AIMS Mathematics, 10(3): 5143–5157.
DOI: 10.3934/math.2025236
Received: 10 January 2025
Revised: 21 February 2025
Accepted: 03 March 2025
Published: 07 March 2025

https://www.aimspress.com/journal/Math

Research article

Fast full conformal prediction for multiple test points

Ilsang Ohn* **and Jisu Park**

Department of Statistics, Inha University, 100 Inha-ro, Incheon 22212, Korea

* **Correspondence:** Email: ilsang.ohn@inha.ac.kr.

Abstract: Conformal prediction has emerged as a useful tool for providing valid predictive inference regardless of the data distribution. However, its implementation can be computationally intensive, even for small-scale data sets. Hence, it is typically prohibitive to construct full conformal prediction intervals for multiple test points, which limits its practicality. As an alternative, a sample-split approach can be used, but it usually provides wider prediction intervals, as it does not use all observations in the data for training. This paper attempts to fill this gap by developing a scalable conformal prediction algorithm for multiple test points. We find that when we use kernel ridge regression for the underlying prediction method, it is possible to reuse some computation in constructing prediction intervals across multiple test points, which enables us to avoid repeating the heavy computation of a matrix inverse for each test point. We propose an efficient algorithm that employs this fact, dramatically reducing the computational cost. We demonstrate the effectiveness and practical usefulness of the proposed algorithm in numerical experiments.

Keywords: conformal prediction; kernel ridge regression; block matrix inversion
Mathematics Subject Classification: 62G08, 62J02

1. Introduction

Machine learning methods, which aim to learn patterns from data and build predictive models, have achieved remarkable success in various fields. For example, in the manufacturing sector, machine learning is applied to optimize production processes and enable predictive maintenance, improving productivity [17]. In medicine, it helps improve diagnostic accuracy and personalized treatment [6]. In finance, machine learning is widely used for credit scoring, risk management, and fraud detection through anomaly analysis [9]. The increasing utility and reliance on machine learning in various industries highlights its growing importance.

However, over-confidence issues in machine learning methods carry inherent risks. For example, in finance, erroneous credit assessments or failures in fraud detection can result in substantial financial

damage, while in healthcare, misdiagnoses can pose serious threats to patient safety. Minimizing these risks is, therefore, a critical concern in machine learning. It is crucial to quantify prediction uncertainty and make cautious, reliable decisions based on the results.

In this context, several methods for quantifying uncertainty in machine learning have been developed and widely adopted. These include Bayesian neural networks [8], ensemble-based methods [14], and conformal prediction [13]. Among these, conformal prediction has received significant attention due to its appeal in that it provides a prediction interval of desired coverage property, only requiring very minimal assumption of the exchangeability of data without any additional distributional assumptions (e.g., the normality assumption in linear regression). Furthermore, conformal prediction can be applied to any machine learning algorithm, enabling uncertainty quantification without modifying existing models, making it highly versatile [13]. Conformal prediction has been applied effectively in diverse fields such as medical diagnosis [15], image classification [11], and natural language processing [5].

However, conformal prediction, which is also referred to as *full conformal prediction*, is computationally very inefficient as it requires repeat training of the machine learning model many times. This becomes more problematic when we compute prediction intervals for multiple test points. On the other hand, *split conformal prediction* is a substantially faster alternative, which divides the training data into two subsets: one for training the machine learning model and the other for the calibration of uncertainty. This approach requires only a single training step, which makes it computationally efficient. However, the width of an induced prediction interval is typically wider than that of full conformal prediction, as it does not use the entire dataset for training.

The main thrust of this paper is to improve the computational efficiency of full conformal prediction for multiple test points in the nonparametric regression framework. Specifically, we demonstrate that when the prediction of the employed machine learning method is a linear function of the output vector in the training data, such as kernel ridge regression, the computational cost of full conformal prediction for multiple test points can be significantly reduced.

The rest of this paper is organized as follows. In Section 2, we provide some background materials, including full conformal prediction, split conformal prediction, and kernel ridge regression. In Section 3, we propose a novel conformal prediction algorithm for multiple test points, which has substantially smaller computational complexity compared to a naive application of full conformal prediction. In Section 4, we conduct numerical studies to compare the efficiency of the proposed algorithm with existing methods. In Section 5, we discuss some possible extensions of the proposed approach to other setups. In Section 6, we conclude with the implications of our findings and possible future work.

For a set A, $|A|$ denotes its cardinality. For a real value z, we let $\lceil z \rceil$ denote the smallest integer larger than or equal to z, and $\lfloor z \rfloor$ the largest integer smaller than or equal to z. We use a standard big-O notation. Let $\mathbb{I}$ be the indicator function.

2. Preliminary

2.1. Conformal prediction

Suppose that we have a training sample $(X_1, Y_1), \ldots, (X_n, Y_n)$ where each X_i and Y_i denote the input (or feature, or predictor)) and output (or response) variables of the i-th observation, respectively. In

this paper, we focus on the regression setup where the output variables are real-valued, and the input variables are multivariate. An aim of conformal prediction is to construct a prediction interval for a test output Y_{n+1} associated with a test input X_{n+1} with guaranteed coverage. The desired coverage can be attained only assuming the exchangeability of the data as follows.

Assumption 1 (Exchangeability)**.** Let $Z_i = (X_i, Y_i)$ for $i = 1, \ldots, n, n+1$. Then $Z_1, \ldots, Z_n, Z_{n+1}$ are exchangeable, that is, for any permutation $\pi : \{1, \ldots, n, n+1\} \mapsto \{1, \ldots, n, n+1\}$,

$$(Z_1, \ldots, Z_n, Z_{n+1}) \stackrel{d}{=} (Z_{\pi(1)}, \ldots, Z_{\pi(n)}, Z_{\pi(n+1)}),$$

where $\stackrel{d}{=}$ means that the left and right sides are equal in distribution.

We introduce full and split conformal prediction approaches.

2.1.1. Full conformal prediction

Full conformal prediction is conducted as follows. For each $y \in \mathbb{R}$, a "candidate" of the unknown output Y_{n+1}, we train a machine learning method, say $\hat{f}$, based on the "augmented" training sample $\{(X_1, Y_1), \ldots, (X_n, Y_n), (X_{n+1}, y)\}$. We denote by $\hat{f}^y_{1:(n+1)}$ the trained prediction function. Then we compute the absolute value of the residual $R^y_i := Y_i - \hat{f}^y_{1:(n+1)}(X_i)$ for $i = 1, \ldots, n$ and $R^y_{n+1} := y - \hat{f}^y_{1:(n+1)}(X_{n+1})$. We compute the number of residuals whose absolute values are not larger than the one R^y_{n+1} associated with the test point, which is denoted as

$$\xi(y, X_{n+1}) := \sum_{i=1}^{n+1} \mathbb{I}\left(|R^y_i| \leq |R^y_{n+1}|\right) = 1 + \sum_{i=1}^{n} \mathbb{I}\left(|R^y_i| \leq |R^y_{n+1}|\right).$$

Lastly, we construct a prediction interval with coverage $1 - \alpha \in (0, 1)$ as

$$\hat{C}^{\text{full}}_\alpha(X_{n+1}) := \{y \in \mathbb{R} : \xi(y, X_{n+1}) \leq \lceil (1-\alpha)(n+1) \rceil\}.$$

Under Assumption 1, this interval achieves the desired coverage:

$$\mathbb{P}\left(Y_{n+1} \in \hat{C}^{\text{full}}_\alpha(X_{n+1})\right) \geq 1 - \alpha.$$

The proof can be found in Theorem 2.1 in [10]. But computing such an interval is impossible (except for some machine learning methods) since we cannot repeat the procedure for any real value y. In practice, an approximate "grid" approach can be used, which only considers a finite number of candidate values, say $y^*_1, \ldots, y^*_q$. Then, the full conformal prediction interval is computed as

$$\hat{C}^{\text{full, grid}}_\alpha(X_{n+1}) = \text{Range}\left(\left\{y^*_j : \xi(y^*_j, X_{n+1}) \leq \lceil (1-\alpha)(n+1) \rceil\right\}\right),$$

where $\text{Range}(A)$ denotes the interval $[\min_i a_i, \max_i a_i]$ for a set $A = \{a_i\}$. This is computationally intensive as this requires q training processes.

2.1.2. Split conformal prediction

Split conformal prediction, unlike the full conformal method, requires only one training process. This approach first splits the training sample into two disjoint sets $\mathcal{I}_1$ and $\mathcal{I}_2$ with $\mathcal{I}_1 \cup \mathcal{I}_2 = \{1, \ldots, n\}$. Usually, these two subsets are equal in size. Then we train a machine learning method, say $\hat{f}$, on the sub-sample $\{(X_i, Y_i) : i \in \mathcal{I}_1\}$. We denote by $\hat{f}_{\mathcal{I}_1}$ the trained prediction function. Then we compute the absolute residuals $\bar{R}_i := |Y_i - \hat{f}_{\mathcal{I}_1}(X_i)|$ for $i \in \mathcal{I}_2$, that is, for the observations in $\mathcal{I}_2$. Lastly, we construct a prediction interval with coverage $1 - \alpha \in (0, 1)$ as

$$\hat{C}_\alpha^{\text{split}}(X_{n+1}) := \left[\hat{f}_{\mathcal{I}_1}(X_{n+1}) - \gamma, \hat{f}_{\mathcal{I}_1}(X_{n+1}) + \gamma\right],$$

where γ is the t-th smallest value in $\{\bar{R}_i : i \in \mathcal{I}_2\}$ with $t := \lceil (1 - \alpha)(|\mathcal{I}_2| + 1)\rceil$. Under Assumption 1, this interval attains the desired coverage:

$$\mathbb{P}\left(Y_{n+1} \in \hat{C}_\alpha^{\text{split}}(X_{n+1})\right) \geq 1 - \alpha.$$

The proof can be found in Theorem 2.2 in [10]. However, this approach is statistically less inefficient than the full conformal prediction approach since only a part of the data is used for training. This means that $\hat{C}_\alpha^{\text{split}}(X_{n+1})$ is typically wider than $\hat{C}_\alpha^{\text{full}}(X_{n+1})$.

2.2. Kernel ridge regression

Kernel ridge regression (KRR) aims to estimate the conditional expectation $f_\star(x) := \mathbb{E}(Y|X = x)$ of the output $Y \in \mathbb{R}$ given an input value $X = x \in \mathbb{R}^d$, based on a sample $(X_1, Y_1), \ldots (X_n, Y_n)$ which are assumed to be i.i.d. copies of (X, Y). For notational simplicity, we write

$$X_{1:n} = (X_1, \ldots, X_n)^\top \in \mathbb{R}^{n\times d}, \quad Y_{1:n} = (Y_1, \ldots, Y_n)^\top \in \mathbb{R}^n.$$

Let $\kappa : \mathbb{R}^d \times \mathbb{R}^d \mapsto \{z \in \mathbb{R} : z \geq 0\}$ be a positive definite kernel function and $\mathcal{H}$ be the associated reproducing kernel Hibert space (RKHS) endowed with the RKHS norm $\|\cdot\|_{\mathcal{H}}$. A KRR estimator of $f_\star$ is obtained by solving the following regularized empirical risk minimization problem

$$\hat{f} = \underset{f\in\mathcal{H}}{\operatorname{argmin}} \left[\sum_{i=1}^{n}(Y_i - f(X_i))^2 + \lambda\|f\|_{\mathcal{H}}\right]$$

over the RKHS $\mathcal{H}$, where $\lambda > 0$ is a regularization parameter that will be tuned to optimize the bias-variance trade-off. This has a closed-form solution given by

$$\hat{f}(x) = \kappa(x, X_{1:n})(\kappa(X_{1:n}, X_{1:n}) + \lambda I)^{-1} Y_{1:n}, \tag{2.1}$$

where we denote

$$\kappa(x, X_{1:n}) := (\kappa(x, X_1), \ldots, \kappa(x, X_n))^\top \in \mathbb{R}^{1\times n}$$

and

$$\kappa(X_{1:n}, X_{1:n}) := (\kappa(X_i, X_j))_{i,j=1,\ldots,n} \in \mathbb{R}^{n\times n}.$$

2.3. Conformal prediction with KRR

In this subsection, we introduce some implementations of full conformal prediction using KRR as a machine learning method. The KRR estimator trained based on the augmented sample $\{(X_1, Y_1), \ldots, (X_n, Y_n), (X_{n+1}, y)\}$ is given by

$$\hat{f}^y_{1:(n+1)}(x) := \kappa(x, X_{1:(n+1)l})(K_{1:(n+1)} + \lambda I)^{-1} Y_{1:n,y},$$

where

$$\kappa(x, X_{1:(n+1)}) := (\kappa(x, X_1), \ldots, \kappa(x, X_n)), \kappa(x, X_{n+1})^\top \in \mathbb{R}^{1\times n},$$
$$K_{1:(n+1)} := \kappa(X_{1:(n+1)}, X_{1:(n+1)}) := (\kappa(X_i, X_j))_{i,j=1,\ldots,n,n+1} \in \mathbb{R}^{(n+1)\times(n+1)},$$

and

$$Y_{1:n,y} := \begin{pmatrix} Y_{1:n} \\ y \end{pmatrix} \in \mathbb{R}^{n+1}.$$

The predicted value for $Y_{1:n,y}$ is then

$$\begin{aligned} \hat{Y}^y_{1:(n+1)} &:= (\hat{f}^y_{1:(n+1)}(X_1), \cdots, \hat{f}^y_{1:(n+1)}(X_n), \hat{f}^y_{1:(n+1)}(X_{n+1}))^\top \\ &= K_{1:(n+1)}(K_{1:(n+1)} + \lambda I)^{-1} Y_{1:n,y}. \end{aligned}$$

For convenience, we define

$$S^\lambda_{1:(n+1)} := K_{1:(n+1)}(K_{1:(n+1)} + \lambda I)^{-1},$$

which is the "smoother" matrix multiplied to the observed output vector $Y_{1:n,y}$ to produce its predicted value. The vector of the residuals

$$R^y_{1:(n+1)} = (R^y_1, \ldots, R^y_n, R^y_{n+1})$$

is represented as

$$(I - S^\lambda_{1:(n+1)})Y_{1:n,y}.$$

Furthermore, it is easy to see that

$$R^y_{1:(n+1)} = a_{1:(n+1)} + y b_{1:(n+1)}$$

where

$$\begin{aligned} a_{1:(n+1)} &= (a_1, \ldots, a_n, a_{n+1})^\top := (I - S^\lambda_{1:(n+1)})(Y_1, \ldots, Y_n, 0)^\top, \\ b_{1:(n+1)} &= (b_1, \ldots, b_n, b_{n+1})^\top := (I - S^\lambda_{1:(n+1)})(0, \ldots, 0, 1)^\top. \end{aligned}$$

This implies that each residual is a linear function of the candidate value y. Based on this observation, [16] developed an efficient algorithm for constructing the full conformal prediction set with KRR. The detailed algorithm is given in Algorithm 1. Provided that the computation of the smoother matrix $S^\lambda_{1:(n+1)}$ is done, the computational complexity of this algorithm is $O(n \log n)$, which is smaller than the $O(n^2 q)$ of the grid approach described in Section 2.1.1.

Algorithm 1 Full conformal prediction with KRR by [16]

Set $\mathcal{Y} = \emptyset$
for $i \leftarrow 1$ to $n+1$ **do**
  if $b_i < 0$ **then**
    Change the signs of a_i and b_i, i.e., $a_i := -a_i$ and $b_i := -b_i$
  end if
  if $b_i \neq b_{n+1}$ **then**
    Update $\mathcal{Y}$ as $\mathcal{Y} := \mathcal{Y} \cup \{-(a_i - a_{n+1})/(b_i - b_{n+1}), -(a_i + a_{n+1})/(b_i + b_{n+1})\}$
  end if
  if $b_i = b_{n+1} \neq 0$ and $a_i \neq a_{n+1}$ **then**
    Update $\mathcal{Y}$ as $\mathcal{Y} := \mathcal{Y} \cup \{-(a_1 + a_{n+1})/(2b_i)\}$
  end if
end for
Sort the elements of $\mathcal{Y}$ to get $y_{(1)} \leq \cdots \leq y_{(r)}$ with $r := |\mathcal{Y}|$
Set $y_{(0)} := -\infty$ and $y_{(r+1)} := \infty$
for $j \leftarrow 0$ to r **do**
  Compute $N_j := \sum_{j=0}^{n} \mathbb{I}((y_{(j)}, y_{(j+1)}) \subset A_i)$ and $M_j := \sum_{j=0}^{n} \mathbb{I}(y_{(j)} \in A_i)$ where $A_i := \left\{y \in \mathbb{R} : |R_i^y| \geq |R_{n+1}^y|\right\}$
end for
return $\hat{C}_\alpha^{\text{full, KRR}}(X_{n+\ell}) := \bigcup_{j:N_j > (n+1)\alpha}(y_{(j)}, y_{(j+1)}) \bigcup \left\{y_{(j)} : M_j > (n+1)\alpha\right\}$.

The work [2] and the subsequent work [3] proposed a different approach for conformal prediction with KRR. Instead of using the absolute value of the residual $|R_i^y|$, they proposed to use the "two-sided exceptionality" of the residual given as

$$G_i^y := \min\left\{\sum_{j=1}^{n+1} \mathbb{I}(R_j^y \geq R_i^y), \sum_{j=1}^{n+1} \mathbb{I}(R_j^y \leq R_i^y)\right\}$$

as the nonconformity measure for the i-th observation. Then, the full conformal prediction set is given by

$$\hat{C}_\alpha^{\text{full, KRR-ts}}(X_{n+1}) := \{y \in \mathbb{R} : \xi_G(y, X_{n+1}) \leq \lceil (1-\alpha)(n+1) \rceil\},$$

where

$$\xi_G(y, X_{n+1}) = \sum_{i=1}^{n+1} \mathbb{I}(G_i^y \leq G_{n+1}^y).$$

The construction of such a prediction set can be done with $O(n \log n)$ complexity provided that the computation of the smoother matrix $S_{1:(n+1)}^{\lambda}$ is done. The detailed algorithm is given in Algorithm 2.

These two algorithms improve the computational efficiency of the construction of the prediction interval but do not concern the efficiency of the computation of the smoother matrix. But when there are multiple test points, we need to compute the smoother matrix for each test point, so it would be a substantial computational bottleneck. This paper proposes an efficient computational algorithm to resolve this issue.

Algorithm 2 Two-sided full conformal prediction with KRR by [2, 3]

for $i \leftarrow 1$ to n **do**
  if $b_{n+1} - b_i > 0$ **then**
    Set $u_i := l_i = (a_i - a_{n+1})/(b_{n+1} - b_i)$
  else
    Set $l_i =: -\infty$ and $u_i := \infty$.
  end if
end for
Sort $u_1, \ldots, u_n$ to get $u_{(1)} \leq \cdots \leq u_{(n)}$ and $l_1, \ldots, l_n$ to get $l_{(1)} \leq \cdots \leq l_{(n)}$
return $\hat{C}_\alpha^{\text{full, KRR-ts}}(X_{n+\ell}) := \left[l_{(\lfloor(\alpha/2)(n+1)\rfloor)}, u_{(\lceil(1-\alpha/2)(n+1)\rceil)}\right]$.

3. Full conformal prediction with KRR for multiple test points

In this section, we describe our efficient algorithm for full conformal prediction based on KRR. Suppose that we are given m test inputs $X_{n+1}, \ldots, X_{n+m}$ as well as n training samples $(X_1, Y_1), \ldots, (X_n, Y_n)$. For each test input $X_{n+\ell}$, we define the augmented input matrix as

$$X_{1:n,n+\ell} := \begin{pmatrix} X_{1:n} \\ X_{n+\ell}^\top \end{pmatrix} \in \mathbb{R}^{(n+1)\times d}.$$

Then, in the same way illustrated in Section 2.3, we know that the predicted values for $Y_{1:n,y}$ is given by

$$\hat{Y}^y_{1:n,n+\ell} = S^\lambda_{1:n,n+\ell} Y_{1:n,y},$$

where

$$S^\lambda_{1:n,n+\ell} := K_{1:n,n+\ell}(K_{1:n,n+\ell} + \lambda I)^{-1},$$

where

$$K_{1:n,n+\ell} := \kappa(X_{1:n,n+\ell}, X_{1:n,n+\ell}) := (\kappa(X_i, X_j))_{i,j=1,\ldots,n,n+\ell} \in \mathbb{R}^{(n+1)\times(n+1)}.$$

A naive approach to compute $S^\lambda_{1:n,n+\ell}$ at each $X_{n+\ell}$ has the computational complexity of $O(n^3 m)$. However, we can substantially reduce it by leveraging the following black matrix inversion formula.

Lemma 1. *Let B be a square matrix such that*

$$B = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix},$$

where B_{11} and B_{22} are square matrices. Assume that B_{11} and $B_{22} - B_{21}B_{11}^{-1}B_{12}$ are invertible. Then the inverse B^{-1} is given by

$$B^{-1} = \begin{pmatrix} B_{11}^{-1} + B_{11}^{-1}B_{12}\Delta B_{21}B_{11}^{-1} & -B_{11}^{-1}B_{12}\Delta \\ -\Delta B_{21}B_{11}^{-1} & \Delta \end{pmatrix},$$

where we define $\Delta := (B_{22} - B_{21}B_{11}^{-1}B_{12})^{-1}$.

Proof. This result is very well known, but we give the proof for the completeness. We start with the identity

$$\begin{pmatrix} I & O \\ -B_{21}B_{11}^{-1} & I \end{pmatrix}\begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}\begin{pmatrix} I & -B_{11}^{-1}B_{12} \\ O & I \end{pmatrix} = \begin{pmatrix} B_{11} & O \\ O & B_{22} - B_{21}B_{11}^{-1}B_{12} \end{pmatrix},$$

where O is a matrix of zeros. Taking the inverse to both sides, we get

$$B^{-1} = \begin{pmatrix} I & -B_{11}^{-1}B_{12} \\ O & I \end{pmatrix}\begin{pmatrix} B_{11}^{-1} & O \\ O & \Delta \end{pmatrix}\begin{pmatrix} I & O \\ -B_{21}B_{11}^{-1} & I \end{pmatrix},$$

which proves the desired. □

Now, we apply Lemma 1 to compute the inverse of the matrix

$$K_{1:n,n+\ell} + \lambda I = \begin{pmatrix} K_{1:n} + \lambda I & L_{n+\ell} \\ L_{n+\ell}^\top & 1 + \lambda \end{pmatrix},$$

where we normalize the kernel function so that $\kappa(x, x) = 1$ for any $x \in \mathbb{R}^d$, and we denote

$$\begin{aligned} K_{1:n} &:= \kappa(X_{1:n}, X_{1:n}) := (\kappa(X_i, X_j))_{i,j=1,\dots,n}, \\ L_{n+\ell} &:= \kappa(X_{n+\ell}, X_{1:n}) := (\kappa(X_{n+\ell}, X_1), \dots, \kappa(X_{n+\ell}, X_n))^\top. \end{aligned}$$

We first define $\delta_{n+\ell}$ as

$$\delta_{n+\ell} := \frac{1}{(1+\lambda) - L_{n+\ell}^\top H_{1:n}^\lambda L_{n+\ell}} \tag{3.1}$$

where we define $H_{1:n}^\lambda := (K_{1:n,n+\ell} + \lambda I)^{-1}$. Then by Lemma 1, we have

$$H_{1:n,n+\ell}^\lambda := (K_{1:n,n+\ell} + \lambda I)^{-1} = \begin{pmatrix} H_{1:n}^\lambda + \delta_{n+\ell} H_{1:n}^\lambda L_{n+\ell} L_{n+\ell}^\top H_{1:n}^\lambda & -\delta_{n+\ell} H_{1:n}^\lambda L_{n+\ell} \\ -\delta_{n+\ell} L_{n+\ell}^\top H_{1:n}^\lambda & \delta_{n+\ell} \end{pmatrix}.$$

Since we have

$$\begin{aligned} K_{1:n,n+\ell} H_{1:n,n+\ell}^\lambda &= (K_{1:n,n+\ell} + \lambda I) H_{1:n,n+\ell}^\lambda - \lambda H_{1:n,n+\ell}^\lambda \\ &= I - \lambda H_{1:n,n+\ell}^\lambda, \end{aligned}$$

the smoother of the KRR estimator is given by

$$\begin{aligned} S_{1:n,n+\ell}^\lambda = K_{1:n,n+\ell} H_{1:n,n+\ell}^\lambda &= \begin{pmatrix} K_{1:n} & L_{n+\ell} \\ L_{n+\ell}^\top & 1 \end{pmatrix}\begin{pmatrix} H_{1:n}^\lambda + \delta_{n+\ell} H_{1:n}^\lambda L_{n+\ell} L_{n+\ell}^\top H_{1:n}^\lambda & -\delta_{n+\ell} H_{1:n}^\lambda L_{n+\ell} \\ -\delta_{n+\ell} L_{n+\ell}^\top H_{1:n}^\lambda & \delta_{n+\ell} \end{pmatrix} \\ &= \begin{pmatrix} K_{1:n} H_{1:n}^\lambda + \delta_{n+\ell}(K_{1:n} H_{1:n}^\lambda - I) L_{n+\ell} L_{n+\ell}^\top H_{1:n}^\lambda & -\delta_{n+\ell}(K_{1:n} H_{1:n}^\lambda - I) L_{n+\ell} \\ \delta_{n+\ell}(\delta_{n+\ell}^{-1} + L_{n+\ell}^\top H_{1:n}^\lambda L_{n+\ell} - 1) L_{n+\ell}^\top H_{1:n}^\lambda & \delta_{n+\ell}(1 - L_{n+\ell}^\top H_{1:n}^\lambda L_{n+\ell}) \end{pmatrix} \\ &= \begin{pmatrix} I - \lambda H_{1:n}^\lambda - \lambda \delta_{n+\ell} H_{1:n}^\lambda L_{n+\ell} L_{n+\ell}^\top H_{1:n}^\lambda & \lambda \delta_{n+\ell} H_{1:n}^\lambda L_{n+\ell} \\ \lambda \delta_{n+\ell} L_{n+\ell}^\top H_{1:n}^\lambda & 1 - \lambda \delta_{n+\ell} \end{pmatrix}. \end{aligned} \tag{3.2}$$

Here the last identity holds due to

$$\begin{aligned}\delta_{n+\ell}(1 - L_{n+\ell}^{\top} H_{1:n}^{\lambda} L_{n+\ell}) &= \delta_{n+\ell}(\delta_{n+\ell}^{-1} - \lambda) \\ &= 1 - \lambda\delta_{n+\ell}.\end{aligned}$$

In the computation in (3.2), computing $H_{1:n}^{\lambda}$, which has the computational complexity of $O(n^3)$ and so is the main computational bottleneck, can be conducted only once, since it does not depend on test inputs. What we need to compute for every test input is the multiplication of $L_{n+\ell}^{\top}$ and $H_{1:n}^{\lambda}$, which has the complexity of $O(n^2)$. Thus, as the construction algorithm in either Algorithm 1 or Algorithm 2 requires $O(n \log n)$ complexity, the computational complexity of computing the full conformal prediction intervals for m test points can be reduced to

$$O(n^3 + n^2 m).$$

This is substantially smaller than $O(n^3 m)$, which is the complexity of simply applying full conformal prediction with KRR to multiple test points.

The proposed procedure is summarized in Algorithm 3.

Algorithm 3 Full conformal prediction with KRR for multiple test points

Input: Training sample $(X_i, Y_i), i = 1, \ldots, n$, Test inputs $X_{n+1}, \ldots, X_{n+m}$, Confidence level $\alpha \in (0, 1)$, Regularization parameter λ, Kernel function κ

Output: Prediction intervals for $Y_{n+1} \ldots, Y_{n+\ell}$.

Compute $H_{1:n}^{\lambda}$
for $\ell \leftarrow 1$ to m **do**
  Compute $\delta_{n+\ell}$ according to (3.1)
  Compute $S_{1:n,n+\ell}^{\lambda}$ according to (3.2)
  Compute $a_{1:n,n+\ell} := (I - S_{1:n,n+\ell}^{\lambda})(Y_1, \ldots, Y_n, 0)^{\top}$ and $b_{1:n,n+\ell} := (I - S_{1:n,n+\ell}^{\lambda})(0, \ldots, 0, 1)^{\top}$
  Construct a $1 - \alpha$ prediction interval for $Y_{n+\ell}$ using either Algorithm 1 or Algorithm 2.
end for

4. Numerical experiments

In this section, we conduct a numerical study to support the effectiveness and efficiency of the proposed algorithm for full conformal prediction.

4.1. Data

For our simulation, we consider both synthetic datasets and real-world datasets.

Synthetic data We generate both training and test samples from the following Gaussian linear regression model

$$Y_i | X_i \overset{\text{ind}}{\sim} N(X_i^{\top}\beta, 1), \quad X_i \overset{\text{iid}}{\sim} N(0, I)$$

for $i = 1, \ldots, n+m$. For true regression coefficients β, we generate each element from the uniform distribution on $[-1, 1]$ independently. We set the dimension of the input as $d = 20$, the number of training points as $n = 200$ and that of test points as $m = 100$.

Real data We consider the following two real-world datasets, the Communities and Crime dataset [12] and the BlogFeedback dataset [4]. The first dataset contains 1,994 data points. There remain $d = 99$ covariates after preprocessing the data following the same procedure as in [1]. The second dataset contains information on 52,397 observations, each of which has $d = 280$ covariates. We apply the log transform to the output variable since it is extremely skewed. For each simulation, we randomly select $n = 200$ training points from the whole dataset and then randomly select $n = 100$ test points from the remaining data points.

4.2. Methods

For the full conformal prediction method with KRR, we use the radial basis kernel function given as

$$\kappa(x, x') = \exp\left(-\|x - x'\|^2/d\right)$$

where d denotes the input dimension. Such a choice of the scale parameter is the same as the default setup `Scikit-learn` package in `Python`. The regularization parameter λ is chosen by a cross-validation process. For prediction interval construction, we consider the two approaches described in Section 2.3: Algorithm 1 (will be abbreviated as Full-KRR-Alg1), and Algorithm 2 (Full-KRR-Alg2).

We compare the proposed full conformal prediction method with split conformal prediction with three different machine learning methods: KRR, random forest (RF) and Lasso. For KRR, we use the same kernel function and the regularization parameter used for our full conformal prediction with KRR approach. The split conformal prediction method with RF is implemented with the default settings provided by the `conformalInference` package in `R`. For the use of Lasso, the tuning parameter was selected through a cross-validation process. In addition, for the synthetic data, we construct a parametric prediction interval for Gaussian linear models, which provides valid coverage under the data-generating process of the synthetic data.

4.3. Results

For each dataset, we report the averaged coverage, width, and computation time of the predictive intervals with their standard errors across 50 independent trials. We choose a target coverage of

$$1 - \alpha = 0.9.$$

Table 1 presents the results on the synthetic datasets generated from the linear regression model. Notably, the performance of the full conformal prediction approach is comparable to that of the parametric method, whose assumptions align with our data-generating process. Otherwise, all split conformal methods provide wider prediction intervals than the full conformal method. In particular, using non-linear function estimation methods such as KRR and RF performs substantially worse than the other methods.

Tables 2 and 3 present the results on the real-world datasets. All methods attain the desired coverage. The proposed full conformal prediction method with Algorithm 1 gives narrower prediction intervals than all split conformal methods for both datasets. This illustrates the statistical efficiency of our method. On the other hand, using Algorithm 2, which uses a different nonconformity score than the first method, provides less efficient results. Moreover, the computation time of the proposed Algorithm 3 is

comparable to the split conformal prediction methods and even smaller than those with RF and Lasso. This result illustrates the practicality of our method.

Table 1. The coverage, width and computation time of the predictive intervals for all methods, averaged over 50 repetitions of synthetic data generation.

	Coverage	**Width**	**Time (sec)**
Full-KRR-Alg1	0.895 (0.039)	3.456 (0.237)	0.138 (0.035)
Full-KRR-Alg2	0.901 (0.039)	3.528 (0.246)	0.076 (0.022)
Split-KRR	0.897 (0.044)	3.729 (0.352)	0.002 (0.003)
Split-RF	0.901 (0.037)	6.189 (0.651)	0.092 (0.007)
Split-Lasso	0.906 (0.047)	3.742 (0.382)	0.023 (0.01)
Parametric	0.899 (0.038)	3.432 (0.188)	0.002 (0)

Table 2. The coverage, width and computation time of the predictive intervals for all methods, averaged over 50 repetitions of data extraction from Communities and Crime data set.

	Coverage	**Width**	**Time (sec)**
Full-KRR-Alg1	0.896 (0.043)	0.387 (0.033)	0.141 (0.024)
Full-KRR-Alg2	0.899 (0.04)	0.44 (0.042)	0.078 (0.011)
Split-KRR	0.907 (0.043)	0.493 (0.072)	0.005 (0.001)
Split-RF	0.904 (0.043)	0.508 (0.074)	0.332 (0.014)
Split-Lasso	0.897 (0.049)	0.513 (0.081)	0.582 (0.161)

Table 3. The coverage, width and computation time of the predictive intervals for all methods, averaged over 50 repetitions of data extraction from the BlogFeedback dataset.

	Coverage	**Width**	**Time (sec)**
Full-KRR-Alg1	0.899 (0.034)	2.082 (0.21)	0.124 (0.012)
Full-KRR-Alg2	0.906 (0.031)	2.773 (0.257)	0.074 (0.02)
Split-KRR	0.897 (0.041)	3.522 (0.768)	0.009 (0.004)
Split-RF	0.902 (0.038)	2.777 (0.429)	0.401 (0.032)
Split-Lasso	0.906 (0.038)	3.366 (0.727)	0.146 (0.04)

4.4. Time comparison with the naive approach

In this experiment, we aim to illustrate how much computation time is reduced by using the proposed algorithm compared to the naive approach that repeats computing the smoother matrix for each test point. We set the number of training points to 200 but vary the number of test points from 100 to 1,000 with an increment of 100. The result is presented in Figure 1, where we see that the computation time of our method shows a much slower rate of increase than the naive approach, indicating its superior efficiency and scalability.

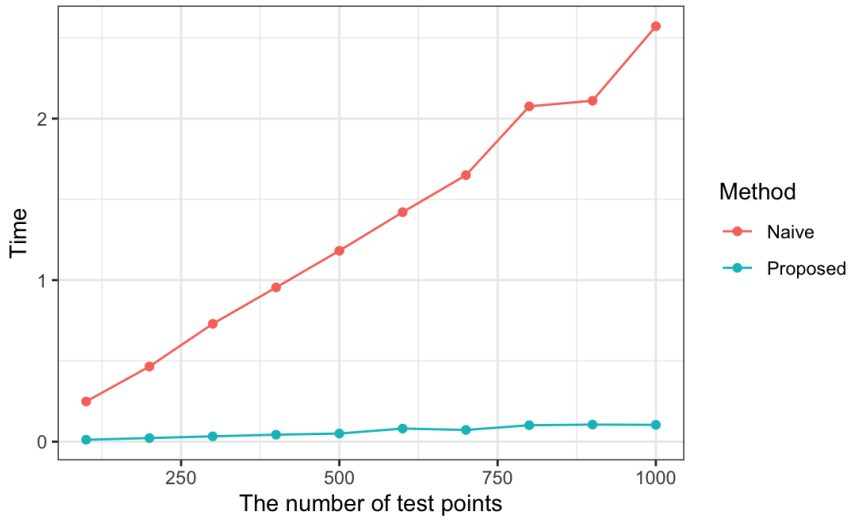

Figure 1. Computation time versus the number of test points.

5. Possible extensions

In this section, we discuss some possible extensions of the proposed approach.

5.1. Online learning

Our Algorithm 3 is naturally adapted to an online learning setup. In this setup, we assume that the pair of input and output comes sequentially and aim to predict the next output given the previously observed pairs. That is, we construct a prediction interval for Y_{t+1} given the observed sample $(X_1, Y_1), \ldots, (X_t, Y_t)$ for each t. The corresponding smoother of KRR is given by

$$S^{\lambda}_{1:(t+1)} := K_{1:(t+1)}(K_{1:(t+1)} + \lambda I)^{-1}.$$

But, according to Eq 3.2, this can be updated efficiently from the previous computed inverse matrix $H^{\lambda}_{1:t} := (K_{1:t} + \lambda I)^{-1}$ which was used to construct a prediction interval for Y_t. Namely,

$$S^{\lambda}_{1:(t+1)} = \begin{pmatrix} I - \lambda H^{\lambda}_{1:t} - \lambda\delta_{t+1} H^{\lambda}_{1:t} L_{t+1} L_{t+1}^{\top} H^{\lambda}_{1:t} & \lambda\delta_{t+1} H^{\lambda}_{1:t} L_{t+1} \\ \lambda\delta_{t+1} L_{t+1}^{\top} H^{\lambda}_{1:n} & 1 - \lambda\delta_{t+1} \end{pmatrix}$$

where $L_{t+1} := \kappa(X_{t+1}, X_{1:t})$ and $\delta_{t+1} := (1 + \lambda - L_{t+1}^{\top} H^{\lambda}_{1:t} L_{t+1})^{-1}$.

5.2. Other machine learning methods

This study focuses on the KRR prediction method, but the proposed algorithm can be applied to any least-squares or ridge-type prediction method. Consider a prediction function of the form

$$f(x) = \phi(x)^{\top}\beta,$$

where ϕ is a fixed "feature" map chosen by an user and β is a learnable coefficient. For example, if $\phi(x) = x$, this recovers the standard linear model. Also, this model includes various basis expansion

methods, such as splines and piecewise polynomials. Given the augmented data $(X_1, Y_1), \ldots, (X_n, Y_n), (X_{n+1}, y)$ for full conformal prediction, suppose that we estimate β by minimizing the following objective function

$$\sum_{i=1}^{n}(Y_i - \phi(X_i)^\top\beta)^2 + (y - \phi(X_{n+1})^\top\beta)^2 + \lambda\|\beta\|_2^2$$

with tuning parameter $\lambda \geq 0$. Note that the choice of $\lambda = 0$ leads to the least-squares estimation. Then the resulting ridge-type predicted value is given by

$$\hat{Y}^y_{1:(n+1)} = \Phi_{1:(n+1)}(\Phi^\top_{1:(n+1)}\Phi_{1:(n+1)} + \lambda I)^{-1}\Phi^\top_{1:(n+1)}Y_{1:n,y}$$

where

$$\Phi_{1:(n+1)} := (\phi(X_1), \ldots, \phi(X_n), \phi(X_{n+1}))^\top$$

denotes the matrix of the feature values. The computation of the above estimator requires the computation of the inverse matrix $(\Phi^\top_{1:(n+1)}\Phi_{1:(n+1)} + \lambda I)^{-1}$. Therefore, to construct multiple conformal prediction intervals, we need to compute many inverse matrices, which can be a substantial computational burden. We can solve this issue using the same idea as the proposed algorithm. We first note that the ridge-type estimator admits the dual form

$$(\Phi^\top_{1:(n+1)}\Phi_{1:(n+1)} + \lambda I)^{-1}\Phi^\top_{1:(n+1)} = \Phi^\top_{1:(n+1)}(\Phi_{1:(n+1)}\Phi^\top_{1:(n+1)} + \lambda I)^{-1},$$

which allows the following representation of the predicted value

$$\hat{Y}^y_{1:(n+1)} = \Phi_{1:(n+1)}\Phi^\top_{1:(n+1)}(\Phi_{1:(n+1)}\Phi^\top_{1:(n+1)} + \lambda I)^{-1}Y_{1:n,y}.$$

In the above display, the matrix to be inversed has a block structure such that

$$\Phi_{1:(n+1)}\Phi^\top_{1:(n+1)} + \lambda I = \begin{pmatrix} \Phi_{1:n}\Phi^\top_{1:n} + \lambda I & \Phi_{1:n}\phi(X_{n+1}) \\ \phi(X_{n+1})^\top\Phi^\top_{1:n} & \phi(X_{n+1})^\top\phi(X_{n+1}) + \lambda \end{pmatrix}.$$

Thus, employing Lemma 1, a similar efficient algorithm as Algorithm 3 is easily derived.

6. Conclusions

In this paper, we propose an efficient algorithm for computing multiple prediction intervals using the full conformal prediction approach. The proposed algorithm employs KRR as a prediction method, which requires the computation of complexity $O(n^3)$ for training. Thus, for m test points, directly applying KRR results in a computational complexity of $O(n^3m)$. However, we find that by using a computational trick for matrix inversion, we can reduce this computational complexity to $O(n^3 + n^2m)$. Based on this observation, we propose an efficient computational algorithm for the construction of full conformal prediction intervals for multiple test points. The numerical studies demonstrate the superior performance and practicality of the proposed algorithm.

There are several interesting avenues for future work. First, since the proposed approach works only for regression, extension to other learning tasks, such as classification, can be considered.

Second, although the computation with complexity $O(n^3)$ can be conducted only once, even for multiple test points, this might be prohibitive for large data sets. There are several approaches to reduce the computational complexity of computing the KRR estimate by approximating the kernel matrix, including the Nyström method [7] and the randomized sketch [18]. It would be interesting to study how such computationally fast approximation methods for KRR can be used for full conformal prediction and reduce the computational complexity.

Author contributions

Ilsang Ohn: Conceptualization, Formal analysis, Funding acquisition, Investigation, Methodology, Software, Supervision, Validation, Visualization, Writing-original draft, Writing–review and editing; Jisu Park: Formal analysis, Investigation, Methodology, Software, Writing-original draft. All authors have approved the final version of the manuscript for publication.

Use of Generative-AI tools declaration

The authors declare that they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work was supported by INHA UNIVERSITY Research Grant.

Conflict of interest

The authors declare no potential conflicts of interest.

References

1. R. F. Barber, E. J. Candes, A. Ramdas, R. J. Tibshirani, Predictive inference with the jackknife+, *Ann. Statist.*, **49** (2021), 486–507. http://doi.org/10.1214/20-AOS1965
2. E. Burnaev, V. Vovk, Efficiency of conformalized ridge regression, In: *Proceedings of The 27th Conference on Learning Theory*, **35** (2014), 605–622.
3. E. Burnaev, I. Nazarov, Conformalized kernel ridge regression, In: *2016 15th IEEE International Conference on Machine Learning and Applications, Anaheim, USA*, 2016, 45–52. https://doi.org/10.1109/ICMLA.2016.0017
4. K. Buza, Feedback prediction for blogs, In: *Data analysis, machine learning and knowledge discovery*, Berlin: Springer, 2013, 145–152. https://doi.org/10.1007/978-3-319-01595-8_16
5. M. M. Campos, A. Farinhas, C. Zerva, M. A. T. Figueiredo, A. F. T. Martins, Conformal prediction for natural language processing: a survey, *Trans. Assoc. Comput. Linguist.*, **12** (2024), 1497–1516. https://doi.org/10.1162/tacl_a_00715

6. T. J. Cleophas, A. H. Zwinderman, *Machine learning in medicine-a complete overview*, Berlin: Springer, 2015. https://doi.org/10.1007/978-3-319-15195-3
7. P. Drineas, M. W. Mahoney, N. Cristianini, On the Nyström method for approximating a gram matrix for improved Kernel-based learning, *J. Mach. Learn. Res.*, **6** (2005), 2153–2175.
8. F. Fiedler, S. Lucia, Improved uncertainty quantification for neural networks with Bayesian last layer, *IEEE Access*, **11** (2023), 123149–123160. https://doi.org/10.1109/ACCESS.2023.3329685
9. H. Gao, G. Kou, H. Liang, H. Zhang, X. Chao, C. C. Li, et al., Machine learning in business and finance: a literature review and research opportunities, *Financ. Innov.*, **10** (2024), 86. https://doi.org/10.1186/s40854-024-00629-z
10. J. Lei, M. G'Sell, A. Rinaldo, R. J. Tibshirani, L. Wasserman, Distribution-free predictive inference for regression, *J. Amer. Statist. Assoc.*, **113** (2018), 1094–1111. https://doi.org/10.1080/01621459.2017.1307116
11. L. E. Makili, J. A. Vega Sánchez, S. Dormido-Canto, Active learning using conformal predictors: application to image classification, *Fusion Sci. Technol.*, **62** (2012), 347–355. https://doi.org/10.13182/FST12-A14626
12. M. Redmond, A. Baveja, A data-driven software tool for enabling cooperative information sharing among police departments, *Eur. J. Oper. Res.*, **141** (2002), 660–678. https://doi.org/10.1016/S0377-2217(01)00264-8
13. G. Shafer, V. Vovk, A tutorial on conformal prediction, *J. Mach. Learn. Res.*, **9** (2008), 371–421.
14. M. H. Shaker, E. Hüllermeier, Ensemble-based uncertainty quantification: Bayesian versus credal inference, In: *Proceedings. 31. Workshop Computational Intelligence, Berlin*, 2021.
15. J. Vazquez, J. C. Facelli, Conformal prediction in clinical medical sciences, *J. Healthc. Inform. Res.*, **6** (2022), 241–252. https://doi.org/10.1007/s41666-021-00113-8
16. V. Vovk, A. Gammerman, G. Shafer, *Algorithmic learning in a random world*, New York: Springer, 2005.
17. T. Wuest, D. Weimer, C. Irgens, K. D. Thoben, Machine learning in manufacturing: advantages, challenges, and applications, *Prod. Manuf. Res.*, **4** (2016), 23–45. https://doi.org/10.1080/21693277.2016.1192517
18. Y. Yang, M. Pilanci, M. J. Wainwright, Randomized sketches for kernels: fast and optimal nonparametric regression, *Ann. Statist.*, **45** (2017), 991–1023. https://doi.org/10.1214/16-AOS1472

AIMS Press