A Conformal Predictive System for Distribution Regression with Random Features

Wei Zhang (448072640@qq.com)
Tianjin University https://orcid.org/0000-0002-6517-4373
Zhen He
Tianjin University
Di WANG
Tianjin University

Research Article

Keywords: Distribution regression, Conformal predictive system, Kernel mean embedding, Statistical postprocessing

Posted Date: December 3rd, 2021

DOI: https://doi.org/10.21203/rs.3.rs-331250/v1

A conformal predictive system for distribution regression with random features

Wei Zhang[1], Zhen He[1], Di Wang[2,*]

Abstract

Distribution regression is the regression case where the input objects are distributions. Many machine learning problems can be analysed in this framework, such as multi-instance learning and learning from noisy data. This paper attempts to build a conformal predictive system(CPS) for distribution regression, where the prediction of the system for a test input is a cumulative distribution function(CDF) of the corresponding test label. The CDF output by a CPS provides useful information about the test label, as it can estimate the probability of any event related to the label and be transformed to prediction interval and prediction point with the help of the corresponding quantiles. Furthermore, a CPS has the property of validity as the prediction CDFs and the prediction intervals are statistically compatible with the realizations. This property is desired for many risk-sensitive applications, such as weather forecast. To the best of our knowledge, this is the first work to extend the learning framework of CPS to distribution regression problems. We first embed the input distributions to a reproducing kernel Hilbert space using kernel mean embedding approximated by random Fourier features, and then build a fast CPS on the top of the embeddings. While inheriting the property of validity from the learning framework of CPS, our algorithm is simple, easy to implement and fast. The proposed approach is tested on synthetic data sets and can be used to tackle the problem of statistical postprocessing of ensemble forecasts, which demonstrates the effectiveness of our algorithm for distribution regression problems.

Keywords*:* Distribution regression; Conformal predictive system; Kernel mean embedding; Statistical postprocessing

1. Introduction

In the distribution regression cases, the input objects are represented by probability distributions or empirical probability distributions instead of feature vectors. One example is multi-instance learning whose input object is a bag of instances. The goal of distribution regression is to build a model from the input distributions to the related labels. Some distribution regression problems are risk-sensitive. Two representative examples are medical diagnosis where the patient's repeated measurements of the medical conditions are considered as the input object, and statistical postprocessing of ensemble forecasts whose input object is the forecasts of the same meteorological elements from multiple members of different numerical weather prediction models. For these risk-sensitive applications, predicting

*Corresponding author
Email address: *wangdi2015@tju.edu.cn* (Di Wang)

[1]College of Management and Economics, Tianjin University, 300072, Tianjin, China
[2]School of Electrical and Information Engineering, Tianjin University, 300072, Tianjin, China

the label of a test input is not enough since this bare point prediction has no useful information about the uncertainty and confidence of the prediction and the probability distribution of the label is preferred. Also, it is desired that the prediction distributions of the regression model have the property of validity (Gneiting & Katzfuss 2014; Vovk et al. 2019; Vovk 2019), which means that the predictions have statistical compatibility with the realizations(Vovk et al. 2018a).

Many algorithms developed from statistics and machine learning can output cumulative distribution functions(CDFs) for test labels. However, the representative algorithms such as the ones built on Gaussian process regression or Bayesian regression are sensitive to their prior distribution assumptions about the applications. If the prior assumptions are wrong, the CDFs output by them can be far away from validity which can not be trusted and used with confidence(Melluish et al. 2001; Balasubramanian et al. 2014). This issue can be tackled by the recently proposed pioneer works about conformal predictive systems(CPSs) (Vovk et al. 2019; Vovk 2019; Vovk et al. 2018a, b), which are proved to have the property of validity even in the small-sample case with the assumption not more than the data are independent and identically distributed(i.i.d.). With little restriction about the data, the CPSs are more useful and practical than Bayesian methods.

CPSs were first proposed by Vovk et al. (2019) to design systems which output valid CDFs for test labels. They are based on conformal prediction(Vovk et al. 2005; Balasubramanian et al. 2014), which can output valid prediction sets for labels and have been successfully applied to many applications where the prediction errors need to be controlled(Bosc et al. 2019; Cortés-Ciriano & Bender 2019; Nouretdinov et al. 2011; Papadopoulos et al. 2009; Laxhammar & Falkman 2011,2013). The p values calculated using conformity scores are the essential elements of constructing valid prediction sets for conformal prediction. With the i.i.d. assumption about the data, the p values are uniformly distributed on [0, 1]. This character is very promising since using this property we can transform the complex uncertainty from data to a very familiar distribution, which we can use to do many interesting things such as constructing valid prediction intervals. CPSs make full use of the p values of conformal prediction and relate them to the CDFs of the test labels. This is the reason why CPSs are valid even in the small-sample case. As CPSs are based on conformal prediction, the original approaches proposed in Vovk et al. (2019), Vovk (2019) and Vovk et al. (2018a) have computational issue inherited from conformal prediction, which severely limits the applicability of CPSs to real-time applications. This issue can be tackled in two ways. The first is to design more computationally efficient variants, which leads to the following works in Vovk et al. (2018b, 2020a) where split conformal predictive systems(SCPSs) and cross- conformal predictive systems(CCPSs) were proposed. The second is to use a fast and well-performed underlying regression algorithm, which motivates our recently proposed CPS combing Leave-One-Out CCPS with regularized extreme learning machine(RELM) (Huang et al. 2011) named as LOO-CCPS-RELM(Wang et al. 2020). The property of validity of LOO-CCPS-RELM were proved theoretically in the asymptotic setting and proved empirically by the experiments. Also, the comparison study with the other representative CPSs was conducted, which verified the effective- ness of LOO-CCPS-RELM. As such, we employ LOO-CCPS-RELM to build our CPS for distribution regression in this paper.

To handle input distributions, we follow the works Szabó et al. (2015) and Szabó et al. (2016) by embedding the input distributions to a reproducing kernel Hilbert space(RKHS) with kernel mean embedding. Different from mapping

feature vectors to points in RKHS with kernel method for pattern analysis(Shawe-Taylor & Cristianini 2004), kernel mean embedding maps input distributions to points in RKHS, each of which is a new representation of the related distribution. Using these new representations as inputs and the corresponding labels as outputs, a regression algorithm can be trained to establish a regression model. Thus, the CPS we develop for distribution regression in this paper comprises the following two steps. First, the input distributions are embedded to a RKHS by kernel mean embedding. Second, LOO-CCPS-RELM is trained from the embeddings to the labels. Moreover, to make the learning process of our CPS fast, we use random Fourier features(Rahimi & Recht 2007, 2008a, 2008b) to approximate the kernel of kernel mean embedding, which is inspired by the works in Jitkrittum et al. (2015) and Lopez-Paz et al. (2015).

The main contributions of this paper are two parts. First, to our knowledge, this is the first CPS which are built to tackle the distribution regression problems. Our approach is simple, easy to implement and applicable to real-time applications with the property of validity inherited from the learning framework of CPS, which is welcome by high-risk and real-time application areas. Second, the CPS is applied to statistical postprocessing of ensemble forecasts for temperature and precipitation, which is the first attempt of using a CPS to tackle the statistical postprocessing problems. Besides, the experimental results in this paper confirm the promising performance of our approaches for distribution regression both in the prediction CDFs and prediction intervals compared with other widely-used Bayesian methods.

The rest of this paper is organized as follows. Section 2 reviews kernel mean embedding of empirical distribution, regularized extreme learning machine and computationally efficient conformal predictive systems. Section 3 introduces our proposed algorithm. In Section 4, experiments are conducted, which includes the empirical study on synthetic data sets and applying our algorithm to statistical postprocessing of ensemble forecasts. The conclusions in this paper are drawn in Section 5.

2. Literature Review

We focus on the setting where the input distributions are empirical distribution , as it is more common to observe the samples than the probability measures itself in real applications(Póczos et al. 2013; Szabó et al. 2015). Thus, this section first reviews kernel mean embedding of empirical distribution and its approximation with random Fourier features, which was proposed in Jitkrittum et al. (2015) and Lopez-Paz et al. (2015). Then, we review SCPS and CCPS, which are computationally efficient versions of CPSs.

Some notations are needed throughout this paper. Let the training set be denoted by $z^l = \{(x_i, y_i), i = 1,\cdots\cdots, l\}$, where $x_i = \{\boldsymbol{x}_{i,j}, j = 1,\cdots\cdots, N_i\}$is the empirical distribution of $\boldsymbol{x}_i$, i.e., $\boldsymbol{x}_{i,j}$s are the samples of $\boldsymbol{x}_i$. y_i is the corresponding label. All samples are assumed to be taken from the sample space $\boldsymbol{X} \subset \boldsymbol{R^n}$ and the labels are real numbers. Here is an example of a medical application about this setting. x_i can be repeated blood tests of the ith patient and the sample $\boldsymbol{x}_{i,j}$ represents the n-dimensional feature vector from the jth blood test. y_i is a health indicator of the ith patient related to x_i. Another example is related to statistical postprocessing of ensemble forecast for precipitation. In this case, x_i is the forecasts of the precipitation from the N_i members of ensemble numerical models with y_i being the corresponding observed precipitation.

For a test input object x_0, the goal of a CPS is to construct a CDF on $y \in \boldsymbol{R}$, such that the CDF

describes the uncertainty of the corresponding label y_0.

2.1. Kernel mean embedding of empirical distribution and its approximation with random Fourier features

Suppose that $k: \boldsymbol{X} \times \boldsymbol{X} \to \boldsymbol{R}$ is a positive definite kernel and the RKHS with k is represented by H_k. Referring to Muandet et al. (2017), the kernel mean embedding $\hat{\mu}_i$ of the empirical distribution x_i to the space H_k can be formularized as

$$\mu_i = \frac{1}{N_i}\sum_{j=1}^{N_i} k(\boldsymbol{x}_{i,j}, \cdot). \tag{1}$$

If k is a characteristic kernel such as Gaussian kernel(Muandet et al. 2017), the embedding of formula (1) can capture all of the information of the distribution of x_i. That is why kernel mean embedding is a popular way to deal with distributional data. In general, the distribution regression based on formula (1) need to resort to dual optimization of learning with kernels. This is time consuming since the complexity of constructing the kernel matrices is at least $O(l^2)$ (Lopez-Paz et al. 2015). To address this issue, an alternative way is to approximate μ_i with a finite representation using random Fourier features(Rahimi & Recht 2007; Jitkrittum et al. 2015; Lopez-Paz et al. 2015). The approximation using random Fourier features explicitly maps $\boldsymbol{x}_i$ to a D-dimensional Euclidean space by a randomized feature map $\emptyset: \boldsymbol{X} \to \boldsymbol{R^D}$. This map satisfies that

$$k(\boldsymbol{x}_{i,p}, \boldsymbol{x}_{i,q}) \approx \phi(\boldsymbol{x}_{i,p})^{\mathrm{T}} \phi(\boldsymbol{x}_{i,q}).$$

Here is an example of Gaussian kernel, i.e.,

$$k(\boldsymbol{x}_{i,p}, \boldsymbol{x}_{i,q}) = exp(-\gamma \left\| \boldsymbol{x}_{i,p} - \boldsymbol{x}_{i,q} \right\|^2),$$

where γ is the kernel parameter. With Gaussian kernel, it follows that

$$\phi(\boldsymbol{x}_{i,p}) = \frac{1}{\sqrt{D}}\left[cos(\boldsymbol{\omega}_1^{\mathrm{T}} \boldsymbol{x}_{i,p} + b_1), \cdots, cos(\boldsymbol{\omega}_D^{\mathrm{T}} \boldsymbol{x}_{i,p} + b_{\mathrm{D}}) \right]^{\mathrm{T}} \tag{2}$$

Each component of $\boldsymbol{w_i}$ in the above formula is randomly drawn from the univariate gaussian distribution with mean being 0 and variance being 2γ, and b_i is drawn from the interval $[-\pi, \pi]$ uniformly. Therefore, the embedded mean μ_i in formula (1) can be approximated by $\hat{\mu}_i$ with the help of the randomized map $\boldsymbol{\emptyset}$, where $\hat{\mu}_i$ can be written as

$$\hat{\mu}_i = \frac{1}{N_i}\sum_{j=1}^{N_i} \phi(\boldsymbol{x}_{i,j}) \tag{3}$$

For shift-invariant kernels, the work in Lopez-Paz et al. (2015) analyzed the approximation error of $\hat{\mu}_i$. Throughout this paper, we use this approximation to design CPSs for distribution regression. Since the original distribution x_i is transformed to a D-dimensional feature vector with the approximation (3), it is easy to apply any regression algorithm including a CPS on the data set $\hat{z}^l = \{(\hat{\mu}_i, y_i), i = 1, \cdots, l\}$ for distribution regression problems. With this idea in mind, the CPSs for distribution regression are built on $\hat{z}^l$ in this paper. Thus, for the remaining parts of Section 2, we use the notation $\hat{\mu}_i$s as the input objects of CPSs.

2.2. Regularized extreme learning machine

RELM is a single-hidden-layer feedforward neural network with randomly chosen parameters of hidden nodes(Huang et al. 2011), which can be written as

$$f(\hat{\mu}) = \boldsymbol{h}(\hat{\mu})^{\mathrm{T}} \boldsymbol{\beta},$$

where L denotes the number of hidden nodes, $\hat{\mu} \in \boldsymbol{R}^D$ the input feature vector and $\beta \in \mathbf{R}^L$ the output weights learned from training data. Also,

$$h(\hat{\mu}) = \frac{1}{\sqrt{L}}\left[h(\hat{\mu};\theta_1),\cdots,h(\hat{\mu};\theta_L)\right]^{\mathrm{T}},$$

where $h(\hat{\mu};\theta_i)$ denotes the ith activation function whose parameters $\theta_i \in \mathbf{R}^{D+1}$.

With the training set the training set $\hat{z}^l$, a chosen activation function $h(\hat{\mu};\theta_i)$ and a fixed number L, the $l \times L$ matrix $\mathbf{H}$ of RELM is first calculated and can be written as

$$\mathbf{H} = [\boldsymbol{h}(\hat{\mu}_1)\boldsymbol{h}(\hat{\mu}_2)\cdots\boldsymbol{h}(\hat{\mu}_l)]^{\mathrm{T}},$$

where $\{\theta_1,\cdots,\theta_L\}$ are randomly drawn from a probability distribution on $\mathbf{R}^{D+1}$.

After that, RELM aims at minimizing the following optimization problem

$$\min_{\beta}\frac{1}{l}\|\mathbf{H}\boldsymbol{\beta}-\boldsymbol{y}\|^2 + \lambda\|\boldsymbol{\beta}\|^2,$$

where $\boldsymbol{y} = [y_1, y_2, \cdots, y_l]^T$ is a column vector and y_is are labels.

Then, the regressor output by the RELM algorithm can be expressed as

$$\hat{f}(\hat{\mu}) = \boldsymbol{h}(\hat{\mu})^{\mathrm{T}}(\mathbf{H}^{\mathrm{T}}\mathbf{H} + \lambda l\mathbf{I}_{L\times L})^{-1}\mathbf{H}^{\mathrm{T}}\boldsymbol{y}, \tag{4}$$

where $\mathbf{I}_{L\times L}$ is the $L \times L$ identity matrix.

RELM can learn fast and the leave-one-out predictions of RELM on the training set, $\hat{f}_i$s, can also be calculated fast by the following formula(Wang et al. 2018)

$$\hat{f}_i = \frac{\hat{f}(\hat{\mu}_i) - y_i \times hat_{ii}}{1 - hat_{ii}}, \tag{5}$$

where hat_{ii} is the entry in the ith element of the diagonal of $\mathbf{H}(\mathbf{H}^{\mathrm{T}}\mathbf{H} + \lambda l\mathbf{I}_{L\times L})^{-1}\mathbf{H}^{\mathrm{T}}$. This is the reason of using RELM as the underlying algorithm of LOO- CCPS- RELM.

2.3. Computationally efficient conformal predictive systems

As the original learning framework of CPS proposed in Vovk et al. (2019) has computational issue inherit from conformal prediction, two variants were proposed in Vovk et al. (2018b, 2020a) to speed up the learning process, which are SCPS and CCPS. As we also focus on fast CPSs in this paper, we only introduce SCPS and CCPS in this section.

2.3.1. Split conformal predictive system

Just like conformal prediction, every CPS needs a conformity measure $A(S,\hat{z})$ to calculate the conformity scores of observations. Here $A(S,\hat{z})$ is a function of a data set S and an observation $\hat{z}$ whose purpose is to measure the degree of agreement between the observations in S and the data $\hat{z}$. However, different from conformal prediction, the conformity measure of CPSs should satisfy the conditions more than being measurable. As stated in Vovk et al. (2018b), in the context of SCPSs, the conformity measure $A(S,\hat{z})$ must be a balanced isotonic function. Generally, with an underlying regression algorithm r, a balanced isotonic function $A(S,\hat{z})$ can be defined as

$$A(S,\hat{z}) = y - \hat{r}(\hat{\mu}), \tag{6}$$

which where $\hat{r}$ is the regression model learned from S using the regression algorithm r. This design for conformity measure was used in Vovk et al. (2018a, b) to build CPSs. Also, in this paper, we use it as the conformity measure to establish our CPSs.

Suppose the training $\hat{z}^l$ is split into two parts, which are the proper training set $\hat{z}_1^m = \{(\hat{\mu}_j, \mathrm{y_j}), \mathrm{j} = 1,2,\cdots,\mathrm{m}\}$ and the calibration set $\hat{z}_m^l = \{(\hat{\mu}_j, \mathrm{y_j}), \mathrm{j} = m+1,\cdots,\mathrm{l}\}$. For each possible label $y \in \boldsymbol{R}$, SCPS calculates $l - m + 1$ conformity scores as

$$\alpha_i = A\left(\hat{z}_1^m, (\hat{\mu}_i, \mathrm{y_i})\right), \tag{7}$$

$$\alpha_0^y = A\left(\hat{z}_1^m, (\hat{\mu}_0, \mathrm{y})\right), \tag{8}$$

where $i = m+1, m+2, \cdots, l$. Then the function

Q can be obtained as Vovk et al. (2018b)

$$Q_t(y) = \frac{1}{l-m+1}\left|\{i\in\{m+1,\cdots,l\}|\alpha_i < \alpha_0^y\}\right| + \frac{t}{l-m+1}\left|\{i\in\{m+1,\cdots,l\}|\alpha_i = \alpha_0^y\}\right| + \frac{t}{l-m+1}. \quad (9)$$

The theory in Vovk et al. (2018b) shows that the function $Q_t(y)$ above is a randomized predictive system, which theoretically has the property of validity. Referring to Vovk et al. (2018b, 2020b), if $A(S,\hat{z})$ is a strictly increasing continuous function of y, then C_i can be defined by the condition $A(\boldsymbol{z}_1^m, \boldsymbol{z}_{m+i}) = A(\boldsymbol{z}_1^m, (\boldsymbol{x}_0, C_i))$, where $i\in\{1,\cdots,l-m\}$. In the case where formula (6) is used as the conformity measure, we have

$$C_i = \hat{r}(\hat{\mu}_0) + y_{m+i} - \hat{r}(\hat{\mu}_{m+i}). \quad (10)$$

By sorting C_i in the increasing order we have $C_{(1)} \le \cdots \le C_{(l-m)}$. Let $C_{(0)} = -\infty$ and $C_{(l-m+1)} = \infty$, and then the function Q can be written as the following formula(Vovk et al. 2018b)

$$Q_t(y) = \begin{cases} \frac{i+t}{l-m+1} & \text{if } y\in(C_{(i)}, C_{(i+1)}) \text{ for } i\in\{0,\cdots,l-m\} \\ \frac{i'-1+(i''-i'+2)t}{l-m+1} & \text{if } y = C_{(i)} \text{ for } i\in\{1,\cdots,l-m\} \end{cases} \quad (11)$$

where $i' = min\{j|C_{(j)} = C_{(i)}\}$ and $i'' = max\{j|C_{(j)} = C_{(i)}\}$.

Formula (11) can not be used directly to represent the CDF of y_0 as it depends on t. To remove the impact of t, it is recommended in Vovk et al. (2020a) to use a modification of formula (11) in applications, such as

$$Q(y) = \begin{cases} \frac{i}{l-m} & \text{if } y\in(C_{(i)}, C_{(i+1)}) \text{ for } i\in\{0,\cdots,l-m\} \\ \frac{i}{l-m} & \text{if } y = C_{(i)} \text{ and } y\neq C_{(i+1)} \text{ for } i\in\{1,\cdots,l-m\} \end{cases} \quad (12)$$

That is, formula (12) is used as the CDF predicted for y_0 in this paper. Formula (12) is actually the empirical cumulative distribution function of $\{C_{(i)}, i = 1,\cdots,l-m\}$. From the approach above, we can see that SCPSs only use the calibration set to obtain $l-m$ conformity scores, which seems to be less informationally efficient than using the full training data. This is the very reason why CCPSs were also proposed.

2.3.2. Cross-conformal predictive system

The CCPS borrows the idea of cross validation and partitions the training set into k non-empty folds first. Let o_i denote the set of the ordinals in the i th fold and $\hat{\mu}_{(o_i)}^l$ denote the training data set leaving the ith fold out. Then, for each $i\in\{1,\cdots,k\}$, a SCPS with conformity measure $A(S,\hat{\mu})$ is used to obtain the conformity scores with $\boldsymbol{z}_{(o_i)}^l$ being the proper training set and $\{\boldsymbol{z}_j|j\in o_i\}$ being the calibration set. Thus, the related conformity scores can be written as

$$\alpha_{j,i} = A(\mathbf{z}_{(o_i)}^l, \mathbf{z}_j)$$

and

$$\alpha_{0,i}^y = A(\mathbf{z}_{(o_i)}^l, (\boldsymbol{x}_0, y))$$

where $j\in o_i$. Finally, the function $Q_t(y)$ of the CCPS is obtained as

$$Q_t(y) = \frac{1}{l+1}\sum_{i=1}^{k}\left|\left\{j\in o_i|\alpha_{j,i} < \alpha_{0,i}^y\right\}\right| + \frac{t}{l+1}\sum_{i=1}^{k}\left|\left\{j\in o_i|\alpha_{j,i} = \alpha_{0,i}^y\right\}\right| + \frac{t}{l+1}. \quad (13)$$

Just like SCPSs, if $A(S,\hat{z})$ is a strictly increasing continuous function of y, $C_{j.i}$ can be defined by the condition $A(\hat{z}_{(o_i)}^l, \hat{z}_j) = A\left(\hat{z}_{(o_i)}^l, (\hat{\mu}_0, C_{j,i})\right)$, where $i = \{1,\cdots,k\}$. Let $C_{(1)} \le \cdots \le C_{(l)}$ be all $C_{j,i}$ s sorted in the increasing order and $C_{(0)} = -\infty$ and $C_{(l+1)} = \infty$. Then the function $Q_t(y)$ of CCPS is determined by the following formula(Vovk et al. 2018b))

$$Q_t(y) = \begin{cases} \frac{i+t}{l+1} & \text{if } y\in(C_{(i)}, C_{(i+1)}) \text{ for } i\in\{0,\cdots,l\} \\ \frac{i'-1+(i''-i'+2)t}{l+1} & \text{if } y = C_{(i)} \text{ for } i\in\{1,\cdots,l\} \end{cases} \quad (14)$$

where $i' = min\{j|C_{(j)} = C_{(i)}\}$ and $i'' =$

$max\{j|C_{(j)} = C_{(i)}\}$.

To remove the impact of t in formula (14), it is recommended in Vovk et al. (2020a) to use a modification of formula (14) as follows,

$$Q(y) = \begin{cases} \frac{i}{l} & \text{if } y \in (C_{(i)}, C_{(i+1)}) \text{ for } i \in \{0, \cdots, l\} \\ \frac{i}{l} & \text{if } y = C_{(i)} \text{ and } y \neq C_{(i+1)} \text{ for } i \in \{1, \cdots, l\} \end{cases} \quad (15)$$

That is, formula (15) is used as the CDF predicted for y_0. Also, formula (15) represents the empirical cumulative distribution function of $\{C_{(i)}, i = 1, \cdots, l\}$. Then Leave-One-Out CCPS can be obtained by setting $k = l$.

2.4. LOO-CCPS-RELM

With $A(S, \hat{z})$ as formula (6) and underlying regression algorithm r as RELM, we only have Leave-One-Out CCPS with RELM, which has to compute RELM l times to calculated C_i s. A further step to obtain LOO-CCPS-RELM is to utilize a slight modification of Leave-One-Out CCPS with RELM by observing that RELM is a uniformly stable algorithm(Bousquet & Elisseeff 2002) whose output regressors are similar to each other before and after one training datum is removed. By this modification, RELM needs to be computed only once in LOO-CCPS-RELM and the details are summarized in Algorithm 1, where we see λl as a whole from formula (4).

Algorithm 1 LOO-CCPS-RELM

Input:

Training set $\hat{z}^l$, test object $\hat{z}_0$, meta-parameter L, regularization parameter λl for RELM.

Output:

CDF predicted for y_0.

1: Calculate C_i for $i = 1, 2, \cdots, l$ as

$$C_i = \hat{f}(\hat{\mu}_0) + y_i - \hat{f}_i,$$

where $\hat{f}_i$s are obtained by formula (5).

2: Return the function $Q(y)$, which is the empirical CDF of $\{C_i, i = 1, \cdots, l\}$.

The theoretical analysis in the asymptotic setting and empirical explorations of LOO-CCPS-RELM can be found from our previous work in Wang et al. (2020).

3. The proposed conformal predictive system for distribution regression

Recall that the training set is z^l and our purpose is to build a CPS to predict the prediction CDF for the test input x_0. After the review of Section 2, we can formulate our proposed CPS in the following two steps. First, the inputs x_is are mapped to $\hat{\mu}_i$s with approximated kernel mean embedding using formula (3). Then LOO-CCPS-RELM is applied to the embedded data set $\hat{z}^l$ to obtain function Q for predicting the CDF of x_0. We use random Fourier features to approximate the kernel of kernel mean embedding. Also, like the work in Wang et al. (2020), we use random Fourier features as $h(\hat{\mu}; \theta_i)$ s to build RELM. Thus, the whole learning process of our proposed CPS is very fast which can be applied to real- time applications. Algorithm 2 summarizes our proposed CPS for distribution regression with random Fourier features.

Algorithm 2 The proposed CPS for distribution regression with random Fourier features

Input:

Training data z^l, meta-parameters $D, L, \lambda l$.

Output:

CDF predicted for y_0.

1: Obtain the data set $\hat{z}^l$ and $\hat{\mu}_0$ using formula (3).

2: Return the function $Q(y)$, which is the

empirical CDF of $\{C_i, i = 1, \cdots, l\}$ calculated by Algorithm 1.

Here gives some analysis of the complexity of Algorithm 2. As the training process of Algorithm 2 contain training RELM on the data set $\hat{z}^l$ and calculating $y_i - \hat{f}_i$s with formula (5), for fixed meta-parameters D, L and λ, the complexity of the training process is $O(l)$, which is the same as that of LOO-CCPS-RELM(Wang et al. 2020). Hence, the training of Algorithm 2 can be fast even when l is very large. Also, for fixed meta-parameters, the testing process of Algorithm 2 is including embedding x_0 to $\hat{\mu}_0$ and calculating $\hat{f}(\hat{\mu}_0)$ as in Algorithm 1, whose computational complexity is only $O(1)$ since the times of multiplication computation is not dependent on l. Therefore, the computational complexity of Algorithm is low and can be applied to real-time applications properly.

As Algorithm 2 outputs a prediction CDF for the test input x_0, it is easy to derive a prediction interval from the CDF by the corresponding quantiles. For example, the prediction interval with expected coverage rate of $1 - \eta$ can be represented by

$$\left[q_{x_0}^{(\eta/2)}, q_{x_0}^{(1-\eta/2)}\right],$$

where $q_{x_0}^{(\eta/2)}$ and $q_{x_0}^{(1-\eta/2)}$ are $\eta/2$ and $1 - \eta/2$ quantiles of $Q_{x_0}(y)$ respectively. Thus, Algorithm 2 can also produce prediction intervals for the test inputs.

Although Algorithm 2 is very simple and easy to implement, it surprisingly has the property of validity inherited from CPSs and the error rate of the prediction intervals derived from the prediction CDFs can be controlled by the significance level η, which are verified empirically in the next section.

4. Experimental result and analysis

Let $\{(x_{0,i}, y_{0,i}), i = 1, \cdots, n\}$ be the test set. Recall that a CDF $Q(y)$ of any random variable Y has the property that for any $\alpha \in (0,1)$, there holds

$$P\{Q(Y) \leq \alpha\} = \alpha. \tag{16}$$

Thus, referring to Vovk et al. (2019) and Vovk(2019), Algorithm 2 having the property of validity means that the prediction CDFs are compatible with the realizations, which can be verified empirically by the fact that the frequency of the events $Q(y_{0,i}) \leq \alpha$ for $i \in \{1, \cdots, n\}$ is near α, i.e.,

$$\frac{|\{i \in \{1, \cdots, n\} | Q(y_{0,i}) \leq \alpha\}|}{n} \approx \alpha. \tag{17}$$

Also, another important property we care about is whether the error rate of the prediction intervals derived from the CDFs of Algorithm 2 can be controlled by the significance level η, which can be verified if the frequency of $y_{0,i}$s being out of the prediction interval $\left[q_{x_{0,i}}^{(\eta/2)}, q_{x_{0,i}}^{(1-\eta/2)}\right]$s is near or less than η.

In this section, we first explore Algorithm 2 using synthetic data sets to test whether Algorithm 2 has the property of validity in the sense of formula (16) and whether the error rate of the prediction intervals derived from Algorithm 2 can be controlled. After that, we build probabilistic forecast models using Algorithm 2 to forecast temperature and precipitation based on ensemble fore- casts of numerical models, and compare them with other widely-used Bayesian models.

For all of the following experiments, we set $D = 1000$ because of no significant improvements with more random features just like the empirical observation in Lopez-Paz et al. (2015). Following our previous work about LOO-CCPS-RELM, we set $L = 1000$ and select the meta-parameter λl from $\{10^{-5}, 10^{-4}, \cdots, 10^4, 10^5\}$ with the least of leave-

one-out square error in the training set. All of features and labels of the data sets in this section were first normalized to [0,1] with min-max normalization before they were fed to the algorithms. The experimental results were collected with ten-fold cross-validation, i.e., the algorithms were trained on nine folds and tested on the rest fold ten times to obtain the testing experimental results. Algorithm 2 was implemented with Python language(Van Rossum & Drake 1995) and its numpy library(Oliphant 2006).

4.1. Explorations on synthetic data sets

Synthetic data sets were generated to demonstrate the applicability of Algorithm 2. Referring to Póczos et al. (2013), each input distribution followed a $beta(a,3)$ distribution with 500 sample points, whose parameter a was uniformly chosen from the interval [3,20] and the corresponding label is the skewness of the distribution plus a noise variable ϵ, which can be written as

$$\frac{2(3-a)\sqrt{a+4}}{(a+3+2\sqrt{3a})}+\epsilon.$$

We collected six data sets in the above way, whose numbers of data were all set to 1000. The first data set is denoted by $beta_{0.00}$ with $\epsilon=0$ and the other five data sets are denoted by $beta_{0.04}$, $beta_{0.08}$, $beta_{0.12}$, $beta_{0.16}$ and $beta_{0.20}$, whose ϵs are uniformly drawn from $[-0.04,0.04]$, $[-0.08,0.08]$, $[-0.12,0.12]$, $[-0.16,0.16]$ and $[-0.20,0.20]$ respectively.

We first check whether Algorithm 2 has the property of validity using formula (17). Algorithm 2 was trained and tested on the six data sets individually and experimental results are shown in Figure 1, which records the relations between the frequency of $Q(y_{0,i})\leq\alpha$ and α. The curves are all closed to the diagonal, which means that Algorithm 2 is valid and is insensitive to the variance of the noise variable. Then, the error rates of the prediction intervals of Algorithm 2 were calculated on all of the data sets. Figure 2 shows the error rates can be controlled by ηs for all of the data sets, which demonstrates that the prediction intervals output by Algorithm 2 are reliable and are robust to the variance of the noise variable. Figure 3 shows the average interval sizes of Algorithm for different ηs and different data sets. It can be seen that the average interval size is related to η and the variance of the noise variable. As the prediction error rates are robust to the variance of the noise variable, a larger variance of the noise variable leads to a larger average interval size. Also, as the error rate can be controlled by η, a larger η leads to a smaller average interval size. As η increases, the average interval size is forced to become short to increase the error rate, which shows the strict control over the error rate by tuning η. Thus, from Figure 2 and Figure 3 we can see that there is a balance between the error rate and the informational efficiency of the prediction intervals. In high-risk applications, as the informational efficiency must be sacrificed to make sure that the error rate of the interval predictions is low enough, one can choose η from $[0.05,0.2]$ to balance the error rate and the informational efficiency.

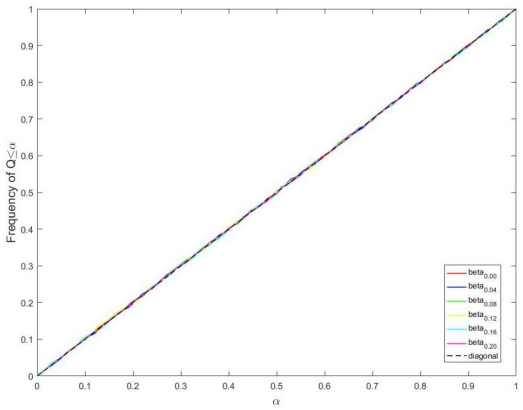

Figure 1: Tests the validity of Algorithm 2 on six data sets. The curves are all closed to the diagonal, which indicates that Algorithm 2 is valid in the sense of formula (16) on all six

data sets.

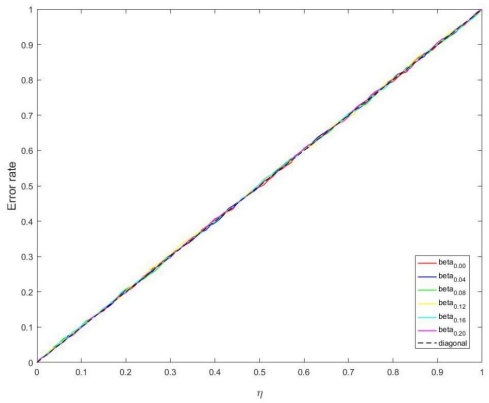

Figure 2: The relation curves of ηs and the error rates of the prediction intervals derived from the CDFs output by Algorithm 2. All curves are closed to the diagonal, which demonstrates that the error rates can be controlled for all six data sets.

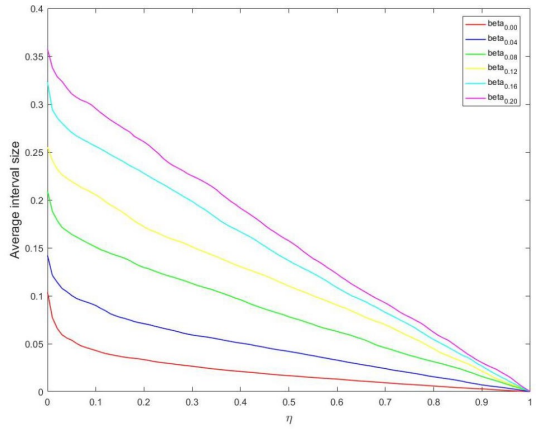

Figure 3: The relation curves of ηs and the average interval sizes. Large variance of labels leads to large average interval size, as the error rates are strictly controlled by ηs.

4.2. The proposed algorithm for statistical postprocessing of ensemble forecasts

Ensemble forecasts from numerical models for probabilistic weather forecasting are not valid since they tend to be biased and under dispersed. This inspires many researches on statistical postprocessing of ensemble forecasts. The task is to build a regression model, whose inputs are ensemble forecasts of some meteorological element and labels are the corresponding observations. This is a typical distribution regression problem since the inputs can be seen as the estimated empirical distribution of the meteorological element.

In this section, we apply Algorithm 2 to the task of postprocessing of ensemble forecasts for temperature and precipitation. The two data sets are all from ensemblepp package(Messner 2017) of R language(R Core Team 2018). For the task of postprocessing of ensemble forecasts for temperature, we use the temp data set, which has 18−30 hour minimum temperature ensemble forecasts and observations at Innsbruck. For precipitation, we use the rain data set, which has accumulated 18−30 hour precipitation ensemble forecasts and observations which are also at Innsbruck. All the two data sets have 2749 samples and are from 2000-01-02 to 2016-01-01. As the distribution of precipitation is highly biased, it is common in the literature to transform precipitation data before applying postprocessing methods. Referring to Vannitsem et al. (2018), we use the square root to transform all forecasts and observations of precipitation data prior to the process of min-max normalization. As the data were collected in chronological order, they were partitioned with ten sequential parts for ten-fold cross-validation experiments.

Two kinds of widely-used postprocessing algorithms are compared with Algorithm 2. The first kind is Bayesian Model Averaging(BMA)(Raftery et al. 2005) and the second kind is Ensemble Model Output Statistics(EMOS)(Gneiting & Raftery 2005). For temperature, as the assumed distribution of temperature is normal distribution, the corresponding algorithms are named as BMA-n(Raftery et al. 2005) and EMOS-n(Gneiting & Raftery 2005). For precipitation, BMA-g(Sloughter et al. 2007), EMOS-csg(Scheuerer &

Hamill 2015) and EMOS-gev(Scheuere 2014) are used as comparative algorithms, where the assumed distributions of precipitation are gamma distribution, censored and shifted gamma distribution, and censored generalized extreme value distribution. The packages ensembleBMA(Fraley et al. 2018) and ensembleMOS(Yuen et al. 2018) of R language were used to build BMA and EMOS based algorithms respectively.

For temperature, BMA-n, EMOS-n and Algorithm 2 were tested on the temp data set. First, the test of whether the algorithms are valid in the sense of formula (16) was conducted. We changed α from 0 to 1 to see if the frequency of $Q(y_{0,i}) \leq \alpha$ is near α. Figure 4 shows the curves obtained by BMA-n, EMOS-n and Algorithm 2. It can be seen that Algorithm 2 is valid as the curve is very closed to the diagonal, while the other two algorithms are not valid at all. This is reasonable since Algorithm 2 inherits the property of validity from CPSs. Second, the error rates of the prediction intervals output by the three algorithms were calculated. We also changed η from 0 to 1 to see whether the error rates are controlled by ηs, i.e., the frequency of the real labels being out of the prediction intervals is near or under η for each $\eta \in (0,1)$. Figure 5 shows the experimental results of error rates of prediction intervals. From Figure 5, we can see that the error rates of prediction intervals of Algorithm 2 are controlled by ηs while those of the other algorithms are not controlled. Thus, Figure 4 and Figure 5 confirm that Algorithm 2 is compatible with the realizations while BMA-n, EMOS-n are not. Also, the average interval sizes were obtained to see if the intervals predicted by Algorithm 2 are informationally efficient. Figure 6 shows how the average interval sizes changed when η varied from 0 to 1 for different algorithms. It is shown that the average interval size of EMOS-n is the lowest for any η. For $\eta \in [0.1,1]$, the average interval size of Algorithm 2 is lower than that of BMA-n while for $\eta \in [0,1)$, the average interval size of Algorithm 2 is nearly the same as that of BMA-n. Since we focus on high-risk applications, we care more about the performance for $\eta \in [0.05,0.2]$. Thus, as the error rates of the prediction intervals of EMOS-n can not be controlled for $\eta \in [0.05,0.2]$, the prediction intervals of EMOS-n for $\eta \in [0.05,0.2]$ are meaningless. Then, we can conclude that the prediction intervals of Algorithm 2 are informationally efficient since the average interval size compares favourably with BMA-n for $\eta \in [0.05,0.2]$. We summarize the experimental results in Table 1 to make them more concise. From Table 1, we can conclude that the CDF output by Algorithm 2 is compatible with the realizations as it is valid in the sense of formula (16) and the error rates of its prediction intervals can be controlled. Besides, the prediction intervals of Algorithm 2 are informationally efficient.

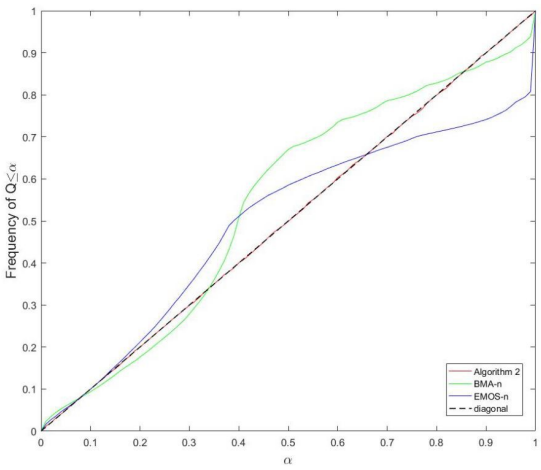

Figure 4: Tests whether Algorithm 2, BMA-n and EMOS-n are valid for probabilistic fore- casting temperature. Algorithm 2 is closed to the diagonal while the other two algorithms are not, means that Algorithm 2 is valid and BMA-n and EMOS-n are not valid in the sense of formula (16).

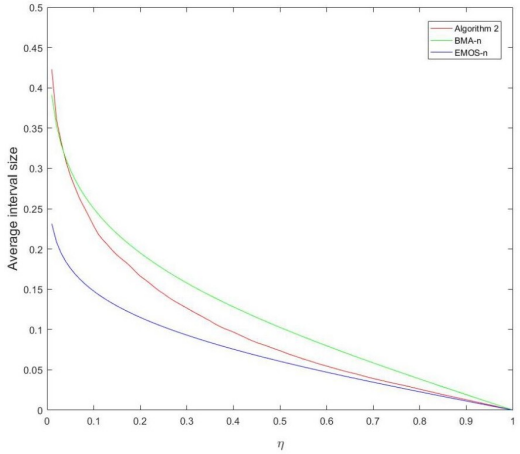

Figure 5: The relation curves of ηs and the error rates of the prediction intervals derived from the CDFs output by Algorithm 2, BMA-n and EMOS-n. The prediction intervals of Algorithm 2 is reliable for all $\eta \in (0,1)$ since its curve is near or below the diagonal. The prediction intervals of BMA-n and EMOS-n are not reliable for some $\eta \in (0,1)$.

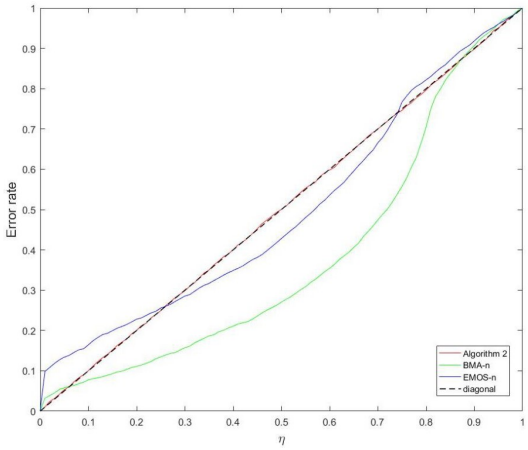

Figure 6: The relation curves of ηs and the average interval sizes of Algorithm 2, BMA-n and EMOS-n. On the condition that the prediction intervals are reliable, Algorithm 2 are informationally efficient for $\eta \in [0.05,0.2]$.

Table 1: Results of postprocessing of ensemble forecasts for temperature

	Frequency of $Q \leq \alpha$			Error rate			Average interval size		
	α=0.9	α=0.5	α=0.1	η=0.2	η=0.1	η=0.05	η=0.2	η=0.1	η=0.05
Algorithm 2	0.899	0.499	0.099	0.200	0.100	0.053	0.166	0.228	0.290
BMA-n	0.877	0.670	0.093	0.110	0.077	0.057	0.194	0.250	0.297
EMOS-n	0.741	0.585	0.099	0.228	0.164	0.133	0.115	0.147	0.175

For precipitation, similar experiments were conducted with the rain data set to compare Algorithm 2 with BMA-g, EMOS-csg and EMOS-gev. As precipitation is always nonnegative, we force the negative samples in the empirical distribution predicted by Algorithm 2 to be 0. The meanings of Figure 7, Figure 8 and Figure 9 are similar with those of Figure 4, Figure 5 and Figure 6 respectively. From Figure 7, it can be seen that Algorithm 2 is valid while the other three algorithms are not. Figure 8 shows that the error rates of the prediction intervals of all four algorithms can be controlled by η s, which indicates that the prediction intervals of all four algorithms are reliable. At last, Figure 9 demonstrates that the average interval sizes of Algorithm 2 are lower than those of BMA-g and EMOS-gev for $\eta \in (0,1)$ and are similar to those of EMOS-csg for $\eta \in (0,0.2)$, which confirms that the prediction intervals output by Algorithm 2 are informationally efficient for forecasting precipitation. The experimental results are summarized in Table 2, where we can conclude that the CDF output by Algorithm 2 is compatible with the realizations and the prediction intervals of Algorithm 2 are informationally efficient.

In summary, we can conclude that Algorithm 2 is valid in the sense of formula (16) for ensemble forests for temperature and precipitation while the other comparative algorithms are not. Besides, the prediction intervals of Algorithm 2 are reliable and

informationally efficient, which verifies the effectiveness of Algorithm 2.

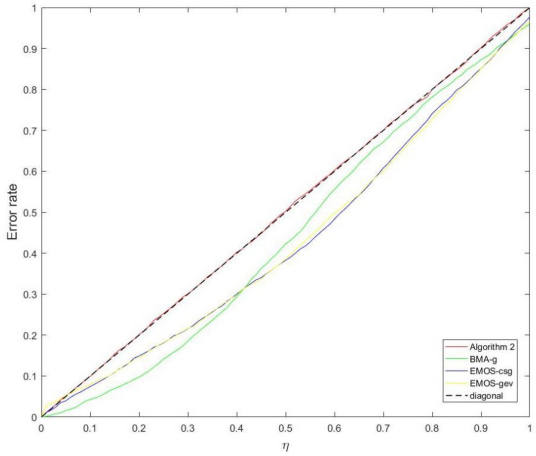

Figure 7: Tests whether Algorithm 2, BMA-g, EMOS-csg and EMOS-gev are valid for probabilistic forecasting precipitation. Algorithm 2 is closed to the diagonal while the other three algorithms are not, means that Algorithm 2 is valid and EMA-g, EMOS-csg and EMOS-gev are not valid in the sense of formula (16).

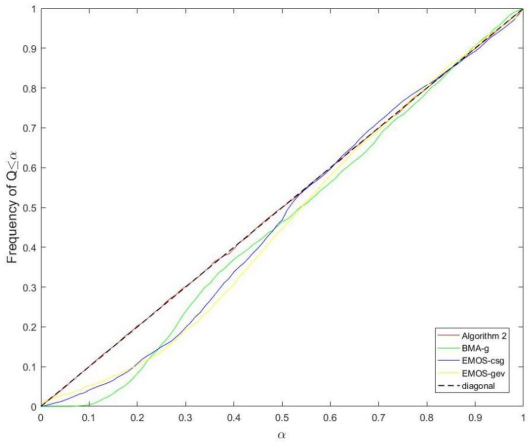

Figure 8: The relation curves of ηs and the error rates of the prediction intervals derived from the CDFs output by Algorithm 2, BMA-g, EMOS-csg and EMOS-gev. The prediction intervals of all algorithms are reliable since their curves are all near or below the diagonal.

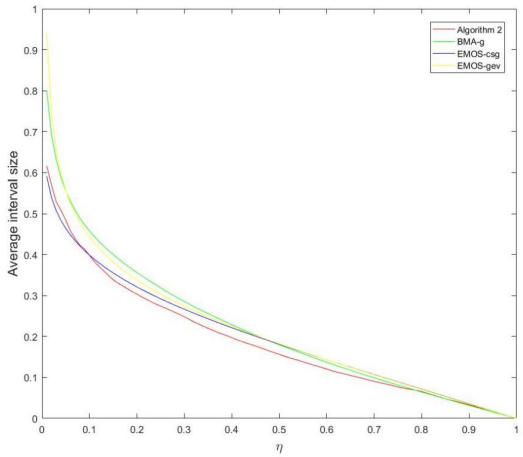

Figure 9: The relation curves of ηs and the average interval sizes of Algorithm 2, BMA-g, EMOS-csg and EMOS-gev. On the condition that the prediction intervals are reliable, Algorithm 2 and EMOS-csg are both informationally efficient for $\eta \in [0.05,0.2]$.

Table 2: Results of postprocessing of ensemble forecasts for precipitation

	Frequency of Q≤α			Error rate			Average interval size		
	α=0.9	α=0.5	α=0.1	η=0.2	η=0.1	η=0.05	η=0.2	η=0.1	η=0.05
Algorithm 2	0.898	0.499	0.100	0.201	0.099	0.052	0.303	0.398	0.484
BMA-g	0.906	0.464	0.004	0.097	0.042	0.016	0.356	0.457	0.555
EMOS-csg	0.892	0.468	0.041	0.149	0.074	0.039	0.321	0.398	0.463
EMOS-gev	0.906	0.445	0.052	0.145	0.078	0.052	0.337	0.443	0.556

5. Conclusion

This paper builds a conformal predictive system for distribution regression. A two-stage process is employed where the input distribution is first embedded to an approximated reproducing kernel Hilbert space and then the LOO-CCPS-RELM is used to build the conformal predictive system. The experiments confirm the validity of the prediction CDFs and the reliability of the prediction intervals of the proposed algorithm. Comparisons with other widely-used Bayesian methods for postprocessing of ensemble forecasts verify the effectiveness of our algorithm for probabilistic forecasts for temperature and precipitation. Our approach is easy to implement, fast and valid, which is promising in real-time and high-risk applications related to distribution regression.

Acknowledgement This work was supported by the National Natural Science Foundation of China under Grant 61972282 and 71661147003.The authors would like to thank the anonymous editor and reviewers for their valuable comments and suggestions which improved this work.

Author contributions All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by Wei Zhang and Di Wang. All authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Compliance with ethical standards

Conflict of interest Wei Zhang, Zhen He and Di Wang declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

Balasubramanian, V., Ho, S.-S., & Vovk, V. (2014). Conformal Prediction for Reliable Machine Learning: Theory, Adaptations and Applications. Newnes.

Bosc, N., Atkinson, F., Felix, E., Gaulton, A., Hersey, A., & Leach, A. R. (2019). Large scale comparison of QSAR and conformal prediction methods and their applications in drug discovery. J Cheminform, 11(1), 4. doi:10.1186/s13321-018-0325-4

Bourouis, S., Al-Osaimi, F. R., Bouguila, N., Sallay, H., Aldosari, F., & Al Mashrgy, M. (2019). Bayesian inference by reversible jump MCMC for clustering based on finite generalized inverted Dirichlet mixtures. Soft Computing, 23(14), 5799-5813. doi:10.1007/s00500-018-3244-4

Bousquet, O., & Elisseeff, A. (2002). Stability and generalization. journal of machine learning research, 2(3), 499-526. doi:Doi 10.1162/153244302760200704

Cortés-Ciriano,, I., & Bender, A. (2019). Reliable Prediction Errors for Deep Neural Networks Using Test-Time Dropout. Journal of Chemical Information and Modeling, 59(7), 3330-3339. doi:10.1021/acs.jcim.9b00297

Fraley, C., Raftery, A. E., Gneiting, T., & Sloughter, J. M. (2018). ensembleBMA: An R Package for Probabilistic Forecasting using Ensembles and Bayesian Model Averaging, r package version 5.1.5.[Available online at https://CRAN.R-project.org/package=ensembleBMA.].

Gneiting, T., & Katzfuss, M. (2014). Probabilistic Forecasting. Annual Review of Statistics and Its Application, Vol 1, 1, 125-151. doi:10.1146/annurev-statistics-062713-085831

Gneiting, T., & Raftery, A. E. (2005). Atmospheric science. Weather forecasting with ensemble methods. science, 310(5746), 248-249. doi:10.1126/science.1115255

Huang, G. B., Wang, D. H., & Lan, Y. (2011). Extreme learning machines: a survey. International Journal of Machine Learning and Cybernetics, 2(2), 107-122. doi:10.1007/s13042-011-0019-y

Jitkrittum, W., Gretton, A., Heess, N., Eslami, S. M. A., Lakshminarayanan, B., Sejdinovic, D., & Szabó, Z. (2015). Kernel-based Just-In-Time learning for passing expectation propagation messages. In UAI'15 Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence (pp. 405–414).

Laxhammar, R., & Falkman, G. (2011, July). Sequential conformal anomaly detection in trajectories based on hausdorff distance. In 14th International Conference on Information Fusion (pp. 1-8). IEEE.

Laxhammar, R., & Falkman, G. (2013). Online Learning and Sequential Anomaly Detection in Trajectories. IEEE Trans Pattern Anal Mach Intell, 36(6), 1158-1173. doi:10.1109/TPAMI.2013.172

Lopez-Paz, D., Muandet, K., lkopf, B. S., & Tolstikhin, I. (2015). Towards a Learning Theory of Cause-Effect Inference. In Proceedings of The 32nd International Conference on Machine Learning (pp. 1452–1461).

Melluish, T., Saunders, C., Nouretdinov, I., & Vovk, V. (2001, September). Comparing the Bayes and typicalness frameworks. In European Conference on Machine Learning (pp. 360-371). Springer, Berlin, Heidelberg.

Messner, J. (2017). Ensemble Postprocessing Data Sets. R package ensemblepp version 0.1-0.[Available online at https://CRAN.R-project.org/package=ensemblepp.]

Muandet, K., Fukumizu, K., Sriperumbudur, B., & Scholkopf, B. (2017). Kernel Mean Embedding of Distributions: A Review and Beyond. Foundations and Trends in Machine Learning, 10(1-2), 1-+. doi:10.1561/2200000060

Nouretdinov, I., Costafreda, S. G., Gammerman, A., Chervonenkis, A., Vovk, V., Vapnik, V., & Fu, C. H. (2011). Machine learning classification with confidence: application of transductive conformal predictors to MRI-based diagnostic and prognostic markers in depression. Neuroimage, 56(2), 809-813. doi:10.1016/j.neuroimage.2010.05.023

Oliphant, T. E. (2006). A guide to NumPy (Vol. 1): Trelgol Publishing USA.

Papadopoulos, H., Gammerman, A., & Vovk, V. (2009, April). Confidence predictions for the diagnosis of acute abdominal pain. In IFIP International Conference on Artificial Intelligence Applications and Innovations (pp. 175-184). Springer, Boston, MA.

Póczos, B., Singh, A., Rinaldo, A., & Wasserman, L. A. (2013, April). Distribution-Free Distribution Regression. In Artificial Intelligence and Statistics (pp. 507–515).

R Core Team. (2018). R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria. [Available online at https://www.R-project.org/]

Raftery, A. E., Gneiting, T., Balabdaoui, F., & Polakowski, M. (2005). Using Bayesian model averaging to calibrate forecast ensembles. Monthly Weather Review, 133(5), 1155-1174. doi:Doi 10.1175/Mwr2906.1

Rahimi, A., & Recht, B. (2007). Random Features for Large-Scale Kernel Machines. In Advances in Neural Information Processing Systems 20 (Vol. 20, pp. 1177–1184).

Rahimi, A., & Recht, B. (2008a). Uniform approximation of functions with random bases. In 2008 46th Annual Allerton Conference on Communication, Control, and Computing (pp. 555-561). IEEE.

Rahimi, A., & Recht, B. (2008b). Weighted Sums of Random Kitchen Sinks: Replacing

minimization with randomization in learning. In Advances in Neural Information Processing Systems 21 (Vol. 21, pp. 1313–1320).

Ren, W. J., Wang, Y. W., & Han, M. (2021). Time series prediction based on echo state network tuned by divided adaptive multi-objective differential evolution algorithm. Soft Computing. doi:10.1007/s00500-020-05457-8

Scheuerer, M. (2014). Probabilistic quantitative precipitation forecasting using Ensemble Model Output Statistics. quarterly journal of the royal meteorological society, 140(680), 1086-1096. doi:10.1002/qj.2183

Scheuerer, M., & Hamill, T. M. (2015). Statistical Postprocessing of Ensemble Precipitation Forecasts by Fitting Censored, Shifted Gamma Distributions. Monthly Weather Review, 143(11), 4578-4596. doi:10.1175/Mwr-D-15-0061.1

Shawe-Taylor, J., & Cristianini, N. (2004). Kernel methods for pattern analysis: Cambridge university press.

Sloughter, J. M., Raftery, A. E., Gneiting, T., & Fraley, C. (2007). Probabilistic quantitative precipitation forecasting using Bayesian model averaging. Monthly Weather Review, 135(9), 3209-3220. doi:10.1175/Mwr3441.1

Szabó, Z., Gretton, A., Póczos, B., & Sriperumbudur, B. (2015, February). Two-stage sampled learning theory on distributions. In Artificial Intelligence and Statistics (pp. 948-957).

Szabó,, Z., Sriperumbudur, B. K., Poczos, B., & Gretton, A. (2016). Learning Theory for Distribution Regression. journal of machine learning research, 17(1), 5272-5311.

Van Rossum, G., & Drake Jr, F. L. (1995). Python tutorial (Vol. 620): Centrum voor Wiskunde en Informatica Amsterdam.

Vannitsem, S., Wilks, D. S., & Messner, J. (2018). Statistical postprocessing of ensemble forecasts: Elsevier.

Vovk, V. (2019). Universally consistent conformal predictive distributions. In Conformal and Probabilistic Prediction and Applications (pp. 105-122).

Vovk, V., Gammerman, A., & Shafer, G. (2005). Algorithmic learning in a random world. Springer Science & Business Media.

Vovk, V., Nouretdinov, I., Manokhin, V., & Gammerman, A. (2018a). Conformal Predictive Distributions with Kernels. Braverman Readings in Machine Learning: Key Ideas from Inception To Current State, 11100, 103-121. doi:10.1007/978-3-319-99492-5_4

Vovk, V., Nouretdinov, I., Manokhin, V., & Gammerman, A. (2018b). Cross-conformal predictive distributions. In Conformal and Probabilistic Prediction and Applications (pp. 37-51).

Vovk, V., Petej, I., Nouretdinov, I., Manokhin, V., & Gammerman, A. (2020). Computationally efficient versions of conformal predictive distributions. Neurocomputing, 397, 292-308. doi:10.1016/j.neucom.2019.10.110

Vovk, V., Petej, I., Toccaceli, P., Gammerman, A., Ahlberg, E., & Carlsson, L. (2020b). Conformal calibrators. In Conformal and Probabilistic Prediction and Applications (pp. 84-99).

Vovk, V., Shen, J. L., Manokhin, V., & Xie, M. G. (2019). Nonparametric predictive distributions based on conformal prediction. Machine Learning, 108(3), 445-474. doi:10.1007/s10994-018-5755-8

Wang, D., Wang, P., & Shi, J. Z. (2018). A fast and efficient conformal regressor with regularized extreme learning machine. Neurocomputing, 304, 1-11. doi:10.1016/j.neucom.2018.04.012

Wang, D., Wang, P., Yuan, Y., Wang, P., & Shi, J. (2020). A fast conformal predictive system with regularized extreme learning machine.

Neural Netw, 126, 347-361. doi:10.1016/j.neunet.2020.03.022

Wang, J. S., Han, S., & Guo, Q. P. (2014). Echo state networks based predictive model of vinyl chloride monomer convention velocity optimized by artificial fish swarm algorithm. Soft Computing, 18(3), 457-468. doi:10.1007/s00500-013-1068-9

Yan, D., Chu, Y., Zhang, H., & Liu, D. (2018). Information discriminative extreme learning machine. Soft Computing, 22(2), 677-689. doi:10.1007/S00500-016-2372-Y

Yuen, R. A., Baran, S., Fraley, C., Gneiting, T., Lerch, S., Scheuerer, M., & Thorarinsdottir, T. L. (2018). ensembleMOS: Ensemble model output statistics. R package version 0.8.2.[Available online at http://CRAN. R-project. org/package= ensembleMOS.].

Zhai, J. H., Xu, H. Y., & Wang, X. Z. (2012). Dynamic ensemble extreme learning machine based on sample entropy. Soft Computing, 16(9), 1493-1502. doi:10.1007/s00500-012-0824-6