Calibration via Regression

Dean P. Foster
Statistics Department
University of Pennsylvania
Email: dean@foster.net

Sham M. Kakade
Toyota Technological Institute
Email: sham@tti-c.org

Abstract— **In the online prediction setting, the concept of *calibration* entails having the empirical (conditional) frequencies match the claimed predicted probabilities. This contrasts with more traditional online prediction goals of getting a low cumulative loss. The differences between these goals have typically made them hard to compare with each other. This paper shows how to get an approximate form of calibration out of a traditional online loss minimization algorithm, namely online regression. As a corollary, we show how to construct calibrated forecasts on a collection of subsequences.**

I. Introduction

Consider an online (sequential) prediction setting where, at each timestep t, the learner must predict some value $y_t \in [0,1]$ (it need not be binary) after being given some input from the set $x_t \in \mathcal{X}$ — the task is to accurately predict the next label y_t, given that we have observed the sequences $\{x_1, \dots x_t\}$ and $\{y_1, \dots y_{t-1}\}$. We study the case the where the sequence $\{(x_t, y_t)\}$ is arbitrary and make no statistical assumptions. There is a growing body of work showing that many of the learnable properties in an i.i.d. setting also hold in this adversarial setting.

Let us start by considering one such setting — the case of linear regression. For now, consider the case in which inputs are binary vectors, i.e. $x_t \in \{0,1\}^d$. Here, we predict with $\widehat{y}_t = \theta_t \cdot x_t$ at time t (where $\theta_t \in \mathcal{R}^d$) and suffer the instant loss $(\widehat{y}_t - y_t)^2$. For this case, even though the sequence is arbitrary, one can show that (minor variants of) ridge regression perform well in the online setting — these algorithms choose the parameter θ_t using a ridge regression algorithm on the prior sequence before time t (see [Foster(1991)], [Vovk(2001)], [Azoury and Warmuth(2001)], [Kakade and Ng(2004)]). They enjoy the guarantee that for all 'comparison' θ

$$\sum_{t=1}^{T}(\widehat{y}_t - y_t)^2 \leq \sum_{t=1}^{T}(\theta \cdot x_t - y_t)^2 + o(T)$$

i.e. these online ridge regression algorithms (asymptotically, on average) perform as well as the best single linear predictor.

Is this all we should expect from an online linear regression algorithm? To answer this, let us briefly reexamine the i.i.d. setting, where $(x,y) \sim \mathcal{D}$, to see an additional property that we might desire. Here, we chose the θ^* which minimizes

$$E_{\mathcal{D}}(\theta \cdot x - y)^2$$

By taking derivatives, we see that this θ^* has the property that for all i

$$E\left(x_i(\widehat{y} - y)\right) = 0 \qquad (1)$$

where $\widehat{y} = \theta^* \cdot x$. The above is the so called *normal* equations — the linear equations (in θ^*) used to solve for θ^*. For the case of binary $x \in \{0,1\}^d$, the normal equations state that the optimal predictions will be unbiased on those times when $x_i = 1$.

We might hope that in the online setting, for all i,

$$\left|\sum_{t=1}^{T} x_{t,i}(\widehat{y}_t - y_t)\right| = o(T) \qquad (2)$$

i.e. that (asymptotically, on average) our predictions are unbiased on the subsequence where $x_{t,i} = 1$. It turns out that standard online linear regression algorithms do not satisfy this unbiasedness property.[1]

The notion of calibration in essence tries to address issues of this form (see [Dawid(1984)])) — making calibrated forecasts involves having the empirical frequencies match their predictions, on average, under various checks. One way of formalizing this is as follows: a prediction rule is calibrated, with respect to a *test function* $f : [0,1] \to [0,1]$, if

$$\left|\sum_{t=1}^{T} f(\widehat{y}_t)(\widehat{y}_t - y_t)\right| = o(T)$$

The left hand side of this equation is referred to as the *calibration error* with respect to f. This is the notion put forth in [Kakade and Foster(2004)], which provided a (deterministic) algorithm which was calibrated with respect to *all* (Lipschitz) continuous test functions on all sequences. [2]

Suppose that we now want to have this weak form of calibration on a variety of subsequences. If we have d (potentially overlapping) subsequences that we actually care about, then we can define 2^d boolean combinations of these

[1] We provide an example of how they fail. Consider the case in which $x_t = 1$ always, so our predictions are $\widehat{y}_t = \theta_t \in \mathcal{R}$. Here, we show how the algorithm makes biased predictions on the entire sequence. Consider the sequence which is $y_t = 0$ for the first half and $y_t = 1$ for the second half. Common online regression algorithms essentially predict using the historical average. This works fine in terms of the square loss. However the bias in the predictions, $|\sum_{t=1}^{T}(\widehat{y}_t - y_t)|$, will be $\Omega(T)$.

[2] Stronger notions of calibration have been considered. The original notion of calibration allows one to use discontinuous test functions. Here, one needs to use randomization to prove existence of such calibrated algorithms. See [Foster and Vohra(1999)], [Kakade and Foster(2004)]

subsequences, which are disjoint. If we are calibrated on all 2^d disjoint subsequences, this implies the weaker statement that we are calibrated on the original d subsequences. In the linear regression setting (with binary inputs), to achieve calibration on the d subsequences, one could run a calibration algorithm separately on each of the 2^d possible settings of x_t and this would clearly guarantee the unbiased condition on each subsequence (i.e. it would satisfy Equation 2). However, the convergence rate would be be exponential in d.

Instead, this paper directly focuses on calibrating with respect to a few test functions. Let us extend the calibration definition to allow the test function to depend on auxiliary information (say the information provided by x_t). We do this by making the test functions time dependent — this time dependence implicitly allows dependence on events that occur before time t. The calibration condition with respect to f_t would then be

$$\left|\sum_{t=1}^{T} f_t(\widehat{y}_t)(\widehat{y}_t - y_t)\right| = o(T) \tag{3}$$

Again, the expression on the left hand side is referred to as the *calibration error* with respect to f_t. For example, if we choose d functions $f_{t,1}, f_{t,2}, \dots f_{t,d}$ such that $f_{t,i}(\widehat{y}) = x_{t,i}$ then calibrating with respect to these functions corresponds to being unbiased on each subsequence (as in Equation 2).

It is worthwhile to understand the relationship between this calibration condition and the square loss. If the calibration error is not sublinear in T, then one can show that there exists some β such that the following regret

$$\left[\sum_{t=1}^{T}(\widehat{y}_t - y_t)^2 - \sum_{t=1}^{T}(\widehat{y}_t + \beta f_t(\widehat{y}_t) - y_t)^2\right]_+$$

is not sublinear in T. [3] In other words, in retrospect, had we predicted $\widehat{y}_t + \beta f_t(\widehat{y}_t)$ instead of $\widehat{y}_t$ then our loss would have been significantly lower. Roughly speaking, the calibration condition stipulates that the test function f_t must be uncorrelated (i.e. orthogonal) to the error $\widehat{y}_t - y_t$. If the two were correlated (i.e. if the calibration condition were not satisfied), then this means that adding in some amount of $f_t(\widehat{y}_t)$ to our predictions $\widehat{y}_t$ would have improved our performance.

We focus on how to (quickly) calibrate with respect to a finite set of functions $f_{t,1}, f_{t,2}, \dots f_{t,d}$. The motivation is that we might have some tests for which we care to be unbiased on. For example, as discussed earlier, the choice $f_{t,i}(\widehat{y}) = x_{t,i}$ corresponds to a preference to satisfy the online analogue of the normal equations (Equation 1). Alternatively, we could chose the functions to be low order polynomials of $\widehat{y}$. This corresponds to the desire that, in retrospect, there should not exist a low-order polynomial transformation of our predictions that improve our performance.

Our algorithm uses a simple modification on the pre-existing machinery of online regression, and we show that this is sufficient to calibrate with respect to these functions. [Vovk(2005)] also considers a this setting and shows how to calibrate using kernels corresponding to an RKHS — the main differences are that we focus on how to calibrate on a finite set of (preferred) functions and our algorithm is just a simple variant of ridge regression. The basic idea of our algorithm is to consider our past predictions when making future predictions — to let the regression algorithm determine how our predictions themselves may be correlating with the prediction error. Our main results show that for any set of test functions $\{f_{t,1}(\cdot), f_{t,2}(\cdot), \dots f_{t,d}(\cdot)\}$ (bounded in $[0,1]$), there is an algorithm such that for all i:

$$\left|\sum_{t=1}^{T} f_{t,i}(\widehat{y}_t)(\widehat{y}_t - y_t)\right| \le O(\sqrt{Td\ln T})$$

which is sublinear in T as desired.

The remainder of the paper is organized as follows. As regression is a tool in our algorithm, we start with a theorem from online regression. Then we present our main result on how to calibrate on a finite set of functions (and we briefly discuss how to (asymptotically) achieve calibration on all (Lipschitz) continuous test functions). We then return to the case of linear regression and show how to satisfy the normal equations. We briefly discuss the i.i.d. case and differences to the online setting.

II. Online Regression

We now state a theorem from Azoury and Warmuth (2001) on regression in the online setting (Theorem 4.6 in [Azoury and Warmuth(2001)]). The ridge regression algorithm is shown in Algorithm 1. The algorithm takes as input some x_t (which need not be binary, but, for simplicity, we restrict it to be in the unit interval) and at each step it outputs $\widehat{y}_t$ (which we always take to be in $[0,1]$). Since a linear predictor may sometimes output a value out of the range $[0,1]$, the algorithm clips the output so that it always predicts in the range $[0,1]$.

The following theorem bounds the performance of the algorithm in terms of the performance of a constant linear predictor.

Theorem 2.1: : [Azoury and Warmuth(2001)] For all sequences $\{x_t\}$ (such that $||x_t|| < 1$) and $\{y_t\}$ (bounded in $[0,1]$), and for all θ, the performance of Algorithm 1 is bounded as follows:

$$\sum_{t=1}^{T}(\widehat{y}_t - y_t)^2 \le \sum_{t=1}^{T}(\theta \cdot x_t - y_t)^2 + ||\theta||^2 + 2d\ln(T+1)$$

where $||\cdot||$ denotes the ℓ_2 norm.

Similar theorems have appeared in [Foster(1991)], [Vovk(2001)], [Kakade and Ng(2004)].

[3] To see this, let $R = -\sum_{t=1}^{T} f_t(\widehat{y}_t)(\widehat{y}_t - y_t)$. One can show that the regret is bounded by to $[2\beta R - \beta^2 T]_+$, using $\sum_{t=1}^{T} f_t^2(\widehat{y}_t) \le T$. Now if $|R|$ is not sublinear in T one can choose a sufficiently small value of $|\beta|$ (β could be negative) such that this regret is above δT (for some $\delta > 0$) infinitely often, so the regret will not be sublinear it T.

```
Input  : At each time $t$, we receive $x_t \in [0,1]^d$.
Output: At each time $t$, we predict $\widehat{y}_t \in [0,1]$
for $t = 1, 2, \dots T$ do
    1. Set $\theta_t$ to be
       $$\theta_t = \mathrm{argmin}_\theta \left\{ \sum_{\tau=1}^{t-1} (\theta \cdot x_\tau - y_\tau)^2 + ||\theta||^2 \right\}$$
    2. Predict with
       $$\widehat{y}_t = \mathrm{clip}(\theta_t \cdot x_t)$$
       where $\mathrm{clip}()$ clips the input to be in $[0,1]$, i.e. it
       outputs the point in $[0,1]$ closest to the input.
end
```

Algorithm 1: Online Linear Ridge Regression

```
Input  : At time $t$, we have $f_t : [0,1] \to [0,1]^{d+1}$.
Output: At each time $t$, we predict $\widehat{y}_t \in [0,1]$
for $t = 1, 2, \dots T$ do
    1. Set $\theta_t$ to be
       $$\theta_t = \mathrm{argmin}_\theta \left\{ \sum_{\tau=1}^{t-1} (\theta \cdot f_\tau(\widehat{y}_\tau) - y_\tau)^2 + ||\theta||^2 \right\}$$
    2. Predict with any $\widehat{y}_t$ such that
       $$\widehat{y}_t = \mathrm{clip}(\theta_t \cdot f_t(\widehat{y}_t))$$
       where $\mathrm{clip}()$ clips the input to be in $[0,1]$, i.e. it
       outputs the point in $[0,1]$ closest to the input point.
end
```

Algorithm 2: Calibrated Regression

III. Calibrating with Test Functions

Now consider the test functions $f_{t,1}(\widehat{y}), f_{t,1}(\widehat{y}), \dots f_{t,d}(\widehat{y})$. Just performing online regression, where we regress off of these variables, would not be enough to satisfy our calibration conditions in general. For instance, as discussed in the Introduction, if we choose $f_{t,i} = x_{t,i}$, then regression off of f_t corresponds to linear regression, which does not calibrate with respect to x_t (i.e. Equation 2 is not satisfied in general). However, consider including a function $f_{t,0}(\widehat{y}) = \widehat{y}$, for all t. It turns out with this simple modification, the online regression algorithm is sufficient to obtain our calibration goals.

However, note that that regressing off of these $f_t(\widehat{y})$ is rather subtle, since these functions depend on the predictions themselves. This means at the time of prediction, a fixed point condition must be solved (though this can be done with a one dimensional line search, if not analytically). In the case of linear regression (treated in the next section) this fixed point condition can be solved analytically.

For notational convenience, we will write $f_t(\widehat{y}) = (f_{t,0}(\widehat{y}), f_{t,1}(\widehat{y}), \dots f_{t,d}(\widehat{y}))$, so f_t is really a $d{+}1$ dimensional vector.

A. A Calibrated Regression Algorithm

Algorithm 2 is the procedure used to calibrate with respect to these test functions. Intuitively, the algorithm is trying to regress off of the the predictions it makes in order to achieve the calibration condition. This involves solving a fixed point equation, in Step 2.

We now show this fixed point exists.

Theorem 3.1:: If f is continuous, then the algorithm exists, i.e. a fixed point $\widehat{y}_t$ exists in Step 2.

Proof: First, if there exists a $\widehat{y}_t \in [0,1]$ such that $\widehat{y}_t = \theta_t \cdot f_t(\widehat{y}_t)$, then we are done, since clipping does not alter this prediction. So assume this is not the case. In other words, assume that the function $\theta_t \cdot f_t(\widehat{y})$ does not cross the function $g(\widehat{y}) = \widehat{y}$ in the interval $\widehat{y} \in [0,1]$, else the crossing point would be a fixed point. Since $f_t(\cdot)$ is continuous, then either $\theta_t \cdot f_t(\cdot)$ must lie completely above or completely below the curve $g(\cdot)$. If it lies below the curve, then $\theta_t \cdot f_t(0) < g(0) = 0$. Hence, for this case, $\widehat{y}_t = 0$ is a fixed point, since $0 = \mathrm{clip}(\theta_t \cdot f_t(0))$. Similarly, if $\theta_t \cdot f_t(\cdot)$ lies above $g(\cdot)$, then $\widehat{y}_t = 1$ is a fixed point. ■

B. Convergence Rates

The following Corollary bounds the performance of the algorithm.

Corollary 3.2:: For all sequences $\{\widehat{y}_t\}$ and continuous $\{f_t\}$ (both bounded in $[0,1]$), and for all θ, the performance of Algorithm 2 is bounded as follows:

$$\sum_{t=1}^{T} (\widehat{y}_t - y_t)^2 \leq \sum_{t=1}^{T} (\theta \cdot f_t(\widehat{y}_t) - y_t)^2 + ||\theta||^2 + 2(d+1)\ln(T+1)$$

Proof: The proof follows directly from Theorem 2.1. To see this, set $x_t = f_t(\widehat{y}_t)$ in Algorithm 1. The fixed point condition in Step 2 assures that this is consistent. ■

Using this, we can state our main theorem.

Theorem 3.3:: Let $f_t(\widehat{y}) = (f_{t,0}(\widehat{y}), f_{t,1}(\widehat{y}), \dots f_{t,d}(\widehat{y})$ be continuous test functions (bounded in $[0,1]$) such that $f_{t,0}(\widehat{y}) = \widehat{y}$. Let $\{\widehat{y}_t\}$ be bounded in $[0,1]$ and define

$$\tau = \sum_{t=1}^{T} f_{t,i}^2(\widehat{y}_t) + 1$$

Algorithm 2 has the following bound on the calibration error for all test functions $i = 1, 2, \dots d$

$$\begin{aligned} \left| \sum_{t=1}^{T} f_{t,i}(\widehat{y}_t)(\widehat{y}_t - y_t) \right| &\leq 2\sqrt{\tau(d+1)\ln(T+1))} \\ &\leq 2\sqrt{(T+1)(d+1)\ln(T+1)} \end{aligned}$$

Proof: Consider setting $\theta_0 = 1$ and $\theta_i = \beta$, where $i \neq 0$ and the remainder of the components to be 0. By Corollary 3.2

$$\sum_{t=1}^{T} (\widehat{y}_t - y_t)^2 \leq \sum_{t=1}^{T} (\widehat{y}_t + \beta f_{t,i}(\widehat{y}_t) - y_t)^2 + 1 + \beta^2 + 2(d+1)\ln(T+1) \quad (4)$$

since $f_0(\widehat{y}) = \widehat{y}$ and $||\theta||^2 = 1 + \beta^2$. Now consider the β which minimizes the right hand side. It is straightforward to show (by setting the first derivative equal to 0) that this β is just

$$\beta = \frac{-\sum_{t=1}^{T} f_{t,i}(\widehat{y}_t)(\widehat{y}_t - y_t)}{\sum_{t=1}^{T}(f_{t,i}(\widehat{y}_t))^2 + 1} \equiv \frac{R}{\tau}$$

where R and τ are are defined to be the numerator and denominator of this expression. We seek to bound R.

Simple algebra leads to:

$$\sum_{t=1}^{T}(\widehat{y}_t + \beta f_{t,i}(\widehat{y}_t) - y_t)^2 + \beta^2 = \sum_{t=1}^{T}(\widehat{y}_t - y_t)^2 - \frac{R^2}{\tau}$$

where we have used the definitions of R and τ. Using Equation 4, this implies:

$$\frac{R^2}{\tau} \leq 1 + 2(d+1)\ln(T+1) \leq 4(d+1)\ln(T+1)$$

Hence, we have that R is bounded by $2\sqrt{(d+1)\tau\ln(T+1)}$. Noting that $\tau \leq T + 1$ completes the proof. ■

C. Asymptotic Calibration

The previous algorithm can also be used for (asymptotic) calibration — meaning we can drive the calibration error to 0 for all (Lipschitz) continuous test functions. To see this, just consider using a (countable) sequence of test functions $f^1(\cdot), f^2(\cdot), \ldots$ (superscripts are used since these are not functions of time) such that these test functions form a basis for all (Lipschitz) continuous functions. Then just add in more test functions (sufficiently slowly) such that the calibration error is forced to 0 for an any of these basis functions.

IV. Calibrating on Subsequences

Now consider the case where at each time we receive binary $x_t \in \{0,1\}^d$. Let us examine how we can modify the online regression algorithm such that the predictions are unbiased on each of the subsequences i, where $x_{t,i} = 1$, i.e.

$$\left|\sum_{t=1}^{T} x_{t,i}(\widehat{y}_t - y_t)\right| = o(T)$$

This is the analogue of the normal equations (Equation 1) in the online setting. This goal naturally generalizes to the case where x_t is non-binary. As discussed in the Introduction, simply running an online linear regression algorithm (such as Algorithm 1) is not sufficient to achieve this unbiased condition.

However, the Calibrated Regression algorithm (Algorithm 2) presents a simple fix, just add one more regression variable, namely $\widehat{y}$. More formally, set $f_{t,0}(\widehat{y}_t) = \widehat{y}_t$ and set $f_{t,i}(\widehat{y}_t) = x_{t,i}$ for $i = 1, 2 \ldots d$.

Note that for prediction the fixed point condition (without the clipping) is just $\widehat{y}_t = \theta \cdot f_t(\widehat{y}_t)$, which is equivalent to:

$$\widehat{y}_t = \frac{\sum_{i=1}^{d} \theta_i x_{t,i}}{1 - \theta_0}$$

If this $\widehat{y}_t \in [0,1]$, then we predict with this value. Else, we must chose either 0 or 1, whichever solves the fixed point condition $\widehat{y}_t = \text{clip}(\theta \cdot f_t(\widehat{y}_t))$.

This algorithm enjoys the following performance bound.

Corollary 4.1:: The Calibrated Regression algorithm, with $f_{t,0}(\widehat{y}_t) = \widehat{y}_t$ and $f_{t,i}(\widehat{y}_t) = x_{t,i}$ for $i = 1, 2 \ldots d$, achieves the following bound on the calibration error, for each i

$$\left|\sum_{t=1}^{T} x_{t,i}(\widehat{y}_t - y_t)\right| \leq 2\sqrt{(T_i + 1)(d+1)\ln(T+1)}$$

where T_i is the total length of subsequence i, i.e. the number of times $x_{t,i} = 1$.

Proof: The proof follows by noting that for $i \neq 0$ $\sum_{t=1}^{T} f_{t,i}^2(\widehat{y}_t) = \sum_{t=1}^{T} x_{t,i}^2 = T_i$. ■

Hence, this bound shows that we can simultaneously bound the bias on all subsequences, such that the bias on each subsequence grows as square root the number of times that sequence was present.

A. Asymptotic Calibration on Subsequences

Now let us consider how to calibrate on each subsequence. By this, we desire that, on each subsequence, we desire that the calibration condition be satisfied for all (Lipschitz) continuous test functions. Recall for obtaining calibration on one sequence we considered using a (countable) sequence of test functions $f^1(\cdot), f^2(\cdot), \ldots$ such that these test functions form a basis for all continuous functions. Here, we consider d sequences, each of the form $x_{t,i}f^1(\cdot), x_{t,i}f^2(\cdot), \ldots$. As before, we slowly add more of these test functions.

V. Why not for the IID case also?

Using the strong similarity of the IID case to the individual sequence case, we are lead to ask what would happen if we added a $\hat{y}$ to the right hand side of a regression problem. But since this is a cross-sectional setting, the $\hat{y}$ is always a fixed linear combination of the other x's. So it doesn't change the subspace spanned by the original list of x's–it is already there. One can think of this as being the reason that we get the unbiased result of equation (1) for free.

But we also considered looking at higher powers of $\hat{y}$. In the IID case this can be thought of as estimating the link function h, where $Ey = h(\beta \cdot x)$. A variant of this has been useful in an applied setting, namely that of bankruptcy from credit card data [Foster and Stine(2004)].

References

[Azoury and Warmuth(2001)] K. S. Azoury and M. Warmuth. Relative loss bounds for on-line density estimation with the exponential family of distributions. *Machine Learning*, 43(3), 2001.

[Dawid(1984)] A. Dawid. Statistical theory: The prequential approach. *J. Royal Statistical Society*, 1984.

[Foster(1991)] D. P. Foster. Prediction in the worst case. *Annals of Statistics*, 19, 1991.

[Foster and Stine(2004)] Dean P. Foster and Robert A. Stine. Variable selection in data mining: Building a predictive model for bankruptcy. *JASA*, 99:303–313, 2004.

[Foster and Vohra(1999)] Dean P. Foster and Rakesh V. Vohra. Regret in the on-line decision problem. *Games and Economic Behavior*, pages 7 – 36, 1999.

[Kakade and Ng(2004)] S. M. Kakade and A. Y. Ng. Online bounds for bayesian algorithms. *Proceedings of Neural Information Processing Systems*, 2004.

[Kakade and Foster(2004)] Sham M. Kakade and Dean P. Foster. Deterministic calibration and nash equilibrium. *The Seventeenth Annual Conference on Learning Theory (COLT)*, 2004.

[Vovk(2001)] V. Vovk. Competitive on-line statistics. *International Statistical Review*, 69, 2001.

[Vovk(2005)] V. Vovk. Non-asymptotic calibration and resolution. *Algorithmic Learning Theory, 16th International Conference, ALT 2005*, 2005.