Faster Recalibration of an Online Predictor via Approachability

Princewill Okoroafor[1] Robert Kleinberg[1] Wen Sun[1]

[1] Cornell University

Setup

Online Recalibration Problem

Each of $t \in \{1, \ldots, T\}$ rounds:

- **Nature** selects context $x_t \in \mathbb{R}^d$
- Oracle predicts $q_t \in [0,1]$
- **Algorithm**
  - observes x_t, q_t
  - predicts $p_t \in [0,1]$
- **Nature** reveals $y_t \in \{0,1\}$

Learning Objectives

- Low-regret with respect a fixed proper scoring rule:
  $\frac{1}{T}\sum_{t=1} S(p_t, y_t) - S(q_t, y_t)$
- ℓ_1 –Calibration Error:
  $\sum_{v \in \{p_1,\ldots,p_T\}} \left|\frac{1}{T}\sum_{t=1}(y_t - v) \cdot \mathbb{I}[p_t = v]\right|$

Model	Calibration	Regret	Method
Online calibration [Abernethy et al 2011]	$O(T^{\frac{2}{3}})$	N/A	Blackwell's approachability
Online recalibration [Kuleshov et al 2017]	$O(T^{\frac{3}{4}})$	$O(T^{\frac{3}{4}})$	Internal regret minimization
Online recalibration [this work]	$O(T^{\frac{2}{3}})$ ← tunable tradeoff →	$O(T^{\frac{1}{2}})$	Blackwell's approachability theorem

Overview

Challenges:

- Calibration might require algorithm to deviate from oracle predictions
- Algorithm must achieve no-regret i.e deviations should only improve predictions

Our Work: We show an algorithm with a tunable linear tradeoff that achieves calibration error $O(T^{2x})$ and regret $O(T^{2x})$ for any $x \in \left[\frac{1}{3}, \frac{2}{5}\right]$

Solution Outline

Reduction to Two Player Vector Payoff Game [Abernethy et al]

- **Player 1** selects sample $y_t \in [0,1]$
- **Player 2** chooses $w_t \in \Delta_{m+1}$
- Payoff vector $u(w_t, q_t, y_t)$

$$\begin{pmatrix} w_t(0)\left(y_t - \frac{0}{m}\right) \\ \vdots \\ w_t(m)\left(y_t - \frac{m}{m}\right) \\ \sum_{i \in [m+1]} w_t(i)\left(S\left(\frac{i}{m}, y_t\right) - S(q_t, y_t)\right) \end{pmatrix}$$

Goal: Is there a strategy for player 2 that guarantees that the average payoff vector converges to the set?

$$\left\{(x, z) \in \mathbb{R}^{m+1} \times \mathbb{R} \mid \|x\|_1 \le \frac{1}{m}, z \le \frac{4L_s}{m^2}\right\}$$

Constructing a Halfspace Oracle

Given halfspace $\theta \in K$, we must construct $w_t \in \Delta_{m+1}$ such that $\langle \theta, u(w_t, q_t, y_t) \rangle \le \epsilon$

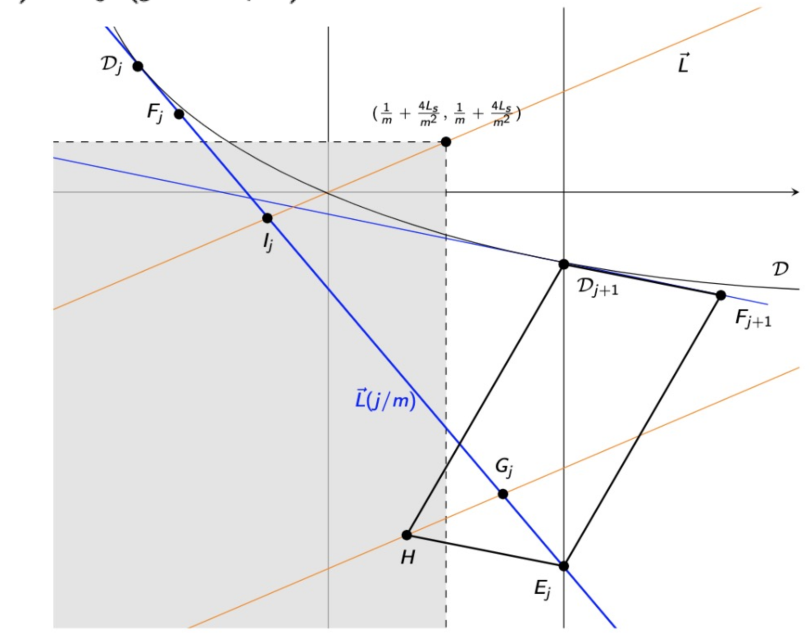
Halfspace Oracle

$$F_i = \begin{bmatrix} f(i,0) \\ f(i,1) \end{bmatrix} \quad \text{where} \quad f(i,y) = \theta_i\left(\frac{i}{m} - y\right) + \theta_{m+2}\left[S\left(\frac{i}{m}, y\right) - S(q_t, y)\right]$$

$$w_t(j) = \frac{f(j+1,1) - f(j+1,0)}{f(j,0) - f(j+1,0) - f(j,1) + f(j+1,1)}$$

$$w_t(j+1) = \frac{f(j,0) - f(j,1)}{f(j,0) - f(j+1,0) - f(j,1) + f(j+1,1)}$$

Find index j where F_j is in the third quadrant or crosses from second to fourth quadrant



Algorithm 1 Online Recalibration Algorithm

Input: some natural number $m \ge \sqrt{4L_S}$
Initialize: $\boldsymbol{\theta}_1 = \mathbf{0}, \mathbf{w}_1 \in \Delta_{m+1}$
for $t = 1, \ldots, T$ **do**
  Observe q_t from black-box prediction oracle
  Sample $i_t \sim \mathbf{w}_t$, predict $p_t = \frac{i_t}{m}$, observe y_t
  Set $l_t := -\boldsymbol{\ell}_t(\mathbf{w}_t, y_t)$
  Query learning algorithm: $\boldsymbol{\theta}_{t+1} \leftarrow \mathrm{OGD}(\boldsymbol{\theta}_t | l_t)$
  // Online Gradient Descent step
  Query halfspace oracle: $\mathbf{w}_{t+1} \leftarrow \mathrm{Approach}(\boldsymbol{\theta}_{t+1})$
  // Obtain $\mathbf{w}_{t+1} \in \Delta_{m+1}$ from $\boldsymbol{\theta}_{t+1}$
end for