Reliable Confidence Estimation via Online Learning

Volodymyr Kuleshov
Stanford University
Stanford, CA 94305
kuleshov@cs.stanford.edu

Stefano Ermon
Stanford University
Stanford, CA 94305
ermon@cs.stanford.edu

Abstract

Assessing uncertainty within machine learning systems is an important step towards ensuring their safety and reliability. Existing uncertainty estimation techniques may fail when their modeling assumptions are not met, e.g. when the data distribution differs from the one seen at training time. Here, we propose techniques that assess a classification algorithm’s uncertainty via calibrated probabilities (i.e. probabilities that match empirical outcome frequencies in the long run) and which are guaranteed to be reliable (i.e. accurate and calibrated) on out-of-distribution input, including input generated by an adversary. Our methods admit formal bounds that can serve as confidence intervals, and process data in an online manner, which obviates the need for a separate calibration dataset. We establish theoretical guarantees on our methods’ accuracies and convergence rates, and we validate them on two real-world problems: question answering and medical diagnosis from genomic data.

1 Introduction

Assessing uncertainty within machine learning systems is an important step towards ensuring their safety and reliability. Indeed, in many applications of machine learning — including medical diagnosis [1], natural language understanding [2], speech recognition [3], weather forecasting [4], econometric analysis [5] — assessing confidence can be as important as obtaining high accuracy. This work explores robust, online confidence estimation methods for classification problems.

An important limitation of existing methods is the assumption that data is sampled i.i.d. from a distribution \( P(x,y) \); when test-time data is distributed according to a different \( P^* \), these methods may become overly confident on incorrect predictions. Here, we introduce new, robust uncertainty estimation algorithms guaranteed to produce reliable confidence estimates on out-of-distribution input, including input generated by an adversary. Our algorithms are based on online learning techniques and, unlike earlier methods, they do not require data to be set aside solely for uncertainty estimation.

We focus primarily on the classification setting, where the most common and natural way of measuring an algorithm’s uncertainty is via calibrated probability estimates that match the true empirical frequencies of an outcome. For example, if an algorithm predicted a 60% chance of rain 100 times in a given year, we would say that its forecast was calibrated if it rained on about 60 of those 100 days.

Calibrated confidence estimates are typically constructed via recalibration, using methods such as Platt scaling or isotonic regression. In the context of binary classification, these methods reduce recalibration to a one-dimensional regression problem that, given data \( (x_i, y_i)_{i=1}^n \), trains a model \( g(s) \) (e.g. logistic regression) to predict probabilities \( p_i = g(s_i) \) from uncalibrated scores \( s_i = h(x_i) \) produced by a classifier \( h \) (e.g. SVM margins). Fitting \( g \) is equivalent to performing density estimation targeting \( P(Y = 1|h(X) = s_i) \) and hence may fail on out-of-distribution testing data.

The methods we introduce in this work are instead based on calibration techniques developed in the literature on online learning in mathematical games [6,7]. These classical methods are not directly applicable to standard prediction tasks in their current form. For one, they do not admit covariates \( x_i \).
We also demonstrate empirically on two real-world problems that our method converges quickly and makes no assumption on the data generating distribution, and of batch probability recalibration, which admits covariates and guarantees forecast sharpness.

As a concrete motivating example, consider a medical system that diagnoses a long stream of patients indexed by \( t = 1, 2, ..., \) outputting a disease risk \( p_t \in \{0, 1\} \) for each patient based on their medical record \( x_t \). Provably calibrated probabilities in this setting may be helpful for making informed policy decisions (e.g. by providing guaranteed upper bounds on the number of patients readmitted after a discharge) and may be used to communicate risks to patients in a more intuitive way. This setting is also inherently online, since patients are typically observed one at a time, and may not satisfy the i.i.d. assumption due to e.g. seasonal disease outbreaks.

More formally, our paper makes the following contributions:

- We formulate a new problem called online recalibration, which requires producing calibrated outcome probabilities on potentially adversarial input, while retaining the predictive power of a given baseline uncalibrated forecaster.
- We propose a meta-algorithm for online recalibration that uses a classical online calibration algorithm \( \mathcal{A} \) as a black box.
- We show that our technique can recalibrate the forecasts of any existing classifier at the cost of an \( O(1/\sqrt{T}) \) overhead in the convergence rate of \( \mathcal{A} \), where \( \epsilon > 0 \) is the desired level of accuracy.
- Surprisingly, both online and standard batch recalibration (e.g. Platt scaling) may be performed only when accuracy is measured using specific loss functions; our work characterizes the losses which admit a recalibration procedure in both the online and batch settings.

We also demonstrate empirically on two real-world problems that our method converges quickly and retains an accuracy comparable to that of the input baseline uncalibrated forecaster.

## 2 Background

Below, we will use \( \mathbb{1}_E \) denote the indicator function of \( E \). \([N]\) and \([N]_0\) to (respectively) denote the sets \( \{1, 2, ..., N\} \) and \( \{0, 1, 2, ..., N\} \), and \( \Delta_d \) to denote the \( d \)-dimensional simplex.

### 2.1 Learning with Expert Advice

Learning with expert advice [8] is a special case of the general online optimization framework [9] that underlies online calibration algorithms. At each time \( t = 1, 2, ..., \) the forecaster \( F \) receives advice from \( N \) experts and chooses a distribution \( w_t \in \Delta_{N-1} \) over their advice. Nature then reveals an outcome \( y_t \) and \( F \) incurs an expected loss of \( \sum_{t=1}^{N} w_t \ell(i, y_t) \), where \( \ell(i, y_t) \) is the loss under expert \( i \)’s advice. Performance in this setting is measured using two notions of regret.

**Definition 1.** The external regret \( R_{e}^T \) and the internal regret \( R_{i}^T \) are defined as

\[
R_{e}^T = \sum_{t=1}^{T} \bar{\ell}(p_t, y_t) - \min_{i \in [N]} \sum_{t=1}^{T} \ell(i, y_t), \quad R_{i}^T = \max_{i, j \in [N]} \sum_{t=1}^{T} p_{t,i} (\ell(i, y_t) - \ell(j, y_t)),
\]

where \( \bar{\ell}(p, y) = \sum_{i=1}^{N} p_i \ell(i, y) \) is the expected loss.

External regret measures loss with respect to the best fixed expert, while internal regret is a stronger notion that measures the gain from retrospectively switching all the plays of action \( i \) to \( j \). Both definitions admit algorithms with sublinear, uniformly bounded regret.

In this paper, we will be particularly interested in proper losses \( \ell \), whose expectation over \( y \) is minimized by the probability corresponding to the average \( y \).

**Definition 2.** A loss \( \ell(p, y) : [0, 1] \times \{0, 1\} \rightarrow \mathbb{R}_+ \) is proper if \( p \in \arg \min_{q} \mathbb{E}_{y \sim \text{Ber}(p)} \ell(y, q) \) \( \forall p \).
Examples of proper losses include the L2 loss $\ell_2(y, p) = (y - p)^2$, the log-loss $\ell_{\log}(y, p) = y \log(p) + (1 - y) \log(1 - p)$, and the the misclassification loss $\ell_{mc}(y, p) = (1 - y)\cdot \mathbb{1}_{p < 0.5} + y\cdot \mathbb{1}_{p \geq 0.5}$. Counter-examples include the L1 and the hinge losses.

### 2.2 Calibration in Online Learning

Intuitively, calibration means that the true and predicted frequencies of an event should match. For example, if an algorithm predicts a 60% chance of rain 100 times in a given year, then we should see rain on about 60 of those 100 days. More formally, let $F^{\text{cal}}$ be a forecaster making predictions in the set $\{\frac{i}{N} \mid i = 0, \ldots, N\}$, where $1/N$ is called the resolution of $F^{\text{cal}}$; consider the quantities

$$\rho_T(p) = \frac{\sum_{t=1}^{T} y_t \cdot \mathbb{1}_{p_t = p}}{\sum_{t=1}^{T} \mathbb{1}_{p_t = p}} \quad C_T^p = \sum_{i=0}^{N} \left| \rho_T\left(\frac{i}{N}\right) - \frac{i}{N} \right| \left( \frac{1}{T} \sum_{t=1}^{T} \mathbb{1}_{\{p_t = \frac{i}{N}\}} \right).$$

The term $\rho_T(p)$ denotes the frequency at which event $y = 1$ occurred over the times when we predicted $p$. Our intuition was that $\rho_T(p)$ and $p$ should be close to each other; we capture this using the notion of calibration error $C_T^p$ for $p \geq 1$; this corresponds to the weighted $\ell_p$ distance between the $p$-calibration probabilities $\frac{i}{N}$; typically one assumes that $p = 1$ or $p = 2$. To simplify notation, we will use the term $C_T$ when the exact $p$ is unambiguous.

**Definition 3.** We say that $F^{\text{cal}}$ is an $(\epsilon, \ell_p)$-calibrated algorithm with resolution $1/N$ if a.s. we have

$$\lim_{T \to \infty} \sup_{p \geq 1} C_T^p \leq \epsilon.$$

There exists a vast literature on calibration in the online setting [8] which is primarily concerned with constructing calibrated predictions $p_t \in [0, 1]$ of a binary outcome $y_t \in \{0, 1\}$ based solely on the past sequence $y_1, \ldots, y_{t-1}$. Surprisingly, this is possible even when the $y_t$ are chosen adversarially by reducing the problem to internal regret minimization relative to $N + 1$ experts with losses $(y_t - i/N)^2$ and proposed predictions $i/N$ for $i \in [N]_0$. See Chapter 4 in Cesa-Bianchi and Lugosi [8] for details.

### 3 Online Recalibration

Unfortunately, existing online calibration methods are not directly applicable in real-world settings. For one, they do not take into account covariates $x_t$ that might be available to improve the prediction of $y_t$. As a consequence, they cannot produce accurate forecasts: for example, they would constantly predict 0.5 on a sequence 01010... formed by alternating 0s and 1s.

To address these shortcomings, we define here a new problem called online recalibration, in which the task is to transform a sequence of uncalibrated forecasts $p_t^F$ into predictions $p_t$ that are calibrated and almost as accurate as the original $p_t^F$. The forecasts $p_t^F$ may come from any existing machine learning system $F$; our methods treat it as a black box and preserve its favorable convergence properties.

Formally, we define the online recalibration task as a generalization of the classical online optimization framework [2][8]. At every step $t = 1, 2, \ldots$:

1. Nature reveals features $x_t \in \mathbb{R}^d$.
2. Forecaster $F$ predicts $p_t^F = \sigma(w_{t-1} \cdot x) \in [0, 1]$.
3. A recalibration algorithm $A$ produces a calibrated probability $p_t = A(p_t^F) \in [0, 1]$.
4. Nature reveals label $y_t \in \{0, 1\}$; $F$ incurs loss of $\ell(p_t, y_t)$, where $\ell : [0, 1] \times \{0, 1\} \to \mathbb{R}^+$ is convex in $p_t$ for all $y_t$.
5. $F$ chooses $w_{t+1}$; $A$ updates itself based on $y_t$.

In the medical diagnosis example, $x_t$ represents medical or genomic features for patient $t$; we use feature weights $w_t$ to predict the probability $p_t^F$ that the patient is ill; the true outcome is encoded by $y_t$. We would like $A$ to produce $p_t$ that are accurate and well-calibrated in the following sense.

**Definition 4.** We say that $A$ is an $(\epsilon, \ell_{\text{cal}})$-accurate online recalibration algorithm for the loss $\ell_{\text{acc}}$ if (a) the forecasts $p_t = A(p_t^F)$ are $(\epsilon, \ell_{\text{cal}})$-calibrated and (b) the regret of $p_t$ with respect to $p_t^F$ is a.s. small in terms of $\ell_{\text{acc}}$.

$$\lim_{T \to \infty} \sup_{p_t} \frac{1}{T} \sum_{t=1}^{T} \left( \ell_{\text{acc}}(p_t, y_t) - \ell_{\text{acc}}(p_t^F, y_t) \right) \leq \epsilon.$$ (3)
4 Algorithms for Online Recalibration

Next, we propose an algorithm for performing online probability recalibration: we refer to our approach as a meta-algorithm because it repeatedly invokes a regular online calibration algorithm as a black-box subroutine. Algorithm 1 outlines this procedure.

At a high level, Algorithm 1 partitions the uncalibrated forecasts $p^F_t$ into $M$ buckets/intervals $\mathcal{I} = \{[0, \frac{1}{M}), [\frac{1}{M}, \frac{2}{M}), ..., [\frac{M-1}{M}, 1]\}$; it trains an independent instance of $F_{\text{cal}}$ on the data $\{p^F_t, y_t \mid p^F_t \in I_j\}$ belonging to each bucket $I_j \in \mathcal{I}$; at prediction time, it calls the instance of $F_{\text{cal}}$ associated with the bucket of the uncalibrated forecast $p^F_t$. Algorithm 1 works because a calibrated predictor is at least as accurate as any constant predictor; in particular, each subroutine $F_{\text{cal}}^j$ is at least as accurate as the prediction $\frac{j}{M}$, which also happens to be approximately $p^F_t$ when $F_{\text{cal}}^j$ was called. Thus, each $F_{\text{cal}}^j$ is as accurate as its input sequence of $p^F_t$. One can then show that if each each $F_{\text{cal}}^j$ is accurate and calibrated, then so is their aggregate, Algorithm 1.

The rest of this section provides a formal version of this argument; due to space limitations, we defer most of our full proofs to the appendix.

4.1 Calibration and Accuracy of the Online Recalibration Procedure

Notation. We define the calibration error of $F_{\text{cal}}^j$ and of Algorithm 1 at $i/N$ as (respectively)

$$C^{(j)}_{T,i} = \left| \rho_T^{(j)}(i/N) - \frac{i}{N} \right| p \left( \frac{1}{T} \sum_{t=1}^{T} \mathbb{1}_{\rho_T^{(j)}(i/N) = t} \right) \quad C_T = \left| \rho_T(i/N) - \frac{i}{N} \right| p \left( \frac{1}{T} \sum_{t=1}^{T} \mathbb{1}_{\rho_T(i/N) = t} \right),$$

where $\mathbb{1}_{\cdot} = \mathbb{1}\{p_t = i/N\}$. Terms marked with a $(j)$ denote the restriction of the usual definition to the input of subroutine $F_{\text{cal}}^j$ (see the appendix for details). We may write the calibration losses of $F_{\text{cal}}^j$ and Algorithm 1 as $C^{(j)}_T = \sum_{i=0}^{N} C^{(j)}_{T,i}$ and $C_T = \sum_{i=0}^{N} C_{T,i}$.

Assumptions. In this section, we will assume that the subroutine $F_{\text{cal}}$ used in Algorithm 1 is $(\epsilon, \ell_1)$-calibrated and that $C^{(j)}_T \leq R_T + \epsilon$ uniformly (for $\ell_1 = o(1)$ as $T \rightarrow \infty$; $T$ is the number of calls to instance $F_{\text{cal}}^j$). This also implies $\ell_2$-calibration (by continuity of $\ell_2$), albeit with different rates $R_T$ and a different $\epsilon$. Abernethy et al. [7] introduce $(\epsilon, \ell_1)$-calibrated $F_j$. We also provide proofs for the $\ell_2$ loss in the appendix.

Crucially, we assume that the loss $\ell$ used for measuring accuracy is proper and bounded with $\ell(\cdot, i/N) < B$ for $i \in [N]$; since the set of predictions is finite, this is a mild requirement. Finally, we make additional continuity assumptions on $\ell$ in Lemma 2.

Recalibration with proper losses. Surprisingly, not every loss $\ell$ admits a recalibration procedure. Consider, for example, the following continuously repeating sequence 001001001... of $y_t$'s. A calibrated forecaster must converge to predicting 1/3 (a constant prediction) with an $\ell_1$ loss of $\approx 0.44$; however predicting 0 for all $t$ has an $\ell_1$ loss of 1/3 < 0.4. Thus we cannot recalibrate this sequence and also remain equally accurate under the $\ell_1$ loss. The same argument also applies to batch recalibration (e.g. Platt scaling): we only need to assume that $y_t \sim \text{Ber}(1/3)$ i.i.d.

However, recalibration is possible for a very large class of proper losses. Establishing this fact will rely on the following key technical lemma.
Lemma 1. If \( \ell \) is a bounded proper loss, then an \((\epsilon, \ell_1)\)-calibrated \( F^{\text{cal}} \) a.s. has a small internal regret w.r.t. \( \ell \) and satisfies uniformly over time \( T \) the bound
\[
R_T^{\text{int}} = \max_{i,j} \sum_{t=1}^{T} \mathbb{I}_{p_{t} = i/N} (\ell(i/N, y_t) - \ell(j/N, y_t)) \leq 2B(R_T + \epsilon).
\] (4)

According to Lemma 1, if a set of predictions is calibrated, then we never want to retrospectively switch to predicting \( p_2 \) at times when we predicted \( p_1 \). Intuitively, this makes sense: if predictions are calibrated, then \( p_1 \) should minimize the total (or average) loss \( \sum_{t=1}^{T} \ell(p_t, y_t) \) over the times \( t \) when \( p_1 \) was predicted at least better than \( p_2 \). However, our \( \ell_1 \) counter-example above shows that this intuition does not hold for every loss; we need to explicitly enforce our intuition, which amounts to assuming that \( \ell \) is proper, i.e. that \( p \in \arg \min_q \mathbb{E}_{y \sim y \sim p(q)} \ell(y, q) \).

Accuracy and calibration. An important consequence of Lemma 1 is that a calibrated algorithm has vanishing regret relative to any fixed prediction (since minimizing internal regret also minimizes external regret). Using this fact, it becomes straightforward to establish that Algorithm 1 is at least as accurate as the baseline forecaster \( F \).

Lemma 2 (Recalibration preserves accuracy). Consider Algorithm 1 with parameters \( M \geq N > 1/\epsilon \) and let \( \ell \) be a bounded proper loss for which

1. \( \ell(p_t, y_t) \leq \ell(y_t, j/M) + B/M \) whenever \( p_t \in [j/M, (j+1)/M) \);
2. \( \ell(p_t, y_t) \leq \ell(y_t, j/N) + B/N \) whenever \( p_t \in [j/N, (j+1)/N) \);

Then the recalibrated \( p_t \) a.s. have vanishing \( \ell \)-loss regret relative to \( p_t^{\ell} \) and we have uniformly:
\[
\frac{1}{T} \sum_{t=1}^{T} \ell(y_t, p_t) - \frac{1}{T} \sum_{t=1}^{T} \ell(y_t, p_t^{\ell}) < NB \sum_{j=1}^{M} \frac{T_j}{T} R_{T_j} + 3B\epsilon.
\] (5)

Proof (sketch). When \( p_t \) is the output of a given \( F_j \), we have \( \ell(y_t, p_t) \approx \ell(y_t, j/M) \approx \ell(y_t, i_j/M) \) (since \( p_t \) is in the \( j \)-th bucket, and since \( M \geq N \) is sufficiently high resolution). Since \( F_j \) is calibrated, Lemma 1 implies the \( p_t \) have vanishing regret relative to the fixed prediction \( i_j/N \); aggregating over \( j \) yields our result.

The assumptions of Lemma 2 essentially require that \( \ell \) be Lipschitz with constant \( B \), which holds e.g. for convex bounded losses that are studied in online learning. Our assumption is slightly more general since \( \ell \) may also be discontinuous (like the misclassification loss). When \( \ell \) is unbounded (like the log-loss), its values at the baseline algorithm’s predictions must be bounded away from infinity.

Next, we also establish that combining the predictions of each \( F_j^{\text{cal}} \) preserves their calibration.

Lemma 3 (Preserving calibration). If each \( F_j^{\text{cal}} \) is \((\epsilon, \ell_p)\)-calibrated, then Algorithm 1 is also \((\epsilon, \ell_p)\)-calibrated and the following bound holds uniformly over \( T \):
\[
C_T \leq \sum_{j=1}^{M} \frac{T_j}{T} R_{T_j} + \epsilon.
\] (6)

These two lemmas lead to our main claim: that Algorithm 1 solves the online recalibration problem.

Theorem 1. Let \( F^{\text{cal}} \) be an \((\ell_1, \epsilon/3B)\)-calibrated online subroutine with resolution \( N \geq 3B/\epsilon \) and let \( \ell \) be a proper loss satisfying the assumptions of Lemma 2. Then Algorithm 1 with parameters \( F^{\text{cal}} \) and \( N \) is an \( \epsilon \)-accurate online recalibration algorithm for the loss \( \ell \).

Proof. By Lemma 3, Algorithm 1 is \((\ell_1, \epsilon/3B)\)-calibrated and by Lemma 2, its regret w.r.t. the raw \( p_t^{\ell} \) tends to \( < 3B/N < \epsilon \). Hence, Theorem 1 follows.
Theorem 2. If $\ell$ is not proper, then there is no recalibration algorithm w.r.t. $\ell$.

The proof of this algorithm is a slight generalization of the counter-example provided for the $\ell_1$ loss. Interestingly, it holds equally for online and batch settings. To our knowledge, it is one of the first characterizations of the limitations of recalibration algorithms.

4.2 Extensions

Convergence rates. Next, we are interested in the rate of convergence $R_T$ of the calibration error $C_T$ of Algorithm 1. For most online calibration subroutines $F^{\text{cal}}$, $R_T \leq f(\epsilon)/\sqrt{T}$ for some $f(\epsilon)$. In such cases, we can further bound the calibration error in Equation 1 as

$$\sum_{j=1}^{M} \frac{T_j}{T} R_{T_j} \leq \sum_{j=1}^{M} \frac{\sqrt{T_j} f(\epsilon)}{T} \leq f(\epsilon)/\sqrt{T}.$$  

Table 1: Time and space complexity and convergence rate of Algorithm 1 using different subroutines.

| Subroutine | Regret min. [8] | Abernethy et al. [7] |
|------------|-----------------|----------------------|
| Time / step | $O(\frac{1}{\epsilon})$ | $O(\log(\frac{1}{\epsilon}))$ |
| Space / step | $O(\frac{1}{\epsilon^2})$ | $O(\frac{1}{\epsilon})$ |
| Calibration | $O(\frac{1}{\epsilon^2\sqrt{T}})$ | $O(\frac{1}{\epsilon\sqrt{T}})$ |
| Advantage    | Simplicity       | Efficiency           |

In the second inequality, we set the $T_j$ to be equal. Thus, our recalibration procedure introduces an overhead of $\frac{1}{\epsilon}$ in the convergence rate of the calibration error $C_T$ and of the regret in Equation 5. In addition, Algorithm 1 requires $\frac{1}{\epsilon}$ times more memory (we run $1/\epsilon$ instances of $F^{\text{cal}}_j$), but has the same per-iteration runtime (we activate one $F^{\text{cal}}_j$ per step).

Table 1 summarizes the convergence rates of Algorithm 1 when the subroutine is either the method of Abernethy et al. [7] or the simpler but slower approach based on internal regret minimization [10].

Multiclass prediction. In the multiclass setting, we seek a recalibrator $A : \Delta_K \to \Delta_K$ producing calibrated probabilities $p_t \in \Delta_K$ that target class labels $y_t \in \{1, 2, ..., K\}$. In analogy to binary recalibration, we may discretize the input space $\Delta_K$ into a $K$-dimensional grid and train a classical multi-class calibration algorithm $F^{\text{cal}}$ [3] on each subset of $p_t^f$ associated with a cell. Just like in the binary setting, a classical calibration method $F^{\text{cal}}$ predicts calibrated $p_t \in \Delta_K$ based solely on past multiclass labels $y_1, y_2, ..., y_{t-1}$; it can serve as a subroutine within Algorithm 1.

However, in the multi-class setting, this construction will require $O(1/\epsilon^K)$ running time per iteration, $O(1/\epsilon^{2K})$ memory, and will have a convergence rate of $O(1/(\epsilon^{2K}\sqrt{T}))$. The exponential dependence on $K$ cannot be avoided, since the calibration problem is fundamentally PPAD-hard [11]. However, there may exist practical workarounds inspired by popular heuristics for the batch setting, such as one-vs-all classification [12].

5 Experiments

We now proceed to study Algorithm 1 empirically. Our experiments will make use of a standard online calibration algorithm based on internal regret minimization [8] which we will refer to as REGMIN; we use it both as a subroutine in Algorithm 1 as well as for comparison.

Predicting a Bernoulli sequence. We start with a simple setting where we observe an i.i.d. sequence of $y_t \sim \text{Ber}(p)$ as well as uncalibrated predictions $(p_t^F)_{t=1}^T$ that equal 0.3 whenever $y_t = 0$ and 0.7 when $y_t = 1$. The forecaster $F$ is essentially a perfect predictor, but is not calibrated.

In Figure 1, we compare the performance of REGMIN (which does not observe $p_t^F$) to Algorithm 1 and to the uncalibrated predictor $F$. Both REGMIN and Algorithm 1 achieve low calibration error after about 300 observations, while the expert is clearly uncalibrated (Figure 1a); however, REGMIN is a terrible predictor: it always forecasts $p_t = 0.5$ and therefore has high $L_2$ loss (Figure 1b). Algorithm 1, on the other hand, makes perfect predictions by recalibrating the input $p_t^F$.

Prediction against an adversary. Next, we test the ability of our method to achieve calibration on adversarial input. At each step $t$, we choose $y_t = 0$ if $p_t > 0.5$ and $y_t = 1$ otherwise; we sample $p_t^F \sim \text{Ber}(0.5)$, which is essentially a form of noise. In Figure 1(c, d), we see that Algorithm 1 successfully ignores the noisy forecaster $F$ and instead quickly converges to making calibrated (albeit not very accurate) predictions. In this setting, Algorithm 1 essentially behaves like REGMIN.
Algorithm 1 is used to recalibrate probabilities from a question answering system (left) and a medical diagnosis system (right; both in blue). We track prediction (a) and calibration error (b) over time; plot (c) displays calibration curves after seeing all the data; circle sizes are proportional to the number of predictions in the corresponding bucket.

**Natural language understanding.** We used Algorithm 1 to recalibrate a state-of-the-art question answering system on the popular Free917 dataset. Free917 is split into 641 training and 276 testing examples; we first trained the system on the training set in batch mode as described in [14]; afterwards, we performed one round of online learning using Adagrad [15] on first the training, and then the testing examples; we applied Algorithm 1 on the uncalibrated system probabilities produced during this pass. This setup was meant to resemble a pre-trained system that further improves itself from user feedback; confidence scores may help determine when to ask users for clarifications.

![Figure 2](image)

Figure 2 (left) compares the calibration of $p_t$ to that of the raw system probabilities $p^F_t$ via calibration curves. Given pairs of predictions and outcomes $p_t$, $y_t$, we compute for each of $N$ buckets $B \in \left\{ \left\{ \frac{i}{N} \right\} \mid 0 \leq i \leq 1 \right\}$, averages $\bar{p}_B = \sum_{t \in B} p_t / |B|$ and $\bar{y}_B = \sum_{t \in B} y_t / |B|$, where $|B| = |\{ p_t \in B \}|$. A calibration curve plots the $\bar{y}_B$ as a function of $\bar{p}_B$; perfect calibration corresponds to a straight line.

For buckets below 0.9, the calibration curve for the $p^F_t$ indicates poor calibration, while the curve for Algorithm 1 follows the straight line much more closely; typically larger buckets (represented by circle size) are better calibrated. Figure 2 confirms that our accuracy (measured by the $\ell_2$ loss) closely tracks the original algorithm.

**Medical diagnosis.** Finally, we consider the problem of predicting the risk of type 1 diabetes from genomic data. We use data from 3,443 subjects — 1,963 patients and 1,480 healthy controls — that were genotyped at 447,221 SNPs as part of a genome-wide association study [17]. At each position, we encoded the major, heterozygous and minor homozygous alleles as 0, 1, 2 respectively, and used them as our feature representation. We ranked SNPs in increasing order of association with disease using the $\chi^2$ test for independence and kept the 2000 top SNPs; we found that this did not degrade our classification accuracy. We used an online $\ell_1$-regularized linear support vector machine (SVM) to predict patient outcome with the regularization weight chosen to $10^{-3}$ via batch cross-validation; we observe the $T = 3,443$ patients one at a time and report our accuracy and calibration as a function of $t \in [T]$. Uncalibrated probabilities for the SVM were computed by normalizing the
raw score $s_i$ within the range of all the previously seen scores, i.e. $p_t^F = (s_t + m_t)/2m_t$, where $m_t = \max_{1 \leq r \leq t} |s_r|$.

Figure 2 (right) shows a calibration curve for raw and recalibrated $p_t$ measured after all the data has been seen. Raw scores are not well-calibrated outside of the interval $[0.4, 0.6]$; recalibration makes them almost perfectly calibrated. Figure 2 further shows that the calibration error of Algorithm 1 is consistently lower throughout the entire learning process, while accuracy approaches to within 0.01 of that of $p_t^F$.

6 Previous Work

Calibrated probabilities are widely used as confidence measures in the context of binary classification. Such probabilities are obtained via recalibration methods, of which Platt scaling [18] and isotonic regression [16] are by far the most popular. Recalibration methods also possess multiclass extensions, which typically involve training multiple one-vs-all predictors [12], as well as extensions to ranking losses [19], combinations of estimators [20], and structured prediction [21].

In the online setting, the calibration problem was formalized by Dawid [22]; online calibration techniques were first proposed by Foster and Vohra [6]. Existing algorithms are based on internal regret minimization [8] or on Blackwell approachability [23]; recently, these approaches were shown to be closely related [7, 10]. Recent work has shown that online calibration is PPAD-hard [11].

The concepts of calibration and sharpness were first formalized in the statistics literature [24, 5]. These metrics are captured by a class of proper losses and can be used both for evaluating [25, 4] and constructing [21] calibrated forecasts.

7 Discussion and Conclusion

Comparison to the batch setting. Algorithm 1 can be understood as a direct analogue of a simple density estimation technique called the histogram method. This technique divides the $p_t^F$ into $N$ bins and estimates the average $y$ in each bin. By the i.i.d. assumption, output probabilities will be calibrated; sharpness will be determined by the bin width. Note that by Hoeffding’s inequality, the average in a given bin with converge at a rate of $O(\sqrt{T_j})$ [26]. This is faster than the $O(1/\sqrt{T_j})$ rate of Abernethy et al. [7] and suggests that calibration is more challenging in the online setting.

Comparison to checking rules. An alternative way to avoid uninformative predictions (e.g. 0.5 on 010101...) is via the framework of checking rules [8]. However, these rules must be specified in advance (e.g. the pattern 010101 must be known) and this framework does not explicitly admit covariates $x_t$. Our approach on the other hand recalibrates any $x_t, y_t$ in a black-box manner.

Conclusion. The need for calibrated probability estimates naturally arises in many applications, including medical diagnosis and natural language processing. Current recalibration techniques implicitly require that the data is distributed i.i.d., which potentially makes them unreliable when this assumption does not hold. Although the online learning literature introduces several techniques that forgo this assumption, these algorithms are not suitable for machine learning applications, since they do not admit covariates, and therefore cannot produce useful predictions.

In this work, we introduced the first recalibration technique that provably recalibrates any existing forecaster with a vanishingly small degradation in accuracy. This method does make i.i.d. assumptions, and is provably calibrated even on adversarial input. We analyzed our method’s theoretical properties and showed excellent empirical performance on several real-world benchmarks, where the method converges quickly and retains good accuracy.
References

[1] Xiaoqian Jiang, Melanie Osl, Jihoon Kim, and Lucila Ohno-Machado. Calibrating predictive model estimates to support personalized medicine. *JAMIA*, 19(2):263–274, 2012.

[2] Khanh Nguyen and Brendan O’Connor. Posterior calibration and exploratory analysis for natural language processing models. *CoRR*, abs/1508.05154, 2015.

[3] Dong Yu, Jinyu Li, and Li Deng. Calibration of confidence measures in speech recognition. *Trans. Audio, Speech and Lang. Proc.*, 19(8):2461–2473, November 2011. ISSN 1558-7916.

[4] J. Brocker. Reliability, sufficiency, and the decomposition of proper scores. *Quarterly Journal of the Royal Meteorological Society*, 135(643):1512–1519, 2009.

[5] Tilmann Gneiting, Fadoua Balabdaoui, and Adrian E. Raftery. Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society: Series B*, 69(2):243–268, 2007.

[6] Dean P. Foster and Rakesh V. Vohra. Asymptotic calibration, 1998.

[7] Jacob Abernethy, Peter L. Bartlett, and Elad Hazan. Blackwell approachability and no-regret learning are equivalent. In *COLT 2011 - The 24th Annual Conference on Learning Theory*, pages 27–46, 2011.

[8] Nicolo Cesa-Bianchi and Gabor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, New York, NY, USA, 2006. ISBN 0521841089.

[9] Shai Shalev-Shwartz. *Online Learning: Theory, Algorithms, and Applications*. Phd thesis, Hebrew University, 2007.

[10] Shie Mannor and Gilles Stoltz. A geometric proof of calibration. *Math. Oper. Res.*, 35(4):721–727, 2010.

[11] Elad Hazan and Sham M. Kakade. (weak) calibration is computationally hard. In *COLT 2012 - The 25th Annual Conference on Learning Theory*, June 25-27, 2012, Edinburgh, Scotland, pages 3.1–3.10, 2012.

[12] Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Eighth ACM Conference on Knowledge Discovery and Data Mining*, pages 694–699, 2002.

[13] Jonathan Berant and Percy Liang. Semantic parsing via paraphrasing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 1415–1425, 2014.

[14] Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic parsing on freebase from question-answer pairs. In *EMNLP 2013*, pages 1533–1544, 2013.

[15] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12:2121–2159, July 2011. ISSN 1532-4435.

[16] Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *Proceedings of the 22Nd International Conference on Machine Learning*, ICML ’05, 2005.

[17] The Wellcome Trust Case Control Consortium. Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls. *Nature*, 447(7145):661–678, June 2007.

[18] John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *ADVANCES IN LARGE MARGIN CLASSIFIERS*, pages 61–74. MIT Press, 1999.

[19] Aditya Krishna Menon, Xiaoqian Jiang, Shankar Vembu, Charles Elkan, and Lucila Ohno-Machado. Predicting accurate probabilities with a ranking loss. In *29th International Conference on Machine Learning*, 2012.

[20] Leon Wenliang Zhong and James T. Kwok. Accurate probability calibration for multiple classifiers. *IJCAI '13*, pages 1939–1945. AAAI Press, 2013. ISBN 978-1-57735-633-2.

[21] Y. Kuleshov and P. Liang. Calibrated structured prediction. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[22] A. Philip Dawid. The well-calibrated bayesian. *Journal of the American Statistical Association*, 77(379):605–610, 1982.

[23] Dean P. Foster. A Proof of Calibration Via Blackwell’s Approachability Theorem. Discussion Papers 1182, Northwestern University, February 1997.

[24] A. H. Murphy. A new vector partition of the probability score. *Journal of Applied Meteorology*, 12(4):595–600, 1973.

[25] Andreas Buja, Werner Stuetzle, and Yi Shen. Loss functions for binary class probability estimation and classification: Structure and applications, 2005.

[26] Luc Devroye, László Györfi, and Gábor Lugosi. *A probabilistic theory of pattern recognition*. Applications of mathematics. Springer, New York, Berlin, Heidelberg, 1996. ISBN 978-0-387-94618-4.
A Correctness of the recalibration procedure

In the appendix, we formally prove the theorems that are presented in the body of the main paper. In this section, we start by proving the results that demonstrate that Algorithm 1 produces forecasts that are calibrated and have vanishing regret relative to the input baseline forecaster \( F \).

We assume, again, for expository purposes, that calibration is measured using the \( \ell_1 \) norm. This loss is very often considered in the literature; in particular the current best recalibration algorithm by Abernethy et al. \cite{Abernethy} targets the \( \ell_1 \) loss directly. We will extend our discussion to general norms in the next section.

A.1 Notation

For now, we focus on the \( \ell_1 \) norm, and we define the calibration error of a forecaster \( F_{\text{cal}} \) as

\[
C_T = \sum_{i=0}^{N} \left| \rho_T(i/N) - \frac{i}{N} \right| \left( \frac{1}{T} \sum_{t=1}^{T} \mathbb{I}_{\{p_t = \frac{i}{M}\}} \right),
\]

where \( \rho_T(p) = \sum_{t=1}^{T} \mathbb{I}_{\{y_t = p\} / T} \) denotes the frequency at which event \( y = 1 \) occurred over the times when we predicted \( p \).

We further define the calibration error when \( F_{\text{cal}} \) or Algorithm 1 predicts \( i/N \) as (respectively)

\[
C_{T,i}^{(j)} = \left| \rho_T^{(j)}(i/N) - \frac{i}{N} \right| \left( \frac{1}{T} \sum_{t=1}^{T} \mathbb{I}^{(j)}_{t,i} \right),
\]

\[
C_T = \left| \rho_T(i/N) - \frac{i}{N} \right| \left( \frac{1}{T} \sum_{t=1}^{T} \mathbb{I}_{t,i} \right),
\]

where \( \mathbb{I}^{(j)}_{t,i} = \mathbb{I}\{p_t = \frac{i}{M} \} \cap \mathbb{I}^{(j)} \) is an indicator for the event that \( F_{\text{cal}} \) is triggered at time \( t \) and predicts \( i/N \). Similarly, \( \mathbb{I}_{t,i} = \mathbb{I}\{p_t = i/N \} = \sum_{j=1}^{M} \mathbb{I}^{(j)}_{t,i} \) indicates that \( i/N \) was predicted at time \( t \), and \( T_j = \sum_{t=1}^{T} \sum_{i=0}^{N} \mathbb{I}^{(j)}_{t,i} \) is the number of calls to \( F_{\text{cal}} \). Also,

\[
\rho_T^{(j)}(i/N) = \frac{\sum_{t=1}^{T} \mathbb{I}^{(j)}_{t,i} y_t}{\sum_{t=1}^{T} \mathbb{I}^{(j)}_{t,i}},
\]

\[
\rho_T(i/N) = \frac{\sum_{t=1}^{T} \mathbb{I}_{t,i} y_t}{\sum_{t=1}^{T} \mathbb{I}_{t,i}},
\]

are the empirical success rates for \( F_{\text{cal}} \) and Algorithm 1 respectively.

Note that with these definitions, we may write the calibration losses of \( F_{\text{cal}} \) and Algorithm 1 as respectively \( C_{T,i}^{(j)} = \sum_{i=0}^{N} C_{T,i}^{(j)} \) and \( C_T = \sum_{i=0}^{N} C_{T,i} \).

A.2 Assumptions

Our lemmas will make the same assumptions as in the paper. In particular, recall that we assume that the subroutines \( F_{\text{cal}} \) are (\( \epsilon, \ell_1 \))-calibrated and that \( C_{T,j}^{(j)} \leq R_{T_j} + \epsilon \) uniformly \( (R_{T_j} = o(1)) \) as \( T_j \to \infty \); \( T_j \) is the number of calls to instance \( F_{\text{cal}}^{(j)} \).

Recall that we also assume that the loss \( \ell \) used for measuring accuracy is proper. We also require that it is bounded with \( \ell(i, i/N) < B \) for \( i \in [N]_0 \); since the set of predictions is finite, this is a mild requirement. Finally, we will make additional continuity assumptions on \( \ell \) in Lemma 1.

A.3 Calibration implies no internal regret

Here, we show that a calibrated forecaster also has small internal regret relative to any bounded proper loss.

**Lemma 1.** If \( \ell \) is a bounded proper loss, then an \( (\epsilon, \ell_1) \)-calibrated \( F_{\text{cal}} \) a.s. has a small internal regret w.r.t. \( \ell \) and satisfies uniformly over time \( T \) the bound

\[
R_{T,j}^{\text{int}} = \max_{i,j} \sum_{t=1}^{T} \mathbb{I}_{p_t = i/N} (\ell(i/N, y_t) - \ell(j/N, y_t)) \leq 2B(R_T + \epsilon).
\]
Proof. Let $T$ be fixed for the rest of this proof. Let $I_{ti} = \mathbb{1}_{p_t = i/N}$ be the indicator of $F^{\text{cal}}$ outputting prediction $i/N$ at time $t$, let $T_i = \sum_{t=1}^{T} I_{ti}$ denote the number of time $i/N$ was predicted, and let

$$R_{T,i,j}^{\text{int}} = \sum_{t=1}^{T} I_{ti} (\ell(i/N, y_t) - \ell(j/N, y_t))$$

denote the gain (measured using the proper loss $\ell$) from retrospectively switching all the plays of action $i$ to $j$. This value forms the basis of the definition of internal regret (Section 2).

Let $T(i, y) = \sum_{t=1}^{T} I_{ti} \mathbb{1}\{y_t = y\}$ denote the total number of $i/N$ forecasts at times when $y_t = y \in \{0, 1\}$. Observe that we have

$$T(i, y) = \sum_{t=1}^{T} I_{ti} \mathbb{1}\{y_t = y\} = \frac{\sum_{t=1}^{T} I_{ti} \mathbb{1}\{y_t = y\} T_i}{\sum_{t=1}^{T} I_{ti}} = \frac{\sum_{t=1}^{T} I_{ti} \mathbb{1}\{y_t = y\}}{\sum_{t=1}^{T} I_{ti}} T_i$$

$$= q(i, y) T_i + T_i \left( \frac{\sum_{t=1}^{T} I_{ti} \mathbb{1}\{y_t = y\}}{\sum_{t=1}^{T} I_{ti}} - q(i, y) \right)$$

$$= q(i, y) T_i + T_i \left( \rho_T(i/N) - i/N \right),$$

where $q(i, y) = i/N$ if $y = 1$ and $1 - i/N$ if $y = 0$. The last equality follows using some simple algebra after adding and subtracting one inside the parentheses in the second term.

We now use this expression to bound $R_{T,i,j}^{\text{int}}$:

$$R_{T,i,j}^{\text{int}} = \sum_{t=1}^{T} I_{ti} (\ell(i/N, y_t) - \ell(j/N, y_t))$$

$$= \sum_{y \in \{0, 1\}} T(i, y) (\ell(i/N, y) - \ell(j/N, y))$$

$$\leq \sum_{y \in \{0, 1\}} q(i, y) T_i (\ell(i/N, y) - \ell(j/N, y)) + \sum_{y \in \{0, 1\}} BT_i |\rho_T(i/N) - i/N|$$

$$\leq 2BT_i |\rho_T(i/N) - i/N|,$$

where in the first inequality, we used $\ell(i/N, y) - \ell(j/N, y) \leq \ell(i/N, y) \leq B$, and in the second inequality we used the fact that $\ell$ is a proper loss.

Since internal regret equals $R_T^{\text{int}} = \max_{i,j} R_{T,i,j}^{\text{int}}$, we have

$$R_T^{\text{int}} \leq \sum_{i=1}^{N} \max_{j} R_{T,i,j}^{\text{int}} \leq 2B \sum_{i=0}^{N} T_i |\rho(i/N) - i/N| \leq 2B(R_T + \epsilon).$$

\[\square\]

A.4 Recalibrated forecasts have low regret relative to uncalibrated forecasts

Next, we use the above result to prove Lemma 2, i.e. we show that the forecasts recalibrated using Algorithm 1 have low regret relative to the baseline uncalibrated forecasts.

**Lemma 2 (Recalibration preserves accuracy).** Consider Algorithm 1 with parameters $M \geq N > 1/\epsilon$ and let $\ell$ be a bounded proper loss for which

1. $\ell(p, y) \leq \ell(y_t, j/M) + B/M$ whenever $p \in [j/M, (j+1)/M]$;
2. $\ell(p, y) \leq \ell(y_t, i/N) + B/N$ whenever $p \in [i/N, (i+1)/N]$;

Then the recalibrated $p_t$ a.s. have vanishing $\ell$-loss regret relative to $p_t^*$ and we have uniformly:

$$\frac{1}{T} \sum_{t=1}^{T} \ell(y_t, p_t) - \frac{1}{T} \sum_{t=1}^{T} \ell(y_t, p_t^*) < NB \sum_{j=1}^{M} \frac{T_j}{T} R_{T_j} + 3B\epsilon. \quad (9)$$
Proof. By the previous lemma, we know that an algorithm with resolution \( \frac{1}{N} \) whose calibration error is bounded by \( R_T = o(1) \) also minimizes internal regret at a rate of \( 2BR_T \), and thus external regret at a rate of \( 2NBR_T \).

Next, let us use \( I_{j,t} = \mathbb{I}\{p_t^F \in \left[ \frac{j-1}{M}, \frac{j}{M} \right) \} \) to indicate that \( F_{j,t}^\text{cal} \) was called at time \( t \). Also, let \( i_j \) denote the index \( i \in [N] \) associated with the interval \( [i/N, (i+1)/N) \) in which \( j/M \) falls.

We establish our main claim as follows:

\[
\frac{1}{T} \sum_{t=1}^{T} \ell(y_t, p_t) - \frac{1}{T} \sum_{t=1}^{T} \ell(y_t, p_t^F) = \frac{1}{T} \sum_{t=1}^{T} \left( \sum_{j=1}^{M} \left( \ell(y_t, p_t) - \ell(y_t, p_t^F) \right) I_{j,t} \right) \\
< \frac{1}{T} \sum_{t=1}^{T} \left( \sum_{j=1}^{M} \left( \ell(y_t, p_t) - \ell(y_t, \frac{j}{M}) \right) I_{j,t} + B \right) \\
< \frac{1}{T} \sum_{t=1}^{T} \left( \sum_{j=1}^{M} \left( \ell(y_t, p_t) - \ell(y_t, \frac{i_j}{N}) \right) I_{j,t} + \frac{2B}{N} \right) \\
\leq NB \sum_{j=1}^{M} \frac{T_j}{T} R_{T_j} + \frac{3B}{N},
\]

where \( R_{T_j} \) is a bound on the calibration error of \( F_{j,t}^\text{cal} \) after \( T_j \) plays.

In the first two inequalities, we use our assumption on the loss \( \ell \), and that \( \frac{1}{M} \leq \frac{1}{N} \). The last inequality follows because \( F_{j,t}^\text{cal} \) minimizes external regret w.r.t. the constant action \( i_j \) at a rate of \( NBR_{T_j} \).

A.5 Correctness of Algorithm 1

We now prove our main result about the correctness of Algorithm 1.

Theorem 1. Let \( F^\text{cal} \) be an \( (\ell_1, \epsilon/3B) \)-calibrated online subroutine with resolution \( N \geq 3B/\epsilon \), and let \( \ell \) be a proper loss satisfying the assumptions of Lemma 2. Then Algorithm 1 with parameters \( F^\text{cal} \) and \( N \) is an \( \epsilon \)-accurate online recalibration algorithm for the loss \( \ell \).

Proof. It is easy to show that Algorithm 1 is \( (\ell_1, \epsilon/3B) \)-calibrated by the same argument as Lemma 1 (see the next section for a formal proof). By Lemma 4, its regret w.r.t. the raw \( p_t^F \) tends to \( < 3B/N < \epsilon \). Hence, the theorem follows.

Finally, we would like to instantiate this theorem with the misclassification loss \( \ell_{mc}(y, p) = (1 - y)p < 0.5 + yI_{p < 0.5} \), which is arguably the most interesting and general loss.

Corollary 1. Let \( F^\text{cal} \) be an \( (\ell_1, \epsilon/3) \)-calibrated online algorithm with resolution \( N \geq 3/\epsilon \), where \( N \) is a power of two. Then Algorithm 1 is an \( \epsilon \)-accurate online recalibration algorithm for the loss \( \ell_{mc} \).

Proof. We only need to show that \( \ell_{mc} \) satisfies the requirements of the above theorem. Clearly, we have \( B = 1 \). Note that if \( N \) is a power of two, the point \( 0.5 \) is contained in an interval \( [0.5, 0.5 + 1/N) \); hence, by construction, \( \ell_{mc}(y, p) \) satisfies the other two conditions on the proper loss, since changing \( p \) doesn’t affect \( \ell_{mc} \), as long as \( p \) is in the same interval.

B Calibration with \( \ell_p \) norms

In this section, we further discuss how our method behaves when we enforce calibration in a norm that is different from the \( \ell_1 \) norm.
B.1 Notation

Recall that we defined the calibration error as

$$C_T^p = \sum_{i=0}^N |\rho_T(i/N) - i/N|^p \left( \frac{1}{T} \sum_{t=1}^T \mathbb{1}_{\{p_t = \frac{i}{N}\}} \right),$$

(10)

which is the weighted distance between the $\rho_T(i/N)$ and the predicted probabilities $i/N$. We say that $F^\text{cal}$ is an $(\epsilon, \ell_p)$-calibrated algorithm with resolution $1/N$ if

$$\lim_{T \to \infty} \sup_{i} C_T^p \leq \epsilon.$$  

(11)

Abernethy et al. (2011) proposed the fastest online calibration algorithm; this method uses the $\ell_1$ norm. Since L1 is the largest norm, their algorithm extends directly to other values of $p$.

However, there exist simpler algorithms that target the $\ell_2$ norm; we would like our results to naturally extend to these algorithms as well.

B.2 Proving that calibration holds under any norm

First, we want to give a proof of Lemma 3; this proof holds for any norm $\ell_p$.

**Lemma 3** (Preserving calibration). If each $F^\text{cal}_T$ is $(\epsilon, \ell_p)$-calibrated, then Algorithm 1 is also $(\epsilon, \ell_p)$-calibrated and the following bound holds uniformly over $T$:

$$C_T \leq \sum_{j=1}^{M} \frac{T_j}{T} R_{T_j} + \epsilon.$$  

(12)

**Proof.** Let $\ell_{t,i}^{(j)} = \sum_{t=1}^{T} \ell_{t,i}^{(j)}$ and note that $\sum_{t=1}^{T} \ell_{t,i}^{(j)} = \sum_{j=1}^{M} \ell_{t,i}^{(j)}$. We may write

$$C_{T,i} = \frac{\sum_{t=1}^{T} \ell_{t,i}^{(j)}}{T} \left| \rho_T(i/N) - i/N \right|^p = \frac{\sum_{j=1}^{M} \ell_{t,i}^{(j)}}{T} \left| \sum_{j=1}^{M} \frac{\ell_{t,i}^{(j)}}{\sum_{j=1}^{M} \ell_{t,i}^{(j)}} \rho_T(j/N) - i/N \right|^p$$

$$= \frac{\sum_{j=1}^{M} \ell_{t,i}^{(j)}}{T} \left| \sum_{j=1}^{M} \frac{\ell_{t,i}^{(j)}}{\sum_{j=1}^{M} \ell_{t,i}^{(j)}} \rho_T(j/N) - i/N \right|^p$$

$$\leq \frac{\sum_{j=1}^{M} \ell_{t,i}^{(j)}}{T} \left| \rho_T(j/N) - i/N \right|^p = \sum_{j=1}^{M} \frac{T_{T,j}}{T} C_{T,i}^{(j)},$$

where in the last line we used Jensen’s inequality. Plugging in this bound in the definition of $C_T$, we find that

$$C_T = \sum_{i=1}^{N} C_{T,i} \leq \sum_{j=1}^{M} \sum_{i=1}^{N} \frac{T_{T,j}}{T} C_{T,i}^{(j)} \leq \sum_{j=1}^{M} \frac{T_{T,j}}{T} R_{T_j} + \epsilon,$$

Since each $R_{T_j} \to 0$, Algorithm 1 will be $\epsilon$-calibrated. \qed

B.3 Recalibration accuracy under the $\ell_2$ norm

Next, we would like to prove an analogue of Lemma 2 for the popular $\ell_2$ norm. Although calibration follows from the $\ell_1$ norm version, here, we use a slightly different argument to derive simpler and better convergence rates. To obtain this result, we make the additional assumption that the proper loss that we are using to measure accuracy is the $\ell_2$ loss.

**Lemma 4** (Recalibration preserves accuracy). Consider Algorithm 1 with parameters $M \geq N > 1/\epsilon$. Suppose that the $F^\text{cal}_T$ are $(\epsilon, \ell_2)$-calibrated. Then the recalibrated $p_t$, a.s. have vanishing $\ell_2$ loss regret relative to $p^*_t$:

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

(13)
Proof. Our proof will use the fact that an algorithm with resolution $\frac{1}{N}$ whose $\ell_2$ calibration error is bounded by $R_T = o(1)$ also minimizes external regret (relative to the $\ell_2$ loss) at a rate of $N R_T$. See e.g. Lemma 4.4 in [3] for a proof of this fact.

Let us use $I_{j,t} = I\{p_t^F \in [\frac{j-1}{M}, \frac{j}{M})\}$ to indicate that $F_j^{cal}$ was called at time $t$. Also, let $i_j$ denote an index $i \in [N]$ close to $j$ in the sense of $|\frac{i_j}{N} - \frac{j}{M}| \leq \frac{1}{N}$; by our assumption that $M \geq N$, this index exists.

We establish our main claim as follows:

$$\frac{1}{T} \sum_{t=1}^{T} (y_t - p_t)^2 - \frac{1}{T} \sum_{t=1}^{T} (y_t - p_t^F)^2$$

$$= \frac{1}{T} \sum_{t=1}^{T} \left( \sum_{j=1}^{M} \left((y_t - p_t)^2 - (y_t - p_t^F)^2\right) I_{j,t} \right)$$

$$< \frac{1}{T} \sum_{t=1}^{T} \left( \sum_{j=1}^{M} \left((y_t - p_t)^2 - (y_t - \frac{j}{M})^2\right) I_{j,t} + \frac{2}{N}\right)$$

$$< \frac{1}{T} \sum_{t=1}^{T} \left( \sum_{j=1}^{M} \left((y_t - p_t)^2 - (y_t - \frac{i_j}{N})^2\right) I_{j,t} + \frac{4}{N}\right)$$

$$\leq N \sum_{t=1}^{T} \sum_{j=1}^{M} \frac{T_j}{T} R_{T_j} + \frac{4}{N},$$

where $R_{T_j}$ is a bound on the calibration error of $F_j^{cal}$ after $T_j$ plays.

The first inequality holds because $|p_t^F - \frac{j}{M}| \leq \frac{1}{M} \leq \frac{1}{N}$ when $I_{j,t} = 1$ and because $\ell_2(p_t^F, y_t) \geq \ell_2(p_t^F, y_t) + \frac{\ell_2}{M}(p, y_t)(\frac{j}{M} - p_t^F)$. We repeat the same argument in the second inequality using the fact that $|\frac{i_j}{N} - \frac{j}{M}| \leq \frac{1}{N}$ for some $i_j$. The last inequality follows because $F_j^{cal}$ minimizes external regret w.r.t. the constant action $i_j$ at a rate of $N R_{T_j}$.

\[Q.E.D.\]

B.4 Discussing general $\ell_p$ norms

Finally, let us briefly consider what happens when calibration is measured according to a different $\ell_p$ norm.

Since we can make $C_{T,\ell_1} < \epsilon$ for an arbitrarily small $\epsilon > 0$ by increasing $N$ and $T$, by continuity and by the equivalence of norms, this means that we can make $C_{T,\ell_p}$ arbitrarily small as well. The only question has to do with the rate at which $C_{T,\ell_p} \to 0$, and this varies depending on the loss.

As an example, let us derive the correct convergence rate for the simple case of the $\ell_2$ loss. It follows from the bound $||x||_1 \leq \sqrt{d}||x||_2$ that

$$\sum_{i=0}^{N} w_i |a_i| \leq \sqrt{\sum_{i=0}^{N} w_i^2} \|a_i\| \leq \sqrt{N + 1} \sqrt{\sum_{i=0}^{N} w_i a_i^2},$$

where we assume that $0 \leq w_i \leq 1$. Using this with our definition of calibration, we find that

$$\sqrt{\sum_{i=0}^{N} \rho_T(i/N) - \frac{i}{N}} \left( \frac{1}{T} \sum_{t=1}^{T} I_{(p_t = \frac{i}{N})} \right) \leq \sqrt{N + 1} \left( \sqrt{R_T + \epsilon} \right)$$

when $C_{T,\ell_2} \leq R_T + \epsilon$. This means that if an algorithm minimizes L2 calibration, the L1 calibration term will converge more slowly to a larger value. To achieve a level of accuracy we need to (among other things) square the resolution parameter $N$.

C Impossibility of recalibrating non-proper losses

We conclude the appendix by explaining why non-proper losses cannot be calibrated.
Theorem 2. If $\ell$ is not proper, then there is no recalibration algorithm w.r.t. $\ell$.

Proof. If $\ell$ is not proper, there exist a $p'$ and $q$ such that $E_{y \sim \text{Ber}(p')} \ell(y, q) < E_{y \sim \text{Ber}(p')} \ell(y, p')$.

Consider a sequence $y_t$ for which $y_t \sim \text{Ber}(p')$ for all $t$. Clearly the prediction of a calibrated forecaster $p_t$ much converge to $p'$ and the average loss will approach $\ell(y, p')$. This means that we cannot recalibrate the constant predictor $p_t = q$ without making its loss $\ell(y, q)$ higher. We thus have a forecaster that cannot be recalibrated with respect to $\ell$. □