On the Generalization Ability of Online Learning Algorithms for Pairwise Loss Functions

Purushottam Kar*, Bharath Sriperumbudur†, Prateek Jain§ and Harish Karnick*

* Indian Institute of Technology Kanpur
† Center for Mathematical Sciences, University of Cambridge
§ Microsoft Research India

International Conference on Machine Learning 2013
Pointwise Loss Functions

Loss functions for classification, regression ..

\[ \ell : \mathcal{H} \times \mathcal{Z} \rightarrow \mathbb{R} \]

.. look at only one point \( z = (x, y) \) at a time

Examples:

- Hinge loss: \( \ell(h, z) = [1 - y \cdot h(x)]_+ \)
- \( \epsilon \)-insensitive loss: \( \ell(h, z) = [|y - h(x)| - \epsilon]_+ \)
- Logistic loss: \( \ell(h, z) = \ln (1 + \exp (y \cdot h(x))) \)
Metric Learning for Classification

Metric needs to be penalized for bringing **blue** and **red** points together
Metric Learning for Classification

Metric needs to be penalized for bringing blue and red points together

- Loss function needs to consider two data points at a time
  - in other words, a pairwise loss function
- **Example:** \( \ell(d_M, z_1, z_2) = \phi(y_1y_2(1 - d_M^2(x_1, x_2))) \)
  - where \( \phi \) is the hinge loss function
Learning with Pairwise Loss Functions

\[ \ell : \mathcal{H} \times Z \times Z \rightarrow \mathbb{R} \]

Examples:

- Mahalanobis metric learning
- Bipartite ranking / maximizing area under ROC curve
- Preference learning
- Two-stage Multiple kernel learning
- Similarity (indefinite kernel) learning
Learning with Pairwise Loss Functions

\[ \ell : \mathcal{H} \times \mathcal{Z} \times \mathcal{Z} \rightarrow \mathbb{R} \]

Online Learning for Pairwise Loss Functions

- **Algorithmic Challenges**
  - Attempts to reduce to pointwise learning
  - Treat pairs \((z_i, z_j)\) as elements of a superdomain \(\tilde{\mathcal{Z}} = \mathcal{Z} \times \mathcal{Z}\)

- **Problem**: one does not receive pairs in the data stream!

- **Solution**: an online learning model for pairwise loss functions
Online Learning Model for Pairwise Loss Functions

**Learner**

\[ \ell : \mathcal{H} \times \mathcal{Z} \times \mathcal{Z} \rightarrow \mathbb{R} \]

**Adversary**

- At each time \( t \), adversary gives us a **single data point** \( z_t = (x_t, y_t) \)
- Loss \( \ell_t \) on hypothesis \( h_{t-1} \) calculated by pairing \( z_t \) with past points
Online Learning Model for Pairwise Loss Functions

At each time $t$, adversary gives us a single data point $z_t = (x_t, y_t)$

Loss $\ell_t$ on hypothesis $h_{t-1}$ calculated by pairing $z_t$ with past points

Buffer $B$

Pair up with all previous points

Incur loss

$$\hat{\mathcal{L}}^\infty_t(h_{t-1}) = \frac{1}{t-1} (\ell(h_{t-1}, z_t, z_1) + \ell(h_{t-1}, z_t, z_2) + \ldots + \ell(h_{t-1}, z_t, z_{t-1}))$$
Online Learning Model for Pairwise Loss Functions

\[ \ell : \mathcal{H} \times \mathcal{Z} \times \mathcal{Z} \rightarrow \mathbb{R} \]

- At each time \( t \), adversary gives us a **single data point** \( z_t = (x_t, y_t) \)
- Loss \( \ell_t \) on hypothesis \( h_{t-1} \) calculated by pairing \( z_t \) with **some** past points

**Finite Buffer** \( B \)

- Capacity to store \( s \) **data items** at a time
Online Learning Model for Pairwise Loss Functions

- At each time $t$, adversary gives us a **single data point** $z_t = (x_t, y_t)$
- Loss $\ell_t$ on hypothesis $h_{t-1}$ calculated by pairing $z_t$ with (some) past points

**Finite Buffer $B$**

- Can pair up only with buffer points $(z_t, z_{i_1})$, $(z_t, z_{i_2})$, \ldots, $(z_t, z_{i_5})$
- Incur loss

$$\hat{L}^\text{buf}_t(h_{t-1}) = \frac{1}{s} (\ell(h_{t-1}, z_t, z_{i_1}) + \ell(h_{t-1}, z_t, z_{i_2}) + \ldots + \ell(h_{t-1}, z_t, z_{i_s}))$$
Online Learning Model for Pairwise Loss Functions

\[ \ell : \mathcal{H} \times \mathcal{Z} \times \mathcal{Z} \rightarrow \mathbb{R} \]

Regret Bounds in this Model:

- How well are we able to do on all possible pairs

  - All-pairs Regret Bound:
    \[
    \frac{1}{n-1} \sum_{t=1}^{n-1} \hat{L}_t^\infty(h_t) \leq \inf_{h \in \mathcal{H}} \frac{1}{n-1} \sum_{t=2}^{n} \hat{L}_t^\infty(h) + \mathcal{R}_n^\infty
    \]
Online Learning Model for Pairwise Loss Functions

Learner: \[ \ell : \mathcal{H} \times \mathcal{Z} \times \mathcal{Z} \to \mathbb{R} \]

Regret Bounds in this Model:

- How well are we able to do on all possible pairs
  - All-pairs Regret Bound:
    \[ \frac{1}{n-1} \sum_{t=1}^{n-1} \hat{L}_t^\infty(h_t) \leq \inf_{h \in \mathcal{H}} \frac{1}{n-1} \sum_{t=2}^{n} \hat{L}_t^\infty(h) + \mathcal{R}_n^\infty \]

- How well are we able to do on pairs that we have seen
  - Finite-buffer Regret Bound:
    \[ \frac{1}{n-1} \sum_{t=1}^{n-1} \hat{L}_t^{\text{buf}}(h_t) \leq \inf_{h \in \mathcal{H}} \frac{1}{n-1} \sum_{t=2}^{n} \hat{L}_t^{\text{buf}}(h) + \mathcal{R}_n^{\text{buf}} \]
Learning with Pairwise Loss Functions

\[ \ell : \mathcal{H} \times \mathcal{Z} \times \mathcal{Z} \rightarrow \mathbb{R} \]

Offline Learning for Pairwise Loss Functions?

- Online techniques used for several batch applications
  - PEGASOS, LASVM ..
  - Even more important for pairwise loss functions
    - Expensive latency costs in sampling i.i.d. pairs from disk.
Learning with Pairwise Loss Functions

\[ \ell : \mathcal{H} \times \mathbb{Z} \times \mathbb{Z} \rightarrow \mathbb{R} \]

Offline Learning for Pairwise Loss Functions?

- **Problem**: Generalization Bounds for Online Algorithms
  - Online learning process generates hypothesis \( \bar{h} \)
  - Generalization performance \( \mathcal{L}(h) := \mathbb{E}_{z_1, z_2} [\ell(h, z_1, z_2)] \)
  - Wish to bound *excess risk*: \( \mathcal{E}_n = \mathcal{L}(\bar{h}) - \inf_{h \in \mathcal{H}} \mathcal{L}(h) \)

- **Solution**: Online-to-batch conversion bounds
  - Bound \( \mathcal{E}_n \) for learned predictor in terms of in terms of \( \mathcal{R}_{n}^{\text{buf}} \) or \( \mathcal{R}_{n}^{\infty} \)
  - **Problem** (for later): Existing OTB techniques *don’t work* here
Learning with Pairwise Loss Functions

\[ \ell : \mathcal{H} \times \mathcal{Z} \times \mathcal{Z} \rightarrow \mathbb{R} \]

- **Online AUC Maximization**
  [Zhao et al, ICML 2011]
  - Use classical stream sampling algorithm RS
  - All-pairs regret bound needs fixing
  - Finite-buffer regret bound holds (implicit)
Learning with Pairwise Loss Functions

\[ \ell : \mathcal{H} \times \mathcal{Z} \times \mathcal{Z} \rightarrow \mathbb{R} \]

- **Online AUC Maximization**
  [Zhao et al, ICML 2011]
  - Use classical stream sampling algorithm RS
  - All-pairs regret bound needs fixing
  - Finite-buffer regret bound holds (implicit)

- **OLP: Online Learning for PLF**
  [This work]
  - Use a novel stream sampling algorithm RS-x
  - Guaranteed sublinear regret w.r.t all-pairs
  - Finite-buffer regret bound holds
Learning with Pairwise Loss Functions

\[ \ell : \mathcal{H} \times \mathcal{Z} \times \mathcal{Z} \to \mathbb{R} \]

- **OTB conversion Bounds for PLF**
  
  \[ \text{[Wang et al, COLT 2012]} \]

  - Work only w.r.t all-pairs regret bounds
  - Unable to handle
    \[ \text{[Zhao et al, ICML 2011]} \]
  - Bounds depend linearly on **input dimension**
  - Dont handle **sparse learning** formulations
  - Basic rates of convergence
Learning with Pairwise Loss Functions

\[ \ell : \mathcal{H} \times \mathcal{Z} \times \mathcal{Z} \rightarrow \mathbb{R} \]

- **OTB conversion Bounds for PLF**
  - [Wang et al, COLT 2012]
    - Work only w.r.t all-pairs regret bounds
    - Unable to handle
      - [Zhao et al, ICML 2011]
    - Bounds depend linearly on input dimension
    - Don’t handle sparse learning formulations
    - Basic rates of convergence

- **OTB conversion Bounds for PLF**
  - [This work]
    - Work with all-pairs and finite-buffer regret
    - Able to handle
      - [Zhao et al, ICML 2011]
    - Bounds independent of input dimension
    - Handle sparse learning formulations
    - Fast rates for strongly convex pairwise loss functions
Online Learning with Pairwise Loss Functions

Learning Algorithm:

- Hypothesis update
- Buffer update
  - Guarantees

Regret Bounds:

- Finite-buffer regret
- All-pairs regret
Online Learning with Pairwise Loss Functions

\[ \ell : \mathcal{H} \times \mathcal{Z} \times \mathcal{Z} \rightarrow \mathbb{R} \]

**Learning Algorithm:**

- **Hypothesis update**
- **Buffer update**
  - Guarantees

**Regret Bounds:**

- Finite-buffer regret
- All-pairs regret

**OLP : Online Learning for Pairwise Loss Functions**

1. Start off with \( h_0 = 0 \) and empty buffer \( B \)

At each time step \( t = 1 \ldots n \)

2. Receive new training point \( z_t \)

3. Construct loss function \( \ell_t = \hat{\mathcal{L}}_{buf}^t \)

4. \( h_t \leftarrow \Pi_{\Omega} \left[ h_{t-1} - \frac{\eta}{\sqrt{t}} \nabla_{h} \ell_t(h_{t-1}) \right] \)

5. Update buffer \( B \) with \( z_t \)

6. Return \( \bar{h} = \frac{1}{n} \sum_{t=0}^{n-1} h_t \)
Online Learning with Pairwise Loss Functions

\[ \ell : \mathcal{H} \times \mathcal{Z} \times \mathcal{Z} \to \mathbb{R} \]

Learning Algorithm:

- Hypothesis update
- Buffer update
  - Guarantees

RS-x : Reservoir Sampling with Replacement

[ ]

Regret Bounds:

- Finite-buffer regret
- All-pairs regret
Online Learning with Pairwise Loss Functions

**Learning Algorithm:**
- Hypothesis update
- **Buffer update**
  - Guarantees

**Regret Bounds:**
- Finite-buffer regret
- All-pairs regret

\[ \ell : \mathcal{H} \times \mathcal{Z} \times \mathcal{Z} \to \mathbb{R} \]
Online Learning with Pairwise Loss Functions

Learning Algorithm:
- Hypothesis update
- Buffer update
  - Guarantees

RS-x : Reservoir Sampling with Replacement

Regret Bounds:
- Finite-buffer regret
- All-pairs regret
Online Learning with Pairwise Loss Functions

Learning Algorithm:
- Hypothesis update
- Buffer update
  - Guarantees

Regret Bounds:
- Finite-buffer regret
- All-pairs regret

RS-x : Reservoir Sampling with Replacement

\[ \ell : \mathcal{H} \times \mathcal{Z} \times \mathcal{Z} \rightarrow \mathbb{R} \]
Online Learning with Pairwise Loss Functions

Learner $\ell : H \times Z \times Z \rightarrow \mathbb{R}$

Adversary

Learning Algorithm:
- Hypothesis update
- Buffer update
  - Guarantees

RS-x : Reservoir Sampling with Replacement

Regret Bounds:
- Finite-buffer regret
- All-pairs regret
Online Learning with Pairwise Loss Functions

Learning Algorithm:
- Hypothesis update
- Buffer update
  - Guarantees

Regret Bounds:
- Finite-buffer regret
- All-pairs regret

\[ \ell : \mathcal{H} \times \mathcal{Z} \times \mathcal{Z} \rightarrow \mathbb{R} \]

\[ RS-x : \text{Reservoir Sampling with Replacement} \]

\[ [z_0 \ z_1 \ z_2] \]
Online Learning with Pairwise Loss Functions

**Learning Algorithm:**
- Hypothesis update
- **Buffer update**
  - Guarantees

**Regret Bounds:**
- Finite-buffer regret
- All-pairs regret

\[ \ell : \mathcal{H} \times \mathcal{Z} \times \mathcal{Z} \to \mathbb{R} \]
Online Learning with Pairwise Loss Functions

\[ \ell: \mathcal{H} \times \mathcal{Z} \times \mathcal{Z} \rightarrow \mathbb{R} \]

Learning Algorithm:
- Hypothesis update
- Buffer update
  - Guarantees

Regret Bounds:
- Finite-buffer regret
- All-pairs regret
Online Learning with Pairwise Loss Functions

Learning Algorithm:
- Hypothesis update
- Buffer update
  - Guarantees

Regret Bounds:
- Finite-buffer regret
- All-pairs regret

\[
\ell : \mathcal{H} \times \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}
\]

RS-x : Reservoir Sampling with Replacement

\[
\sim B(1/t)
\]
Online Learning with Pairwise Loss Functions

Learning Algorithm:
- Hypothesis update
- Buffer update
  - Guarantees

Regret Bounds:
- Finite-buffer regret
- All-pairs regret

RS-x : Reservoir Sampling with Replacement

\( \ell : H \times Z \times Z \rightarrow R \)

Our Contributions
Online Learning with Pairwise Loss Functions

Learning Algorithm:

- Hypothesis update
- Buffer update
  - Guarantees

Regret Bounds:

- Finite-buffer regret
- All-pairs regret

\[ \ell : \mathcal{H} \times \mathbb{Z} \times \mathbb{Z} \rightarrow \mathbb{R} \]

RS-x : Reservoir Sampling with Replacement

\[
\begin{bmatrix}
Z_{i_0} & Z_t & Z_{i_2} & Z_{i_3} & Z_t & Z_{i_5}
\end{bmatrix}
\]
Online Learning with Pairwise Loss Functions

**Learner**

\[ \ell : \mathcal{H} \times \mathcal{Z} \times \mathcal{Z} \rightarrow \mathbb{R} \]

**Adversary**

**Learning Algorithm:**
- Hypothesis update
- Buffer update
  - Guarantees

**Regret Bounds:**
- Finite-buffer regret
- All-pairs regret

**RS-x : Reservoir Sampling with Replacement**

**Sampling Guarantee for RS-x:**

**Theorem:** At any fixed time \( t > s \), every buffer element is an i.i.d. sample from the set \( \{z_1, \ldots, z_{t-1}\} \)
Online Learning with Pairwise Loss Functions

Learner $\ell : \mathcal{H} \times \mathcal{Z} \times \mathcal{Z} \rightarrow \mathbb{R}$

Adversary

Learning Algorithm:
- Hypothesis update
- Buffer update
  - Guarantees

Regret Bounds:
- Finite-buffer regret
- All-pairs regret

Finite-buffer regret bound for OLP

How well are we able to do on pairs that we have seen

Theorem: $\mathcal{R}_n^{buf} \leq \frac{1}{\sqrt{n}}$

Proof: OLP is a GIGA variant: the analysis follows.
Online Learning with Pairwise Loss Functions

Learner

\( \ell : \mathcal{H} \times \mathcal{Z} \times \mathcal{Z} \rightarrow \mathbb{R} \)

Adversary

Learning Algorithm:

- Hypothesis update
- Buffer update
  - Guarantees

Regret Bounds:

- Finite-buffer regret
- All-pairs regret

All-pairs regret bound for OLP

How well are we able to do on all pairs

**Theorem:** \( \mathcal{R}_n^\infty \leq C_d \sqrt{\frac{\log n}{s}} \) w.h.p.

**Proof:** Use properties of RS-x to show that w.h.p.

\[ \hat{L}_t^{\text{buf}} - \epsilon \leq \hat{L}_t^\infty \leq \hat{L}_t^{\text{buf}} + \epsilon \]

Use regret bound on \( \mathcal{R}_n^{\text{buf}} \) to finish off.
Generalization Bounds for Online Algorithms for Pairwise Loss Functions

Generalization Bounds for Pairwise Loss Functions

• Recall: Online learning process generates hypothesis \( \overline{h} = \frac{1}{n} \sum_{t=0}^{n-1} h_t \)
  
  ○ Wish to bound excess risk: \( \mathcal{E}_n = \mathcal{L}(\overline{h}) - \inf_{h \in \mathcal{H}} \mathcal{L}(h) \)

  • **Online-to-batch conversion**: bound \( \mathcal{E}_n \) in terms of \( \mathcal{R}_n^{\text{buf}} \) (or \( \mathcal{R}_n^{\infty} \))
Generalization Bounds for Online Algorithms for Pairwise Loss Functions

Generalization Bounds for Pairwise Loss Functions

- Recall: Online learning process generates hypothesis $\bar{h} = \frac{1}{n} \sum_{t=0}^{n-1} h_t$
  - Wish to bound excess risk: $\mathcal{E}_n = \mathcal{L}(\bar{h}) - \inf_{h \in \mathcal{H}} \mathcal{L}(h)$

- **Online-to-batch conversion**: bound $\mathcal{E}_n$ in terms of $\mathcal{R}_n^{buf}$ (or $\mathcal{R}_n^\infty$)

- **Classical Proof Techniques**: for pointwise loss functions
  - $\{\ell_t(h_{t-1}) - \mathcal{L}(h_{t-1})\}$ forms an MDS
  - [Cesa-Bianchi et al, NIPS 2001], Azuma-Heoffding
  - [Kakade and Tewari, NIPS 2008], Bernstein
Generalization Bounds for Online Algorithms for Pairwise Loss Functions

Generalization Bounds for Pairwise Loss Functions

- **Problem**: Existing techniques do not apply
  - \( \{\ell_t(h_{t-1}) - L(h_{t-1})\} \) not an MDS due to **coupling**

- **Solution**: decompose \( \{\ell_t(h_{t-1}) - L(h_{t-1})\} \) into MDS and residual terms
  - First proposed by [Wang et al, COLT 2012]
  - Apply Azuma-Hoeffding to one and Uniform Convergence to other
  - We use Rademacher average route: great **flexibility** and **tight** bounds
Generalization Bounds for Online Algorithms for Pairwise Loss Functions

Generalization Bounds for Pairwise Loss Functions

- **Problem**: Existing techniques do not apply
  - \( \{\ell_t(h_{t-1}) - \mathcal{L}(h_{t-1})\} \) not an MDS due to **coupling**

- **Solution**: decompose \( \{\ell_t(h_{t-1}) - \mathcal{L}(h_{t-1})\} \) into MDS and residual terms
  - First proposed by [Wang et al, COLT 2012]
  - Apply Azuma-Hoeffding to one and Uniform Convergence to other
  - We use Rademacher average route: great **flexibility** and **tight** bounds

- **Problem**: Coupling yet again prevents classical symmetrization

- **Solution**: Symmetrization of Expectations!
Generalization Bounds for Online Algorithms for Pairwise Loss Functions

Generalization Bounds for Pairwise Loss Functions

- **Problem**: What should be notion of Rademacher averages?

- **Solution**: We define

\[ R_n(H) := \mathbb{E}_{z,z_\tau,\epsilon_\tau} \left[ \sup_{h \in H} \frac{1}{n} \sum_{\tau=1}^{n} \epsilon_\tau h(z, z_\tau) \right] \]

  - One head term and \( n \) tail terms
  - We show that for several problems, the R.A. have the following form

\[ R_n(H) \sim C_d \cdot \frac{1}{\sqrt{n}} \]

- Derivations do not follow directly from existing techniques
Generalization Bounds for Online Algorithms for Pairwise Loss Functions

Our Online-to-batch Conversion Bounds

\[ \mathcal{L}(\overline{h}) \leq \inf_{h \in \mathcal{H}} \mathcal{L}(h) + \mathcal{E}_n \]

- **Bounded Losses**
  - All-pairs regret bounds, w.h.p. \[ \mathcal{E}_n \leq \mathcal{R}_n^\infty + \frac{C_d + \sqrt{\log n}}{\sqrt{n}} \]
  - Finite-buffer regret bounds, w.h.p. \[ \mathcal{E}_n \leq \mathcal{R}_n^{\text{buf}} + \frac{C_d + \sqrt{\log n}}{\sqrt{s}} \]
  - **Proofs**: Uniform convergence with SoE + Azuma-Hoeffding inequality
Generalization Bounds for Online Algorithms for Pairwise Loss Functions

Our Online-to-batch Conversion Bounds

\[ \mathcal{L}(\bar{h}) \leq \inf_{h \in \mathcal{H}} \mathcal{L}(h) + \mathcal{E}_n \]

- **Strongly Convex Losses**
  - All-pairs regret bounds, w.h.p. \( \mathcal{E}_n \leq \mathcal{R}_n^\infty + \frac{C_d^2 \log^2 n}{n} \)
  - Finite-buffer regret bounds, w.h.p. \( \mathcal{E}_n \leq \mathcal{R}_n^{buf} + \frac{C_d^2 \log n}{s} \)
  - **Proofs**: Novel use of fast rate results for batch algorithms + Bernstein-type martingale inequalities
Applications

\[
\mathcal{R}_n^\infty \leq C_d \sqrt{\frac{\log n}{s}}, \quad \mathcal{E}_n \leq \mathcal{R}_n^\infty + \frac{C_d^2 \log^2 n}{n}
\]

Bipartite Ranking

- **Objective**: \( h : x \mapsto \langle w, x \rangle \) such that \( h(x_1) > h(x_2) \) if \( y_1 = 1, y_2 = -1 \)
- Equivalent to maximizing the area under the ROC curve
- Loss function: \( \ell(w, z_1, z_2) = \phi((y_1 - y_2)w^\top (x_1 - x_2)) \)
- **Rademacher Averages**:
  - \( L_p \) regularized \( w \), \( p > 1 \): \( C_d = \mathcal{O}(1) \)
  - \( L_1 \) regularized **sparse** \( w \): \( C_d = \mathcal{O}(\sqrt{\log d}) \)
Applications

\[ \mathcal{K}_n^\infty \leq C_d \sqrt{\frac{\log n}{s}}, \quad \mathcal{E}_n \leq \mathcal{K}_n^\infty + \frac{C_d^2 \log^2 n}{n} \]

Mahalanobis Metric Learning

- **Objective**: \( d^2 : (x_1, x_2) \mapsto (x_1 - x_2)^\top M (x_1 - x_2) \) such that
  - \( d^2(x_1, x_2) > 1 \) if \( y_1 \neq y_2 \)
  - \( d^2(x_1, x_2) < 1 \) if \( y_1 = y_2 \)

- **Loss function**: \( \ell(M, z_1, z_2) = \phi (y_1 y_2 (1 - d_M^2(x_1, x_2))) \)

- **Rademacher Averages**:
  - Frobenius norm regularized \( M \): \( C_d = \mathcal{O}(1) \)
  - Trace norm regularized \( M \): \( C_d = \mathcal{O}(\sqrt{\log d}) \)
Applications

\[
\mathcal{R}_n^\infty \leq C_d \sqrt{\log n}, \quad \mathcal{E}_n \leq \mathcal{R}_n^\infty + \frac{C_d^2 \log^2 n}{n}
\]

Two-stage Multiple Kernel Learning

- **Objective**: \( K : (x_1, x_2) \mapsto K_\mu(x_1, x_2) \) such that \( K_\mu = \sum_{i=1}^p \mu_i K_i \)
- Desire *kernel-target alignment*
- Loss function: \( \ell(\mu, z_1, z_2) = \phi(y_1 y_2 K_\mu(x_1, x_2)) \)
- **Rademacher Averages**:
  - \( L_2 \) norm regularized \( \mu \): \( C_d = O(\sqrt{p}) \)
  - \( L_1 \) norm regularized \( \mu \): \( C_d = O(\sqrt{\log p}) \)
Future Work

1. Our all-pairs regret bound for $\text{OLP} + \text{RS-x}$ is $\sqrt{\frac{\log n}{s}}$
   - Is $\omega(\log n)$ buffer size necessary for sublinear regret?

2. Our OTB results for finite-buffer regret bounds behave as $\sqrt{\frac{\log n}{s}}$ (resp. $\frac{\log n}{s}$)
   - Can we get $O\left(\frac{1}{f(n)}\right)$ rates?

3. Our generalization bounds require buffer update policies to be stream oblivious
   - Update algorithm cannot look at $z_t$, just the index $t$
   - **Examples**: FIFO/LRU, RS, RS-x ..
   - Guarantees for (suitable) *stream aware* policies?
Thank You!

For more, visit our poster this evening !!!