Benelearn, 19 June 2015

Second-order Quantile Methods for Online Sequential Prediction

Tim van Erven

Universiteit Leiden

Joint work with: **Wouter Koolen**

Berkeley University of California

QUT Queensland University of Technology

Outline

- **Prediction with Expert Advice**
  - Setting
  - Standard algorithm (and its limitations)
- Improvements
  - Either second-order bounds
  - Or quantile bounds
- How to get both improvements
- Online shortest path

Sequential Prediction with Expert Advice

- K experts **sequentially** predict data $x_1, x_2, \ldots$
- Goal: predict (almost) as well as the **best expert** on average

Sequential Prediction with Expert Advice

- K experts **sequentially** predict data $x_1, x_2, \ldots$
- Goal: predict (almost) as well as the **best expert** on average
- Applications:
  - online **classification**, e.g. spam detection
  - online convex **optimization**
  - **boosting**
  - **differential privacy**
  - predicting time series like electricity consumption or air pollution levels

Formal Setting

- Every round $t = 1, \ldots, T$:
  1. Predict **probability distribution** w_t on K experts
  2. Observe expert losses $\ell_t^1, \ldots, \ell_t^K \in [0,1]$
  3. Our expected loss is $\hat{\ell}_t = \mathop{\mathbf{E}}_{w_t(k)} [\ell_t^k]$

Formal Setting

- Every round $t = 1, \ldots, T$:
  1. Predict **probability distribution** w_t on K experts
  2. Observe expert losses $\ell_t^1, \ldots, \ell_t^K \in [0, 1]$
  3. Our expected loss is $\hat{\ell}_t = \underset{w_t(k)}{\mathbf{E}} [\ell_t^k]$
- Goal: small ***regret*** for every expert k

$$R_T^k = \sum_{t=1}^{T} \hat{\ell}_t - \sum_{t=1}^{T} \ell_t^k$$

Standard Algorithm

- **Exponential weights** with **prior** π :

$$w_t(k) = \frac{\pi(k) e^{\eta R^k_{t-1}}}{\text{normalisation}}$$

- **learning rate** η is a parameter
  - large η : aggressive learning
  - small η : conservative learning

Standard Algorithm

- **Exponential weights** with **prior** π :

$$w_t(k) = \frac{\pi(k) e^{\eta R^k_{t-1}}}{\text{normalisation}}$$

- **learning rate** η is a parameter
  - large η : aggressive learning
  - small η : conservative learning
- w_t is **gradient of potential function** $\ln \sum_k \pi(k) e^{\eta R^k_{t-1}}$

Basic Regret Guarantee

- For learning rate $\eta = \sqrt{8 \ln(K)/T}$ [Freund, Schapire, 1997]

$$R_T^k \prec \sqrt{T \ln(K)} \quad \text{for all experts k}$$

- Average regret per round **goes to 0**
- T does not measure **inherent difficulty**
- $\ln(K)$ does not count **effective nr of experts**

Outline

- Prediction with Expert Advice
  - Setting
  - Standard algorithm (and its limitations)
- **Improvements**
  - Either second-order bounds
  - Or quantile bounds
- How to get both improvements
- Online shortest path

Improvement 1: Second-order Bounds

$$R_T^k \prec \sqrt{T \ln(K)}$$

- T simply **counts** nr of rounds
- Want to replace by some measure of the **variance** in the losses
- Different proposals [Cesa-Bianchi, Mansour, Stoltz, 2007], [Hazan, Kale, 2010], [Gaillard, Stoltz, vE, 2014]

$$V_T^k := \sum_{t=1}^{T} (\hat{\ell}_t - \ell_t^k)^2$$

Improvement 1: Second-order Bounds

$$R_T^k \prec \sqrt{V_T^k \ln(K)}$$

- **Different specialized algorithms**
- **Different clever tricks to choose learning rate** adaptively over time

[Cesa-Bianchi, Mansour, Stoltz, 2007], [Hazan, Kale, 2010], [Gaillard, Stoltz, vE, 2014]

Improvement 2: Quantile Bounds

$$R_T^k \prec \sqrt{T \ln(K)}$$

- $\ln(K)$ is **nr. of bits** to identify best expert
- But suppose multiple experts $\mathcal{K} \subset \{1, \ldots, K\}$ are **all good**
- Want to replace by

$$\ln \frac{1}{\pi(\mathcal{K})}$$

for **prior** π on experts

[Chaudhuri, Freund, Hsu, 2009]

Improvement 2: Quantile Bounds

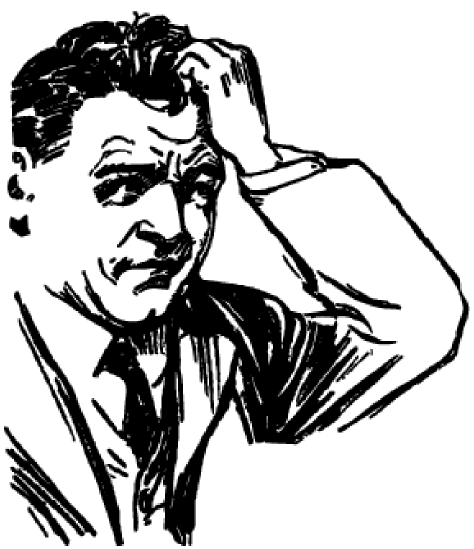
$$R_T^{\mathcal{K}} \prec \sqrt{T \ln \frac{1}{\pi(\mathcal{K})}}$$

- $R_T^{\mathcal{K}} = \min_{k \in \mathcal{K}} R_T^k$ good when all $k \in \mathcal{K}$ good
- **Specialized algorithms**
- **Clever tricks to choose learning rate** adaptively over time

[Chaudhuri, Freund, Hsu, 2009]

Both Improvements: Second-order Quantile Bounds?

$$R_T^{\mathcal{K}} \prec \sqrt{V_T^{\mathcal{K}} \ln \frac{1}{\pi(\mathcal{K})}}$$



- Different **specialized algorithms**
- Incompatible clever tricks to **tune learning rate**

Need something simple!

Outline

- Prediction with Expert Advice
  - Setting
  - Standard algorithm (and its limitations)
- Improvements
  - Either second-order bounds
  - Or quantile bounds
- **How to get both improvements**
- Online shortest path

New Algorithm

- **Exponential weights** with **prior** π:

$$w_t(k) = \frac{\pi(k) e^{\eta R^k_{t-1}}}{\text{normalisation}}$$

- 1. Incorporate **variance**:

$$w_t(k) = \frac{\pi(k) e^{\eta R^k_{t-1} - \eta^2 V^k_{t-1}}}{\text{normalisation}}$$

New Algorithm

- **Exponential weights** with **prior** π:

$$w_t(k) = \frac{\pi(k) e^{\eta R_{t-1}^k}}{\text{normalisation}}$$

- 2. Add **prior on learning rates**:

$$w_t(k) = \frac{\int \gamma(\eta) \pi(k) e^{\eta R_{t-1}^k - \eta^2 V_{t-1}^k} \eta \, \mathrm{d}\eta}{\text{normalisation}}$$

New Regret Bound

Thm. **Any** $\pi(k)$, **right choice** of $\gamma(\eta)$ achieves

$$R_T^{\mathcal{K}} \prec \sqrt{V_T^{\mathcal{K}} \left(\ln \frac{1}{\pi(\mathcal{K})} + \ln \ln T \right)} \quad \text{for all } \mathcal{K}$$

- **Averages** under the prior **instead of worst** in $\mathcal{K}$:

$$R_T^{\mathcal{K}} = \mathop{\mathbf{E}}_{\pi(k|\mathcal{K})} [R_T^k] \qquad V_T^{\mathcal{K}} = \mathop{\mathbf{E}}_{\pi(k|\mathcal{K})} [V_T^k]$$

New Regret Bound

Thm. **Any** $\pi(k)$, **right choice** of $\gamma(\eta)$ achieves

$$R_T^{\mathcal{K}} \prec \sqrt{V_T^{\mathcal{K}} \left(\ln \frac{1}{\pi(\mathcal{K})} + \ln \ln T \right)} \quad \text{for all } \mathcal{K}$$

- **Averages** under the prior **instead of worst** in $\mathcal{K}$:

$$R_T^{\mathcal{K}} = \mathop{\mathbf{E}}_{\pi(k|\mathcal{K})}[R_T^k] \qquad V_T^{\mathcal{K}} = \mathop{\mathbf{E}}_{\pi(k|\mathcal{K})}[V_T^k]$$

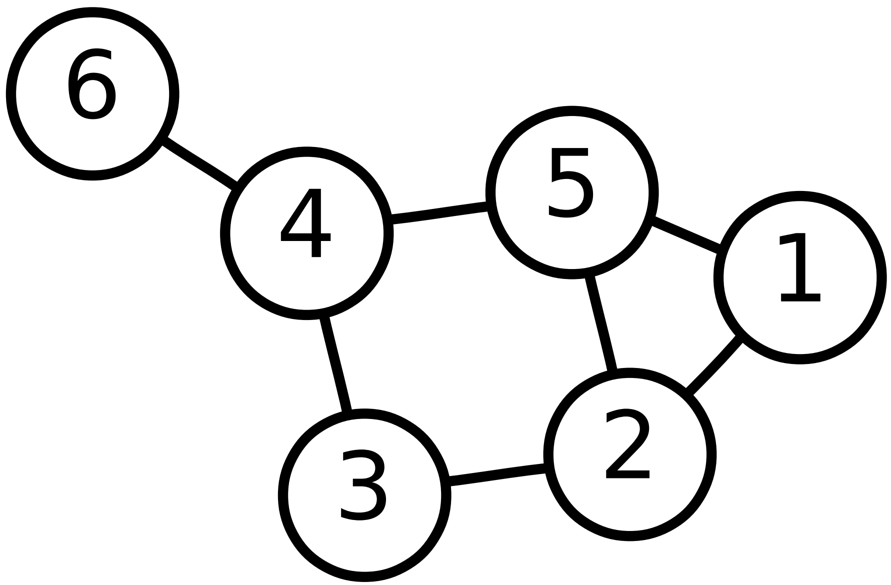
- If T = **age of universe** in μs: $\ln \ln T < 4$

Outline

- Prediction with Expert Advice
  - Setting
  - Standard algorithm (and its limitations)
- Improvements
  - Either second-order bounds
  - Or quantile bounds
- How to get both improvements
- **Online shortest path**

Learn Shortest Path in a Graph

- Every round t, **each edge incurs loss**
- v : **best distribution on paths**
- Goal: learn v



K edges

$$R_T^v \prec \sqrt{V_T^v \left(\text{complexity}(v) + K \ln \ln T\right)}$$

Summary

- Improvements over standard exponential weights algorithm:
  - Either **second-order bounds**
  - Or **quantile bounds**
- New algorithm gets **both improvements**
  - Surprisingly simple generalization of exponential weights
- Extension to online shortest path
  (for other combinatorial problems, come to poster)