Conformalized and Fast Bayesian Nonparametric Ensemble

Aditya Makkar, Jeremiah Liu, Sebastian Rowland, Lawrence Chillrud,
Marianthi-Anna Kioumourtzoglou, Brent Coull, John Paisley

Abstract

Bayesian nonparametrics provide a theoretically well-founded framework to construct machine learning models by convenient incorporation of modeling assumptions. In this project we develop a Bayesian Nonparametric Ensemble (BNE) that not only provides a flexible way to weight different underlying models in an ensemble according to the strengths of each model on different regions of input space, but also allows for quantification of the ensemble's uncertainty. Our model is motivated by the problem of $\mathrm{PM}_{2.5}$ prediction across the contiguous USA. We solve two obstacles in our model–first, the size of the dataset necessitates the development of a fast approximate inference for our BNE model, which we do using random Fourier features and Laplace approximation, and second, to close the unavoidable gap between reality and our modeling assumptions we use conformal prediction to get accurate coverage estimation. Finally, we demonstrate the practicality and usefulness of this model by doing thorough experiments on the $\mathrm{PM}_{2.5}$ dataset.

1 Introduction

Ensemble methods are a well-developed framework of machine learning and statistics [Dietterich, 2000, Claeskens and Hjort, 2008, Zhou, 2012], with the winning solution of the Netflix Prize challenge being a notable example [Bell and Koren, 2007]. In this approach, multiple predictions of the same event made by predetermined models are ensembled in a way that improves the overall prediction. Let $\mathcal{D} = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$ be a training set of n data pairs drawn i.i.d. from some unknown distribution $\mathbb{P}$ on the set $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X} \subseteq \mathbb{R}^d$ and $\mathcal{Y} \subseteq \mathbb{R}$. In this paper, we focus on the model ensembling problem for regression with spatio-temporally evolving data, i.e., $\mathcal{X}$ will be $\mathbb{R}^3$ with a typical point being $(\text{latitude}, \text{longitude}, \text{time})$. The problem of designing an ensemble method can be broken into two primary tasks: 1) create a collection $\mathcal{S} = \{S_1, \dots, S_L\}$ of individual models such that each S_ℓ is a mapping from $\mathcal{X}$ to $\mathcal{Y}$, possibly using data other than $\mathcal{D}$; and 2) develop a combination rule $\widehat{S} = f(S_1, \dots, S_L) \colon \mathcal{X} \to \mathcal{Y}$. In this paper, we focus on the second task and assume the models $\mathcal{S}$ are already known.

The most basic ensemble is of the form

$$\widehat{S} = \textstyle\sum_{\ell=1}^{L} c_\ell S_\ell,$$

where the weights $c_1, \dots, c_L$ are nonnegative and sum to 1. Traditional ensemble methods work with fixed weights $c_1, \dots, c_L$. However, it is often desirable that the weights be *adaptive*, i.e., are functions mapping $\mathcal{X}$ to $[0, 1]$ with the constraint that at every point in $\mathcal{X}$ sums to 1. This allows the ensemble $\widehat{S}$ to weight models flexibly depending on which regions of the input space $\mathcal{X}$ they perform better. It is also desirable to provide uncertainty estimates for the ensemble prediction [Liu et al., 2019].

As a concrete example that motivates our proposed model, we consider the problem of predicting $\mathrm{PM}_{2.5}$ concentration in the contiguous United States. Several existing models exist for this problem developed by research groups in the environmental sciences. These models employ different inputs (e.g., remote sensing, chemical transport models, land use variables) and different algorithms (e.g., neural networks, random forests, generalized additive models). They make fine scale predictions across all locations within the USA. Meanwhile, throughout the USA there also exist about 1000 monitors that regularly collect reading of $\mathrm{PM}_{2.5}$. Using such data, an ensemble naturally would allow us to combine the strengths of these models to obtain better predictions than each of the individual models. Furthermore, a spatio-temporally adaptive ensemble would add flexibility to the model averaging. Such a framework would provide a means for employing expert knowledge contained in the carefully-crafted models while improving predictions with interpretable spatio-temporal model averaging. We also emphasize that, for many problems, such as $\mathrm{PM}_{2.5}$ prediction, the models $\mathcal{S}$ are developed by experts using painstaking years of research, and therefore it is essential in such fields to create an ensemble of existing models in $\mathcal{S}$ using training data $\mathcal{D}$ rather than create a new model from scratch using only the training data $\mathcal{D}$. In this paper, we propose a method that uses set of spatio-temporally adaptive Gaussian processes, which is made scalable using the random Fourier feature approach.

Finally, how do we know if our model correctly represents the reality? More precisely, suppose, as before, we have the training data $\mathcal{D} = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$ drawn i.i.d. from $\mathbb{P}$, using which we fit a Bayesian model $\widehat{f} \colon \mathcal{X} \to \mathcal{P}(\mathcal{Y})$, where $\mathcal{P}(\mathcal{Y})$ denotes the set of probability measures on $\mathcal{Y}$, i.e., at each point $x \in \mathcal{X}$, we get a posterior distribution $\widehat{f}(x)$ on $\mathcal{Y}$. Since we are in Euclidean spaces, we can disintegrate (see [Çınlar, 2011] Theorem 2.18) $\mathbb{P}$ as follows

$$\mathbb{P}(\mathrm{d}x, \mathrm{d}y) = \mathbb{P}_{\mathcal{X}}(\mathrm{d}x)\mathbb{P}_{\mathcal{Y}|\mathcal{X}}(x, \mathrm{d}y),$$

where $\mathbb{P}_{\mathcal{X}}$ denotes the marginal distribution of $\mathbb{P}$ on $\mathcal{X}$ and $\mathbb{P}_{\mathcal{Y}|\mathcal{X}}$ denotes the transition probability kernel from $\mathcal{X}$ to $\mathcal{Y}$, which at each point $x \in \mathcal{X}$ gives a probability measure $\mathbb{P}_{\mathcal{Y}|\mathcal{X}}(x, \cdot)$ on $\mathcal{Y}$. Now suppose we draw a new point (X_{n+1}, Y_{n+1}) from $\mathbb{P}$ independently of $\mathcal{D}$. If our model $\widehat{f}$ is indeed correct, then we would expect the distribution $\widehat{f}(X_{n+1})$ to equal the distribution $\mathbb{P}_{\mathcal{Y}|\mathcal{X}}(X_{n+1}, \cdot)$. This, however, is too strong a requirement and instead we would like a more relaxed requirement: if $C_\alpha(X_{n+1}) \subseteq \mathcal{Y}$ is a set with $\widehat{f}(X_{n+1})-$measure at least $1 - \alpha$, then we would also like to have valid coverage

$$\mathbb{P}\left\{Y_{n+1} \in C_\alpha(X_{n+1})\right\} \geq 1 - \alpha.$$

Conformal prediction is a technique tailor-made for constructing such sets C_α that attain valid coverage in finite samples without making any assumptions on our model.

In the next section, Section 2, we provide the relevant technical background, namely Bayesian nonparametric models, Gaussian processes, random Fourier features (RFF), and conformal prediction. In Section 3 we describe in detail our Bayesian nonparametric ensemble and how to do fast inference on it using RFF and do conformal prediction for valid coverage. Finally, in Section 4 we perform extensive experiments on the $\mathrm{PM}_{2.5}$ dataset to test our model.

2 Technical Background

2.1 Bayesian Nonparametric Models

Throughout the paper we assume an underlying probability space $(\Omega, \mathscr{F}, \mathbb{P})$ which serves as the common domain for random variables unless specified otherwise. A *statistical model* $\mathbf{M}$ on a *sample* *space* $\mathcal{Z}$ is a set of probability measures on $\mathcal{Z}$ (assuming some suitable $\sigma-$algebra on $\mathcal{Z}$). If there exists a *parameter space* $\Xi \subseteq \mathbb{R}^d$ for some $d \in \mathbb{N}$ such that the elements of $\mathbf{M}$ can be indexed as $\mathbf{M} = \{\mathbb{P}_\theta : \theta \in \Xi\}$, then $\mathbf{M}$ is called a *parametric model*, otherwise $\mathbf{M}$ is called a *nonparametric model* [Orbanz, 2014]. We focus on nonparametric models $\mathbf{M}$ for which we can write $\mathbf{M} = \{\mathbb{P}_\theta : \theta \in \Xi\}$, and thus Ξ is necessarily infinite-dimensional. As an example of a parametric model, if $\mathbf{M}$ is taken to be the set of all Gaussian measures on $\mathbb{R}$, then $\Xi = \mathbb{R} \times [0, \infty) \subset \mathbb{R}^2$ serves as the parameter space with each element of $\mathbf{M}$ indexed by its mean μ and variance σ^2 as $(\mu, \sigma^2) \in \Xi$. An example of a nonparametric model is Gaussian processes which we discuss in the section 2.2.

In the Bayesian setting, we treat all unknown quantities as random variables, and since the parameter of the data generating distribution is unknown, we treat it as a random variable $\Theta \colon \Omega \to \Xi$ (assuming some suitable $\sigma-$algebra on Ξ). The distribution $\pi = \mathbb{P} \circ \Theta^{-1}$ of Θ is called the *prior distribution*, or simply *prior*, is a measure on Ξ, and represents our state of knowledge of the parameter before seeing any data. The data are represented by a random sequence $Z = (Z_1, Z_2, \ldots)$ such that each $Z_i \colon \Omega \to \mathcal{Z}$ is a random variable. We observe n samples $\{z_1, \ldots, z_n\}$ which are interpreted as observed values of the first n random variables of Z. Under the Bayesian setting, the data are assumed to be generated using the two steps: $\Theta \sim \pi$ and $Z_1, Z_2, \ldots \mid \Theta \overset{\text{i.i.d.}}{\sim} \mathbb{P}_\Theta$. The learning process is then computing the *posterior distribution*, or simply *posterior*, $\pi\left[\Theta \in \cdot \mid Z_1 = z_1, \ldots, Z_n = z_n\right]$, which is the conditional distribution of Θ given the observed data and, intuitively, provides us with an updated beliefs about the parameters.

2.2 Gaussian Process

Recall that for an arbitrary index set T, we call the collection $\Theta = \{\Theta_t\}_{t \in T}$ of random variables a *stochastic process.*

Definition 2.1 (Gaussian process)**.** A stochastic process $\{\Theta_t\}_{t \in T}$ is called a Gaussian process if for any $n \in \mathbb{N}$ and any $t_1, t_2, \ldots, t_n \in T$, $(\Theta_{t_1}, \Theta_{t_2}, \ldots, \Theta_{t_n})$ is a (possibly degenerate) Gaussian random vector.

The distributions of $(\Theta_{t_1}, \Theta_{t_2}, \ldots, \Theta_{t_n})$ for all $t_1, t_2, \ldots, t_n \in T$ are called the *finite-dimensional marginals* of the stochastic process Θ. A Gaussian process is completely specified by its mean function $m(t) := \mathbb{E}\left[\Theta_t\right]$ and covariance function $k(s, t) := \mathbb{E}\left[(\Theta_s - m(s))(\Theta_t - m(t))\right]$. We denote the Gaussian process with mean function m and covariance function k by $\mathcal{GP}(m, k)$.

For our ensemble model, we are interested in learning adaptive weights for each expert. To this end, let $\mathcal{Z}$, the sample space, be the Cartesian product $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X} \neq \varnothing$ is the domain of the experts and $\mathcal{Y} \neq \varnothing$ is the codomain of the experts. Let Ξ, the parameter space, be the set of functions mapping $\mathcal{X}$ to $\mathcal{Y}$, i.e., $\Xi = \mathcal{Y}^{\mathcal{X}}$. Gaussian processes provide a convenient prior over the space of functions. To see this, consider the following alternate viewpoint of Gaussian processes. Taking T to be $\mathcal{X}$ and $\Theta_t \colon \Omega \to \mathcal{Y}$ for each $t \in \mathcal{X}$ in the definition above, we can consider Θ as a random function $\Theta \colon \Omega \to \mathcal{Y}^{\mathcal{X}}$. Then the pushforward measure $\mathbb{P} \circ \Theta^{-1}$ defines a measure on the function space $\mathcal{Y}^{\mathcal{X}}$. Therefore, we can we view the Gaussian process Θ as inducing a distribution over the function space $\Xi = \mathcal{Y}^{\mathcal{X}}$ and choose its distribution $\mathbb{P} \circ \Theta^{-1}$ to be a prior over Ξ.

2.2.1 Gaussian Process Regression

The classical Bayesian linear regression model with Gaussian noise can be written as

$$f(x) = \phi(x)^\top w, \quad Y = f(x) + \varepsilon \tag{1}$$

where $x \in \mathcal{X} \subseteq \mathbb{R}^d$ is the input vector, $\phi \colon \mathbb{R}^d \to \mathbb{R}^D$ takes x into a higher dimensional feature space, $w \in \mathbb{R}^D$ is a vector of weights, Y is the target random variable whose instances are observed, and $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ is independent noise. Suppose we induce a prior $w \sim \mathcal{N}(0, \lambda^{-1} I)$ on the weights. Then it can be shown [Rasmussen and Williams, 2005] that f is a Gaussian process with mean function $\mathbb{E}\left[f(x)\right] = \phi(x)^\top \mathbb{E}\left[w\right] = 0$ and covariance function $\mathbb{E}\left[f(x)f(x')\right] = \phi(x)^\top \phi(x')/\lambda$.

This marginal can be written as a Gaussian process $\Theta \sim \mathcal{GP}(0, k)$, with

$$Y = \Theta_x + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma^2) \tag{2}$$

where ε is independent noise. Given the data $\{(x_i, y_i)\}_{i=1,\dots,n}$ we are interested in the posterior distribution

$$\mathbb{P}\left[\Theta \in \cdot \mid Y_1 = y_1, \dots, Y_n = y_n\right]$$

which as we claimed before is uniquely specified by the finite-dimensional marginals, i.e., the distributions of

$$(\Theta_{x_{n+1}}, \dots, \Theta_{x_{n+m}}) \mid Y_1 = y_1, \dots, Y_n = y_n \tag{3}$$

for all $m \geq 1$. We can view $(x_{n+1}, \dots, x_{n+m})$ as test data for which we need model predictions. It is easily shown that the posterior is also a Gaussian process.

2.3 Kernel Methods

We briefly introduce the relevant ideas from kernel theory. See [Christmann and Steinwart, 2008] (Chapter-4) or [Berlinet and Thomas-Agnan, 2004] for more details.

Definition 2.2 (Positive definite kernel)**.** A function $k \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a positive definite kernel on $\mathcal{X}$, if it is symmetric, i.e., $k(x_1, x_2) = k(x_2, x_1)$ for all $x_1, x_2 \in \mathcal{X}$, and if for any $n \in \mathbb{N}$, $\{a_1, \dots, a_n\} \subset \mathbb{R}$ and $\{x_1, \dots, x_n\} \subseteq \mathcal{X}$, we have

$$\sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j k(x_i, x_j) \geq 0. \tag{4}$$

Condition (4) can be equivalently stated as the Gram matrix, i.e., the matrix $\mathbf{K}$ with elements $\mathbf{K}_{i,j} = k(x_i, x_j)$, is positive semi-definite. $k \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a positive definite kernel if and only if there exists a $\mathbb{R}-$Hilbert space $\mathcal{H}$ and a map $\phi \colon \mathcal{X} \to \mathcal{H}$ such that for all $x_1, x_2 \in \mathcal{X}$ we have

$$k(x_1, x_2) = \langle \phi(x_1), \phi(x_2) \rangle_{\mathcal{H}}$$

This is the well-known *kernel trick* allowing us to implicitly compute inner products in a possibly infinite dimensional feature space $\mathcal{H}$.

Definition 2.3 (Reproducing Kernel Hilbert Space)**.** Let $\mathcal{H}$ be a $\mathbb{R}-$Hilbert function space over $\mathcal{X}$, i.e., a $\mathbb{R}-$Hilbert space that consists of functions mapping from $\mathcal{X}$ into $\mathbb{R}$.

(i) A function $k \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a reproducing kernel of $\mathcal{H}$ if we have $k(\cdot, x) \in \mathcal{H}$ for all $x \in \mathcal{X}$ and the reproducing property

$$f(x) = \langle f, k(\cdot, x) \rangle_{\mathcal{H}}$$

(ii) The space $\mathcal{H}$ is called a reproducing kernel Hilbert space (RKHS) over $\mathcal{X}$ if for all $x \in \mathcal{X}$ the evaluation functionals $\delta_x \colon \mathcal{H} \to \mathbb{R}$ defined by

$$\delta_x(f) = f(x), \quad f \in \mathcal{H}$$

are continuous.

Reproducing kernels are positive definite kernels, every RKHS has a unique reproducing kernel, and every positive definite kernel has a unique RKHS. Thus, positive definite kernels and RKHSs are in one-to-one correspondence.

2.3.1 Kernel Ridge Regression

Suppose our observed data $\mathcal{D}$ are generated by some unknown function with additive noise, i.e.,

$$y_i = g(x_i) + \varepsilon_i \tag{5}$$

where, like before, the noise is independent of each other, and $g \colon \mathcal{X} \to \mathcal{Y}$ is some unknown function. We wish to find a function f^* from an RKHS that best approximates g as follows

$$\begin{aligned} J(f^*) &= \inf_{f \in \mathcal{H}} J(f) \\ &= \inf_{f \in \mathcal{H}} R\left(\|f\|^2_{\mathcal{H}}\right) + L_n(f(x_1), \ldots, f(x_n)) \end{aligned}$$

where, R is a nondecreasing function and L_n is a continuous function. Representer theorem [Schölkopf et al., 2001] guarantees that f^* is of the form $f^* = \sum_{i=1}^n \alpha_i k(\cdot, x_i)$. In kernel ridge regression, we take $R\left(\|f\|^2_{\mathcal{H}}\right) = \sigma^2 \|f\|^2_{\mathcal{H}}$ and $L_n(f(x_1), \ldots, f(x_n)) = \sum_{i=1}^n (f(x_i) - y_i)^2$. See supplementary material section 2.2 or [Kanagawa et al., 2018] for the connection between kernel ridge regression and Gaussian process regression.

2.3.2 Random Fourier Features

Notwithstanding their theoretical simplicity, kernel methods show poor scalability due to their reliance on the Gram matrix. For a data set of n points simply storing the Gram matrix requires $O(n^2)$ space and simple algorithms like Gaussian process regression and kernel ridge regression discussed above require $O(n^3)$ time due to the matrix inversion or equivalent computations. To mitigate this issue many randomized kernel methods have been proposed, among which random Fourier features [Rahimi and Recht, 2008] is widely popular.

Random Fourier features relies on Bochner's theorem [Rudin, 1990], which states that $k \colon \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{C}$ is a translation invariant (i.e., $k(x, y) = k(x - y, 0)$) positive definite kernel if and only if it is the Fourier transform of a finite nonnegative Borel measure Γ, called the *spectral measure*, on $(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$, i.e.,

$$k(x, y) = \int_{\mathbb{R}^d} e^{i\gamma^\top (x-y)} \, \mathrm{d}\Gamma(\gamma)$$

where i is the imaginary number satisfying $i^2 = 1$ (and not the index). Since k is real-valued and symmetric we can simplify the above equation to get

$$k(x, y) = \int_{\mathbb{R}^d} \cos\left(\gamma^\top (x - y)\right) \, \mathrm{d}\Gamma(\gamma)$$

Since $\Gamma(\mathbb{R}^d) = k(0,0)$, by appropriate scaling of the kernel we may, without loss of generality, assume Γ to be a probability measure. For example, if k is the Gaussian kernel $k(x,y) = \exp\left\{-\frac{\|x-y\|_2^2}{2h}\right\}$, where $h > 0$ is the length scale, then since $k(0,0) = 1$, the corresponding measure Γ is a probability measure and is easily checked to be $\mathcal{N}\left(0, \frac{1}{h}\mathbf{I}_d\right)$.

The idea of Rahimi and Recht was to use a Monte Carlo sum to approximate the integral above: if $\gamma_1, \ldots, \gamma_D \overset{\text{i.i.d.}}{\sim} \Gamma$ then using the fact that $\cos\left(\gamma^\top(x-y)\right) = \cos\left(\gamma^\top x\right)\cos\left(\gamma^\top y\right) + \sin\left(\gamma^\top x\right)\sin\left(\gamma^\top y\right)$, we can write

$$k(x,y) \approx \frac{1}{D}\sum_{i=1}^{D}\cos\left(\gamma_i^\top(x-y)\right) = z(x)^\top z(y)$$

where $z\colon \mathbb{R}^d \to \mathbb{R}^{2D}$ is given by

$$z(x) = \\ \frac{1}{\sqrt{D}}\left[\cos\left(\gamma_1^\top x\right), \sin\left(\gamma_1^\top x\right), \ldots, \cos\left(\gamma_D^\top x\right), \sin\left(\gamma_D^\top x\right)\right]^\top$$

An alternative mapping (see [Sutherland and Schneider, 2015] for a comparison) is $z\colon \mathbb{R}^d \to \mathbb{R}^D$ given by

$$z(x) = \sqrt{\frac{2}{D}}\left[\cos\left(\gamma_1^\top x + b_1\right), \ldots, \cos\left(\gamma_D^\top x + b_D\right)\right]^\top$$

where again $\gamma_1, \ldots, \gamma_D \overset{\text{i.i.d.}}{\sim} \Gamma$ and $b_1, \ldots, b_D \overset{\text{i.i.d.}}{\sim} \text{Uniform}[0, 2\pi]$. It is easy to see that in each case $k(x,y) = \mathbb{E}\left[z(x)^\top z(y)\right]$.

For any choice of such a mapping, z, which we call *approximate feature mapping*, we can construct a *substitute kernel* $\tilde{k}$ defined on $\mathbb{R}^d$ by $\tilde{k}(x,y) := z(x)^\top z(y)$ and see that it is in fact a positive definite kernel. Note that it is not necessary that its corresponding RKHS $\tilde{\mathcal{H}}$ is contained in $\mathcal{H}$. Suppose $\mathbf{Z} \in \mathbb{R}^{n\times D}$ is the matrix whose i^{th} row is $z(x_i)^\top$, and let $\tilde{\mathbf{K}} = \mathbf{Z}\mathbf{Z}^\top$. $\tilde{\mathbf{K}}$ is an approximation of $\mathbf{K}$ and is the Gram matrix for the substitute kernel $\tilde{k}$. Using the low-rank approximation $\tilde{\mathbf{K}}$ instead of $\mathbf{K}$ decreases the time complexity of kernel methods from $O(n^3)$ to $O(D^2n)$ and space complexity from $O(n^2)$ to $O(Dn)$, a significant improvement.

2.4 Conformal Prediction

Conformal prediction was introduced by Vovk, Gammerman and Shafer in the early 2000s. See the book [Vovk et al., 2005] and the tutorials [Shafer and Vovk, 2008, Angelopoulos and Bates, 2021] for a detailed introduction.

The setting is the same as in the introduction: we have the training data $\mathcal{D} = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ drawn i.i.d. from an unknown distribution $\mathbb{P}$ on the product space $\mathcal{X} \times \mathcal{Y}$. We draw an independent test point (X_{n+1}, Y_{n+1}) from $\mathbb{P}$, and we would like to construct a *prediction interval* $C_\alpha(X_{n+1})$ for $\alpha \in (0,1)$ such that

$$\mathbb{P}\left\{Y_{n+1} \in C_\alpha(X_{n+1})\right\} \geq 1 - \alpha.$$

Suppose we train a model $\mathcal{X} \ni x \mapsto \widehat{f}(x) = (\widehat{f}_1(x), \widehat{f}_2(x)) \in (\mathcal{Y}, \mathcal{P}(\mathcal{Y}))$ on the training data $\mathcal{D}$. Here the output of $\widehat{f}$ is a pair containing a prediction $\widehat{f}_1$–say, the mode of the posterior predictive in Bayesian models, and a probability distribution $\widehat{f}_2$–say, the posterior predictive in the case of Bayesian models. The prediction interval computed using this distribution may not be accurate, perhaps due to various approximations at different stages of the inference or because the model itself does not correctly represent the reality. Therefore, a natural empirical method to compute the prediction intervals would then be to let

$$C_\alpha(X_{n+1}) = \left[\widehat{f}_1(X_{n+1}) - \mathsf{Q}_{1-\alpha}\left(\left\{\left|Y_i - \widehat{f}_1(X_i)\right|\right\}_{i=1}^{n}\right),\ \widehat{f}_1(X_{n+1}) + \mathsf{Q}_{1-\alpha}\left(\left\{\left|Y_i - \widehat{f}_1(X_i)\right|\right\}_{i=1}^{n}\right)\right],$$

where Q_τ denotes the $\tau-$quantile of a distribution and $\mathsf{Q}_{1-\alpha}\left(\left\{\left|Y_i - \widehat{f}_1(X_i)\right|\right\}_{i=1}^{n}\right)$ is assumed to mean the $1-\alpha$ empirical quantile for the points $\left\{\left|Y_i - \widehat{f}_1(X_i)\right|\right\}_{i=1}^{n}$, i.e., $\mathsf{Q}_{1-\alpha}\left(\left\{\left|Y_i - \widehat{f}_1(X_i)\right|\right\}_{i=1}^{n}\right)$ is the $\lceil(1-\alpha)n\rceil^{\text{th}}$ smallest value in the set $\left\{\left|Y_i - \widehat{f}_1(X_i)\right|\right\}_{i=1}^{n}$.

However, the residuals $\left|Y_i - \widehat{f}_1(X_i)\right|$ for the training data will typically be smaller than the residuals for unobserved data, and thus this interval would typically undercover. Therefore, another approach could be to do $K-$fold cross-validation, also known as jacknife in the literature [Barber et al., 2021]. That is, we split the training data $\mathcal{D}$ into K disjoint subsets $\mathcal{D}_1, \dots, \mathcal{D}_K$ each of size approximately n/K. Then train $\widehat{f}_{-i}$ on $\mathcal{D} \setminus \mathcal{D}_i$ and use

$$C_\alpha(X_{n+1}) = \left[\widehat{f}_1(X_{n+1}) - \mathsf{Q}_{1-\alpha}\left(\left\{\left|Y_i - \widehat{f}_{-k(i),1}(X_i)\right|\right\}_{i=1}^{n}\right),\ \widehat{f}_1(X_{n+1}) + \mathsf{Q}_{1-\alpha}\left(\left\{\left|Y_i - \widehat{f}_{-k(i),1}(X_i)\right|\right\}_{i=1}^{n}\right)\right]$$

as the prediction interval, where $k(i) \in \{1, \dots, K\}$ is such that $i \in \mathcal{D}_{k(i)}$. Rich literature exists for this method, see [Stone, 1974, Geisser, 1975, Butler and Rothman, 1980, Barber et al., 2021], but coverage guarantees exist only in the asymptotic regime.

Conformal prediction allows for finite sample coverage guarantees. We will use the *split conformal prediction*, which despite being fast gives prediction intervals with statistical guarantees. We start by splitting the data $\mathcal{D}$ into two disjoint subsets, $\mathcal{D}_T$ and $\mathcal{D}_C$, such that $\mathcal{D}_T$ is now used for training the model and $\mathcal{D}_C$ is left untouched during training. We design a *score function* $S\colon \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, and this is important for the success of conformal prediction, such that $S(X, Y)$ measures how "unusual" the pair (X, Y) is according to our trained model. Two examples of score functions are $\left|Y - \widehat{f}_1(X)\right|$ and $\left|Y - \widehat{f}_1(X)\right|/u(X)$, where $u(X)$ measures the uncertainty at X according to the model. We will choose the second choice, since it gives us more flexible prediction intervals. Then compute

$$\widehat{q} = \mathsf{Q}_{1-\alpha}\left(\{S(X, Y)\}_{(X,Y)\in\mathcal{D}_C}\right).$$

Finally, the prediction interval for a new test point (X_{n+1}, Y_{n+1}) is given by

$$C_\alpha(X_{n+1}) = \left[\widehat{f}_1(X_{n+1}) - u(X_{n+1})\widehat{q},\ \widehat{f}_1(X_{n+1}) + u(X_{n+1})\widehat{q}\right]$$

Conformal prediction guarantees that

$$\mathbb{P}\left\{Y_{n+1} \in C_\alpha(X_{n+1})\right\} \geq 1 - \alpha.$$

For a proof, see [Vovk et al., 2005]. There the assumption of i.i.d. data is relaxed to exchangeability of data. It turns out that even exchangeability can be relaxed, see [Barber et al., 2022].

The prediction intervals obtained using conformal prediction satisfy some desired properties: 1) they are smaller when α gets bigger, and 2) they are as small as possible.

3 Bayesian Nonparametric Ensemble

The model weights c_ℓ's are highly dependent on each other and simple Bayesian nonparametric models don't provide the flexibility to model this dependence. Dependent tail-free process of [Jara and Hanson, 2011], which we recall next, provide a natural and powerful framework for inducing a prior on the model weights c_ℓ's.

Let $q \geq 2$, $E = \{1, 2, \ldots, q\}$ and $E^* = \bigcup_{m=1}^{\infty} E^m$. Consider the sequence of partitions of $\mathbb{R}$ given by $\pi_0 = \{\mathbb{R}\}$, $\pi_1 = \{B_1, \ldots, B_q\}$, $\pi_2 = \{B_{11}, \ldots, B_{1q}, B_{21}, \ldots, B_{2q}, \ldots, B_{q1}, \ldots, B_{qq}\}$, …, such that $B_1 \cup \cdots \cup B_q = \mathbb{R}$, $B_i \cap B_j = \varnothing$ for any $i \neq j$, and for each $m \in \mathbb{N}$ and every $\varepsilon = \varepsilon_1 \cdots \varepsilon_m \in E^m$, $B_\varepsilon = B_{\varepsilon 1} \cup \cdots \cup B_{\varepsilon q}$ and $B_{\varepsilon i} \cap B_{\varepsilon j} = \varnothing$ for any $i \neq j$. Assume that for any $\varepsilon \in E^*$, B_ε is a left-open right-closed interval unless ε is a string of q's only, and that the elements of each partition π_m are arranged in the natural order, i.e., $B_{1\cdots 1}$ comes first, then $B_{1\cdots q}$, and so on. Let $\pi^* = \bigcup_{m=0}^{\infty} \pi_m$ and further assume that the $\sigma-$algebra generated by π^* contains the Borel $\sigma-$algebra $\mathcal{B}(\mathbb{R})$. Let $h \colon \mathbb{R}^q \to \{u \in [0,1]^q \ : \ u_1 + \cdots + u_q = 1\}$ be a continuous function such that it is strictly increasing in each of its arguments. For example, h could be the softmax function.

Definition 3.1. Let $\mathcal{A} = \{V_\varepsilon \ : \ \varepsilon \in E^*\}$ be a set of covariance functions and $\mathcal{P}(\mathbb{R})$ be the set of all Borel probability measures on $\mathbb{R}$. Let $G = \{G_x \ : \ x \in \mathcal{X}\}$ be a $\mathcal{P}(\mathbb{R})-$valued stochastic process on the underlying probability space $(\Omega, \mathscr{F}, \mathbb{P})$, i.e., each $G_x \colon \Omega \to \mathcal{P}(\mathbb{R})$ is a random probability measure. Suppose G is such that:

1. the sets $\{g_{\varepsilon,x} \ : \ x \in \mathcal{X}\}$, for every $\varepsilon \in E^*$, are mutually independent zero mean Gaussian processes with covariance functions V_ε respectively;

2. for every $x \in \mathcal{X}$, $\omega \in \Omega$ and every $\varepsilon = \varepsilon_1 \cdots \varepsilon_m \in E^*$, $Y_{\varepsilon_1 \cdots \varepsilon_{m-1} i}(x, \omega) = h(g_{\varepsilon,x}(\omega))_i$ for every $i = 1, \ldots, L$;

3. for every $x \in \mathcal{X}$ and $\varepsilon = \varepsilon_1 \cdots \varepsilon_m \in E^*$,

$$G_x(\omega)(B_\varepsilon) = \prod_{j=1}^{m} Y_{\varepsilon_1 \cdots \varepsilon_j}(x, \omega)$$

Then the process G is called a dependent tail-free process. We denote it as $G \sim \mathrm{DTFP}(\mathcal{X}, h, \mathcal{A}, \pi^*)$.

An example of dependent tail-free processes are Pólya trees, which themselves are generalization of Dirichlet processes [Lavine, 1992, Ferguson, 1974].

Recall the training data set $\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ and the collection $\mathcal{S} = \{S_1, \ldots, S_L\}$ of base models from Section 1. The Bayesian nonparametric ensemble $\widehat{S} \colon \mathcal{X} \to \mathcal{Y}$ is constructed using a dependent tail-free process. Suppose we believe a-priori that the collection $\mathcal{S}$ can be arranged in a hierarchical order as a tree $\mathcal{T}_\mathcal{S}$, called *model tree*, possibly using knowledge of the data or the training process used to get the base models. Suppose the height of $\mathcal{T}_\mathcal{S}$ is m and the maximum degree is q. See Figure 1 for an example with $m = 2$ and $q = 3$.

We can then construct a *partition tree*, $\mathcal{T}_{\mathbb{R},\mathcal{S}}$, which corresponds naturally to the model tree $\mathcal{T}_\mathcal{S}$ such that the root of $\mathcal{T}_{\mathbb{R},\mathcal{S}}$ is $\mathbb{R}$, the children of each node form the node's partition, the height is m, and the degree of every non-leaf node is q, possibly using empty sets as nodes. Note that the leaf nodes of $\mathcal{T}_{\mathcal{S},\mathbb{R}}$ form a partition of $\mathbb{R}$. We can associate every node in this tree to a set of models. See Figure 2 for the construction corresponding to Figure 1. Let $B_\varepsilon \subseteq \mathbb{R}$ be the node in $\mathcal{T}_{\mathbb{R},\mathcal{S}}$ corresponding to $\mathcal{S}_\varepsilon \subseteq \mathcal{S}$, where $\mathcal{S}_\varepsilon$ is a node from $\mathcal{T}_\mathcal{S}$. Then if we have a dependent tail-free

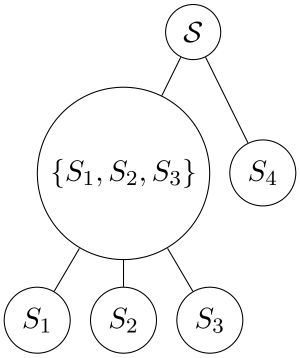


Figure 1: An example of a model tree obtained from some hierarchical structure for $\mathcal{S} = \{S_1, S_2, S_3, S_4\}$.

process $G \sim \text{DTFP}(\mathcal{X}, h, \mathcal{A}, \pi^*)$ where π^* is chosen such that it contains all aforementioned B_ε's, $G_x(\omega)(B_\varepsilon)$ gives us the probability that at point $x \in \mathcal{X}$ the set of models $\mathcal{S}_\varepsilon$ explains the output. These probabilities could equivalently be used to weight the L models to get an ensemble:

$$\widehat{S}(x)(\omega) = \sum_{\ell=1}^{L} G_x(\omega)(B_{\varepsilon_\ell}) S_\ell(x)$$

where B_{ε_ℓ} is the partition that corresponds to S_ℓ. Since all B_{ε_ℓ} are the leaves in the partition tree, and all other leaves correspond to empty sets, we have $\sum_{\ell=1}^{L} G_x(\omega)(B_{\varepsilon_\ell}) = 1$.

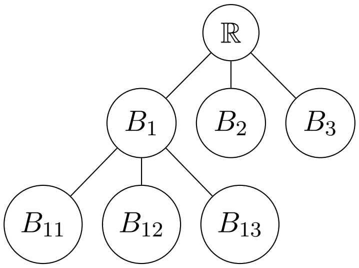
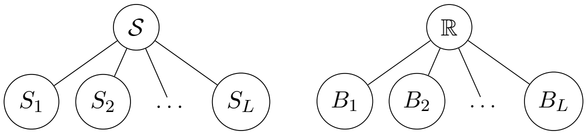


Figure 2: With reference to Figure 1, $\mathbb{R}$ corresponds to $\mathcal{S}$, B_1 corresponds to the set $\{S_1, S_2, S_3\}$ of models, B_{11} corresponds to S_1, B_{12} corresponds to S_2, B_{13} corresponds to S_3, B_2 corresponds to S_4, and $B_3 = \varnothing$ and corresponds to no model. $\{B_1, B_2, B_3\}$ form a partition of $\mathbb{R}$, $\{B_{11}, B_{12}, B_{13}\}$ form a partition of B_1, and therefore the leaves $\{B_{11}, B_{12}, B_{13}, B_2, B_3\}$ form a partition of $\mathbb{R}$.

We augment the collection $\mathcal{S}$ with a model S_0 which is always 1 and serves as a bias term. This serves to mitigate the collection $\{S_1, \ldots, S_L\}$'s systematic bias. We focus on a special case of the Bayesian nonparametric ensemble described above. We assume no a-priori knowledge of the models and therefore assume the model tree $\mathcal{T}_\mathcal{S}$ to be of height 1 as shown in Figure 3a. The corresponding partition tree $\mathcal{T}_{\mathcal{S},\mathbb{R}}$ is shown in Figure 3b.

The dependent tail-free process we use is such that the set of covariance functions $\mathcal{A}$ contains only one kernel k on $\mathcal{X} \subseteq \mathbb{R}^d$, the function h on $\mathbb{R}^L$ is the softmax function $h(v)_\ell = \exp\{v_\ell\} / \sum_{p=1}^{L} \exp\{v_p\}$ for $v \in \mathbb{R}^L$, and π^* contains the sets $B_1, \ldots, B_L$. Then if $G \sim \text{DTFP}(\mathcal{X}, h, \mathcal{A}, \pi^*)$ and we denote $c_\ell(x)(\omega) = G_x(\omega)(B_\ell)$, we could describe our model by the following simple genera-



(a) Model tree $\mathcal{T}_{\mathcal{S}}$. (b) Partition tree $\mathcal{T}_{\mathcal{S},\mathbb{R}}$.

Figure 3: The model and partition trees for our model.

tive process:

$$
\begin{aligned}
& y_i \mid \widehat{S}, x_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\widehat{S}(x_i), \sigma^2), \quad i = 1, \dots, n \\
& \widehat{S}(x_i) = \sum_{\ell=0}^{L} c_\ell(x_i) S_\ell(x_i), \quad i = 1, \dots, n \\
& c_0 = g_0, \ c_\ell = \frac{\exp\{g_\ell\}}{\sum_{p=1}^{L} \exp\{g_p\}}, \quad \ell = 1, \dots, L \\
& g_\ell \overset{\text{i.i.d.}}{\sim} \mathcal{GP}(0, k), \quad \ell = 0, 1, \dots, L
\end{aligned}
\tag{6}
$$

The model weights c_ℓ allow for adaptive weighting of different models. Modeling these weights using a function of Gaussian processes allows for uncertainty quantification. Intuitively, in regions of input space $\mathcal{X}$ with low overlap with $\mathcal{D}$ or where the models $\{S_1, \dots, S_L\}$ disagree, the uncertainty in weights will be high. The inference task we are interested in is learning the posterior distribution for the model weights.

3.1 Approximation Using RFF

Inference for even the simple model described above is unfeasible for large n due to the time and memory requirements. Random Fourier features allow us to tackle this limitation by using the approximate Gram matrix $\tilde{\mathbf{K}}$ defined in Section 2.3.2.

Recall the Gaussian processes-view of regression from Section 2.2.1 and combine that with the RFF approximation $k(x, x') \approx z(x)^\top z(x')$. Then if we impose the priors $w_\ell \sim \mathcal{N}\left(0, \lambda^{-1} \mathbf{I}_D\right)$ on the $L + 1$ weight vectors from equation (1), we can approximate the Gaussian processes in (6) as

$$
g_\ell(x) \approx \hat{g}_\ell(x) = z(x)^\top w_\ell, \quad x \in \mathcal{X}, \ \ell = 0, 1, \dots, L.
$$

Approximations to the model weights become

$$
\hat{c}_0 = \hat{g}_0, \ \hat{c}_\ell = \frac{\exp\{\hat{g}_\ell\}}{\sum_{p=1}^{L} \exp\{\hat{g}_p\}}, \quad \ell = 1, \dots, L.
$$

The ensemble becomes

$$
\widehat{S}(x) = \sum_{\ell=0}^{L} \hat{c}_\ell(x) S_\ell(x) = S(x)^\top \hat{c}(x) + z(x)^\top w_0
$$

where $S(x) := [S_1(x), \dots, S_L(x)]^\top$ and $\hat{c}(x) := [\hat{c}_1(x), \dots, \hat{c}_L(x)]^\top$. Like before, we are interested in learning the posterior distribution over the model weights $\hat{c}_\ell$'s, which now reduces to learning the posterior distribution over the weight vectors w_ℓ's. We have tamed our nonparametric model by converting it into a parametric model.

3.2 Estimation

Getting the exact posterior or an approximation using Gibbs sampling is impossible since the posterior or the conditional terms needed for Gibbs sampling are not closed form. Markov Chain Monte Carlo methods are prohibitively slow. Although mean-field variational inference provides a convenient way to approximate the posterior, the latent variables would not remain interdependent, and it necessitates variance reduction techniques [Ranganath et al., 2014]. We therefore do the simple yet powerful maximum a posteriori (MAP) & Laplacian approximation.

The generative model allows us to get MAP and Laplace estimates by maximizing the log joint likelihood (ignoring the terms constant with respect to the weights)

$$\begin{aligned}\mathcal{L} &= -\frac{1}{2\sigma^2}\sum_{i=1}^{n}\Big(y_i - \widehat{S}(x_i)\Big)^2 - \frac{\lambda}{2}\sum_{\ell=0}^{L}\|w_\ell\|^2 \qquad (7)\\ &= -\frac{1}{2\sigma^2}\sum_{i=1}^{n}\Big(y_i - S(x_i)^\top \hat{c}(x_i) - z(x_i)^\top w_0\Big)^2\\ &- \frac{\lambda}{2}\sum_{\ell=0}^{L}\|w_\ell\|^2\end{aligned}$$

3.2.1 Gradients And Hessians

We need to compute the gradients of the log joint likelihood (7) with respect to the weight vectors w_ℓ's and compute the Hessian to be able to find MAP and Laplace estimate. The gradient with respect to w_0 is simple:

$$\nabla_{w_0}\mathcal{L} = \frac{1}{\sigma^2}\sum_{i=1}^{n}\Big(y_i - \widehat{S}(x_i)\Big)\, z(x_i) - \lambda w_0.$$

Before we compute the gradients $\nabla_{w_\ell}\mathcal{L}$ for $\ell = 1, \dots, L$, we note that if $A_j = \frac{\exp\{g(a_j)\}}{\sum_{p=1}^{L}\exp\{g(a_p)\}}$, where g is a real-valued differential function, then if $\delta_{j\ell}$ denotes the Kronecker delta which equals 1 if $j = \ell$ and 0 otherwise, we have

$$\begin{aligned}\frac{\partial A_j}{\partial a_\ell} &= \frac{1}{\sum_{p=1}^{L}\exp\{g(a_p)\}}\frac{\partial \exp\{g(a_j)\}}{\partial a_\ell} - \exp\{g(a_j)\}\frac{\partial(\sum_{p=1}^{L}\exp\{g(a_p)\})^{-1}}{\partial a_\ell}\\ &= \delta_{j\ell}A_j\frac{\partial g(a_\ell)}{\partial a_\ell} - \frac{A_j}{\sum_{p=1}^{L}\exp\{g(a_p)\}}\frac{\partial g(a_\ell)}{\partial a_\ell}\\ &= A_j\frac{\partial g(a_\ell)}{\partial a_\ell}\left(\delta_{j\ell} - A_\ell\right).\end{aligned}$$

Thus, $\nabla_{w_\ell}\hat{c}_j(x_i) = \hat{c}_j(x_i)z(x_i)\left(\delta_{j\ell} - \hat{c}_\ell(x_i)\right)$, and we get

$$\begin{aligned}\nabla_{w_\ell}\mathcal{L} &= \frac{1}{\sigma^2}\sum_{i=1}^{n}\Big(y_i - \widehat{S}(x_i)\Big)\nabla_{w_\ell}S(x_i)^\top \hat{c}(x_i)\\ &= \frac{1}{\sigma^2}\sum_{i=1}^{n}\Big(y_i - \widehat{S}(x_i)\Big)\hat{c}_\ell(x_i)z(x_i)\left[S_\ell(x_i) - S(x_i)^\top \hat{c}(x_i)\right].\end{aligned}$$

Then gradient ascent allows us to quickly find the MAP. To get the Laplace approximation, we need to compute the Hessians. The Hessian will be a $D(L+1) \times D(L+1)$ matrix storing all second order gradients. We compute each block $D \times D$ of the Hessian one by one. The top left $D \times D$ block is $\nabla^2_{w_0} \mathcal{L}$ given by

$$\nabla^2_{w_0} \mathcal{L} = -\frac{1}{\sigma^2} \sum_{i=1}^{n} z(x_i) z(x_i)^\top - \lambda \mathbf{I}_D.$$

The L blocks along the first column and the first row are

$$\nabla_{w_\ell} \nabla_{w_0} \mathcal{L} = -\frac{1}{\sigma^2} \sum_{i=1}^{n} z(x_i) z(x_i)^\top \hat{c}_\ell(x_i) \left[S_\ell(x_i) - S(x_i)^\top \hat{c}(x_i) \right], \quad \ell = 1, \ldots, L$$

Note that $\nabla_{w_0} \nabla_{w_\ell} \mathcal{L} = \nabla_{w_\ell} \nabla_{w_0} \mathcal{L}$. The L blocks along the diagonal of the Hessian are

$$\nabla^2_{w_\ell} \mathcal{L} = -\frac{1}{\sigma^2} \sum_{i=1}^{n} z(x_i) z(x_i)^\top \left(u_i^{(1)} + u_i^{(2)} + u_i^{(3)} \right) - \lambda \mathbf{I}_D, \quad \ell = 1, \ldots, L$$

where

$$\begin{aligned}
u_i^{(1)} &= \hat{c}_\ell^2(x_i) \left[S_\ell(x_i) - S(x_i)^\top \hat{c}(x_i) \right]^2 \\
u_i^{(2)} &= -\hat{c}_\ell(x_i)(1 - \hat{c}_\ell(x_i)) \left(y_i - \widehat{S}(x_i) \right) \left[S_\ell(x_i) - S(x_i)^\top \hat{c}(x_i) \right] \\
u_i^{(3)} &= \hat{c}_\ell^2(x_i) \left(y_i - \widehat{S}(x_i) \right) \left[S_\ell(x_i) - S(x_i)^\top \hat{c}(x_i) \right].
\end{aligned}$$

The off-diagonal blocks for $j \neq \ell$ are given by

$$\nabla_{w_j} \nabla_{w_\ell} \mathcal{L} = -\frac{1}{\sigma^2} \sum_{i=1}^{n} z(x_i) z(x_i)^\top \left(v_i^{(1)} + v_i^{(2)} + v_i^{(3)} \right)$$

where

$$\begin{aligned}
v_i^{(1)} &= \hat{c}_j(x_i) \hat{c}_\ell(x_i) \left[S_j(x_i) - S(x_i)^\top \hat{c}(x_i) \right] \left[S_\ell(x_i) - S(x_i)^\top \hat{c}(x_i) \right] \\
v_i^{(2)} &= \hat{c}_j(x_i) \hat{c}_\ell(x_i) \left(y_i - \widehat{S}(x_i) \right) \left[S_j(x_i) - S(x_i)^\top \hat{c}(x_i) \right] \\
v_i^{(3)} &= \hat{c}_j(x_i) \hat{c}_\ell(x_i) \left(y_i - \widehat{S}(x_i) \right) \left[S_\ell(x_i) - S(x_i)^\top \hat{c}(x_i) \right].
\end{aligned}$$

Then the posterior is approximated as a Gaussian distribution with mean being the MAP and covariance being the inverse of the negative of the Hessian [Bishop, 2006].

4 Experiments

Particulate matter 2.5, or PM$_{2.5}$, are fine inhalable particles with diameters that are 2.5 micrometers or smaller. Excess exposure to them can cause multiple respiratory problems and therefore monitoring of their levels is an important environmental problem. For our experiments we focus on the problem of predicting the PM$_{2.5}$ concentration within the contiguous United States. Throughout the contiguous United States there are about 1000 ground monitoring sites which provide daily or

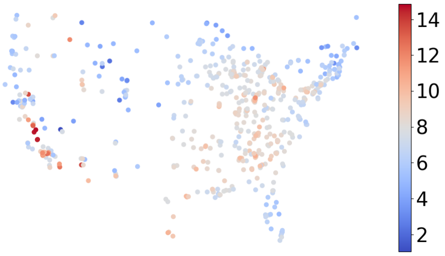


Figure 4: Annualized average $PM_{2.5}$ concentration readings for the year 2015 by the ground monitors.

weekly readings depending on the monitoring site. See Figure 4 for the locations and the average readings for the year 2015 for all the ground monitors. All the readings are in $\mu g/m^3$.

As can be seen there are large areas where the readings are sparse and therefore multiple approaches have been developed by various environmental teams [Di et al., 2019, Hammer et al., 2020, Shaddick et al., 2016, Kim et al., 2020] which use this ground monitoring data along with satellite-derived aerosol optical depth, land-use variables, chemical transport model predictions, and several meteorological variables to make much more fine-grained (1 km × 1 km resolution, for example) predictions for $PM_{2.5}$. Since these approaches use differing data and differing models they have own strengths and weaknesses. This data provides a perfect test-bed for BNE: we use 6 such meteorological models as our base models and attempt to learn an ensemble model on top of them. The expectation is that BNE would perform better than each of the base models.

4.1 Experimental Setup

The data $\mathcal{D} = \{x_i, y_i\}_{i=1,\ldots,n}$ we use for our experiments are the daily readings by the ground monitors for the six years 2010–2015. The number of readings n during this time is about 600k. Each $x_i \in \mathbb{R}^3$ and is the triple (latitude, longitude, time stamp). Each $y_i \in \mathbb{R}$ is the reading for $PM_{2.5}$ by a ground monitor at x_i. We use a Gaussian kernel with different length scales for the spatial and temporal coordinates since they are not comparable. The hyperparameters for BNE are the number of random Fourier features, D, the parameter λ for the prior of the weights w_ℓ, the length scales h and h_t for spatial and temporal coordinates respectively, and the noise variance σ^2.

We perform 20−fold cross validation to tune the hyperparameters and measure different models' performance as follows: fix the hyperparameters; randomly partition the data $\mathcal{D}$ into 20 blocks; for the i^{th} fold run, keep the i^{th} block separate and train using gradient ascent on the remaining 19 blocks; then measure performance metrics on the i^{th} block like mean absolute error, root mean square error and coverage (percentage of the points in i^{th} block whose true values lie within two standard deviations of the predicted values); finally compute the median of the performance metrics obtained in the 20 runs. See Figure 5 for a plot of the mean absolute errors for different values of three hyperparameters. The hyperparameter values for BNE in table 1 are $D = 500$, $\lambda = 0.0005$, $h = 20.0$, $h_t = 25.0$, and $\sigma^2 = \text{Var}\left(\{y_i\}_{i=1}^n\right)/8$ (i.e., a signal to noise ratio of 8).

Figure 6 visualizes how the BNE weights different base models adaptively depending on the spatial as well as the temporal location. Each row in the figure corresponds to a single day – winter days are 1 January of the corresponding year, spring days are 1 April of the corresponding year, summer days are 1 July of the corresponding year, and autumn days are 1 October of the corre-

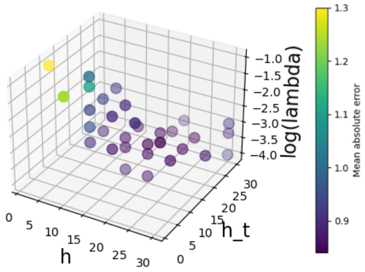


Figure 5: Hyperparameters.

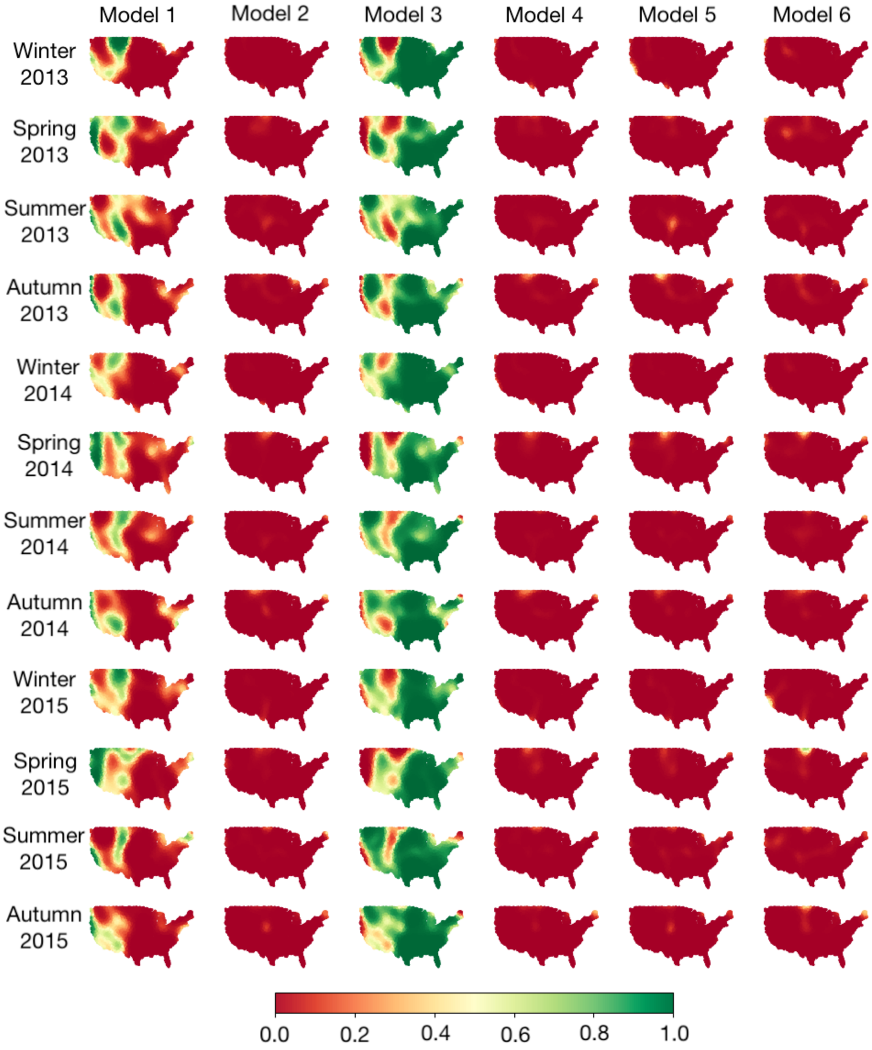


Figure 6: Weights given by BNE to different base models. Each row represents a particular day in a season and colors denote the weights $c_\ell(x)$ given by the BNE to each model at every location.

Table 1: Different models' performance.

Model	**MAE**	**RMSE**
Model 1	1.26	1.97
Model 2	3.93	5.96
Model 3	0.93	1.59
Model 4	3.97	5.22
Model 5	3.82	5.11
Model 6	3.62	4.81
Non-adaptive ensemble	0.88	1.47
Gaussian process regression	3.77	5.05
BNE	**0.84**	**1.39**

sponding year. We can see that model 1 is given high weight on the west coast during spring. Model 3 stands out due to the high weights in many regions – not surprising considering its performance is much better than other base models as can seen from Table 1.

4.2 Baselines

As a baseline, we compare BNE against a non-adaptive ensemble $\tilde{S}$, given by

$$\tilde{S}(x) = \textstyle\sum_{\ell=0}^{L} \tilde{c}_\ell S_\ell(x),$$

where $\sum_{\ell=1}^{L} \tilde{c}_\ell = 1$ and recall $S_0 = 1$. The weights $\tilde{c}$ are *not* a function of x unlike in BNE. Assuming a prior $\tilde{w}_\ell \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \lambda^{-1})$ where $\tilde{c}_\ell = \exp\{\tilde{w}_\ell\} / \sum_{p=1}^{L} \exp\{\tilde{w}_p\}$ and the likelihood function as $y_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\tilde{S}(x_i), \sigma^2)$, the joint log likelihood function is

$$\tilde{\mathcal{L}} = -\frac{1}{2\sigma^2} \sum_{i=1}^{n} \Big(y_i - \tilde{S}(x_i)\Big)^2 - \frac{\lambda}{2} \sum_{\ell=0}^{L} \tilde{w}_\ell^2$$

Table 1 shows that this non-adaptive ensemble performs better than each of the base models, but worse than BNE, showing that adaptivity plays an important role in BNE's performance.

For another baseline we use Gaussian process regression on the data. The large size of the data necessitates approximation using random Fourier features. This baseline does not use the six base models at all and uses only the data $\mathcal{D}$. As expected this method performs poorly as can be seen from Table 1, since it doesn't exploit the expertise of the base models.

4.3 Uncertainty Estimation

To use conformal prediction, we change the experimental setup. The data $\mathcal{D} = \{x_i, y_i\}_{i=1,\dots,n}$ we use for our experiments are still the daily readings by the ground monitors for the six years 2010–2015. We then split $\mathcal{D}$ into three disjoint subsets $\mathcal{D}_{\text{Tr}}, \mathcal{D}_C$ and $\mathcal{D}_{\text{Te}}$: $\mathcal{D}_{\text{Tr}}$ is used for training and contains

90% of the full data $\mathcal{D}$, $\mathcal{D}_C$ is used for calibration for conformal prediction and is 5% of $\mathcal{D}$, and $\mathcal{D}_{\mathrm{Te}}$ is used for testing at the end to get coverage estimates and is 5% of $\mathcal{D}$.

We train the model on $\mathcal{D}_{\mathrm{Tr}}$ to get the MAP estimate and Laplace approximation of the posterior. We then compute $\widehat{q}$ on the calibration data $\mathcal{D}_C$ by using the estimate by posterior predictive given by a Monte Carlo estimate

$$p(y \mid X_{n+1}, \mathcal{D}_{\mathrm{Tr}}) = \int p(y \mid X_{n+1}, \theta) p(\theta \mid \mathcal{D}_{\mathrm{Tr}})\, \mathrm{d}\theta \approx \frac{1}{S} \sum_{s=1}^{S} p(y \mid X_{n+1}, \theta_s)$$

as the uncertainty function u mentioned in Section 2.4. We use $\alpha = 0.05$ to get the 95% prediction intervals. We also calculate the 95% prediction interval estimated using the model's uncertainty.

Finally, we compare the coverage of the prediction intervals on the test data $\mathcal{D}_{\mathrm{Te}}$. The coverage for the model's inherent uncertainty is only 30.5%, while the coverage for conformal prediction is 99.9%. But note that the prediction intervals using conformal prediction are much wider and therefore it's no surprise that the coverage is higher. For example, the average width of the prediction intervals according to the model is only 1.33 μg/m^3, while for conformal prediction the average width is 13.4 μg/m^3. On the other hand, the model severely under-covers and conformal prediction was necessary to get more accurate coverage.

References

[Angelopoulos and Bates, 2021] Angelopoulos, A. N. and Bates, S. (2021). A gentle introduction to conformal prediction and distribution-free uncertainty quantification.

[Barber et al., 2022] Barber, R. F., Candes, E. J., Ramdas, A., and Tibshirani, R. J. (2022). Conformal prediction beyond exchangeability.

[Barber et al., 2021] Barber, R. F., Candès, E. J., Ramdas, A., and Tibshirani, R. J. (2021). Predictive inference with the jackknife+. *The Annals of Statistics*, 49(1):486 – 507.

[Bell and Koren, 2007] Bell, R. M. and Koren, Y. (2007). Lessons from the netflix prize challenge. *SIGKDD Explor. Newsl.*, 9(2):75–79.

[Berlinet and Thomas-Agnan, 2004] Berlinet, A. and Thomas-Agnan, C. (2004). *Reproducing Kernel Hilbert Spaces in Probability and Statistics.* Springer US.

[Bishop, 2006] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning.* Springer-Verlag New York.

[Butler and Rothman, 1980] Butler, R. and Rothman, E. D. (1980). Predictive intervals based on reuse of the sample. *Journal of the American Statistical Association*, 75(372):881–889.

[Christmann and Steinwart, 2008] Christmann, A. and Steinwart, I. (2008). *Support Vector Machines.* Springer New York.

[Çınlar, 2011] Çınlar, E. (2011). *Probability and Stochastics.* Graduate Texts in Mathematics. Springer New York.

[Claeskens and Hjort, 2008] Claeskens, G. and Hjort, N. L. (2008). *Model Selection and Model Averaging*. Cambridge Series In Statistical and Probabilistic Mathematics, Cambridge University Press.

[Di et al., 2019] Di, Q., Amini, H., Shi, L., Kloog, I., Silvern, R., Kelly, J., Sabath, M. B., Choirat, C., Koutrakis, P., Lyapustin, A., Wang, Y., Mickley, L. J., and Schwartz, J. (2019). An ensemble-based model of pm2.5 concentration across the contiguous united states with high spatiotemporal resolution. *Environment International*, 130:104909.

[Dietterich, 2000] Dietterich, T. G. (2000). Ensemble methods in machine learning. In *Multiple Classifier Systems*, pages 1–15, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Ferguson, 1974] Ferguson, T. S. (1974). Prior Distributions on Spaces of Probability Measures. *The Annals of Statistics*, 2(4):615 – 629.

[Geisser, 1975] Geisser, S. (1975). The predictive sample reuse method with applications. *Journal of the American Statistical Association*, 70(350):320–328.

[Hammer et al., 2020] Hammer, M. S., van Donkelaar, A., Li, C., Lyapustin, A., Sayer, A. M., Hsu, N. C., Levy, R. C., Garay, M. J., Kalashnikova, O. V., Kahn, R. A., Brauer, M., Apte, J. S., Henze, D. K., Zhang, L., Zhang, Q., Ford, B., Pierce, J. R., and Martin, R. V. (2020). Global estimates and long-term trends of fine particulate matter concentrations (1998–2018). *Environmental Science & Technology*, 54(13):7879–7890. PMID: 32491847.

[Jara and Hanson, 2011] Jara, A. and Hanson, T. E. (2011). A class of mixtures of dependent tail-free processes. *Biometrika*, 98(3):553–566.

[Kanagawa et al., 2018] Kanagawa, M., Hennig, P., Sejdinovic, D., and Sriperumbudur, B. K. (2018). Gaussian processes and kernel methods: A review on connections and equivalences.

[Kim et al., 2020] Kim, S.-Y., Bechle, M., Hankey, S., Sheppard, L., Szpiro, A., and Marshall, J. (2020). Concentrations of criteria pollutants in the contiguous u.s., 1979 – 2015: Role of prediction model parsimony in integrated empirical geographic regression. *PLOS ONE*, 15:e0228535.

[Lavine, 1992] Lavine, M. (1992). Some Aspects of Polya Tree Distributions for Statistical Modelling. *The Annals of Statistics*, 20(3):1222 – 1235.

[Liu et al., 2019] Liu, J., Paisley, J., Kioumourtzoglou, M.-A., and Coull, B. (2019). Accurate uncertainty estimation and decomposition in ensemble learning. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

[Orbanz, 2014] Orbanz, P. (2014). Lecture notes on bayesian nonparametrics. http://www.gatsby.ucl.ac.uk/~porbanz/papers/porbanz_BNP_draft.pdf.

[Rahimi and Recht, 2008] Rahimi, A. and Recht, B. (2008). Random features for large-scale kernel machines. In Platt, J., Koller, D., Singer, Y., and Roweis, S., editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc.

[Ranganath et al., 2014] Ranganath, R., Gerrish, S., and Blei, D. (2014). Black Box Variational Inference. In Kaski, S. and Corander, J., editors, *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, volume 33 of *Proceedings of Machine Learning Research*, pages 814–822, Reykjavik, Iceland. PMLR.

[Rasmussen and Williams, 2005] Rasmussen, C. E. and Williams, C. K. I. (2005). *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press.

[Rudin, 1990] Rudin, W. (1990). *Fourier Analysis on Groups*. John Wiley & Sons, Ltd.

[Schölkopf et al., 2001] Schölkopf, B., Herbrich, R., and Smola, A. J. (2001). A generalized representer theorem. In Helmbold, D. and Williamson, B., editors, *Computational Learning Theory*, pages 416–426, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Shaddick et al., 2016] Shaddick, G., Thomas, M., Jobling, A., Brauer, M., Donkelaar, A., Burnett, R., Chang, H., Cohen, A., Van Dingenen, R., Dora, C., Gumy, S., Liu, Y., Martin, R., Waller, L., West, J., Zidek, J., and Prüss-Ustün, A. (2016). Data integration model for air quality: A hierarchical approach to the global estimation of exposures to ambient air pollution. *Journal of the Royal Statistical Society. Series C: Applied Statistics*, 67.

[Shafer and Vovk, 2008] Shafer, G. and Vovk, V. (2008). A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9(12):371–421.

[Stone, 1974] Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(2):111–147.

[Sutherland and Schneider, 2015] Sutherland, D. J. and Schneider, J. (2015). On the error of random fourier features. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, UAI'15, page 862–871, Arlington, Virginia, USA. AUAI Press.

[Vovk et al., 2005] Vovk, V., Gammerman, A., and Shafer, G. (2005). *Algorithmic Learning in a Random World*. Springer US.

[Zhou, 2012] Zhou, Z.-H. (2012). *Ensemble Methods: Foundations and Algorithms*. CRC Press.