Learning with Pairwise Losses

Problems, Algorithms and Analysis

Purushottam Kar

Microsoft Research India

Outline

- Part I: Introduction to pairwise loss functions
  - Example applications
- Part II: Batch learning with pairwise loss functions
  - Learning formulation: no algorithmic details
  - Generalization bounds
    - The **coupling** phenomenon
    - Decoupling techniques
- Part III: Online learning with pairwise loss functions
  - A generic online algorithm
    - Regret analysis
  - Online-to-batch conversion bounds
    - A decoupling technique for online-to-batch conversions

Part I: Introduction

What is a loss function?

$$\boxed{\ell : \mathcal{H} \to \mathbb{R}^+}$$

- We observe empirical losses on data $S = \{x_1, \ldots x_n\}$

$$\ell_{x_i}(\cdot) = \ell(h, x_i)$$

- … and try to minimize them (e.g. classfn, regression)

$$\hat{h} = \inf_{h \in \mathcal{H}} \hat{\mathcal{L}}_S(h), \qquad \hat{\mathcal{L}}_S(h) = \frac{1}{n} \sum \ell_{x_i}(h)$$

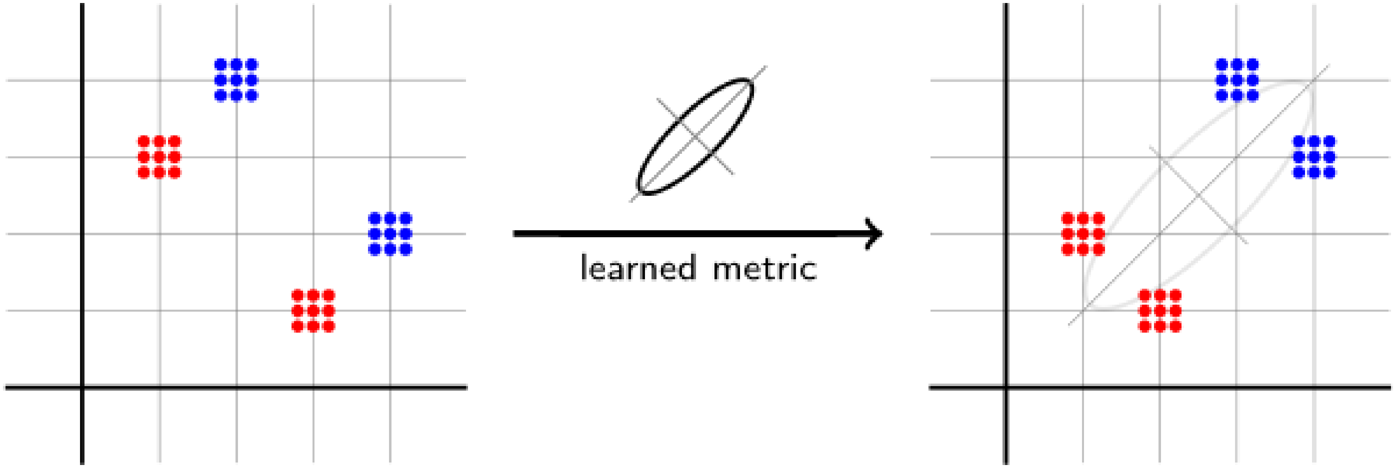
- … in the hope that

$$\left\| {}^{1}/_{n} \sum \ell_{x_i}(\cdot) - \mathbb{E}\ell_x(\cdot) \right\|_\infty \le \epsilon$$

- … so that

$$\mathcal{L}\left(\hat{h}\right) \le \mathcal{L}(h^*) + \epsilon, \qquad \mathcal{L}(h) = \mathbb{E}\ell_x(h)$$

Metric Learning



- Penalize metric for bringing **blue** and **red** points close
- Loss function needs to consider two points at a time!
  - … in other words a **pairwise loss function**
- E.g. $\ell_{(x_1,x_2)}(M) = \begin{cases} 1, y_1 \neq y_2 \text{ and } M(x_1,x_2) < \gamma_1 \\ 1, y_1 = y_2 \text{ and } M(x_1,x_2) > \gamma_2 \\ 0, \text{otherwise} \end{cases}$

Pairwise Loss Functions

- Typically, loss functions are based on **ground truth**
$$\ell_x(h) = \ell\big(h(x), y(x)\big)$$
- Thus, for metric learning, loss functions look like
$$\ell_{(x_1,x_2)}(h) = \ell\big(h(x_1, x_2), y(x_1, x_2)\big)$$
- In previous example, we had
$$h(x_1, x_2) = M(x_1, x_2) \text{ and } y(x_1, x_2) = y_1 y_2$$

- Useful to learn patterns that capture data interactions

Pairwise Loss Functions

Examples: (ϕ is any margin loss function e.g. hinge loss)

- Metric learning [Jin *et al* NIPS '09]

$$\ell_{(x_1,x_2)}(M) = \phi\Big(y_1 y_2\big(1 - M(x_1, x_2)\big)\Big)$$

- Preference learning [Xing *et al* NIPS '02]
- S-goodness [Balcan-Blum ICML '06]

$$\ell_{(x_1,x_2)}(K) = \phi\big(y_1 y_2 K(x_1, x_2)\big)$$

- Kernel-target alignment [Cortes *et al* ICML '10]
- Bipartite ranking, (p)AUC [Narasimhan-Agarwal ICML '13]

$$\ell_{(x_1,x_2)}(f) = \phi\Big(\big(f(x_1) - f(x_2)\big)(y_1 - y_2)\Big)$$

Learning Objectives in Pairwise Learning

- Given training data $x_1, x_2, \dots x_n$
- Learn $\hat{h}: \mathcal{X} \times \mathcal{X} \to \mathcal{Y}$ such that

$$\mathcal{L}(\hat{h}) \leq \mathcal{L}(h^*) + \epsilon$$

(will define $\mathcal{L}(\cdot)$ and $\hat{\mathcal{L}}(\cdot)$ shortly)

Challenges:

- Training data given as singletons, not pairs
- Algorithmic efficiency
- Generalization error bounds

Part II: Batch Learning

Part II: Batch Learning

Batch Learning for Unary Losses

Training with Unary Loss Functions

- Notion of **empirical loss**

$$\hat{\mathcal{L}}: \mathcal{H} \to \mathbb{R}^+$$

- Given training data $S = \{x_1, \ldots, x_n\}$, natural notion

$$\hat{\mathcal{L}}_S(\cdot) = \frac{1}{n} \sum \ell(\cdot, x_i)$$

- Empirical risk minimization dictates us to find $\hat{h}$, s.t.

$$\hat{\mathcal{L}}_S(\hat{h}) \leq \inf_{h \in \mathcal{H}} \hat{\mathcal{L}}_S(h)$$

- Note that $\hat{\mathcal{L}}(\cdot)$ is a U-statistic
- **U-statistic**: a notion of "training loss" $\hat{\mathcal{L}}_S: \mathcal{H} \to \mathbb{R}^+$ s.t.

$$\forall h \in \mathcal{H}, \mathbb{E}\left(\hat{\mathcal{L}}_S(h)\right) = \mathcal{L}(h)$$

Generalization bounds for Unary Loss Functions

- **Step 1**: Bound excess risk by suprēmus excess risk

$$\mathcal{L}(\hat{h}) - \hat{\mathcal{L}}_S(\hat{h}) \leq \sup_{h \in \mathcal{H}} \mathcal{L}(h) - \hat{\mathcal{L}}_S(h)$$

- **Step 2**: Apply McDiarmid's inequality

$\hat{\mathcal{L}}_S(h)$ is not perturbed by changing any x_i

$$\mathcal{L}(\hat{h}) - \hat{\mathcal{L}}_S(\hat{h}) \leq \mathbb{E}\left[\sup_{h \in \mathcal{H}} \mathcal{L}(h) - \hat{\mathcal{L}}_S(h)\right] + \tilde{\mathcal{O}}\left({}^{1}\!/_{\sqrt{n}}\right)$$

- **Step 3**: Analyze the expected suprēmus excess risk

$$\mathbb{E}\left[\sup_{h \in \mathcal{H}} \mathcal{L}(h) - \hat{\mathcal{L}}_S(h)\right] = \mathbb{E}\left[\sup_{h \in \mathcal{H}} \mathbb{E}\left[\hat{\mathcal{L}}_{\tilde{S}}(h)\right] - \hat{\mathcal{L}}_S(h)\right]$$

$$\leq \mathbb{E}\left[\sup_{h \in \mathcal{H}} \hat{\mathcal{L}}_{\tilde{S}}(h) - \hat{\mathcal{L}}_S(h)\right] \text{(Jensen's inequality)}$$

Analyzing the Expected Suprēmus Excess Risk

$$\mathbb{E}\left[\sup_{h\in\mathcal{H}} \hat{\mathcal{L}}_{\tilde{S}}(h) - \hat{\mathcal{L}}_{S}(h)\right]$$

- For unary losses $\hat{\mathcal{L}}_S(\cdot) = \sum \ell_{x_i}(\cdot)$
- Analyzing this term through symmetrization easy

$$\frac{1}{\mathrm{n}}\mathbb{E}\left[\sup_{h\in\mathcal{H}} \sum \ell_{x_i}(h) - \ell_{\tilde{x}_i}(h)\right] \leq \frac{2}{n}\mathbb{E}\left[\sup_{h\in\mathcal{H}} \sum \epsilon_i \ell_{x_i}(h)\right]$$

$$\leq \frac{2L}{n}\mathbb{E}\left[\sup_{h\in\mathcal{H}} \sum \epsilon_i h(x_i)\right] \approx \mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$$

Part II: Batch Learning

Batch Learning for Pairwise Loss Functions

Training with Pairwise Loss Functions

- Given training data $x_1, x_2, \ldots x_n$, choose a U-statistic
- U-statistic should use terms like $\ell_{(x_i,x_j)}(h)$ (**the kernel**)
- Population risk defined as $\mathcal{L}(\cdot) = \mathbb{E}\ell_{(x,x')}(\cdot)$

Examples:

- For any index set $\Omega \subset [n] \times [n]$, define

$$\hat{\mathcal{L}}_S(\cdot;\Omega) = \frac{1}{|\Omega|}\sum_{(i,j)\in\Omega} \ell_{(x_i,x_j)}(\cdot)$$

- Choice of $\Omega = \{(i,j): i \neq j\}$ maximizes data utilization
- Various ways of optimizing $\inf_{h\in\mathcal{H}} \hat{\mathcal{L}}_S(h)$ (e.g. SSG)

Generalization bounds for Pairwise Loss Functions

- **Step 1**: Bound excess risk by suprēmus excess risk

$$\mathcal{L}(\hat{h}) - \hat{\mathcal{L}}_S(\hat{h}) \le \sup_{h \in \mathcal{H}} \mathcal{L}(h) - \hat{\mathcal{L}}_S(h)$$

- **Step 2**: Apply McDiarmid's inequality

  Check that $\hat{\mathcal{L}}_S(h)$ is not perturbed by changing any x_i

$$\mathcal{L}(\hat{h}) - \hat{\mathcal{L}}_S(\hat{h}) \le \mathbb{E}\left[\sup_{h \in \mathcal{H}} \mathcal{L}(h) - \hat{\mathcal{L}}_S(h)\right] + \tilde{\mathcal{O}}\left({}^{1}\!/_{\sqrt{n}}\right)$$

- **Step 3**: Analyze the expected suprēmus excess risk

$$\mathbb{E}\left[\sup_{h \in \mathcal{H}} \mathcal{L}(h) - \hat{\mathcal{L}}_S(h)\right] = \mathbb{E}\left[\sup_{h \in \mathcal{H}} \mathbb{E}\left[\hat{\mathcal{L}}_{\tilde{S}}(h)\right] - \hat{\mathcal{L}}_S(h)\right]$$

$$\le \mathbb{E}\left[\sup_{h \in \mathcal{H}} \hat{\mathcal{L}}_{\tilde{S}}(h) - \hat{\mathcal{L}}_S(h)\right] \text{(Jensen's inequality)}$$

Analyzing the Expected Suprēmus Excess Risk

$$\mathbb{E}\left[\sup_{h\in\mathcal{H}} \hat{\mathcal{L}}_{\tilde{S}}(h) - \hat{\mathcal{L}}_S(h)\right]$$

- For pairwise losses $\hat{\mathcal{L}}_S(\cdot) = \sum_{i\neq j} \ell_{(x_i,x_j)}(\cdot)$
- Clean symmetrization not possible due to **coupling**

$$2\mathbb{E}\left[\sup_{h\in\mathcal{H}} \sum_i \sum_j \ell_{(\tilde{x}_i,\tilde{x}_j)}(h) - \ell_{(x_i,x_j)}(h)\right]$$

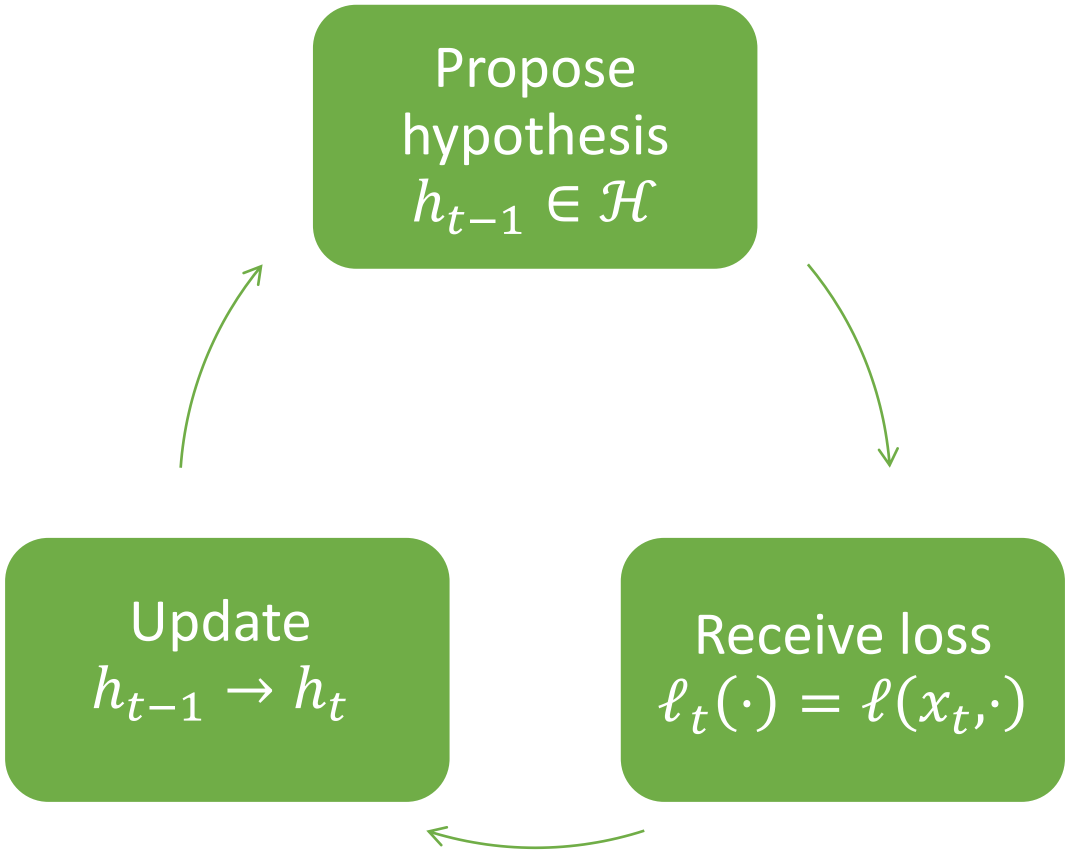
- Solutions [see Clémençon *et al* Ann. Stat. '08]
  - Alternate representation of U-statistics
  - Hoeffding decomposition

Part III: Online Learning

Part III: Online Learning

A Whirlwind Tour of Online Learning for Unary Losses

Model for Online Learning with Unary Losses



- Regret

$$\mathfrak{R}_T = \sum \ell_t(h_{t-1}) - \inf_{h \in \mathcal{H}} \sum \ell_t(h)$$

Online Learning Algorithms

- Generalized Infinitesimal Gradient Ascent (GIGA) [Zinkevich '03]

$$h_t = h_{t-1} - \eta_t \nabla_h \ell_t(h_{t-1})$$

- Follow the Regularized Leader (FTRL) [Hazan *et al* '06]

$$h_t = \underset{h \in \mathcal{H}}{\operatorname{argmin}} \sum_{\tau=1}^{t-1} \ell_\tau(h) + \sigma_t \|h\|^2$$

- Under some conditions

$$\mathfrak{R}_T \leq \mathcal{O}\left(\sqrt{T}\right)$$

- Under **strong**er conditions

$$\mathfrak{R}_T \leq \mathcal{O}(\log T)$$

Online to Batch Conversion for Unary Losses

- Key insight: h_{t-1} is evaluated on an unseen point [Cesa-Bianchi *et al* ‘01]
  $\mathbb{E}[\ell_t(h_{t-1})|\sigma(x_1, \ldots, x_{t-1})] = \mathbb{E}\ell(h_{t-1}, x_t) = \mathcal{L}(h_{t-1})$
- Set up a martingale difference sequence

$$V_t = \mathcal{L}(h_{t-1}) - \ell_t(h_{t-1})$$
$$\mathbb{E}[V_t|\sigma(x_1, \ldots, x_{t-1})] = 0$$

- Azuma-Hoeffding gives us

$$\sum \mathcal{L}(h_{t-1}) \leq \sum \ell_t(h_{t-1}) + \tilde{O}\left(\sqrt{T}\right)$$
$$\sum \ell_t(h^*) \geq T\mathcal{L}(h^*) - \tilde{O}\left(\sqrt{T}\right)$$

- Together we get

$$\sum \mathcal{L}(h_{t-1}) - T\mathcal{L}(h^*) \leq \mathfrak{R}_T + \tilde{O}\left(\sqrt{T}\right)$$

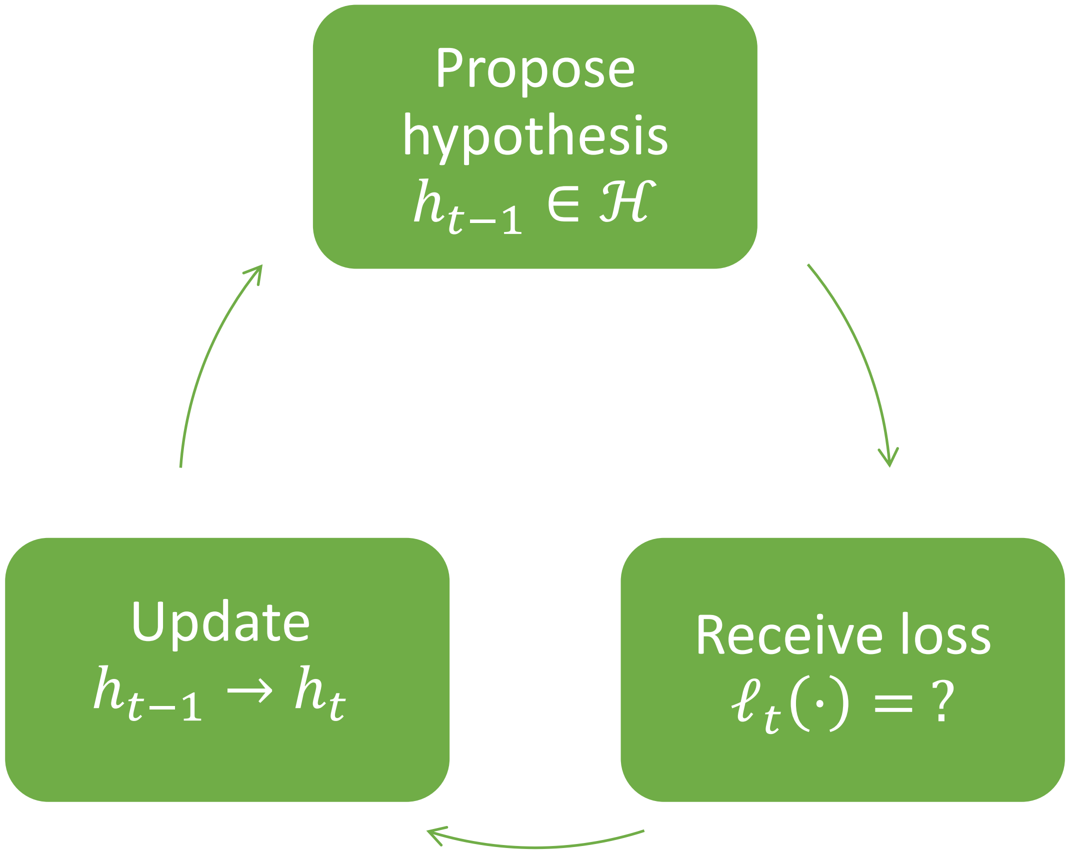
Online to Batch Conversion for Unary Losses

- Hypothesis selection
  - Convex loss function $\hat{h} = \frac{1}{T}\sum h_t$
$$\mathcal{L}(\hat{h}) \leq \frac{1}{T}\sum \mathcal{L}(h_t) \leq \mathcal{L}(h^*) + \frac{\mathfrak{R}_T}{T} + \tilde{\mathcal{O}}\left(\frac{1}{\sqrt{T}}\right)$$
  - More involved for non convex losses
- Better results possible [Tewari-Kakade '08]
  - Assume strongly convex loss functions
$$\sum \mathcal{L}(h_{t-1}) \leq T\mathcal{L}(h^*) + \mathfrak{R}_T + \tilde{\mathcal{O}}\left(\sqrt{\mathfrak{R}_T}\right)$$
  - For $\mathfrak{R}_T = \mathcal{O}(\log T)$, this reduces to
$$\mathcal{L}(\hat{h}) \leq \frac{1}{T}\sum \mathcal{L}(h_t) \leq \mathcal{L}(h^*) + \tilde{\mathcal{O}}\left(\frac{\log T}{T}\right)$$

Part III: Online Learning

Online Learning for Pairwise Loss Functions

Model for Online Learning with Pairwise Losses



- Regret

$$\mathfrak{R}_T = ?$$

Defining Instantaneous Loss and Regret

- At time t, we receive point x_t
- Natural definition of instantaneous loss:
  All the pairwise interactions x_t has with previous points

$$\ell_t(\cdot) = \sum_{\tau=1}^{t-1} \ell_{(x_t,x_\tau)}(\cdot)$$

- Corresponding notion of regret

$$\mathfrak{R}_T = \sum \ell_t(h_{t-1}) - \inf_{h\in\mathcal{H}} \sum \ell_t(h)$$

- Note that this notion of instantaneous loss satisfies

$$\forall h \in \mathcal{H}, \sum \ell_t(h) = \sum_{i<j} \ell_{(x_i,x_j)}(h) = \frac{1}{2}\hat{\mathcal{L}}_S(h)$$

Online Learning Algorithm with Pairwise Losses

- For regularity, we use a normalized loss

$$\ell_t(\cdot) = \frac{1}{t-1}\sum_{\tau=1}^{t-1} \ell_{(x_t, x\_\tau)}(\cdot)$$

- Note that $\ell_t(\cdot)$ is convex, bounded and Lipchitz if ℓ is so
- Turns out GIGA works just fine

$$h_t = h_{t-1} - \eta_t \nabla_h \ell_t(h_{t-1})$$

- Guarantees similar regret bounds

$$\mathfrak{R}_T \leq \mathcal{O}\left(\sqrt{T}\right)$$

Online Learning Algorithm with Pairwise Losses

- Implementing GIGA requires storing previous history

$$\nabla_h \ell_t(\cdot) = \frac{1}{t-1} \sum_{\tau=1}^{t-1} \nabla_h \ell_{(x_t, x_\tau)}(\cdot)$$

- To reduce memory usage, keep a **snapshot** of history
- Limited memory **buffer** $B = [\square_1, \square_2, \ldots, \square_s]$
- Modified instantaneous loss

$$\ell_t^{\text{buf}}(\cdot) = \frac{1}{s} \sum_{x \in B_{t-1}} \ell_{(x_t, x)}(\cdot)$$

- Responsibilities: at each time step t
  - Update hypothesis $h_{t-1} \to h_t$ (same as GIGA but with $\ell_t^{\text{buf}}(\cdot)$)
  - Update buffer $\text{UPDATE}\,(B_{t-1}, x_t) \to B_t$

Buffer Update Algorithm

- Online sampling algorithm for i.i.d. samples [K. *et al* ‘13]

RS-x: Reservoir sampling with replacement

Regret Analysis for GIGA with RS-x

- RS-x gives the following guarantee
  At any fixed time t, the buffer B contains s i.i.d. samples from the previous history $H_t = \{x_1, \ldots, x_{t-1}\}$
- Use this to prove a **Regret Conversion Bound**
- Basic idea
  - Prove a **finite buffer regret** bound

$$\frac{1}{\mathrm{T}}\sum \ell_t^{\text{buf}}(h_{t-1}) \leq \inf_{h \in \mathcal{H}} \frac{1}{T}\sum \ell_t^{\text{buf}}(h) + \mathcal{O}\left(\frac{1}{\sqrt{T}}\right)$$

  - Use **uniform convergence** style bounds to show

$$\ell_t(h_{t-1}) \approx \ell_t^{\text{buf}}(h_{t-1}) \pm \tilde{\mathcal{O}}\left(\frac{1}{\sqrt{s}}\right)$$

Regret Analysis for GIGA with RS-x: Step 1

Finite Buffer Regret

- The modified algo. uses $\ell_t^{\text{buf}}(\cdot)$to update hypothesis
- $\ell_t^{\text{buf}}(\cdot)$ is also convex, bounded and Lipchitz given B
- Standard GIGA analysis gives us

$$\frac{1}{\mathrm{T}}\sum \ell_t^{\text{buf}}(h_{t-1}) \leq \inf_{h\in\mathcal{H}} \frac{1}{T}\sum \ell_t^{\text{buf}}(h) + \mathcal{O}\left(\frac{\mathfrak{R}_T^{\text{buf}}}{T}\right),$$

where $\mathfrak{R}_T^{\text{buf}} = \mathcal{O}\left(\sqrt{T}\right)$

Regret Analysis for GIGA with RS-x: Step 2

Uniform convergence

- Think of H_t as population and B as i.i.d. sample of size s
- Define $g_x(\cdot) = \ell_{(x_t,x)}(\cdot)$ and set unif. dist. over H_t
  - Population risk analysis

$$\mathcal{G}(\cdot) = \mathbb{E}g_x(\cdot) = \frac{1}{t-1}\sum_{\tau=1}^{t-1} \ell_{(x_t,x_\tau)}(\cdot) = \ell_t(\cdot)$$

  - Empirical risk analysis

$$\hat{\mathcal{G}}(\cdot) = \frac{1}{s}\sum\nolimits_{x\in B_{t-1}} g_{(x)}(\cdot) = \frac{1}{s}\sum\nolimits_{x\in B_{t-1}} \ell_{(x_t,x)}(\cdot) = \ell_t^{\mathrm{buf}}(\cdot)$$

- Finish off using $\left\|\mathcal{G}(\cdot) - \hat{\mathcal{G}}(\cdot)\right\|_\infty \leq \tilde{\mathcal{O}}\left(\frac{1}{\sqrt{s}}\right)$

Regret Analysis for GIGA with RS-x: Wrapping up

- Convert finite buffer regret to true regret
- Three results:

$$\forall t, \ell_t(h_{t-1}) \leq \ell_t^{\text{buf}}(h_{t-1}) + \tilde{\mathcal{O}}(1/\sqrt{s})$$
$$\forall h, \forall t, \ell_t^{\text{buf}}(h) \leq \ell_t(h) + \tilde{\mathcal{O}}(1/\sqrt{s})$$
$$\forall h, \frac{1}{T}\sum \ell_t^{\text{buf}}(h_{t-1}) \leq \frac{1}{T}\sum \ell_t^{\text{buf}}(h) + \frac{\mathfrak{R}_T^{\text{buf}}}{T}$$

- Combine to get

$$\frac{1}{\mathrm{T}}\sum \ell_t(h_{t-1}) \leq \inf_{h \in \mathcal{H}} \frac{1}{T}\sum \ell_t(h) + \tilde{\mathcal{O}}\left(\frac{1}{\sqrt{s}}\right) + \frac{\mathfrak{R}_T^{\text{buf}}}{T}$$

i.e. $\mathfrak{R}_T \leq \mathfrak{R}_T^{\text{buf}} + \tilde{\mathcal{O}}\left(\frac{T}{\sqrt{s}}\right) = \tilde{\mathcal{O}}\left(\frac{T}{\sqrt{s}}\right)$

Regret Analysis for GIGA with RS-x

- Better results possible for strongly convex losses
- For any $\epsilon > 0$, we can show

$$\frac{1}{\mathrm{T}}\sum \ell_t(h_{t-1}) \leq (1+\epsilon)\left(\inf_{h\in\mathcal{H}} \frac{\sum \ell_t(h)}{T} + \frac{\mathfrak{R}_T}{T}\right) + \tilde{\mathcal{O}}\left(\frac{1}{\epsilon s}\right)$$

- For realizable cases (i.e. $\mathcal{L}(h^*) = 0$), we can also show

$$\frac{1}{\mathrm{T}}\sum \ell_t(h_{t-1}) \leq \inf_{h\in\mathcal{H}} \frac{1}{T}\sum \ell_t(h) + \frac{\mathfrak{R}_T}{T} + \tilde{\mathcal{O}}\left(\frac{\sqrt{\mathfrak{R}_T}}{s}\right)$$

Online to Batch Conversion for Pairwise Losses

- Recall that in unary case, we had an MDS
$$V_t = \mathcal{L}(h_{t-1}) - \ell_t(h_{t-1})$$
- Recall, in pairwise case, we have
$$\mathcal{L}(\cdot) = \mathbb{E}\ell_{(x,x')}(\cdot)$$
$$\ell_t(\cdot) = \frac{1}{t-1}\sum_{\tau=1}^{t-1} \ell_{(x_t,x_\tau)}(\cdot)$$
- No longer an MDS since V_t and $V_\tau, \tau < t$ are coupled
$$\mathbb{E}[V_t|\sigma(H_t)] = \mathcal{L}(h_{t-1}) - \mathbb{E}[\ell_t(h_{t-1})|\sigma(H_t)] \neq 0$$

Online to Batch Conversion for Pairwise Losses

Solution:

- Martingale creation: let $\bar{\ell}_t(\cdot) = \mathbb{E}[\ell_t(\cdot) | \sigma(H_t)]$

$$V_t = \mathcal{L}(h_{t-1}) - \bar{\ell}_t(h_{t-1}) + \bar{\ell}_t(h_{t-1}) - \ell_t(h_{t-1})$$
$$V_t = P_t + Q_t$$

- Sequence Q_t is an MDS by construction: A.H. bounds
- Bounding P_t using uniform convergence
  - Be careful during symmetrization step
- End Result

$$\frac{1}{T}\sum \mathcal{L}(h_{t-1}) \leq \mathcal{L}(h^*) + \frac{\mathfrak{R}_T}{T} + \tilde{\mathcal{O}}\left(\frac{1}{\sqrt{T}}\right)$$

Faster Rates for Strongly Convex Losses

- Have to use *fast* rate results to bound both P_t and Q_t
- Fast rates for P_t
  For strongly **unary** convex loss functions $\ell_x(\cdot)$, we have
  $$\mathcal{L}(h) - \mathcal{L}(h^*) \leq (1+\epsilon)\left(\hat{\mathcal{L}}_S(h) - \hat{\mathcal{L}}_S(h^*)\right) + \tilde{O}\left(\frac{1}{\epsilon n}\right)$$
- Fast rates for Q_t
  Use Bernstein inequality for martingales
- End result
  $$\frac{1}{T}\sum \mathcal{L}(h_{t-1}) \leq \mathcal{L}(h^*) + \frac{\mathfrak{R}_T}{T} + \tilde{O}\left(\frac{\sqrt{\mathfrak{R}_T}}{T}\right)$$

Hidden Constants

- All our analyses involved Rademacher averages
  - Even for regret analysis and bounding P_t for slow/fast rates
  - Get dimension independent bounds for regularized classes
  - Weak dependence on dimensionality for sparse formulations
  - Earlier work [Wang *et al* '12] used covering number methods

- If constants not imp. then can try analyzing V_t directly
  - Use covering number arguments to get linear dep. on d

Some Interesting Projects

- Regret bounds require $s = \omega(\log T)$
  - Is this necessary: regret lower bound

- Learning higher order tensors
  - Scalability issues

- RS-x is a data oblivious sampling algorithm
  - Can throw away useful points by chance
  - Data aware sampling methods + corresponding regret bounds

That's all!

Get slides from the following URL
http://research.microsoft.com/en-us/people/t-purkar/

References

- Balcan, Maria-Florina and Blum, Avrim. On a Theory of Learning with Similarity Functions. In ICML, pp. 73-80, 2006.
- Cesa-Bianchi, Nicolo, Conconi, Alex, and Gentile, Claudio. On the Generalization Ability of On-Line Learning Algorithms. In NIPS, pp. 359-366, 2001.
- Clemencon, Stephan, Lugosi, Gabor, and Vayatis, Nicolas. Ranking and empirical minimization of Ustatistics. Annals of Statistics, 36:844-874, 2008.
- Cortes, Corinna, Mohri, Mehryar, and Rostamizadeh, Afshin. Two-Stage Learning Kernel Algorithms. In ICML, pp. 239-246, 2010.
- Hazan, Elad, Kalai, Adam, Kale, Satyen, and Agarwal, Amit. Logarithmic Regret Algorithms for Online Convex Optimization. In COLT, pp. 499-513,2006.

References

- Jin, Rong, Wang, Shijun, and Zhou, Yang. Regularized Distance Metric Learning: Theory and Algorithm. In NIPS, pp. 862-870, 2009.
- Kakade, Sham M. and Tewari, Ambuj. On the Generalization Ability of Online Strongly Convex Programming Algorithms. In NIPS, pp. 801-808, 2008.
- Kar, Purushottam, Sriperumbudur, Bharath, Jain, Prateek, and Karnick, Harish, On the Generalization Ability of Online Learning Algorithms for Pairwise Loss Functions. In ICML, 2013.
- Narasimhan, Harikrishna and Agarwal, Shivani, A Structural SVM Based Approach for Optimizing Partial AUC, In ICML, 2013.
- De la Peña, Victor H. and Giné, Evariste, Decoupling: From Dependence to Independence. Springer, New York, 1999.

References

- Sridharan, Karthik, Shalev-Shwartz, Shai, and Srebro, Nathan. Fast Rates for Regularized Objectives. In NIPS, pp. 1545-1552, 2008.
- Wang, Yuyang, Khardon, Roni, Pechyony, Dmitry, and Jones, Rosie. Generalization Bounds for Online Learning Algorithms with Pairwise Loss Functions. In COLT 2012.
- Zhao, Peilin, Hoi, Steven C. H., Jin, Rong, and Yang, Tianbao. Online AUC Maximization. In ICML, pp. 233-240, 2011.
- Xing, Eric P., Ng, Andrew Y., Jordan, Michael I., and Russell, Stuart J. Distance Metric Learning with Application to Clustering with Side-Information. In NIPS, pp. 505-512, 2002.
- Zinkevich, Martin. Online Convex Programming and Generalized Infinitesimal Gradient Ascent. In ICML, pp. 928-936, 2003.