive an image. Context $x_t \in \mathcal{X}$
- Assign a probability. Prediction $\hat{p}_t \in [0, 1]$
- Observe the true label. Observation $y_t \in \{0, 1\}$
- Incur penalty. Loss $\ell_{\log}(\hat{p}_t, y_t) = -y_t \log(\hat{p}_t) - (1 - y_t) \log(\hat{p}_t)$

Notice that $\ell_{\log}$ equals the negative log likelihood of y_t under the model $\hat{p}_t$.

Challenges

- We do not rely on data-generating assumptions.

Contextual Online Learning with Log Loss

Example: Image Identification

For rounds $t = 1, \dots, n$:

- Receive an image. Context $x_t \in \mathcal{X}$
- Assign a probability. Prediction $\hat{p}_t \in [0, 1]$
- Observe the true label. Observation $y_t \in \{0, 1\}$
- Incur penalty. Loss $\ell_{\log}(\hat{p}_t, y_t) = -y_t \log(\hat{p}_t) - (1 - y_t) \log(\hat{p}_t)$

Notice that $\ell_{\log}$ equals the negative log likelihood of y_t under the model $\hat{p}_t$.

Challenges

- We do not rely on data-generating assumptions.
- $\ell_{\log}$ is neither bounded nor Lipschitz.

Measuring Performance with Regret

Without model assumptions, guaranteed small loss on predictions is impossible.

Measuring Performance with Regret

Without model assumptions, guaranteed small loss on predictions is impossible.

If I can't promise about the future, can I say something about the past?

Measuring Performance with Regret

Without model assumptions, guaranteed small loss on predictions is impossible.
If I can't promise about the future, can I say something about the past?

Consider a relative notion of performance in hindsight.

- Relative to a class $\mathcal{F} \subseteq \{f : \mathcal{X} \to [0,1]\}$, consisting of **experts** $f \in \mathcal{F}$.
- Compete against the optimal $f \in \mathcal{F}$ on the actual sequence of observations.

Measuring Performance with Regret

Without model assumptions, guaranteed small loss on predictions is impossible.
If I can't promise about the future, can I say something about the past?

Consider a relative notion of performance in hindsight.

- Relative to a class $\mathcal{F} \subseteq \{f : \mathcal{X} \to [0,1]\}$, consisting of **experts** $f \in \mathcal{F}$.
- Compete against the optimal $f \in \mathcal{F}$ on the actual sequence of observations.

Regret: $$R_n(\hat{\mathbf{p}}; \mathcal{F}, \mathbf{x}, \mathbf{y}) = \sum_{t=1}^{n} \ell_{\log}(\hat{p}_t, y_t) - \inf_{f \in \mathcal{F}} \sum_{t=1}^{n} \ell_{\log}(f(x_t), y_t).$$

Measuring Performance with Regret

Without model assumptions, guaranteed small loss on predictions is impossible.
If I can't promise about the future, can I say something about the past?

Consider a relative notion of performance in hindsight.

- Relative to a class $\mathcal{F} \subseteq \{f : \mathcal{X} \to [0,1]\}$, consisting of **experts** $f \in \mathcal{F}$.
- Compete against the optimal $f \in \mathcal{F}$ on the actual sequence of observations.

$$\text{Regret:} \qquad R_n(\hat{\mathbf{p}}; \mathcal{F}, \mathbf{x}, \mathbf{y}) = \sum_{t=1}^{n} \ell_{\log}(\hat{p}_t, y_t) - \inf_{f \in \mathcal{F}} \sum_{t=1}^{n} \ell_{\log}(f(x_t), y_t).$$

This quantity depends on

- $\hat{\mathbf{p}}$: Player predictions,
- $\mathcal{F}$: Expert class,
- $\mathbf{x}$: Observed contexts,
- $\mathbf{y}$: Observed data points.

Summary of Results

We control the **minimax regret** using the **sequential entropy** of the experts $\mathcal{F}$.

Summary of Results

We control the **minimax regret** using the **sequential entropy** of the experts $\mathcal{F}$.

- **Minimax regret:** the *smallest possible* regret under *worst-case* observations.
- **Sequential entropy:** a *data-dependent complexity measure* for $\mathcal{F}$.

Summary of Results

We control the **minimax regret** using the **sequential entropy** of the experts $\mathcal{F}$.

- **Minimax regret:** the *smallest possible* regret under *worst-case* observations.
- **Sequential entropy:** a *data-dependent complexity measure* for $\mathcal{F}$.

Contributions

Summary of Results

We control the **minimax regret** using the **sequential entropy** of the experts $\mathcal{F}$.

- **Minimax regret:** the *smallest possible* regret under *worst-case* observations.
- **Sequential entropy:** a *data-dependent complexity measure* for $\mathcal{F}$.

Contributions

- Improved upper bound for expert classes with polynomial sequential entropy.

Summary of Results

We control the **minimax regret** using the **sequential entropy** of the experts $\mathcal{F}$.

- **Minimax regret:** the *smallest possible* regret under *worst-case* observations.
- **Sequential entropy:** a *data-dependent complexity measure* for $\mathcal{F}$.

Contributions

- Improved upper bound for expert classes with polynomial sequential entropy.
- Novel proof technique that exploits the curvature of log loss to avoid a key "truncation step" used by previous works.

Summary of Results

We control the **minimax regret** using the **sequential entropy** of the experts $\mathcal{F}$.

- **Minimax regret:** the *smallest possible* regret under *worst-case* observations.
- **Sequential entropy:** a *data-dependent complexity measure* for $\mathcal{F}$.

Contributions

- Improved upper bound for expert classes with polynomial sequential entropy.
- Novel proof technique that exploits the curvature of log loss to avoid a key "truncation step" used by previous works.
- Resolve the minimax regret with log loss for Lipschitz experts on $[0, 1]^p$ with matching lower bounds.

Summary of Results

We control the **minimax regret** using the **sequential entropy** of the experts $\mathcal{F}$.

- **Minimax regret:** the *smallest possible* regret under *worst-case* observations.
- **Sequential entropy:** a *data-dependent complexity measure* for $\mathcal{F}$.

Contributions

- Improved upper bound for expert classes with polynomial sequential entropy.
- Novel proof technique that exploits the curvature of log loss to avoid a key "truncation step" used by previous works.
- Resolve the minimax regret with log loss for Lipschitz experts on $[0,1]^p$ with matching lower bounds.
- Conclude the minimax regret with log loss cannot be completely characterized using sequential entropy.

Minimax Regret

Regret: $$R_n(\hat{\mathbf{p}}; \mathcal{F}, \mathbf{x}, \mathbf{y}) = \sum_{t=1}^{n} \ell_{\log}(\hat{p}_t, y_t) - \inf_{f \in \mathcal{F}} \sum_{t=1}^{n} \ell_{\log}(f(x_t), y_t).$$

Minimax regret: an **algorithm-free quantity** on **worst-case observations**.

$$R_n(\mathcal{F}) = \sup_{x_1} \inf_{\hat{p}_1} \sup_{y_1} \sup_{x_2} \inf_{\hat{p}_2} \sup_{y_2} \cdots \sup_{x_n} \inf_{\hat{p}_n} \sup_{y_n} R_n(\hat{\mathbf{p}}; \mathcal{F}, \mathbf{x}, \mathbf{y}).$$

Minimax Regret

Regret: $$R_n(\hat{\mathbf{p}}; \mathcal{F}, \mathbf{x}, \mathbf{y}) = \sum_{t=1}^{n} \ell_{\log}(\hat{p}_t, y_t) - \inf_{f \in \mathcal{F}} \sum_{t=1}^{n} \ell_{\log}(f(x_t), y_t).$$

Minimax regret: an **algorithm-free quantity** on **worst-case observations**.

$$R_n(\mathcal{F}) = \mathbf{\sup_{\boldsymbol{x_1}}} \inf_{\hat{p}_1} \sup_{y_1} \; \sup_{x_2} \inf_{\hat{p}_2} \sup_{y_2} \; \cdots \; \sup_{x_n} \inf_{\hat{p}_n} \sup_{y_n} R_n(\hat{\mathbf{p}}; \mathcal{F}, \mathbf{x}, \mathbf{y}).$$

The first context is observed.

Minimax Regret

Regret: $$R_n(\hat{\mathbf{p}}; \mathcal{F}, \mathbf{x}, \mathbf{y}) = \sum_{t=1}^{n} \ell_{\log}(\hat{p}_t, y_t) - \inf_{f \in \mathcal{F}} \sum_{t=1}^{n} \ell_{\log}(f(x_t), y_t).$$

Minimax regret: an **algorithm-free quantity** on **worst-case observations**.

$$R_n(\mathcal{F}) = \sup_{x_1} \boldsymbol{\inf_{\hat{p}_1}} \sup_{y_1} \; \sup_{x_2} \inf_{\hat{p}_2} \sup_{y_2} \; \cdots \; \sup_{x_n} \inf_{\hat{p}_n} \sup_{y_n} R_n(\hat{\mathbf{p}}; \mathcal{F}, \mathbf{x}, \mathbf{y}).$$

The player makes their prediction.

Minimax Regret

Regret: $$R_n(\hat{\mathbf{p}}; \mathcal{F}, \mathbf{x}, \mathbf{y}) = \sum_{t=1}^{n} \ell_{\log}(\hat{p}_t, y_t) - \inf_{f \in \mathcal{F}} \sum_{t=1}^{n} \ell_{\log}(f(x_t), y_t).$$

Minimax regret: an **algorithm-free quantity** on **worst-case observations**.

$$R_n(\mathcal{F}) = \sup_{x_1} \inf_{\hat{p}_1} \boldsymbol{\sup_{y_1}} \ \sup_{x_2} \inf_{\hat{p}_2} \sup_{y_2} \ \cdots \ \sup_{x_n} \inf_{\hat{p}_n} \sup_{y_n} R_n(\hat{\mathbf{p}}; \mathcal{F}, \mathbf{x}, \mathbf{y}).$$

The adversary plays an observation.

Minimax Regret

Regret: $$R_n(\hat{\mathbf{p}}; \mathcal{F}, \mathbf{x}, \mathbf{y}) = \sum_{t=1}^{n} \ell_{\log}(\hat{p}_t, y_t) - \inf_{f \in \mathcal{F}} \sum_{t=1}^{n} \ell_{\log}(f(x_t), y_t).$$

Minimax regret: an **algorithm-free quantity** on **worst-case observations**.

$$R_n(\mathcal{F}) = \sup_{x_1} \inf_{\hat{p}_1} \sup_{y_1} \boldsymbol{\sup_{x_2} \inf_{\hat{p}_2} \sup_{y_2}} \cdots \sup_{x_n} \inf_{\hat{p}_n} \sup_{y_n} R_n(\hat{\mathbf{p}}; \mathcal{F}, \mathbf{x}, \mathbf{y}).$$

This repeats for all n rounds.

Minimax Regret

Regret: $$R_n(\hat{\mathbf{p}}; \mathcal{F}, \mathbf{x}, \mathbf{y}) = \sum_{t=1}^{n} \ell_{\log}(\hat{p}_t, y_t) - \inf_{f \in \mathcal{F}} \sum_{t=1}^{n} \ell_{\log}(f(x_t), y_t).$$

Minimax regret: an **algorithm-free quantity** on **worst-case observations**.

$$R_n(\mathcal{F}) = \sup_{x_1} \inf_{\hat{p}_1} \sup_{y_1} \; \sup_{x_2} \inf_{\hat{p}_2} \sup_{y_2} \; \cdots \; \boldsymbol{\sup_{x_n} \inf_{\hat{p}_n} \sup_{y_n}} R_n(\hat{\mathbf{p}}; \mathcal{F}, \mathbf{x}, \mathbf{y}).$$

This repeats for all n rounds.

Minimax Regret

Regret: $$R_n(\hat{\mathbf{p}}; \mathcal{F}, \mathbf{x}, \mathbf{y}) = \sum_{t=1}^{n} \ell_{\log}(\hat{p}_t, y_t) - \inf_{f \in \mathcal{F}} \sum_{t=1}^{n} \ell_{\log}(f(x_t), y_t).$$

Minimax regret: an **algorithm-free quantity** on **worst-case observations**.

$$R_n(\mathcal{F}) = \sup_{x_1} \inf_{\hat{p}_1} \sup_{y_1} \; \sup_{x_2} \inf_{\hat{p}_2} \sup_{y_2} \; \cdots \; \boldsymbol{\sup_{x_n} \inf_{\hat{p}_n} \sup_{y_n}} R_n(\hat{\mathbf{p}}; \mathcal{F}, \mathbf{x}, \mathbf{y}).$$

Interpretation: The experts $\mathcal{F}$ are *minimax online learnable* if $R_n(\mathcal{F}) < o(n)$.

- slow rate: $R_n(\mathcal{F}) = \Theta(\sqrt{n})$
- fast rate: $R_n(\mathcal{F}) \leq \mathcal{O}(\log(n))$

Covering Numbers

Goal: Obtain regret bounds using a notion of *complexity* of the expert class $\mathcal{F}$.

Covering Numbers

Goal: Obtain regret bounds using a notion of *complexity* of the expert class $\mathcal{F}$.

Covering Numbers

Covering Numbers

Goal: Obtain regret bounds using a notion of *complexity* of the expert class $\mathcal{F}$.

Covering Numbers

- Define a notion of distance between experts, $d(f, g)$.

Covering Numbers

Goal: Obtain regret bounds using a notion of *complexity* of the expert class $\mathcal{F}$.

Covering Numbers

- Define a notion of distance between experts, $d(f, g)$.
- Find the *smallest* $\mathcal{G} \subseteq \mathcal{F}$ so that for each $f \in \mathcal{F}$, there is a $g \in \mathcal{G}$ with $d(f, g) \leq \gamma$.

Covering Numbers

Goal: Obtain regret bounds using a notion of *complexity* of the expert class $\mathcal{F}$.

Covering Numbers

- Define a notion of distance between experts, $d(f, g)$.
- Find the *smallest* $\mathcal{G} \subseteq \mathcal{F}$ so that for each $f \in \mathcal{F}$, there is a $g \in \mathcal{G}$ with $d(f, g) \leq \gamma$.
- The covering number for $\mathcal{F}$ is $|\mathcal{G}|$, and the *entropy* is $\log(|\mathcal{G}|)$.

Covering Numbers

Goal: Obtain regret bounds using a notion of *complexity* of the expert class $\mathcal{F}$.

Covering Numbers

- Define a notion of distance between experts, $d(f, g)$.
- Find the *smallest* $\mathcal{G} \subseteq \mathcal{F}$ so that for each $f \in \mathcal{F}$, there is a $g \in \mathcal{G}$ with $d(f, g) \leq \gamma$.
- The covering number for $\mathcal{F}$ is $|\mathcal{G}|$, and the *entropy* is $\log(|\mathcal{G}|)$.

Uniform Covering

$$d(f, g) = \sup_{x \in \mathcal{X}} \sup_{y \in \{0,1\}} |\ell_{\log}(f(x), y) - \ell_{\log}(g(x), y)|$$

Covering Numbers

Goal: Obtain regret bounds using a notion of *complexity* of the expert class $\mathcal{F}$.

Covering Numbers

- Define a notion of distance between experts, $d(f, g)$.
- Find the *smallest* $\mathcal{G} \subseteq \mathcal{F}$ so that for each $f \in \mathcal{F}$, there is a $g \in \mathcal{G}$ with $d(f, g) \leq \gamma$.
- The covering number for $\mathcal{F}$ is $|\mathcal{G}|$, and the *entropy* is $\log(|\mathcal{G}|)$.

Uniform Covering

$$d(f, g) = \sup_{x \in \mathcal{X}} \sup_{y \in \{0,1\}} |\ell_{\log}(f(x), y) - \ell_{\log}(g(x), y)|$$

A uniform covering may be infinite for large expert classes.

Covering Numbers

Goal: Obtain regret bounds using a notion of *complexity* of the expert class $\mathcal{F}$.

Covering Numbers

- Define a notion of distance between experts, $d(f, g)$.
- Find the *smallest* $\mathcal{G} \subseteq \mathcal{F}$ so that for each $f \in \mathcal{F}$, there is a $g \in \mathcal{G}$ with $d(f, g) \leq \gamma$.
- The covering number for $\mathcal{F}$ is $|\mathcal{G}|$, and the *entropy* is $\log(|\mathcal{G}|)$.

Uniform Covering

$$d(f, g) = \sup_{x \in \mathcal{X}} \sup_{y \in \{0,1\}} |\ell_{\log}(f(x), y) - \ell_{\log}(g(x), y)|$$

A uniform covering may be infinite for large expert classes.

Instead, we use *sequential covering* from Rakhlin and Sridharan (2014).

Sequential Covering

Key characteristics of sequential covering:

- Only need to cover the expert predictions on the actual observed contexts.
- The cover respects the sequential dependency of the online game.

Sequential Covering

Key characteristics of sequential covering:

- Only need to cover the expert predictions on the actual observed contexts.
- The cover respects the sequential dependency of the online game.

$$R_n(\mathcal{F}) = \sup_{x_1} \inf_{\hat{p}_1} \sup_{y_1} \sup_{x_2} \inf_{\hat{p}_2} \sup_{y_2} \cdots \sup_{x_n} \inf_{\hat{p}_n} \sup_{y_n} R_n(\hat{\mathbf{p}}; \mathcal{F}, \mathbf{x}, \mathbf{y}).$$

We encode the sequential nature of x_t and y_t using **binary trees:**

Sequential Covering

Key characteristics of sequential covering:

- Only need to cover the expert predictions on the actual observed contexts.
- The cover respects the sequential dependency of the online game.

$$R_n(\mathcal{F}) = \sup_{x_1} \inf_{\hat{p}_1} \sup_{y_1} \ \sup_{x_2} \inf_{\hat{p}_2} \sup_{y_2} \ \cdots \ \sup_{x_n} \inf_{\hat{p}_n} \sup_{y_n} R_n(\hat{\mathbf{p}}; \mathcal{F}, \mathbf{x}, \mathbf{y}).$$

We encode the sequential nature of x_t and y_t using **binary trees:**

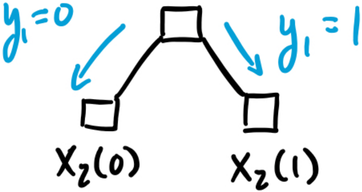
Sequential Covering

Key characteristics of sequential covering:

- Only need to cover the expert predictions on the actual observed contexts.
- The cover respects the sequential dependency of the online game.

$$R_n(\mathcal{F}) = \sup_{x_1} \inf_{\hat{p}_1} \sup_{y_1} \; \sup_{x_2} \inf_{\hat{p}_2} \sup_{y_2} \; \cdots \; \sup_{x_n} \inf_{\hat{p}_n} \sup_{y_n} R_n(\hat{\mathbf{p}}; \mathcal{F}, \mathbf{x}, \mathbf{y}).$$

We encode the sequential nature of x_t and y_t using **binary trees:**



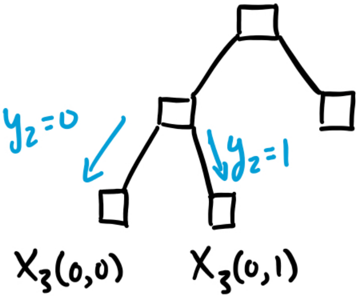
Sequential Covering

Key characteristics of sequential covering:

- Only need to cover the expert predictions on the actual observed contexts.
- The cover respects the sequential dependency of the online game.

$$R_n(\mathcal{F}) = \sup_{x_1} \inf_{\hat{p}_1} \sup_{y_1} \; \sup_{x_2} \inf_{\hat{p}_2} \sup_{y_2} \; \cdots \; \sup_{x_n} \inf_{\hat{p}_n} \sup_{y_n} R_n(\hat{\mathbf{p}}; \mathcal{F}, \mathbf{x}, \mathbf{y}).$$

We encode the sequential nature of x_t and y_t using **binary trees:**



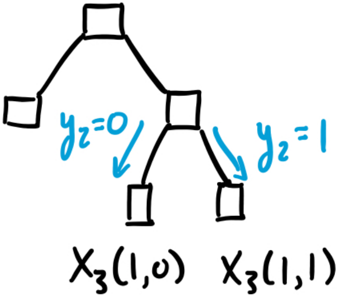
Sequential Covering

Key characteristics of sequential covering:

- Only need to cover the expert predictions on the actual observed contexts.
- The cover respects the sequential dependency of the online game.

$$R_n(\mathcal{F}) = \sup_{x_1} \inf_{\hat{p}_1} \sup_{y_1} \; \sup_{x_2} \inf_{\hat{p}_2} \sup_{y_2} \; \cdots \; \sup_{x_n} \inf_{\hat{p}_n} \sup_{y_n} R_n(\hat{\mathbf{p}}; \mathcal{F}, \mathbf{x}, \mathbf{y}).$$

We encode the sequential nature of x_t and y_t using **binary trees:**



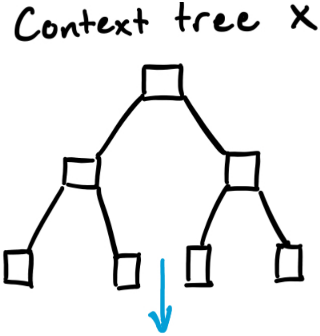
Sequential Covering

Key characteristics of sequential covering:

- Only need to cover the expert predictions on the actual observed contexts.
- The cover respects the sequential dependency of the online game.

$$R_n(\mathcal{F}) = \sup_{x_1} \inf_{\hat{p}_1} \sup_{y_1} \sup_{x_2} \inf_{\hat{p}_2} \sup_{y_2} \cdots \sup_{x_n} \inf_{\hat{p}_n} \sup_{y_n} R_n(\hat{\mathbf{p}}; \mathcal{F}, \mathbf{x}, \mathbf{y}).$$

We encode the sequential nature of x_t and y_t using **binary trees:**



Sequential Covering

A class of trees V sequentially covers $\mathcal{F}$ at margin γ on context tree $\mathbf{x}$ if:

$$\sup_{f \in \mathcal{F}} \sup_{\mathbf{y} \in \{0,1\}^n} \inf_{\mathbf{v} \in V} \sup_{t \in [n]} |f(x_t(\mathbf{y})) - v_t(\mathbf{y})| \leq \gamma.$$

Sequential Covering

A class of trees V sequentially covers $\mathcal{F}$ at margin γ on context tree $\mathbf{x}$ if:

$$\sup_{f \in \mathcal{F}} \sup_{\mathbf{y} \in \{0,1\}^n} \inf_{\mathbf{v} \in V} \sup_{t \in [n]} |f(x_t(\mathbf{y})) - v_t(\mathbf{y})| \leq \gamma.$$

Observations

- V is chosen after observing $\mathbf{x}$, so it doesn't have to apply to all of $\mathcal{X}$.
- $\mathbf{v} \in V$ is chosen with knowledge of $\mathbf{y}$, the actual path of observations.

Sequential Covering

A class of trees V sequentially covers $\mathcal{F}$ at margin γ on context tree $\mathbf{x}$ if:

$$\sup_{f \in \mathcal{F}} \sup_{\mathbf{y} \in \{0,1\}^n} \inf_{\mathbf{v} \in V} \sup_{t \in [n]} |f(x_t(\mathbf{y})) - v_t(\mathbf{y})| \leq \gamma.$$

Observations

- V is chosen after observing $\mathbf{x}$, so it doesn't have to apply to all of $\mathcal{X}$.
- $\mathbf{v} \in V$ is chosen with knowledge of $\mathbf{y}$, the actual path of observations.

Definitions

- The size of the smallest such V for $\mathbf{x}$ is $\mathcal{N}_\infty(\mathcal{F} \circ \mathbf{x}, \gamma)$.
- *Sequential entropy* for n rounds is $\mathcal{H}_\infty(\mathcal{F}, \gamma, n) = \sup_{\mathbf{x}} \log(\mathcal{N}_\infty(\mathcal{F} \circ \mathbf{x}, \gamma))$.

Improved Minimax Bounds

Theorem (BFR '20)

There exists $c > 0$ such that for all convex $\mathcal{F}$,

$$R_n(\mathcal{F}) \leq \inf_{\gamma > 0} \Big\{ 4n\gamma + c\,\mathcal{H}_\infty\left(\mathcal{F}, \gamma, n\right) \Big\}.$$

Improved Minimax Bounds

Theorem (BFR '20)

There exists $c > 0$ such that for all convex $\mathcal{F}$,

$$R_n(\mathcal{F}) \leq \inf_{\gamma > 0} \Big\{ 4n\gamma + c\,\mathcal{H}_\infty\left(\mathcal{F}, \gamma, n\right) \Big\}.$$

Upper Bound (Computation)

If $\mathcal{H}_\infty\left(\mathcal{F}, \gamma, n\right) = \Theta(\gamma^{-p})$ for $p > 0$,

$$R_n(\mathcal{F}) \leq \mathcal{O}(n^{\frac{p}{p+1}}).$$

Improved Minimax Bounds

Theorem (BFR '20)

There exists $c > 0$ *such that for all convex* $\mathcal{F}$*,*

$$R_n(\mathcal{F}) \leq \inf_{\gamma > 0} \Big\{ 4n\gamma + c\,\mathcal{H}_\infty(\mathcal{F}, \gamma, n) \Big\}.$$

Upper Bound (Computation)

If $\mathcal{H}_\infty(\mathcal{F}, \gamma, n) = \Theta(\gamma^{-p})$ for $p > 0$,

$$R_n(\mathcal{F}) \leq \mathcal{O}(n^{\frac{p}{p+1}}).$$

Theorem (BFR '20)

If $p \in \mathbb{N}$*, there exists an* $\mathcal{F}$ *with* $\mathcal{H}_\infty(\mathcal{F}, \gamma, n) = \Theta(\gamma^{-p})$ *and*

$$R_n(\mathcal{F}) \geq \Omega(n^{\frac{p}{p+1}}).$$

Applications

Applications

- **1-Lipschitz:**
  $\mathcal{F} = \{f \mid f : [0,1]^p \to [0,1], |f(x) - f(y)| \leq \|x - y\| \ \forall x, y \in [0,1]^p\}$.

$$\mathcal{H}_\infty\left(\mathcal{F}, \gamma, n\right) = \Theta\left(\gamma^{-p}\right).$$

Applications

- **1-Lipschitz:**
  $\mathcal{F} = \{f \mid f : [0,1]^p \to [0,1], |f(x) - f(y)| \leq \|x - y\| \ \forall x, y \in [0,1]^p\}$.

$$\mathcal{H}_\infty\left(\mathcal{F}, \gamma, n\right) = \Theta\left(\gamma^{-p}\right).$$

  We have matching upper and lower bounds for this class, so:

$$R_n(\mathcal{F}) = \Theta(n^{\frac{p}{p+1}}).$$

Applications

- **Linear Predictors:**
  $\mathcal{F} = \{f \mid \exists w \text{ s.t. } \|w\|_2 \leq 1, f(x) = \frac{1}{2}[1 + \langle w, x\rangle] \ \forall \|x\|_2 \leq 1\}$.

$$\mathcal{H}_\infty(\mathcal{F}, \gamma, n) = \tilde{\Theta}\left(\gamma^{-2}\right).$$

Applications

- **Linear Predictors:**
  $\mathcal{F} = \{f \mid \exists w \text{ s.t. } \|w\|_2 \leq 1, f(x) = \frac{1}{2}[1 + \langle w, x\rangle] \; \forall \|x\|_2 \leq 1\}$.

$$\mathcal{H}_\infty\left(\mathcal{F}, \gamma, n\right) = \tilde{\Theta}\left(\gamma^{-2}\right).$$

  Our upper bound prescribes:

$$R_n(\mathcal{F}) \leq \tilde{\mathcal{O}}(n^{2/3}).$$

Applications

- **Linear Predictors:**
  $\mathcal{F} = \{f \mid \exists w \text{ s.t. } \|w\|_2 \leq 1, f(x) = \frac{1}{2}[1 + \langle w, x \rangle] \; \forall \, \|x\|_2 \leq 1\}$.

  $$\mathcal{H}_\infty(\mathcal{F}, \gamma, n) = \tilde{\Theta}\left(\gamma^{-2}\right).$$

  Our upper bound prescribes:

  $$R_n(\mathcal{F}) \leq \tilde{\mathcal{O}}(n^{2/3}).$$

  However, Rakhlin & Sridharan (2015) showed (with an explicit algorithm)

  $$R_n(\mathcal{F}) \leq \tilde{\mathcal{O}}(\sqrt{n}).$$

Applications

- **Linear Predictors:**
  $\mathcal{F} = \{f \mid \exists w \text{ s.t. } \|w\|_2 \leq 1, f(x) = \frac{1}{2}[1 + \langle w, x \rangle] \; \forall \|x\|_2 \leq 1\}$.

  $$\mathcal{H}_\infty(\mathcal{F}, \gamma, n) = \tilde{\Theta}\left(\gamma^{-2}\right).$$

  Our upper bound prescribes:

  $$R_n(\mathcal{F}) \leq \tilde{\mathcal{O}}(n^{2/3}).$$

  However, Rakhlin & Sridharan (2015) showed (with an explicit algorithm)

  $$R_n(\mathcal{F}) \leq \tilde{\mathcal{O}}(\sqrt{n}).$$

Our upper bound cannot be improved, so the minimax regret under log loss cannot be characterized solely by sequential entropy.