Competing With Strategies

Wei Han  Alexander Rakhlin  Karthik Sridharan

University of Pennsylvania
Online Learning with Strategies

Learner picks $f_t \in F$

Adversary simultaneously picks instance $z_t \in Z$

Learner suffers loss $\ell(f_t, z_t)$

**Goal**: minimize regret

$$\text{Reg}_T = T \sum_{t=1}^{T} \ell(f_t, z_t) - \inf_{f \in F} T \sum_{t=1}^{T} \ell(f, z_t)$$

Wednesday, July 24, 13
For $t = 1$ to $T$

End
For \( t = 1 \) to \( T \)

Learner picks \( f_t \in \mathcal{F} \)

End
For $t = 1$ to $T$
   Learner picks $f_t \in \mathcal{F}$
   Adversary simultaneously picks instance $z_t \in \mathcal{Z}$

End
For $t = 1$ to $T$
  Learner picks $f_t \in \mathcal{F}$
  Adversary simultaneously picks instance $z_t \in \mathcal{Z}$
  Learner suffers loss $\ell(f_t, z_t)$
End
For $t = 1$ to $T$
   Learner picks $f_t \in \mathcal{F}$
   Adversary simultaneously picks instance $z_t \in \mathcal{Z}$
   Learner suffers loss $\ell(f_t, z_t)$
End

Goal: minimize regret

$$\text{Reg}_T = \sum_{t=1}^{T} \ell(f_t, z_t) - \inf_{f \in \mathcal{F}} \sum_{t=1}^{T} \ell(f, z_t)$$
Online Learning with Strategies

For $t = 1$ to $T$

Learner picks $f_t \in \mathcal{F}$
Adversary simultaneously picks instance $z_t \in \mathcal{Z}$
Learner suffers loss $\ell(f_t, z_t)$

End

Goal: minimize regret

$$\text{Reg}_T = \sum_{t=1}^{T} \ell(f_t, z_t) - \inf_{\pi \in \Pi} \sum_{t=1}^{T} \ell(\pi_t(z_{1:t-1}), z_t)$$

where each $\pi \in \Pi$ is a sequence $(\pi_1, \ldots, \pi_T)$ with $\pi_t : \mathcal{Z}^{t-1} \rightarrow \mathcal{F}$

$\Pi$: set of online learning algorithms we want to compete with
1. Tools to study learning rates while competing with strategies

2. Generic recipe for building algorithms for these problems.
\[ \mathcal{V}_T(\Pi, \ell) := \inf_{\text{Randomized Algorithm}} \sup_{\text{Adversary}} \mathbb{E}[\text{Reg}_T] \]
\[ \mathcal{V}_T(\Pi, \ell) := \inf_{\text{Randomized Algorithm}} \sup_{\text{Adversary}} \mathbb{E}[\text{Reg}_T] \]

\[ := \inf_{q_1 \in \Delta(\mathcal{F})} \sup_{z_1 \in \mathcal{Z}} \mathbb{E}_{f_1 \sim q_1} \ldots \inf_{q_T \in \Delta(\mathcal{F})} \sup_{z_T \in \mathcal{Z}} \mathbb{E}_{f_T \sim q_T} [\text{Reg}_T] \]
Minimax Rate

\[ \mathcal{V}_T(\Pi, \ell) := \inf_{\text{Randomized Algorithm}} \sup_{\text{Adversary}} \mathbb{E}[\text{Reg}_T] \]

\[ := \inf_{q_1 \in \Delta(\mathcal{F})} \sup_{z_1 \in \mathcal{Z}} \mathbb{E}[f_1 \sim q_1 \ldots \inf_{q_T \in \Delta(\mathcal{F})} \sup_{z_T \in \mathcal{Z}} \mathbb{E}[f_T \sim q_T]\text{[Reg}_T]\]

- Best possible regret guarantee
  - There exists an algorithm with expected regret at most \( \mathcal{V}_T(\Pi, \ell) \)
  - No algorithm can guarantee expected regret smaller than \( \mathcal{V}_T(\Pi, \ell) \)
Minimax Rate

$$\mathcal{V}_T(\Pi, \ell) := \inf_{\text{Randomized Algorithm}} \sup_{\text{Adversary}} \mathbb{E}[\text{Reg}_T]$$

$$:= \inf_{q_1 \in \Delta(\mathcal{F})} \sup_{z_1 \in Z} \mathbb{E}_{f_1 \sim q_1} \ldots \inf_{q_T \in \Delta(\mathcal{F})} \sup_{z_T \in Z} \mathbb{E}_{f_T \sim q_T} [\text{Reg}_T]$$

- Best possible regret guarantee
  - There exists an algorithm with expected regret at most $$\mathcal{V}_T(\Pi, \ell)$$
  - No algorithm can guarantee expected regret smaller than $$\mathcal{V}_T(\Pi, \ell)$$

- Key idea: bound $$\mathcal{V}_T(\Pi, \ell)$$ in terms of a stochastic quantity
  [Abernethy et al ‘09], [Rakhlin, S., Tewari ’10]
\[ \mathcal{V}_T(\Pi, \ell) := \inf_{\text{Randomized Algorithm}} \sup_{\text{Adversary}} \mathbb{E}[\text{Reg}_T] \]

\[ := \inf_{q_1 \in \Delta(\mathcal{F})} \sup_{z_1 \in \mathcal{Z}} \mathbb{E}_{f_1 \sim q_1} \cdots \inf_{q_T \in \Delta(\mathcal{F})} \sup_{z_T \in \mathcal{Z}} \mathbb{E}_{f_T \sim q_T} \text{Reg}_T \]

- **Best possible regret guarantee**
  - There exists an algorithm with expected regret at most \( \mathcal{V}_T(\Pi, \ell) \)
  - No algorithm can guarantee expected regret smaller than \( \mathcal{V}_T(\Pi, \ell) \)

- **Key idea:** bound \( \mathcal{V}_T(\Pi, \ell) \) in terms of a stochastic quantity
  - [Abernethy et al ’09], [Rakhlin, S., Tewari ’10]

- **On similar lines as** [Rakhlin, S., Tewari ’10] introduce sequential complexity tools for strategies
**Sequential Rademacher Complexities**

\[
\mathcal{R}_T(\ell, \Pi) \triangleq \sup_{w,z} \mathbb{E}_\epsilon \sup_{\pi \in \Pi} \left[ \sum_{t=1}^{T} \epsilon_t \ell(\pi_t(w_1(\epsilon), \ldots, w_{t-1}(\epsilon)), z_t(\epsilon)) \right]
\]

where \(w, z\) are \(Z\)-valued trees of depth \(T\). (Each \(z_t: \{\pm 1\}^{t-1} \rightarrow Z\).)

**Lemma**

For any class \(\Pi\) of strategies,

\[
\mathcal{R}_T(\ell, \Pi) \leq 2 R_T(\ell, \Pi)
\]

**Examples:**

- **History independent strategies:** \(\pi_1(\epsilon) = \pi_2(\epsilon) = \cdots = \pi_T(\epsilon)\) recovers sequential Rademacher complexity of \(F\).
  
  [Rakhlin, S., Tewari ’10]

- **Static experts:** \(\pi_t(z_1: t-1) = \pi_t(z)\) recovers classical Rademacher complexity.
  
  [Cesa-Bianchi, Lugosi ’99]
**Sequential Rademacher Complexity**

\[ \mathcal{R}_T(\ell, \Pi) = \sup_{w,z} \mathbb{E}_\epsilon \sup_{\pi \in \Pi} \left[ \sum_{t=1}^T \epsilon_t \ell(\pi_t(w_1(\epsilon), \ldots, w_{t-1}(\epsilon)), z_t(\epsilon)) \right] \]

where \( w, z \) are \( \mathcal{Z} \)-valued tree of depth \( T \). (each \( z_t : \{\pm 1\}^{t-1} \rightarrow \mathcal{Z} \).)
**Sequential Rademacher Complexity**

\[
\mathcal{R}_T(\mathcal{L}, \mathcal{A}) \triangleq \sup_{\mathbf{w}, \mathbf{z}} \mathbb{E}_\epsilon \sup_{\pi \in \Pi} \left[ \sum_{t=1}^{T} \epsilon_t \ell(\pi_t(\mathbf{w}_1(\epsilon), \ldots, \mathbf{w}_{t-1}(\epsilon)), \mathbf{z}_t(\epsilon)) \right]
\]

where \( \mathbf{w}, \mathbf{z} \) are \( \mathcal{Z} \)-valued tree of depth \( T \). (each \( \mathbf{z}_t : \{\pm 1\}^{t-1} \mapsto \mathcal{Z} \)).

\( \epsilon = (+1, -1, -1) \)
Sequential Rademacher Complexity

\[ \mathcal{R}_T(\ell, \Pi) \triangleq \sup_{w,z} \mathbb{E}_\epsilon \sup_{\pi \in \Pi} \left[ \sum_{t=1}^{T} \epsilon_t \ell(\pi_t(w_1(\epsilon), \ldots, w_{t-1}(\epsilon)), z_t(\epsilon)) \right] \]

where \( w, z \) are \( \mathcal{Z} \)-valued tree of depth \( T \). (each \( z_t : \{\pm 1\}^{t-1} \rightarrow \mathcal{Z} \)).
\[ \mathcal{R}_T(\ell, \Pi) \equiv \sup_{w, z} \mathbb{E}_\epsilon \sup_{\pi \in \Pi} \left[ \sum_{t=1}^{T} \epsilon_t \ell(\pi_t(w_1(\epsilon), \ldots, w_{t-1}(\epsilon)), z_t(\epsilon)) \right] \]

where \( w, z \) are \( \mathcal{Z} \)-valued tree of depth \( T \). (each \( z_t : \{\pm 1\}^{t-1} \rightarrow \mathcal{Z} \)).

\[ \sum_{t=1}^{T} \epsilon_t \ell(\pi_t(w_1(\epsilon), \ldots, w_{t-1}(\epsilon)), z_t(\epsilon)) = +\ell(\pi_1(\cdot), z_1) - \ell(\pi_2(w_1), z_3) - \ell(\pi_3(w_1, w_3), z_6) \]
**Sequential Rademacher Complexity**

\[
\mathcal{R}_T (\ell, \Pi) \triangleq \sup_{w,z} \mathbb{E}_\epsilon \sup_{\pi \in \Pi} \left[ \sum_{t=1}^{T} \epsilon_t \ell(\pi_t(w_1(\epsilon), \ldots, w_{t-1}(\epsilon)), z_t(\epsilon)) \right]
\]

where \( w, z \) are \( \mathcal{Z} \)-valued tree of depth \( T \). (each \( z_t : \{\pm 1\}^{t-1} \mapsto \mathcal{Z} \)).

\[
\sum_{t=1}^{T} \epsilon_t \ell(\pi_t(w_1(\epsilon), \ldots, w_{t-1}(\epsilon)), z_t(\epsilon)) = \ell(\pi_1(\cdot), z_1) - \ell(\pi_2(w_1), z_3) - \ell(\pi_3(w_1, w_3), z_6)
\]

\( \epsilon = (+1, -1, -1) \)
Sequential Rademacher Complexity

\[ \mathcal{R}_T(\ell, \Pi) = \sup_{w, z} \mathbb{E}_\varepsilon \sup_{\pi \in \Pi} \left[ \sum_{t=1}^{T} \varepsilon_t \ell(\pi_t(w_1(\varepsilon), \ldots, w_{t-1}(\varepsilon), z_t(\varepsilon)) \right] \]

where \( w, z \) are \( \mathcal{Z} \)-valued tree of depth \( T \). (each \( z_t : \{\pm 1\}^{t-1} \rightarrow \mathcal{Z} \).

\[ \sum_{t=1}^{T} \varepsilon_t \ell(\pi_t(w_1(\varepsilon), \ldots, w_{t-1}(\varepsilon), z_t(\varepsilon)) \]

\[ = +\ell(\pi_1(\cdot), z_1) - \ell(\pi_2(w_1), z_3) - \ell(\pi_3(w_1, w_3), z_6) \]
Sequential Rademacher Complexity

\[
\mathcal{R}_T(\ell, \Pi) \triangleq \sup_{w,z} \mathbb{E}_\pi \sup_{\pi \in \Pi} \left[ \sum_{t=1}^{T} \epsilon_t \ell(\pi_t(w_1(\epsilon), \ldots, w_{t-1}(\epsilon)), z_t(\epsilon)) \right]
\]

where \(w, z\) are \(\mathcal{Z}\)-valued tree of depth \(T\). (each \(z_t : \{\pm 1\}^{t-1} \mapsto \mathcal{Z}\)).

Lemma

For any class \(\Pi\) of strategies,

\[
\mathcal{V}_T(\Pi, \ell) \leq 2 \mathcal{R}_T(\ell, \Pi)
\]

Wednesday, July 24, 13
**Sequential Rademacher Complexity**

\[ \mathcal{R}_T(\ell, \Pi) \triangleq \sup_{w,z} \mathbb{E}_\epsilon \sup_{\pi \in \Pi} \left[ \sum_{t=1}^{T} \epsilon_t \ell(\pi_t(w_1(\epsilon), \ldots, w_{t-1}(\epsilon)), z_t(\epsilon)) \right] \]

where \( w, z \) are \( \mathcal{Z} \)-valued tree of depth \( T \). (each \( z_t : \{\pm 1\}^{t-1} \mapsto \mathcal{Z} \)).

**Lemma**

*For any class \( \Pi \) of strategies,*

\[ \nu_T(\Pi, \ell) \leq 2 \mathcal{R}_T(\ell, \Pi) \]

**Examples:**

- **History independent strategies:** \( \pi_1 = \pi_2 = \ldots = \pi_T \) recovers sequential Rademacher complexity of \( \mathcal{F} \). [Rakhlin, S., Tewari ’10]
- **Individual sequence prediction:** \( \pi_t(z_{1:t-1}) = \pi_t \) recovers classical Rademacher complexity. [Cesa-Bianchi, Lugosi ’99]
Other Sequential Complexity Measures

- Sequential covering number for strategies
- Bound minimax rate in terms of these covering numbers
- Contraction lemma and other structural properties
Other Sequential Complexity Measures

- Sequential covering number for strategies
- Bound minimax rate in terms of these covering numbers
- Contraction lemma and other structural properties

_bounds are non-constructive._
Sequential covering number for strategies

Bound minimax rate in terms of these covering numbers

Contraction lemma and other structural properties

Bounds are non-constructive.

What about the algorithms?
THE RELAXATION MECHANISM

Initial condition: [Rakhlin, Shamir, S. ’12]

\[ \text{Rel}_T(\prod|z_1, \ldots, z_T) \geq -\inf_{\pi \in \Pi} \sum_{t=1}^{T} \ell(\pi_t(z_{1:t-1}), z_t) \]

Admissibility:

\[ \inf_{q_t \in \Delta(F)} \sup_{z_t \in Z} \left\{ \mathbb{E}_{f_t \sim q_t} \ell(f_t, z_t) + \text{Rel}_T(\prod|z_1, \ldots, z_t) \right\} \leq \text{Rel}_T(\prod|z_1, \ldots, z_{t-1}) \]
**THE RELAXATION MECHANISM**

Initial condition: \[ \text{Rel}_T(\prod|z_1, \ldots, z_T) \geq -\inf_{\pi \in \Pi} \sum_{t=1}^{T} \ell(\pi_t(z_{1:t-1}), z_t) \]

Admissibility:
\[
\inf_{q_t \in \Delta(\mathcal{F})} \sup_{z_t \in \mathcal{Z}} \left\{ \mathbb{E}_{f_t \sim q_t} \ell(f_t, z_t) + \text{Rel}_T(\prod|z_1, \ldots, z_t) \right\} \leq \text{Rel}_T(\prod|z_1, \ldots, z_{t-1})
\]

Algorithm:
\[
q_t = \arg\min_{q_t \in \Delta(\mathcal{F})} \sup_{z_t \in \mathcal{Z}} \left\{ \mathbb{E}_{f_t \sim q_t} \ell(f_t, z_t) + \text{Rel}_T(\prod|z_1, \ldots, z_t) \right\}
\]

[Рахлин, Шамир, С. ’12]
The Relaxation Mechanism

Initial condition: [Rakhlin, Shamir, S. ’12]
\[ \text{Rel}_T(\prod|z_1, \ldots, z_T) \geq - \inf_{\pi \in \prod} \sum_{t=1}^{T} \ell(\pi_t(z_{1:t-1}), z_t) \]

Admissibility:
\[ \inf_{q_t \in \Delta(F)} \sup_{z_t \in Z} \left\{ \mathbb{E}_{f_t \sim q_t} \ell(f_t, z_t) + \text{Rel}_T(\prod|z_1, \ldots, z_t) \right\} \leq \text{Rel}_T(\prod|z_1, \ldots, z_{t-1}) \]

Algorithm:
\[ q_t = \arg\min_{q_t \in \Delta(F)} \sup_{z_t \in Z} \left\{ \mathbb{E}_{f_t \sim q_t} \ell(f_t, z_t) + \text{Rel}_T(\prod|z_1, \ldots, z_t) \right\} \]

Lemma

Expected regret of Algorithm associated with \( \text{Rel}_T \) is bounded as
\[ \mathbb{E} \left[ \text{Reg}_T \right] \leq \text{Rel}_T(\prod) \]
Sequential Rademacher relaxation for strategies:

$$\mathcal{R}_T(\Pi|z_{1:t}) = \sup_{z,w} \mathbb{E}_e \sup_{\pi \in \Pi} \left[ 2 \sum_{s=t+1}^T \epsilon_s \ell(\pi_s((z_{1:t}, w_{1:s-t-1}(\epsilon)), z_{s-t}(\epsilon)) - \sum_{s=1}^t \ell(\pi_s(z_{1:s-1}), z_s) \right]$$
Sequential Rademacher relaxation for strategies:

$$\mathcal{R}_T(\Pi|z_{1:t}) = \sup_{z, w} \sup_{\pi \in \Pi} \mathbb{E}_\epsilon \left[ 2 \sum_{s=t+1}^{T} \epsilon_s \ell(\pi_s((z_{1:t}, w_{1:s-t-1}(\epsilon)), z_{s-t}(\epsilon)) - \sum_{s=1}^{t} \ell(\pi_s(z_{1:s-1}), z_s) \right]$$

**Lemma**

The sequential Rademacher relaxation for any class of strategies $\Pi$ is admissible and regret of corresponding algorithm is bounded as:

$$\mathbb{E} \left[ \text{Reg}_T \right] \leq \mathcal{R}_T(\Pi)$$
Steps:

1. Formulate sequential Rademacher relaxation, $\mathcal{R}_T(\Pi|z_{1:t})$ for give problem.
Steps:

1. Formulate sequential Rademacher relaxation, $\mathcal{R}_T(\Pi|z_{1:t})$ for give problem.

2. Get rid of the trees $w$ and $z$ and go to upper bound $\text{Rel}_T$:

$$\mathcal{R}_T(\ell, \Pi|z_{1:t}) \leq \text{Rel}_T (\Pi|z_{1:t})$$
Steps:

1. Formulate sequential Rademacher relaxation, $\mathcal{R}_T(\Pi|z_{1:t})$ for given problem.

2. Get rid of the trees $w$ and $z$ and go to upper bound $\text{Rel}_T$:

   $$\mathcal{R}_T(\ell, \Pi|z_{1:t}) \leq \text{Rel}_T(\Pi|z_{1:t})$$

3. Ensure admissibility of $\text{Rel}_T$
Steps:

1. Formulate sequential Rademacher relaxation, $\mathcal{R}_T(\Pi|z_{1:t})$ for give problem.

2. Get rid of the trees $w$ and $z$ and go to upper bound $\text{Rel}_T$:

   $\mathcal{R}_T(\ell, \Pi|z_{1:t}) \leq \text{Rel}_T(\Pi|z_{1:t})$

3. Ensure admissibility of $\text{Rel}_T$

4. Algorithm: Solve optimization problem

   $$q_t = \arg\min_{q_t} \sup_{z_t \in \mathcal{Z}} \left\{ \mathbb{E}_{f_t \sim q_t} \ell(f_t, z_t) + \text{Rel}_T(\Pi|z_1, \ldots, z_t) \right\}$$
Problem setup: binary sequence prediction
**Example : MAP algorithms**

Problem setup : binary sequence prediction

- \( Z = \{0, 1\} \), \( \ell(f, z) = |f - z| \), \( \Pi = \{\pi^{\alpha, \beta} : \alpha > 1, \beta \in (1, C_\beta]\} \)

\[
\pi_t^{\alpha, \beta}(z_1, \ldots, z_{t-1}) = \frac{\sum_{i=1}^{t-1} z_i + \alpha - 1}{t - 1 + \alpha + \beta - 2}
\]
EXAMPLE: MAP ALGORITHMS

Problem setup: binary sequence prediction
- $\mathcal{Z} = \{0, 1\}$, $\ell(f, z) = |f - z|$, $\Pi = \{\pi^{\alpha, \beta} : \alpha > 1, \beta \in (1, C_\beta]\}$

$$\pi_t^{\alpha, \beta}(z_1, \ldots, z_{t-1}) = \frac{\sum_{i=1}^{t-1} z_i + \alpha - 1}{t - 1 + \alpha + \beta - 2}$$

- MAP estimate under Bernoulli likelihood and $Beta(\alpha, \beta)$ prior
EXAMPLE : MAP ALGORITHMS

Problem setup : binary sequence prediction
- \( \mathcal{Z} = \{0, 1\} \), \( \ell(f, z) = |f - z| \), \( \Pi = \{ \pi_{\alpha, \beta} : \alpha > 1, \beta \in (1, C_\beta) \} \)

\[ \pi_{t}^{\alpha, \beta}(z_1, \ldots, z_{t-1}) = \frac{\sum_{i=1}^{t-1} z_i + \alpha - 1}{t - 1 + \alpha + \beta - 2} \]

- MAP estimate under Bernoulli likelihood and \( \text{Beta}(\alpha, \beta) \) prior
- Relaxation :

\[ \text{Rel}(z_{1:t}) = \mathbb{E}_{\varepsilon_{t+1:T}} \sup_{\alpha, \beta} \left[ 2 \sum_{s=t+1}^{T} \varepsilon_s \cdot \frac{s + \alpha - 2}{s + \alpha + \beta - 3} - \sum_{s=1}^{t} \left| \frac{\sum_{i=1}^{s-1} z_i}{s + \alpha + \beta - 3} - z_s \right| \right] \]
Example: MAP Algorithms

Problem setup: binary sequence prediction
- \( \mathcal{Z} = \{0, 1\} \), \( \ell(f, z) = |f - z| \), \( \Pi = \{\pi^{\alpha, \beta} : \alpha > 1, \beta \in (1, C_\beta]\} \)

\[ \pi_t^{\alpha, \beta}(z_1, \ldots, z_{t-1}) = \frac{\sum_{i=1}^{t-1} z_i + \alpha - 1}{t - 1 + \alpha + \beta - 2} \]

- MAP estimate under Bernoulli likelihood and \( Beta(\alpha, \beta) \) prior
- Relaxation:

\[ \text{Rel}(z_{1:t}) = \mathbb{E}_{\epsilon_{t+1:T}} \sup_{\alpha, \beta} \left[ 2 \sum_{s=t+1}^{T} \epsilon_s \cdot \frac{s + \alpha - 2}{s + \alpha + \beta - 3} - \sum_{s=1}^{t} \left| \frac{\sum_{i=1}^{s-1} z_i}{s + \alpha + \beta - 3} - z_s \right| \right] \]

- Algorithm:

\[
q_t = \frac{1}{2} \left\{ \sup_{\alpha, \beta} \left[ 2 \sum_{s=t+1}^{T} \epsilon_s \cdot \frac{s + \alpha - 2}{s + \alpha + \beta - 3} - \sum_{s=1}^{t-1} (1 - 2z_s) \cdot \frac{\sum_{i=1}^{s-1} z_i}{s + \alpha + \beta - 3} + \frac{\sum_{i=1}^{t-1} z_i}{t + \alpha + \beta - 3} \right] 
- \sup_{\alpha, \beta} \left[ 2 \sum_{s=t+1}^{T} \epsilon_s \cdot \frac{s + \alpha - 2}{s + \alpha + \beta - 3} - \sum_{s=1}^{t-1} (1 - 2z_s) \cdot \frac{\sum_{i=1}^{s-1} z_i}{s + \alpha + \beta - 3} - \frac{\sum_{i=1}^{t-1} z_i}{t + \alpha + \beta - 3} \right] \right\}
\]
Problem setup: binary sequence prediction

- $\mathcal{Z} = \{0, 1\}$, $\ell(f, z) = |f - z|$, $\Pi = \{\pi_{\alpha, \beta} : \alpha > 1, \beta \in (1, C_{\beta}]\}$

$$\pi_{t}^{\alpha, \beta}(z_1, \ldots, z_{t-1}) = \frac{\sum_{i=1}^{t-1} z_i + \alpha - 1}{t - 1 + \alpha + \beta - 2}$$

- MAP estimate under Bernoulli likelihood and $\text{Beta}(\alpha, \beta)$ prior
- Relaxation:

$$\text{Rel}(z_{1:t}) = \mathbb{E}_{e_{t+1:T}} \sup_{\alpha, \beta} \left[ 2 \sum_{s=t+1}^{T} e_s \cdot \frac{s + \alpha - 2}{s + \alpha + \beta - 3} - \sum_{s=1}^{t} \left| \frac{\sum_{i=1}^{s-1} z_i}{s + \alpha + \beta - 3} - z_s \right| \right]$$

- Algorithm:
  - Time complexity for iteration $t$: $O(t \log t)$
  - Expected regret bounded as: $O(\sqrt{T})$
Example: MAP Algorithms

Problem setup: binary sequence prediction
- $\mathcal{Z} = \{0, 1\}$, $\ell(f, z) = |f - z|$, $\Pi = \{\pi^{\alpha, \beta} : \alpha > 1, \beta \in (1, C_\beta]\}$

- $\pi_t^{\alpha, \beta}(z_1, \ldots, z_{t-1}) = \frac{\sum_{i=1}^{t-1} z_i + \alpha - 1}{t - 1 + \alpha + \beta - 2}$

- MAP estimate under Bernoulli likelihood and $\text{Beta}(\alpha, \beta)$ prior
- Relaxation:

\[
\text{Rel}(z_1:t) = \mathbb{E}_{\epsilon_{t+1:T}} \sup_{\alpha, \beta} \left[ 2 \sum_{s=t+1}^{T} \epsilon_s \cdot \frac{s + \alpha - 2}{s + \alpha + \beta - 3} - \sum_{s=1}^{t} \left| \frac{\sum_{i=1}^{s-1} z_i}{s + \alpha + \beta - 3} - z_s \right| \right]
\]

Algorithm:
- Time complexity for iteration $t$: $O(t \log t)$
- Expected regret bounded as: $O(\sqrt{T})$
- Experts gives $O(\sqrt{T \log T})$ regret bound and worse time complexity in $T$
Example: Auto-regressive Model

- $\mathcal{F} = \mathcal{Z} = [-1, 1], \ l(f, z) = (f - z)^2, \ \Pi = \{\pi^\theta : \|\theta\|_1 \leq 1, \theta \in \mathbb{R}^T\}$

$$\pi_t^\theta (z_1, \ldots, z_{t-1}) = \sum_{i=1}^{t-1} \theta_i z_i$$
Example: Auto-regressive Model

- $\mathcal{F} = \mathcal{Z} = [-1, 1]$,
  $\ell(f, z) = (f - z)^2$,
  $\Pi = \{\pi^\theta : \|\theta\|_1 \leq 1, \theta \in \mathbb{R}^T\}$

$$
\pi^\theta_t(z_1, \ldots, z_{t-1}) = \sum_{i=1}^{t-1} \theta_i z_i
$$

- Relaxation: let $a^t_s(\epsilon) = 2\epsilon_s$ for $s > t$, and $-z_s$ otherwise

$$
\text{Rel}(z_{1:t}) = \mathbb{E}_{\epsilon_{t+1:T}} \max_{1 \leq s \leq T} \left| \sum_{i=s}^{T} a^t_i(\epsilon) \right|
$$
**Example: Auto-regressive Model**

- \( \mathcal{F} = \mathcal{Z} = [-1, 1] \), \( \ell(f, z) = (f - z)^2 \), \( \Pi = \{\pi^\theta : \|\theta\|_1 \leq 1, \theta \in \mathbb{R}^T\} \)

\[
\pi_t^\theta (z_1, \ldots, z_{t-1}) = \sum_{i=1}^{t-1} \theta_i z_i
\]

- Relaxation: let \( a^t_s(\epsilon) = 2\epsilon_s \) for \( s > t \), and \(-z_s\) otherwise

\[
\text{Rel}(z_1:t) = \mathbb{E}_{\epsilon_{t+1:T}} \max_{1 \leq s \leq T} \left| \sum_{i=s}^{T} a^t_i(\epsilon) \right|
\]

- Algorithm:

\[
q_t(\epsilon) = \frac{1}{2} \left( \max \left\{ \max_{s=1,\ldots,t} \left| - \sum_{i=s}^{t-1} z_i + 1 + 2 \sum_{i=t+1}^{T} \epsilon_i \right| , \max_{s=t+1,\ldots,T} \left| 2 \sum_{i=s}^{T} \epsilon_i \right| \right\}
\]

\[
- \max \left\{ \max_{s=1,\ldots,t} \left| - \sum_{i=s}^{t-1} z_i - 1 + 2 \sum_{i=t+1}^{T} \epsilon_i \right| , \max_{s=t+1,\ldots,T} \left| 2 \sum_{i=s}^{T} \epsilon_i \right| \right\}
\]
**Example: Auto-regressive Model**

- \( \mathcal{F} = \mathcal{Z} = [-1, 1] \), \( \ell(f, z) = (f - z)^2 \), \( \Pi = \{ \pi^\theta : \|\theta\|_1 \leq 1, \theta \in \mathbb{R}^T \} \)

\[
\pi^\theta_t (z_1, \ldots, z_{t-1}) = \sum_{i=1}^{t-1} \theta_i z_i
\]

- Relaxation: let \( a^t_s (\varepsilon) = 2\varepsilon_s \) for \( s > t \), and \( -z_s \) otherwise

\[
\text{Rel}(z_{1:t}) = \mathbb{E}_{\varepsilon_{t+1:T}} \max_{1 \leq s \leq T} \left| \sum_{i=s}^T a^t_i (\varepsilon) \right|
\]

- Algorithm:

\[
q_t (\varepsilon) = \frac{1}{2} \left( \max \left\{ \max_{s=1,\ldots,t} \left| - \sum_{i=s}^{t-1} z_i + 1 + 2 \sum_{i=t+1}^T \varepsilon_i \right|, \max_{s=t+1,\ldots,T} \left| 2 \sum_{i=s}^T \varepsilon_i \right| \right\} 
- \max \left\{ \max_{s=1,\ldots,t} \left| - \sum_{i=s}^{t-1} z_i - 1 + 2 \sum_{i=t+1}^T \varepsilon_i \right|, \max_{s=t+1,\ldots,T} \left| 2 \sum_{i=s}^T \varepsilon_i \right| \right\} \right)
\]

- Regret bound \( O(\sqrt{T}) \), time complexity per round \( O(T) \)
- Time complexity can be improved to \( O(1) \) per round
Finite-order Markov strategies

Finite look-back autoregression models

Autoregression with full dependence on history and but geometric restrictions on $\theta$’s

Strategies that compress past into some *sufficient statistic*

Competing with families of regularized least squares algorithms

Competing with Follow the Regularized Leader Strategies
FURTHER DIRECTIONS

Competing with general class of Bayesian algorithms indexed by family of priors
Competing with strategies when instances are more predictive rather than worst-case
Derive efficient algorithms for competing with family of regularized least-squares
Derive more efficient algorithms for competing with family of follow regularized leader algorithms
Competing with general class of Bayesian algorithms indexed by family of priors
Further Directions

- Competing with general class of Bayesian algorithms indexed by family of priors
- Competing with strategies when instances are more *predictive* rather than worst-case
Competing with general class of Bayesian algorithms indexed by family of priors

Competing with strategies when instances are more *predictive* rather than worst-case

Derive *efficient* algorithms for competing with family of regularized least-squares
Further Directions

- Competing with general class of Bayesian algorithms indexed by family of priors
- Competing with strategies when instances are more predictive rather than worst-case
- Derive efficient algorithms for competing with family of regularized least-squares
- Derive more efficient algorithms for competing with family of follow regularized leader algorithms
Thank you

?