Calibrated Confidence Measures With Guarantees

Volodymyr Kuleshov, Stefano Ermon

Department of Computer Science, Stanford University

"It ain't what you don't know that gets you into trouble. It's what you know for sure that just ain't so."
– Mark Twain

Motivation

In healthcare, assessing forecast confidence is often as important as achieving high accuracy, e.g.:

- How certain are we that this patent has cancer?

This work studies *calibrated* confidence estimates

- For *online* prediction problems
- With *guaranteed* accuracy (even vs. adversary).

Calibration

We assess confidence via *calibrated* probabilities: e.g., if forecaster $F(x)$ predicts 70% chance of rain on some days, it should rain on 70% of these days.

Formally, given forecasts $(p_t)_{t=1}^T \in [0,1]$ and outcomes $(y_t)_{t=1}^T \in \{0,1\}$, the *calibration error* is

$$C_T = \hat{\mathbb{E}}\left[\left(\rho_T(i/N) - \frac{i}{N}\right)^2\right]$$

where $\rho_T(p) = \hat{\mathbb{E}}[y \mid p]$ is the frequency at which event $y = 1$ occurred over the times when we predicted p.

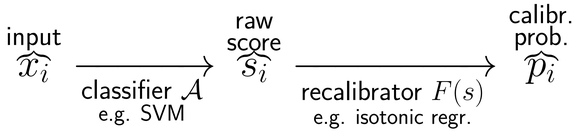
Calibration in Online Learning

Current methods are motivated by game theory:

- Task: predicting binary outcome y_{t+1} from past outcomes $y_{1:t}$.
- Surprising result: calibration algorithms for adversarial y_t
- Idea: model as a game with $N+1$ actions and losses $(\frac{i}{N} - y_t)^2$; minimize internal regret.

Recalibration in Batch Learning

Popular methods like Platt scaling or isotonic regression remap raw scores into probabilities.



F is trained on a calibration set labeled with y_i.

Shortcomings of Existing Methods

Existing online methods:

- Do not allow covariates x_t (e.g. genomic data)
- Are not *sharp*: predicting 0.5 on 0101... is okay.

Existing batch methods:

- Are heuristics with no guarantees (especially if underlying data distribution changes)
- Need separate calibration set for batch training

Online Recalibration

- We transform raw p_t^F from uncalibrated forecaster F into calibrated p_t.
- We maintain the accuracy (regret) and convergence of F (e.g. learning rate adaptivity)

Formal Setup

Formally, at every step $t = 1, ..., T$:

- Nature chooses $(x_t, y_t) \in \mathbb{R}^d \times \{0,1\}$, reveals x_t.
- Forecaster F predicts $p_t^F = \sigma(w_{t-1} \cdot x) \in [0,1]$.
- A recalibration algorithm A produces a calibrated probability $p_t = A(p_t^F) \in [0,1]$.
- Nature reveals y_t; F incurs loss of $\ell(p_t, y_t)$.
- F updates w_t; A updates itself based on y_t.

Algorithm

Idea:

- Divide p_t^F into M buckets $[\frac{j}{M}, \frac{j+1}{M})$.
- Train M independent instances of online calibration algorithm F^{cal} on each bucket.
- At new p_t^F, call its bucket's F^{cal}.

1: Let $\mathcal{I} = \{[0, \frac{1}{M}), [\frac{1}{M}, \frac{2}{M}), ..., [\frac{M-1}{M}, 1]\}$ be a set of intervals that partition $[0,1]$.
2: Let $\mathcal{F} = \{F_j^{\text{cal}} \mid i = 0, ..., M-1\}$ be a set of M independent instances of F^{cal}.
3: **for** $t = 1, ..., T$: **do**
4: Let $[\frac{j}{M}, \frac{j+1}{M})$ be the interval containing p_t^F.
5: Let p_t be the forecast of F_j^{cal}. Output p_t.
6: Observe y_t and pass it to $F_j^{\text{cal}} \in \mathcal{F}$.

Guarantees

Suppose that:

- Subroutine F^{cal} is ϵ-calibrated.
- Number of buckets $M > 1/\epsilon$

Then, the recalibrated p_t:

- Are ϵ-calibrated
- Are not worse than the raw p_t^F: the regret

$$\lim_{T\to\infty}\left(\frac{1}{T}\sum_{t=1}^{T}(y_t - p_t)^2 - \frac{1}{T}\sum_{t=1}^{T}(y_t - p_t^F)^2\right) < 4/N,$$

can be made arbitrarily small.

Intuition:

- Calibration $\implies$ low regret $\implies$ we do as well as predicting constant $\frac{j}{M} \approx p_t^F$ on bucket j.
- Since we do as well as p_t^F on each bucket's predictions, we do as well on average.
- We are similarly calibrated because each F_j^{cal} is calibrated

Convergence analysis

Our algorithm has $O(1/\epsilon)$ space and $O(1/\sqrt{\epsilon})$ convergence rate overhead over subroutine F_j^{cal}:

	F^{cal} subroutine	Our method
Time/Iter.	$O(\log(\frac{1}{\epsilon}))$	$O(\log(\frac{1}{\epsilon}))$
Memory	$O(\frac{1}{\epsilon})$	$O(\frac{1}{\epsilon^2})$
C_T conv. rate	$O(\frac{1}{\epsilon\sqrt{\epsilon T}})$	$O(\frac{1}{\epsilon\sqrt{T}})$

Extensions

Recalibration of multiple forecasters. Reduction to standard case:

- Run regret-minimization algorithm on the $(F_k)_{k=1}^K$.
- Recalibrate output: ensures low regret w.r.t. all F_k

K classes. Extension of partitioning idea:

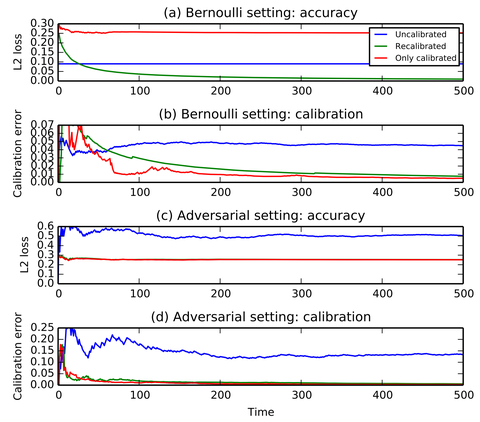
- Partition space into grid with $\frac{1}{\epsilon^K}$ cells
- Overhead of $O(\frac{1}{\epsilon^K})$ over subroutine
- Cannot do better: problem PPAD-complete

Highlights

- Recalibration of any online forecaster F on adversarial input.
- Recalibrated forecasts have approximately the same ℓ_2 error as raw forecasts.
- Tight finite-time bounds on calibration error.
- Performs well on real-world diabetes dataset.

Synthetic experiments

Top: $z_t \sim \text{Ber}(0.5)$; $p_t^F = 0.2$ if $z_t = 0$, else $p_t^F = 0.8$. Our algorithm learns to perfectly predict z_t.



Bottom: on adversarial input and random p_t^F, we are calibrated and match standard recalibration.

Medical diagnosis experiment

We effectively recalibrate SVM predictions of T1 diabetes from WTCCC genotypes.