Bellman Conformal Inference: Calibrating Prediction Intervals For Time Series

Zitong Yang [◇] Emmanuel Candès [◇] Lihua Lei [†◇]
{zitong, lihualei, candes}@stanford.edu
[◇]Department of Statistics, Stanford University
[†]Stanford Graduate School of Bussiness

Abstract

We introduce Bellman Conformal Inference (BCI), a framework that wraps around any time series forecasting models and provides approximately calibrated prediction intervals. Unlike existing methods, BCI is able to leverage multi-step ahead forecasts and explicitly optimize the average interval lengths by solving a one-dimensional stochastic control problem (SCP) at each time step. In particular, we use the dynamic programming algorithm to find the optimal policy for the SCP. We prove that BCI achieves long-term coverage under arbitrary distribution shifts and temporal dependence, even with poor multi-step ahead forecasts. We find empirically that BCI avoids uninformative intervals that have infinite lengths and generates substantially shorter prediction intervals in multiple applications when compared with existing methods.

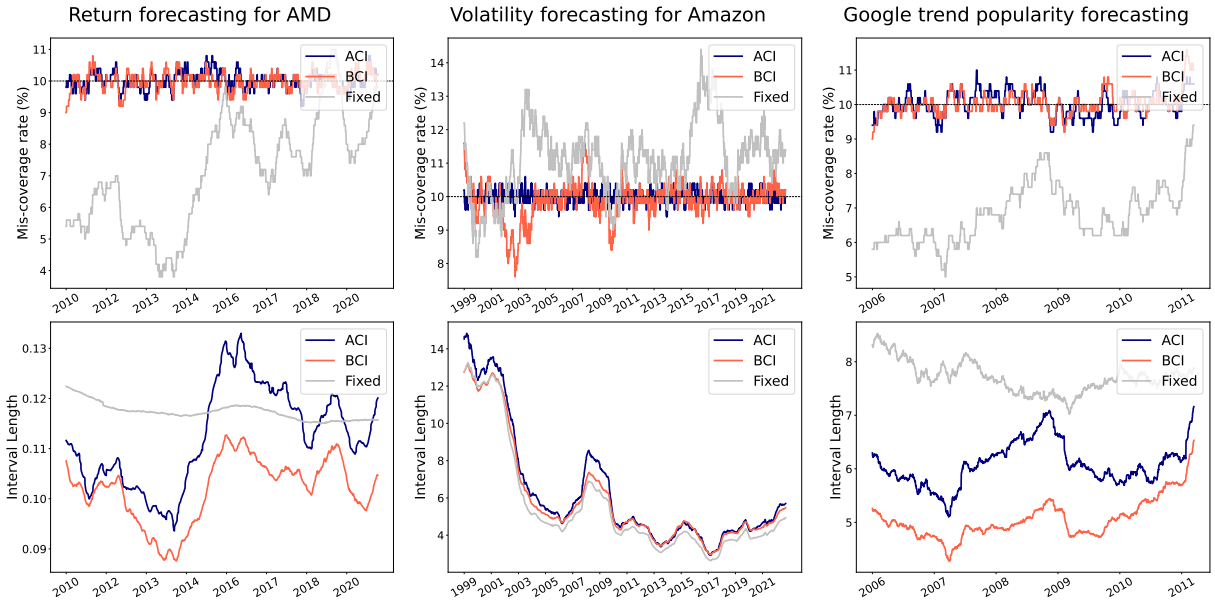
1 Introduction

Uncertainty quantification for time series nowcasting and forecasting is crucial in many areas such as climate science, epidemiology, industrial engineering, and macroeconomics. Ideally, the forecaster would generate a prediction interval at each time period that is calibrated in the sense that the fraction of intervals covering the true outcomes is approximately equal to the target coverage level in the long run. Classical approaches for generating prediction intervals are mostly model-based Box and Jenkins [1976], Engle [1982a], Stock and Watson [2010], Brown [1964], Jorda [2005]. However, time series models are often mis-specified due to non-stationarity or changing environments. As a result, the model-based prediction intervals tend to be poorly calibrated (see for instance the gray curves in Figure 1). Moreover, many forecasters have upgraded their workflows by incorporating black-box machine learning algorithms [e.g. Taylor and Letham, 2018, Makridakis et al., 2018, Herzen et al., 2022], for which valid uncertainty quantification proves to be challenging.

Due to complex temporal dependence and distribution shifts, distribution-free uncertainty quantification techniques such as conformal inference [e.g. Saunders et al., 1999, Papadopoulos et al., 2002, Vovk et al., 2005, Lei et al., 2015, 2018, Angelopoulos and Bates, 2021] are not guaranteed to achieve calibration. While many variants of conformal inference have been developed under weaker assumptions, most require restrictive dependence structure Chernozhukov et al. [2018], Tibshirani et al. [2019], Lei and Candès [2021], Oliveira et al. [2022], Candès et al. [2023], or limited distribution shifts Barber et al. [2023], or accurate model estimates Xu and Xie [2021, 2023], or multiple independent copies of the time series Stankeviciute et al. [2021], Dietterich and Hostetler [2022], Sun and Yu [2023].

An important departure from the aforementioned methods is Adaptive Conformal Inference (ACI) proposed by Gibbs and Candès [2021], as well as its variants Gibbs and Candès [2022], Zaffran et al. [2022], Feldman et al. [2023], Angelopoulos et al. [2023], which can produce approximately calibrated prediction intervals without making any assumptions on the time series. ACI generalizes standard conformal prediction for exchangeable data by choosing a time-varying nominal miscoverage rate to achieve calibration. Alternatively, ACI can be formulated as an online gradient descent algorithm that adjusts the nominal miscoverage rate downwards following each failure to cover the true outcome and upwards otherwise.

While the appeal of assumption-free calibration is evident, ACI lacks a mechanism to explicitly optimize the average interval lengths. For most practical time series forecasting models, multi-step ahead prediction intervals are readily available Box and Jenkins [1976], Fan and Yao [2003a], West and Harrison [2006], Politis and Wu [2023]. These intervals could be used to trade off between present and future interval lengths. For instance, if a two-day ahead interval accurately represents the one-day ahead interval for the following day, its length (at any given nominal miscoverage rate) could provide valuable guidance for judiciously selecting the nominal miscoverage rate to prevent unnecessarily wide intervals in two days. In this paper, we introduce Bellman Conformal Inference (BCI) which wraps around any multi-step ahead prediction intervals and formulates a stochastic control problem (SCP) to explicitly optimize the average interval lengths. In



(a) Return forecasting (b) Volatility forecasting (c) Google trend

Figure 1: Online time series forecasting for three different tasks: return forecasting on AMD stock data, volatility forecasting on Amazon stock data, and Google search popularity data for keyword "deep leanring". **Top panel:** moving averages of miscoverage rates over 500 data points. **Bottom panel:** moving averages of prediction interval lengths. In all figures, the red curves correspond to our proposed BCI algorithm; the blue curves correspond to the ACI algorithm with stepsize 0.1; the gray curves correspond to setting $\alpha_t = \overline{\alpha}$ for all t.

particular, at each time point, BCI formulates an SCP by taking the nominal miscoverage rate as the action to optimize an objective function that trades off between the average length of multi-step ahead prediction intervals, as proxies for the actual future prediction intervals, and the estimated average future coverage. By virtue of the scalar action, the SCP problem can be efficiently solved by dynamic programming (DP).

As with ACI, BCI is guaranteed to generate calibrated prediction intervals without making any assumptions on the data generating process – in particlar, it does not require the nominal multi-step prediction intervals to be well-calibrated. Instead of updating the nominal miscoverage rates directly, BCI applies the online gradient descent to adjust another parameter in the SCP that controls the trade-off between average interval lengths and short-term coverage rate. This step can be viewed as an instantiation of the Rolling RC method introduced by Feldman et al. [2023], an extension of ACI from online uncertainty quantification to online risk control that is akin to the extension of conformal inference in the offline setting Bates et al. [2021], Angelopoulos et al. [2021, 2022]. We apply BCI to forecast stock price volatility, absolute return, and the popularity of Google search trend. Our findings suggest that when the nominal multi-step ahead prediction intervals are poorly-calibrated, BCI can generate substantially shorter intervals compared to ACI as seen in Figure 1(c) and Figure 4(a). Conversely, when the nominal intervals are well-calibrated, BCI generates intervals of comparable lengths to ACI and prevents the occurrence of infinitely long intervals as seen in Figure 1(b) and Figure 4(b). We discuss this correspondence in detail in Section 4.3.

2 Setup

Consider a time series $Y_1, Y_2, \ldots$, where $Y_t \in \mathcal{Y}$ is the outcome of interest the analyst wants to predict. We assume Y_t is not observed until time $t+1$. Further, let $\mathcal{F}_{t-1}$ denote the σ-algebra generated by all information available at time t. In particular, $\mathcal{F}_{t-1}$ includes all past outcomes $Y_{t-1}, Y_{t-2}, \ldots, Y_1$. It can also include other variables that have been observed before the prediction interval for Y_t is generated.

2.1 Multi-step ahead prediction intervals

The analyst applies the forecasting algorithm that can generate T-step ahead prediction intervals, where T is a positive integer. At each time t and future time s for some $t \le s \le t+T-1$, let $C_{s|t}(1-\beta) \subset \mathcal{Y}$ denote the $(1-\beta)$ prediction interval for Y_s produced by the forecasting algorithm. Here, β stands for the nominal miscoverage rate. We say $C_{s|t}(1-\beta)$ is a nominal prediction interval because $\mathbb{P}(Y_s \in C_{s|t}(1-\beta)) = 1-\beta$ when the model behind the forecasting algorithm is correctly specified. However, when the model is misspecified, $\mathbb{P}(Y_s \in C_{s|t}(1-\beta))$ might deviate from $1-\beta$.

Any time series model can be used to generate multi-step prediction intervals. Below are two classical examples.

- ARMA model Box and Jenkins [1976], Fan and Yao [2003b]: an ARMA(p, q) model assumes

$$Y_t = b_1 Y_{t-1} + \ldots + b_p Y_{t-p} + \nu_t + a_1 \nu_{t-1} + \ldots + a_q \nu_{t-q},$$

  where $\nu_1, \nu_2, \ldots \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2)$. Then $C_{s|t}(\beta) = [\hat{Y}_{t+s-1} + z_{\beta/2}\hat{\sigma}_{t+s-1}, \hat{Y}_{t+s-1} + z_{1-\beta/2}\hat{\sigma}_{t+s-1}]$, where $\hat{Y}_{t+s-1}$ is the estimate of the mean of Y_{t+s-1} conditional on the past, $\hat{\sigma}_{t+s-1}$ is the estimated standard deviation of $Y_{t+s-1} - \hat{Y}_{t+s-1}$, and $z_{\beta/2}$ is the $\beta/2$-th quantile of the standard normal distribution.

- GARCH model Engle [1982b], Bollerslev [1986]: a GARCH(p, q) model is a generalization of the ARMA model that allows for conditional heteroskedasticitiy. In particular, it assumes $Y_t \sim \mathcal{N}(0, \sigma_t^2)$ where

$$\sigma_t^2 = \omega + b_1 \sigma_{t-1}^2 + \ldots + b_p \sigma_{t-p}^2 + a_1 \epsilon_{t-1}^2 + \ldots + a_q \epsilon_{t-q}^2,$$

  and $\epsilon_1, \epsilon_2, \ldots \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2)$. When the GARCH model is applied to forecast Y_t^2, $C_{s|t}(\beta)$ can be formed by the $(\beta/2)$-th and $(1-\beta/2)$-th quantile of $(\hat{\sigma}_{t+s-1}Z)^2$ where $Z \sim \mathcal{N}(0,1)$ and $\hat{\sigma}_{t+s-1}^2$ is the estimate of σ_{t+s-1}^2.

Even without a complete time series model, we can obtain multi-step ahead prediction intervals by fitting a generic likelihood function. We will also use this approach in the experiments later.

- Sequence-to-sequence neural networks Hochreiter and Schmidhuber [1997], Salinas et al. [2020], Vaswani et al. [2017]: this approach first models the marginal distribution of each Y_t (e.g., $Y_t \sim \mathcal{N}(\mu_t, \sigma_t^2)$). Then it uses a sequence-to-sequence neural networks such as a transformer to predict the likelihood parameters of a future observation Y_t (e.g. $\hat{\mu}_t, \hat{\sigma}_t$) from the lagged observations $\{Y_{t-1}, Y_{t-2}, \ldots\}$. The prediction intervals $C_{s|t}(\beta)$ can be formed using $\beta/2$ and $1-\beta/2$-th quantiles of Y_t's distribution (e.g. $\mathcal{N}(\hat{\mu}_t, \hat{\sigma}_t)$) with likelihood parameters imputed by the model forecast.

Throughout the paper we make the following mild assumptions on the nominal prediction intervals, akin to Gupta et al. [2022] and Feldman et al. [2023].

Assumption 1. *For any $t, s \ge 1$, the prediction interval $C_{s|t}(\beta) \subset \mathcal{Y}$ satisfies the following two conditions:*

- ***Monotonicity:*** *$C_{s|t}(\beta_1) \subset C_{s|t}(\beta_2)$ if $\beta_1 > \beta_2$,*
- ***Safeguard:*** *$C_{s|t}(1) = \mathcal{Y}$, i.e. $\mathbb{P}(Y \in C_{s|t}(1)) = 1$.*

The monotonicity condition assumes a smaller nominal miscoverage rate gives a wider interval, which is a natural condition for any reasonable prediction interval. The safeguard condition assumes that it is always safe to set $\beta = 0$. It is important if we want to achieve calibration without any distributional assumptions. Otherwise, nature can always choose the time series adversarially to escape the prediction interval. Note that when the conditions fail, we can enforce them by redefining $C_{s|t}(1-\beta) = \bigcup_{\beta' \le \beta} C_{s|t}(1-\beta')$ and $C_{s|t}(1) = \mathcal{Y}$.

2.2 Calibrating prediction intervals

Given the multi-step ahead prediction intervals $C_{s|t}(\cdot)$, BCI selects a nominal coverage index α_t and outputs $C_{t|t}(\alpha_t) \subset \mathcal{Y}$ as the prediction interval for Y_t. We write $C_t(\cdot) := C_{t|t}(\cdot)$ for notational convenience. Following Gibbs and Candès [2021], our goal is to generate a sequence of nominal miscoverage indices $\{\alpha_t : t \in [K]\}$ such that α_t only depends on $\mathcal{F}_{t-1}$ and

$$\lim_{K \to \infty} \frac{1}{K} \sum_{t=1}^{K} 1(Y_t \notin C_t(1-\alpha_t)) \le \overline{\alpha} \quad \text{almost surely} \tag{1}$$

for some pre-specified target miscoverage level $\overline{\alpha}$ (e.g. $\overline{\alpha} = 0.1$). In particular, the inequality needs to hold uniformly over any joint distribution of $\{Y_t : t \geq 1\}$. This includes the case where the sequence is deterministic. The ACI algorithm chooses an initial nominal coverage α_0 and updates α_t by

$$\alpha_t = \alpha_{t-1} + \gamma(\overline{\alpha} - \text{err}_{t-1}), \tag{2}$$

where $\text{err}_{t-1} = 1(Y_{t-1} \notin C_{t-1}(1 - \alpha_{t-1}))$. Above, γ is a fixed stepsize, which can be made data-adaptive using more advanced techniques in online learning Gibbs and Candès [2022]. Gibbs and Candès [2021] prove that, the average coverage over the first T time periods is at most $\overline{\alpha} + 2/T\gamma$.

Let β_t be the largest nominal miscoverage rate at which Y_t is covered,

$$\beta_t = \sup_{Y_t \in C_t(1-\beta)} \beta. \tag{3}$$

We shall refer to β_t as the uncalibrated probability inverse transform (PIT). Note that at time t, β_{t-1} is observed as Y_{t-1} is. If $C_t(1 - \beta)$ satisfies Assumption 1 and is continuous in β, Y_t is not covered if and only if $\alpha_t > \beta_t$: $\text{err}_t = 1(\alpha_t > \beta_t)$. Thus, for ACI, α_t depends exclusively on $(\alpha_{t-1}, \beta_{t-1})$. While ACI indeed provides valid calibration guarantee, it disregards two sources of information: the first is the length of $C_t(1-\beta)$ and multi-step ahead intervals which can inform a better choice of α_t. The second is the historical sequence of uncalibrated PITs $\{\beta_j\}_{j<t}$ that allows a better estimate of β_{t+1} than just using β_t alone. In particular, Angelopoulos et al. [2023] show that incorporating the historical errors into the update (2) can stabilize the large variability in the prediction intervals generated by ACI. Unlike all previous works, the BCI method we introduce in this paper chooses α_t as a function of the past actions $\alpha_{t-1}, \alpha_{t-2}, \ldots$, the past uncalibrated PITs $\beta_{t-1}, \beta_{t-2}, \ldots$, and the multi-step ahead intervals $C_{t|t}(\cdot), \ldots, C_{t+T-1|t}(\cdot)$.

3 Bellman Conformal Inference

3.1 BCI as Model Predictive Control

At a high level, BCI is analogous to Model Predictive Control (MPC) Borrelli et al. [2017]. At each time point t, BCI models the "dynamics" of the process $(Y_1, \beta_1), (Y_2, \beta_2), \ldots$ from past observations, simulates the "system" over the next T steps, and plans the "control" α_t by minimizing the "cost" driven by both the average interval length and miscoverage. Unlike traditional reactive control approaches, the proactive MPC approach is more suitable for forecasting problems under substantial distribution shifts.

To set up the cost minimization problem at time t, we denote by $L_{s|t}(\beta) = |C_{s|t}(1-\beta)|$ the function that maps the miscoverage rate to the length of the nominal multi-step ahead prediction interval for Y_s and $F_{s|t}$ the marginal distribution of β_s estimated using the past observations. We do not make any assumption on how $L_{s|t}(\cdot)$ and $F_{s|t}$ are generated. For all experiments in this paper, we simply set $F_{s|t} \equiv F_t$ where F_t is the empirical CDF of $\{\beta_{t-1}, \ldots, \beta_{t-B}\}$ for some large B. In our simulation, we set $B = 100$.

At time t, we solve the following optimization problem:

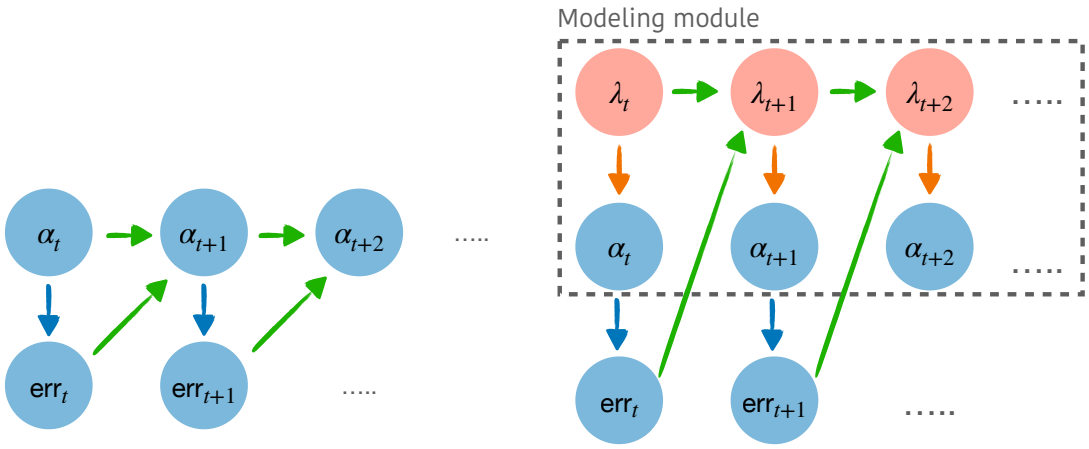
$$\min_{\alpha_{t|t},\ldots,\alpha_{t+T-1|t}} \mathbb{E}_{\substack{(\beta_{t|t},\ldots,\beta_{t+T-1|t}) \\ \sim F_{t|t} \otimes \ldots F_{t+T-1|t}}} \left[\overbrace{\sum_{s=t}^{t+T-1} L_{s|t}(\alpha_{s|t})}^{\text{Efficiency: interval length}} + \lambda_t \underbrace{\max\left(\frac{1}{T} \sum_{s=t}^{t+T-1} \text{err}_{s|t} - \bar{\alpha}, 0 \right)}_{\text{Validity: miscoverage rate}} \right], \tag{4}$$

where $\alpha_{s|t}$ denotes the planned action for time s, $\beta_{s|t}$ denotes a draw from $F_{s|t}$, $\text{err}_{s|t} = 1(\alpha_{s|t} > \beta_{s|t})$ denotes the error indicator for the realized $\beta_{s|t}$, and λ_t denotes the relative weight on the miscoverage that BCI uses to achieve the coverage guarantee. The first term measures the average interval length and the second term measures the rescaled average miscoverage rate within the receding horizon. Here we assume $\beta_{t|t}, \ldots, \beta_{t+T-1|t}$ are independent so that the cost function only depends on the marginal distributions. This is not required for the coverage guarantee (see Theorem 1) but makes the cost-minimization problem more tractable.

Under Assumption 1,

$$\lambda_t \leq 0 \Longrightarrow \alpha^*_{s|t} = 1, \ s = t, \ldots, t+T-1. \tag{5}$$

The analyst can apply any algorithm to minimize (4) as long as it satisfies (5). In particular, we discuss a DP-based algorithm in Section 3.3 that provides the exact solution and is computationally efficient for



(a) ACI (b) BCI

Figure 2: Schematic illustrations of standard ACI and BCI.

moderate values of T. Let $\alpha^*_{s|t}$ be the solution, which does not have to be exact. Clearly, $\alpha^*_{s|t}$ only depends on the model forecasts $(L_{s|t}(\cdot), F_{s|t})$ and the relative weight λ_t. The standard MPC would set $\alpha_t = \alpha^*_{t|t}$ and discard all other planned actions. To achieve the distribution-free coverage, BCI would modify α_t by considering both $\alpha^*_{t|t}$ and λ_t.

3.2 BCI update rule for λ_t and α_t

The key lever of BCI is the relative weight λ_t in (4). Intuitively, we shall increase λ_t to penalize the miscoverage term more when errors occur too often and decrease λ_t otherwise. Motivated by ACI Gibbs and Candès [2021] and the Rolling RC method Feldman et al. [2023], we update λ_t as follows:

$$\lambda_{t+1} = \lambda_t - \gamma[\overline{\alpha} - \text{err}_t], \tag{6}$$

where γ is a user-defined step size that controls how fast the procedure adapts to past errors and $\text{err}_t = 1(\alpha_t > \beta_t)$ as before.

As mentioned in Section 2, the safeguard property in Assumption 1 is crucial to guarantee long-term coverage even in the adversarial case. Here, we force α_t to be zero, so that $C_t(\alpha_t) = \mathcal{Y}$ under Assumption 1, when λ_t passes some pre-specified threshold $\lambda_{\max} > 0$:

$$\alpha_t = \begin{cases} 0, & \text{if } \lambda_t \geq \lambda_{\max}, \\ \alpha^\star_{t|t}, & \text{otherwise}, \end{cases} \tag{7}$$

where $\alpha^*_{t|t}$ is the minimizer of (4).

To match the scale, we can set $\gamma = c\lambda_{\max}$ for some $c \in (0, 1)$. Unlike ACI that updates α_t directly through a step of online gradient descent, BCI updates α_t indirectly through an online gradient descent-type update on λ_t. This allows BCI to take input other than past (α_t, β_t)'s. We illustrate the difference between ACI and BCI in Figure 2.

BCI with the above update rules achieves long-term coverage:

Theorem 1. *Let λ_t and α_t be defined by (6) and (7), respectively. Assume $\alpha^*_{t|t}$ is obtained by an algorithm satisfying (5). Under Assumption 1, when $\gamma = c\lambda_{\max}$ and $\gamma \in (0, \lambda_{\max})$, for any $m \geq 0$,*

$$\left| \frac{1}{K} \sum_{t=m+1}^{m+K} \text{err}_t - \overline{\alpha} \right| \leq \frac{c+1}{cK}. \tag{8}$$

In particular, (1) holds by letting $m = 0$ and $T \to \infty$.

The proof for Theorem 1 can be found in Appendix 9.1.

3.3 Solving $\alpha^*_{t|t}$ via dynamic programming

When the value of T is moderate (e.g. T=3), we can minimize (4) exactly and efficiently by DP. In fact, we do not recommend choosing a large T because the multi-step ahead prediction intervals tend to be less accurate or informative for events way into the future. For all our experiments, we choose $T = 3$.

The reader may refer to Appendix 7.1 for some background on the usage of DP in solving stochastic control problems. To set up the DP algorithm for our problem, we define the state variable $\rho_{s|t}$ as

$$\rho_{t|t} = 0,\ \rho_{s|t} = \sum_{k=t}^{s-1} 1(\alpha_{k|t} > \beta_{k|t}),\ s = t+1, \ldots, t+T.$$

The dynamics for the state variable can be written as

$$\rho_{s+1|t} = \rho_{s|t} + 1(\alpha_{s|t} > \beta_{s|t}).$$

Clearly, $\rho_{s|t}$ takes values in $\{0, 1, \ldots, s-t\}$. DP then minimizes (4) in a backward fashion. Following the standard DP terminology (see Appendix 7.1), the cost-to-go function at time $t+T$ is

$$J_{t+T|t}(\rho) = \lambda_t \max\left(\rho/T - \bar{\alpha}, 0\right), \rho \in \{0, \ldots, T\}.$$

This measures the loss incurred in the last step when $\rho_{t+T|t} = \rho$. The cost-to-go functions for time $s = t+T-1, t+T-2, \ldots, t$ are recursively defined through the Bellman equation: $J_{s|t}(\rho) =$

$$\min_{\alpha_{s|t}} \mathbb{E}_{\beta_{s|t} \sim F_{s|t}} \left[L_{s|t}(\alpha_{s|t}) + J_{s+1|t}(\rho + 1(\alpha_{s|t} > \beta_{s|t})) \right],$$

for $\rho \in \{0, \ldots, s-t\}$. It measures the optimal cumulative loss incurred from time $s+1$ onwards when $\rho_{s|t} = \rho$. Since $\beta_{s|t} \sim F_{s|t}$, we can rewrite $J_{s|t}(\rho)$ as

$$J_{s|t}(\rho) = \min_{\alpha_{s|t}} \Big\{ L_{s|t}(\alpha_{s|t}) + J_{s+1|t}\left(\rho+1\right) F_{s|t}(\alpha_{s|t}) + J_{s+1|t}\left(\rho\right) \left(1 - F_{s|t}(\alpha_{s|t})\right) \Big\}.$$

To simplify the expression, we define

$$D_{s|t}(\rho) = J_{s+1|t}\left(\rho+1\right) - J_{s+1|t}\left(\rho\right).$$

Then

$$J_{s|t}(\rho) = J_{s+1|t}\left(\rho\right) + \min_{\alpha} \left\{ L_{s|t}(\alpha) + D_{s|t}(\rho) F_{s|t}(\alpha) \right\},$$

and the optimal policy at time s can be characterized as

$$\tilde{\alpha}_{s|t}(\rho) = \arg\min_{\alpha} \left\{ L_{s|t}(\alpha) + D_{s|t}(\rho) F_{s|t}(\alpha) \right\}. \tag{9}$$

The following Proposition provides a useful property of the objective function in (9).

Proposition 3.1. *At any time s, the cost-to-go function $J_{s|t}(\rho)$ is nonnegative and non-decreasing in ρ.*

Proposition 3.1 implies that the objective function $L_{s|t}(\alpha) + D_{s|t}(\rho)F_t(\alpha)$ is a sum of increasing function $D_{s|t}(\rho)F_t(\alpha)$ and a decreasing function $L_{s|t}(\alpha)$, which depicts the efficiency-coverage tradeoff discussed earlier.

Since the optimization problem in (9) is one-dimensional, it can be efficiently solved with a grid search. In addition, when we choose $F_{s|t}$ to be the empirical CDF of $\{\beta_{t-1}, \ldots, \beta_{t-B}\}$ as discussed earlier, it is easy to see that $\alpha^*_{s|t}(\rho) \in \{\beta_{t-1}, \ldots, \beta_{t-B}\}$ if $L_{s|t}(\cdot)$ is continuous. As a result, (9) can be solved exactly.

Finally, since $\rho_{t|t} = 0$, the optimal solution $\alpha^*_{t|t}$ in (7) is given by $\tilde{\alpha}_{t|t}(0)$. Since $\rho_{s|t}$ is integer-valued, the computation of $\tilde{\alpha}_{t|t}(0)$ involves solving $(T+1)+T+\ldots+1 = O(T^2)$ one-dimensional optimization problems in the form of (9). When T is moderate, it is computationally efficient.

We summarize BCI with the DP algorithm described in this subsection in Algorithm 1.

Algorithm 1: Bellman conformal inference at time t

Hyperparameters: target miscoverage level $\overline{\alpha}$; length of receding horizon T; maximum weight $\lambda_{\max}$; relative step size $c \in (0, 1)$.
Input: Estimated marginal CDF of future uncalibrated PITs $F_{t|t}, \ldots, F_{t+T-1|t}$;
Length functions for multi-step ahead prediction intervals $L_{t|t}(\cdot), \ldots, L_{t+T-1|t}(\cdot)$;
Weight λ_{t-1} and miscoverage indicator err_{t-1} from the previous iteration.
Step 1: Update the security parameter as $\lambda_t = \lambda_{t-1} - \gamma[\overline{\alpha} - \mathrm{err}_{t-1}]$.
Step 2: Using $L_{s|t}$ and F_t, instantiate the stochastic control problem defined in (4) from Section 3.1.
Step 3: Apply the DP algorithm described in Section 3.3 to get $\tilde{\alpha}_{t+T-1|t}(\cdot), \ldots, \tilde{\alpha}_{t|t}(\cdot)$
Output: $\alpha_t = \tilde{\alpha}_{t|t}(0)$ if $\lambda_t \leq \lambda_{\max}$ and $\alpha_t = 0$ otherwise.

4 Empirical results

In this section, we present empirical experiments on real time series forecasting problems to demonstrate the effectiveness of BCI. We consider auto-regressive time series forecasting task for three datasets: daily Google trend popularity for keyword `deep learning`, daily stock return for companies `AMD, Amazon, Nvidia`, and stock volatility for the same companies. We use the ACI procedure (2) as a baseline for comparison. Making fair comparisons between ACI and BCI is not straightforward and we propose a basis for comparison in Section 4.2. The code for reproducing the results in the paper can be found at https://github.com/ZitongYang/bellman-conformal-inference.git.

4.1 Dataset and model fitting

Return forecasting. Our first example studies the relative return of stock prices from various companies. We download the daily stock price of companies `Amazon, AMD, Nvidia` from the *Wall Street Journal Market Data*.[1] The length of history varies from company to company. That said, for all companies we have approximately 15 years of data with roughly 250 trading days each year. For each company, we compute the one-day log return Y_t of the stock as

$$Y_t = \log(P_t / P_{t-1}).$$

On each day t, we use the lagged sequence $\{Y_{t-1}, \ldots, Y_{t-100}\}$ as our predictors. We use a decoder-only transformer Vaswani et al. [2017] of embedding size 16 with 8 heads and 2 layers. This choice leads to a head size of $16/8 = 2$, which makes this a low-dimensional self-attention operation that is suitable for univariate time series. For the output, we use a linear layer to map the embedding to a $2 \times T$ dimensional vector, representing the predicted mean and standard deviation of the for the next T days. Mathematically, we can think of our transformer as a mapping from $\mathbb{R}^{100}$ to $\mathbb{R}^{2\times T}$, denoted by $[\mu_{s|t}(\boldsymbol{X}_t; \boldsymbol{\theta}), \sigma_{s|t}(\boldsymbol{X}_t; \boldsymbol{\theta})]$ for $s = t, \ldots, t+T-1$. Here, $\boldsymbol{X}_t = \{Y_{t-1}, \ldots, Y_{t-100}\}$ are the lagged returns and $\boldsymbol{\theta}$ denotes the weights of the neural network. At time t, we estimate $\boldsymbol{\theta}$ by solving the following optimization problem:

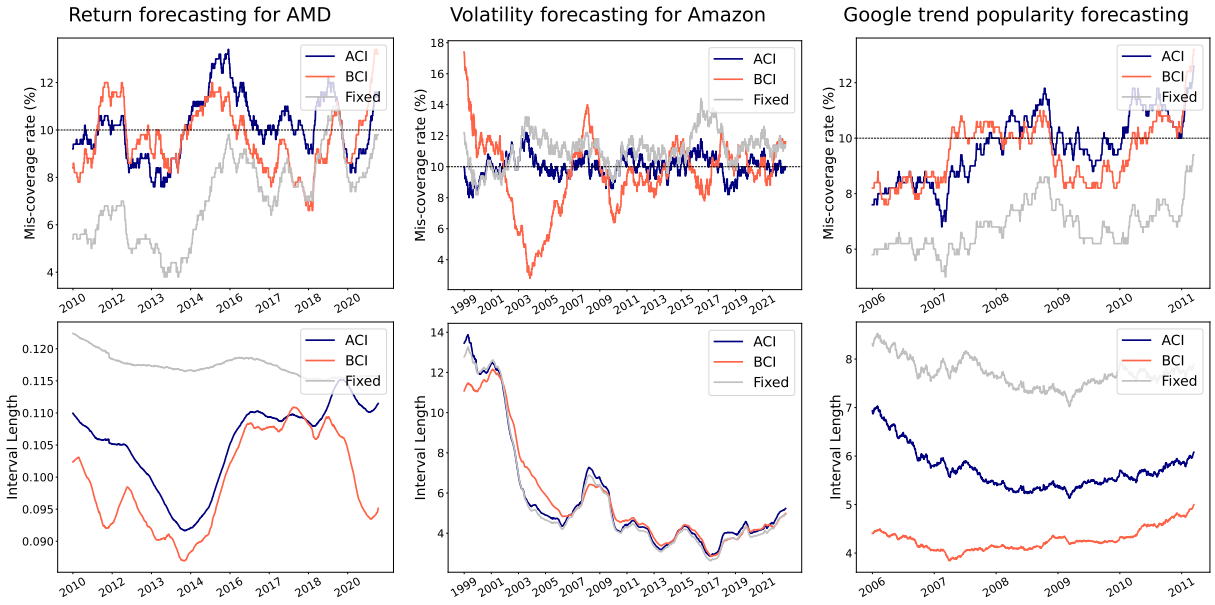
$$\min_{\boldsymbol{\theta}} \sum_{\tau<T} \sum_{s=\tau-T}^{\tau-1} -\log \mathcal{N}\left(Y_{t+s}; \mu_{s|t}(\boldsymbol{X}_t; \boldsymbol{\theta}), \sigma_{s|t}(\boldsymbol{X}_t; \boldsymbol{\theta})\right).$$

This formulation assumes the distribution of Y_{t+s} follows a normal distribution $\mathcal{N}\left(\mu_{s|t}(\boldsymbol{X}_t; \boldsymbol{\theta}), \sigma_{s|t}(\boldsymbol{X}_t; \boldsymbol{\theta})\right)$. It is a straightforward adaptation from the standard techniques for training variational auto-encoders Kingma and Welling [2014]. Then we can construct $C_{s|t}(1-\beta)$ as $[\mu_{s|t}(\boldsymbol{X}_t; \boldsymbol{\theta}) \pm z_{1-\beta/2} \cdot \sigma_{s|t}(\boldsymbol{X}_t; \boldsymbol{\theta})]$ where $z_{1-\beta/2}$ is the $(1-\beta/2)$-th quantile of the standard normal distribution.

Volatility forecasting. Our next example is the volatility forecasting problem explored in Gibbs and Candès [2021]. We work with the same daily stock data as in the return forecasting problem. Instead of the log return, we compute the squared volatility as

$$Y_t = (P_t / P_{t-1} - 1)^2.$$

[1] https://www.wsj.com/market-data



(a) Return forecasting (b) Volatility forecasting (c) Google trend

Figure 3: Same as Figure 1, except that the stepsize γ for ACI is 0.08 for looser control.

Task and dataset	Miscoverage rate (%)		Average length*		Frac. days with ∞	
	BCI	ACI	BCI	ACI	BCI	ACI
Return-Nvidia	10.04%	10.01%	0.08	0.09	0.0%	3.76%
Return-AMD	9.81%	9.99%	0.1	0.11	0.0%	2.40%
Return-Amazon	9.86%	9.97%	0.07	0.08	0.0%	2.37%
Volatility-Nvidia	10.00%	10.00%	4.72	4.85	0.0%	2.19%
Volatility-Amazon	9.98%	10.00%	4.06	4.09	0.0%	1.89%
Volatility-AMD	9.97%	9.99%	7.91	8.07	0.0%	9.79%
Google Trend Popularity	10.00%	10.00%	5.07	6.11	0.0%	1.75%

Table 1: Summary statistics for ACI with $\gamma = 0.1$ and BCI with a stepsize that matches the variance of LocalMiscov$_t$. (* We compute the average length removing the ∞-sized prediction intervals.)

We then use the lagged volatility $\{Y_{t-1}, \ldots, Y_{t-100}\}$ as a the predictors. Instead of using modern neural networks, we use the classical time series model `GARCH(1, 1)`Engle [1982b], Bollerslev [1986] that is often used to model stock price volatility. As we will see in Section 4.3, this classic model delivers nearly (marginally) calibrated prediction intervals. The `GARCH(1, 1)` model assumes that the squared volatility Y_s conditioned on the current observation follows a non-central χ-squared distribution and we can use its $\beta/2$- and $1-\beta/2$-th quantile to form the prediction interval $C_{s|t}(1-\beta)$. We provide a detailed description of the `GARCH(1, 1)` model in Appendix 8.

Google trend popularity. Finally, we apply BCI to non-financial data. We consider the daily Google search popularity for the keyword `deep learning` from 2006 to 2011. For this dataset, we use the same neural-network-based fitting procedure as in the return forecasting task, except that we use a 5-layer LSTM Hochreiter and Schmidhuber [1997] recurrent network instead of the transformer to demonstrate the compatibility of BCI with a broader class of forecasters.

Dataset	Miscoverage rate (%)		Average length*		Frac. days with ∞	
	BCI	ACI	BCI	ACI	BCI	ACI
Return-Nvidia	9.12%	9.97%	0.08	0.09	0.0%	0.0%
Return-AMD	9.6%	9.99%	0.1	0.11	0.0%	0.0%
Return-Amazon	9.6%	9.97%	0.06	0.07	0.0%	0.0%
Volatility-Nvidia	9.75%	9.85%	4.52	4.47	0.0%	0.0%
Volatility-Amazon	9.81%	9.82%	3.85	3.79	0.0%	0.0%
Volatility-AMD	10.01%	10.47%	7.46	7.3	0.0%	0.0%
Google Trend	9.90%	9.71%	4.24	5.64	0.0%	0.0%

Table 2: Same as Table 1, except that we run ACI with $\gamma = 0.08$ for a somewhat loose control of local miscoverage rate.

4.2 Performance evaluation

We turn to discussing how we evaluate the performance of BCI. In particular, we point out some caveats when comparing the performance of different online prediction intervals and propose an approach for a fair evaluation.

Evaluation metrics. We evaluate the performance of online prediction intervals using two metrics: average miscoverage rate and average interval length. Following Gibbs and Candès [2021], we measure the local average of both measures over a moving window of size 500:

$$\begin{bmatrix}\text{LocalMiscov}_t \\ \text{LocalLength}_t\end{bmatrix} = \frac{1}{500}\sum_{s=t-250}^{t+250}\begin{bmatrix}\text{err}_s \\ |C_s(1-\alpha_s)|\end{bmatrix}.$$

An approach for fair comparison. For both ACI and BCI, the stepsize γ can trade off the tightness of the coverage and the rate of change of the interval length. Typically, a smaller stepsize means that the local miscoverage rate will exhibit larger excursions away from the target, hence looser control, and that the length of the prediction intervals will be smootherdue to the smaller increment in α_t's. We illustrate this tradeoff in Figure 5. Ideally, we want to compare the average interval lengths when both methods achieve similar levels of coverage control. To make the comparison fair, we first choose a set of ACI parameters ($\gamma = 0.1$ for tight control and $\gamma = 0.008$ for loose control). For each choice of γ for ACI, we perform a grid search on the stepsize of BCI, and choose the one that matches the sample variance of ACI's LocalMiscov_t.

4.3 Empirical results

For each forecasting problem, we apply ACI with $\gamma = 0.008$ (tight coverage control) and $\gamma = 0.1$ (loose coverage control). We also run BCI with stepsizes calibrated to match each version of ACI as discussed in the last subsection, as well as the naive benchmark with $\alpha_t = \bar{\alpha}$.

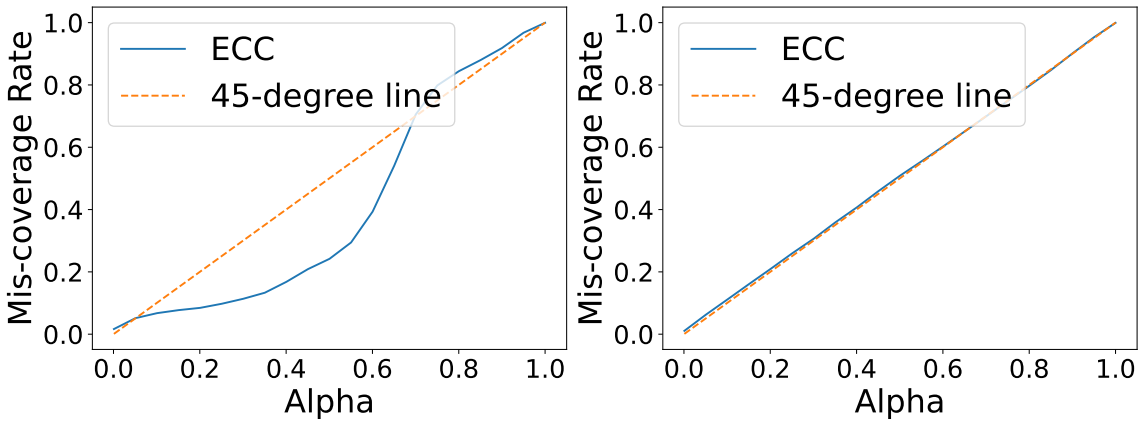
The time series of LocalMiscov_t and LocalLength_t are plotted in Figure 1 and Figure 3 for a subset of experiments. The average miscoverage rate and interval lengths over all time periods are summarized in Table 1 and 2 for all experiments. By design, ACI and BCI achieve similar levels of coverage control. In most of the cases, BCI outperforms ACI in terms of the average interval lengths, especially for return forecasting and Google trend forecasting. The rest of the experiments are plotted in Appendix 10.

Uninformative infinite-length intervals. One issue pointed out by Angelopoulos et al. [2023] is that ACI may generate infinite-length intervals. From Table 1, we observe that ACI generates a moderate fraction of infinite-length prediction intervals under tight coverage control. In contrast, BCI completely avoids infinite intervals and generally tend to produce shorter intervals. Under loose control, neither ACI nor BCI generate any uninformative intervals.

Quality of nominal prediction intervals. From Figure 1 and Figure 3, we can see that BCI performs generally better than ACI for both tight and loose control. For return forecasting and Google trend forecasting, the gain is more prominent than for volatility forecasting. Intuitively, if the nominal prediction intervals are well-calibrated, BCI would not gain much from using the multi-step ahead prediction intervals. To formalize it, we define the expected calibration curve for nominal prediction intervals $C_t(\cdot)$ as follows:

$$\mathsf{ECC}(\alpha) = \frac{1}{K}\sum_{t=1}^{K} 1\{Y_t \notin C_t(1-\alpha)\}.$$

In words, $\mathsf{ECC}(\alpha)$ is the average miscoverage rate when $\alpha_t = \alpha$. If $\mathsf{ECC}(\alpha) \approx \alpha$ for all $\alpha \in (0,1)$, it means that the prediction sets are well-calibrated. From Figure 4, we can see that the Google trend forecasting



(a) Google trend forecasting

(b) Volatility forecasting

Figure 4: Expected calibration curve of nominal prediction intervals for Google trend forecasting and volatility forecasting.

with LSTM neural network (Figure 4(a)) is too conservative for small α and too aggressive for large α. In contrast, the `GARCH(1, 1)` generates nearly calibrated prediction invervals (Figure 4(b)), which means that, on average, $C_t(1-\alpha)$ covers the true Y_t $(1-\alpha)$ fraction of the time, leaving little possibility for BCI to alter the prediction intervals. This supports our intuition that BCI improves upon ACI more substantially when the nominal prediction intervals are poorly calibrated.

5 Conclusion

We propose Bellman Conformal Inference (BCI) as an extension of ACI that calibrates nominal prediction intervals produced by any forecasting algorithms. BCI leverages multi-step ahead prediction intervals and applies Model Predictive Control (MPC) techniques to explicitly optimize the interval lengths. When the nominal prediction intervals are poorly calibrated, BCI improves substantially upon ACI in terms of the average interval lengths given the same level of coverage control; otherwise, BCI has comparable performance to ACI.

6 Acknowledgement

Zitong Yang would like to acknowledge the support of Albion Walter Hewlett Stanford Graduate Fellowship. Lihua Lei is grateful for the support of National Science Foundation grant DMS-2338464. E.J.C. was supported by the Office of Naval Research grant N00014-20-1-2157, the National Science Foundation grant DMS-2032014, the Simons Foundation under award 814641, and the ARO grant 2003514594.

References

Anastasios N Angelopoulos and Stephen Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv preprint arXiv:2107.07511*, 2021.

Anastasios N Angelopoulos, Stephen Bates, Emmanuel J Candès, Michael I Jordan, and Lihua Lei. Learn then test: Calibrating predictive algorithms to achieve risk control. *arXiv preprint arXiv:2110.01052*, 2021.

Anastasios N Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. Conformal risk control. *arXiv preprint arXiv:2208.02814*, 2022.

Anastasios N Angelopoulos, Emmanuel J Candes, and Ryan J Tibshirani. Conformal PID control for time series prediction. *arXiv preprint arXiv:2307.16895*, 2023.

Rina Foygel Barber, Emmanuel J Candes, Aaditya Ramdas, and Ryan J Tibshirani. Conformal prediction beyond exchangeability. *The Annals of Statistics*, 51(2):816–845, 2023.

Stephen Bates, Anastasios Angelopoulos, Lihua Lei, Jitendra Malik, and Michael Jordan. Distribution-free, risk-controlling prediction sets. *Journal of the ACM (JACM)*, 68(6):1–34, 2021.

Dimitri P. Bertsekas. *Dynamic Programming and Stochastic Control*. Academic Press, Inc., USA, 1976. ISBN 0120932504.

Tim Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of econometrics*, 31(3): 307–327, 1986.

Francesco Borrelli, Alberto Bemporad, and Manfred Morari. *Predictive Control for Linear and Hybrid Systems*. Cambridge University Press, 2017. doi: 10.1017/9781139061759.

George.E.P. Box and Gwilym M. Jenkins. *Time Series Analysis: Forecasting and Control*. Holden-Day, 1976.

Robert Goodell Brown. Smoothing, forecasting and prediction of discrete time series. In ., 1964. URL https://api.semanticscholar.org/CorpusID:120260777.

Emmanuel Candès, Lihua Lei, and Zhimei Ren. Conformalized survival analysis. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 85(1):24–45, 2023.

Victor Chernozhukov, Kaspar Wüthrich, and Zhu Yinchu. Exact and robust conformal inference methods for predictive machine learning with dependent data. In *Conference On learning theory*, pages 732–749. PMLR, 2018.

Thomas G Dietterich and Jesse Hostetler. Conformal prediction intervals for markov decision process trajectories. *arXiv preprint arXiv:2206.04860*, 2022.

Robert Engle. Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica*, 50(4):987–1007, 1982a. URL https://EconPapers.repec.org/RePEc:ecm:emetrp:v:50:y:1982:i:4:p:987-1007.

Robert F Engle. Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica: Journal of the econometric society*, pages 987–1007, 1982b.

Jianqing Fan and Qiwei Yao. *Nonlinear time series: nonparametric and parametric methods*, volume 20. Springer, 2003a.

Jianqing Fan and Qiwei Yao. *Nonlinear Time Series: Nonparametric and Parametric Methods*. Springer, 01 2003b. ISBN 978-0-387-26142-3. doi: 10.1007/978-0-387-69395-8.

Shai Feldman, Liran Ringel, Stephen Bates, and Yaniv Romano. Achieving risk control in online learning settings, 2023.

Isaac Gibbs and Emmanuel Candès. Adaptive conformal inference under distribution shift. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=6vaActvpcp3.

Isaac Gibbs and Emmanuel Candès. Conformal inference for online prediction with arbitrary distribution shifts, 2022.

Chirag Gupta, Arun K Kuchibhotla, and Aaditya Ramdas. Nested conformal prediction and quantile out-of-bag ensemble methods. *Pattern Recognition*, 127:108496, 2022.

Julien Herzen, Francesco LAossig, Samuele Giuliano Piazzetta, Thomas Neuer, LAoo Tafti, Guillaume Raille, Tomas Van Pottelbergh, Marek Pasieka, Andrzej Skrodzki, Nicolas Huguenin, Maxime Dumonal, Jan KoAcisz, Dennis Bader, FrACdAcrick Gusset, Mounir Benheddi, Camila Williamson, Michal Kosinski, Matej Petrik, and GaAl Grosch. Darts: User-friendly modern machine learning for time series. *Journal of Machine Learning Research*, 23(124):1–6, 2022. URL http://jmlr.org/papers/v23/21-1177.html.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, nov 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

Oscar Jorda. Estimation and inference of impulse responses by local projections. *American Economic Review*, 95(1):161–182, March 2005. doi: 10.1257/0002828053828518. URL https://www.aeaweb.org/articles?id=10.1257/0002828053828518.

Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

Jing Lei, Alessandro Rinaldo, and Larry Wasserman. A conformal prediction approach to explore functional data. *Annals of Mathematics and Artificial Intelligence*, 74:29–43, 2015. doi: 10.1007/s10472-013-9366-6.

Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J Tibshirani, and Larry Wasserman. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113(523):1094–1111, 2018.

Lihua Lei and Emmanuel J Candès. Conformal inference of counterfactuals and individual treatment effects. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 83(5):911–938, 2021.

Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. Statistical and machine learning forecasting methods: Concerns and ways forward. *PloS one*, 13(3):e0194889, 2018.

Roberto I Oliveira, Paulo Orenstein, Thiago Ramos, and João Vitor Romano. Split conformal prediction for dependent data. *arXiv preprint arXiv:2203.15885*, 2022.

Harris Papadopoulos, Kostas Proedrou, Volodya Vovk, and Alex Gammerman. Inductive confidence machines for regression. In *Machine Learning: ECML 2002: 13th European Conference on Machine Learning Helsinki, Finland, August 19–23, 2002 Proceedings 13*, pages 345–356. Springer, 2002.

Dimitris N Politis and Kejin Wu. Multi-step-ahead prediction intervals for nonparametric autoregressions via bootstrap: Consistency, debiasing, and pertinence. *Stats*, 6(3):839–867, 2023.

David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3):1181–1191, 2020. ISSN 0169-2070. doi: https://doi.org/10.1016/j.ijforecast.2019.07.001. URL https://www.sciencedirect.com/science/article/pii/S0169207019301888.

Craig Saunders, Alexander Gammerman, and Volodya Vovk. Transduction with confidence and credibility. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, IJCAI '99, page 722–726, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc. ISBN 1558606130.

Kamile Stankeviciute, Ahmed M Alaa, and Mihaela van der Schaar. Conformal time-series forecasting. *Advances in neural information processing systems*, 34:6216–6228, 2021.

James Stock and Mark Watson. *Dynamic Factor Models*, chapter ., page . Oxford University Press, Oxford, 2010. URL http://www.economics.harvard.edu/faculty/stock/files/dfm_oup_4.pdf.

Sophia Sun and Rose Yu. Copula conformal prediction for multi-step time series forecasting, 2023.

Sean J Taylor and Benjamin Letham. Forecasting at scale. *The American Statistician*, 72(1):37–45, 2018.

Ryan J Tibshirani, Rina Foygel Barber, Emmanuel Candes, and Aaditya Ramdas. Conformal prediction under covariate shift. *Advances in neural information processing systems*, 32, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer-Verlag, Berlin, Heidelberg, 2005. ISBN 0387001522.

Mike West and Jeff Harrison. *Bayesian forecasting and dynamic models*. Springer Science & Business Media, 2006.

Chen Xu and Yao Xie. Conformal prediction interval for dynamic time-series. In *International Conference on Machine Learning*, pages 11559–11569. PMLR, 2021.

Chen Xu and Yao Xie. Sequential predictive conformal inference for time series. In *International Conference on Machine Learning*, pages 38707–38727. PMLR, 2023.

Margaux Zaffran, Olivier Féron, Yannig Goude, Julie Josse, and Aymeric Dieuleveut. Adaptive conformal predictions for time series. In *International Conference on Machine Learning*, pages 25834–25866. PMLR, 2022.

7 Background on stochastic control

This is a self-contained section outlining some background on stochastic control Bertsekas [1976] needed for this paper. To be consistent with the control literature, we use the standard notation in this section. Consider a finite horizon stochastic system defined by the update rule

$$x_{t+1} = f_t(x_t, u_t, w_t) \text{ for } t = 1, 2, \ldots, T, \tag{10}$$

where x_t is the state variable, u_t is the control variable chosen by the analyst, and w_t is the random disturbance to the system at time t that affects the next state x_{t+1}. The control u_t can depend on any information available at time t but not on future knowledge. The disturbance w_t are sampled from $w_t \sim P(\cdot|x_t, u_t)$ and can depend on the state x_t and the control u_t. At each time t, the analyst bears the cost

$$g_t(x_t, u_t, w_t). \tag{11}$$

At the terminal time $T+1$, the analyst bears the cost $g_{T+1}(x_{T+1})$. The planning strategy is described by policy $\pi = \{\mu_t\}_{t=1,\ldots,T}$, where each μ_t determines the control at time t through

$$u_t = \mu_t(x_t). \tag{12}$$

Given an initial state x_1, the expected cost under policy π is

$$J_\pi(x_1) = \mathbb{E}_{w_{1:T}}\left[g_{T+1}(x_{T+1}) + \sum_{t=1}^{T} g_t(x_t, \mu_t(x_t), w_t)\right]. \tag{13}$$

The problem of optimal control is to find the optimal policy

$$\pi^*_{x_1} = \arg\min_\pi J_\pi(x_1). \tag{14}$$

7.1 Dynamic programming

The optimal control problem admits exact solution through dynamic programming. To introduce the dynamic programming algorithm, we first define auxiliary functions

$$J_t(x_t) := \text{The optimal expected cost given that we start at state } x_t \text{ at time } t. \tag{15}$$

These functions are called "cost-to-go" function in dynamic programming literature. Under this definition, we have that

$$J_{T+1}(x_{T+1}) = g_{T+1}(x_{T+1}).$$

Now suppose that we have solved function J_{t+1} exactly. We can solve J_t according the update rule

$$J_t(x) = \min_u \mathbb{E}_{w\sim P(\cdot|x,u)}\left[g_t(x, u, w) + J_{t+1}(f_t(x, u, w))\right]. \tag{16}$$

Mathematically, the minimizer u^* of the program in (16) will depend on x, which defines the relation $u^* = \mu_t(x)$. A standard result that establishes the optimality of dynamic programming algorithm dictates the policy $\{\mu_t\}$ find through (16) is the same as the optimal policy in (14).

8 Stochastic model for the volatility series

In the definition below, we introduce the GARCH(1, 1) process which we will later use as a stochastic model for the return series $r_1, \ldots, r_K$.

Definition 1 (GARCH(1, 1))**.** *Let $p_k(r_1, \ldots, r_k; \mu, \omega, a, b)$ be a joint probability density function of $r_1, \ldots, r_k$ with parameters (μ, ω, a, b). p_k is defined through the following sampling process: For each $k \geq 1$*

1. *Define ${\sigma_k}^2 = \omega + a{\epsilon_{k-1}}^2 + b{\sigma_{k-1}}^2$ with the convention that $\epsilon_0 = 0$ and $\sigma_0 = 0$.*
2. *Sample $e_k \sim \mathcal{N}(0, 1)$.*
3. *Set $\epsilon_k = \sigma_k e_k$ and $r_k = \epsilon_k + \mu$.*

Forecasting with GARCH. Now we introduce how to use the GARCH model to generate prediction intervals. The stochastic model specifies the conditional distribution

$$r_{K+1} | r_1, \ldots, r_K \sim \mathcal{N}(\mu, \sigma_{K+1}^2).$$

Therefore, once we know the values of σ_{K+1} and μ, we know the conditional distribution $r_{K+1}|r_1, \ldots, r_K$. With a bit of algebra, we can show that

$$\sigma_{K+1}^2 = \omega \frac{1 - b^{K+1}}{1 - b} + a \sum_{k=1}^{K} b^{K-k} (r_k - \mu)^2$$

The unknown variables in the equation above are (μ, ω, a, b), which can be estimated by applying MLE on historical data $r_1, r_2, \ldots, r_K$:

$$(\hat{\mu}, \hat{\omega}, \hat{a}, \hat{b}) = \arg \max_{(\mu, \omega, a, b)} \log p_K(r_1, \ldots, r_K; \mu, \omega, a, b).$$

We use python package `arch`[2] to perform the fitting. This gives an estimate of the conditional distribution $r_{K+1}|r_1, \ldots, r_K \sim \mathcal{N}(\hat{\mu}, \hat{\sigma}_{K+1}^2)$, where

$$\hat{\sigma}_{K+1}^2 = \hat{\omega} \frac{1 - \hat{b}^{K+1}}{1 - \hat{b}} + \hat{a} \sum_{k=1}^{K} \hat{b}^{K-k} (r_k - \hat{\mu})^2.$$

Let $Q_{\mu,\sigma} : [0, 1] \to [0, \infty]$ be the quantile function of the squared normal distribution $\mathcal{N}(\mu, \sigma)^2$, meaning that

$$\mathbb{P}(\mathcal{N}(\mu, \sigma^2)^2 \leq Q_{\mu,\sigma}(1 - \beta)) = 1 - \beta.$$

Using $Q_{\mu,\sigma}$, a natural prediction interval for r_{K+1} is

$$\hat{C}_{K+1}(\beta) = [Q_{\hat{\mu}, \hat{\sigma}_{K+1}^2}(\beta/2), Q_{\hat{\mu}, \hat{\sigma}_{K+1}^2}(1 - \beta/2)].$$

Intuitively, we expect $\hat{C}_{K+1}(\beta)$ to have mis-coverage rate β.

[2] https://arch.readthedocs.io/en/latest/univariate/introduction.html

9 Proofs

9.1 Proof of Theorem 1

Proof. Note that

$$\lambda_{m+K} = \lambda_m - \gamma \sum_{t=m+1}^{m+K} (\alpha - \mathrm{err}_\tau) \Rightarrow \left| \alpha - \frac{1}{K} \sum_{t=m+1}^{m+K} \mathrm{err}_t \right| = \frac{|\lambda_{m+K} - \lambda_{m+1}|}{K\gamma}. \tag{17}$$

Now we prove that $\lambda_k \in [-\gamma\bar{\alpha}, \lambda_{\max} + \gamma(1-\bar{\alpha})]$ for all k by induction. By assumption, this is true $k = 1$. Suppose the claim holds for some $k > 1$.

1. If $\lambda_k < 0$, by (5), $\alpha_k = 1$ and hence $\mathrm{err}_k = 1$. As a result,
$$-\gamma\bar{\alpha} \le \lambda_k < \lambda_{k+1} = \lambda_k + \gamma(1-\bar{\alpha}) < \gamma \le \lambda_{\max}.$$
2. If $\lambda_k > \lambda_{\max}$, then $\alpha_k = 0$ and hence $\mathrm{err}_k = 0$. As a result,
$$\lambda_{\max} + \gamma(1-\bar{\alpha}) \ge \lambda_k > \lambda_{k+1} = \lambda_k - \gamma\bar{\alpha} > \lambda_{\max} - \gamma\bar{\alpha} \ge \lambda_{\max} - \gamma > 0.$$
3. If $\lambda_k \in [0, \lambda_{\max}]$, then we either subtract $\gamma\bar{\alpha}$ from λ_k or add $\gamma(1-\bar{\alpha})$ onto λ_k. This guarantees that $\lambda_{k+1} \in [-\gamma\bar{\alpha}, \lambda_{\max} + \gamma(1-\bar{\alpha})]$.

The above arguments show that the induction hypothesis holds for $k+1$ and hence for every positive integer k. The proof is then completed by (17) with the observation that

$$|\lambda_{m+K} - \lambda_{m+1}| \le \lambda_{\max} + \gamma(1-\bar{\alpha}) + \gamma\bar{\alpha} = \lambda_{\max} + \gamma.$$

. □

9.2 Proof of Proposition 3.1

Proof. We shall use the induction on s. When $s = t + T$,

$$J_{s|t}(\rho) = \lambda_t \max\left(\rho/T - \bar{\alpha}, 0\right) \tag{18}$$

is increasing. Now we assume that $J_{s+1}(\rho)$ is non-decreasing in ρ. Then for any ρ

$$\begin{aligned} &J_{s|t}(\rho+1) - J_{s|t}(\rho) \\ &\quad = \left(J_{s+1|t}(\rho+1) - J_{s+1|t}(\rho)\right) + \min_\alpha \left\{L_{s|t}(\alpha) + D_{s|t}(\rho+1)F_{s|t}(\alpha)\right\} - \min_\alpha \left\{L_{s|t}(\alpha) + D_{s|t}(\rho)F_{s|t}(\alpha)\right\}. \end{aligned}$$

Write $\tilde{\alpha}_{s|t}(\rho+1)$ as α' for notational convenience. Then

$$\min_\alpha \left\{L_{s|t}(\alpha) + D_{s|t}(\rho+1)F_{s|t}(\alpha)\right\} = L_{s|t}(\alpha') + D_{s|t}(\rho+1)F_{s|t}(\alpha'),$$

and

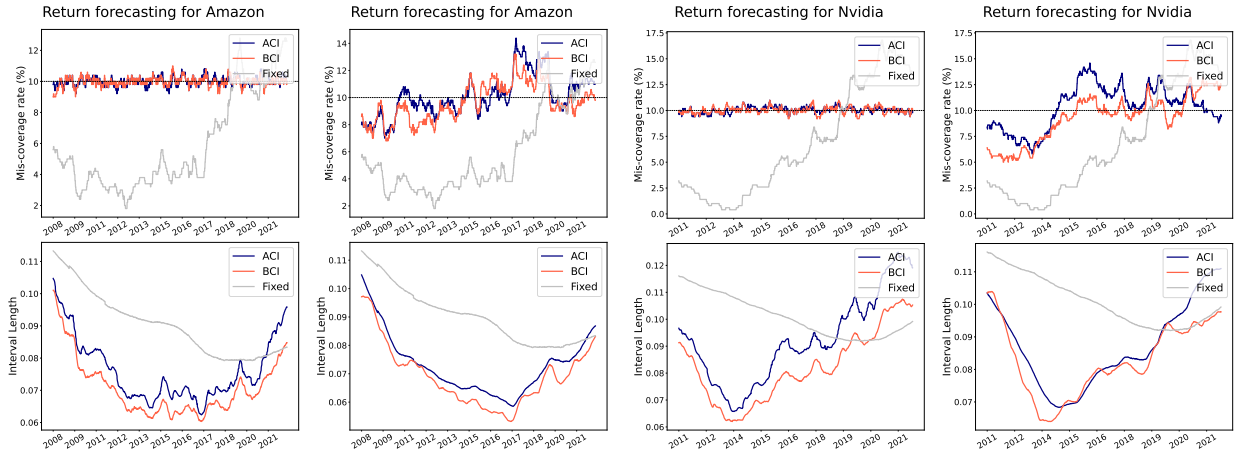
$$-\min_\alpha \left\{L_{s|t}(\alpha) + D_{s|t}(\rho)F_{s|t}(\alpha)\right\} \ge -\left\{L_{s|t}(\alpha') + D_{s|t}(\rho)F_{s|t}(\alpha')\right\}.$$

Therefore

$$\begin{aligned} J_{s|t}(\rho+1) - J_{s|t}(\rho) &\ge D_{s|t}(\rho) + \left\{L_{s|t}(\alpha') + D_{s|t}(\rho+1)F_{s|t}(\alpha')\right\} - \left\{L_{s|t}(\alpha') + D_{s|t}(\rho)F_{s|t}(\alpha')\right\}, \\ &\ge D_{s|t}(\rho)\left(1 - F_{s|t}(\alpha')\right) + D_{s|t}(\rho+1)F_{s|t}(\alpha'), \\ &\ge 0. \end{aligned}$$

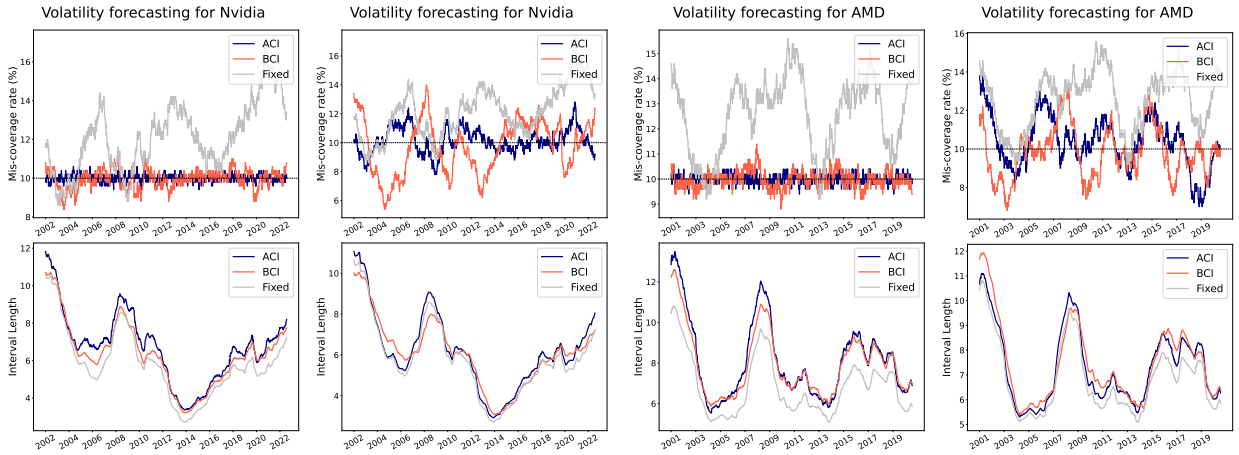
Since ρ is arbitrary, $J_{s|t}(\rho)$ is increasing in ρ. This completes the proof. □

10 More experiments results



(a) Amazon tight v.s. loose control (b) Nvidia tight v.s. loose control

Figure 5: Additional return forecasting problems.



(a) Nvidia tight v.s. loose control (b) AMD tight v.s. loose control

Figure 6: Additional volatility forecasting problems.