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

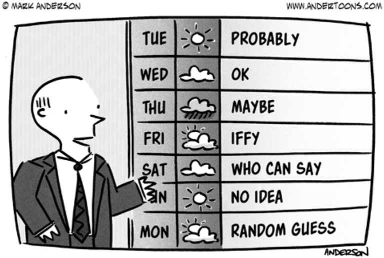
Blair Bilodeau[1,2]
with Dylan J. Foster[3] and Daniel M. Roy[1,2]

March 11, 2020

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

Motivation

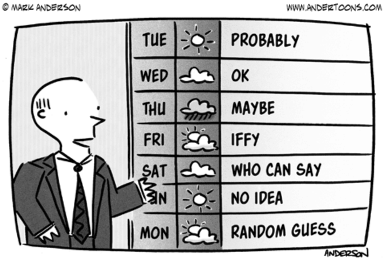
Weather Forecasting



"And now the 7-day forecast..."

Goal: forecast the probability of rain from historical data and current conditions.

Weather Forecasting



"And now the 7-day forecast..."

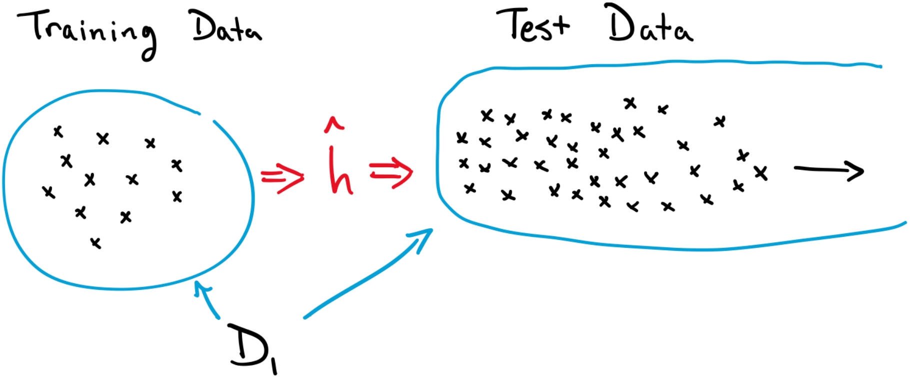
Goal: forecast the probability of rain from historical data and current conditions.

Considerations

- Which assumptions to make about historical trends continuing?
- How many physical relationships should be incorporated in the model?
- Are some missed predictions more expensive than others?

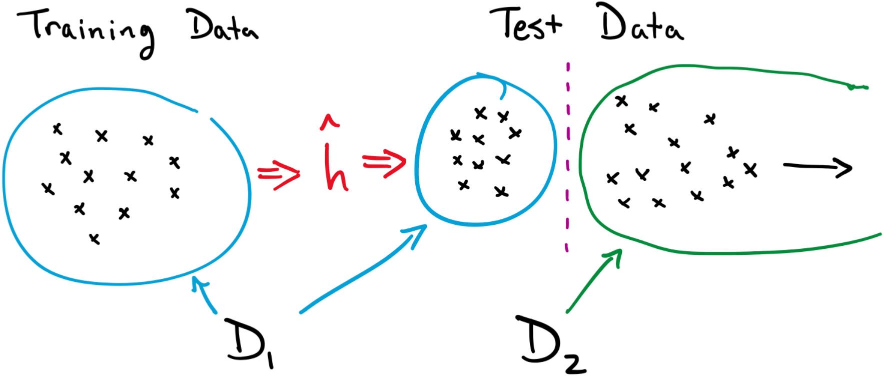
Traditional Statistical Learning

- Receive a batch of data
- Estimate a prediction function $\hat{h}$
- Evaluate performance on new data assumed to be from the same distribution



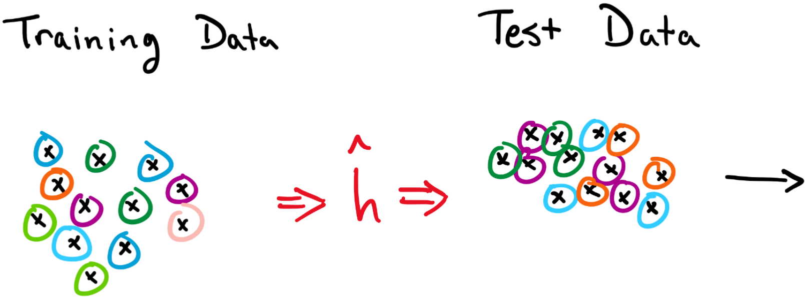
Traditional Statistical Learning

But what if there's a changepoint...



Traditional Statistical Learning

...or your training data isn't even i.i.d.?



Statistical Solutions

We want to remove assumptions about the data generating process.

In particular, **future data may not be i.i.d. with past data**.

Statistical Solutions

We want to remove assumptions about the data generating process.
In particular, **future data may not be i.i.d. with past data**.

Statistics does this with, for example,

- Markov assumption
- stationarity assumption (time series)
- covariance structure assumption (e.g., Gaussian process)

Statistical Solutions

We want to remove assumptions about the data generating process.
In particular, **future data may not be i.i.d. with past data**.

Statistics does this with, for example,

- Markov assumption
- stationarity assumption (time series)
- covariance structure assumption (e.g., Gaussian process)

But these assumptions are often **uncheckable** or **false**.

Online Learning

Online Learning

A framework where **the past may not be indicative of the future**.

Online Learning

A framework where **the past may not be indicative of the future**.

Online Learning

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

- Predict $\hat{y}_t \in \hat{\mathcal{Y}}$
- Observe $y_t \in \mathcal{Y}$
- Incur loss $\ell(\hat{y}_t, y_t)$

Online Learning

A framework where **the past may not be indicative of the future**.

Online Learning

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

- Predict $\hat{y}_t \in \hat{\mathcal{Y}}$
- Observe $y_t \in \mathcal{Y}$ $\longleftarrow$ We do not assume this is generated by a model
- Incur loss $\ell(\hat{y}_t, y_t)$

Online Learning

A framework where **the past may not be indicative of the future**.

Contextual Online Learning

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

- Observe context $x_t \in \mathcal{X}$
- Predict $\hat{y}_t \in \hat{\mathcal{Y}}$
- Observe $y_t \in \mathcal{Y}$ $\longleftarrow$ We do not assume this is generated by a model
- Incur loss $\ell(\hat{y}_t, y_t)$

Online Learning

A framework where **the past may not be indicative of the future**.

Contextual Online Learning

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

- Observe context $x_t \in \mathcal{X}$ $\longleftarrow$ Also has no model assumptions
- Predict $\hat{y}_t \in \hat{\mathcal{Y}}$
- Observe $y_t \in \mathcal{Y}$ $\longleftarrow$ We do not assume this is generated by a model
- Incur loss $\ell(\hat{y}_t, y_t)$

Measuring Performance

In statistical learning, performance is often measured against:

- a ground truth, e.g., parameter estimation
- the best predictor from some class for the underlying probability model

Measuring Performance

In statistical learning, performance is often measured against:

- a ground truth, e.g., parameter estimation
- the best predictor from some class for the underlying probability model

These measures quantify **guarantees about the future given the past**.
Without a probabilistic model:

- no notion of ground truth to compare with
- the "best hypothesis" in a class is not clearly defined
- cannot naively hope to do well on future observations

Measuring Performance

In statistical learning, performance is often measured against:

- a ground truth, e.g., parameter estimation
- the best predictor from some class for the underlying probability model

These measures quantify **guarantees about the future given the past**.
Without a probabilistic model:

- no notion of ground truth to compare with
- the "best hypothesis" in a class is not clearly defined
- cannot naively hope to do well on future observations

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

Measuring Performance

In statistical learning, performance is often measured against:

- a ground truth, e.g., parameter estimation
- the best predictor from some class for the underlying probability model

These measures quantify **guarantees about the future given the past**.
Without a probabilistic model:

- no notion of ground truth to compare with
- the "best hypothesis" in a class is not clearly defined
- cannot naively hope to do well on future observations

Consider a relative notion of performance in hindsight.

- Relative to a class $\mathcal{F} \subseteq \{f : \mathcal{X} \to \hat{\mathcal{Y}}\}$, consisting of **experts** $f \in \mathcal{F}$.
- Compete against the optimal $f \in \mathcal{F}$ on the actual sequence of observations from past rounds.

Regret

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

Regret

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

This quantity depends on

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

Minimax Regret

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

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

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

Minimax Regret

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

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

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

The first context is observed.

Minimax Regret

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

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

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

The player makes their prediction.

Minimax Regret

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

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

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

The adversary plays an observation.

Minimax Regret

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

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

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

This repeats for all n rounds.

Minimax Regret

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

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

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

This repeats for all n rounds.

Minimax Regret

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

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

$$R_n^\ell(\mathcal{F}) = \left\langle\!\!\left\langle \sup_{x_t} \inf_{\hat{y}_t} \sup_{y_t} \right\rangle\!\!\right\rangle_{t=1}^{n} R_n^\ell(\hat{\mathbf{y}}; \mathcal{F}, \mathbf{x}, \mathbf{y}).$$

The notation $\langle\!\langle \cdot \rangle\!\rangle_{t=1}^{n}$ denotes repeated application of operators.

Minimax Regret

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

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

$$R_n^\ell(\mathcal{F}) = \left\langle\!\!\left\langle \sup_{x_t} \inf_{\hat{y}_t} \sup_{y_t} \right\rangle\!\!\right\rangle_{t=1}^n R_n^\ell(\hat{\mathbf{y}}; \mathcal{F}, \mathbf{x}, \mathbf{y}).$$

Interpretation: The tuple $(\ell, \mathcal{F})$ is *online learnable* if $R_n^\ell(\mathcal{F}) < o(n)$.

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

Logarithmic Loss

Problem Formulation

Sequential Probability Assignment

In each round, the prediction is a distribution on possible observations.

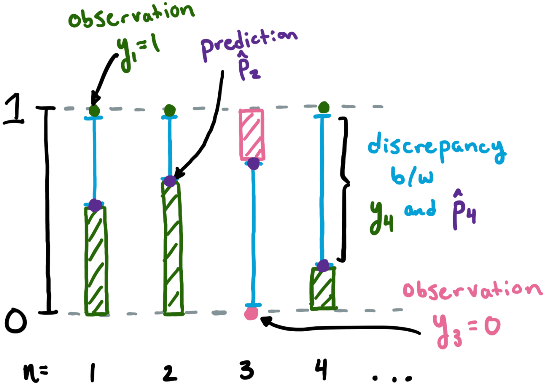
Problem Formulation

Sequential Probability Assignment

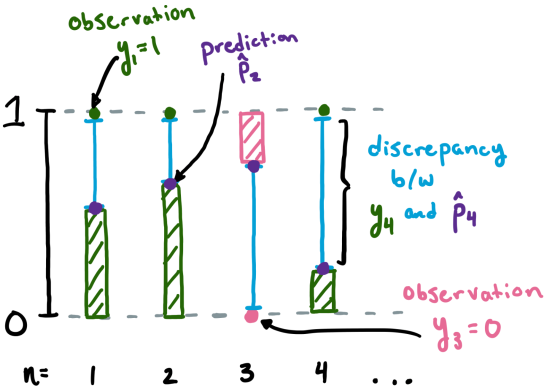
In each round, the prediction is a distribution on possible observations.

Predicting Binary Outcomes

$y \in \mathcal{Y} = \{0, 1\}$ and $\hat{p} \in \hat{\mathcal{Y}} \equiv [0, 1]$

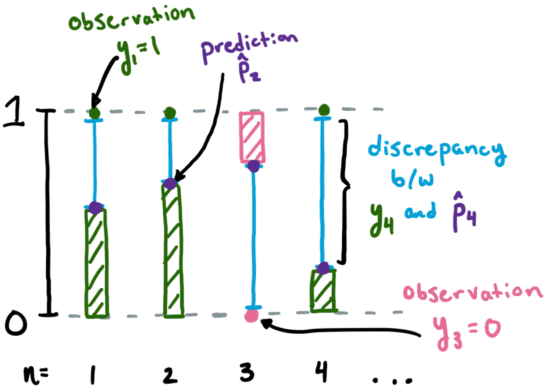


Measuring Loss



What is the correct notion of loss?

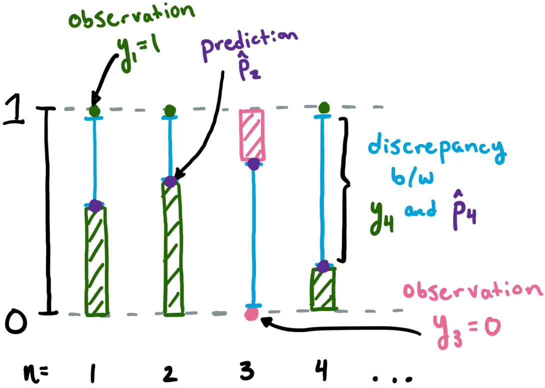
Measuring Loss



Intuition: being confidently wrong is much worse than being indecisive.

Statistical motivation: maximum likelihood estimation for a Bernoulli.

Measuring Loss



Intuition: being confidently wrong is much worse than being indecisive.
Statistical motivation: maximum likelihood estimation for a Bernoulli.

Logarithmic Loss

$$\ell_{\log}(\hat{p}_t, y_t) = -y_t \log(\hat{p}_t) - (1 - y_t) \log(1 - \hat{p}_t).$$

Measuring Loss

Why is this difficult?

Standard online learning techniques rely on loss being bounded or Lipschitz.

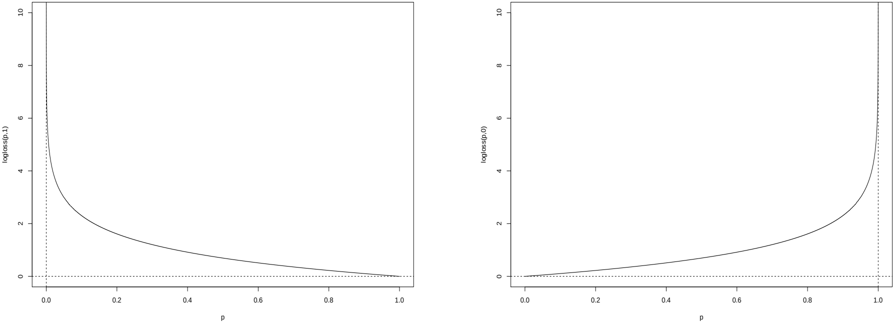
Measuring Loss

Why is this difficult?

Standard online learning techniques rely on loss being bounded or Lipschitz.

$$\ell_{\log}(\hat{p}_t, y_t) = -y_t \log(\hat{p}_t) - (1 - y_t) \log(1 - \hat{p}_t).$$

$y = 1$ $\quad\quad$ $y = 0$



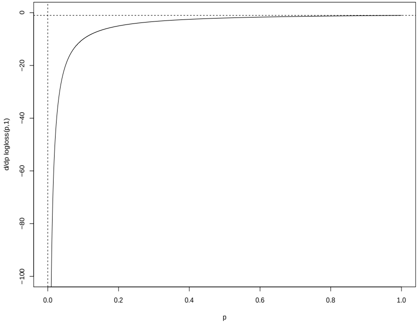
Measuring Loss

Why is this difficult?

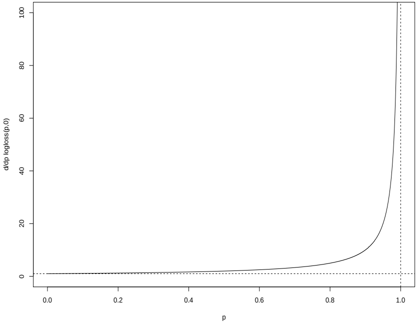
Standard online learning techniques rely on loss being bounded or Lipschitz.

$$\ell_{\log}(\hat{p}_t, y_t) = -y_t \log(\hat{p}_t) - (1 - y_t) \log(1 - \hat{p}_t).$$

$y = 1$



$y = 0$



Bounding Regret

Dual Game

Recall that the minimax regret is

$$R_n^{\log}(\mathcal{F}) = \left\langle\!\!\left\langle \sup_{x_t} \inf_{\hat{p}_t} \sup_{y_t} \right\rangle\!\!\right\rangle_{t=1}^{n} R_n^{\log}(\hat{\mathbf{p}}; \mathcal{F}, \mathbf{x}, \mathbf{y}).$$

Dual Game

Recall that the minimax regret is

$$R_n^{\log}(\mathcal{F}) = \left\langle\!\!\left\langle \sup_{x_t} \inf_{\hat{p}_t} \sup_{y_t} \right\rangle\!\!\right\rangle_{t=1}^{n} R_n^{\log}(\hat{\mathbf{p}}; \mathcal{F}, \mathbf{x}, \mathbf{y}).$$

The worst-case observations can equivalently be viewed as

$$R_n^{\log}(\mathcal{F}) = \left\langle\!\!\left\langle \sup_{x_t} \inf_{\hat{p}_t} \sup_{p_t} \mathop{\mathbb{E}}_{y_t \sim p_t} \right\rangle\!\!\right\rangle_{t=1}^{n} R_n^{\log}(\hat{\mathbf{p}}; \mathcal{F}, \mathbf{x}, \mathbf{y}).$$

Dual Game

Recall that the minimax regret is

$$R_n^{\log}(\mathcal{F}) = \left\langle\!\!\left\langle \sup_{x_t} \inf_{\hat{p}_t} \sup_{y_t} \right\rangle\!\!\right\rangle_{t=1}^{n} R_n^{\log}(\hat{\mathbf{p}}; \mathcal{F}, \mathbf{x}, \mathbf{y}).$$

The worst-case observations can equivalently be viewed as

$$R_n^{\log}(\mathcal{F}) = \left\langle\!\!\left\langle \sup_{x_t} \inf_{\hat{p}_t} \sup_{p_t} \mathop{\mathbb{E}}_{y_t \sim p_t} \right\rangle\!\!\right\rangle_{t=1}^{n} R_n^{\log}(\hat{\mathbf{p}}; \mathcal{F}, \mathbf{x}, \mathbf{y}).$$

(Abernethy et al., 2009, Rakhlin and Sridharan, 2015)
An extension of the minimax theorem gives

$$R_n^{\log}(\mathcal{F}) = \left\langle\!\!\left\langle \sup_{x_t} \sup_{p_t} \mathop{\mathbb{E}}_{y_t \sim p_t} \right\rangle\!\!\right\rangle_{t=1}^{n} R_n^{\log}(\mathbf{p}; \mathcal{F}, \mathbf{x}, \mathbf{y}).$$

Empirical Process Theory

Expanding the regret term, we get

$$R_n^{\log}(\mathcal{F}) = \left\langle\!\!\left\langle \sup_{x_t} \sup_{p_t} \mathop{\mathbb{E}}_{y_t \sim p_t} \right\rangle\!\!\right\rangle_{t=1}^{n} \sup_{f \in \mathcal{F}} \left[\sum_{t=1}^{n} \ell_{\log}(p_t, y_t) - \ell_{\log}(f(x_t), y_t) \right].$$

Empirical Process Theory

Expanding the regret term, we get

$$R_n^{\log}(\mathcal{F}) = \left\langle\!\!\left\langle \sup_{x_t} \sup_{p_t} \mathop{\mathbb{E}}_{y_t \sim p_t} \right\rangle\!\!\right\rangle_{t=1}^{n} \sup_{f \in \mathcal{F}} \left[\sum_{t=1}^{n} \ell_{\log}(p_t, y_t) - \ell_{\log}(f(x_t), y_t) \right].$$

The presence of an **expected supremum** suggests empirical process theory.

Empirical Process Theory

Expanding the regret term, we get

$$R_n^{\log}(\mathcal{F}) = \left\langle\!\!\left\langle \sup_{x_t} \sup_{p_t} \mathop{\mathbb{E}}_{y_t \sim p_t} \right\rangle\!\!\right\rangle_{t=1}^{n} \sup_{f \in \mathcal{F}} \left[\sum_{t=1}^{n} \ell_{\log}(p_t, y_t) - \ell_{\log}(f(x_t), y_t) \right].$$

The presence of an **expected supremum** suggests empirical process theory.

- Discretize the infinite supremum into a finite cover.
- Bound the expected maximum of the finite cover.
- Bound the error from only considering the finite cover.

Uniform Covering Fails

Early work (Cesa-Bianchi and Lugosi, 1999, Opper and Haussler, 1999) used a uniform covering approach, but this is **too coarse for many expert classes**.

Uniform Covering Fails

Early work (Cesa-Bianchi and Lugosi, 1999, Opper and Haussler, 1999) used a uniform covering approach, but this is **too coarse for many expert classes**.

Distance between $f, g \in \mathcal{F}$:

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

Uniform Covering Fails

Early work (Cesa-Bianchi and Lugosi, 1999, Opper and Haussler, 1999) used a uniform covering approach, but this is **too coarse for many expert classes**.

Distance between $f, g \in \mathcal{F}$:

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

Class $\mathcal{G}$ covers class $\mathcal{F}$ at margin γ if:

$$\sup_{f \in \mathcal{F}} \inf_{g \in \mathcal{G}} d(f, g) \le \gamma.$$

Uniform Covering Fails

Early work (Cesa-Bianchi and Lugosi, 1999, Opper and Haussler, 1999) used a uniform covering approach, but this is **too coarse for many expert classes**.

Distance between $f, g \in \mathcal{F}$:

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

Class $\mathcal{G}$ covers class $\mathcal{F}$ at margin γ if:

$$\sup_{f \in \mathcal{F}} \inf_{g \in \mathcal{G}} d(f,g) \leq \gamma.$$

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

Binary Tree Notation

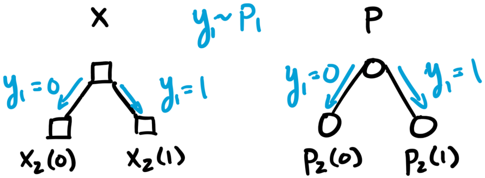
$$R_n^{\log}(\mathcal{F}) = \left\langle\!\!\left\langle \sup_{x_t} \sup_{p_t} \mathop{\mathbb{E}}_{y_t \sim p_t} \right\rangle\!\!\right\rangle_{t=1}^{n} \sup_{f \in \mathcal{F}} \left[\sum_{t=1}^{n} \ell_{\log}(p_t, y_t) - \ell_{\log}(f(x_t), y_t) \right].$$

We can encode the sequential nature of x_t and p_t using binary trees:

Binary Tree Notation

$$R_n^{\log}(\mathcal{F}) = \left\langle\!\!\left\langle \sup_{x_t} \sup_{p_t} \mathop{\mathbb{E}}_{y_t \sim p_t} \right\rangle\!\!\right\rangle_{t=1}^{n} \sup_{f \in \mathcal{F}} \left[\sum_{t=1}^{n} \ell_{\log}(p_t, y_t) - \ell_{\log}(f(x_t), y_t) \right].$$

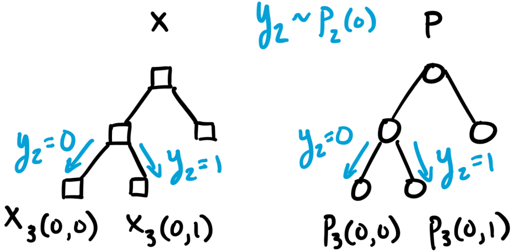
We can encode the sequential nature of x_t and p_t using binary trees:



Binary Tree Notation

$$R_n^{\log}(\mathcal{F}) = \left\langle\!\!\left\langle \sup_{x_t} \sup_{p_t} \mathop{\mathbb{E}}_{y_t \sim p_t} \right\rangle\!\!\right\rangle_{t=1}^{n} \sup_{f \in \mathcal{F}} \left[\sum_{t=1}^{n} \ell_{\log}(p_t, y_t) - \ell_{\log}(f(x_t), y_t) \right].$$

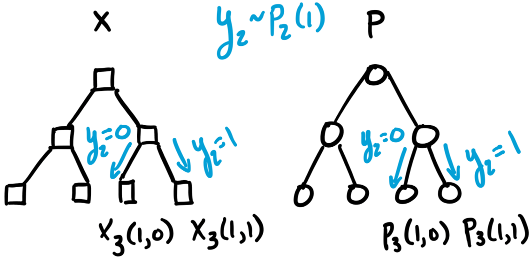
We can encode the sequential nature of x_t and p_t using binary trees:



Binary Tree Notation

$$R_n^{\log}(\mathcal{F}) = \left\langle\!\!\left\langle \sup_{x_t} \sup_{p_t} \mathop{\mathbb{E}}_{y_t \sim p_t} \right\rangle\!\!\right\rangle_{t=1}^{n} \sup_{f \in \mathcal{F}} \left[\sum_{t=1}^{n} \ell_{\log}(p_t, y_t) - \ell_{\log}(f(x_t), y_t) \right].$$

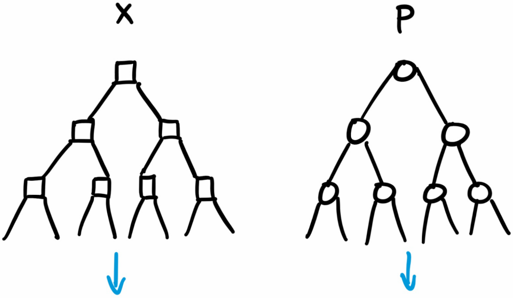
We can encode the sequential nature of x_t and p_t using binary trees:



Binary Tree Notation

$$R_n^{\log}(\mathcal{F}) = \left\langle\!\!\left\langle \sup_{x_t} \sup_{p_t} \mathop{\mathbb{E}}_{y_t \sim p_t} \right\rangle\!\!\right\rangle_{t=1}^{n} \sup_{f \in \mathcal{F}} \left[\sum_{t=1}^{n} \ell_{\log}(p_t, y_t) - \ell_{\log}(f(x_t), y_t) \right].$$

We can encode the sequential nature of x_t and p_t using binary trees:



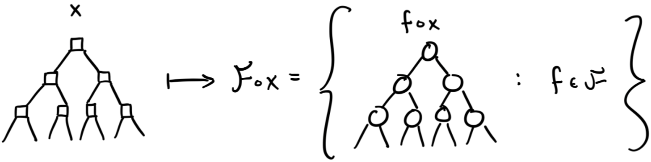
Sequential Covering

$$R_n^{\log}(\mathcal{F}) = \sup_{\mathbf{x}} \sup_{\mathbf{p}} \mathop{\mathbb{E}}_{\mathbf{y}\sim\mathbf{p}} \sup_{f\in\mathcal{F}} \left[\sum_{t=1}^{n} \ell_{\log}(p_t(\mathbf{y}), y_t) - \ell_{\log}(f(x_t(\mathbf{y})), y_t) \right].$$

Sequential Covering

$$R_n^{\log}(\mathcal{F}) = \sup_{\mathbf{x}} \sup_{\mathbf{p}} \mathop{\mathbb{E}}_{\mathbf{y}\sim\mathbf{p}} \sup_{f\in\mathcal{F}} \left[\sum_{t=1}^{n} \ell_{\log}(p_t(\mathbf{y}), y_t) - \ell_{\log}(f(x_t(\mathbf{y})), y_t) \right].$$

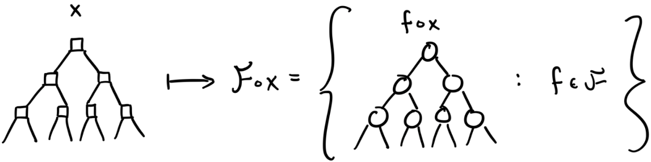
Cover the class of trees $\mathcal{F} \circ \mathbf{x}$ defined by composing $\mathcal{F}$ with a context tree $\mathbf{x}$:



Sequential Covering

$$R_n^{\log}(\mathcal{F}) = \sup_{\mathbf{x}} \sup_{\mathbf{p}} \mathop{\mathbb{E}}_{\mathbf{y}\sim\mathbf{p}} \sup_{f\in\mathcal{F}} \left[\sum_{t=1}^{n} \ell_{\log}(p_t(\mathbf{y}), y_t) - \ell_{\log}(f(x_t(\mathbf{y})), y_t) \right].$$

Cover the class of trees $\mathcal{F} \circ \mathbf{x}$ defined by composing $\mathcal{F}$ with a context tree $\mathbf{x}$:



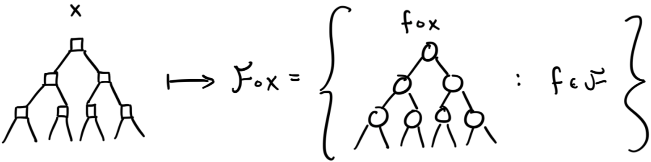
A class of trees V sequentially covers $\mathcal{F} \circ \mathbf{x}$ at margin γ if:

$$\sup_{\mathbf{u}\in\mathcal{F}\circ\mathbf{x}} \; \sup_{\mathbf{y}\in\{0,1\}^n} \; \inf_{\mathbf{v}\in V} \|\mathbf{u}(\mathbf{y}) - \mathbf{v}(\mathbf{y})\|_p \leq \gamma.$$

Sequential Covering

$$R_n^{\log}(\mathcal{F}) = \sup_{\mathbf{x}} \sup_{\mathbf{p}} \mathop{\mathbb{E}}_{\mathbf{y}\sim\mathbf{p}} \sup_{f\in\mathcal{F}} \left[\sum_{t=1}^{n} \ell_{\log}(p_t(\mathbf{y}), y_t) - \ell_{\log}(f(x_t(\mathbf{y})), y_t) \right].$$

Cover the class of trees $\mathcal{F} \circ \mathbf{x}$ defined by composing $\mathcal{F}$ with a context tree $\mathbf{x}$:



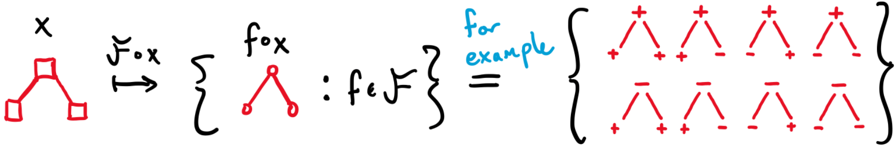
A class of trees V sequentially covers $\mathcal{F} \circ \mathbf{x}$ at margin γ if:

$$\sup_{\mathbf{u}\in\mathcal{F}\circ\mathbf{x}} \sup_{\mathbf{y}\in\{0,1\}^n} \inf_{\mathbf{v}\in V} \|\mathbf{u}(\mathbf{y}) - \mathbf{v}(\mathbf{y})\|_p \leq \gamma.$$

The order of observations and covering elements is **reversed from a uniform cover**.

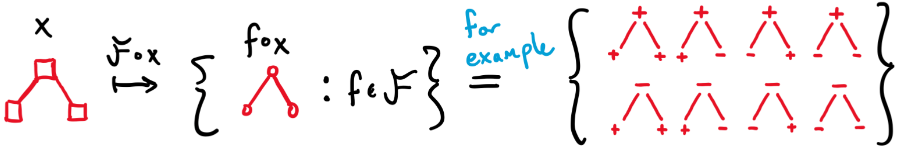
Sequential Covering Example

To illustrate the utility of sequential covering, consider binary experts for $n = 2$:



Sequential Covering Example

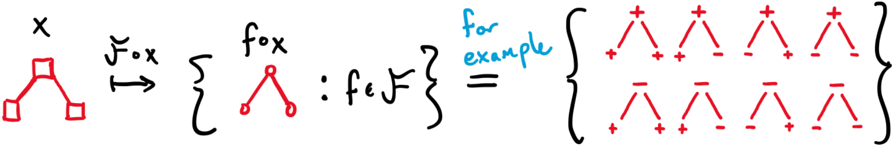
To illustrate the utility of sequential covering, consider binary experts for $n = 2$:



The only uniform cover of $\mathcal{F} \circ \mathbf{x}$ is itself, which has 8 elements.

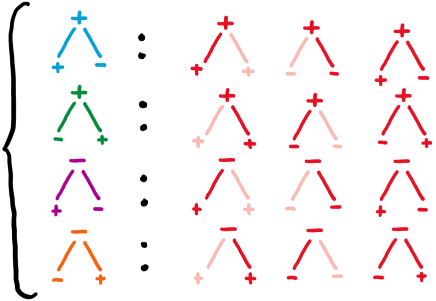
Sequential Covering Example

To illustrate the utility of sequential covering, consider binary experts for $n = 2$:



The only uniform cover of $\mathcal{F} \circ \mathbf{x}$ is itself, which has 8 elements.

For a sequential cover, we can choose **a different element for each path**, so only 4 trees are required.



Sequential Covering Examples

Examples of sequential covering numbers:

Sequential Covering Examples

Examples of sequential covering numbers:

- **Time-Invariant:** $\mathcal{F} = \{f \mid \exists q \in [0,1] \text{ s.t. } f(x) = q \ \forall x \in \mathcal{X}\}$.

$$\sup_{\mathbf{x}} \log\left(\mathcal{N}_\infty\left(\mathcal{F} \circ \mathbf{x}, \gamma\right)\right) \le \log(1/\gamma).$$

Sequential Covering Examples

Examples of sequential covering numbers:

- **Time-Invariant:** $\mathcal{F} = \{f \mid \exists q \in [0,1] \text{ s.t. } f(x) = q \ \forall x \in \mathcal{X}\}$.

$$\sup_{\mathbf{x}} \log\left(\mathcal{N}_\infty\left(\mathcal{F} \circ \mathbf{x}, \gamma\right)\right) \leq \log(1/\gamma).$$

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

$$\sup_{\mathbf{x}} \log\left(\mathcal{N}_\infty\left(\mathcal{F} \circ \mathbf{x}, \gamma\right)\right) = 1/\gamma^2.$$

Sequential Covering Examples

Examples of sequential covering numbers:

- **Time-Invariant:** $\mathcal{F} = \{f \mid \exists q \in [0,1] \text{ s.t. } f(x) = q \ \forall x \in \mathcal{X}\}$.

$$\sup_{\mathbf{x}} \log\left(\mathcal{N}_\infty\left(\mathcal{F} \circ \mathbf{x}, \gamma\right)\right) \leq \log(1/\gamma).$$

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

$$\sup_{\mathbf{x}} \log\left(\mathcal{N}_\infty\left(\mathcal{F} \circ \mathbf{x}, \gamma\right)\right) = 1/\gamma^2.$$

- **1-Lipschitz:** $\mathcal{F} = \{f \mid f : \mathbb{R}^d \to [0,1], \|\nabla f(x)\|_\infty \leq 1\}$.

$$\sup_{\mathbf{x}} \log\left(\mathcal{N}_\infty\left(\mathcal{F} \circ \mathbf{x}, \gamma\right)\right) = 1/\gamma^d.$$

Improved Minimax Bounds

Improved Minimax Bounds

Theorem (B., Foster, Roy, 2020)

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

$$R_n^{\log}(\mathcal{F}) \leq \sup_{\mathbf{x}} \inf_{\gamma > 0} \left\{ 4n\gamma + c \log\left(\mathcal{N}_\infty\left(\mathcal{F} \circ \mathbf{x}, \gamma\right)\right)\right\}.$$

Improved Minimax Bounds

Theorem (B., Foster, Roy, 2020)

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

$$R_n^{\log}(\mathcal{F}) \leq \sup_{\mathbf{x}} \inf_{\gamma > 0} \left\{ 4n\gamma + c \log\left(\mathcal{N}_\infty\left(\mathcal{F} \circ \mathbf{x}, \gamma\right)\right) \right\}.$$

Upper Bound (Computation)

If $\sup_{\mathbf{x}} \log\left(\mathcal{N}_\infty\left(\mathcal{F} \circ \mathbf{x}, \gamma\right)\right) \asymp \gamma^{-p}$,

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

Improved Minimax Bounds

Theorem (B., Foster, Roy, 2020)

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

$$R_n^{\log}(\mathcal{F}) \leq \sup_{\mathbf{x}} \inf_{\gamma > 0} \left\{ 4n\gamma + c \log\left(\mathcal{N}_\infty\left(\mathcal{F} \circ \mathbf{x}, \gamma\right)\right)\right\}.$$

Upper Bound (Computation)

If $\sup_{\mathbf{x}} \log\left(\mathcal{N}_\infty\left(\mathcal{F} \circ \mathbf{x}, \gamma\right)\right) \asymp \gamma^{-p}$,

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

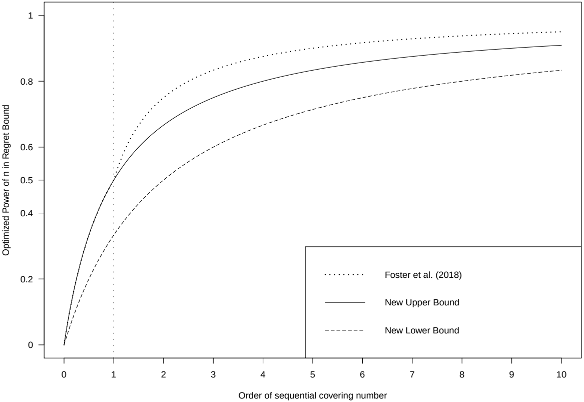
Theorem (B., Foster, Roy, 2020)

If $p > 0$*, there exists an* $\mathcal{F}$ *with* $\sup_{\mathbf{x}} \log\left(\mathcal{N}_\infty\left(\mathcal{F} \circ \mathbf{x}, \gamma\right)\right) \asymp \gamma^{-p}$ *and*

$$R_n^{\log}(\mathcal{F}) \geq \Omega\left(n^{\frac{p}{p+2}}\right).$$

Improved Minimax Bounds Visualized

Our results compared to the previous best upper bound from Foster et al. (2018).



Advances Underlying Results

Truncation Free

The standard procedure to control log loss uses truncation.

Define the truncated expert class $\mathcal{F}^\delta = \{f^\delta : f \in \mathcal{F}\}$ for $\delta \in (0, 1/2)$, where

$$f^\delta(x) = \begin{cases} \delta & f(x) < \delta \\ f(x) & \delta \le f(x) \le 1 - \delta \\ 1 - \delta & f(x) > 1 - \delta \end{cases} .$$

Truncation Free

The standard procedure to control log loss uses truncation.

Define the truncated expert class $\mathcal{F}^\delta = \{f^\delta : f \in \mathcal{F}\}$ for $\delta \in (0, 1/2)$, where

$$f^\delta(x) = \begin{cases} \delta & f(x) < \delta \\ f(x) & \delta \leq f(x) \leq 1 - \delta \\ 1 - \delta & f(x) > 1 - \delta \end{cases} .$$

- Observe that for $p \in [\delta, 1 - \delta]$, $\ell_{\log}(p, y)$ is $1/\delta$-Lipschitz.

Truncation Free

The standard procedure to control log loss uses truncation.

Define the truncated expert class $\mathcal{F}^\delta = \{f^\delta : f \in \mathcal{F}\}$ for $\delta \in (0, 1/2)$, where

$$f^\delta(x) = \begin{cases} \delta & f(x) < \delta \\ f(x) & \delta \le f(x) \le 1 - \delta \\ 1 - \delta & f(x) > 1 - \delta \end{cases} .$$

- Observe that for $p \in [\delta, 1-\delta]$, $\ell_{\log}(p, y)$ is $1/\delta$-Lipschitz.
- It can be shown that $R_n^{\log}(\mathcal{F}) \le R_n^{\log}(\mathcal{F}^\delta) + 2n\delta$.

Truncation Free

The standard procedure to control log loss uses truncation.

Define the truncated expert class $\mathcal{F}^\delta = \{f^\delta : f \in \mathcal{F}\}$ for $\delta \in (0, 1/2)$, where

$$f^\delta(x) = \begin{cases} \delta & f(x) < \delta \\ f(x) & \delta \le f(x) \le 1-\delta \\ 1-\delta & f(x) > 1-\delta \end{cases} .$$

- Observe that for $p \in [\delta, 1-\delta]$, $\ell_{\log}(p, y)$ is $1/\delta$-Lipschitz.
- It can be shown that $R_n^{\log}(\mathcal{F}) \le R_n^{\log}(\mathcal{F}^\delta) + 2n\delta$.

Rakhlin and Sridharan (2015) hypothesize this truncation argument is suboptimal, and pose the open problem of finding a tighter bound without it.

Truncation Free

The standard procedure to control log loss uses truncation.

Define the truncated expert class $\mathcal{F}^\delta = \{f^\delta : f \in \mathcal{F}\}$ for $\delta \in (0, 1/2)$, where

$$f^\delta(x) = \begin{cases} \delta & f(x) < \delta \\ f(x) & \delta \leq f(x) \leq 1 - \delta \\ 1 - \delta & f(x) > 1 - \delta \end{cases} .$$

- Observe that for $p \in [\delta, 1-\delta]$, $\ell_{\log}(p, y)$ is $1/\delta$-Lipschitz.
- It can be shown that $R_n^{\log}(\mathcal{F}) \leq R_n^{\log}(\mathcal{F}^\delta) + 2n\delta$.

Rakhlin and Sridharan (2015) hypothesize this truncation argument is suboptimal, and pose the open problem of finding a tighter bound without it.

Our argument does not require truncation.

Self-Concordance

Self-Concordant (Nesterov and Nemirovski, 1994)
A function $F : \mathbb{R} \to \mathbb{R}$ is self-concordant if

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

Self-Concordance

Self-Concordant (Nesterov and Nemirovski, 1994)
A function $F : \mathbb{R} \to \mathbb{R}$ is self-concordant if

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

Logarithmic loss is self-concordant as a function of p.

Self-Concordance

Self-Concordant (Nesterov and Nemirovski, 1994)
A function $F : \mathbb{R} \to \mathbb{R}$ is self-concordant if

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

Logarithmic loss is self-concordant as a function of p.

Utility: In convex optimization, encoding the constraint boundary with a *self-concordant barrier function* leads to polynomial iterations for high accuracy.

Self-Concordance

Self-Concordant (Nesterov and Nemirovski, 1994)
A function $F : \mathbb{R} \to \mathbb{R}$ is self-concordant if

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

Logarithmic loss is self-concordant as a function of p.

Utility: In convex optimization, encoding the constraint boundary with a *self-concordant barrier function* leads to polynomial iterations for high accuracy.

If F is self-concordant, then $\forall x, y \in \mathbb{R}$

$$F(x) - F(y) \leq (x-y)F'(x) - |x-y|\sqrt{F''(x)} + \log\left(1 + |x-y|\sqrt{F''(x)}\right).$$

Self-Concordance

Self-Concordant (Nesterov and Nemirovski, 1994)
A function $F : \mathbb{R} \to \mathbb{R}$ is self-concordant if

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

Logarithmic loss is self-concordant as a function of p.

Utility: In convex optimization, encoding the constraint boundary with a *self-concordant barrier function* leads to polynomial iterations for high accuracy.

If F is self-concordant, then $\forall x, y \in \mathbb{R}$

$$F(x) - F(y) \leq (x - y)F'(x) - |x - y| \sqrt{F''(x)} + \log\left(1 + |x - y| \sqrt{F''(x)}\right).$$

We use the second term to **control the gradient of logarithmic loss**.

Chaining Free

Recall our upper bound:

$$R_n^{\log}(\mathcal{F}) \leq \sup_{\mathbf{x}} \inf_{\gamma>0} \left\{4n\gamma + c\log\left(\mathcal{N}_\infty\left(\mathcal{F}\circ\mathbf{x},\gamma\right)\right)\right\}.$$

Chaining Free

Recall our upper bound:

$$R_n^{\log}(\mathcal{F}) \leq \sup_{\mathbf{x}} \inf_{\gamma > 0} \left\{ 4n\gamma + c \log\left(\mathcal{N}_\infty \left(\mathcal{F} \circ \mathbf{x}, \gamma \right) \right) \right\}.$$

- Rather than a **single discretization step**, it is common to use **multiple, nested discretizations** of finer sizes – called *chaining*.

Chaining Free

Recall our upper bound:

$$R_n^{\log}(\mathcal{F}) \leq \sup_{\mathbf{x}} \inf_{\gamma>0} \left\{4n\gamma + c\log\left(\mathcal{N}_\infty\left(\mathcal{F}\circ\mathbf{x},\gamma\right)\right)\right\}.$$

- Rather than a **single discretization step**, it is common to use **multiple, nested discretizations** of finer sizes – called *chaining*.
- Our current approach does not permit such a technique, yet improves on previous results which do.

Chaining Free

Recall our upper bound:

$$R_n^{\log}(\mathcal{F}) \leq \sup_{\mathbf{x}} \inf_{\gamma > 0} \left\{ 4n\gamma + c \log \left(\mathcal{N}_\infty \left(\mathcal{F} \circ \mathbf{x}, \gamma \right) \right) \right\}.$$

- Rather than a **single discretization step**, it is common to use **multiple, nested discretizations** of finer sizes – called *chaining*.
- Our current approach does not permit such a technique, yet improves on previous results which do.
- Naive attempts to change our result to allow chaining fail, and we are actively working on this area.

Summary

Summary

Motivation

- Make probabilistic forecasts without making assumptions about the data generating process – whether i.i.d. or more sophisticated dependence structure.

Summary

Motivation

- Make probabilistic forecasts without making assumptions about the data generating process – whether i.i.d. or more sophisticated dependence structure.

Problem Setup

- Bounding minimax regret for arbitrary expert classes under logarithmic loss.

Summary

Motivation

- Make probabilistic forecasts without making assumptions about the data generating process – whether i.i.d. or more sophisticated dependence structure.

Problem Setup

- Bounding minimax regret for arbitrary expert classes under logarithmic loss.

Contributions

- Improved upper bound for complex classes and provided lower bound.
- Proof technique is truncation free and only requires one step discretization.

Summary

Motivation

- Make probabilistic forecasts without making assumptions about the data generating process – whether i.i.d. or more sophisticated dependence structure.

Problem Setup

- Bounding minimax regret for arbitrary expert classes under logarithmic loss.

Contributions

- Improved upper bound for complex classes and provided lower bound.
- Proof technique is truncation free and only requires one step discretization.

Next Steps

- Match upper and lower bounds.
- Obtain bounds that interpolate between stochastic and fully adversarial.

Open Problem

Infinite Dimensional Linear Prediction

- $\mathcal{X} = B_2$, the unit ball in a Hilbert space,
- $\mathcal{F} = \{f(x) = (\langle w, x \rangle + 1)/2 : w \in B_2\}$,
- Log-loss can be written as
  $g_t(w) = -y_t \log(1 + \langle w, x_t \rangle) - (1 - y_t) \log(1 - \langle w, x_t \rangle)$.

Open Problem

Infinite Dimensional Linear Prediction

- $\mathcal{X} = B_2$, the unit ball in a Hilbert space,
- $\mathcal{F} = \{f(x) = (\langle w, x\rangle + 1)/2 : w \in B_2\}$,
- Log-loss can be written as
  $g_t(w) = -y_t \log(1 + \langle w, x_t\rangle) - (1 - y_t) \log(1 - \langle w, x_t\rangle)$.

Constructive Algorithm (Rakhlin and Sridharan, 2015)

- Follow-the-Regularized-Leader with a *self-concordant barrier function* gives
  $R_n^{\log}(\mathcal{F}) \leq \tilde{\mathcal{O}}\left(\sqrt{n}\right)$.

Open Problem

Infinite Dimensional Linear Prediction

- $\mathcal{X} = B_2$, the unit ball in a Hilbert space,
- $\mathcal{F} = \{f(x) = (\langle w, x \rangle + 1)/2 : w \in B_2\}$,
- Log-loss can be written as
  $g_t(w) = -y_t \log(1 + \langle w, x_t \rangle) - (1 - y_t) \log(1 - \langle w, x_t \rangle)$.

Constructive Algorithm (Rakhlin and Sridharan, 2015)

- Follow-the-Regularized-Leader with a *self-concordant barrier function* gives $R_n^{\log}(\mathcal{F}) \leq \tilde{\mathcal{O}}\left(\sqrt{n}\right)$.
- This is tighter than any known upper bounds, including ours, and matches the lower bound.

Open Problem

Infinite Dimensional Linear Prediction

- $\mathcal{X} = B_2$, the unit ball in a Hilbert space,
- $\mathcal{F} = \{f(x) = (\langle w, x \rangle + 1)/2 : w \in B_2\}$,
- Log-loss can be written as
  $g_t(w) = -y_t \log(1 + \langle w, x_t \rangle) - (1 - y_t) \log(1 - \langle w, x_t \rangle)$.

Constructive Algorithm (Rakhlin and Sridharan, 2015)

- Follow-the-Regularized-Leader with a *self-concordant barrier function* gives $R_n^{\log}(\mathcal{F}) \leq \tilde{\mathcal{O}}\left(\sqrt{n}\right)$.
- This is tighter than any known upper bounds, including ours, and matches the lower bound.
- It is not well-defined how to apply a concrete algorithm technique like this to arbitrary expert classes.