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 ..

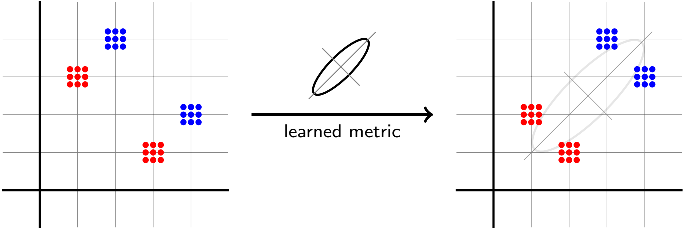
$$\boxed{\ell : \mathcal{H} \times \mathcal{Z} \rightarrow \mathbb{R}}$$

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

Examples:

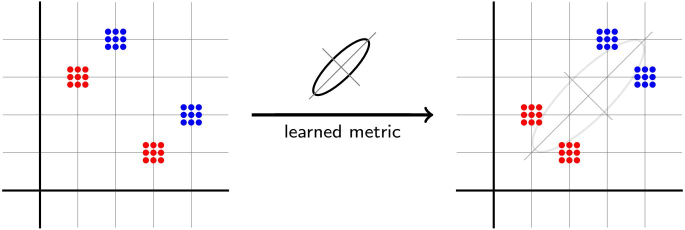
- Hinge loss: $\ell(h, \mathbf{z}) = [1 - y \cdot h(\mathbf{x})]_+$
- ϵ-insensitive loss: $\ell(h, \mathbf{z}) = [|y - h(\mathbf{x})| - \epsilon]_+$
- Logistic loss: $\ell(h, \mathbf{z}) = \ln(1 + \exp(y \cdot h(\mathbf{x})))$

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**: $\ell(d_{\mathbf{M}}, \mathbf{z}_1, \mathbf{z}_2) = \phi\left(y_1 y_2 \left(1 - d^2_{\mathbf{M}}(\mathbf{x}_1, \mathbf{x}_2)\right)\right)$

  where ϕ is the hinge loss function

Learning with Pairwise Loss Functions

$$\boxed{\ell : \mathcal{H} \times \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}}$$

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

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

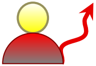
Online Learning for Pairwise Loss Functions ?

- **Algorithmic Challenges**
  - Attempts to reduce to pointwise learning
  - Treat pairs $(\mathbf{z}_i, \mathbf{z}_j)$ 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

$$\boxed{\ell : \mathcal{H} \times \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}}$$



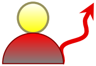
Adversary

- At each time t, adversary gives us a **single data point** $\boxed{\mathbf{z}_t} = (\mathbf{x}_t, y_t)$
- Loss ℓ_t on hypothesis h_{t-1} calculated by pairing $\mathbf{z}_t$ with past points

Online Learning Model for Pairwise Loss Functions

Learner

$$\boxed{\ell : \mathcal{H} \times \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}}$$



Adversary

- At each time t, adversary gives us a **single data point** $\boxed{\mathbf{z}_t} = (\mathbf{x}_t, y_t)$

- Loss ℓ_t on hypothesis h_{t-1} calculated by pairing $\mathbf{z}_t$ with past points

Buffer B $\quad \left[\ \boxed{\mathbf{z}_0}\ \boxed{\mathbf{z}_1}\ \boxed{\mathbf{z}_2}\ \boxed{\mathbf{z}_3}\ \cdots\ \cdots\ \right]$

- Pair up with **all** previous points $\quad (\boxed{\mathbf{z}_t}, \boxed{\mathbf{z}_1})\ (\boxed{\mathbf{z}_t}, \boxed{\mathbf{z}_2}) \cdots (\boxed{\mathbf{z}_t}, \boxed{\mathbf{z}_{t-1}})$

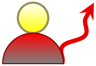
- Incur loss

$$\hat{\mathcal{L}}_t^{\infty}(h_{t-1}) = \frac{1}{t-1}\left(\ell(h_{t-1}, \mathbf{z}_t, \mathbf{z}_1) + \ell(h_{t-1}, \mathbf{z}_t, \mathbf{z}_2) + \ldots + \ell(h_{t-1}, \mathbf{z}_t, \mathbf{z}_{t-1})\right)$$

Online Learning Model for Pairwise Loss Functions

Learner

$$\boxed{\ell : \mathcal{H} \times \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}}$$



Adversary

- At each time t, adversary gives us a **single data point** $\boxed{\mathbf{z}_t} = (\mathbf{x}_t, y_t)$
- Loss ℓ_t on hypothesis h_{t-1} calculated by pairing $\mathbf{z}_t$ with **(some)** past points

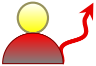
Finite Buffer B

- Capacity to store s **data items** at a time

Online Learning Model for Pairwise Loss Functions

Learner

$$\boxed{\ell : \mathcal{H} \times \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}}$$



Adversary

- At each time t, adversary gives us a **single data point** $\boxed{\mathbf{z}_t} = (\mathbf{x}_t, y_t)$

- Loss ℓ_t on hypothesis h_{t-1} calculated by pairing $\mathbf{z}_t$ with **(some)** past points

Finite Buffer B $\left[\ \boxed{\mathbf{z}_{i_0}}\ \boxed{\mathbf{z}_{i_1}}\ \boxed{\mathbf{z}_{i_2}}\ \boxed{\mathbf{z}_{i_3}}\ \boxed{\mathbf{z}_{i_4}}\ \boxed{\mathbf{z}_{i_5}}\ \right]$

- Can pair up only with buffer points $(\boxed{\mathbf{z}_t}, \boxed{\mathbf{z}_{i_1}})\ (\boxed{\mathbf{z}_t}, \boxed{\mathbf{z}_{i_2}}) \cdots (\boxed{\mathbf{z}_t}, \boxed{\mathbf{z}_{i_5}})$

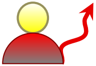
- Incur loss

$$\hat{\mathcal{L}}_t^{\text{buf}}(h_{t-1}) = \frac{1}{s}\left(\ell(h_{t-1}, \mathbf{z}_t, \mathbf{z}_{i_1}) + \ell(h_{t-1}, \mathbf{z}_t, \mathbf{z}_{i_2}) + \ldots + \ell(h_{t-1}, \mathbf{z}_t, \mathbf{z}_{i_s})\right)$$

Online Learning Model for Pairwise Loss Functions


Learner

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


Adversary

Regret Bounds in this Model:

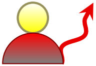
- How well are we able to do on **all possible pairs**
  - **All-pairs Regret Bound**:

$$\frac{1}{n-1}\sum_{t=1}^{n-1} \hat{\mathcal{L}}_t^{\infty}(h_t) \leq \inf_{h \in \mathcal{H}} \frac{1}{n-1}\sum_{t=2}^{n} \hat{\mathcal{L}}_t^{\infty}(h) + \mathfrak{R}_n^{\infty}$$

Online Learning Model for Pairwise Loss Functions

Learner

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



Adversary

Regret Bounds in this Model:

- How well are we able to do on **all possible pairs**
  - **All-pairs Regret Bound**:
  $$\frac{1}{n-1}\sum_{t=1}^{n-1} \hat{\mathcal{L}}_t^{\infty}(h_t) \leq \inf_{h\in\mathcal{H}} \frac{1}{n-1}\sum_{t=2}^{n} \hat{\mathcal{L}}_t^{\infty}(h) + \mathfrak{R}_n^{\infty}$$
- How well are we able to do on **pairs that we have seen**
  - **Finite-buffer Regret Bound**:
  $$\frac{1}{n-1}\sum_{t=1}^{n-1} \hat{\mathcal{L}}_t^{\text{buf}}(h_t) \leq \inf_{h\in\mathcal{H}} \frac{1}{n-1}\sum_{t=2}^{n} \hat{\mathcal{L}}_t^{\text{buf}}(h) + \mathfrak{R}_n^{\text{buf}}$$

Learning with Pairwise Loss Functions

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

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

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

Offline Learning for Pairwise Loss Functions ?

- **Problem**: Generalization Bounds for Online Algorithms
  - Online learning process generates hypothesis $\bar{h}$
  - Generalization performance $\mathcal{L}(h) := \mathop{\mathbb{E}}_{\mathbf{z}_1, \mathbf{z}_2} [\![\ell(h, \mathbf{z}_1, \mathbf{z}_2)]\!]$
  - 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 $\mathfrak{R}_n^{\text{buf}}$ or $\mathfrak{R}_n^{\infty}$
  - **Problem** (for later): Existing OTB techniques **dont work** here

Learning with Pairwise Loss Functions

$$\boxed{\ell : \mathcal{H} \times \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}}$$

- **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

$$\boxed{\ell : \mathcal{H} \times \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}}$$

- **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

$$\boxed{\ell : \mathcal{H} \times \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}}$$

- **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**
  - Dont handle **sparse learning** formulations
  - Basic rates of convergence

Learning with Pairwise Loss Functions

$$\boxed{\ell : \mathcal{H} \times \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}}$$

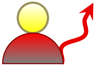
- **OTB conversion Bounds for PLF** [*Wang et al, COLT 2012*]
  - Work only w.r.t all-pairs regret bounds
  - Unable to handle [*Zhao et al, ICML 2011*]
  - Bounds depend linearly on **input dimension**
  - Dont handle **sparse learning** formulations
  - Basic rates of convergence

- **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

Learner

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



Adversary

Learning Algorithm:

- Hypothesis update
- Buffer update
  - Guarantees

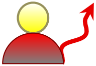
Regret Bounds:

- Finite-buffer regret
- All-pairs regret

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

OLP : **O**nline **L**earning for **P**airwise Loss Functions

1. Start off with $h_0 = \mathbf{0}$ and empty buffer B

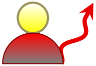
   At each time step $t = 1 \ldots n$

2. Receive new training point $\mathbf{z}_t$
3. Construct loss function $\ell_t = \hat{\mathcal{L}}_t^{\text{buf}}$
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 $\mathbf{z}_t$
6. Return $\bar{h} = \frac{1}{n} \sum_{t=0}^{n-1} h_t$

Online Learning with Pairwise Loss Functions

Learner

$$\boxed{\ell : \mathcal{H} \times \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}}$$

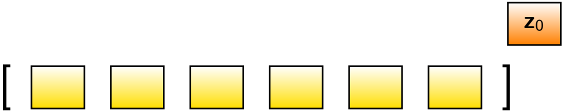


Adversary

Learning Algorithm:

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

RS-x : **R**eservoir **S**ampling with Repla**x**ement



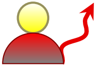
Regret Bounds:

- Finite-buffer regret
- All-pairs regret

Online Learning with Pairwise Loss Functions

Learner

$$\boxed{\ell : \mathcal{H} \times \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}}$$



Adversary

Learning Algorithm:

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

RS-x : **R**eservoir **S**ampling with Repla**x**ement

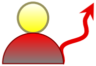
Regret Bounds:

- Finite-buffer regret
- All-pairs regret

Online Learning with Pairwise Loss Functions

Learner

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

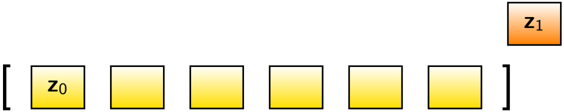


Adversary

Learning Algorithm:

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

RS-x : **R**eservoir **S**ampling with Repla**x**ement



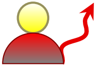
Regret Bounds:

- Finite-buffer regret
- All-pairs regret

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

RS-x : **R**eservoir **S**ampling with Repla**x**ement

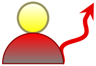
Regret Bounds:

- Finite-buffer regret
- All-pairs regret

Online Learning with Pairwise Loss Functions

Learner

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

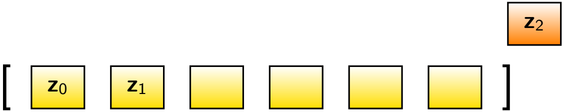


Adversary

Learning Algorithm:

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

RS-x : **R**eservoir **S**ampling with Repla**x**ement



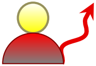
Regret Bounds:

- Finite-buffer regret
- All-pairs regret

Online Learning with Pairwise Loss Functions

Learner

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



Adversary

Learning Algorithm:

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

RS-x : **R**eservoir **S**ampling with Repla**x**ement

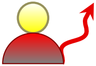
Regret Bounds:

- Finite-buffer regret
- All-pairs regret

Online Learning with Pairwise Loss Functions


Learner

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


Adversary

Learning Algorithm:

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

RS-x : **R**eservoir **S**ampling with Repla**x**ement

...

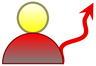
Regret Bounds:

- Finite-buffer regret
- All-pairs regret

Online Learning with Pairwise Loss Functions

Learner

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



Adversary

Learning Algorithm:

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

Regret Bounds:

- Finite-buffer regret
- All-pairs regret

RS-x : **R**eservoir **S**ampling with Repla**x**ement

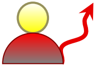
$$\mathbf{z}_t$$

$$\left[\; \mathbf{z}_{i_0} \quad \mathbf{z}_{i_1} \quad \mathbf{z}_{i_2} \quad \mathbf{z}_{i_3} \quad \mathbf{z}_{i_4} \quad \mathbf{z}_{i_5} \;\right]$$

Online Learning with Pairwise Loss Functions


Learner

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


Adversary

Learning Algorithm:

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

RS-x : **R**eservoir **S**ampling with Repla**x**ement

 $\sim B(1/t)$

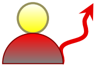
Regret Bounds:

- Finite-buffer regret
- All-pairs regret

Online Learning with Pairwise Loss Functions

Learner

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



Adversary

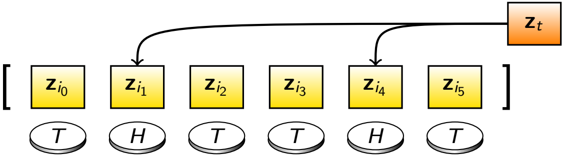
Learning Algorithm:

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

Regret Bounds:

- Finite-buffer regret
- All-pairs regret

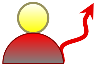
RS-x : **R**eservoir **S**ampling with Repla**x**ement



Online Learning with Pairwise Loss Functions

Learner

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



Adversary

Learning Algorithm:

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

Regret Bounds:

- Finite-buffer regret
- All-pairs regret

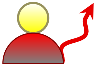
RS-x : **R**eservoir **S**ampling with Repla**x**ement

$$\left[\ \boxed{\mathbf{z}_{i_0}}\ \boxed{\mathbf{z}_{t}}\ \boxed{\mathbf{z}_{i_2}}\ \boxed{\mathbf{z}_{i_3}}\ \boxed{\mathbf{z}_{t}}\ \boxed{\mathbf{z}_{i_5}}\ \right]$$

Online Learning with Pairwise Loss Functions

Learner

$$\boxed{\ell : \mathcal{H} \times \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}}$$



Adversary

Learning Algorithm:

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

Regret Bounds:

- Finite-buffer regret
- All-pairs regret

RS-x : **R**eservoir **S**ampling with Repla**x**ement

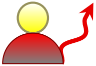
Sampling Guarantee for RS-x :

Theorem: At any fixed time $t > s$, every buffer element is an **i.i.d. sample** from the set $\{\mathbf{z}_1, \ldots, \mathbf{z}_{t-1}\}$

Online Learning with Pairwise Loss Functions

Learner

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



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

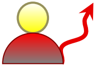
Theorem: $\mathfrak{R}_n^{\text{buf}} \leq \dfrac{1}{\sqrt{n}}$

Proof: **OLP** is a GIGA variant: the analysis follows.

Online Learning with Pairwise Loss Functions


Learner

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


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: $\mathfrak{R}_n^{\infty} \leq \mathbf{C_d}\sqrt{\frac{\log n}{s}}$ w.h.p.

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

$$\hat{\mathcal{L}}_t^{\text{buf}} - \epsilon \leq \hat{\mathcal{L}}_t^{\infty} \leq \hat{\mathcal{L}}_t^{\text{buf}} + \epsilon$$

Use regret bound on $\mathfrak{R}_n^{\text{buf}}$ to finish off.

Generalization Bounds for Online Algorithms for Pairwise Loss Functions

Generalization Bounds for Pairwise Loss Functions

- Recall: Online learning process generates hypothesis $\bar{h} = \frac{1}{n}\sum_{t=0}^{n-1} h_t$
  - 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 $\mathfrak{R}_n^{\text{buf}}$ (or $\mathfrak{R}_n^{\infty}$)

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 $\mathfrak{R}_n^{\text{buf}}$ (or $\mathfrak{R}_n^{\infty}$)
- **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
  - $\{\ell_t(h_{t-1}) - \mathcal{L}(h_{t-1})\}$ 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

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

$$\mathcal{R}_n(\mathcal{H}) := \mathop{\mathbb{E}}_{\mathbf{z},\mathbf{z}_\tau,\epsilon_\tau} \left[\!\!\left[\sup_{h \in \mathcal{H}} \frac{1}{n} \sum_{\tau=1}^{n} \epsilon_\tau h(\mathbf{z}, \mathbf{z}_\tau) \right]\!\!\right]$$

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

$$\mathcal{R}_n(\mathcal{H}) \sim \mathbf{C_d} \cdot \frac{1}{\sqrt{n}}$$

- Derivations **do not** follow directly from existing techniques

Generalization Bounds for Online Algorithms for Pairwise Loss Functions

Our Online-to-batch Conversion Bounds

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

- **Bounded Losses**
  - All-pairs regret bounds, w.h.p. $\mathcal{E}_n \leq \mathfrak{R}_n^{\infty} + \frac{\mathbf{C_d} + \sqrt{\log n}}{\sqrt{n}}$
  - Finite-buffer regret bounds, w.h.p. $\mathcal{E}_n \leq \mathfrak{R}_n^{\text{buf}} + \frac{\mathbf{C_d} + \sqrt{\log n}}{\sqrt{s}}$
  - **Proofs**: Uniform convergence with SoE + Azuma-Hoeffding inequality

Generalization Bounds for Online Algorithms for Pairwise Loss Functions

Our Online-to-batch Conversion Bounds

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

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

Applications

$$\mathfrak{R}_n^\infty \leq \mathbf{C_d}\sqrt{\frac{\log n}{s}}, \qquad \mathcal{E}_n \leq \mathfrak{R}_n^\infty + \frac{\mathbf{C_d^2}\log^2 n}{n}$$

Bipartite Ranking

- **Objective**: $h : \mathbf{x} \mapsto \langle \mathbf{w}, \mathbf{x} \rangle$ such that $h(\mathbf{x}_1) > h(\mathbf{x}_2)$ if $y_1 = 1, y_2 = -1$
- Equivalent to maximizing the area under the ROC curve
- Loss function: $\ell(\mathbf{w}, \mathbf{z}_1, \mathbf{z}_2) = \phi\left((y_1 - y_2)\mathbf{w}^\top(\mathbf{x}_1 - \mathbf{x}_2)\right)$
- **Rademacher Averages**:
  - L_p regularized $\mathbf{w}$, $p > 1$: $\mathbf{C_d} = \mathcal{O}(\mathbf{1})$
  - L_1 regularized **sparse** $\mathbf{w}$: $\mathbf{C_d} = \mathcal{O}\left(\sqrt{\log \mathbf{d}}\right)$

Applications

$$\mathfrak{R}_n^\infty \leq \mathbf{C_d}\sqrt{\frac{\log n}{s}}, \qquad \mathcal{E}_n \leq \mathfrak{R}_n^\infty + \frac{\mathbf{C_d^2}\log^2 n}{n}$$

Mahalanobis Metric Learning

- **Objective**: $d^2 : (\mathbf{x}_1, \mathbf{x}_2) \mapsto (\mathbf{x}_1 - \mathbf{x}_2)^\top \mathbf{M}(\mathbf{x}_1 - \mathbf{x}_2)$ such that
  - $d^2(\mathbf{x}_1, \mathbf{x}_2) > 1$ if $y_1 \neq y_2$
  - $d^2(\mathbf{x}_1, \mathbf{x}_2) < 1$ if $y_1 = y_2$
- Loss function: $\ell(\mathbf{M}, \mathbf{z}_1, \mathbf{z}_2) = \phi\left(y_1 y_2 \left(1 - d^2_{\mathbf{M}}(\mathbf{x}_1, \mathbf{x}_2)\right)\right)$
- **Rademacher Averages**:
  - Frobenius norm regularized $\mathbf{M}$: $\mathbf{C_d} = \mathcal{O}\left(\mathbf{1}\right)$
  - Trace norm regularized $\mathbf{M}$: $\mathbf{C_d} = \mathcal{O}\left(\sqrt{\log \mathbf{d}}\right)$

Applications

$$\mathfrak{R}_n^\infty \leq \mathbf{C_d}\sqrt{\frac{\log n}{s}}, \qquad \mathcal{E}_n \leq \mathfrak{R}_n^\infty + \frac{\mathbf{C_d^2}\log^2 n}{n}$$

Two-stage Multiple Kernel Learning

- **Objective**: $K : (\mathbf{x}_1, \mathbf{x}_2) \mapsto K_{\boldsymbol{\mu}}(\mathbf{x}_1, \mathbf{x}_2)$ such that $K_{\boldsymbol{\mu}} = \sum_{i=1}^p \boldsymbol{\mu}_i K_i$
- Desire *kernel-target alignment*
- Loss function: $\ell(\boldsymbol{\mu}, \mathbf{z}_1, \mathbf{z}_2) = \phi\left(y_1 y_2 K_{\boldsymbol{\mu}}(\mathbf{x}_1, \mathbf{x}_2)\right)$
- **Rademacher Averages**:
  - L_2 norm regularized $\boldsymbol{\mu}$: $\mathbf{C_d} = \mathcal{O}\left(\sqrt{\mathbf{p}}\right)$
  - L_1 norm regularized $\boldsymbol{\mu}$: $\mathbf{C_d} = \mathcal{O}\left(\sqrt{\log \mathbf{p}}\right)$

Future Work

1. Our all-pairs regret bound for **OLP** + **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 $\mathcal{O}\left(\frac{1}{f(n)}\right)$ rates ?
3. Our generalization bounds require buffer update policies to be stream oblivious
   - Update algorithm cannot look at $\mathbf{z}_t$, 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 !!!