ECML 2015 Big Targets Workshop

Paul Mineiro

Extreme Challenges

- How can we generalize well?
- Can we compete with OAA?
- When can we predict quickly?

How can we generalize well?

Chasing Tails

- Typical extreme datasets have many rare classes.

Chasing Tails

- Typical extreme datasets have many rare classes.
- What are the implications for generalization?

Chasing Tails

- Typical extreme datasets have many rare classes.
- What are the implications for generalization?
- Let's use the bootstrap to get intuition.

Bootstrap Lesson

Observation (Tail Frequencies)

The true frequencies of tail classes is not clear given the training set.

Two Loss Patterns

- All classes below have 1 training example.
- Which hypothesis do you like better?

	h_1	h_2
class 1	1	0.6
class 2	1	0.6
class 3	0	0.42
class 4	0	0.42

Two Loss Patterns

- All classes below have 1 training example.
- Which hypothesis do you like better?

	h_1	h_2
class 1	1	0.6
class 2	1	0.6
class 3	0	0.42
class 4	0	0.42

- ERM likes h_1 better.
- I like h_2 better.

The Extreme Deficiencies of ERM

- ERM cares only about average loss.

$$h^* = \operatorname*{argmin}_{h \in \mathcal{H}} \mathbb{E}_{(x,y) \sim D} \left[l(h(x); y) \right]$$

- …but extreme learning empirical losses can have high variance.
- ERM doesn't care about empirical loss variance.
- ERM is based upon a uniform bound on the hypothesis space.

eXtreme Risk Minimization

- Sample Variance Penalization (XRM) penalizes combination of expected loss and loss variance.

$$h^* = \operatorname*{argmin}_{h \in \mathcal{H}} \left(\mathbb{E}\left[l(h(x); y) \right] + \kappa \mathbb{V}\left[l(h(x); y) \right] \right)$$

- (κ is a hyperparameter in practice)
- XRM is based upon empirical Bernstein bounds.

Example: Neural Language Modeling

- Mini-batch XRM gradient:

$$\mathbb{E}_i \left[\left(1 + \kappa \frac{l_i(\phi) - \mathbb{E}_j \left[l_j(\phi) \right]}{\sqrt{\mathbb{E}_j \left[l_j^2(\phi) \right] - \mathbb{E}_j \left[l_j(\phi) \right]^2}} \right) \frac{\partial l_i(\phi)}{\partial \phi} \right]$$

- Smaller than average loss $\implies$ lower learning rate
- Larger than average loss $\implies$ larger learning rate
- Loss variance is the unit of loss measurement

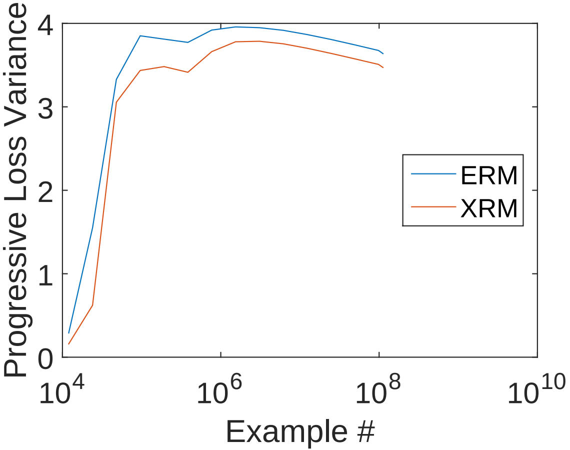
Example: Neural Language Modeling

- `enwiki9` data set
- FNN-LM of Zhang et. al.
- Same everything except κ.

method	perplexity
ERM ($\kappa = 0$)	106.3
XRM ($\kappa = 0.25$)	104.1

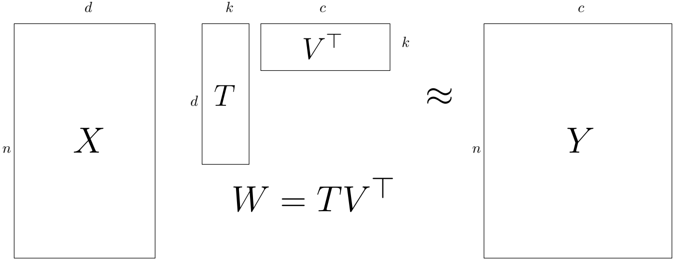
- Modest lift, but over SOTA baseline and with minimal code changes.

Example: Neural Language Modeling



Example: Randomized Embeddings

- Based upon (randomized) SVD.



- How to adapt black-box technique to XRM?
- Idea: proxy model $\Longrightarrow$ importance weights.

Imbalanced binary XRM

- Binary classification with constant predictor.
- $l(y; q) = y \log(q) + (1 - y) \log(1 - q)$

$$1 + \kappa \left. \frac{l(y; q) - \mathbb{E}\left[l(\cdot; q)\right]}{\sqrt{\mathbb{E}\left[l^2(\cdot; q)\right] - \mathbb{E}\left[l(\cdot; q)\right]^2}} \right|_{q=p}$$

$$= \begin{cases} 1 - \kappa \sqrt{\frac{p}{1-p}} & y = 0 \\ 1 + \kappa \sqrt{\frac{1-p}{p}} & y = 1 \end{cases} \qquad (p \leq 0.5)$$

XRM Rembed for ODP

- Compute base rate q_c each class c.
- Importance weight $(1 + \kappa(1/\sqrt{q_{y_i}}))$.

method	error rate (%)
ODP ERM	[80.3, 80.4]
ODP XRM ($\kappa = 1$)	[78.5, 78.7]

- Modest lift, but over SOTA baseline and with minimal code changes.

Summary

- The tail can deviate wildly between train and test.
- Controlling loss variance helps a little bit.
- Speculation: explicitly treat the head and tail differently?