

Modern Paradigms in Learning and Decision-Making

Lecture #6: Reductions between Predictions and Decision Making*

Juan C. Perdomo[†]

Fall 2026

1 Introduction

In the last few lectures, we introduced the problem of online multicalibration. We analyzed a simple algorithm that achieved it for the special setting where functions in the class $\mathcal{C}$ could be written as

$$c(x, p) = \langle \Phi(x, p), \theta \rangle$$

where $\Phi : \mathcal{X} \times [0, 1] \rightarrow \mathbb{R}^d$ was a known feature map.

In this lecture, we will use the online learning tools we developed last time to achieve multicalibration with respect to much richer classes of functions $\mathcal{C}$. We will then illustrate how a similar idea can be used to quantify uncertainty in black-box prediction systems. To do so, we start by taking a small detour into a recently introduced class of optimization problems known as Expected Variational Inequalities.

2 Expected Variational Inequalities

In EVIs, we are given a compact, convex set $\mathcal{K}$ and a bounded operator $S : \mathcal{K} \rightarrow \mathbb{R}^d$, and a tolerance parameter ε . The goal is to find a distribution $\mathcal{D}$ over points in $\mathcal{K}$ such that for every $p' \in \mathcal{K}$

$$\mathbf{E}_{p \sim \mathcal{D}} S(p)^\top (p - p') \leq \varepsilon.$$

That is, we want to find a distribution over points such that we guarantee negative correlation with respect to every point p' in $\mathcal{K}$. EVIs are a powerful optimization framework that generalize important game-theoretic questions, such as the computation of (coarse) correlated equilibria.

However, for now, notice how this primitive generalizes the procedure we saw in the Defensive Forecasting lecture to higher dimensional problems. If we let $\mathcal{K} = [0, 1]$ and let $S(p) = \sum_{i=1}^{t-1} \Phi(x_i, p) \Phi(x_i, p_i) (y_i - p_i)$, we learned in the Defensive Forecasting lecture [PR25] how to solve this problem via a reduction to binary search. If $S(p)$ was continuous in p , we could show that it had a root in the unit interval. Any such root $p_\star$ would satisfy $S(p_\star) = 0$ and hence solve the EVI.

This was neat, but the techniques do not directly generalize to functions $S(p)$ that are vector-valued since root-finding is not nearly as straightforward in higher dimensions.

*These notes are a work in progress and have not been subjected to the usual scrutiny reserved for formal publications. Please email me if you see any mistakes or find something confusing.

[†]© 2026, Juan C. Perdomo. Not to be sold, published, or distributed without the authors' consent.

2.1 Solving EVIs via Online Learning.

Luckily, Zhang et al. [ZAT⁺25] introduced a way to efficiently solve these EVIs in higher dimensions via a reduction to online linear optimization. Recall that in OLO, the goal is: Given a convex set $\mathcal{K}$ produce a sequence of elements $p_t \in \mathcal{K}$ such that for any adaptive sequence of losses $\{\ell_t\}_{t=1}^T$

$$\sup_{p_\star \in \mathcal{K}} \sum_{t=1}^T \ell_t^\top p_t - \sum_{t=1}^T \ell_t^\top p_\star \leq o(T)$$

If we have access to an online algorithm $\mathcal{A}$ that can solve OLO over $\mathcal{K}$, then we also have an algorithm that can solve EVIs for any bounded S . In particular, consider the following procedure run for T rounds

1. Choose decision $p_t \in \mathcal{K}$ using $\mathcal{A}$
2. Define the loss $\ell_t = S(p_t)$ and feed it back to the online algorithm.

Notice how we are exploiting the fact that in OLO, Nature can be adversarial (they can pick any loss ℓ_t) and adaptive (it can pick ℓ_t as a function of p_t).

The reduction works for any no-regret algorithm, but for concreteness assume we ran online gradient descent. Then, after T rounds, we would have that:

$$\sup_{p_\star \in \mathcal{K}} \sum_{t=1}^T \ell_t^\top p_t - \sum_{t=1}^T \ell_t^\top p_\star \leq \frac{3}{2}GD\sqrt{T},$$

where $G = \sup_p \|S(p)\|_2$ is a bound on the norm of the losses and $D = \sup_{p, p' \in \mathcal{K}} \|p - p'\|_2$ is a bound on the diameter of the set $\mathcal{K}$.

If we now set $\mathcal{D}$ to be the uniform distribution over the iterates $\{p_1, \dots, p_T\}$, the above regret guarantee implies that for any $p' \in \mathcal{K}$

$$\begin{aligned} \mathbf{E}_{p \sim \mathcal{D}} S(p)^\top (p - p') &= \frac{1}{T} \sum_{t=1}^T S(p_t)^\top p_t - S(p_t)^\top p' \\ &= \frac{1}{T} \sum_{t=1}^T \ell_t^\top p_t - \ell_t^\top p' \\ &\leq \frac{3}{2} \cdot \frac{GD}{\sqrt{T}}. \end{aligned}$$

Hence, if we wanted to solve the EVI to error ε it suffices to run online gradient descent for $\mathcal{O}(GD/\varepsilon^2)$ many iterations (different OCO algorithms will give you slightly different dependence on parameters like G, D).

This approach is powerful in that it works for *any* function S that is bounded. We did not need to assume any further regularity properties. It is in practice, however, somewhat slow if you need ε to be very small. Zhang et al. [ZAT⁺25] also introduce a faster algorithm based on the ellipsoid which addresses this issue and runs in time $\mathcal{O}(\log(1/\varepsilon))$. I hope to write some lecture notes about it someday.

3 A Black-Box Reduction to Multicalibration

Now we have the two main ingredients (EVIs and OLO) in place to guarantee online multicalibration with respect to general function classes and convex outcomes as defined below:

Let $\mathcal{X}$ be a feature set, $\mathcal{Y} \subseteq \mathbb{R}^d$ a compact convex set of outcomes, and $\mathcal{H} \subseteq \{h : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}^d\}$ a class of vector-valued functions. Consider the online protocol where at each round $t = 1, \dots, T$:

1. Nature first reveals features $x_t \in \mathcal{X}$;
2. Then, the Learner outputs a distribution $\mathcal{D}_t$ over forecasts $p_t \in \mathcal{Y}$;
3. Lastly, Nature reveals an outcome $y_t \in \mathcal{Y}$ (perhaps as a function of $\mathcal{D}_t$)

Given a function $h \in \mathcal{H}$, we define the multicalibration error of h as:

$$\text{MC}_T(h) := \sum_{t=1}^T \mathbf{E}_{p_t \sim \mathcal{D}_t} [\langle h(x_t, p_t), y_t - p_t \rangle] \quad (1)$$

The Learner is online multicalibrated with respect to $\mathcal{H}$ if $\text{MC}_T := \sup_{h \in \mathcal{H}} \text{MC}_T(h) = o(T)$ regardless of Nature's choices.¹

Relative to the setup we saw before, we allow the Learner to output a distribution over predictions $\mathcal{D}_t$ instead of forcing them to commit to a single prediction p_t . This is necessary if we allow functions $h(x, p)$ to be discontinuous in p as per the following example.²

Example 3.1 (discontinuous tests). *Let $\mathcal{Y} = [0, 1]$ and consider the case where $\mathcal{H}$ is equal to*

$$\mathcal{H} = \{\mathbf{1}\{p \geq 1/2\}, -\mathbf{1}\{p < 1/2\}\}.$$

Assume that given p_t , Nature picks $y_t = 0$ if $p_t \geq 1/2$ and $y_t = 1$ if $p_t < 1/2$. Then at every round, for every choice of p_t

$$\min\{1\{p_t \geq 1/2\}(p_t - y_t), -1\{p_t < 1/2\}(p_t - y_t)\} \geq \frac{1}{2}$$

Therefore, $\text{MC}_T(h) \geq T/2$ for at least one of the functions. By randomizing the prediction, we can get around this problem. (Try deriving such a strategy as an exercise for this example).

In Algorithm 1, we present a general black-box recipe for achieving online multicalibration over the class $\mathcal{H}$ developed in [FP26]. It assumes access to two subroutines: an EVI solver, and an online linear optimizer over the set of functions in $\mathcal{H}$:

¹In this definition, we have no absolute values. However, this is without loss of generality if for every $h \in \mathcal{H}$, $-h$ is also in the class.

²We ruled out discontinuities in the previous example since we assumed that $\Phi(x, p)$ was continuous in p .

Algorithm 1: Black-box Reduction from Online Learning over $\mathcal{H}$ to $\mathcal{H}$ -multicalibration

Input: an online learner $\mathcal{A}$ over $\mathcal{H}$, EVI error tolerances ε_t

For $t = 1, 2, \dots$

1. Nature reveals x_t .
2. The online learner $\mathcal{A}$ outputs $h_t \in \mathcal{H}$.
3. The Learner outputs the distribution over forecasts $\mathcal{D}_t$ where $\mathcal{D}_t$ is a solution to the following EVI over $S_t := h_t(x_t, \cdot)$

$$\mathbf{E}_{p \sim \mathcal{D}_t} [\langle h_t(x_t, p), y - p \rangle] \leq \varepsilon_t \quad \text{for all } y \in \mathcal{Y}. \quad (2)$$

4. Nature reveals y_t , and $\mathcal{A}$ receives the loss

$$f_t(h) := - \mathbf{E}_{p_t \sim \mathcal{D}_t} [\langle h(x_t, p_t), y_t - p_t \rangle]. \quad (3)$$

Intuitively, the Online Learner $\mathcal{A}$ plays the role of an Auditor that selects the function h_t that exhibits the biggest calibration errors over the observed trajectory. Very informally,

$$h_t \approx \operatorname{argmax}_{h \in \mathcal{H}} \sum_{i=1}^{t-1} \mathbf{E}_{p_i \sim \mathcal{D}_i} h(x_i, p_i)^\top (y_i - p_i)$$

The EVI solver then picks a distribution over predictions, such that no matter what the outcome is, we have small error with respect to that auditor function. We prove the correctness of the algorithm in the following theorem.

Theorem 3.2 ([FP26]). *Fix $h \in \mathcal{H}$ and let*

$$\operatorname{Regret}_T(h) := \sum_{t=1}^T f_t(h_t) - f_t(h) \quad \text{and} \quad \operatorname{EVI}_T := \sum_{t=1}^T \varepsilon_t$$

denote the regret of $\mathcal{A}$ on the losses f_t defined in Algorithm 1, and the sum of the EVI errors in (2) respectively. Then, the distributions $\mathcal{D}_t$ output by the Learner in Algorithm 1 satisfy

$$\operatorname{MC}_T(h) \leq \operatorname{Regret}_T(h) + \operatorname{EVI}_T.$$

Proof. The crux of the proof is leveraging the EVI guarantee (2) which states that for any outcome $y \in \mathcal{Y}$ (and hence, in particular, for Nature's choice of y_t), we have that:

$$0 \leq \varepsilon_t - \mathbf{E}_{p_t \sim \mathcal{D}_t} [h_t(x_t, p_t)^\top (y_t - p_t)].$$

Therefore, plugging in the definition of regret:

$$\begin{aligned}
\sum_{t=1}^T \mathbf{E}_{p_t \sim \mathcal{D}_t} [h(x_t, p_t)^\top (y_t - p_t)] &\leq \sum_{t=1}^T \mathbf{E}_{p_t \sim \mathcal{D}_t} [h(x_t, p_t)^\top (y_t - p_t)] - \mathbf{E}_{p_t \sim \mathcal{D}_t} [h_t(x_t, p_t)^\top (y_t - p_t)] + \varepsilon_t \\
&= \left(\sum_{t=1}^T f_t(h_t) - f_t(h) \right) + \sum_{t=1}^T \varepsilon_t \\
&= \text{Regret}_T(h) + \text{EVI}_T. \quad \square
\end{aligned}$$

Algorithm 1 is not really an algorithm as much as it is a recipe book for many different algorithms to achieve online multicalibration with respect to high-dimensional vector-valued outcomes. We will now discuss a few example instantiations. Throughout, we fix the EVI error $\varepsilon_t = 1/(10t^2)$ so that $\text{EVI}_T = \sum_{t=1}^T \varepsilon_t \leq \sum_{t=1}^\infty 1/(10t^2) \leq 1$.

3.1 Instantiations

Multiplicative Weight Updates (Hedge). Let $\mathcal{H} = \{h^{(1)}, \dots, h^{(n)}\}$ be a finite set of functions normalized so that $|\langle h^{(i)}(x, p), y - p \rangle| \leq 1$ for all i, x, y, p . We can run Algorithm 1 with $\mathcal{A}$ equal to MWU over the n experts where the loss of expert i at time step t is equal to³

$$\ell_{t,i} = f_t(h^{(i)}) + 1 = 1 - \mathbf{E}_{p_t \sim \mathcal{D}_t} h^{(i)}(x_t, p_t)^\top (y_t - p_t) \in [0, 2].$$

MWU maintains weights $\lambda_t \in \Delta_n$ and outputs the average test $h_t := \sum_i \lambda_{t,i} h^{(i)} \in \text{conv}(\mathcal{H})$. The weights are updated in proportion to the Expert's calibration error at each time step:

$$\lambda_{t+1,i} \propto \lambda_{t,i} \exp \left(\eta \mathbf{E}_{p_t \sim \mathcal{D}_t} [\langle h^{(i)}(x_t, p_t), y_t - p_t \rangle] \right).$$

Since $\ell_{t,i}^2 \leq 4$, the Hedge guarantee with $\eta = \sqrt{\log n / (4T)}$ gives regret at most $4\eta T + \log n / \eta = 4\sqrt{T \log n}$ against every expert. Therefore, for every $h \in \mathcal{H}$,

$$\text{MC}_T(h) \leq 4\sqrt{T \log n} + 1.$$

Online Gradient Descent. For $\mathcal{H} = \{h(x, p) = \Gamma(x, p)\theta\}$ where $\Gamma(x, p) \in \mathbb{R}^{d \times p}$ is a matrix-valued function and $\theta \in \mathbb{R}^p$ are vectors with norm at most B , we get that the losses in Algorithm 1 can be written as:

$$f_t(h) = - \mathbf{E}_{p_t \sim \mathcal{D}_t} (y_t - p_t)^\top \Gamma(x_t, p_t)^\top \theta = \ell_t^\top \theta$$

Online Gradient Descent would therefore achieve

$$\text{MC}_T(h) \leq \frac{3}{2} G D \sqrt{T} + 1.$$

where $G = \sup_{x,p,y} \|\Gamma(x, p)(y - p)\|_2$ and $D = \sup \|\theta\|_2 \leq B$.

³The added 1 is to make the loss nonnegative.

3.2 Further Applications.

Since the reduction is black-box, it allows us to draw on a rich set of tools and algorithms that have been developed within the extensive literature on online linear optimization to design sequential prediction algorithms with provable guarantees. One particular such connection is that it enables us to design prediction algorithms that handle *delayed observations*.

In the setup we’ve discussed so far, the Learner immediately observes y_t after issuing their predictions $\mathcal{D}_t$. However, imagine that we need to make many predictions before we see the outcome for any data point. In the language of the reduction, this amounts to having the online learner observe the losses $f_t(h)$ associated with round t at some later round $t' > t$. By plugging in algorithms that guarantee low regret even with delayed losses, we use the recipe to design online algorithms that can handle delayed observations. See the Cesa-Bianchi and Lugosi [CBL06] and [FP26] for more details.

4 Uncertainty Quantification & Quantile Regression

In the previous section, we demonstrated how algorithms for online decision-making (online linear optimization) could be used to design algorithms for sequential prediction. Eventually, we will go back and show how algorithms for sequential prediction can be used in a similar black-box fashion to produce online decision-making algorithms with even stronger guarantees (Φ -regret minimization). However, here we take a slight detour and show the techniques we’ve developed so far can be used to quantify uncertainty in black-box ML models.

To motivate the setup, assume we have access to a regression model $f_\star : \mathcal{X} \rightarrow \mathbb{R}$ which makes predictions. The goal is to quantify the uncertainty in the model. Given features x_t , we want to produce an interval $[\hat{y}_{\min}, \hat{y}_{\max}]$ such that $f_\star(x) \in [\hat{y}_{\min}, \hat{y}_{\max}]$ with high probability.

To do so, we can reduce the problem to quantile prediction. There are many ways to do this,⁴ but for simplicity, let $e_t = |y_t - f_\star(x_t)|$. Given a confidence level α , if we could predict the $1 - \alpha$ quantile on e_t then we could turn that into a confidence interval. More precisely, if we can produce forecasts p_t such that $\Pr[e_t \geq p_t] = 1 - \alpha$, then the prediction interval $[f_\star(x_t) - p_t, f_\star(x_t) + p_t]$ will contain the true y_t exactly a $1 - \alpha$ fraction of the time.

In this section, we extend the machinery from before to solve online, multicalibrated quantile regression. In particular, we consider the following online protocol

1. Nature reveals features x_t .
2. The learner produces a distribution $\mathcal{D}_t$ over forecasts.
3. Nature reveals the outcome $y_t \in (Y_{\min}, Y_{\max})$.

For simplicity, we denote outcomes by y_t . In the context of uncertainty quantification, these y_t are the errors e_t as defined above.

⁴For instance, we can predict the $\alpha/2$ and $1 - \alpha/2$ quantiles on the errors $e_t = y_t - f_\star(x_t)$ directly.

Given a quantile $q \in [0, 1]$, the goal is for the sequence of quantile forecasts to be calibrated with respect to a set of tests $\mathcal{H}$:

$$\sup_{h \in \mathcal{H}} \sum_{t=1}^T \mathbf{E}_{p_t \sim \mathcal{D}_t} [h(x_t, p_t)(1\{y_t \leq p_t\} - q)] \leq o(T) \quad (4)$$

Following one of the overarching themes of the class, as we increase the richness of $\mathcal{H}$, the better the resulting guarantees on our quantile estimates. For example,

- If $\mathcal{H} = \{1\}$ only contains the constant function, then, achieving the guarantee above does not necessarily imply a good estimate of the quantile, even though, it is the case that as T becomes large, p_t is less than y_t a q fraction of the time:

$$\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \mathbf{E}_{p_t \sim \mathcal{D}_t} 1\{p_t \leq y_t\} = q.$$

In particular, let $q = .5$. The alternating sequence $Y_{\min}, Y_{\max}, Y_{\min}, \dots$ satisfies this above guarantee while being a terrible estimate of the quantile. For every q , we can construct a deterministic sequence that ignores the entire data and achieves exact quantile coverage.⁵

- Furthermore, this strategy that alternates between the max and min value also achieves (4) with respect to any class of functions h that only depend on x .
- As we increase the complexity of $\mathcal{H}$ and enforce that outputs are *self consistent* by also including functions that depend on the outputs p_t themselves, e.g. $h(x, p) = p$, we get stronger notions of approximation. We cannot pass these tests by ignoring the data. Figure 1 illustrates this point on synthetic data.

4.1 Another Reduction to Online Learning

We can apply the same algorithmic ideas as before to derive a general recipe for online quantile regression. The recipe has again two main steps. An online learner returns the function that best summarizes our current errors, and then we find a distribution over forecasts such that, on that round, we look good with respect to the specific test. That is, our quantile coverage error is guaranteed to be negatively correlated with the Auditor's function no matter the outcome:

$$\sup_{y \in [Y_{\min}, Y_{\max}]} \mathbf{E}_{p_t \sim \mathcal{D}_t} h(x_t, p_t)(1\{p_t \leq y\} - q) \leq 0$$

Finding such a distribution $\mathcal{D}$ is very close to a 1 dimensional EVI problem where we want to solve for $\mathcal{D}$ such that $\mathbf{E}_{p \sim \mathcal{D}} S(p)(p - y) \leq 0$. The main difference is that instead of having $p - y$ we have the discontinuous error $1\{y \leq p\} - q$.

While we cannot solve these via linear optimization, we can apply the same 2 point argument from the defensive forecasting lecture. In particular, let S be any function from $[Y_{\min}, Y_{\max}]$ to $\mathbb{R}$ that is bounded by B , $\sup_p |S(p)| \leq B$. Consider the following procedure:

⁵These deterministic sequences yield error sequences that bound $\sum_{t=1}^T \mathbf{E}_{p_t \sim \mathcal{D}_t} [h(x_t, p_t)(1\{y_t \leq p_t\} - q)]$ by a constant. But constant regret is not interesting here.

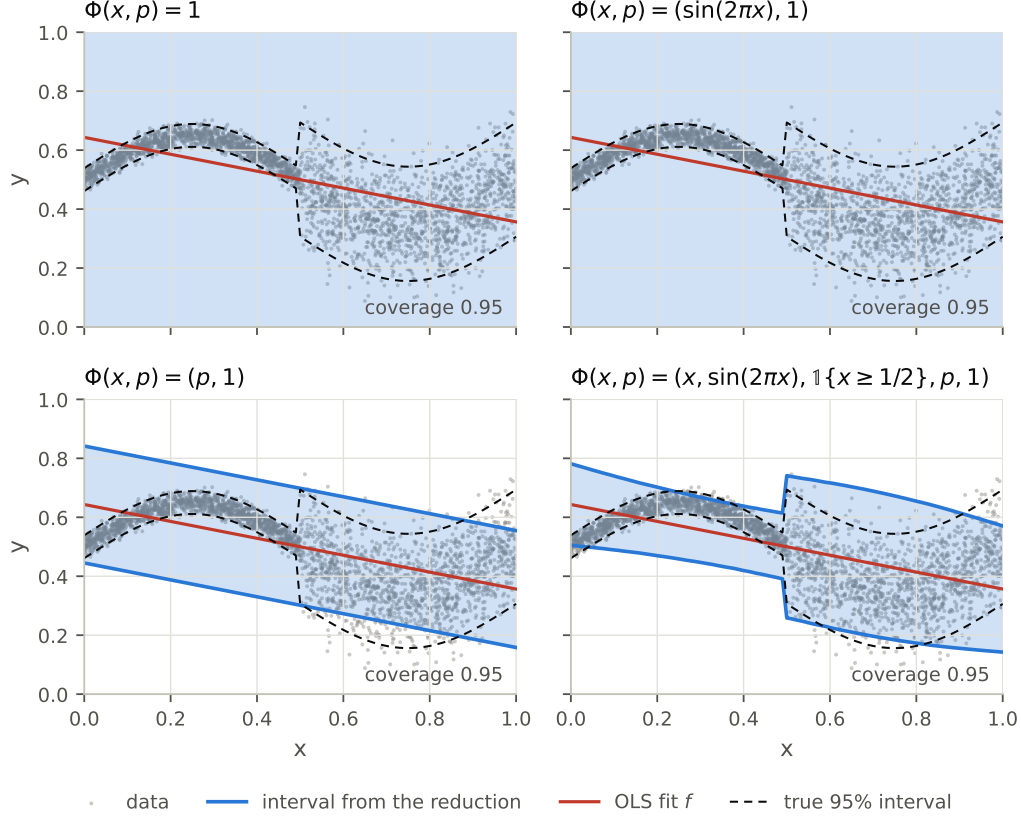


Figure 1: A sample of the data (gray), the least squares fit f (red), the interval $f(x) \pm p(x)$ produced by estimating uncertainty by performing quantile regression on $|f(x) - y|$ with $q = 0.95$ (blue), and the true 95% interval (dashed), for four test classes $\mathcal{H} = \{\langle \Phi(x, p), \theta \rangle\}$. Each panel reports the coverage $\frac{1}{T} \sum_t \mathbf{E}_{p_t \sim \mathcal{D}_t} \mathbf{1}\{e_t \leq p_t\}$ over the run. *Top left:* $\Phi = 1$, the class $\mathcal{H}$ contains the constant function. The algorithm plays $Y_{\max}$ in 95% of rounds and $Y_{\min}$ in the rest, without looking at the data. *Top right:* $\Phi = (\sin(2\pi x), 1)$, so $\mathcal{H}$ contains the functions $a + b \sin(2\pi x)$ of the features alone. Even though the class of tests is richer, we can still pass all the tests while completely ignoring the data. *Bottom left:* $\Phi = (p, 1)$, so $\mathcal{H}$ contains the functions $a + bp$ of the forecast alone. These tests rule out the deterministic strategy, however, it is ok for the algorithm to produce intervals that do not vary with x . *Bottom right:* $\Phi = (x, \sin(2\pi x), \mathbf{1}\{x \geq 1/2\}, p, 1)$, so $\mathcal{H}$ contains the functions $a + bx + c \sin(2\pi x) + d \mathbf{1}\{x \geq 1/2\} + ep$. The intervals now vary with x and track the conditional quantile of the residuals, narrow on the quiet half and wider on the noisy half.

1. If $S(Y_{\min}) \geq 0$, set $\mathcal{D}$ to be a point mass at $Y_{\min}$.
2. If $S(Y_{\max}) \leq 0$, set $\mathcal{D}$ to be a point mass at $Y_{\max}$.
3. Otherwise, find two points p_+ and p_- that are ε apart, $|p_+ - p_-| \leq \varepsilon$ and where $S(p_+) > 0$ and $S(p_-) < 0$. Set $\mathcal{D}$ to be the mixture

$$\mathcal{D} = \tau \delta_{p_-} + (1 - \tau) \delta_{p_+}, \quad \tau = \frac{S(p_+)}{S(p_+) - S(p_-)},$$

where δ_p denotes a point mass at p . The weight $\tau \in (0, 1)$ is chosen so that

$$\tau S(p_-) + (1 - \tau) S(p_+) = 0.$$

We can show that these distributions that mix between 2 close together points always guarantee that

$$\sup_{y \in (Y_{\min}, Y_{\max})} \mathbf{E}_{p \sim \mathcal{D}} S(p) (\mathbf{1}\{p \leq y\} - q) \leq \varepsilon$$

In particular, if $S(Y_{\min}) \geq 0$, then predicting $p = Y_{\min}$ guarantees that $\mathbf{1}\{Y_{\min} \leq y\} - q \geq 0$ for all $y \in (Y_{\min}, Y_{\max})$. A similar fact is true if $S(Y_{\max}) \leq 0$ and we set $p = Y_{\max}$.

If neither of these conditions hold, we mix between 2 points for which S has opposite signs. For any y , we can expand $\mathbf{E}_{p \sim \mathcal{D}} [S(p)(\mathbf{1}\{y \leq p\} - q)]$ using the definition of $\mathcal{D}$ as

$$\begin{aligned} &= \tau S(p_-) (\mathbf{1}\{y \leq p_-\} - \mathbf{1}\{y \leq p_+\}) + [\tau S(p_-) + (1 - \tau) S(p_+)] (\mathbf{1}\{y \leq p_+\} - q) \\ &= \tau |S(p_-)| \cdot \mathbf{1}\{p_- < y \leq p_+\} \\ &\leq B \cdot \mathbf{1}\{p_- < y \leq p_+\}. \end{aligned}$$

Here, we used the fact that $\tau S(p_-) + (1 - \tau) S(p_+) = 0$ by construction. Unlike the previous case, we cannot unconditionally guarantee that the error is small. However, the intuition is that if y is random, then $B \cdot \mathbf{1}\{p_- < y \leq p_+\}$ is naturally going to be very small since it is very unlikely that $y \in [p_-, p_+]$ for p_-, p_+ that are chosen to be exponentially close together.

For instance, if we assume that Nature chooses y from a Lipschitz distribution map:

$$\mathbf{Pr}[y \leq t_1] - \mathbf{Pr}[y \leq t_2] \leq L|t_1 - t_2|,$$

then the quantity above is at most $B \cdot L \cdot \varepsilon$ in expectation. Using the search routine, we can now present the general reduction to online learning. Given a test h , we write its quantile error as

$$\text{QuantileErr}_T(h) := \sum_{t=1}^T \mathbf{E}_{p_t \sim \mathcal{D}_t} [h(x_t, p_t) (\mathbf{1}\{y_t \leq p_t\} - q)].$$

Algorithm 2 illustrates how to achieve online quantile regression with respect to any $\mathcal{H}$ over which we can do online learning.

Algorithm 2: Black-box Reduction from Online Learning over $\mathcal{H}$ to Quantile Regression

Input: an online learner $\mathcal{A}$ over $\mathcal{H}$ and interval widths $\varepsilon_1, \varepsilon_2, \dots > 0$.

For $t = 1, 2, \dots$:

1. Nature reveals x_t .
2. The online learner $\mathcal{A}$ outputs $h_t \in \mathcal{H}$.
3. The Learner outputs a distribution $\mathcal{D}_t$ supported on two points $p_{t,-} \leq p_{t,+}$ with $|p_{t,+} - p_{t,-}| \leq \varepsilon_t$ such that

$$\mathbf{E}_{p \sim \mathcal{D}_t} [h_t(x_t, p)(\mathbf{1}\{y \leq p\} - q)] \leq B \cdot \mathbf{1}\{y \in [p_{t,-}, p_{t,+}]\} \text{ for all } y \in (Y_{\min}, Y_{\max}). \quad (5)$$

4. Nature reveals y_t , and $\mathcal{A}$ receives the loss

$$f_t(h) := - \mathbf{E}_{p_t \sim \mathcal{D}_t} [h(x_t, p_t)(\mathbf{1}\{y_t \leq p_t\} - q)]. \quad (6)$$

In words, the Learner announces two close forecasts and promises that the test h_t cannot be exploited by any outcome that lands outside the interval $I_t := [p_{t,-}, p_{t,+}]$. If the outcome lands inside I_t , the Learner pays at most B for that round. The two-point procedure above computes $\mathcal{D}_t$ with $\lceil \log_2((Y_{\max} - Y_{\min})/\varepsilon_t) \rceil$ evaluations of $h_t(x_t, \cdot)$ via binary search.

Theorem 4.1. Fix $h \in \mathcal{H}$ and let $\text{Regret}_T(h) := \sum_{t=1}^T f_t(h_t) - f_t(h)$ be the regret of $\mathcal{A}$ on the losses (6). The distributions $\mathcal{D}_t$ output by Algorithm 2 satisfy

$$\text{QuantileErr}_T(h) \leq \text{Regret}_T(h) + B \cdot N_T, \quad N_T := \sum_{t=1}^T \mathbf{1}\{y_t \in I_t\}.$$

Proof. Evaluating (5) at $y = y_t$ gives $-f_t(h_t) \leq B \cdot \mathbf{1}\{y_t \in I_t\}$ for every t . Therefore,

$$\text{QuantileErr}_T(h) = - \sum_{t=1}^T f_t(h) = \sum_{t=1}^T (f_t(h_t) - f_t(h)) - \sum_{t=1}^T f_t(h_t) \leq \text{Regret}_T(h) + B N_T.$$

□

The theorem says that the quantile error against any test is at most the regret of the online learner against that test, plus B times the number of rounds in which the outcome landed in the exception interval. Notice that the bound holds for every sequence, and that N_T is observable: the Learner knows I_t and sees y_t , so it can report N_T after the fact.

However, Nature sees I_t before choosing y_t , and nothing stops her from landing in it every round. Some condition on the outcomes is therefore needed for N_T to be small, and this is unavoidable. If $y_t = c$ at every round for some $c \in (Y_{\min}, Y_{\max})$ and $\mathcal{H}$ contains $\mathbf{1}\{p \geq c\}$ and $-\mathbf{1}\{p < c\}$, then one of the two tests has error at least $T \min(q, 1 - q)/2$ against any sequence of forecasts.

However, by placing some kinds of restrictions on Nature, we can guarantee this doesn't happen:

Corollary 4.2. *If Nature draws every y_t from an L -Lipschitz distribution and $\varepsilon_t = 1/t^2$, then $\mathbf{E}[N_T] \leq 2L$ and hence $\mathbf{E}[\text{QuantileErr}_T(h)] \leq \mathbf{E}[\text{Regret}_T(h)] + 2BL$ for every $h \in \mathcal{H}$.*

Proof. Conditionally on the history, $\Pr[y_t \in I_t] \leq L\varepsilon_t$ since I_t has length at most ε_t . Summing, $\mathbf{E}[N_T] \leq L \sum_{t \geq 1} t^{-2} = L\pi^2/6 \leq 2L$. $\square$

References

- [CBL06] Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.
- [FP26] Gabriele Farina and Juan Carlos Perdomo. An efficient black-box reduction from online learning to multicalibration, and a new route to Φ -regret minimization. *arXiv preprint arXiv:2604.19592*, 2026.
- [PR25] Juan Carlos Perdomo and Benjamin Recht. In defense of defensive forecasting. *arXiv preprint arXiv:2506.11848*, 2025.
- [ZAT⁺25] Brian Hu Zhang, Ioannis Anagnostides, Emanuel Tewolde, Ratip Emin Berker, Gabriele Farina, Vincent Conitzer, and Tuomas Sandholm. Expected variational inequalities. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025.