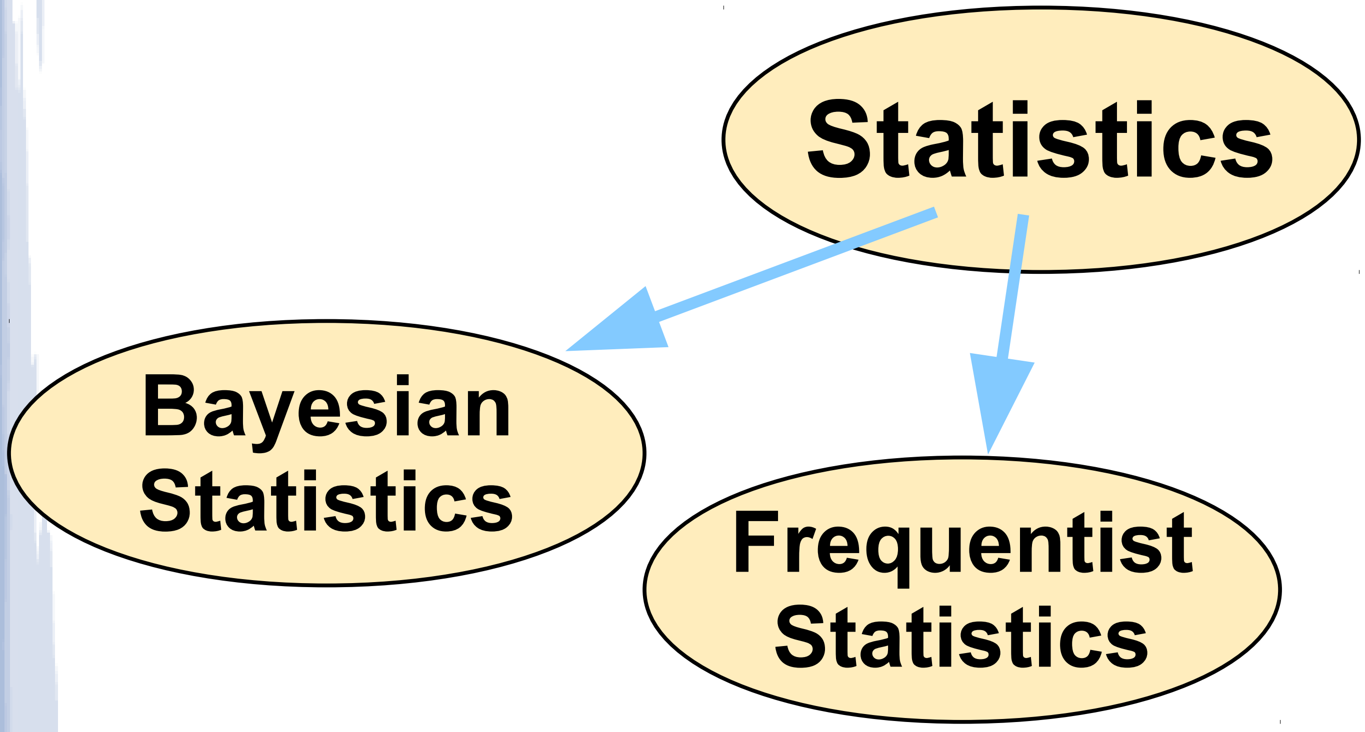
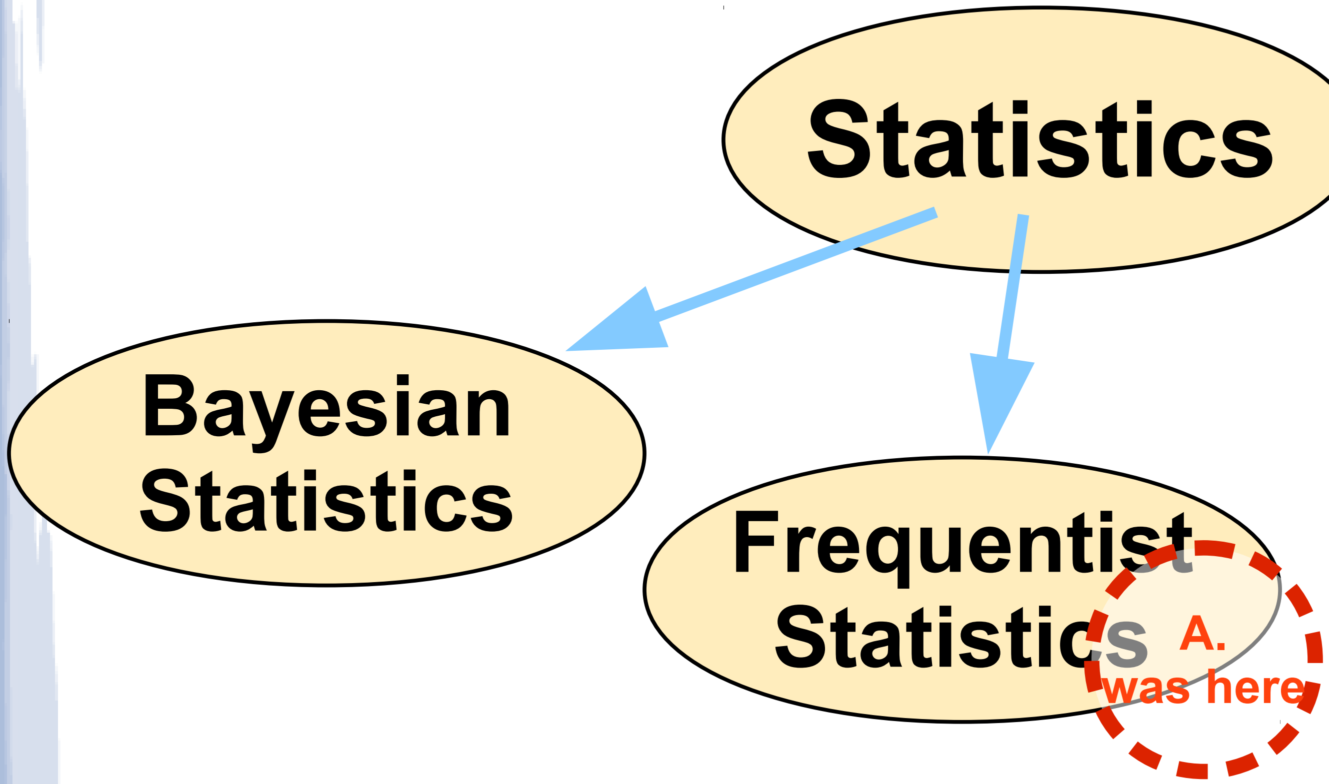
General Mathematics Colloquium Leiden, December 4, 2014

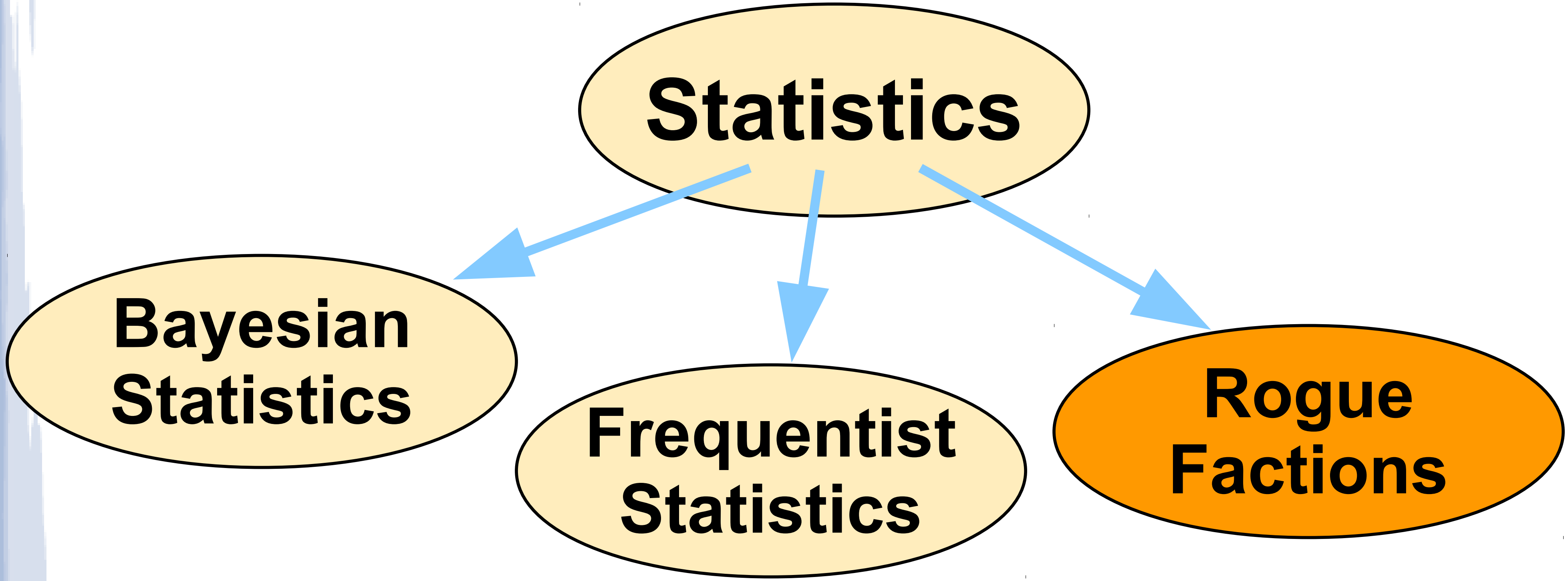
An Introduction to
Game-theoretic Online Learning

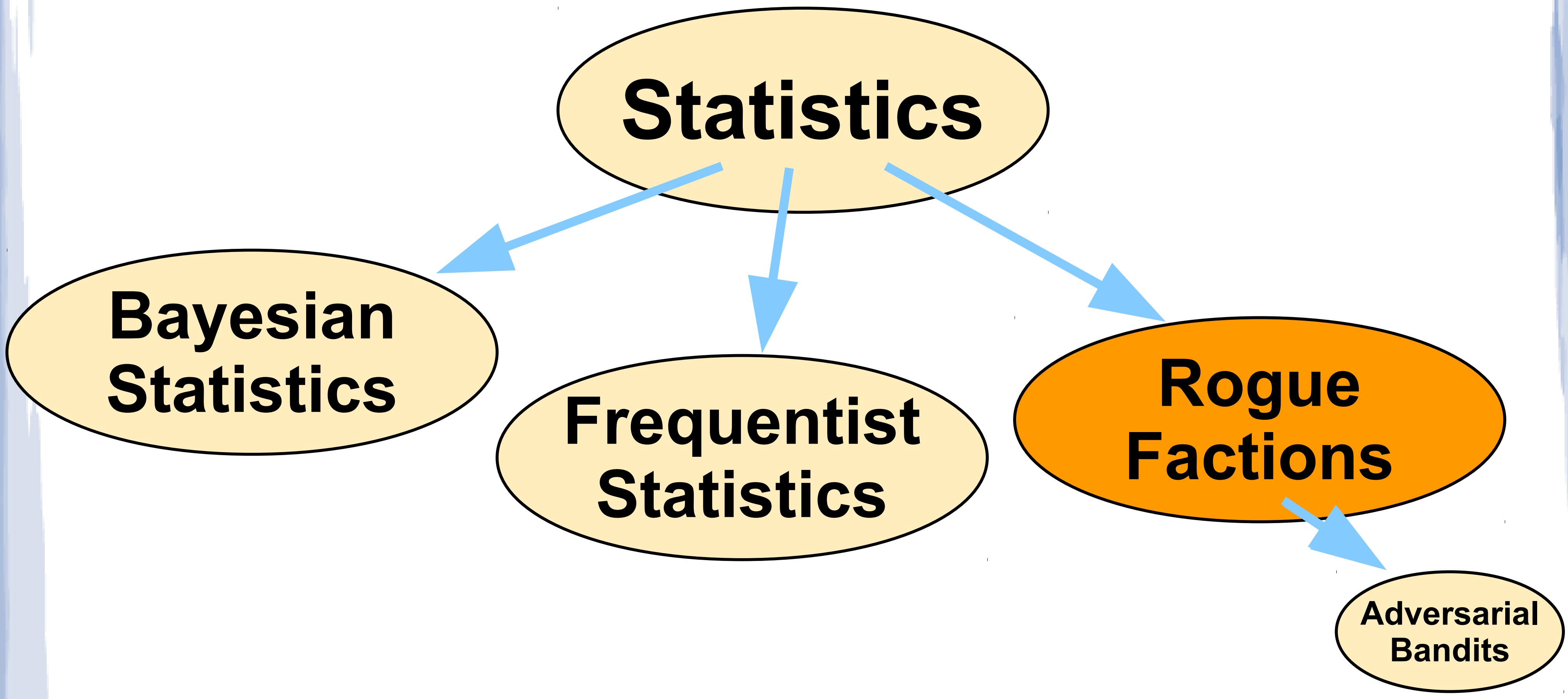
Tim van Erven

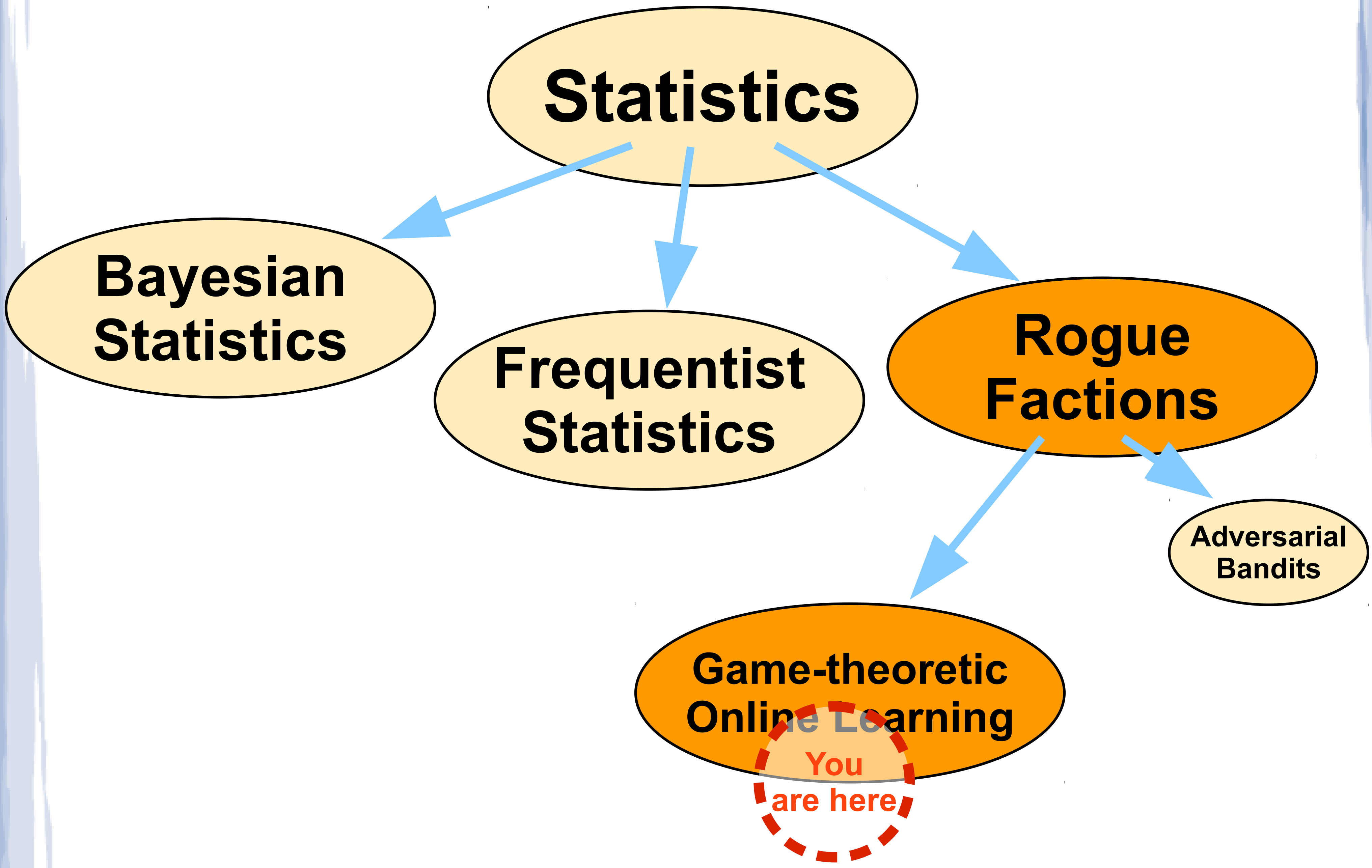
Universiteit Leiden


Statistics
Bayesian Statistics
Frequentist Statistics


Statistics
Bayesian Statistics
Frequentist Statistics
A. was here


Statistics
Bayesian Statistics
Frequentist Statistics
Rogue Factions


Statistics
Bayesian Statistics
Frequentist Statistics
Rogue Factions
Adversarial Bandits



Statistics
Bayesian Statistics
Frequentist Statistics
Rogue Factions
Adversarial Bandits
Game-theoretic Online Learning
You are here

Outline

- Introduction to Online Learning
  - **Game-theoretic Model**
  - Regression example: Electricity
  - Classification example: Spam
- Three algorithms
- Tuning the Learning Rate

Game-Theoretic Online Learning

- Predict data that arrive one by one
- Model: repeated game against an **adversary**
- Applications:
  - spam detection
  - data compression
  - online convex optimization
  - predicting electricity consumption
  - predicting air pollution levels
  - ...

Repeated Game (Informally)

- **Sequentially** predict outcomes $x_1, x_2, \ldots$
- Measure quality of prediction a_t by loss $\ell(x_t, a_t)$

- Before predicting x_t, get predictions (=advice) from K **experts**
- Goal: to predict as well as the best expert over T rounds.

- Data and Advice can be **adversarial**

Repeated Game

- Every round $t = 1, \ldots, T$:
  1. Get expert predictions $a_t^k \quad (k = 1, \ldots, K)$
  2. Predict a_t^*
  3. Outcome x_t is revealed
  4. Measure losses $\ell(x_t, a_t^*), \ell(x_t, a_t^k)$

Repeated Game

- Every round $t = 1, \ldots, T$:
  1. Get expert predictions $a_t^k \quad (k = 1, \ldots, K)$
  2. Predict a_t^*
  3. Outcome x_t is revealed
  4. Measure losses $\ell(x_t, a_t^*), \ell(x_t, a_t^k)$

- Best expert: $L^* = \min_k \sum_{t=1}^{T} \ell(x_t, a_t^k)$
- Goal: minimize ***regret*** $\sum_{t=1}^{T} \ell(x_t, a_t^*) - L^*$

Outline

- Introduction to Online Learning
  - Game-theoretic Model
  - **Regression example: Electricity**
  - Classification example: Spam
- Three algorithms
- Tuning the Learning Rate

Regression Example: Electricity

- Électricité de France: predict electricity demand one day ahead, every day [Devaine, Gaillard, Goude, Stoltz, 2013]
- Experts: K complicated regression models
- Loss: $\ell(x, a) = (a - x)^2$

Regression Example: Electricity

- Électricité de France: predict electricity demand one day ahead, every day [Devaine, Gaillard, Goude, Stoltz, 2013]
- Experts: K complicated regression models
- Loss: $\ell(x, a) = (a - x)^2$
- Best model after one year: $L^* = \min_k \sum_{t=1}^{T} \ell(x_t, a_t^k)$
- How much worse are we? $\sum_{t=1}^{T} \ell(x_t, a_t^*) - L^*$

Outline

- Introduction to Online Learning
  - Game-theoretic Model
  - Regression example: Electricity
  - **Classification example: Spam**
- Three algorithms
- Tuning the Learning Rate

Example: Spam Detection

Subject	From	
Gratis Turkije. . .	Reizen Center	$x_1 =$ **spam**
uitnodiging hoorzitting reorganisatie FEW dinsdag 20 se...	Ivo van Stokkum	$x_2 =$ **ham**
Re: Urgent Business Inquiry.	Ubc Ltd	$x_3 =$ **spam**
Reminder: first colloquium	Jeu, R.M.H. de	$x_4 =$ **ham**
Informatie over VUnet	College van Bestuur	$x_5 =$ **ham**
USD 500 Free Deposit at PartyPoker!	PartyPoker	$x_6 =$ **spam**
YOU ARE A WINNER!!! VERY URGENT NOTIFICATION.	UK INTL. LOTTERY PROMOTION	$x_7 =$ **spam**
bachelor/master diploma uitreiking 14 september	Sotiriou, M.	$x_8 =$ **ham**
HAPPY NEW YEAR 2068	Anil Shilpakar	$x_9 =$ **spam**
Thailand Package	Anil Shilpakar	$x_{10} =$ **spam**

Classification Example: Spam

- Experts: K spam detection algorithms
- Messages: $x \in \{\text{ham}, \text{spam}\}$
  Predictions: $a \in \{\text{ham}, \text{spam}\}$
- Loss:

$$\ell(x, a) = \begin{cases} 0 & \text{if correct: } a = x \\ 1 & \text{if wrong: } a \neq x \end{cases}$$

- Regret: extra mistakes we make over best algorithm on T messages

Outline

- Introduction to Online Learning
- Three algorithms:
    1. **Halving**
    2. Follow the Leader (FTL)
    3. Follow the Regularized Leader (FTRL)
- Tuning the Learning Rate

A First Algorithm: Halving

- Suppose one of the spam detectors is **perfect**

- Keep track of experts without mistakes so far:

$$S_t = \{k \mid \text{expert } k \text{ made no mistakes before round t}\}$$

- Halving algorithm:

$$a_t^* = \text{majority vote among experts in } S_t$$

- **Theorem:** regret $\leq \log_2 K$

A First Algorithm: Halving

Theorem: regret $\leq \log_2 K$

- Does not grow with T

Proof:

- Suppose halving makes m mistakes, regret = $m - 0$
- Every mistake eliminates at least half of S_t
- m is at most $\log_2 |S_1| = \log_2 K$ mistakes

Outline

- Introduction to Online Learning
- Three algorithms:
  1. Halving
  2. **Follow the Leader (FTL)**
  3. Follow the Regularized Leader (FTRL)
- Tuning the Learning Rate

Follow the Leader

- Want to remove unrealistic assumption that one expert is perfect
- FTL: copy the leader's prediction
- The **leader** at time t:

$$\hat{k}_t = \arg\min_k L_{t-1}^k \quad \text{(break ties randomly)}$$

where $L_t^k = \sum_{s=1}^{t} \ell(x_s, a_s^k)$ is cumulative loss for expert k

FTL Works with Perfect Expert

Theorem: Suppose one of the spam detectors is **perfect**. Then Expected regret $= O(\log K)$

Proof:

- Expected regret = E[nr. mistakes] - 0
- Worst case: experts get one loss in turn
- E[nr. mistakes] $= \frac{1}{K} + \frac{1}{K-1} + \ldots + \frac{1}{2} = O(\log K)$

FTL: More Good News

- No assumption of perfect expert

Theorem: regret $\leq \text{nr. leader changes/ties}$

FTL: More Good News

- No assumption of perfect expert

  Theorem: regret $\leq \text{nr. leader changes/ties}$

- **Proof sketch:**
  - No leader change: our loss = loss of leader, so the regret stays the same
  - Leader change: our regret increases at most by 1 (range of losses)

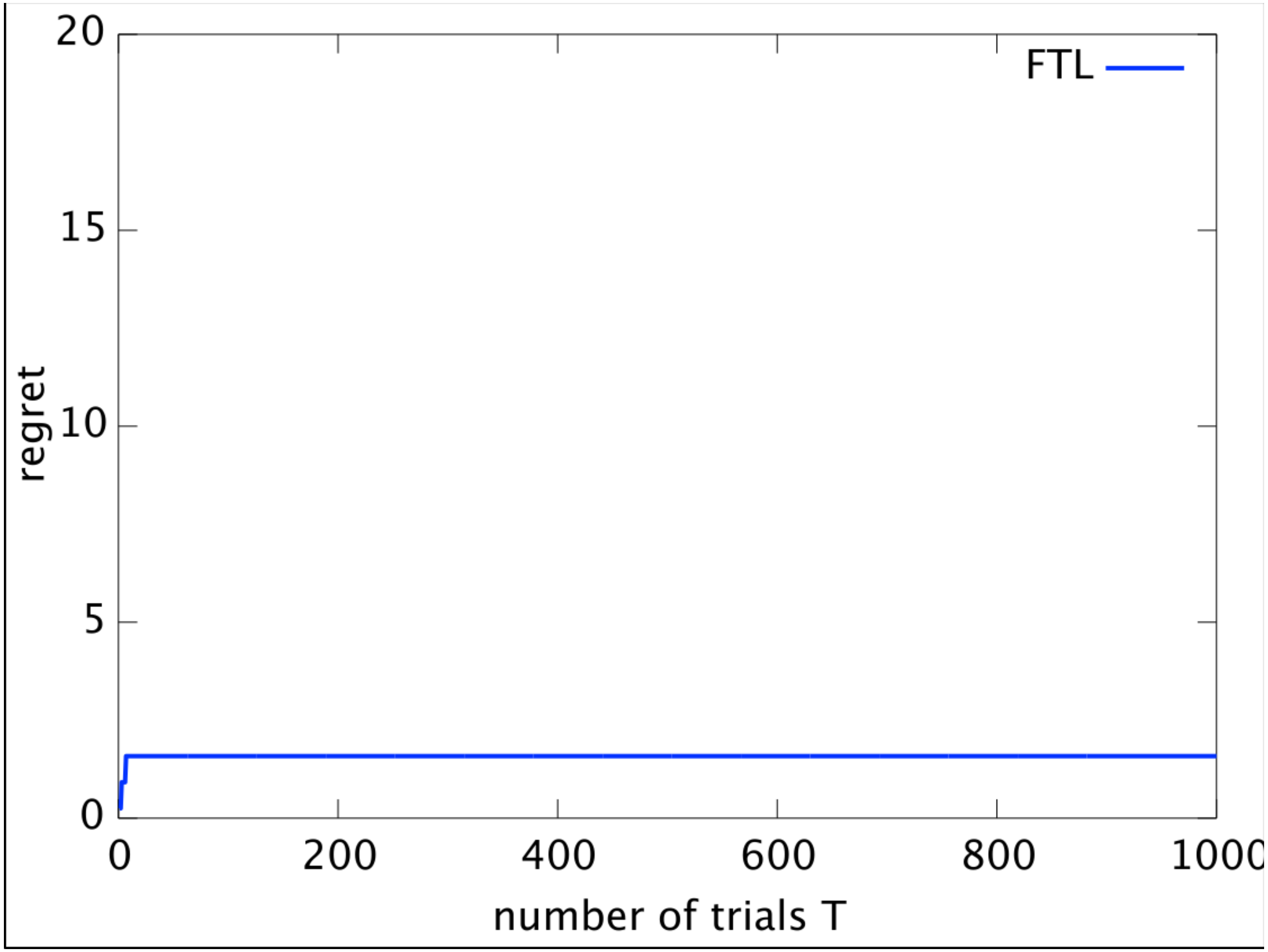
FTL: More Good News

- No assumption of perfect expert

  Theorem: regret $\leq \text{nr. leader changes/ties}$

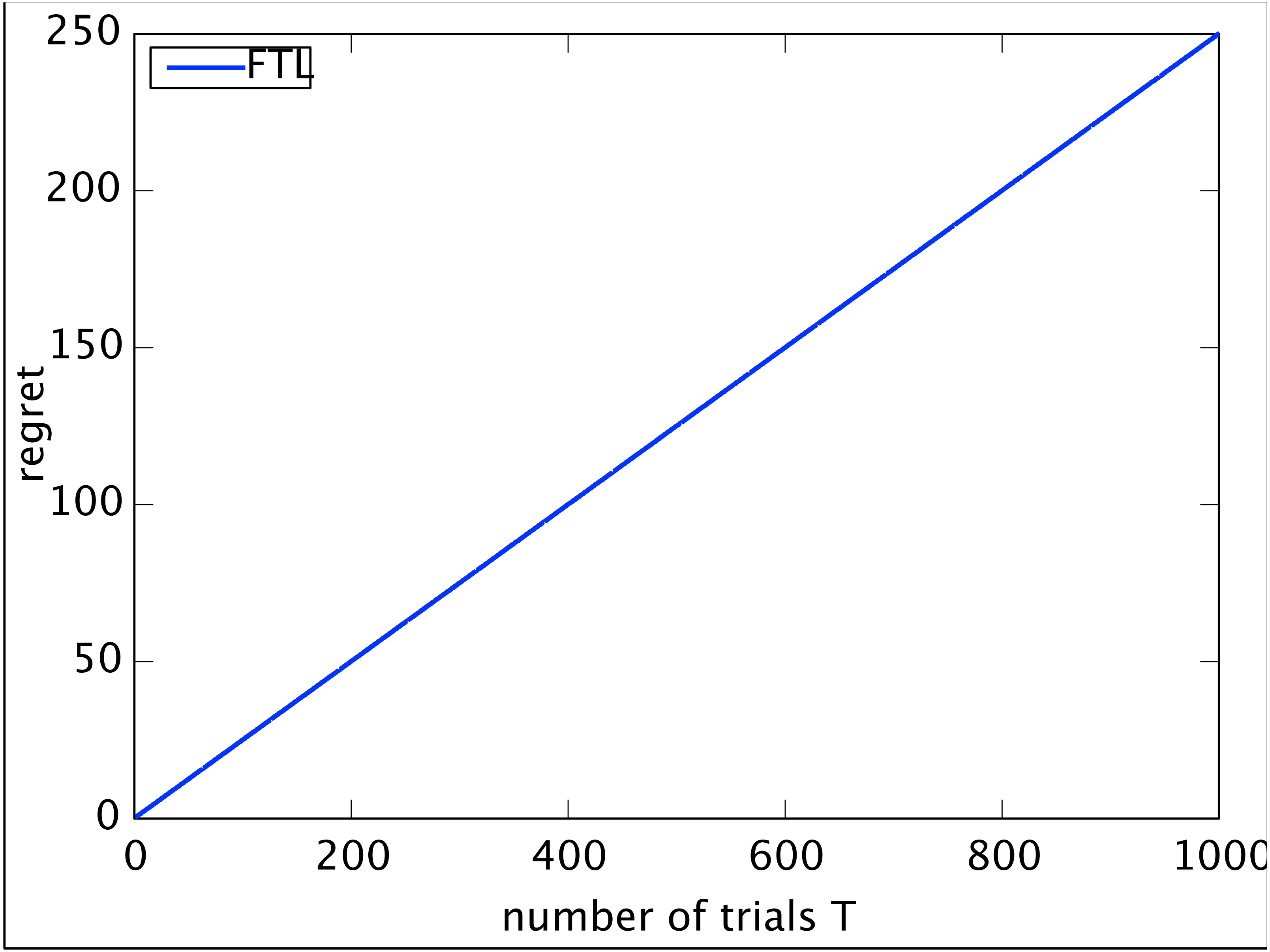
- **Proof sketch:**
  - No leader change: our loss = loss of leader, so the regret stays the same
  - Leader change, our regret increases at most by 1 (range of losses)
- Works well for i.i.d. losses, because the leader changes only finitely many times w.h.p.

FTL on IID Losses



- 4 experts with Bernoulli 0.1, 0.2, 0.3, 0.4 losses; regret = O(log K)

FTL Worst-case Losses



FTL Worst-case Losses

- Two experts with tie/leader change every round:

Expert 1	1	0	1	0	1	0
Expert 2	0	1	0	1	0	1
FTL	1/2	1	1/2	1	1/2	1

- Both experts have cumulative loss: $L^* = \frac{T}{2}$
- Regret $= \frac{3T}{4} - L^* = \frac{T}{4}$ is **linear** in T
- Problem: FTL too sure of itself when no ties!

Outline

- Introduction to Online Learning
- Three algorithms:
    1. Halving
    2. Follow the Leader (FTL)
    3. **Follow the Regularized Leader (FTRL)**
- Tuning the Learning Rate

Solution: Be Less Sure!

- Pull FTL choices towards uniform distribution
- Follow the Leader:
  - leader: $\hat{k}_t = \arg\min_k L_{t-1}^k$
  - as distribution: $\hat{w}_t = \arg\min_w \mathop{\mathbb{E}}_{k \sim w} [L_{t-1}^k]$
- Follow the **Regularized** Leader:

$$\hat{w}_t = \arg\min_w \mathop{\mathbb{E}}_{k \sim w} [L_{t-1}^k] + \frac{1}{\eta} \mathrm{KL}(w \| u)$$

  - add penalty for being away from uniform; Kullback-Leibler divergence in this talk

The Learning Rate

- Follow the Regularized Leader:

$$\hat{w}_t = \arg\min_{w} \mathop{\mathbb{E}}_{k \sim w}[L^k_{t-1}] + \frac{1}{\eta}\mathrm{KL}(w \| u)$$

- Very sensitive to choice of **learning rate** $\eta > 0$

$\eta \to \infty$	$\eta \to 0$
Follow the Leader	Don't learn at all

Outline

- Introduction to Online Learning
- Three algorithms
- Tuning the Learning Rate:
  - **Safe tuning**
  - The Price of Robustness
  - Learning the Learning Rate

The Worst-case Safe Learning Rate

Theorem: For FTRL regret $\leq \frac{\ln K}{\eta} + \frac{\eta T}{8}$

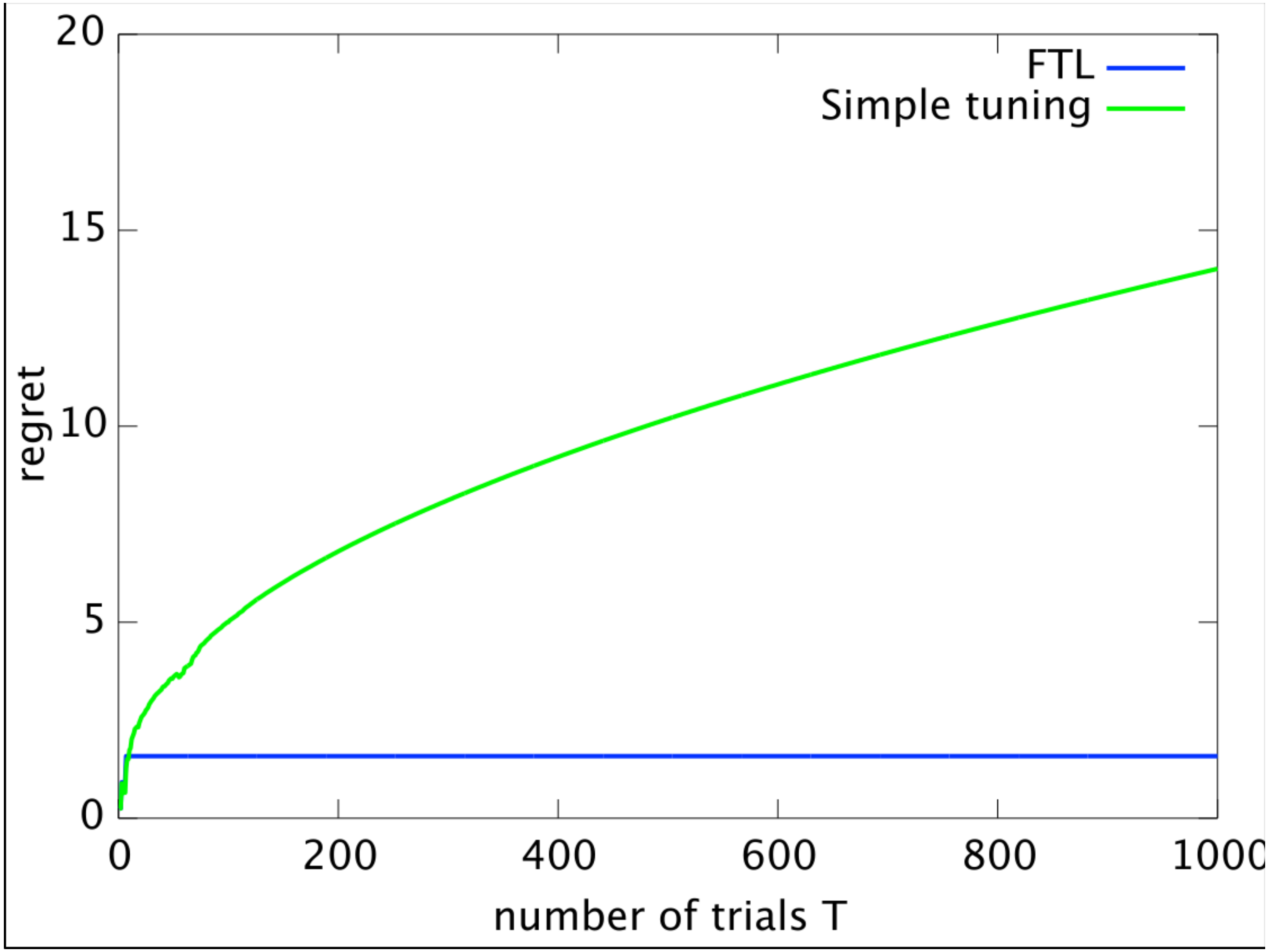
$$\eta = \sqrt{\frac{8 \ln K}{T}} \longrightarrow \text{regret} \leq \sqrt{\frac{T \ln(K)}{2}}$$

- **No (probabilistic) assumptions** about data!
- Optimal
- $O(\sqrt{T})$ is standard in online learning

Outline

- Introduction to Online Learning
- Three algorithms
- Tuning the Learning Rate:
  - Safe tuning
  - **The Price of Robustness**
  - Learning the Learning Rate

The Price of Robustness



- Safe tuning does much worse than FTL on i.i.d. losses

The Price of Robustness

Method	Special Case of FTRL	Perfect Expert Data	IID Data	Worst-case Data
Halving	no	$O(\log K)$	undefined	undefined
Follow the Leader	$\eta = \infty$ (very large)	$O(\log K)$	$O(\log K)$	$\Theta(T)$
FTRL with Worst-case Safe Tuning	$\eta = \sqrt{\frac{8 \ln K}{T}}$ (small)	$O(\sqrt{T \ln K})$	$O(\sqrt{T \ln K})$	$O(\sqrt{T \ln K})$

Can we **adapt** to optimal eta **automatically**?

Outline

- Introduction to Online Learning
- Three algorithms
- Tuning the Learning Rate:
  - Safe tuning
  - The Price of Robustness
  - **Learning the Learning Rate**

A Failed Approach: Being Meta

- We want $\min\{\text{regret}_{\text{FTL}}, \text{regret}_{\text{Safe}}\}$
- Idea: **meta-problem**
  - Expert 1: FTL
  - Expert 2: FTRL with Safe Tuning

A Failed Approach: Being Meta

- We want $\min\{\text{regret}_{\text{FTL}}, \text{regret}_{\text{Safe}}\}$
- Idea: **meta-problem**
  - Expert 1: FTL
  - Expert 2: FTRL with Safe Tuning

- Regret $= \min\{\text{regret}_{\text{FTL}}, \text{regret}_{\text{Safe}}\} + \text{meta-regret}$
- Best of both worlds if meta-regret small!

A Failed Approach: Being Meta

- We want $\min\{\text{regret}_{\text{FTL}}, \text{regret}_{\text{Safe}}\}$
- Idea: **meta-problem**
  - Expert 1: FTL
  - Expert 2: FTRL with Safe Tuning

- Regret $= \min\{\text{regret}_{\text{FTL}}, \text{regret}_{\text{Safe}}\} + \text{meta-regret}$
- Best of both worlds if meta-regret small!
- If $\text{regret}_{\text{FTL}} = O(\log K)$ and $\text{meta-regret} = O(\sqrt{T})$, then $\text{regret} = O(\sqrt{T})$ is **too big!**

w_t

Partial Progress

- Safe tuning: Regret $= O(\sqrt{T \ln(K)})$
- Improvement for small losses:

$$\text{Regret} = O\left(\sqrt{L^* \ln(K)}\right)$$

w_t

Partial Progress

- Safe tuning: Regret $= O(\sqrt{T \ln(K)})$
- Improvement for small losses:

$$\text{Regret} = O\left(\sqrt{L^* \ln(K)}\right)$$

variance of w_t

- Variance Bounds: $O\left(\sqrt{\sum_t v_t \ln(K)}\right)$

• Cesa-Bianchi, Mansour, Stoltz, 2007
• vE, Grünwald, Koolen, De Rooij, 2011
• De Rooij, vE, Grünwald, Koolen, 2014

w_t

Partial Progress

- Safe tuning: Regret $= O(\sqrt{T \ln(K)})$
- Improvement for small losses:

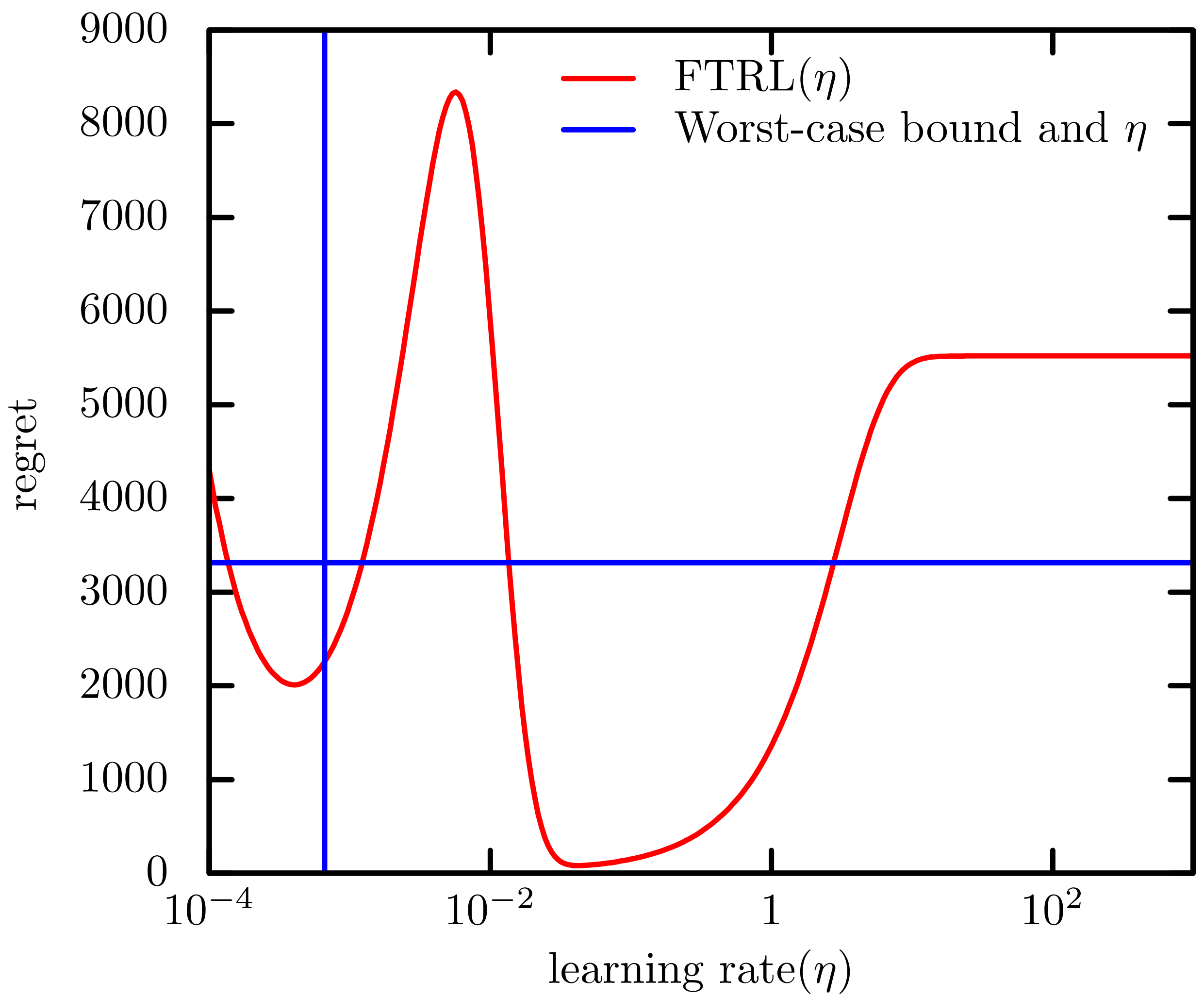
$$\text{Regret} = O\left(\sqrt{L^* \ln(K)}\right)$$

variance of w_t

- Variance Bounds: $O\left(\sqrt{\sum_t v_t \ln(K)}\right)$

$$O\left(\sqrt{\frac{L^*(T - L^*)}{T} \ln(K)}\right)$$

Regret is a Difficult Function 1/2

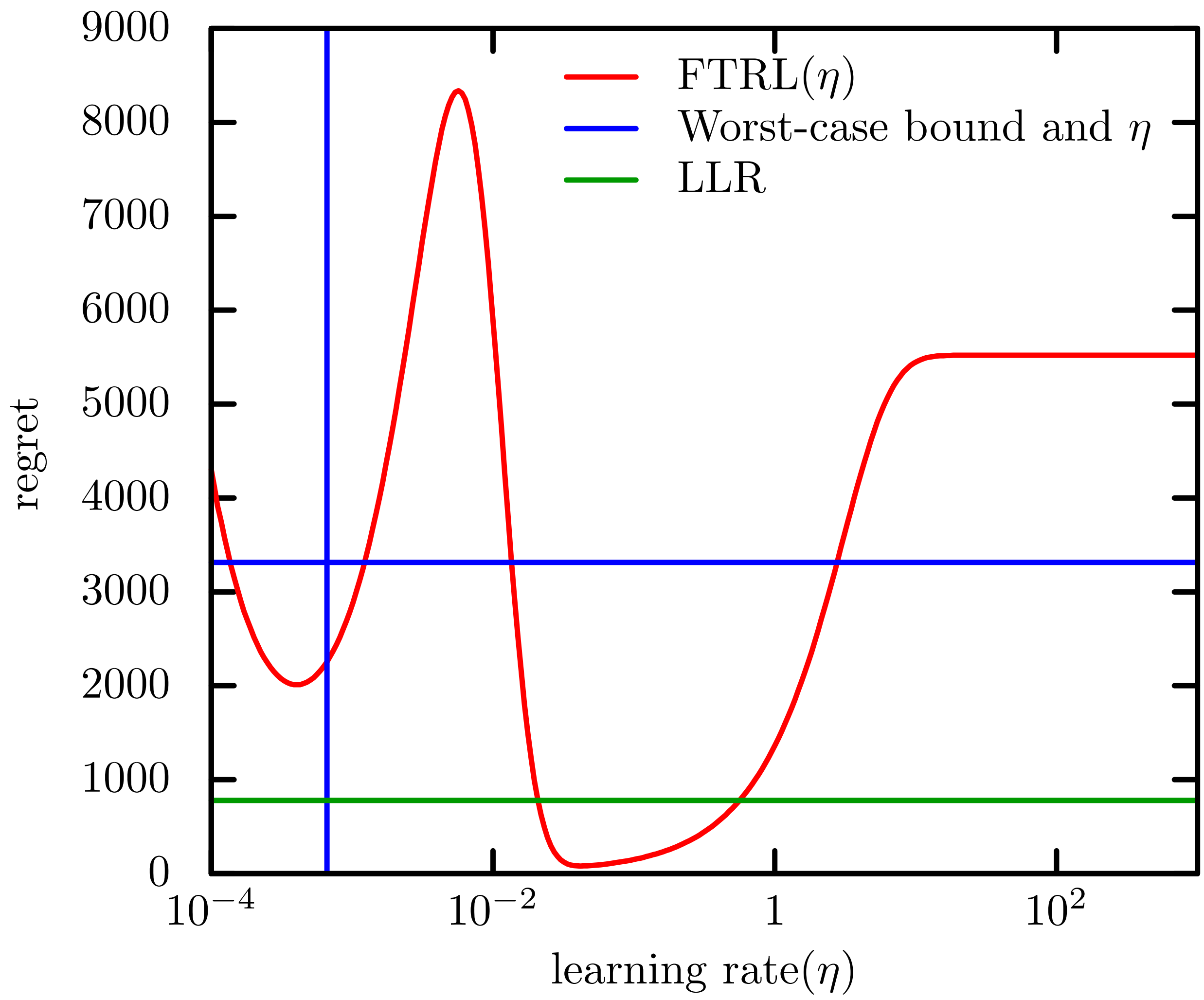


- All the previous solutions could potentially end up in wrong local minimum

Regret is a Difficult Function 2/2

- Regret as a function of T for fixed η is **non-monotonic**.
- This means some η may look very bad for a while, but end up being very good after all
- How do we see which η's are good?
- Use (best-possible) **monotonic lower-bound** per η

Learning the Learning Rate



- Koolen, vE, Grünwald, 2014

Learning the Learning Rate

- **Track performance** for grid of learning rates η
- **Switch** between them
  - Pay for switching, but not too much
- Running time as **fast** as for single fixed η
  - does not depend on size of grid

Learning the Learning Rate

Theorems:

$$\left.\begin{aligned} \text{regret} &\leq C \cdot \text{regret}_{\text{FTL}} \\ \text{regret} &\leq C \cdot \text{regretbound}_{\text{Safe}} \\ \text{regret} &\leq C \cdot \text{regretbound}_{\text{Variance}} \end{aligned}\right\}$$

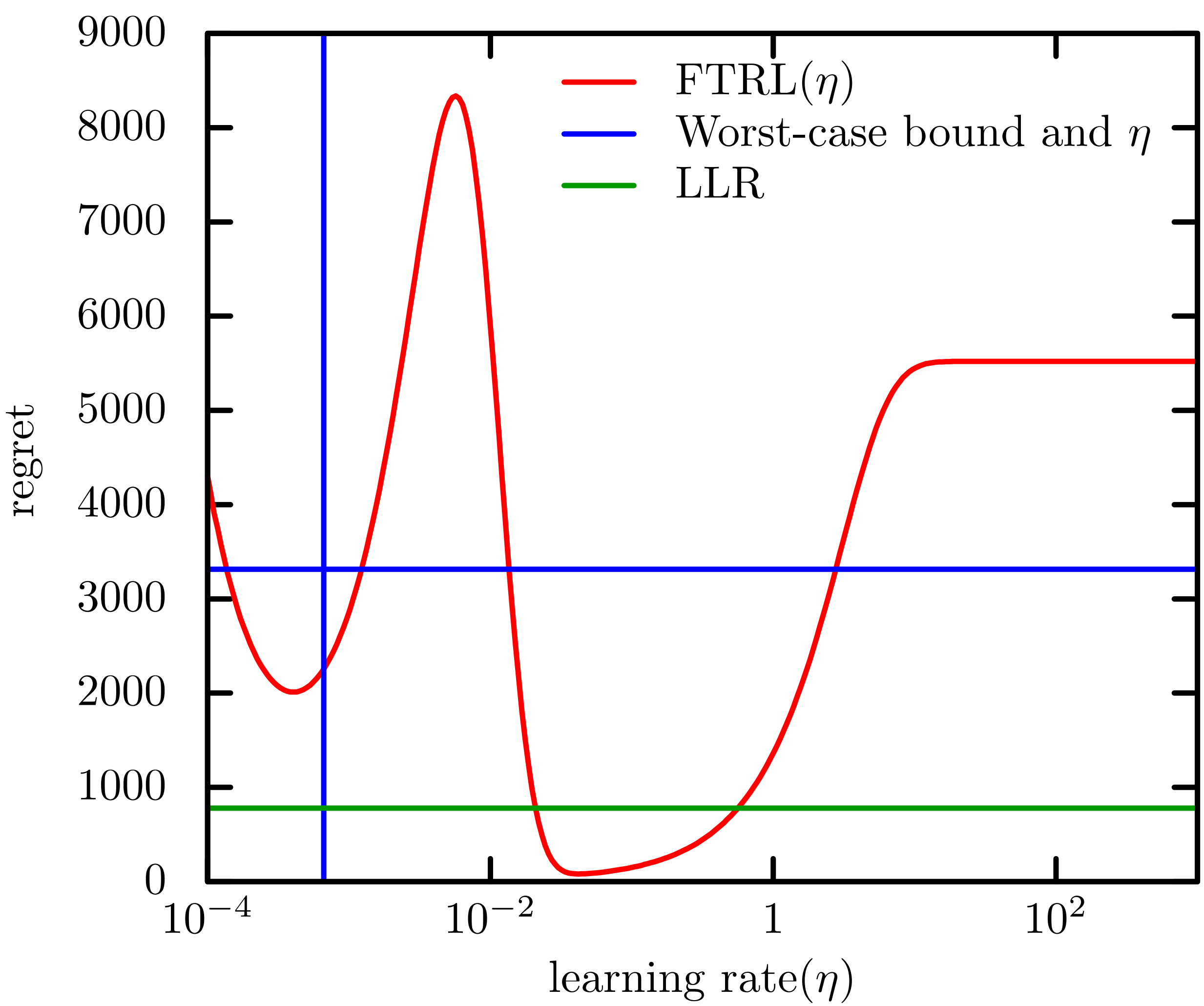
As good as all previous methods

- For **all** interesting η :

$$\text{regret} \leq F \cdot \text{regret}_\eta$$

$$F = O(\ln(K) \ln^{1+\epsilon}(T)) =$$ **polylog(K,T)**

Learning the Learning Rate



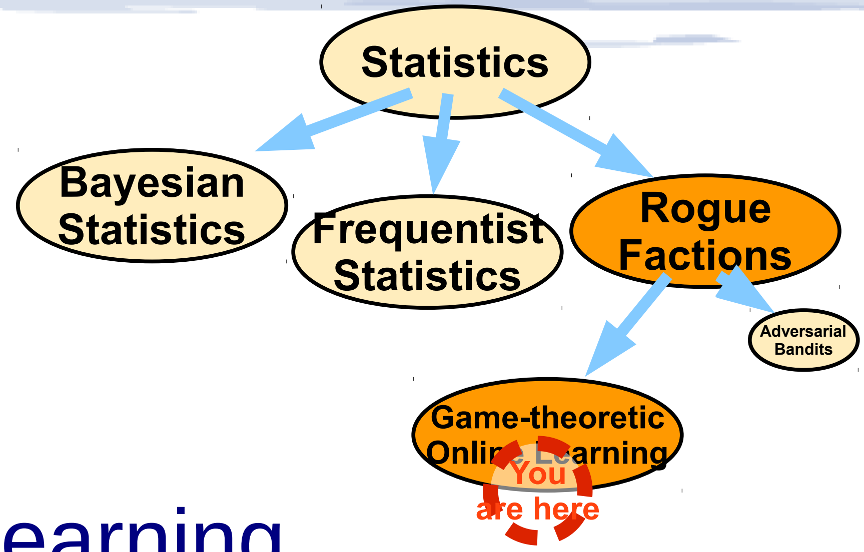
- Koolen, vE, Grünwald, 2014

The Price of Robustness

Method	Special Case of FTRL	Perfect Expert Data	IID Data	Worst-case Data	Compared to Optimal η
Follow the Leader	$\eta = \infty$ (very large)	$O(\log K)$	$O(\log K)$	$\Theta(T)$	no useful guarantees
FTRL with Worst-case Safe Tuning	$\eta = \sqrt{\frac{8 \ln K}{T}}$ (small)	$O(\sqrt{T \ln K})$	$O(\sqrt{T \ln K})$	$O(\sqrt{T \ln K})$	no useful guarantees
LLR	$\eta = \text{adaptive}$	$O(\log K)$	$O(\log K)$	$O(\sqrt{T \ln K})$	polylog(K,T) factor

Can we **adapt** to optimal eta **automatically**? **Yes!**

Summary



- Game-theoretic Online Learning
  - e.g. electricity forecasting, spam detection
- Three algorithms:
  - Halving, Follow the (Regularized) Leader
- Tuning the Learning Rate:
  - Safe tuning pays the **Price of Robustness**
  - Learning the learning rate **adapts** to the optimal learning rate **automatically**

Future Work

- So far: compete with the **best expert**
- Online Convex Optimization: compete with the **best convex combination** of experts
- Future work: extend LLR to online convex optimization

References

- Cesa-Bianchi and Lugosi. Prediction, learning, and games. 2006.
- Cesa-Bianchi, Mansour, Stoltz. Improved second-order bounds for prediction with expert advice. Machine Learning, 66(2/3):321–352, 2007.
- Devaine, Gaillard, Goude, Stoltz. Forecasting electricity consumption by aggregating specialized experts. Machine Learning, 90(2):231-260, 2013.
- Van Erven, Grünwald, Koolen and De Rooij. Adaptive Hedge. NIPS, 2011.
- De Rooij, Van Erven, Grünwald, Koolen. Follow the Leader If You Can, Hedge If You Must. Journal of Machine Learning Research, 2014.
- Koolen, Van Erven, Grünwald. **Learning the Learning Rate for Prediction with Expert Advice**. NIPS, 2014.