Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

Estimating Uncertainty Online Against an Adversary

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

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

Abstract

Assessing uncertainty is an important step towards ensuring the safety and reliability of machine learning systems. 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. This represents an extension of classical online learning that handles uncertainty in addition to guaranteeing accuracy under adversarial assumptions. We establish formal guarantees for our methods, and we validate them on two real-world problems: question answering and medical diagnosis from genomic data.

Introduction

Assessing uncertainty is an important step towards ensuring the safety and reliability of machine learning systems. In many applications of machine learning — including medical diagnosis (Jiang et al. 2012), natural language understanding (Nguyen and O'Connor 2015), and speech recognition (Yu, Li, and Deng 2011) — assessing confidence can be as important as obtaining high accuracy. This work explores confidence estimation for classification problems.

An important limitation of existing methods is the assumption that data is sampled i.i.d. from a distribution $\mathbb{P}(x, y)$; when test-time data is distributed according to a different $\mathbb{P}^*$, these methods may become overconfident and erroneous. 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.

In the classification setting, the most 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, its forecast would be calibrated if it rained on about 60 of those 100 days.

Background. Calibrated confidence estimates are typically constructed via *recalibration*, using methods such as Platt scaling (Platt 1999) or isotonic regression (Niculescu-Mizil and Caruana 2005). 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 $\mathbb{P}(Y = 1|h(X) = s_i)$ 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 (Foster and Vohra 1998; Abernethy, Bartlett, and Hazan 2011). These *classical* methods are not suitable for standard prediction tasks in their current form. For one, they do not admit covariates x_i that might be available to improve the prediction of y_i; hence, they also do not consider the predictive power of the forecasts. For example, predicting 0.5 on a sequence 01010... formed by alternating 0s and 1s is considered a valid calibrated forecaster. The algorithms we present here combine the advantages of online calibration (adversarial assumptions), and of batch probability recalibration (covariates and forecast sharpness).

Online learning with uncertainty. Whereas classical online optimization aims to accurately predict targets y given x (via a convex loss $\ell(x, y)$), our algorithms aim to accurately predict *uncertainties* $p(y = \hat{y})$. The p here are defined as empirical frequencies over data seen so far; it turns out that these probability-like quantities can be estimated under the standard adversarial assumptions of online learning. We thus see our work as extending classical online optimization to handle uncertainty in addition to guaranteeing accuracy.

Example. 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 up-

per 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 be i.i.d. due to e.g., seasonal disease outbreaks.

Contributions. More formally, our contributions are to:

- Formulate a new problem called *online recalibration*, which requires producing calibrated probabilities on potentially adversarial input, while retaining the predictive power of a given baseline uncalibrated forecaster.
- Propose a meta-algorithm for online recalibration that uses classical online calibration as a black box subroutine.
- Show that our technique can recalibrate the forecasts of any existing classifier at the cost of an $O(1/\sqrt{\epsilon})$ 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.

Background

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

Learning with Expert Advice

Learning with expert advice (Cesa-Bianchi and Lugosi 2006) is a special case of the general online optimization framework (Shalev-Shwartz 2007) 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 $\sum_{i=1}^{N} w_{ti}\ell(y_t, a_{it})$, where $\ell(y_t, a_{it})$ is the loss under expert i's advice a_{it}. Performance in this setting is measured using two notions of regret.

Definition 1. *The external regret R_T^{ext} and the internal regret R_T^{int} are defined as*

$$R_T^{\text{ext}} = \sum_{t=1}^{T} \bar{\ell}(y_t, p_t) - \min_{i\in[N]} \sum_{t=1}^{T} \ell(y_t, a_{it})$$

$$R_T^{\text{int}} = \max_{i,j\in[N]} \sum_{t=1}^{T} p_{t,i} \left(\ell(y_t, a_{it}) - \ell(y_t, a_{jt})\right),$$

where $\bar{\ell}(y, p) = \sum_{i=1}^{N} p_i \ell(y, a_{it})$ 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.

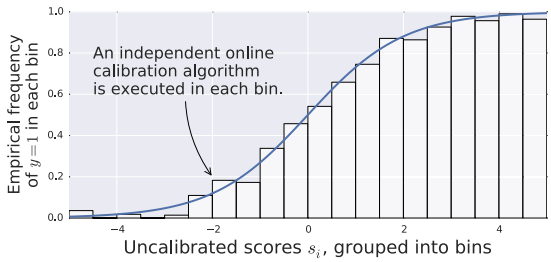

Figure 1: Our method bins uncalibrated scores and runs online calibration subroutines in each bin (not unlike the histogram recalibration method targeting $\mathbb{P}(y = 1 \mid s = t)$).

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

Definition 2. *A loss $\ell(y, p) : \{0, 1\} \times [0, 1] \to \mathbb{R}_+$ is proper if $p \in \arg\min_q \mathbb{E}_{y\sim Ber(p)}\ell(y, q)\ \forall p$.*

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 $\ell_{\text{mc}}(y, p) = (1-y)\mathbb{I}_{p<0.5} + y\mathbb{I}_{p\geq 0.5}$. Counter-examples include the L1 and the hinge losses.

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^{cal} be a forecaster making predictions in the set $\{\frac{i}{N} \mid i = 0, ..., N\}$, where $1/N$ is called the *resolution* of F^{cal}; consider the quantities $\rho_T(p) = \frac{\sum_{t=1}^{T} y_t \mathbb{I}_{p_t=p}}{\sum_{t=1}^{T} \mathbb{I}_{p_t=p}}$ and

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

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 ℓ_p distance between the $\rho_T(i/N)$ and the predicted 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^{cal} is an (ϵ, ℓ_p)-calibrated algorithm with resolution $1/N$ if $\limsup_{T\to\infty} C_T^p \leq \epsilon$ a.s.*

There exists a vast literature on calibration in the online setting (Cesa-Bianchi and Lugosi 2006) 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 $y_1, ..., y_{t-1}$. 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 $i \in [N]_0$. All such algorithms are randomized, hence our results will hold almost surely (a.s.). See Chapter 4 in Cesa-Bianchi and Lugosi for details.

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 (Shalev-Shwartz 2007; Cesa-Bianchi and Lugosi 2006). At every step $t = 1, 2, ...$:

1: Nature reveals features $x_t \in \mathbb{R}^d$.
2: Forecaster F predicts $p_t^F = \sigma(w_{t-1} \cdot x_t) \in [0, 1]$.
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 $\ell(y_t, p_t)$, 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.

Here, σ is a transfer function chosen such that the task is convex in w_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^F that are accurate and well-calibrated in the following sense.

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

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \left(\ell^{acc}(y_t, p_t) - \ell^{acc}(y_t, p_t^F)\right) \leq \epsilon. \quad (2)$$

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_t^F 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^{cal} on the data $\{p_t^F, y_t \mid p_t^F \in I_j\}$ belonging to each bucket $I_j \in \mathcal{I}$; at prediction time, it calls the instance of F^{cal} associated with the bucket of the uncalibrated forecast p_t^F.

Algorithm 1 Online Recalibration

Require: Online calibration subroutine F^{cal} and number of buckets M
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 $\mathcal{F} = \{F_j^{\text{cal}} \mid j = 0, ..., M-1\}$ be a set of M independent instances of F^{cal}.
3: **for** $t = 1, 2, ...$: **do**
4: Observe uncalibrated forecast p_t^F.
5: Let $I_j \in \mathcal{I}$ be the interval containing p_t^F.
6: Let p_t be the forecast of F_j^{cal}.
7: Output p_t. Observe y_t and pass it to F_j^{cal}.

Algorithm 1 works because a calibrated predictor is at least as accurate as any constant predictor; in particular, each subroutine F_j^{cal} is at least as accurate as the prediction $\frac{j}{M}$, which also happens to be approximately p_t^F when F_j^{cal} was called. Thus, each F_j^{cal} is as accurate as its input sequence of p_t^F. One can then show that if each each F_j^{cal} is accurate and calibrated, then so it their aggrgate, 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.

Calibration and Accuracy of Online Recalibration

Notation. We define the calibration error of F_j^{cal} and of Algorithm 1 at i/N as (respectively)

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

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

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

Assumptions. In this section, we will assume that the subroutine F^{cal} used in Algorithm 1 is (ϵ, ℓ_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_j^{cal}). This also implies ℓ_p-calibration (by continuity of ℓ_p), albeit with different rates R_{T_j} and a different ϵ. Abernethy, Bartlett, and Hazan introduce (ϵ, ℓ_1)-calibrated F_j. We also provide proofs for the ℓ_2 loss in the appendix.

Crucially, we assume that the loss ℓ used for measuring accuracy is *proper* and bounded with $\ell(\cdot, i/N) < B$ for $i \in [N]_0$; since the set of predictions is finite, this is a mild requirement. Finally, we make additional continuity assumptions on ℓ in Lemma 2.

Recalibration with proper losses. Surprisingly, not every loss ℓ 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 ℓ_1 loss of $\approx$0.44; however predicting 0 for all t has an ℓ_1 loss of $1/3 < 0.44$. Thus we cannot recalibrate this sequence and also remain equally accurate under the ℓ_1 loss. The same argument also applies to batch recalibration (e.g. Platt scaling): we only need to assume that $y_t \sim \mathrm{Ber}(1/3)$ 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 ℓ is a proper loss bounded by $B > 0$, then an (ϵ, ℓ_1)-calibrated F^{cal} a.s. has a small internal regret w.r.t. ℓ and satisfies uniformly over time T the bound*

$$R_T^{\mathrm{int}} = \max_{ij} \sum_{t=1}^{T} \mathbb{I}_{p_t = i/N} \left(\ell(y_t, i/N) - \ell(y_t, j/N)\right) \le 2B(R_T + \epsilon).$$

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 $\sum_{t:p_t=p_1} \ell(y_t, p)$ over the times t when p_1 was predicted (at least better so than p_2). However, our ℓ_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 ℓ is proper, i.e. that $p \in \arg\min_q \mathbb{E}_{y\sim\mathrm{Ber}(p)} \ell(y, q)$.

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 $M \ge N > 1/\epsilon$ and let ℓ be a bounded proper loss for which*

1. *$\ell(y_t, p) \le \ell(y_t, j/M) + B/M$ for $p \in [j/M, (j+1)/M)$;*
2. *$\ell(y_t, p) \le \ell(y_t, i/N) + B/N$ for $p \in [i/N, (i+1)/N)$;*

Then the recalibrated p_t a.s. have vanishing ℓ-loss regret relative to p_t^F and we have uniformly:

$$\frac{1}{T}\sum_{t=1}^{T} \ell(y_t, p_t) - \frac{1}{T}\sum_{t=1}^{T} \ell(y_t, p_t^F) < NB \sum_{j=1}^{M} \frac{T_j}{T} R_{T_j} + 3B\epsilon.$$

Proof (sketch). When p_t is the output of a given F_j, we have $\ell(y_t, p_t^F) \approx \ell(y_t, j/M) \approx \ell(y_t, i_j/M)$ (since p_t^F is in the j-th bucket, and since $M \ge N$ is sufficiently high resolution).

Subroutine	Regret Minimization	Blackwell Approchability
Time / step	$O(1/\epsilon)$	$O(\log(1/\epsilon))$
Space / step	$O(1/\epsilon^2)$	$O(1/\epsilon^2)$
Calibration	$O(1/\epsilon\sqrt{\epsilon T})$	$O(1/\epsilon\sqrt{T})$
Advantage	Simplicity	Efficiency

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

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 ℓ 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 ℓ may also be discontinuous (like the misclassification loss). When ℓ 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^{cal} preserves their calibration.

Lemma 3 (Preserving calibration)**.** *If each F_j^{cal} is (ϵ, ℓ_p)-calibrated, then Algorithm 1 is also (ϵ, ℓ_p)-calibrated and the bound $C_T \le \sum_{j=1}^{M} \frac{T_j}{T} R_{T_j} + \epsilon$ holds uniformly over T.*

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

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

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^F tends to $< 3B/N < \epsilon$. Hence, Theorem 1 follows. □

In the appendix, we provide a detailed argument for how ℓ can be chosen to be the misclassificaiton loss.

Interestingly, it also turns out that if ℓ is not a proper loss, then recalibration is not possible for some $\epsilon > 0$.

Theorem 2. *If ℓ is not proper, then no algorithm achieves recalibration w.r.t. ℓ for all $\epsilon > 0$.*

The proof of this algorithm is a slight generalization of the counter-example provided for the ℓ_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.

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^{cal}, $R_T \le f(\epsilon)/\sqrt{T}$ for some $f(\epsilon)$. In such cases, we can further

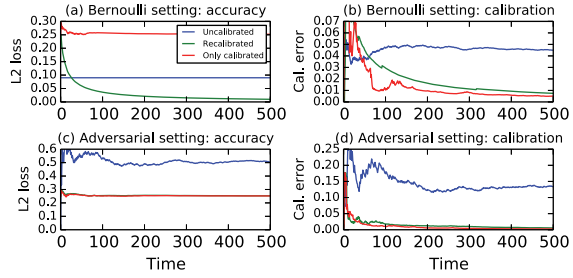

Figure 2: We compare predictions from an uncalibrated expert F (blue), Algorithm 1 (green), and REGMIN (red) on sequences $y_t \sim \text{Ber}(0.5)$ (plots a, b) and on adversarially chosen y_t (plots c, d).

bound the calibration error in Lemma 3 as $\sum_{j=1}^{M} \frac{T_j}{T} R_{T_j} \leq \sum_{j=1}^{M} \frac{\sqrt{T_j} f(\epsilon)}{T} \leq \frac{f(\epsilon)}{\sqrt{\epsilon T}}$. In the second inequality, we set the T_j to be equal.

Thus, our recalibration procedure introduces an overhead of $\frac{1}{\sqrt{\epsilon}}$ in the convergence rate of the calibration error C_T and of the regret in Lemma 2. In addition, Algorithm 1 requires $\frac{1}{\epsilon}$ times more memory (we run $1/\epsilon$ instances of F_j^{cal}), but has the same per-iteration runtime (we activate one F_j^{cal} per step). Table 1 summarizes the convergence rates of Algorithm 1 when the subroutine is either the method of Abernethy, Bartlett, and Hazan based on Blackwell approachability or the simpler but slower approach based on internal regret minimization (Mannor and Stoltz 2010).

Multiclass prediction. In the multiclass setting, we seek a recalibrator $A : \Delta_{K-1} \to \Delta_{K-1}$ producing calibrated probabilities $p_t \in \Delta_{K-1}$ that target class labels $y_t \in \{1, 2, ..., K\}$. In analogy to binary recalibration, we may discretize the input space Δ_{K-1} into a K-dimensional grid and train a classical multi-class calibration algorithm F^{cal} (Cesa-Bianchi and Lugosi 2006) on each subset of p_t^F associated with a cell. Just like in the binary setting, a classical calibration method F_j^{cal} predicts calibrated $p_t \in \Delta_{K-1}$ 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 (Hazan and Kakade 2012). However, there may exist practical workarounds inspired by popular heuristics for the batch setting, such as one-vs-all classification (Zadrozny and Elkan 2002).

Experiments

We now proceed to study Algorithm 1 empirically. Algorithm 1's subroutine is the standard internal regret minimization approach of Cesa-Bianchi and Lugosi ("REGMIN"). We measure calibration and accuracy in the ℓ_2 norm.

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 2, we compare the performance of REGMIN (which does not observe p_t^F) to Algorithm 1 and to the uncalibrated predictor F. Both methods achieve low calibration error after about 300 observations, while the expert is clearly uncalibrated (Figure 2b); however, REGMIN is a terrible predictor: it always forecasts $p_t = 0.5$ and therefore has high ℓ_2 loss (Figure 2a). 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 2 (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 (it reduces to REGMIN).

Natural language understanding. We used Algorithm 1 to recalibrate a state-of-the-art question answering system (Berant and Liang 2014) on the popular Free917 dataset (641 training, 276 testing examples). We trained the system on the training set as described in (Berant et al. 2013) and then calibrated probabilities using Algorithm 1 in one pass over first the training, and then the testing examples. This setup emulates a pre-trained system that further improves itself from user feedback.

Figure 3 (left) compares our predicted p_t to the raw system probabilities p_t^F via *calibration curves*. Given pairs of predictions and outcomes p_t, y_t, we compute for each of N buckets $B \in \{[\frac{i}{N}, \frac{i+1}{N}) \mid 0 \leq i \leq 1\}$, averages $\bar{p}_B = \sum_{t:p_t \in B} p_t / N_B$ and $\bar{y}_B = \sum_{t:p_t \in B} y_t / N_B$, where $N_B = |\{p_t \in B\}|$. A calibration curve plots the $\bar{y}_B$ as a function of $\bar{p}_B$; perfect calibration corresponds to a straight line.

Calibration curves indicate that the p_t^F are poorly calibrated in buckets below 0.9, while Algorithm 1 fares better. Figure 3a confirms that our accuracy (measured by the ℓ_2 loss) tracks the baseline forecaster.

Medical diagnosis. Our last task is predicting the risk of type 1 diabetes from genomic data. We use genotypes of 3,443 subjects (1,963 cases, 1,480 controls) over 447,221 SNPs (The Wellcome Trust Case Control Consortium 2007), with alleles encoded as $0, 1, 2$ (major, heterozygous and minor homozygous resp.). We use an online ℓ_1-regularized linear support vector machine (SVM) to predict outcomes one patient at a time, and report performance for each $t \in [T]$. Uncalibrated probabilities are normalized raw SVM scores s_t, i.e. $p_t^F = (s_t + m_t)/2m_t$, where $m_t = \max_{1 \leq r \leq t} |s_r|$.

Figure 3 (right) measures calibration after observing all the data. Raw scores are not well-calibrated outside of the interval $[0.4, 0.6]$; recalibration makes them almost perfectly

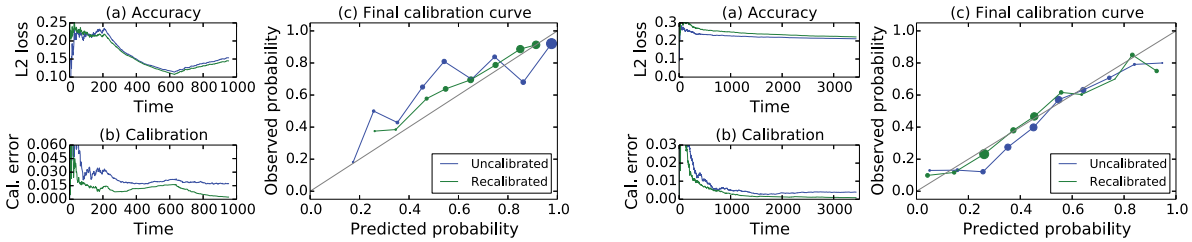

Figure 3: Algorithm 1 (green) is used to recalibrate probabilities from a question answering system (left) and a medical diagnosis system (right; both in blue). We track prediction (a) and calibration error (b) over time; plot (c) displays calibration curves after seeing all the data; circle sizes are proportional to the number of predictions in the corresponding bucket.

calibrated. Figure 3 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.

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 (Platt 1999) and isotonic regression (Niculescu-Mizil and Caruana 2005) are by far the most popular. Recalibration methods also possess multiclass extensions, which typically involve training multiple one-vs-all predictors (Zadrozny and Elkan 2002), as well as extensions to ranking losses (Menon et al. 2012), combinations of estimators (Zhong and Kwok 2013), and structured prediction (Kuleshov and Liang 2015).

In the online setting, the calibration problem was formalized by Dawid; online calibration techniques were first proposed by Foster and Vohra. Existing algorithms are based on internal regret minimization (Cesa-Bianchi and Lugosi 2006) or on Blackwell approachability (Foster 1997); recently, these approaches were shown to be closely related (Abernethy, Bartlett, and Hazan 2011; Mannor and Stoltz 2010). Recent work has shown that online calibration is PPAD-hard (Hazan and Kakade 2012).

The concepts of calibration and sharpness were first formalized in the statistics literature (Murphy 1973; Gneiting, Balabdaoui, and Raftery 2007). These metrics are captured by a class of *proper* losses and can be used both for evaluating (Buja, Stuetzle, and Shen 2005; Brocker 2009) and constructing (Kuleshov and Liang 2015) calibrated forecasts.

Discussion and Conclusion

Online vs batch. 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 $O(1/\sqrt{T_j})$ (Devroye, Györfi, and Lugosi 1996). This is faster than the $O(1/\sqrt{\epsilon T_j})$ rate of Abernethy, Bartlett, and Hazan and suggests that calibration is more challenging in the online setting.

Checking rules. An alternative way to avoid uninformative predictions (e.g. 0.5 on 010101...) is via the framework of *checking rules* (Cesa-Bianchi and Lugosi 2006). 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.

Defensive forecasting. Vovk, Takemura, and Shafer developed simultaneously calibrated and accurate online learning methods under the notion of *weak* calibration (Abernethy and Mannor 2011). We use strong calibration, which implies weak, although it requires different (e.g. randomized) algorithms. Vovk et al. also use a different notion of precision; their algorithm ensures a small difference between average predicted p_t and true y_t at times t when $p_t \approx p^*$ and $x_t \approx x^*$, for any p^*, x^*. The relation $\approx$ is determined by a user-specified kernel (over e.g. sentences or genomes x_t). Our approach, on the other hand, does not require specifying a kernel, and matches the accuracy of any given baseline forecaster; this may be simpler in some settings. Interestingly, we arrive at the same rates of convergence under different assumptions.

Conclusion. Current recalibration techniques implicitly require that the data is distributed i.i.d., which potentially makes them unreliable when this assumption does not hold. 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 not 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.

Acknowledgements. This work is supported by the NSF (grant #1649208) and by the Future of Life Institute (grant 2016-158687).

References

Abernethy, J. D., and Mannor, S. 2011. Does an efficient calibrated forecasting strategy exist? In *COLT 2011 - The 24th Annual Conference on Learning Theory, June 9-11, 2011, Budapest, Hungary*, 809–812.

Abernethy, J.; Bartlett, P. L.; and Hazan, E. 2011. Blackwell approachability and no-regret learning are equivalent. In *COLT 2011 - The 24th Annual Conference on Learning Theory*, 27–46.

Berant, J., and Liang, P. 2014. Semantic parsing via paraphrasing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, 1415–1425.

Berant, J.; Chou, A.; Frostig, R.; and Liang, P. 2013. Semantic parsing on freebase from question-answer pairs. In *EMNLP 2013*, 1533–1544.

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

Buja, A.; Stuetzle, W.; and Shen, Y. 2005. Loss functions for binary class probability estimation and classification: Structure and applications.

Cesa-Bianchi, N., and Lugosi, G. 2006. *Prediction, Learning, and Games*. New York, NY, USA: Cambridge University Press.

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

Devroye, L.; Györfi, L.; and Lugosi, G. 1996. *A probabilistic theory of pattern recognition*. Applications of mathematics. New York, Berlin, Heidelberg: Springer.

Foster, D. P., and Vohra, R. V. 1998. Asymptotic calibration.

Foster, D. P. 1997. A Proof of Calibration Via Blackwell's Approachability Theorem. Discussion Papers 1182, Northwestern University.

Gneiting, T.; Balabdaoui, F.; and Raftery, A. E. 2007. Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society: Series B* 69(2):243–268.

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

Jiang, X.; Osl, M.; Kim, J.; and Ohno-Machado, L. 2012. Calibrating predictive model estimates to support personalized medicine. *JAMIA* 19(2):263–274.

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

Mannor, S., and Stoltz, G. 2010. A geometric proof of calibration. *Math. Oper. Res.* 35(4):721–727.

Menon, A. K.; Jiang, X.; Vembu, S.; Elkan, C.; and Ohno-Machado, L. 2012. Predicting accurate probabilities with a ranking loss. In *29th International Conference on Machine Learning,*.

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

Nguyen, K., and O'Connor, B. 2015. Posterior calibration and exploratory analysis for natural language processing models. *CoRR* abs/1508.05154.

Niculescu-Mizil, A., and Caruana, R. 2005. Predicting good probabilities with supervised learning. In *Proceedings of the 22Nd International Conference on Machine Learning*, ICML '05.

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

Shalev-Shwartz, S. 2007. *Online Learning: Theory, Algorithms, and Applications*. Phd thesis, Hebrew University.

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

Vovk, V.; Takemura, A.; and Shafer, G. 2005. Defensive forecasting. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics, AISTATS 2005, Bridgetown, Barbados, January 6-8, 2005*.

Yu, D.; Li, J.; and Deng, L. 2011. Calibration of confidence measures in speech recognition. *Trans. Audio, Speech and Lang. Proc.* 19(8):2461–2473.

Zadrozny, B., and Elkan, C. 2002. Transforming classifier scores into accurate multiclass probability estimates. In *Eighth ACM Conference on Knowledge Discovery and Data Mining*, 694–699.

Zhong, L. W., and Kwok, J. T. 2013. Accurate probability calibration for multiple classifiers. IJCAI '13, 1939–1945. AAAI Press.