Adaptive Online Learning

Dylan Foster
Cornell University

- Joint work with Alexander Rakhlin and Karthik Sridharan

Cornell University

Penn
University of Pennsylvania

Online Learning Protocol

For $t = 1$ to n

End

Online Learning Protocol

For $t = 1$ to n

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

End

Online Learning Protocol

For $t = 1$ to n

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

Learner picks randomized prediction $q_t \in \Delta(\mathcal{Y})$

End

Online Learning Protocol

For $t = 1$ to n

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

Learner picks randomized prediction $q_t \in \Delta(\mathcal{Y})$

Receive outcome $y_t \in \mathcal{Y}$

End

Online Learning Protocol

For $t = 1$ to n

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

Learner picks randomized prediction $q_t \in \Delta(\mathcal{Y})$

Receive outcome $y_t \in \mathcal{Y}$

Learner draws prediction $\hat{y}_t \sim q_t$ and suffers loss $\ell(\hat{y}_t, y_t)$

End

Online Learning Protocol

For $t = 1$ to n

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

Learner picks randomized prediction $q_t \in \Delta(\mathcal{Y})$

Receive outcome $y_t \in \mathcal{Y}$

Learner draws prediction $\hat{y}_t \sim q_t$ and suffers loss $\ell(\hat{y}_t, y_t)$

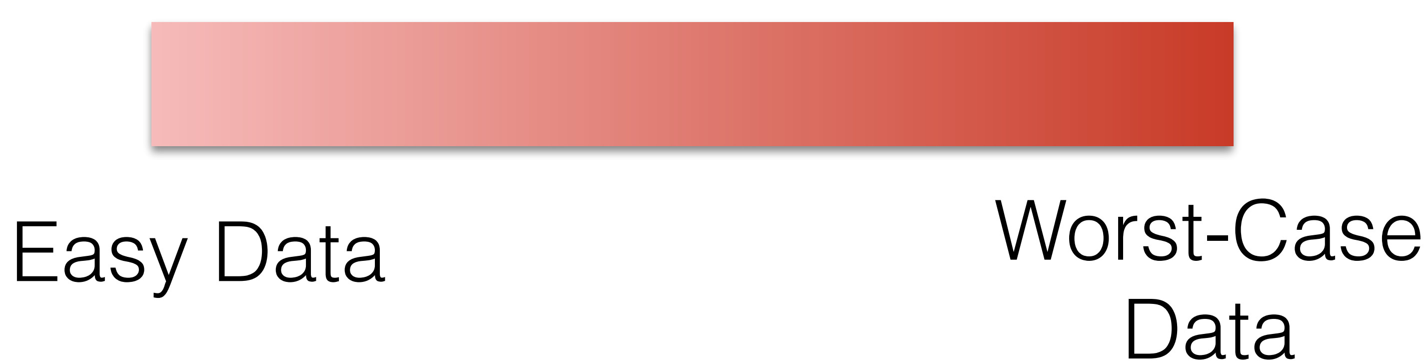
End

Goal: Minimize regret w.r.t. any $f \in \mathcal{F}$

$$\mathbf{Reg}_n(x_{1:n}, y_{1:n}; f) = \sum_{t=1}^{n} \ell(\hat{y}_t, y_t) - \sum_{t=1}^{n} \ell(f(x_t), y_t)$$

Adaptive Learning

Adaptive Learning

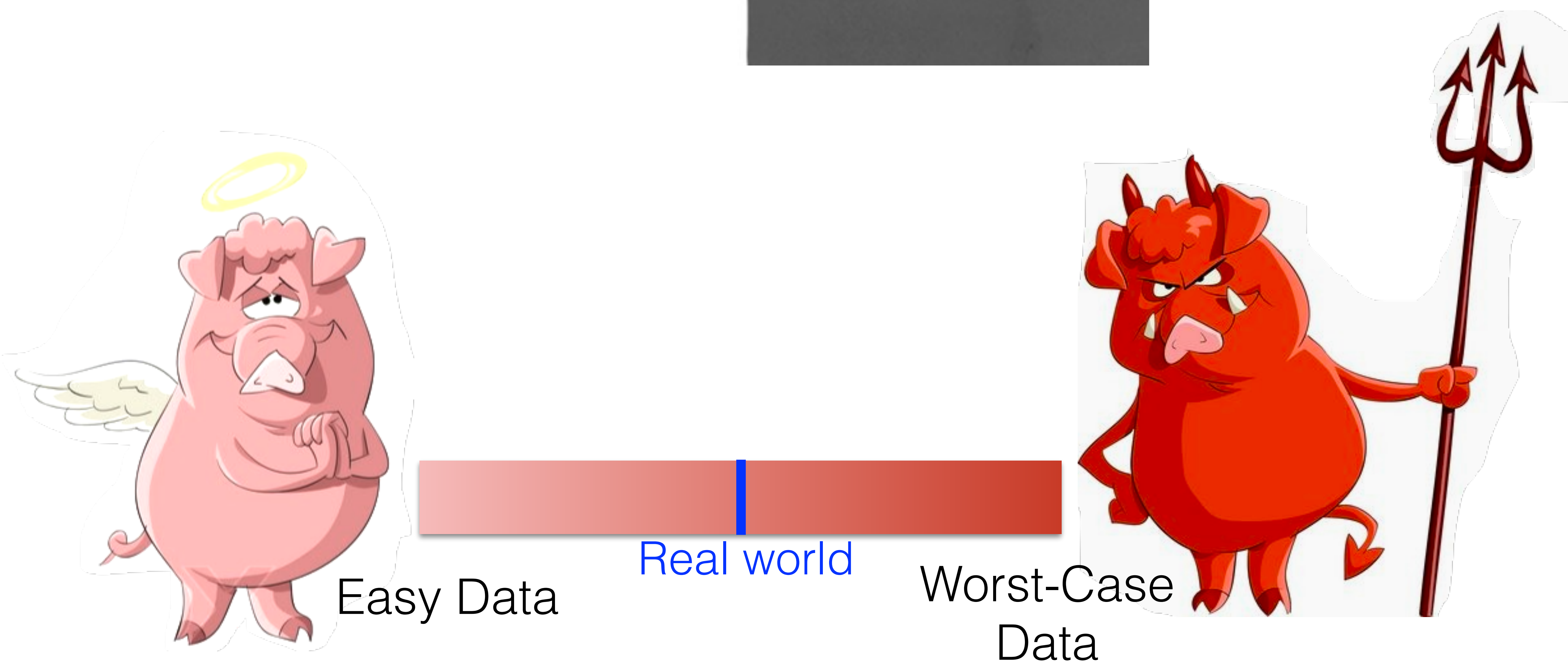


Adaptive Learning

Easy Data

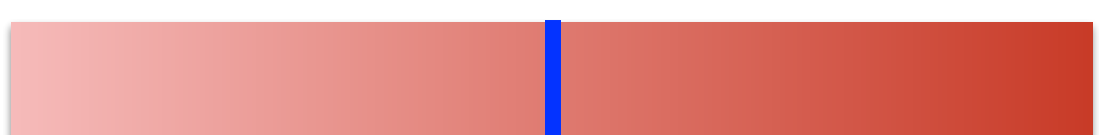
Worst-Case Data

Adaptive Learning



ADAPTIVE LEARNING

Be cautiously optimistic



Easy Data

Worst-Case Data

Uniform Regret Bounds

Uniform bound on regret:

$$\forall f \in \mathcal{F}, \quad \mathbf{Reg}_n(x_{1:n}, y_{1:n}; f) \leq B(n)$$

Examples:

Uniform Regret Bounds

Uniform bound on regret:

$$\forall f \in \mathcal{F}, \quad \mathbf{Reg}_n(x_{1:n}, y_{1:n}; f) \leq B(n)$$

Examples:

- Gradient descent $B(n) = \sqrt{n}$ [Zinkevich'03]

Uniform Regret Bounds

Uniform bound on regret:

$$\forall f \in \mathcal{F}, \quad \mathbf{Reg}_n(x_{1:n}, y_{1:n}; f) \le B(n)$$

Examples:

- Gradient descent $B(n) = \sqrt{n}$ [Zinkevich'03]
- Exponential weights $B(n) = \sqrt{n \log |\mathcal{F}|}$ [Littlestone-Warmuth'94], [Vovk'98]

Adaptive Regret Bounds

Adaptive regret bound:

$$\forall f \in \mathcal{F}, \quad \mathbf{Reg}_n(x_{1:n}, y_{1:n}; f) \leq B(f; x_{1:n}, y_{1:n})$$

Examples:

Adaptive Regret Bounds

Adaptive regret bound:

$$\forall f \in \mathcal{F}, \quad \mathbf{Reg}_n(x_{1:n}, y_{1:n}; f) \leq B(f; x_{1:n}, y_{1:n})$$

Examples:

- Gradient descent $B(f; \nabla_{1:n}) = C\sqrt{\sum_{t=1}^{n} \|\nabla_t\|_2^2}$ e.g. [McMahan-Streeter'10]

Adaptive Regret Bounds

Adaptive regret bound:

$$\forall f \in \mathcal{F}, \quad \mathbf{Reg}_n(x_{1:n}, y_{1:n}; f) \leq B(f; x_{1:n}, y_{1:n})$$

Examples:

- Gradient descent $B(f; \nabla_{1:n}) = C\sqrt{\sum_{t=1}^{n} \|\nabla_t\|_2^2}$ e.g. [McMahan-Streeter'10]
- Exponential weights
  $B(f; x_{1:n}, y_{1:n}) = C\sqrt{\log|\mathcal{F}| \sum_{t=1}^{n} \ell(f(x_t), y_t)} + K \log|\mathcal{F}|$ e.g. [Cesa-Bianchi-Lugosi'06]

Adaptive Regret Bounds

Adaptive regret bound:

$$\forall f \in \mathcal{F}, \quad \mathbf{Reg}_n(x_{1:n}, y_{1:n}; f) \le B(f; x_{1:n}, y_{1:n})$$

Examples:

- Gradient descent $B(f; \nabla_{1:n}) = C\sqrt{\sum_{t=1}^{n} \|\nabla_t\|_2^2}$ e.g. [McMahan-Streeter‘10]
- Exponential weights $B(f; x_{1:n}, y_{1:n}) = C\sqrt{\log|\mathcal{F}| \ \sum_{t=1}^{n} \ell(f(x_t), y_t)} + K \log|\mathcal{F}|$ e.g. [Cesa-Bianchi-Lugosi‘06]
- ...many more!
  [Cesa-Bianchi-Mansour-Stoltz‘07], [Even-Dar-Kearns-Mansour-Wortman‘08]
  [Chaudhuri-Freund-Hsu‘09], [Duchi-Hazan-Singer‘11]
  [Rakhlin-Sridharan‘13], [McMahan-Orabona‘14],
  [Luo-Schapire‘15], [Koolen-van Erven’15]
  ⋮

Adaptive Regret Bounds

Adaptive regret bound:

$$\forall f \in \mathcal{F}, \quad \mathbf{Reg}_n(x_{1:n}, y_{1:n}; f) \leq B(f; x_{1:n}, y_{1:n})$$

What we want from B

Adaptive Regret Bounds

Adaptive regret bound:

$$\forall f \in \mathcal{F}, \;\; \mathbf{Reg}_n(x_{1:n}, y_{1:n}; f) \leq B(f; x_{1:n}, y_{1:n})$$

What we want from B

- More likely models enjoy smaller regret

Adaptive Regret Bounds

Adaptive regret bound:

$$\forall f \in \mathcal{F}, \quad \mathbf{Reg}_n(x_{1:n}, y_{1:n}; f) \leq B(f; x_{1:n}, y_{1:n})$$

What we want from B

- More likely models enjoy smaller regret
- Instances easier to deal with enjoy better bound

Adaptive Regret Bounds

Adaptive regret bound:

$$\forall f \in \mathcal{F}, \quad \mathbf{Reg}_n(x_{1:n}, y_{1:n}; f) \leq B(f; x_{1:n}, y_{1:n})$$

What we want from B

- More likely models enjoy smaller regret
- Instances easier to deal with enjoy better bound
- Retain worst case guarantee, that is

$$\sup_{f; x_{1:n}, y_{1:n}} B(f; x_{1:n}, y_{1:n}) = \tilde{O}(\text{Optimal uniform rate}_n)$$

Adaptive Regret Bounds

Adaptive regret bound:

$$\forall f \in \mathcal{F}, \quad \mathbf{Reg}_n(x_{1:n}, y_{1:n}; f) \le B(f; x_{1:n}, y_{1:n})$$

What we want from B

- More likely models enjoy smaller regret
- Instances easier to deal with enjoy better bound
- Retain worst case guarantee, that is

$$\sup_{f; x_{1:n}, y_{1:n}} B(f; x_{1:n}, y_{1:n}) = \tilde{O}(\text{Optimal uniform rate}_n)$$

What adaptive rates, B's, are achievable?

Achievable Adaptive Bounds

Adaptive rate B is said to be achievable if

$$\left[\mathbf{Reg}_n(x_{1:n}, y_{1:n}; f) - B(f; x_{1:n}, y_{1:n})\right] \le 0$$

Achievable Adaptive Bounds

Adaptive rate B is said to be achievable if

$$\sup_{f \in \mathcal{F}} \left[\mathbf{Reg}_n(x_{1:n}, y_{1:n}; f) - B(f; x_{1:n}, y_{1:n}) \right] \leq 0$$

Achievable Adaptive Bounds

Adaptive rate B is said to be achievable if

$$\max_{\text{instances}} \mathbb{E} \sup_{f \in \mathcal{F}} \left[\mathbf{Reg}_n(x_{1:n}, y_{1:n}; f) - B(f; x_{1:n}, y_{1:n}) \right] \leq 0$$

Achievable Adaptive Bounds

Adaptive rate B is said to be achievable if

$$\mathcal{A}_n := \min_{\substack{\text{Randomized}\\ \text{Algorithms}}} \max_{\text{instances}} \mathbb{E} \sup_{f \in \mathcal{F}} \left[\mathbf{Reg}_n(x_{1:n}, y_{1:n}; f) - B(f; x_{1:n}, y_{1:n}) \right] \leq 0$$

Achievable Adaptive Bounds

Adaptive rate B is said to be achievable if

$$\mathcal{A}_n := \min_{\substack{\text{Randomized}\\\text{Algorithms}}} \max_{\text{instances}} \mathbb{E} \sup_{f \in \mathcal{F}} \left[\mathbf{Reg}_n(x_{1:n}, y_{1:n}; f) - B(f; x_{1:n}, y_{1:n}) \right] \leq 0$$

- To show that a rate $\approx B_n$ is achievable we need to prove $\mathcal{A}_n$ is bounded by a constant or $o(B_n)$ bound.

Achievable Adaptive Bounds

Adaptive rate B is said to be achievable if

$$\mathcal{A}_n := \min_{\substack{\text{Randomized} \\ \text{Algorithms}}} \max_{\text{instances}} \mathbb{E} \sup_{f \in \mathcal{F}} \left[\mathbf{Reg}_n(x_{1:n}, y_{1:n}; f) - B(f; x_{1:n}, y_{1:n}) \right] \leq 0$$

- To show that a rate $\approx B_n$ is achievable we need to prove $\mathcal{A}_n$ is bounded by a constant or $o(B_n)$ bound.
- We analyze $\mathcal{A}_n$ by going to dual game and using idea of symmetrization.

Sequential Rademacher Complexity

Sequential Rademacher complexity: [Rakhlin, Sridharan, Tewari'10]

$$\mathcal{R}_n(\mathcal{F}) := \sup_{\mathbf{x}} \mathbb{E}_{\epsilon}\left[\frac{2}{n}\sup_{f\in\mathcal{F}}\left|\sum_{t=1}^{n}\epsilon_t f(\mathbf{x}_t(\epsilon_{1:t-1}))\right|\right]$$

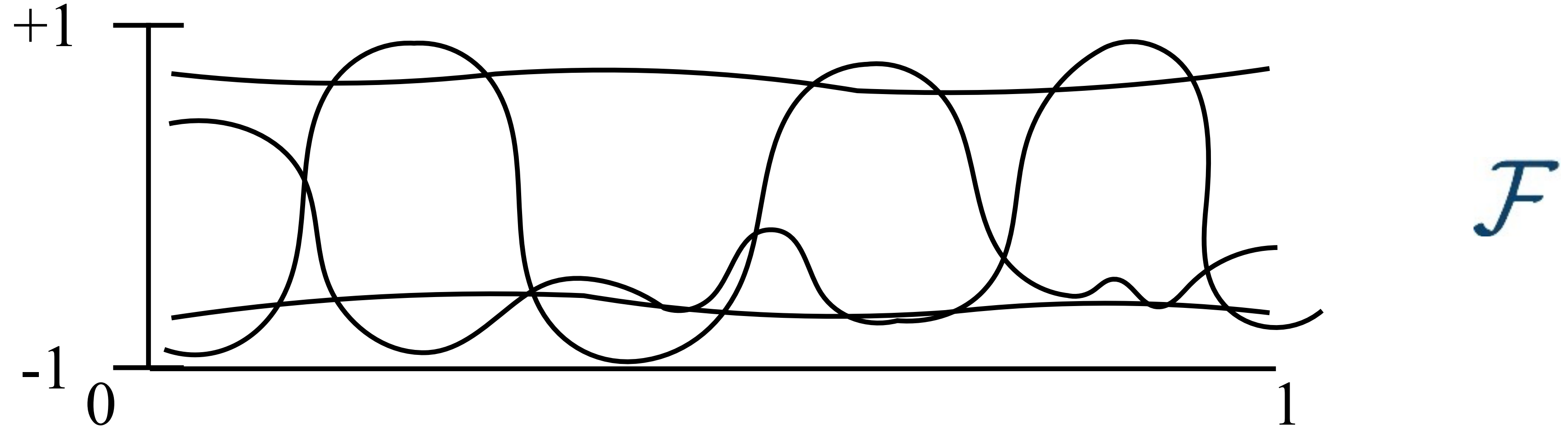
where $\mathbf{x} = (\mathbf{x}_1,\ldots,\mathbf{x}_n)$ is $\mathcal{X}$-valued tree. (each $\mathbf{x}_t : \{\pm 1\}^{t-1} \mapsto \mathcal{X}$)

Sequential Rademacher Complexity

Sequential Rademacher complexity: [Rakhlin, Sridharan, Tewari'10]

$$\mathcal{R}_n(\mathcal{F}) := \sup_{\mathbf{x}} \mathbb{E}_{\epsilon}\left[\frac{2}{n}\sup_{f\in\mathcal{F}}\left|\sum_{t=1}^{n} \epsilon_t f(\mathbf{x}_t(\epsilon_{1:t-1}))\right|\right]$$

where $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ is $\mathcal{X}$-valued tree. (each $\mathbf{x}_t : \{\pm 1\}^{t-1} \mapsto \mathcal{X}$)

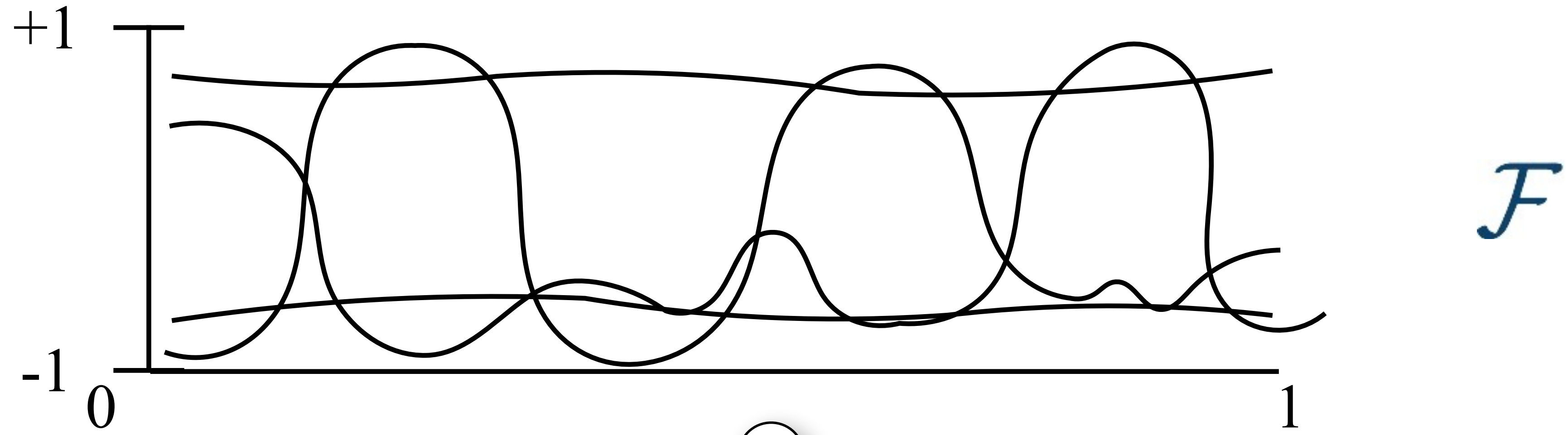


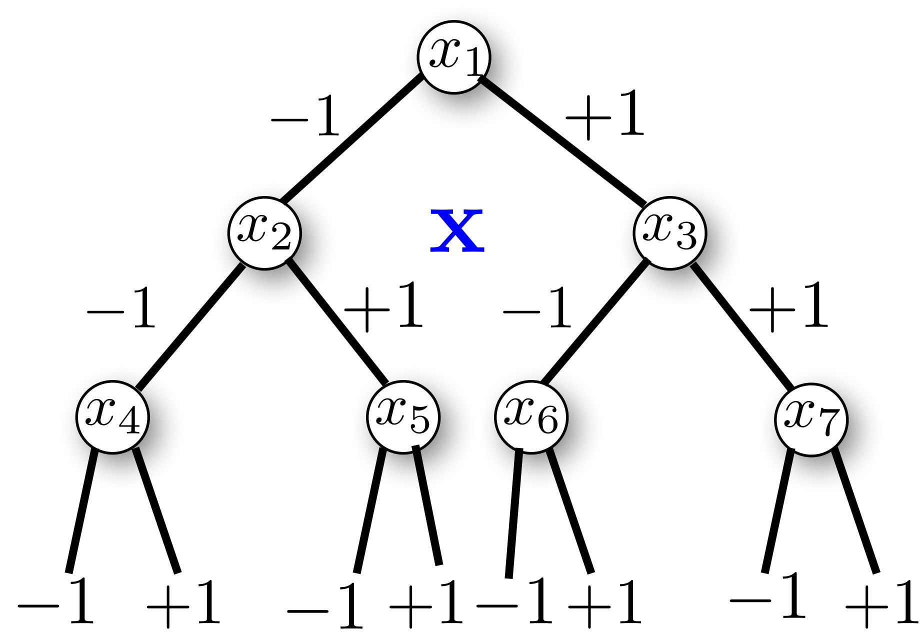
SEQUENTIAL RADEMACHER COMPLEXITY

Sequential Rademacher complexity: [Rakhlin, Sridharan, Tewari'10]

$$\mathcal{R}_n(\mathcal{F}) := \sup_{\mathbf{x}} \mathbb{E}_{\epsilon} \left[\frac{2}{n} \sup_{f \in \mathcal{F}} \left| \sum_{t=1}^{n} \epsilon_t f(\mathbf{x}_t(\epsilon_{1:t-1})) \right| \right]$$

where $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ is $\mathcal{X}$-valued tree. (each $\mathbf{x}_t : \{\pm 1\}^{t-1} \mapsto \mathcal{X}$)



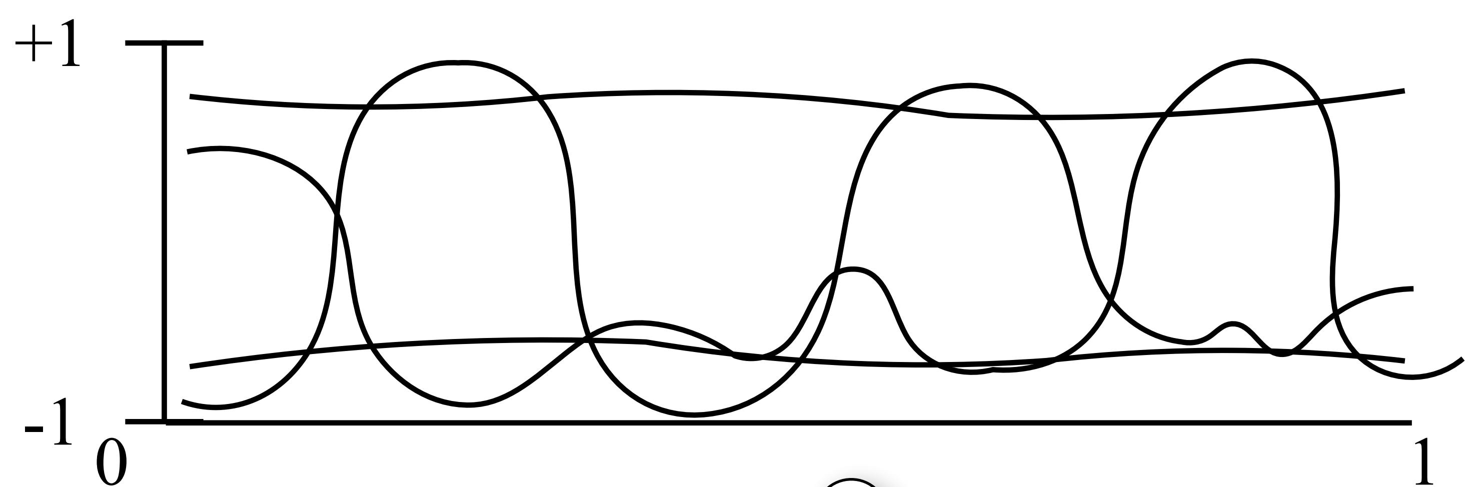


Sequential Rademacher Complexity

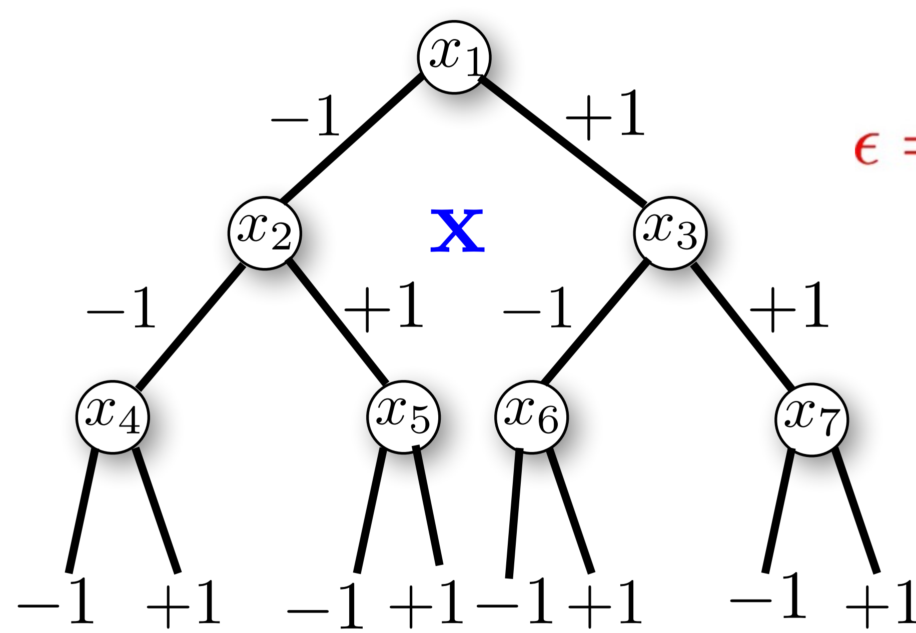
Sequential Rademacher complexity: [Rakhlin, Sridharan, Tewari'10]

$$\mathcal{R}_n(\mathcal{F}) := \sup_{\mathbf{x}} \mathbb{E}_{\epsilon}\left[\frac{2}{n}\sup_{f\in\mathcal{F}}\left|\sum_{t=1}^{n} \epsilon_t f(\mathbf{x}_t(\epsilon_{1:t-1}))\right|\right]$$

where $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ is $\mathcal{X}$-valued tree. (each $\mathbf{x}_t : \{\pm 1\}^{t-1} \mapsto \mathcal{X}$)



$\mathcal{F}$



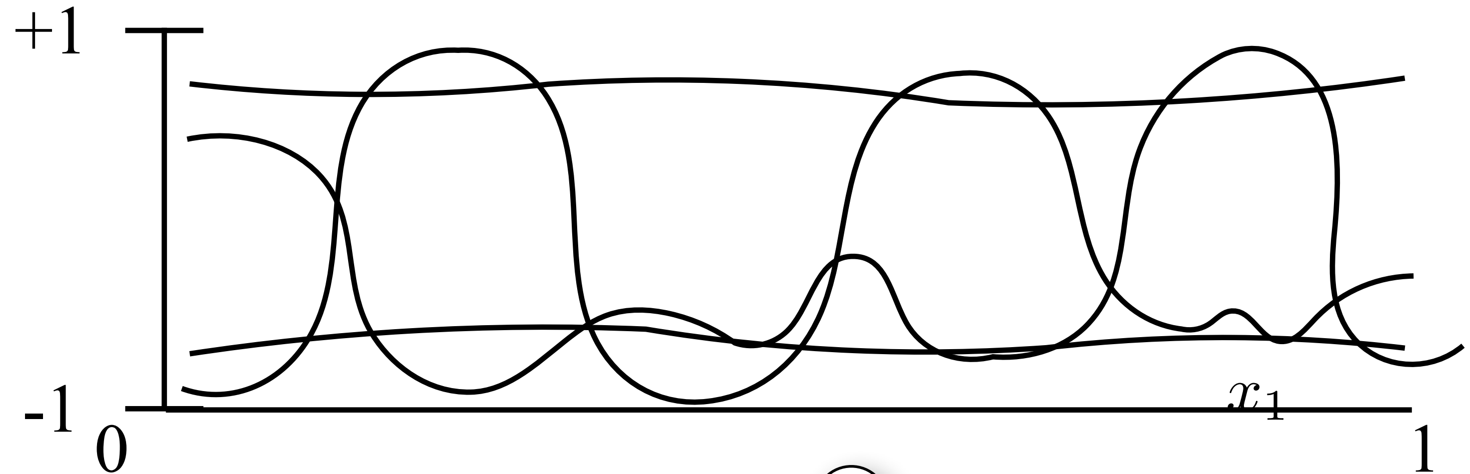
$\epsilon = (+1, -1, -1, \ldots, 1)$

Sequential Rademacher Complexity

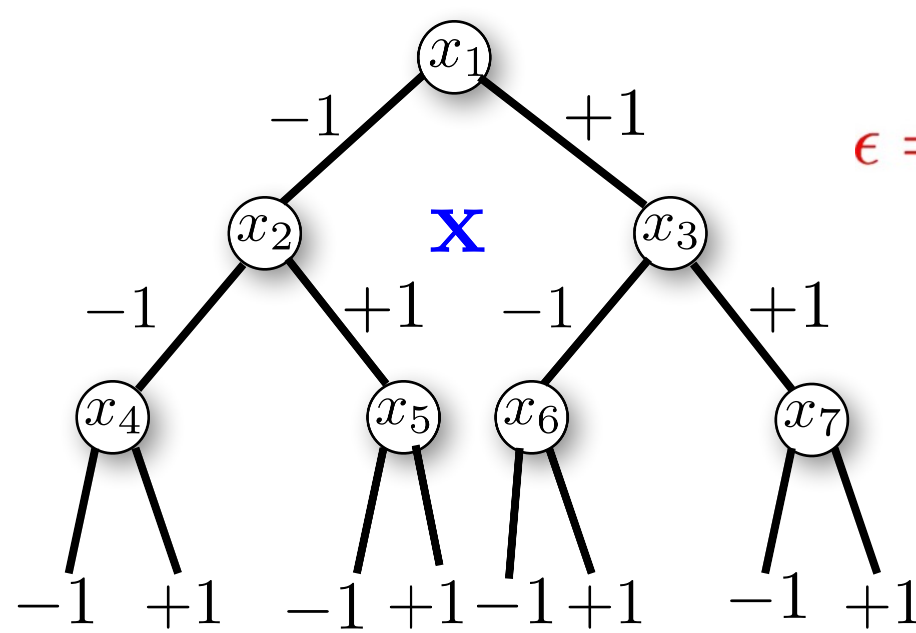
Sequential Rademacher complexity: [Rakhlin, Sridharan, Tewari'10]

$$\mathcal{R}_n(\mathcal{F}) := \sup_{\mathbf{x}} \mathbb{E}_{\epsilon}\left[\frac{2}{n}\sup_{f\in\mathcal{F}}\left|\sum_{t=1}^{n}\epsilon_t f(\mathbf{x}_t(\epsilon_{1:t-1}))\right|\right]$$

where $\mathbf{x} = (\mathbf{x}_1,\ldots,\mathbf{x}_n)$ is $\mathcal{X}$-valued tree. (each $\mathbf{x}_t : \{\pm1\}^{t-1} \mapsto \mathcal{X}$)



$\mathcal{F}$



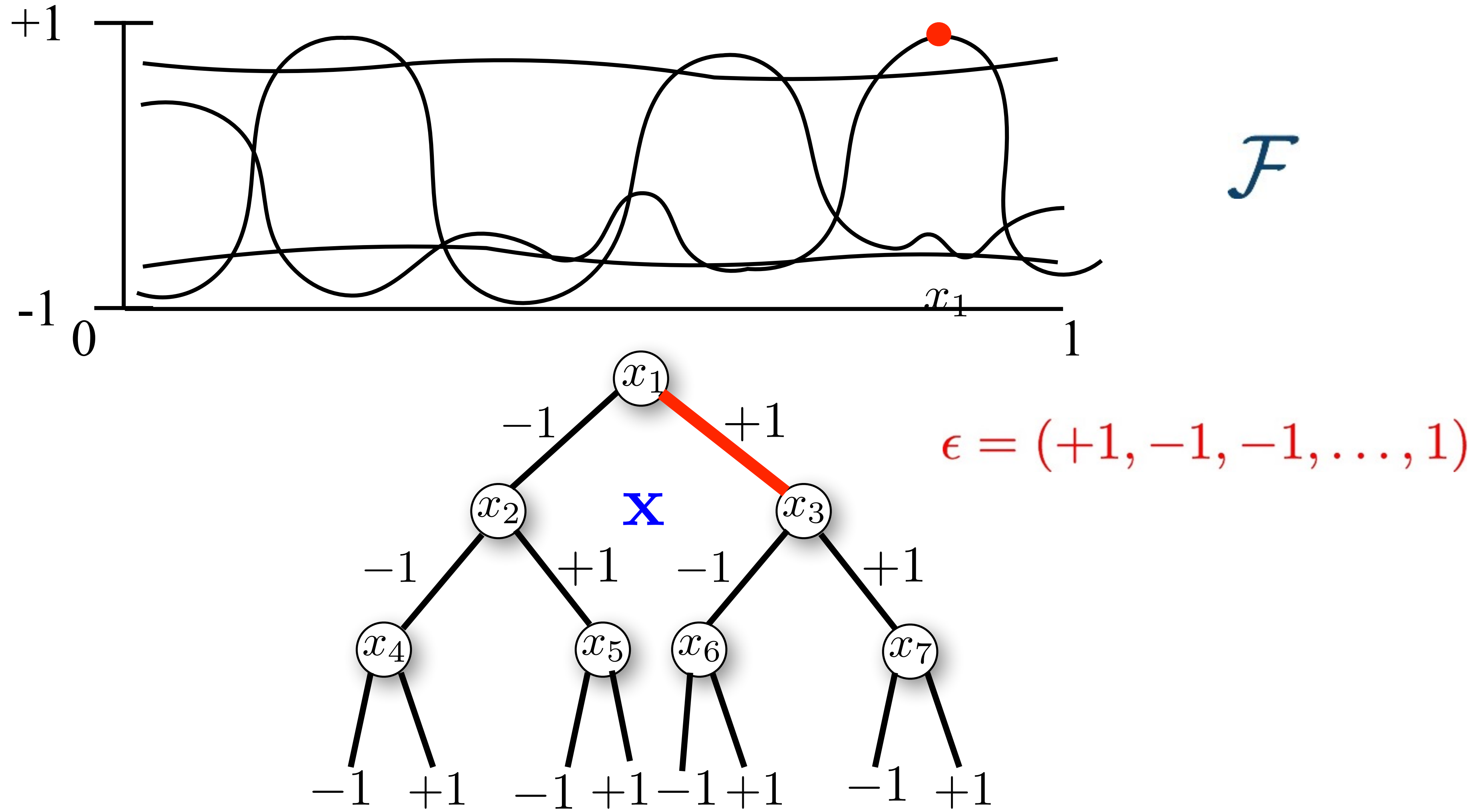
$\epsilon = (+1, -1, -1, \ldots, 1)$

SEQUENTIAL RADEMACHER COMPLEXITY

Sequential Rademacher complexity: [Rakhlin, Sridharan, Tewari'10]

$$\mathcal{R}_n(\mathcal{F}) := \sup_{\mathbf{x}} \mathbb{E}_{\epsilon}\left[\frac{2}{n}\sup_{f\in\mathcal{F}}\left|\sum_{t=1}^{n}\epsilon_t f(\mathbf{x}_t(\epsilon_{1:t-1}))\right|\right]$$

where $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ is $\mathcal{X}$-valued tree. (each $\mathbf{x}_t : \{\pm 1\}^{t-1} \mapsto \mathcal{X}$)

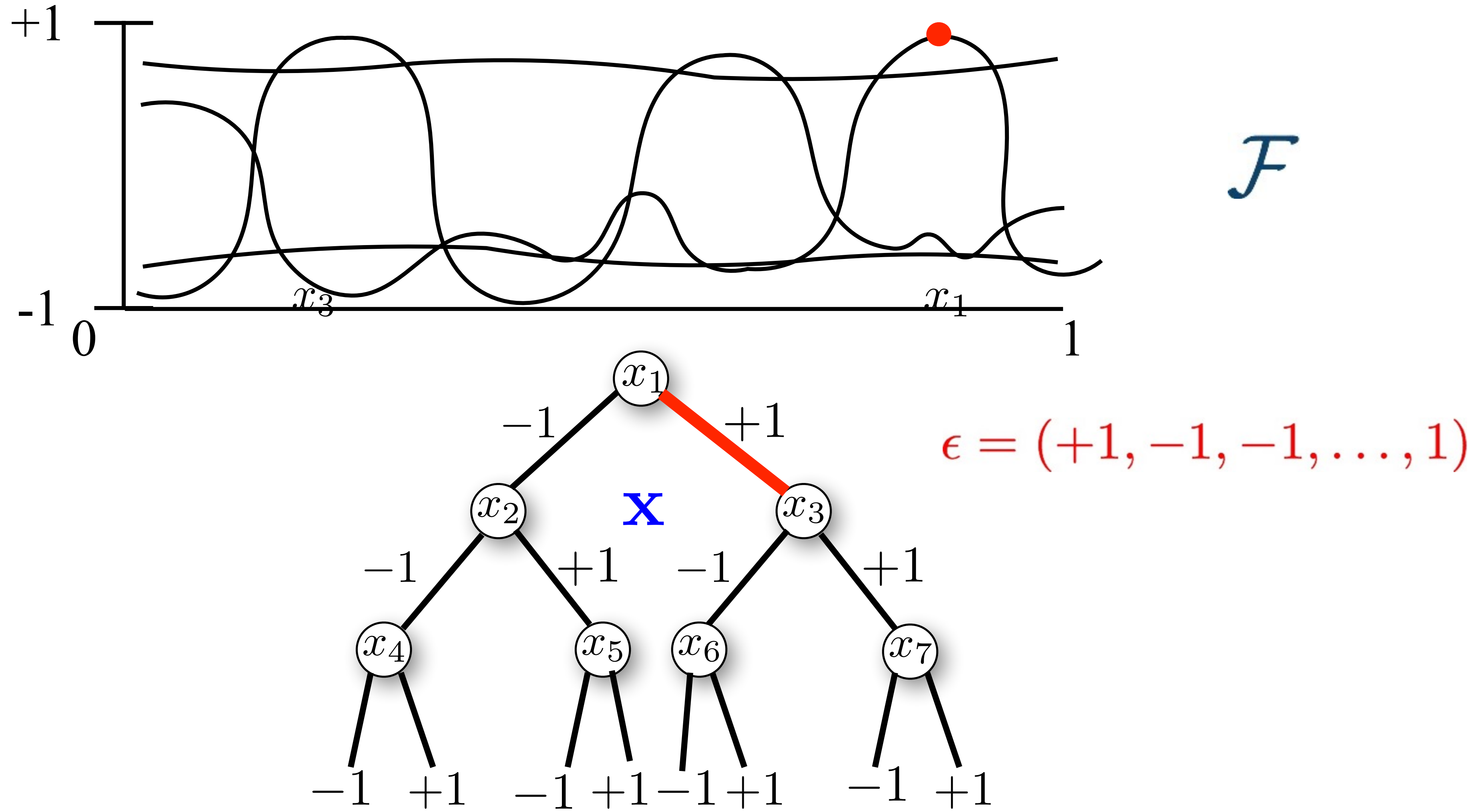


Sequential Rademacher Complexity

Sequential Rademacher complexity: [Rakhlin, Sridharan, Tewari'10]

$$\mathcal{R}_n(\mathcal{F}) := \sup_{\mathbf{x}} \mathbb{E}_{\epsilon}\left[\frac{2}{n}\sup_{f\in\mathcal{F}}\left|\sum_{t=1}^{n}\epsilon_t f(\mathbf{x}_t(\epsilon_{1:t-1}))\right|\right]$$

where $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ is $\mathcal{X}$-valued tree. (each $\mathbf{x}_t : \{\pm 1\}^{t-1} \mapsto \mathcal{X}$)

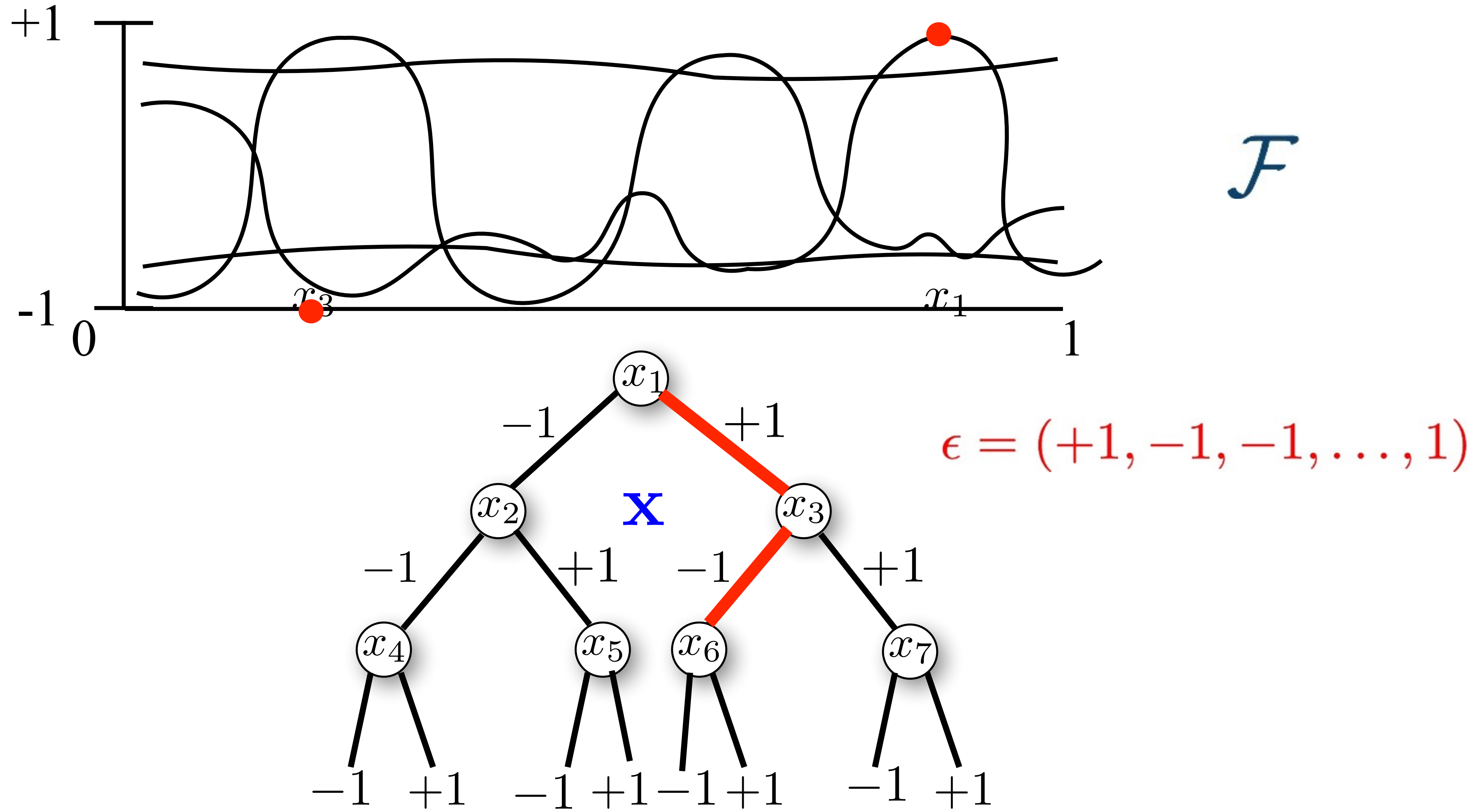


Sequential Rademacher Complexity

Sequential Rademacher complexity: [Rakhlin, Sridharan, Tewari'10]

$$\mathcal{R}_n(\mathcal{F}) := \sup_{\mathbf{x}} \mathbb{E}_{\epsilon}\left[\frac{2}{n}\sup_{f\in\mathcal{F}}\left|\sum_{t=1}^{n}\epsilon_t f(\mathbf{x}_t(\epsilon_{1:t-1}))\right|\right]$$

where $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ is $\mathcal{X}$-valued tree. (each $\mathbf{x}_t : \{\pm 1\}^{t-1} \mapsto \mathcal{X}$)

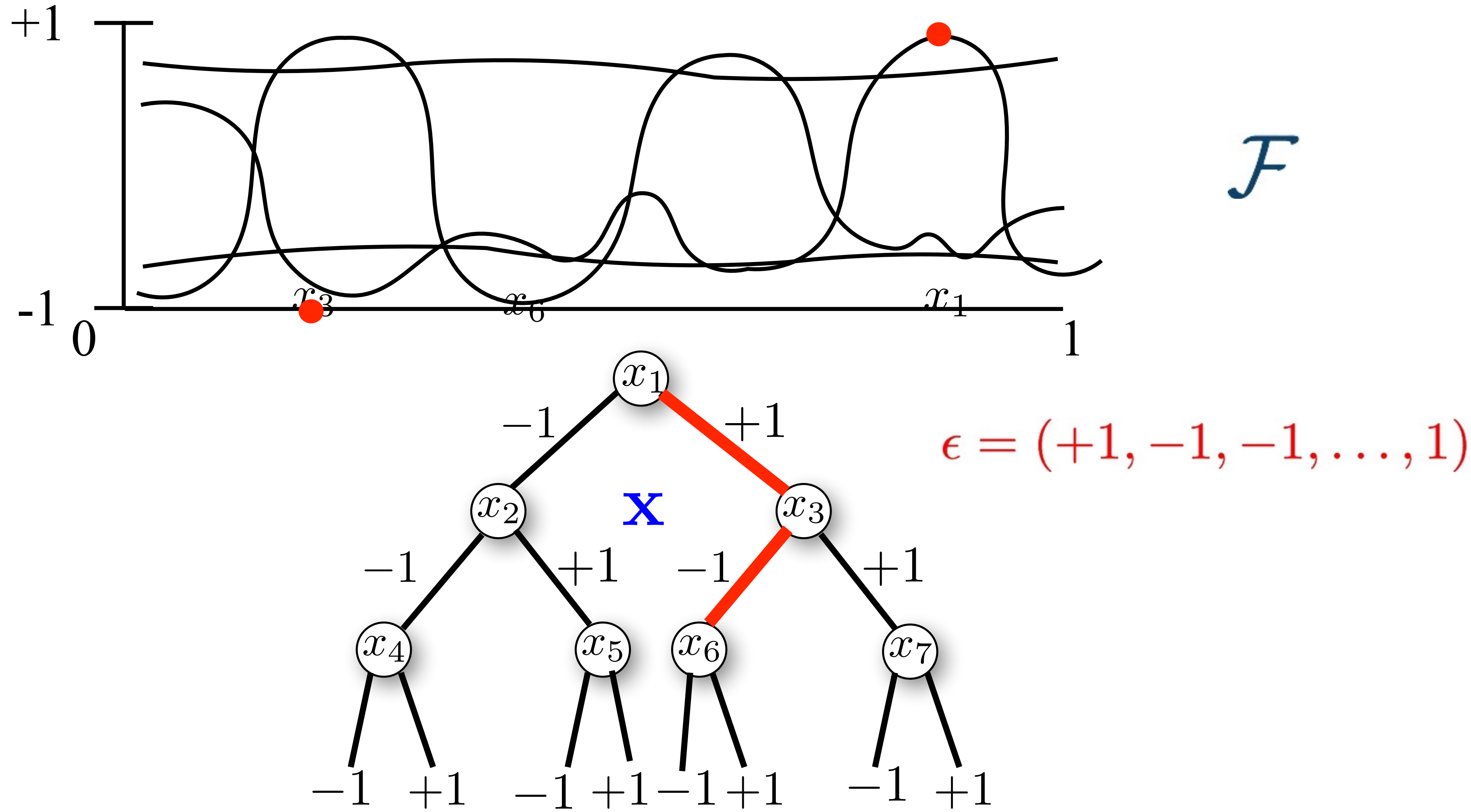


SEQUENTIAL RADEMACHER COMPLEXITY

Sequential Rademacher complexity: [Rakhlin, Sridharan, Tewari'10]

$$\mathcal{R}_n(\mathcal{F}) := \sup_{\mathbf{x}} \mathbb{E}_{\epsilon} \left[\frac{2}{n} \sup_{f \in \mathcal{F}} \left| \sum_{t=1}^{n} \epsilon_t f(\mathbf{x}_t(\epsilon_{1:t-1})) \right| \right]$$

where $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ is $\mathcal{X}$-valued tree. (each $\mathbf{x}_t : \{\pm 1\}^{t-1} \mapsto \mathcal{X}$)

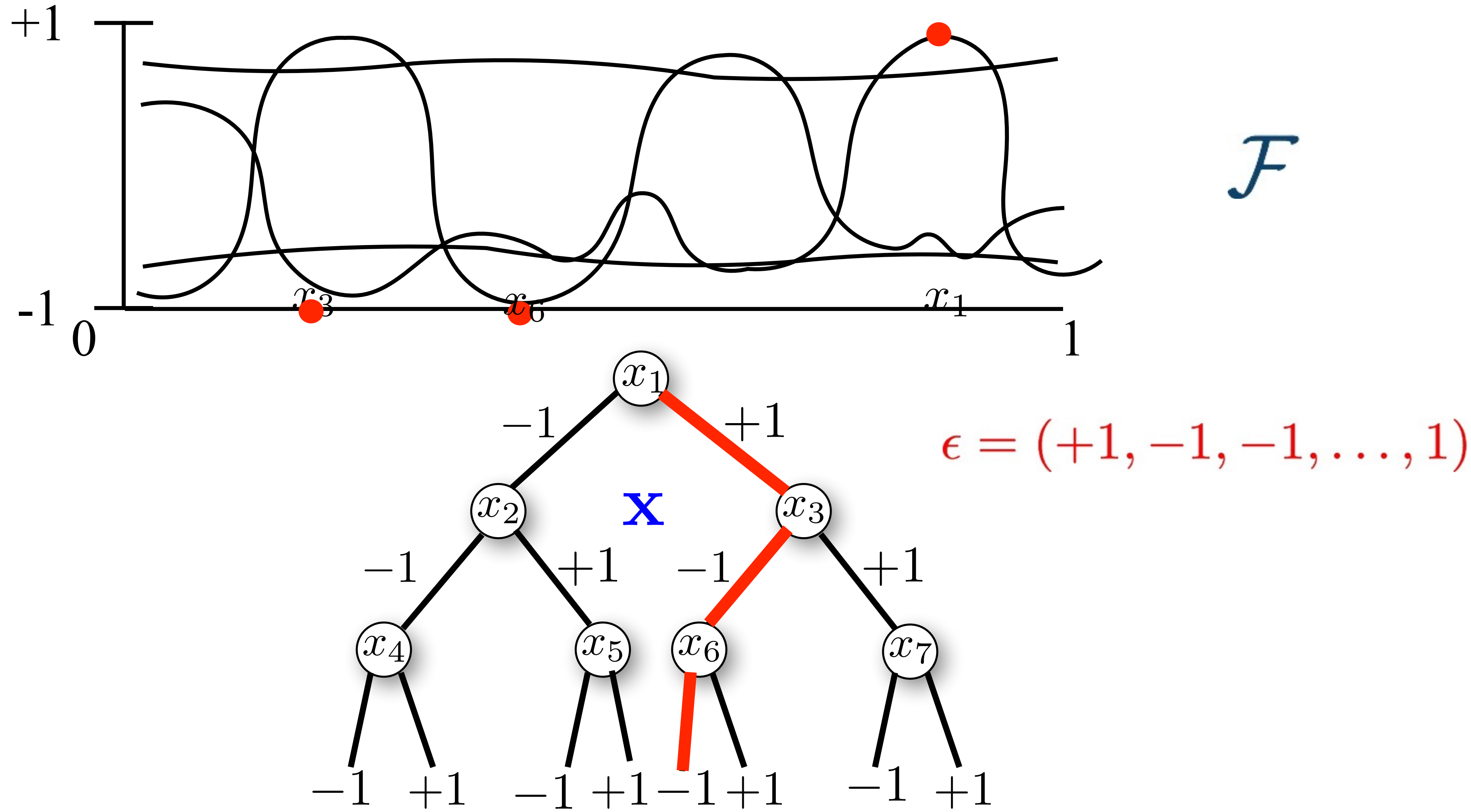


SEQUENTIAL RADEMACHER COMPLEXITY

Sequential Rademacher complexity: [Rakhlin, Sridharan, Tewari'10]

$$\mathcal{R}_n(\mathcal{F}) := \sup_{\mathbf{x}} \mathbb{E}_{\epsilon}\left[\frac{2}{n}\sup_{f\in\mathcal{F}}\left|\sum_{t=1}^{n} \epsilon_t f(\mathbf{x}_t(\epsilon_{1:t-1}))\right|\right]$$

where $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ is $\mathcal{X}$-valued tree. (each $\mathbf{x}_t : \{\pm 1\}^{t-1} \mapsto \mathcal{X}$)

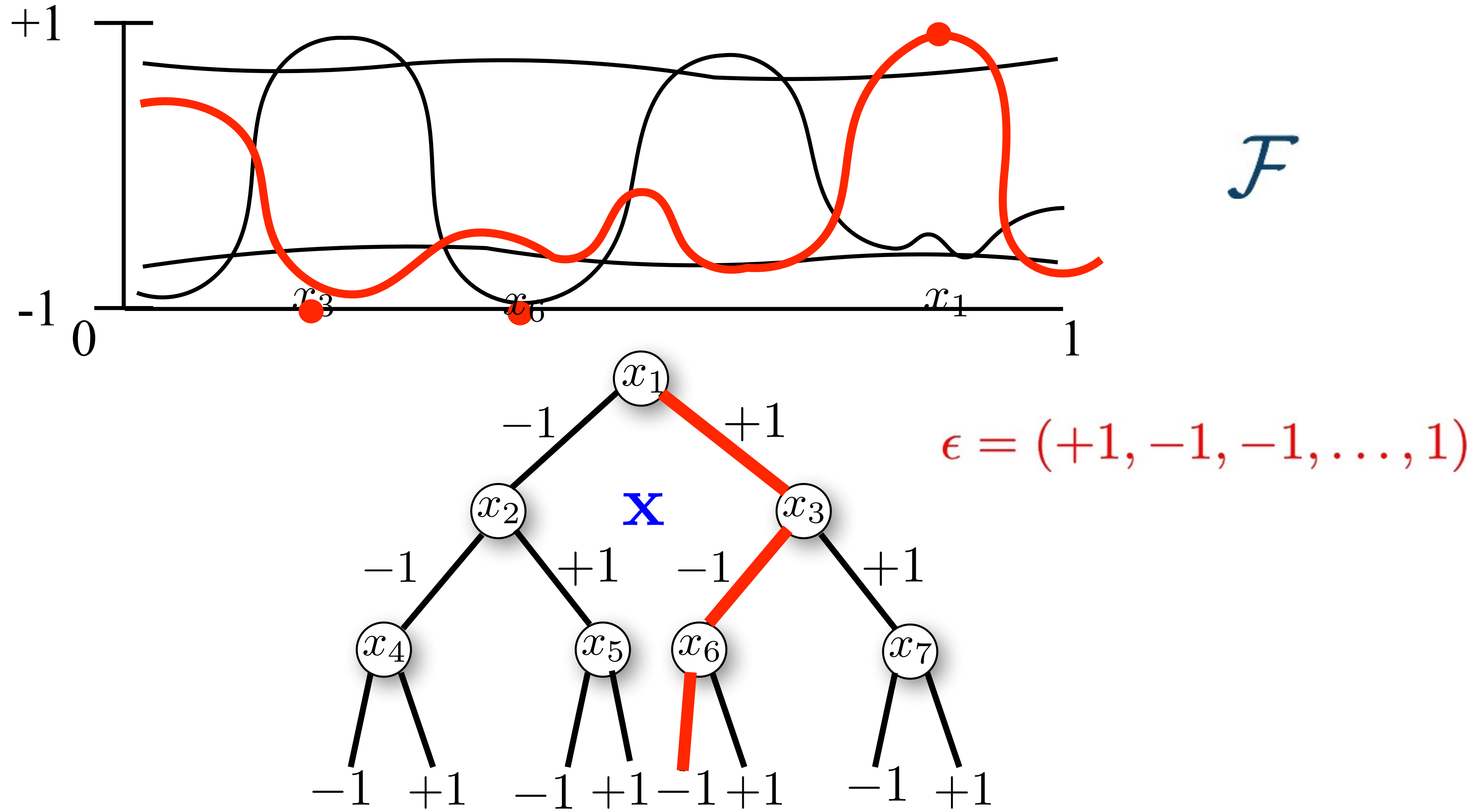


Sequential Rademacher Complexity

Sequential Rademacher complexity: [Rakhlin, Sridharan, Tewari'10]

$$\mathcal{R}_n(\mathcal{F}) := \sup_{\mathbf{x}} \mathbb{E}_{\epsilon}\left[\frac{2}{n}\sup_{f\in\mathcal{F}}\left|\sum_{t=1}^{n}\epsilon_t f(\mathbf{x}_t(\epsilon_{1:t-1}))\right|\right]$$

where $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ is $\mathcal{X}$-valued tree. (each $\mathbf{x}_t : \{\pm 1\}^{t-1} \mapsto \mathcal{X}$)



Sequential Rademacher Complexity

Sequential Rademacher complexity: [Rakhlin, Sridharan, Tewari'10]

$$\mathcal{R}_n(\mathcal{F}) := \sup_{\mathbf{x}} \mathbb{E}_{\epsilon} \left[\frac{2}{n} \sup_{f \in \mathcal{F}} \left| \sum_{t=1}^{n} \epsilon_t f(\mathbf{x}_t(\epsilon_{1:t-1})) \right| \right]$$

where $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ is $\mathcal{X}$-valued tree. (each $\mathbf{x}_t : \{\pm 1\}^{t-1} \mapsto \mathcal{X}$)

Sequential Rademacher Complexity

Sequential Rademacher complexity: [Rakhlin, Sridharan, Tewari'10]

$$\mathcal{R}_n(\mathcal{F}) := \sup_{\mathbf{x}} \mathbb{E}_{\epsilon}\left[\frac{2}{n}\sup_{f\in\mathcal{F}}\left|\sum_{t=1}^{n}\epsilon_t f(\mathbf{x}_t(\epsilon_{1:t-1}))\right|\right]$$

where $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ is $\mathcal{X}$-valued tree. (each $\mathbf{x}_t : \{\pm 1\}^{t-1} \mapsto \mathcal{X}$)

tree — random signs — max correlation on drawn path

Sequential Rademacher Complexity

Sequential Rademacher complexity: [Rakhlin, Sridharan, Tewari'10]

$$\mathcal{R}_n(\mathcal{F}) := \sup_{\mathbf{x}} \mathbb{E}_{\epsilon}\left[\frac{2}{n}\sup_{f\in\mathcal{F}}\left|\sum_{t=1}^{n}\epsilon_t f(\mathbf{x}_t(\epsilon_{1:t-1}))\right|\right]$$

where $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ is $\mathcal{X}$-valued tree. (each $\mathbf{x}_t : \{\pm 1\}^{t-1} \mapsto \mathcal{X}$)

Theorem [Rakhlin, S., Tewari'10]

For any class of predictors $\mathcal{F} \subset \mathbb{R}^{\mathcal{X}}$ and appropriate loss :

$\mathcal{F}$ is online learnable (ie. $\mathcal{V}_n(\mathcal{F}) \to 0$) if and only if $\mathcal{R}_n(\mathcal{F}) \to 0$

Sequential Rademacher Complexity

Sequential Rademacher complexity: [Rakhlin, Sridharan, Tewari'10]

$$\mathcal{R}_n(\mathcal{F}) := \sup_{\mathbf{x}} \mathbb{E}_{\epsilon}\left[\frac{2}{n}\sup_{f\in\mathcal{F}}\left|\sum_{t=1}^{n}\epsilon_t f(\mathbf{x}_t(\epsilon_{1:t-1}))\right|\right]$$

where $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ is $\mathcal{X}$-valued tree. (each $\mathbf{x}_t : \{\pm 1\}^{t-1} \mapsto \mathcal{X}$)

Theorem [Rakhlin, S., Tewari'10]

For any class of predictors $\mathcal{F} \subset \mathbb{R}^{\mathcal{X}}$ *and appropriate loss :*

$\mathcal{F}$ *is online learnable (ie.* $\mathcal{V}_n(\mathcal{F}) \to 0$*) if and only if* $\mathcal{R}_n(\mathcal{F}) \to 0$

For absolute loss , $\frac{1}{2}\mathcal{R}_n(\mathcal{F}) \le \mathcal{V}_n(\mathcal{F}) \le \mathcal{R}_n(\mathcal{F})$

Sequential Rademacher Complexity

Sequential Rademacher complexity: [Rakhlin, Sridharan, Tewari'10]

$$\mathcal{R}_n(\mathcal{F}) := \sup_{\mathbf{x}} \mathbb{E}_{\epsilon}\left[\frac{2}{n}\sup_{f\in\mathcal{F}}\left|\sum_{t=1}^{n}\epsilon_t f(\mathbf{x}_t(\epsilon_{1:t-1}))\right|\right]$$

where $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ is $\mathcal{X}$-valued tree. (each $\mathbf{x}_t : \{\pm 1\}^{t-1} \mapsto \mathcal{X}$)

Theorem [Rakhlin, S., Tewari'10]

For any class of predictors $\mathcal{F} \subset \mathbb{R}^{\mathcal{X}}$ and appropriate loss :

$\mathcal{F}$ is online learnable (ie. $\mathcal{V}_n(\mathcal{F}) \to 0$) if and only if $\mathcal{R}_n(\mathcal{F}) \to 0$

For absolute loss , $\frac{1}{2}\mathcal{R}_n(\mathcal{F}) \le \mathcal{V}_n(\mathcal{F}) \le \mathcal{R}_n(\mathcal{F})$

VC or PAC theory for online learning !

VC or PAC style theory for adaptive online learning?

Sufficient Condition for Achievability

Lemma

Convex and L-Lipschitz supervised learning loss (or 0-1 loss):

$$\sup_{\mathbf{x},\mathbf{y}} \mathbb{E}_{\epsilon}\left[\sup_{f\in\mathcal{F}}\left\{\underbrace{2L\sum_{t=1}^{n}\epsilon_t f(\mathbf{x}_t(\epsilon))}_{\text{Rademacher average}} - \right\}\right].$$

Sufficient Condition for Achievability

Lemma

Convex and L-Lipschitz supervised learning loss (or 0-1 loss):

$$\mathcal{A}_n \leq \sup_{\mathbf{x},\mathbf{y}} \mathbb{E}_\epsilon \left[\sup_{f \in \mathcal{F}} \left\{ \underbrace{2L \sum_{t=1}^n \epsilon_t f(\mathbf{x}_t(\epsilon))}_{\textit{Rademacher average}} - \underbrace{B_n(f; \mathbf{x}_{1:n}(\epsilon), \mathbf{y}_{1:n}(\epsilon))}_{\textit{Offset}} \right\} \right].$$

Sufficient Condition for Achievability

Lemma

Convex and L-Lipschitz supervised learning loss (or 0-1 loss):

$$\mathcal{A}_n \leq \sup_{\mathbf{x},\mathbf{y}} \mathbb{E}_\epsilon \left[\sup_{f\in\mathcal{F}} \left\{ \underbrace{2L \sum_{t=1}^n \epsilon_t f(\mathbf{x}_t(\epsilon))}_{\text{Rademacher average}} - \underbrace{B_n(f; \mathbf{x}_{1:n}(\epsilon), \mathbf{y}_{1:n}(\epsilon))}_{\text{Offset}} \right\} \right].$$

- Similar bounds hold for more general settings.

Sufficient Condition for Achievability

Lemma

Convex and L-Lipschitz supervised learning loss (or 0-1 loss):

$$\mathcal{A}_n \leq \sup_{\mathbf{x},\mathbf{y}} \mathbb{E}_\epsilon \left[\sup_{f \in \mathcal{F}} \left\{ \underbrace{2L \sum_{t=1}^n \epsilon_t f(\mathbf{x}_t(\epsilon))}_{\text{Rademacher average}} - \underbrace{B_n(f; \mathbf{x}_{1:n}(\epsilon), \mathbf{y}_{1:n}(\epsilon))}_{\text{Offset}} \right\} \right].$$

- Similar bounds hold for more general settings.
- When B_n is a uniform rate, recovers sequential Rademacher complexity bound [Rakhlin-Sridharan-Tewari‘10].

Sufficient Condition for Achievability

Lemma

Convex and L-Lipschitz supervised learning loss (or 0-1 loss):

$$\mathcal{A}_n \le \sup_{\mathbf{x},\mathbf{y}} \mathbb{E}_\epsilon \left[\sup_{f\in\mathcal{F}} \left\{ \underbrace{2L \sum_{t=1}^n \epsilon_t f(\mathbf{x}_t(\epsilon))}_{\text{Rademacher average}} - \underbrace{B_n(f;\mathbf{x}_{1:n}(\epsilon),\mathbf{y}_{1:n}(\epsilon))}_{\text{Offset}} \right\} \right].$$

- Similar bounds hold for more general settings.
- When B_n is a uniform rate, recovers sequential Rademacher complexity bound [Rakhlin-Sridharan-Tewari'10].
- Specific settings have matching lower bound.

Example

- To check the adaptive bound from gradient descent, we need to ensure

$$\sup_{\mathbf{y}} \mathbb{E}_{\epsilon}\left[\underbrace{2\left\|\sum_{t=1}^{n} \epsilon_t \mathbf{y}_t\right\|_2}_{\text{Rademacher average}} \underbrace{-C\sqrt{\sum_{t=1}^{n} \|\mathbf{y}_t\|_2^2}}_{\text{Offset}} \right] \leq 0.$$

- Jensen + Pythagoras: Sufficient to take $C = 2$.

EXAMPLE

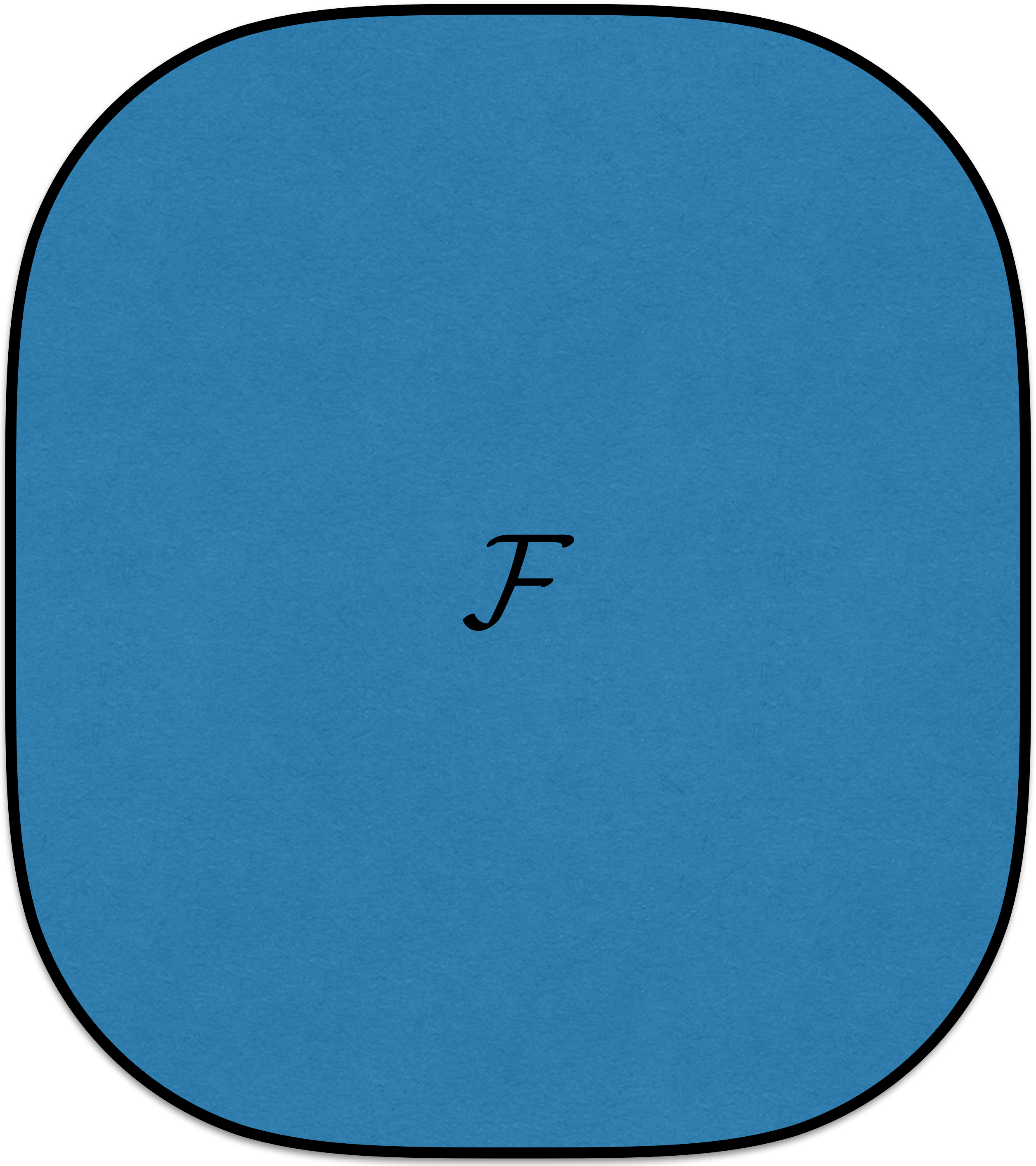
- To check the adaptive bound from gradient descent, we need to ensure

$$\sup_{\mathbf{y}} \mathbb{E}_{\epsilon}\left[\underbrace{2\left\|\sum_{t=1}^{n} \epsilon_t \mathbf{y}_t\right\|_2}_{\text{Rademacher average}} \underbrace{-C\sqrt{\sum_{t=1}^{n} \|\mathbf{y}_t\|_2^2}}_{\text{Offset}} \right] \leq 0.$$

- Jensen + Pythagoras: Sufficient to take $C = 2$.
- Takeaway: To show achievability we need to bound expected random process.

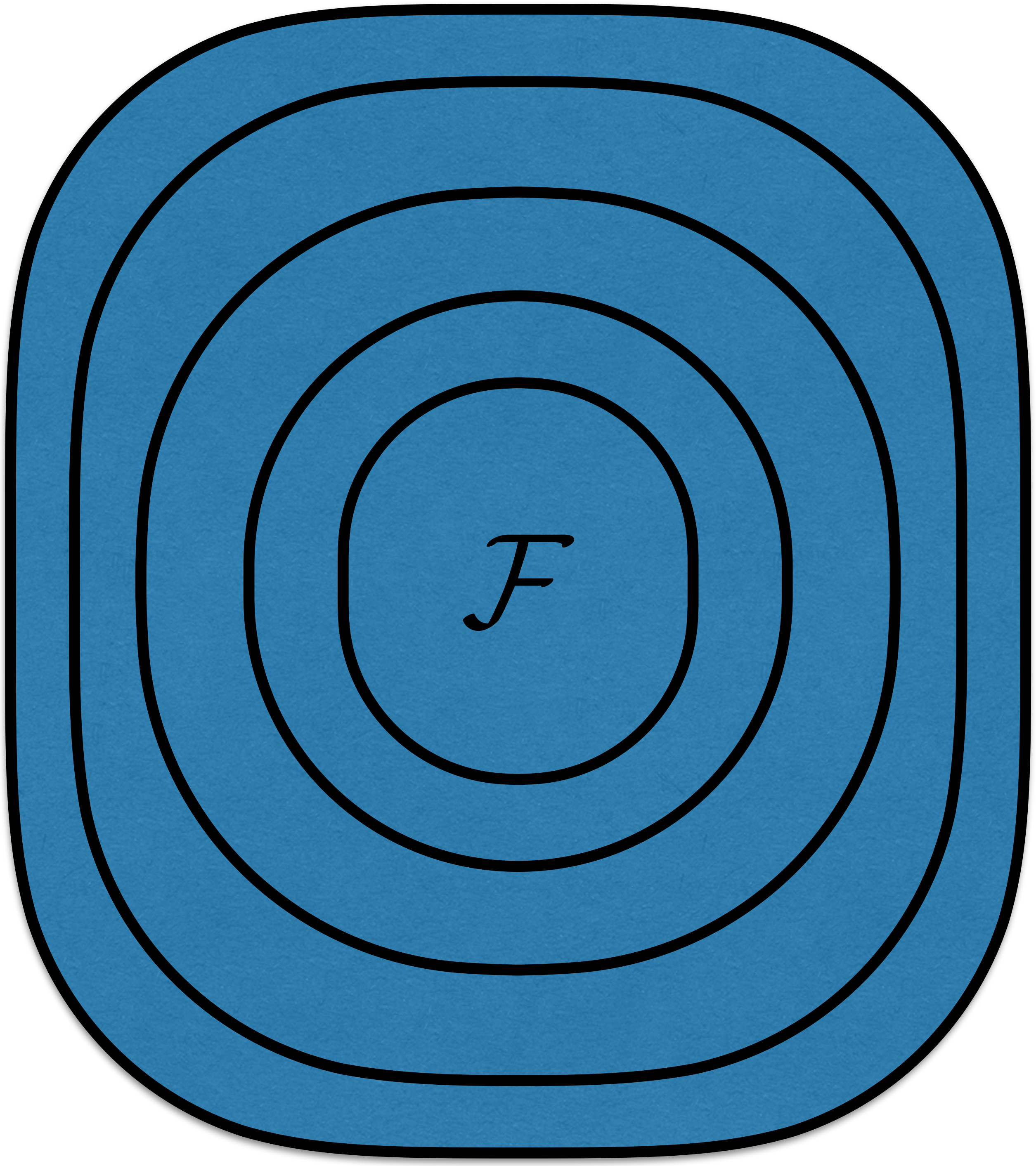
Online Model Selection

Online Model Selection



Uniform $\mathrm{Rate}_n(\mathcal{F})$ is large

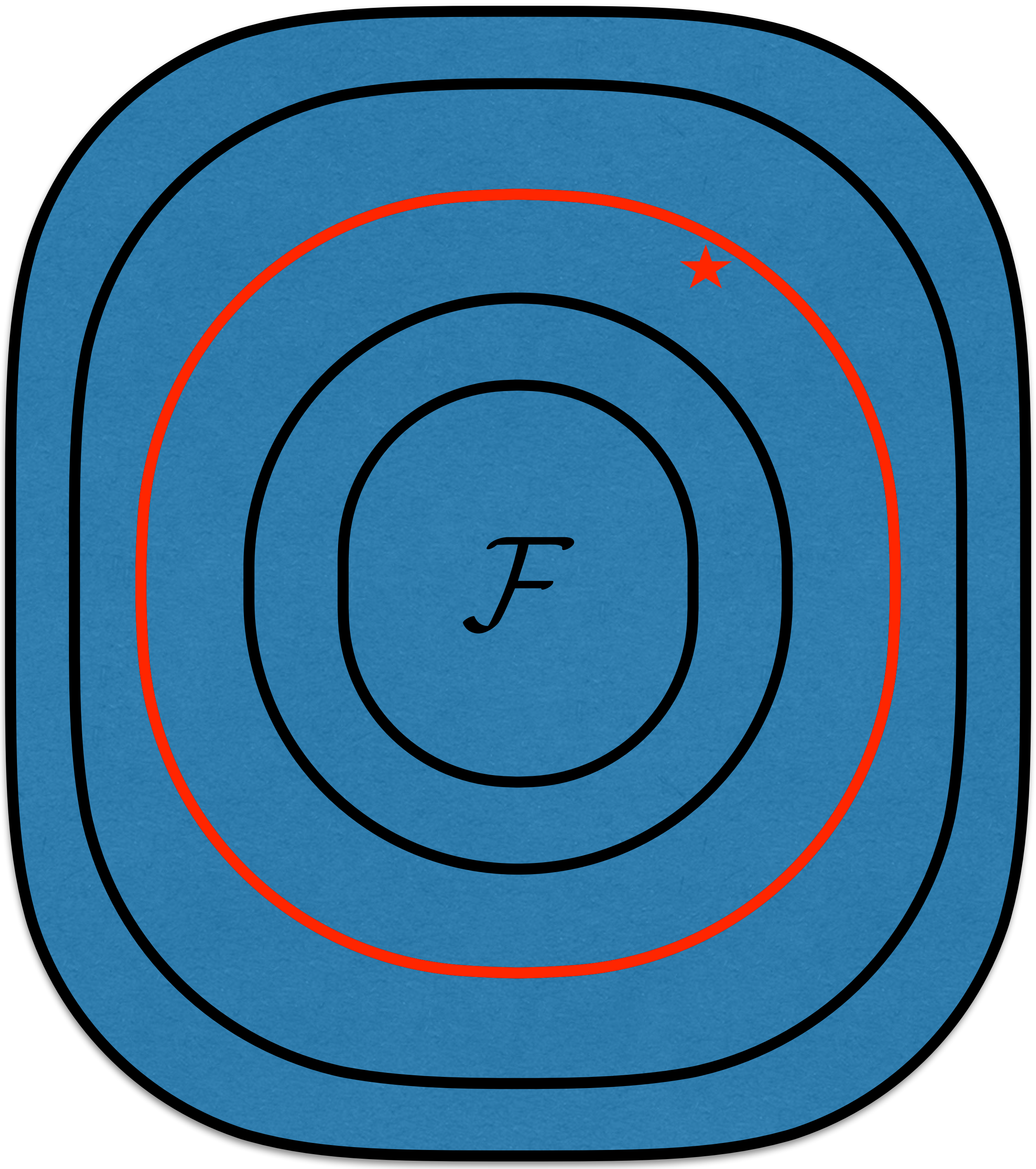
Online Model Selection



Uniform $\text{Rate}_n(\mathcal{F})$ is large

$$\mathcal{F} = \bigcup_r \mathcal{F}_r$$

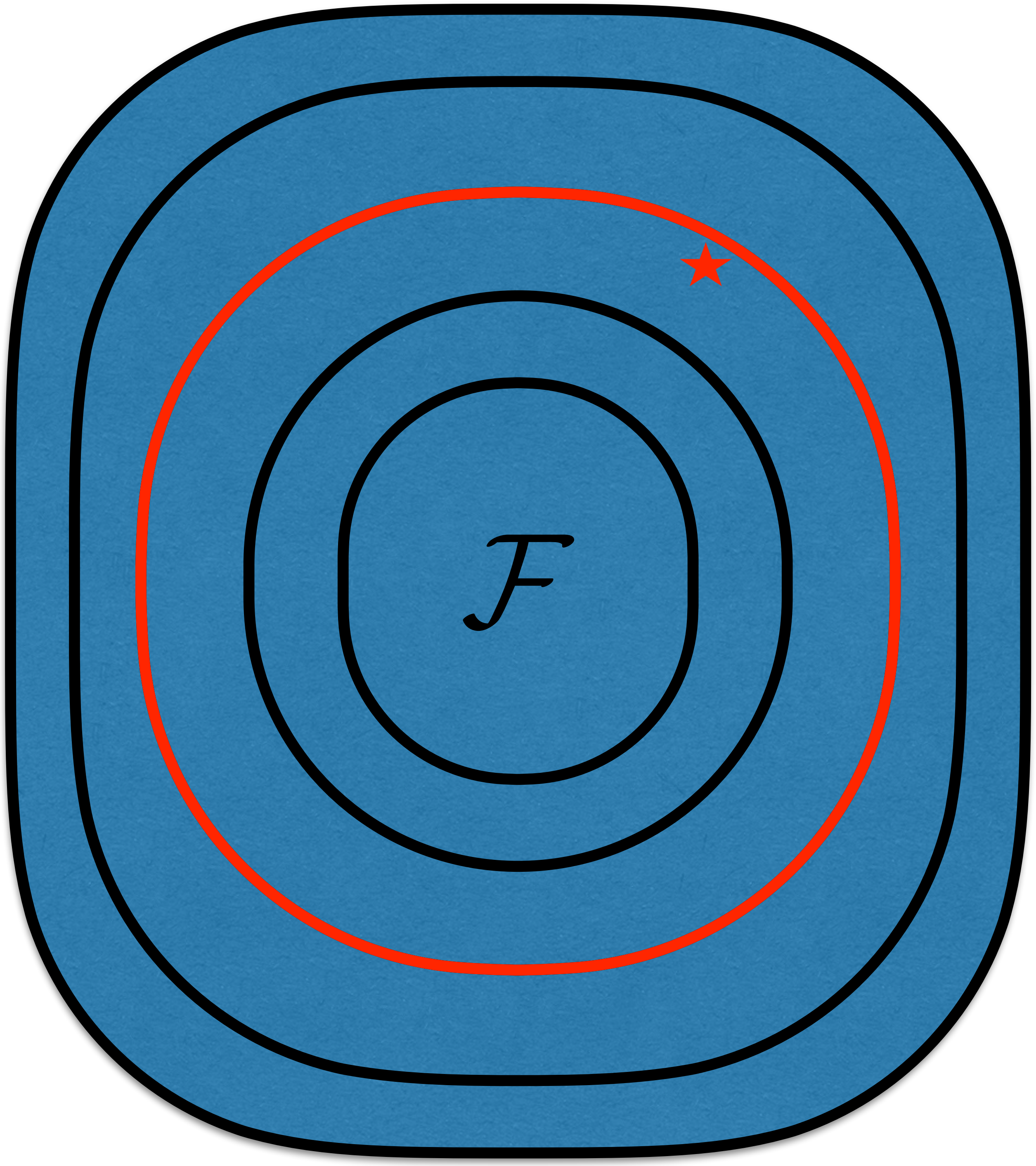
Online Model Selection



Uniform $\text{Rate}_n(\mathcal{F})$ is large

$$\mathcal{F} = \bigcup_r \mathcal{F}_r$$

Online Model Selection

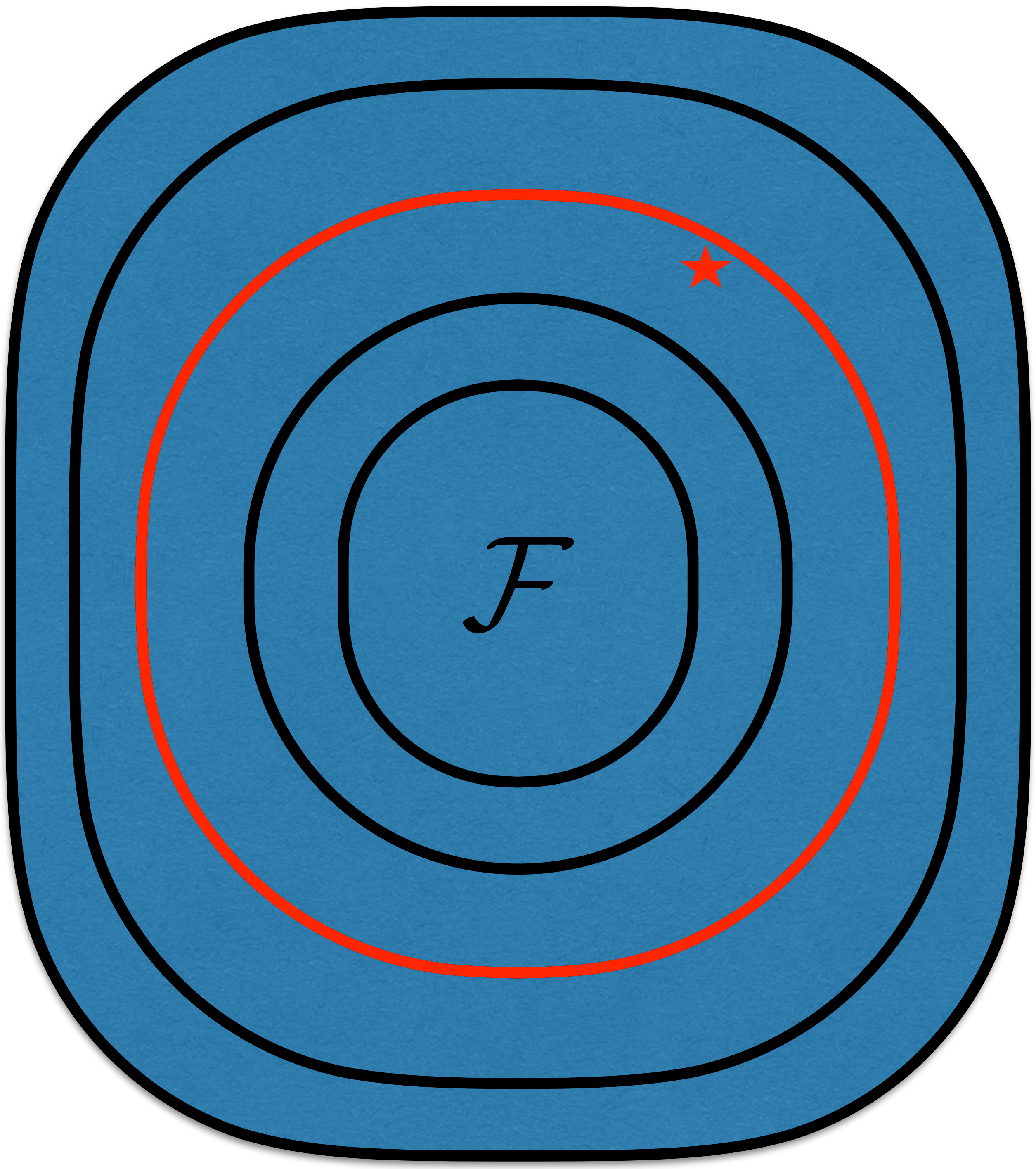


Uniform $\text{Rate}_n(\mathcal{F})$ is large

$$\mathcal{F} = \bigcup_r \mathcal{F}_r$$

$$R(f) = \inf \{r : f \in \mathcal{F}_r\}$$

Online Model Selection



Uniform $\text{Rate}_n(\mathcal{F})$ is large

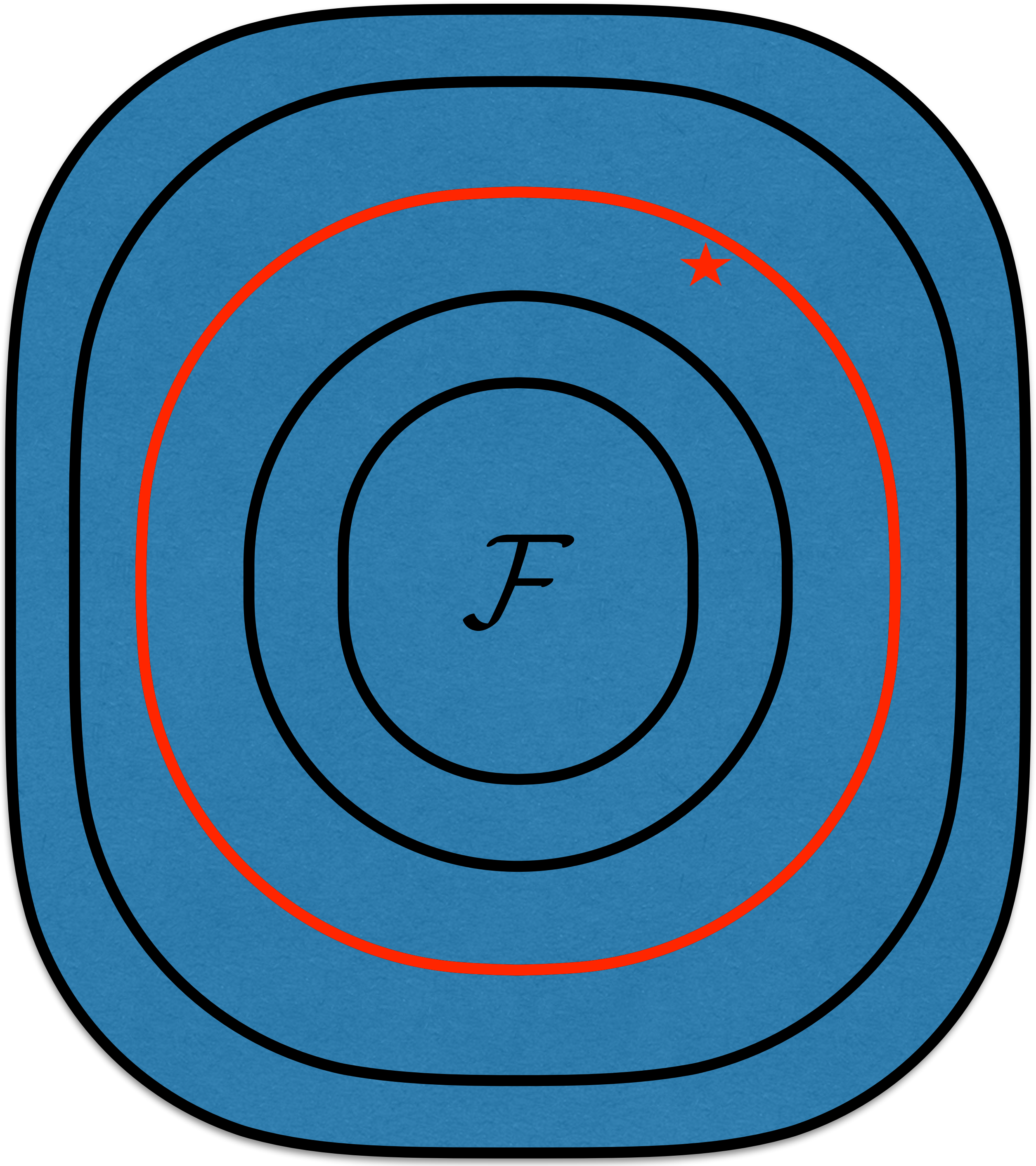
$$\mathcal{F} = \bigcup_r \mathcal{F}_r$$

$$R(f) = \inf \{r : f \in \mathcal{F}_r\}$$

If $R(f)$ is known in advance,

$$\mathbf{Reg}_n(f) \leq \mathcal{R}_n(\mathcal{F}_{R(f)})$$

Online Model Selection



Uniform $\text{Rate}_n(\mathcal{F})$ is large

$$\mathcal{F} = \bigcup_r \mathcal{F}_r$$

$$R(f) = \inf \{r : f \in \mathcal{F}_r\}$$

If $R(f)$ is known in advance,

$$\mathbf{Reg}_n(f) \leq \mathcal{R}_n(\mathcal{F}_{R(f)})$$

How well can we adapt to not knowing $R(f)$?

In statistical learning: [Birge-Massart'98], [Lugosi-Nobel'99], [Bartlett-Boucheron-Lugosi'2002]

Model Adaptation

Corollary

For any class of predictors $\mathcal{F}$ *with* $\mathcal{F}(1)$ *non-empty, for* 1*-Lipschitz loss* ℓ*, the following rate is achievable:*

$$B_n(f) = \tilde{O}\left(\mathcal{R}_n(\mathcal{F}(2R(f)))\sqrt{\log(R(f))}\right)$$

where $R(f) = \min\{r : f \in \mathcal{F}(r)\}$.

Model Adaptation

Corollary

For any class of predictors $\mathcal{F}$ with $\mathcal{F}(1)$ non-empty, for 1-Lipschitz loss ℓ, the following rate is achievable:

$$B_n(f) = \tilde{O}\left(\mathcal{R}_n(\mathcal{F}(2R(f))\sqrt{\log(R(f))}\right)$$

where $R(f) = \min\{r : f \in \mathcal{F}(r)\}$.

Example: unconstrained linear optimization [McMahan-Orabona'14]
$\mathcal{F} = \mathbb{R}^d$, $\mathcal{Y} = \{\mathbf{x} : \|\mathbf{x}\|_2 \leq 1\}$, loss $\ell(\hat{\mathbf{y}}, \mathbf{y}) = \langle \hat{y}, y \rangle$. Define $\mathcal{F}(R) = \left\{f : \|f\|_2 \leq R\right\}$, then,

$$B_n(f) = D\sqrt{n}\left\{8\|f\|_2\left\{1 + \sqrt{\log(2\|f\|_2) + \log\log(2\|f\|_2)}\right\} + 12\right\}.$$

Model Adaptation

Strategy for showing achievability:

- Define collection of RVs in terms of complexity radius: $R_i = \sup_{f \in \mathcal{F}(r_i)} 2 \sum_{t=1}^{n} \epsilon_t f(\mathbf{x}_t(\epsilon))$.
- Establish tail bounds showing $R_i \lesssim B_i$, e.g. $B_i = \mathcal{R}_n(\mathcal{F}(r_i))$.
- Dilate B_i to $B_i \theta_i$ and appeal to **maximal inequality** to bound $\mathbb{E} \sup_i [R_i - B_i \theta_i]$.

Linear example $R_i = 2r_i \left\| \sum_{t=1}^{n} \epsilon_t \mathbf{y}_t(\epsilon) \right\|_2$, $B_i = O(r_i \sqrt{n})$, $\theta_i = O(\sqrt{\log(r_i)})$.

A Simple Probabilistic Tool

Proposition

Let $(R_i)_{i\in I}$ *be a sequence of random variables satisfying: for any* $\tau > 0$,

$$P(R_i - B_i > \tau) \leq C_1 \exp\left(-\tau^2/(2\sigma_i^2)\right)$$

Then $\forall\ \bar{\sigma} \leq \sigma_1$,

$$\mathbb{E}\left[\sup_{i\in I}\left\{R_i - B_i\theta_i\right\}\right] \leq 3C_1\bar{\sigma}$$

where $\theta_i = \frac{\sigma_i}{B_i}\sqrt{2\log(\frac{\sigma_i}{\bar{\sigma}}) + 4\log(i)} + 1.$

A Simple Probabilistic Tool

Proposition

Let $(R_i)_{i \in I}$ *be a sequence of random variables satisfying: for any* $\tau > 0$*,*

$$P(R_i - B_i > \tau) \leq C_1 \exp\left(-\tau^2/(2\sigma_i^2)\right)$$

Then $\forall\ \bar{\sigma} \leq \sigma_1$*,*

$$\mathbb{E}\left[\sup_{i \in I} \{R_i - B_i \theta_i\}\right] \leq 3 C_1 \bar{\sigma}$$

where $\theta_i = \frac{\sigma_i}{B_i}\sqrt{2\log(\frac{\sigma_i}{\bar{\sigma}}) + 4\log(i)} + 1$.

- Model selection example: $\bar{\sigma} = \log^{3/2}(n)\mathcal{R}_n(\mathcal{F}(1))$.

Motivation: Predictable Sequences

- Sequence M_t is our guess for what a good hypothesis looks like.
- Want low regret against hypotheses close to M_t.

Generalized Predictable Sequences

Lemma

Online supervised learning problem with a convex 1*-Lipschitz loss. Let* $(M_t)_{t\geq 1}$ *be any predictable sequence:*

$$B_n(f; x_{1:n}) = \inf_\gamma \left\{ K_1 \sqrt{\log n \cdot \log \mathcal{N}_2(\mathcal{F}, \gamma/2, n) \cdot \left(\sum_{t=1}^n (f(x_t) - M_t)^2\right)} + K_2 \log n \int_{1/n}^{\gamma} \sqrt{n \log \mathcal{N}_2(\mathcal{F}, \delta, n)} d\delta \right\},$$

$\mathcal{N}_2(\mathcal{F}, \gamma, n)$ is sequential analogue of ℓ_2 covering number.

E.G. REGRET TO FIXED VS REGRET TO BEST (SUPERVISED LEARNING)

[Even-Dar-Kearns-Mansour-Wortman'08]

Experts setting: Let $f^\star \in \mathcal{F}$ be a fixed expert chosen in advance:

$$B_n(f, x_{1:n}) = O\left(\log\left(\log N \sum_{t=1}^{n}(f(x_t) - f^\star(x_t))^2\right)\sqrt{\log N \sum_{t=1}^{n}(f(x_t) - f^\star(x_t))^2}\right).$$

In particular, against $f^\star$ we have $B_n(f^\star, x_{1:n}) = O(1)$, and against an arbitrary expert we have $B_n(f, x_{1:n}) = O\left(\sqrt{n \log N}\,(\log(n \cdot \log N))\right)$.

Achieve by taking pred. sequence $M_t = f^\star(x_t)$.

Optimistic Online PAC-Bayes

- Online version of PAC Bayes theorem [McAllester'98].
- $\mathcal{F}$ set of distributions over class of experts, π is some prior over experts

$$B_n(f; y_{1:n}) = O\left(\sqrt{50\left(\mathrm{KL}(f|\pi) + \log(n)\right)\sum_{t=1}^{n}\mathbb{E}_{e\sim f}\ell(e, y_t)^2}\right)$$

Related to [Luo-Schapire'15], [Koolen-van Erven'15]

Optimistic Online PAC-Bayes

- Online version of PAC Bayes theorem [McAllester'98].
- $\mathcal{F}$ set of distributions over class of experts, π is some prior over experts

$$B_n(f; y_{1:n}) = O\left(\sqrt{50\left(\mathrm{KL}(f|\pi) + \log(n)\right)\sum_{t=1}^{n}\mathbb{E}_{e\sim f}\ell(e, y_t)^2}\right)$$

Related to [Luo-Schapire'15], [Koolen-van Erven'15]

- We also recover [Chaudhuri-Freund-Hsu'09]:

$$\forall \epsilon > 0, \text{Regret against top } \epsilon|\mathcal{F}| \text{ experts} \leq \sqrt{n \log \epsilon^{-1}}$$

Adaptive Relaxation for Algorithms

Extends [Rakhlin-Shamir-Sridharan'12]

- Find mapping $\mathbf{Rel}_n : \bigcup_{t=0}^{n} (\mathcal{X} \times \mathcal{Y})^t \to \mathbb{R}$ satisfying initial condition:

$$\mathbf{Rel}_n\left(x_{1:n}, y_{1:n}\right) \geq \sup_{f \in \mathcal{F}} \left\{ -\sum_{t=1}^{n} \ell(f(x_t), y_t) - B_n(f; x_{1:n}, y_{1:n}) \right\}$$

- Admissibility condition,

$$\mathbf{Rel}_n\left(x_{1:t-1}, y_{1:t-1}\right) \geq \sup_{x_t} \inf_{q_t} \sup_{y_t} \mathbb{E}_{\hat{y}_t \sim q_t}\left[\ell(\hat{y}_t, y_t) + \mathbf{Rel}_n\left(x_{1:t}, y_{1:t}\right)\right]$$

- Algorithm:

$$q_t = \operatorname{argmin}_q \sup_{y_t} \mathbb{E}_{\hat{y}_t \sim q}\left[\ell(\hat{y}_t, y_t) + \mathbf{Rel}_n\left(x_{1:t}, y_{1:t}\right)\right]$$

- Algorithm achieves the following bound:

$$\mathbf{Reg}_n \leq B_n(f; x_{1:n}, y_{1:n}) + \mathbf{Rel}_n\left(\cdot\right)$$

SUMMARY

- Sufficient condition for establishing achievability of adaptive rate.
- For specific settings condition also necessary.
- Obtain unconstrained optimization, model adaptation, optimistic PAC Bayes, quantile bound etc.
- Sketch of schema for deriving adaptive algorithms.

Further Directions

- More general techniques for going from bounds to algorithms?
- Apply to game theory.
- Apply to approximation algorithms.
- Further explore data and model priors.