Rainfall modelling with Bayesian tree based models

Simon Philp
201779187

Supervised by Hamish Steptoe (Met Office), Lanpeng Ji and Georgios Aivaliotis.

Submitted in accordance with the requirements for the
module MATH5872M: Dissertation in Data Science and Analytics
as part of the degree of

Master of Science in Data Science and Analytics

The University of Leeds, School of Mathematics

September 2024

UNIVERSITY OF LEEDS

School of Mathematics

FACULTY OF ENGINEERING AND PHYSICAL SCIENCES

Academic integrity statement

I am aware that the University defines plagiarism as presenting someone else's work, in whole or in part, as your own. Work means any intellectual output, and typically includes text, data, images, sound or performance.

I promise that in the attached submission I have not presented anyone else's work, in whole or in part, as my own and I have not colluded with others in the preparation of this work. Where I have taken advantage of the work of others, I have given full acknowledgement. I have not resubmitted my own work or part thereof without specific written permission to do so from the University staff concerned when any of this work has been or is being submitted for marks or credits even if in a different module or for a different qualification or completed prior to entry to the University. I have read and understood the University's published rules on plagiarism and also any more detailed rules specified at School or module level. I know that if I commit plagiarism I can be expelled from the University and that it is my responsibility to be aware of the University's regulations on plagiarism and their importance.

I re-confirm my consent to the University copying and distributing any or all of my work in any form and using third parties (who may be based outside the EU/EEA) to monitor breaches of regulations, to verify whether my work contains plagiarised material, and for quality assurance purposes. I confirm that I have declared all mitigating circumstances that may be relevant to the assessment of this piece of work and that I wish to have taken into account. I am aware of the University's policy on mitigation and the School's procedures for the submission of statements and evidence of mitigation. I am aware of the penalties imposed for the late submission of coursework.

Name Simon Philp

Student ID 201779187

Abstract

In this paper we look to solve the problem of a proliferation of atmospheric datasets concerning extreme precipitation throughout Nepal, which results in a lack of a consistent scientific message for policymakers. This is done through a data blending framework which utilises Bayesian Additive Regression Trees (BART) in order to produce a spatially consistent model which can accurately estimate extreme precipitation for any location. We find such a blended model offers increased out-of-sample performance against other competing models such as Random Forests and, due to its Bayesian framework, also offers a suitable representation of the uncertainty of each estimate. Furthermore, we find that such a model offers similar results to previous research and that its estimation of extreme events compared to that of a single dataset highlights the need for policymakers to consider a blended solution incorporating several sources in order to make reliable decisions.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 The Problem

With the threat of climate change increasing each year, the need for reliable and consistent data to inform policymakers around the globe is ever increasing. However, due to the very nature of data concerning the climate, such a consistent message is often hard to find.

Considering precipitation measurements for example, historically, in-situ readings, such as those from rain gauges, have formed the basis of our understanding and given a good description of how precipitation might have changed since records began. However, although such data may be seen as the closest to 'ground-truth', it is limited due to the logistical constraints of such physical measurements, with rain gauges for example being unable to offer a consistent coverage around the globe. Also, even for locations where coverage is possible, such data often offers a limited spatial resolution, with each measurement site covering a large range. This results in a lack of consistency and often an insufficient resolution of data for our needs.

An alternative to this can be provided by data supplied through remote sensing such as via satellites. Given in a gridded format, such data offers a higher temporal and spatial resolution, with measurements being taken in a consistent format across the globe. However, such data can be prone to bias and random errors, due to the discrepancy between observations recorded from space and those taken at ground level. Furthermore, although given at a higher spatial resolution, due to the gridded nature of this data a full spatial coverage is still unavailable with recordings from specific locations such as meteorological sites often being unavailable, making comparisons between the two methods troublesome.

Finally, by combining historical data and climate model outputs, reanalysis products offer a further source of climate data, often at a high temporal and spatial resolution. Constituting a huge part of modern climate research, such products are being produced with increasing accuracy and have been greatly utilised in all parts of the world. However, as such data is the output of modelling rather than actual data itself, such products can still be prone to bias, especially when used to estimate extreme events. In particular, in relation to extreme precipitation events, Rhodes et al. [2015] performed a thorough analysis of reanalysis products for England

and Wales and found that such products only had hit-rates (defined as an event above the 98th percentile) of approximately 40-65% for extreme precipitation events.

Hence, for a given atmospheric phenomena, such methods or a combination of such methods can offer a wide array of different datasets to choose from. For a given policymaker, this can be extremely problematic with this lack of a consensus result failing to support informed decision making. Additionally, out of such datasets there is no clear optimal choice, with different methods often suiting different problems. How then might a policymaker go about making informed decisions from this proliferation of datasets?

1.2 Research Aims

Providing the inspiration for our research, such an issue was considered by Steptoe and Economou [2023], who specifically investigated the estimation of daily maximum precipitation (RX1day) throughout Nepal for a given year using a range of conflicting datasets. This work is particularly important as, due to the monsoon season between June and September every year, heavy rainfall can lead to several extreme precipitation events, ensuring data analysis has an important bearing on any potential decision-making in the area.

To solve this proliferation of atmospheric datasets, the authors proposed combining all datasets using a data blending framework based on Generalized Additive Models (GAMs) in order to produce combined estimated RX1day values, more suited to inform reliable decision making. Using this, interestingly the authors noted how estimates from a single dataset can seemingly misrepresent extreme precipitation events when compared to estimates given by the blended model which factor in information from several different sources. This was especially prevalent for more extreme events, which is particularly concerning for new enterprises such as building a new dam as such extreme events are the incidents that need to be most accounted for.

Such a blended model also provided a predictive distribution for each estimate which suitably summarised all uncertainty present in each constituent dataset. When a heuristic measure to identify a best dataset doesn't exist, they showed this is extremely important in order to help provide a well-informed and consistent message to policymakers, that incorporates all information available.

Taking inspiration from such work therefore, in this paper we aim to investigate an alternative data blending solution to this specific problem by considering modelling the data instead via Bayesian Additive Regression Trees (BART). Through this it is hoped the added uncertainty incorporated within the Bayesian framework is well suited to the problem at hand, producing a spatially consistent model which gives reliable estimates of Annual RX1day values throughout Nepal in a way that incorporates all information available. By comparing our results to previous work, such further modelling will allow us to gain a deeper understanding of the data itself and the suitability of BART in providing a data blending solution.

To this aim therefore, this report is organized as follows. In Chapter 2 we present a summary of the Bayesian CART model which provides a basis for the BART model which we introduce in Chapter 3. Having introduced our modelling framework, in Chapter 4 we next introduce the data we will be working with, before detailing previous work done with this data by Steptoe and Economou [2023] in Chapter 5. Then, in Chapter 6 we investigate initial results from the modelling on BART on each dataset individually, before finally evaluating the suitability of BART to give a blended solution in Chapter 7, comparing such results to previous work. Chapter 8 then concludes with an overview and discussion of our work.

Chapter 2

Bayesian CART

Given a classification or regression problem, we often use a tree-based approach to represent relationships in the data. In particular, introduced by Breiman et al. [1984], a popular choice is given through Classification and Regression Tree (CART) models, which offer a flexible and easy to interpret method that can be applied to a range of different problems.

Such models use a greedy algorithm to iteratively partition the predictor space into smaller and more homogeneous subsets. However, this greedy algorithm does have its drawbacks, with a locally optimistic approach limiting the full exploration of the tree space. This can result in a failure to capture relationships on a more global scale, which is especially problematic when we wish to quantify the uncertainty of our predictions.

There exists several different tree-based models that offer there own alternatives, such as XGBoost (Chen and Guestrin [2016]) and Random Forest (Breiman [2001]), which both use ensemble methods to produce improved results. However, in this chapter we will instead focus on the Bayesian CART approach introduced by Chipman et al. [1998], which aims to consider a tree-based approach from a Bayesian viewpoint.

Simultaneously solving the greediness and uncertainty problem at once, in essence such a model involves the specification of a prior on the tree space that induces a posterior distribution more likely to find 'better' trees. This has been shown to offer improved results on a wide array of problems and enables posterior inference to be performed on potential trees.

In this chapter therefore, we will give a detailed overview of such an approach, and the advantages such a model can give to our problem. However, before we do this, it is important to first present a brief summary of the general structure of CART, with it acting as a building block to later models.

2.1 Classification and Regression Trees

As specified by Breiman et al. [1984], a CART model aims to predict a response variable y given a set of observations $\mathbf{x} = (x_1, x_2, \ldots, x_p)$. To do this, we define a CART model by two

main components:

- A tree T with b terminal nodes.
- A parameter space $M = (\mu_1, \mu_2, \ldots, \mu_b)$.

The tree T, is composed of a series of binary splitting rules present at each internal (non-terminal) node. This has the effect of partitioning the predictor space into subsets such that each observation x is filtered down the tree until it reaches a terminal node (leaf node), where there are no more splitting rules.

When such a terminal node is reached, y is then estimated through a unique parameter, μ_j, which corresponds to the value from M which is assigned to that node. Such a tree is called a regression or classification tree depending on whether the response variable y is qualitative or quantitative respectively.

In the case of a regression tree, analogous to linear regression, our model may therefore take the form

$$y = g(x|T, M) + \epsilon, \qquad \epsilon \sim N(0, \sigma^2) \tag{2.1}$$

where $g(x|T, M)$ denotes the prediction of y given by the tree T and parameter space M, and ϵ represents an error term, distributed independently of (T, M).

In order to facilitate further analysis, we often assume our observations y are i.i.d for each terminal node and independent across such nodes.

2.1.1 Splitting Rules

As previously stated, to subdivide the predictor space there exists a series of binary splitting rules present at each internal node. Such splitting rules are determined by two factors:

- The predictor variable for the split to be performed on.
- For continuous predictors, the split value s such that each observation is assigned according to $\{x_i \leq s\}$ or $\{x_i > s\}$, or, for categorical predictors, the subset C such that each observation is assigned according to $\{x_i \in C\}$ or $\{x_i \notin C\}$.

2.1.2 Example

To further highlight such ideas, we consider the tree given in Figure 2.1. As we can see the tree consists of 3 splitting rules, one for the categorical variable x_1 and two for the continuous variable x_2.

Given such a tree, let us consider a particular observation $\mathbf{x} = (C, 7.5)$. As $x_1 \notin \{A, B\}$ and $x_2 > 5$, such an observation is partitioned into the terminal node corresponding with $\mu_4 = 1$. Assuming a variance of 3, we would therefore model y as coming from the $N(1, 3)$ distribution.

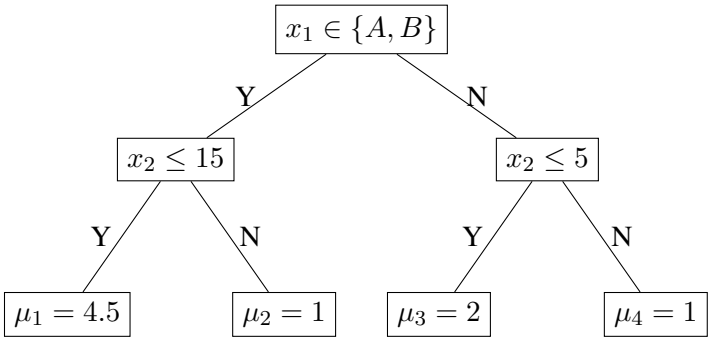


Figure 2.1: A Regression Tree example for $\mathbf{x} = (x_1, x_2)$.

For many CART models, splitting rules and the parameter space M are often chosen within the training of our model through a greedy algorithm, in order to minimise certain optimality criteria such as Mean Square Error or Gini Impurity. However, as we are working in a Bayesian framework, for Bayesian CART models such parameters are instead assumed non-fixed and are specified a prior distribution detailing our beliefs about such values.

2.2 Specification of Priors

Given data therefore, the CART approach uses a greedy algorithm to choose an optimal tree based on the chosen selection criteria. However, as mentioned previously, such an algorithm limits the exploration of the full tree space and fails to incorporate any uncertainty in our estimates. Hence, we would instead like to consider a posterior on the set of potential trees.

This will give us a detailed understanding of the distribution of other trees available to us and therefore allow us to make more informed decisions. In addition to this, unlike for CART models, where the depth of the tree needs to be specified before training, a Bayesian CART approach allows our model to learn from the data and consider tree depths which are most suited to our problem.

Considering this then, in order to represent our initial beliefs we first need to specify a prior on our parameter and Tree space (M, T), remembering to also include a prior for the variance of our error term, σ. Fortunately, we note that this may be simplified as

$$p(M, T, \sigma) = p(M, T)p(\sigma) = p(M|T)p(T)p(\sigma) \tag{2.2}$$

As M is naturally conditioned on a given tree T, this has the advantage of allowing us to specify each component individually, including the less straightforward prior $p(T)$ independent of M. In the following sections, we will go into detail on such specifications.

2.2.1 Tree Prior

First, considering the prior on our tree space, rather than specifying a closed-form expression we instead define $P(T)$ through the use of a tree-generating stochastic process. This is done iteratively as follows:

1. Start with a single node η.

2. Choose to perform a binary split on the node with probability $p_{SPLIT}(\eta)$. If we choose not to split, then this node becomes terminal and is to be ignored in future iterations.

3. If a split is to be performed we assign a splitting rule ρ in accordance with the probability distribution $p_{RULE}(\rho|\eta)$ and assign left and right children nodes accordingly.

4. Steps 2 and 3 are repeated with all available nodes until there exists no more non-terminal nodes to iterate upon or there exists no more non-trivial splits.

Through this process we can consider each generated tree T as an instance from the prior $P(T)$, with explicit probabilities being able to be calculated straightforwardly under many specifications.

2.2.2 Choice of parameters

As our prior $p(T)$ is solely dependent on $p_{SPLIT}(\eta)$ and $p_{RULE}(\rho|\eta)$, our choice in such parameters can greatly effect the trees we are likely to see.

Determining size

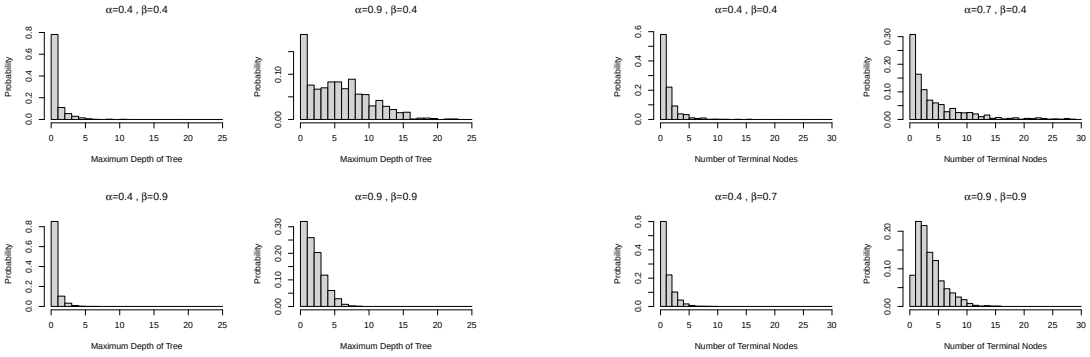
First, considering $p_{SPLIT}(\eta)$, an obvious choice is to let $p_{SPLIT}(\eta) = \alpha$ with $\alpha \in (0, 1)$. However, this has the undesirable effect of splits being equally likely for any depth d of the tree, often resulting in excessively long and straggly trees. Hence, instead we often consider

$$p_{SPLIT}(\eta) = \alpha(1 + d)^{-\beta}, \qquad \alpha \in (0, 1), \beta \in [0, \infty) \tag{2.3}$$

Such a decreasing function ensures that as our tree gets deeper, the probability of further splits diminishes; allowing us to create more desirable trees.

A nice feature of this approach is that, through a suitable choice of (α, β), we can tailor the size and shape of models more likely to be given by the posterior distribution. This gives us great freedom to downplay unwanted characteristics we may not want to see in our final trees.

To highlight this, in Figure 2.2a and Figure 2.2b we use simulations to highlight how the maximum depth and number of terminal nodes differs for different values of α and β.



(a) Difference in maximum depth.

(b) Difference in number of terminal nodes.

Figure 2.2: Plots highlighting the change in structure and shape of trees given by $p_{SPLIT}(\eta)$ as we vary α and β.

Determining splitting rules

Secondly, we consider the probability distribution $p_{RULE}(\rho|\eta)$ which defines the splitting rule ρ.

As mentioned in the previous section, such splitting rules consist of two parts:

- The parameter to split on.
- The splitting value s or subset C to define the split.

Considering the parameter x_j to split on, an obvious choice is to choose x_j uniformly from the set of predictors $\mathbf{x} = (x_1, x_2, \ldots, x_q)$, implicitly assuming that each variable is equally likely to be effective.

Next, given a parameter x_j, by taking all the values of this parameter present in the observed data, we can apply similar ideas to uniformly choose from these values to attain the splitting value s or subset C. Strictly speaking, this method breaks the Bayesian paradigm as we are specifying our prior based on observed data, but nonetheless it is often used as it has the appealing feature of being invariant to monotone transformations of the predictors.

Similarly to $p_{SPLIT}(\eta)$ we can also tailor such methods to make our posterior distribution favour more appealing models given a priori information. For example, if there exists a variable we believe has more of an impact than the rest of the predictors we can apply weights to our uniform distribution accordingly, resulting in the variable being present more often in our splitting rules. The same can also be said to areas of the data that define our splitting value s or subset C. Overall, this gives us a great amount of freedom in tailoring our results to each problem.

2.2.3 Parameter Priors

Next, let us consider the specification of our parameter priors, $p(M|T)$ and $p(\sigma)$.

To ease the computation of our posterior distribution, these are chosen such that may we may marginalise out M in order for us to obtain

$$p(Y|T,\sigma) = \int p(Y|M,T,\sigma)p(M|T,\sigma)dM \tag{2.4}$$

Assuming independence across terminal nodes, as detailed by Chipman et al. [1998] this is often done by choosing conjugate priors as follows:

$$\mu_1,\ldots,\mu_b|T \quad i.i.d \quad \sim \; N(\mu_\mu,\sigma_\mu^2) \tag{2.5}$$

$$\sigma^2 \sim IG(\frac{\nu}{2},\frac{\nu\lambda}{2})(\Leftrightarrow \frac{\nu\lambda}{\sigma^2} \sim \mathcal{X}_\nu^2) \tag{2.6}$$

In choosing the hyperparameters, $(\nu,\lambda,\mu_\mu,\sigma_\mu)$, we may make use of the observed Y values to guide ourselves to realistic choices. For example, choosing (μ_μ,σ_μ) such that each μ is likely to fall within (y_{min},y_{max}).

As before, we may also alter such hyperparameters in order to tailor the results to our needs. In particular, considering σ, we may use overestimates such as the sample standard deviation s^* to choose (ν,λ), in order to attain a desired size and shape of our distribution.

2.3 Posterior Distribution

After the specification of a suitable prior and observation of data, importantly our Bayesian framework induces a posterior distribution:

$$P(T,M,\sigma|Y) \tag{2.7}$$

We note that this may also be written as:

$$P(T,M,\sigma|Y) = P(\sigma|Y)P(T|\sigma,Y)P(M|T,\sigma,Y) \tag{2.8}$$

To draw from such a distribution, we first note that we can draw from $P(\sigma|Y)$ via an inverse gamma distribution (using results from Chipman et al. [2010]) and so σ may be extracted by common methods.

Next, with our conjugate priors making equation 2.4 analytically feasible, we may compute the posterior upon the tree space $P(T|\sigma,Y)$ up to a normalizing constant:

$$P(T|Y,\sigma) \propto P(Y|T,\sigma)P(T) \tag{2.9}$$

However, although this can be used to evaluate individual trees, due to the sheer number of potential trees in the tree space, it is unfeasible to evaluate such a distribution as a whole. Hence,

we are unable to compute the normalizing constant or posterior probabilities which will allow us to perform effective posterior inference.

To solve this, we will therefore use Monte-Carlo Markov Chain (MCMC) methods, which are often used to approximate probability distributions for which direct sampling proves difficult. In particular, we consider the Metropolis-Hastings algorithm.

2.3.1 Metropolis-Hastings

Named after methods introduced by Metropolis et al. [1953] and Hastings [1970], such an algorithm produces a Markov chain sequence of samples which converges in distribution to the probability distribution we wish to approximate. Given the current sample x_t at time t, this is done by considering a new sample $x^{'}$ drawn from a proposal distribution $q(x^{'}|x_t)$. Such a proposed value is then accepted or rejected dependent on the probability distribution at that point.

In particular, for our posterior $P(T|Y,\sigma)$ the algorithm works explicitly as follows:

1. **Initialise**: Choose an initial tree T^0, set $t = 0$ and choose a proposal distribution $q(T^*|T^t)$.
2. **Iterate**:
   - Propose a new candidate tree T^* from the proposal distribution $q(T^t|T^*)$.
   - Compute the acceptance probability

$$\alpha(T^t, T^*) = \min\left\{\frac{q(T^t|T^*)}{q(T^*|T^t)}\frac{P(T^*|Y,\sigma)}{P(T^t|Y,\sigma)}, 1\right\} \tag{2.10}$$

     where $\frac{P(T^*|Y,\sigma)}{P(T^t|Y,\sigma)} = \frac{P(Y|T^*,\sigma)P(T^*)}{P(Y|T^t,\sigma)P(T^t)}$ as in equation 2.9.
   - Generate a random uniform number $u \in [0,1]$ and
     - Set $T^{t+1} = T^*$ if $u \leq \alpha(T^t, T^*)$
     - Set $T^{t+1} = T^t$ if $u > \alpha(T^t, T^*)$

Importantly, this sequence of samples converges to the true posterior (Tierney [1994]), and hence due to calculations performed in equation 2.10, such an algorithm allows us to draw samples from the posterior distribution without computing the normalizing constant.

Proposal Distribution

Considering such an algorithm for our problem then, for a given tree T^t, the proposal distribution $q(T^*|T^t)$ is defined by randomly altering the tree to T^* through one of the four following steps

- **GROW:** Randomly choose a terminal node and assign a new splitting rule according to $p_{RULE}(\rho|\eta)$.

- **PRUNE:** Randomly choose a parent of two terminal nodes and collapse the split.

- **CHANGE:** Randomly pick an internal node and choose a new splitting rule according to $p_{RULE}(\rho|\eta)$.

- **SWAP:** Randomly choose an internal node and swap the splitting rules of its two children nodes. If they are the same then swap the rule of each children node with the rule of the parent node.

Such a proposal distribution is chosen as it has many benefits in reducing computation times of the iteration given by the algorithm. In particular, we note $q(T^*|T^t)$ and $q(T^t|T^*)$ are equivalent for both the **CHANGE** and **SWAP** steps, hence reducing the calculation of (2.10) as $\frac{q(T^t|T^*)}{q(T^*|T^t)}$ is equal to 1. Also, due to the use of $p_{RULE}(\rho|\eta)$, both the **GROW** and **PRUNE** steps offer a reduction in calculations due to a cancelling out of terms used in the calculation of $P(T^t)$ in Equarion (2.10).

2.3.2 Evaluation

Using MCMC methods then, we may sample trees from the posterior $P(T|Y,\sigma)$. For each sample and for each terminal node, we may then easily draw independently from $P(M|T,\sigma,Y)$ via a normal distribution (Chipman et al. [2010]) in order to consider full samples from our posterior distribution $P(T,M,\sigma|Y)$.

Due to the nature of the Metropolis-Hastings algorithm, we are more likely to see so called 'better' trees that have a high probability in relation to $P(T|Y,\sigma)$. However, experiments by Chipman et al. [1998] showed that such an algorithm is prone to get 'stuck' in a local space of trees, with potential trees not radically changing much over several hundred iterations. This can be seen as a result of the algorithm slowly moving locally around many of the different peaks in a multimodal posterior distribution.

As such an algorithm converges, it will eventually evaluate all of the potential tree space, but in order to reduce computation times it has been found necessary to restart the algorithm several times such that it may explore different regions of our posterior more quickly. This is a particular downside to the model at hand, with several runs being needed to produce adequate results.

Having produced many samples from such a posterior, there also exists many different ways in which we may choose to take estimates from our given model. One popular example however, is to choose 'good' trees by comparing marginal likelihoods $P(Y|X,T)$ and to predict from such. Alternatively, inspired by model averaging (Oliver and Hand [1995]) we could use weights proprtional to $p(Y|X,T)P(T)$ for the estimate from each tree in order to approximate the posterior mean.

2.3.3 Alternatives

Although such analysis proposed by Chipman et al. [1998] is the basis of our study, it is important to note that this work was not the first to consider tree-based models from a Bayesian perspective. In particular, Buntine [1992] proposed a Bayesian analysis of a tree-based model by using a deterministic rather than stochastic prior on the tree space. However, such work was limited to classification trees unlike work discussed in this chapter.

Also, as discussed by Malehi and Jahangiri [2019], publishing of Chipman et al. [1998] coincided with a near identical study of Bayesian tree-based models by Denison et al. [1998] . They instead differed by considering a prior distribution defined over the splitting node (S), splitting variable (V), splitting rule (R), tree size (κ) and parameters in terminal node distributions (Ψ), with the prior on κ in particular helping to reduce overfitting of our model by avoiding excessively long trees.

Furthermore, there exists numerous alternatives to the proposal distribution suggested for use in our Metropolis-Hastings algorithm, for example by Wu et al. [2007]. They proposed including an additional RESTRUCTURE step which alters the structure of the tree by a large amount. Appealingly, this has the effect of reducing computation time of our algorithm as such a large change in structure removes the need to continuously restart our algorithm as it no longer gets 'stuck' in areas of the tree space.

Chapter 3

BART

In this chapter we extend the ideas introduced in the last chapter by considering such methods applied to a sum-of-trees model. This allows for the modelling of more complex relationships with improved performance.

3.1 Motivation

Introduced in the last chapter, the Bayesian CART model offers a useful way in which we can utilise tree-based models to perform statistical inference. However, due to its single tree structure it can often result in a failure to capture more complex relationships found by alternative and competing models, limiting its use to providing a search of the tree space which is likely to find 'good' trees.

Such alternative models often consider an ensemble of trees, which, by combining the strengths of several different trees, can often drastically outperform single tree-based methods, offering improved accuracy and robustness to new data. In particular, two such ensemble methods are boosting and bagging introduced by Freund and Schapire [1997] and Breiman [1996] respectively.

For such, boosting involves using a collection of trees which each aim to explain the variation of a different part of the data through low-order interaction effects. Denoted as 'weak' learners, these trees are then combined, allowing the model to more accurately fit the data as a whole. On the other hand, bagging involves using a collection of trees trained on different samples of the data, helping to improve the robustness of the model and avoid the dangers of overfitting.

Hence, taking inspiration from such ensemble methods, Chipman et al. [2010] extended previous work done on Bayesian CART by proposing a new model called Bayesian Additive Regression Trees (BART). Such a model instead aims to carry out statistical inference through a sum-of-trees model which, like boosting, uses multiple weak learners that each aim to explain relationships in different parts of the data. However, differently to boosting, BART enforces such weak learners using ideas from the Bayesian CART model, by including the specification

of a regularisation prior upon the tree space. Each component tree is then created by taking successive draws from a Bayesian backfitting MCMC algorithm similarly to as described in Chapter 2.

Improving on previous research, such work is attractive as it offers a much higher performing model than Bayesian CART, whilst still keeping the desired uncertainty encapsulation given through a full posterior distribution on the tree space. Hence, to understand more about such work, let us therefore first expand on the structure of the model itself.

3.2 Model structure

For a continuous y, as for Bayesian CART we wish to use a function $f(x)$ to model the relationship between y and x:

$$y = f(x) + \epsilon, \qquad \epsilon \sim N(0, \sigma^2) \tag{3.1}$$

However, unlike in equation 2.1, we instead express $f(x)$ through a sum of m trees such that

$$y = \sum_{j=1}^{m} g(x|T_j, M_j) + \epsilon, \qquad \epsilon \sim N(0, \sigma^2) \tag{3.2}$$

where $g(x|T_j, M_j)$ is a function which gives the i-th parameter $\mu_{ij} \in M_j$ which is assigned to x for the j-th tree T_j.

For such a model, y is therefore given as a sum of m different parameters u_{ij} for each value of x, allowing the incorporation of additive effects into our model similarly to other models such as Generalized Additive Models (GAMs). However, unlike GAMs, as we will see later on, due to the flexibility of each tree in this sum, BART models can more easily incorporate main and interaction effects of varying orders depending on the size and structure of each tree. This allows BART to model more complex relationships.

3.3 Regularisation Prior

Along with the sum-of-trees model, the second main feature of BART is a regularisation prior, which enforces each tree to be a 'weak' learner of the data. This structure is advantageous as it allows for the modelling of complex relationships without using computationally demanding individual trees which are prone to overfit to the data, but rather a collection of simple trees which together form a strong model.

Notably, unlike for boosting, in our BART model such a regularisation is enforced through the specification of the prior imposed on each tree through the Bayesian framework introduced in Chapter 2. To show this, let us first consider the full prior on our sum of trees model:

$$P((T_1, M_1), \ldots, (T_m, M_m), \sigma) \tag{3.3}$$

Fortunately, similarly to the Bayesian CART model, we assume independence between the variables for each tree and also between the trees themselves so that we may greatly simplify this prior as follows:

$$\begin{aligned} P((T_1, M_1), \ldots, (T_m, M_m), \sigma) &= P(\sigma) \prod_{j=1}^{m} P(T_j, M_j) \\ &= P(\sigma) \prod_{j=1}^{m} P(T_j) P(M_j | T_j) \end{aligned}$$

Importantly, this allows us to specify each prior separately in order to ease the complexity of our work. Considering this, let us now specify each component prior in turn.

3.3.1 Tree priors

Firstly, considering the priors $P(T_j)$ for our BART model, for each tree we simply use the specification as given in Chapter 2, with its nature proving cohesive with MCMC calculations for our posterior distribution. However, to ensure regularization of our trees, we carefully choose the values α and β as we recall their importance on the size and structure of each tree.

In particular, values $\alpha = 0.95$ and $\beta = 2$ are used as they will mainly restrict our model to trees with less than 5 terminal nodes. Fortunately however, unlike for boosting where the tree depth is fixed, the Bayesian nature of our model allows for longer trees to be grown if suggested by the data.

3.3.2 M priors

Next, like in Chapter 2, we assume independence of the parameters μ_{ij} across each node for each tree, allowing a general specification for $P(\mu_{ij}|T_j)$. In order to simplify posterior computation, this is again chose such that for each tree

$$\mu_{1j}, \ldots, \mu_{bj} | T_j \quad i.i.d \quad \sim N(\mu_\mu, \sigma_\mu^2) \tag{3.4}$$

with μ_m and σ_μ again chosen to produce reasonable results in accordance with the observed data. However, in our BART model there is a slight change in the importance of the specification of such hyperparameters.

Considering this in more detail, we note that, under such priors, our sum-of-trees model, $f(x)$, has an induced prior which is normally distributed according to $N(m\mu_\mu, m\sigma_\mu)$ for each observation y. Hence, in order for our model to behave reasonably to the data, for our BART model we choose μ_m and σ_μ such that for a given k:

$$m\mu_\mu - k\sqrt{m\sigma_\mu} = y_{min} \tag{3.5}$$

$$m\mu_\mu + k\sqrt{m\sigma_\mu} = y_{max} \tag{3.6}$$

Such a k value acts as an additional hyperparameter for our prior, determining the extent to which each μ_{ij} is likely to fall within (y_{min}, y_{max}). Although this can be chosen optimally through cross-validation, often we use a default value of $k = 2$, indicating the likelihood of such an occurrence at 95%.

However, this is not all, as before training our BART model we also choose to shift and scale our y values such that all our data lies in the interval $(-0.5, 0.5)$. Returning to the simultaneous equations in Equations (3.5) and (3.6), we note that for each tree we now have

$$\mu_{1j}, \ldots, \mu_{bj}|T_j \quad i.i.d \quad \sim N(0, \sigma_\mu^2) \tag{3.7}$$

where $\sigma = \frac{0.5}{k\sqrt{m}}$.

With $E(\mu_{mu}) = 0$ for each μ_{ij}, this has the desired effect of weakening each tree by limiting the effect any individual can have on the full sum-of-trees model as a whole. As m is increased, due to a shrinkage of σ_μ such an effect is only increased, ensuring the desired nature of our regularisation prior.

3.3.3 σ prior

Finally, identically to Bayesian CART, we choose $P(\sigma)$ such that

$$\sigma^2 \sim IG(\frac{\nu}{2}, \frac{\nu\lambda}{2}) (\Leftrightarrow \frac{\nu\lambda}{\sigma^2} \sim \mathcal{X}_\nu^2) \tag{3.8}$$

with values (ν, λ) estimated from the data.

3.4 Posterior Distribution

After observation of the data, such priors therefore induce a posterior distribution upon our parameter space:

$$P((T_1, M_1), \ldots, (T_m, M_m), \sigma|y)$$

Due to the sheer size of such a distribution, drawing samples directly from our posterior is unfeasible and so instead we make use of another MCMC algorithm called Gibbs sampling. Such an algorithm allows the sampling from high-dimensional distributions through the consideration of several lower order conditional distributions which are much easier to work with.

Specifically, for our posterior, let us denote $T_{(j)}$ as all the trees but the j-th tree T_j, and $M_{(j)}$ similarly. Our Gibbs sampler therefore works by taking m successive draws from:

$$(T_j, M_j)|T_{(j)}, M_{(j)}, \sigma, y \qquad j = 1, \ldots, m \tag{3.9}$$

and then a draw of σ from:

$$\sigma|(T_1, M_1), \ldots, (T_m, M_m), y \tag{3.10}$$

Crucially for this algorithm, for each draw we must use the **current** value of each conditioned parameter, ensuring the algorithm updates appropriately.

Introduced by Chipman et al. [2010], the authors discuss how Hastie and Tibshirani [2000] showed that such an algorithm was a stochastic generalization of the backfitting algorithm used for generalised additive models. Hence, they referred to the algorithm as a backfitting MCMC, which we shall denote as such from now.

3.4.1 Backfitting MCMC

To initialise our backfitting MCMC algorithm, we first start off with m single-node trees. Then, we iteratively draw from each conditional until it converges to our desired distribution.

Firstly, as in Bayesian CART, we may draw σ from an inverse gamma distribution using routine methods. Next, we consider draws from each (T_j, M_j) pairs, which proves slightly more tricky.

Importantly however, we note that (T_j, M_j) depends on $T_{(j)}, M_{(j)}, y$ only through

$$R_j = y - \sum_{k \neq j} g(x|T_k, M_k) \tag{3.11}$$

Hence, by considering this partial residual R_j, we can draw from each (T_j, M_j) by taking two successive draws from:

$$T_j|R_j, \sigma$$
$$M_j|T_j, R_j, \sigma$$

Replacing y by the partial residuals, we can draw trees from $(T_j|R_j, \sigma)$ using the MCMC algorithm proposed last chapter. Then, we can again use similar methods to draw each μ_{ij} from a normal distribution as in the last chapter, allowing the partial residuals, R_{j+1}, to be calculated to allow for the next iteration of the algorithm.

Hence, at each iteration of the algorithm each of the m different trees can alter by one move at a time, whether that be growing, shrinking or staying the same. As each tree is built using partial residuals, R_j, conceptually as such an algorithm runs, each tree is changing and altering in order to explain relationships in different parts of the data.

3.4.2 Posterior Inference

Using such an algorithm, we therefore produce a sequence of draws $(T_1^*, M_1^*), \ldots, (T_m^*, M_m^*), \sigma^*$ which converge in distribution to the posterior $P((T_1, M_1), \ldots, (T_m, M_m), \sigma|y)$. Using as such, for each iteration, we can also produce a sequence of samples $f^*(x)$:

$$f^*(x) = \sum_{j=1}^{m} g(x|T_j, M_j) \tag{3.12}$$

This converges to the induced distribution of the so-called 'true' function $f(x)$ from equation 3.1 which we believe models the underlying relationships in the data. Fortunately, unlike for Bayesian CART, such an algorithm doesn't get 'stuck' in parts of the posterior, with the additive nature of our model ensuring restarts of the algorithm aren't needed.

Hence, after a suitable burn-in period needed to stabilise our algorithm, such a sequence of samples $f^*(x)$ can be used to perform a full posterior inference for each value of x. In particular, such a model can be used to produce point estimates of y for a given x by taken the average of all k samples after burn-in:

$$\frac{1}{K} \sum_{k=1}^{K} f_k^*(x) \tag{3.13}$$

In addition to this, posterior uncertainty of $f(x)$ can be determined through the variation in such a sequence, such as using sample kernel density estimates to represent the behaviour of each prediction.

3.5 Use in Literature

Evaluating such work, Chipman et al. [2010] tested BART on a wide variety of different data sets, displaying similar performance compared to several other reference methods such as boosting, random forests and neural nets when using default parameter values. Interestingly, they also highlighted that after tailoring the hyperparameters through cross-validation, the model actually surpassed all other methods when considering out-of-sample performance in the form of RMSE.

Such investigation into the performance of the model has been greatly extended in the years since its introduction, with BART being used in several different research areas encompassing different fields, such as for prediction of credit risk (Zhang and Härdle [2010]) and hospital performance evaluation (Liu et al. [2015]) to name a few.

The theory of BART has also been greatly extended, with a plethora of variations now being available as discussed by Hill et al. [2020]. Such variations include Spatial BART (Müller et al. [2007]) which modifies BART to include spatial information and Soft BART (Linero and Yang [2018]) which incorporates soft decision trees into the general framework in order to produce smoother predictions.

Chapter 4

Data Evaluation

Having introduced our modelling framework, in this chapter we present a full evaluation of the data used by Steptoe and Economou [2023], which has provided the inspiration for our work. Such data will be used to evaluate the suitability of BART models in estimating extreme rainfall throughout Nepal in the remainder of this report.

4.1 Data Sources

We consider 6 datasets previously used by Steptoe and Economou [2023] which each provide values of Annual daily maximum precipitation (RX1day) for a number of years and locations across Nepal.

However, due to the difficulties in achieving consistent recordings in such a sparse and mountainous region of the world, each dataset differs slightly in the methodologies in which they were put together. It is such an issue which forms the basis of our problem and so we first present a brief description on each dataset and their different origins below:

- **Multi-Source Weighted-Ensemble Prediction (MSWEP) v2.8**- From Beck et al. [2019], a global precipitation product which uniquely combines measurements from rain gauges, satellites and reanalyses products to offer high quality precipitation estimates at a high resolution with full global coverage.

- **High Asia Refined analysis (HAR) v2**- From Wang et al. [2020], a regional atmospheric dataset which is generated by dynamically downscaling ERA5 reanalysis data to focus on high mountain Asia.

- **Indian Monsoon Data Assimilation and Analysis (IMDAA) v0.3**- From Rani et al. [2021], a high resolution regional reanalysis model focused on the Indian monsoon region which makes use of both conventional and satellite observations from many different sources.

- **GloSea5**- A seasonal prediction system developed and run by the Met Office (MacLachlan et al. [2015]) which is based on the Met Office climate prediction model, HadGEM3.
- **APHRODITE-2**- From Yatagai et al. [2012], a daily gridded precipitation dataset made from the collection and analysis of in-situ measurements from rain gauges across more than 5000 stations across Asia. An extensive dataset, it covers a period of more than 57 years and will be used as a baseline dataset to evaluate our results.
- **ECMWF Reanalysis v5 (ERA5)**- A global reanalysis dataset from Hersbach et al. [2020] which offers a detailed description of the Earth's atmosphere from 1950 onwards. Such recordings are given hourly with a gridded dataset of 31km spatial resolution which is an improvement on previous models. This will also be used as a baseline dataset to evaluate results.

4.2 Data Structure

As well as providing annual RX1day values, each dataset also includes many other variables which we will use to try and estimate such precipitation recordings. Hence, to understand more about our data, let us first consider the size and structure of each of our constituent data sets through Table 4.1. As we will only be using them as baseline sources, we will not include APHRODITE-2 or ERA5 in such discussion.

Dataset	**No. of Entries**	**No. of Variables**	**Years Recorded**
MSWEP	18,900	7	1979-2020
HAR	9,450	7	2000-2020
IMDAA	18,450	7	1979-2019
GloSea5	259,200	8	1992-2016

Table 4.1: Summary of the datasets we will use for modeling.

Through this we note two key things. Firstly, we see that each dataset has been recorded over slightly different time periods, ensuring a lack of temporal consistency. Due to this, before modelling we might wish to alter our data such that each source includes data from a common time span. This will ensure any temporal relationships do not interfere with our results.

Secondly, the other important thing we note is a big difference in the structure of the GloSea5 dataset compared to the others, with a much larger collection of recordings and an additional variable. To understand why this may be the case we present a summary of the main variables present in these datasets in Table 4.2.

Variable	Type	Datasets	Description
Annual RX1day (mm)	Continuous	All	Detailing the maximum rainfall experienced for each year and location in our data, this is the response variable we wish to estimate through modeling.
Latitude	Continuous	All	The latitude of each recording.
Longitude	Continuous	All	The longitude of each recording.
Year	Continuous	All	The specific year of each recording.
Realisation	Categorical	GloSea5 only	The realisation of the model making up our GloSea5 dataset which led to the specific RX1day value.

Table 4.2: Summary of the main variables included within the data.

Interestingly, we see that the reason for this different structure comes as a result of the addition of a *Realisation* variable, unique to the GloSea5 dataset. This is included as each recording from the GloSea5 dataset comes from the Met Office climate prediction model, which is subject to a certain level of randomness. Hence, for each location and year, the GloSea5 dataset includes 24 realisations of such a prediction model in order to account for the randomness in the model itself.

As this results in a much larger dataset than the others, when we consider blending our data later on, it will be important to account for such size discrepancies to ensure the GloSea5 dataset does not dominate our results.

4.3 Summary of Main Variables

One of the most important aspects of modelling comes through the consideration of which predictor variables we wish to include. To this aim, we next include an analysis of the main variables in our data which are likely to influence estimates of Annual RX1day values throughout Nepal.

4.3.1 Latitude and Longitude

Within our data, the most influential factor in estimating extreme precipitation is through the spatial information given for each data entry, with rainfall norms differing greatly as we move throughout Nepal. Due to a lack of additional topological features such as elevation, this information is encapsulated solely through the latitude and longitude variables detailing the position of each recording in co-ordinate form.

Analysing such variables, we see that, common for each dataset, there in fact exists 450 *Latitude* and *Longitude* pairings across a 18x25 grid at a consistent resolution. Plotted in Figure 4.1 such

recording locations offer full coverage over Nepal as well as parts of neighbouring countries China and India. Such a coverage incorporates a wide variety of different geographical regions characteristic to the area, ranging from lowland plains to the mountains of the Himalayas.

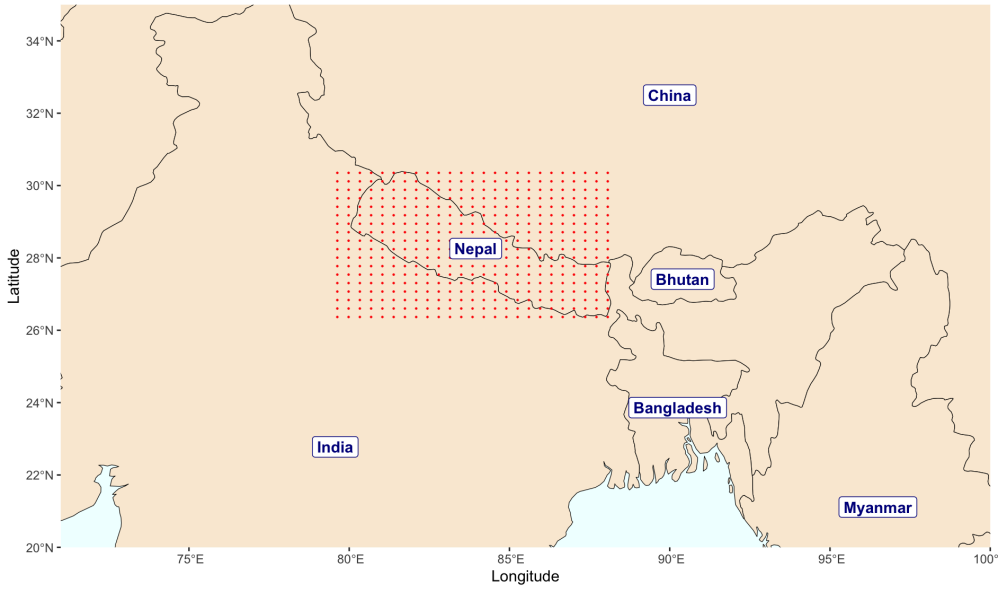


Figure 4.1: Location of the 450 observation sites which are present in our data.

It is important to note that for each dataset, each location also has the same number of recordings, ensuring a high level of consistency in our data.

4.3.2 Year

The only other variable within our data which may prove influential to extreme precipitation recordings comes through the *Year* variable. In fact, due to the known impacts of climate change in the last 20 years, we have especially large reason to suspect extreme precipitation is highly dependent on temporal information present in our data.

As we previously saw in Table 4.1, the range of such a variable differs greatly by dataset. If we were therefore to consider including such a variable in our model, it would be wise to alter our data to ensure each dataset includes data from a common year span. By doing as such we ensure no single dataset alters our model by allowing it to learn temporal relationships not present in all of our data.

However, considering Figure 4.2 , we note that is little evidence that precipitation recordings are dependent on any temporal effects, with there being no clear trend in such plots. Hence, it may be wise to not include such a variable in our models, in order to reduce unnecessary complexity and high computation times.

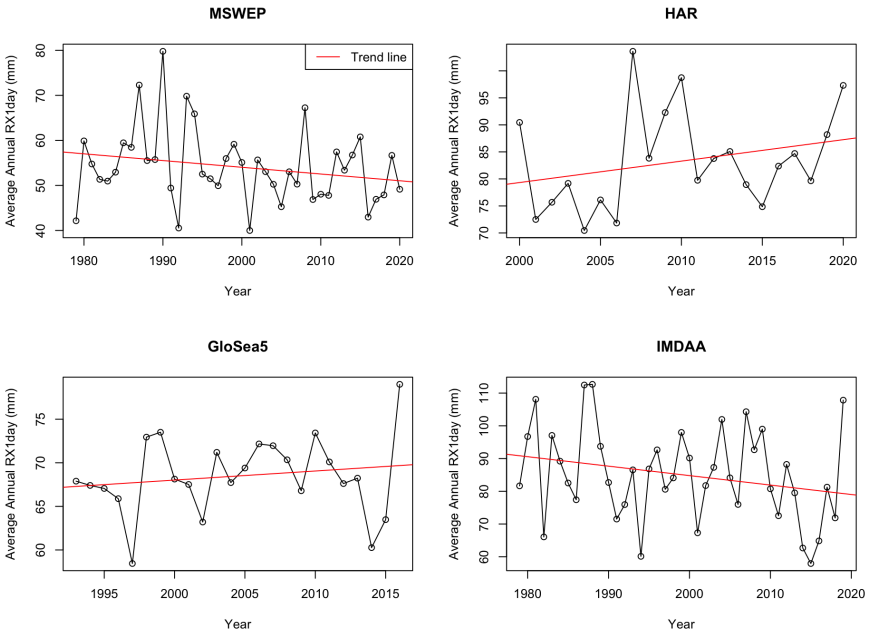


Figure 4.2: Plots showing the average annual RX1day value for each year and for each dataset, with trend lines fitted.

4.3.3 Annual RX1day

The final variable present in our data that we wish to analyse is the response variable that we wish to estimate, Annual RX1day (mm).

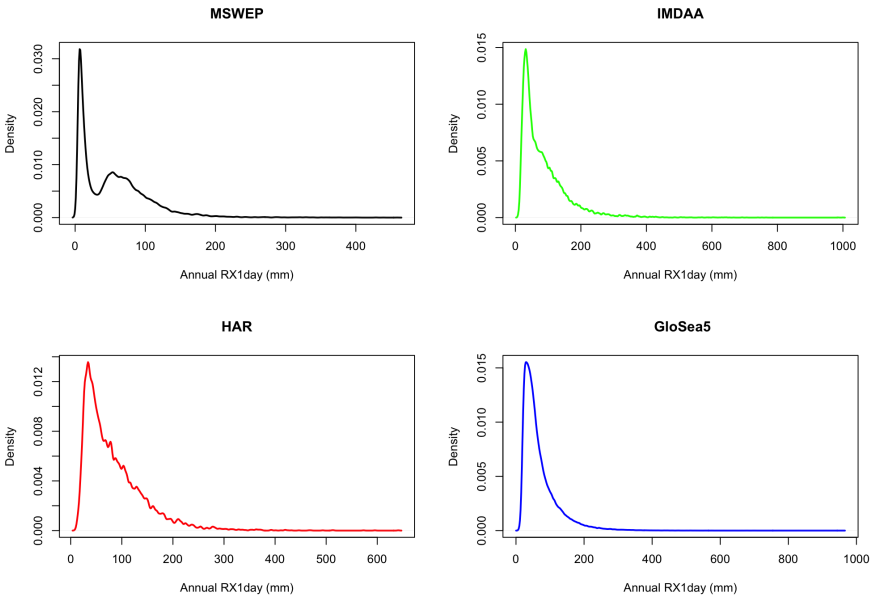


Figure 4.3: Plots showing Annual RX1day values for each dataset.

First of all, for each dataset we amalgamate all observations in order to see the overall variation of extreme precipitation estimates and how they differ. Plotted in Figure 4.3, interestingly we see the extent of such a difference with the MSWEP dataset in particular notably exhibiting a twin-peaked plot.

However, as we are considering the spatial dependency of such values through our modelling, it may be more revealing to consider the difference of such precipitation values across all locations for each dataset. To this aim, in Figure 4.4 we include chloropleth plots of the average annual RX1day value for each location within Nepal and for each dataset, with the red line denoting the outline of Nepal.

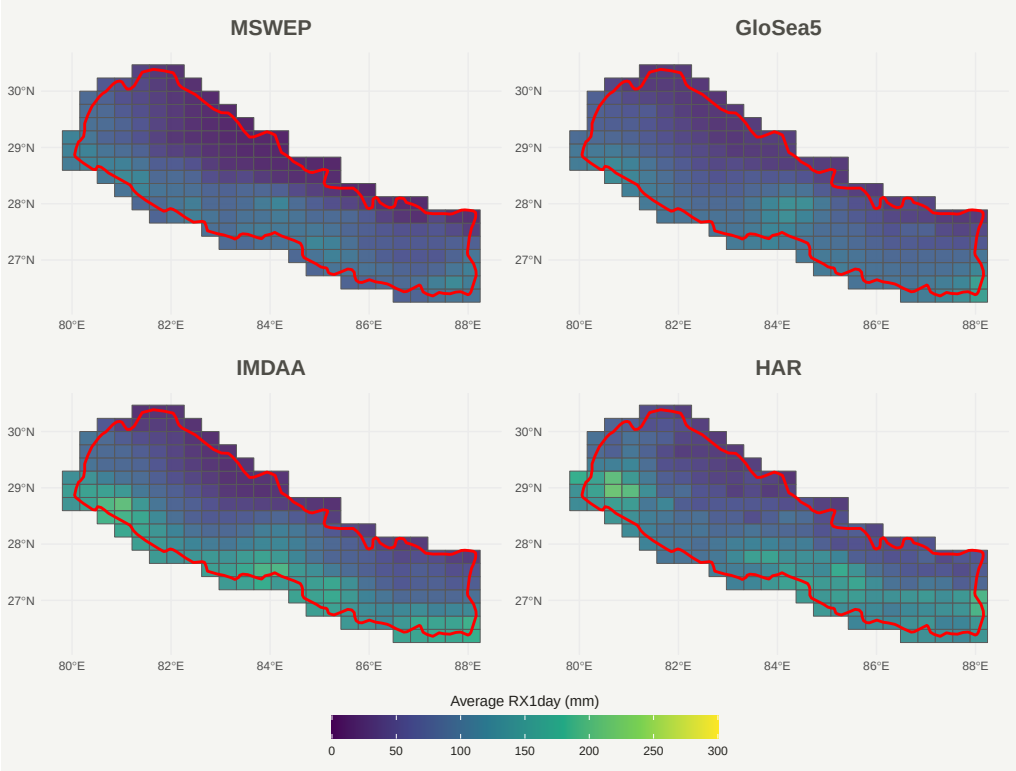


Figure 4.4: Chloropleth plots showing the average annual RX1day value for each location in each dataset, with the outline of Nepal given in red.

Importantly, we see how such plots represent known rainfall trends present in the region, with high values in the South and lower values in the more mountainous North. However, as also found by Steptoe and Economou [2023], each dataset gives varying estimates, with the IMDAA and HAR datasets in particular estimating much more extreme values in the south than the others. This highlights the lack of a consistent message available for policymakers in the region and the need for a reliable blended solution which incorporates all information available into a single model, providing the inspiration for our work.

4.3.4 Plausibility of BART Assumptions

We recall that one assumption of the BART model given in chapter 3 is that the errors are normally distributed. Hence, as we will be using such a model, it would be also useful to check that such assumptions seems reasonable for particular locations in our data.

To do this, let us consider the 4 locations marked in Figure 4.5. Such locations will also be used throughout the remainder of the paper to evaluate different aspects of our modelling and hence are chosen to represent the different precipitation environments present within Nepal.

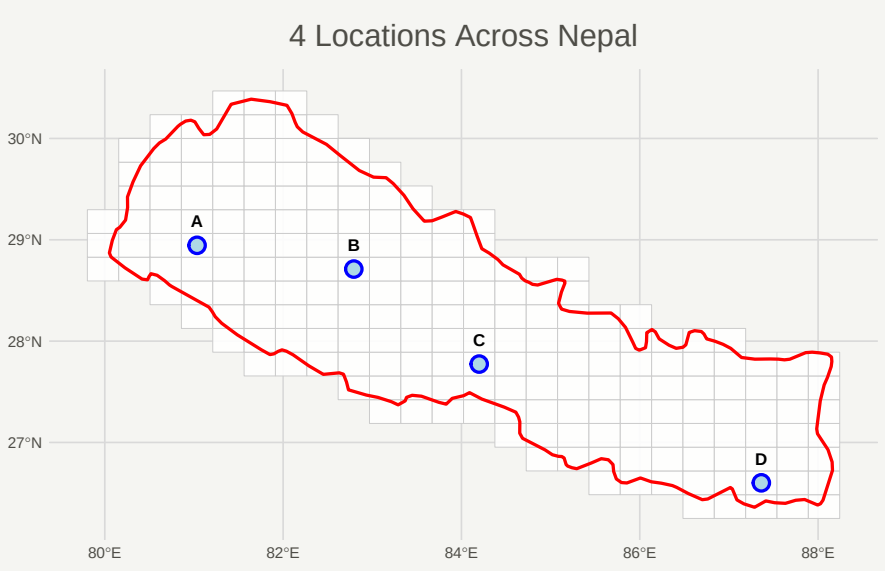


Figure 4.5: Plot showing the location of 4 observational points we will use for further analysis across the next 3 chapters.

Considering these locations then, in Figure 4.6 we plot kernel density estimates given by the MSWEP dataset. As we can see, this normality assumption appears to be violated, with the distinctive bell-shaped not being present for all location points. In particular, for locations A and D we have left-skewed distributions.

To address this therefore, we may consider applying a transformation from the Box-Cox family (Box and Cox [1964]) such that our data is more suited to the assumptions needed for successful modelling with BART. In particular, we consider this in Figure 4.7 by assessing the 4 locations from the previous plot but this time with log transformed data. As we can see the data is now more representative of normally distributed data and hence we may consider performing such a transformation when modelling.

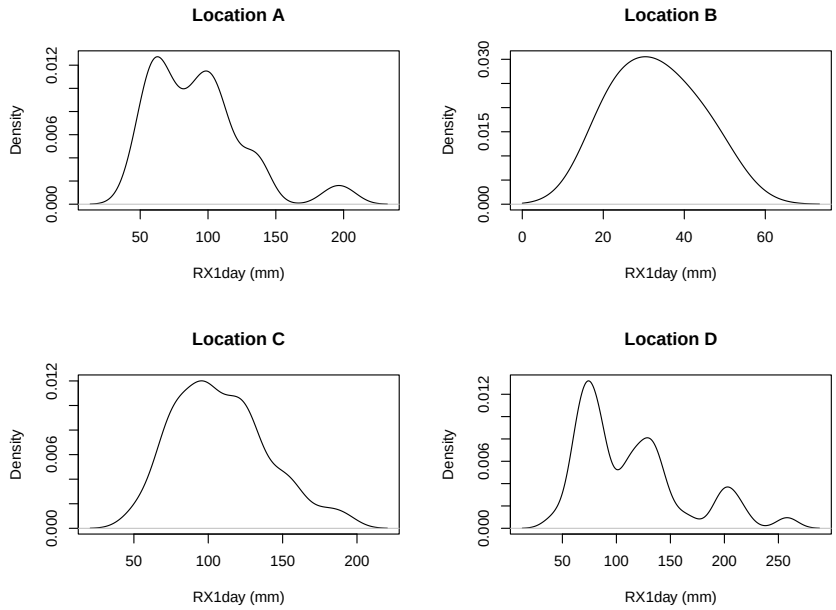


Figure 4.6: Kernel density estimates of annual RX1day values for 4 locations in the MSWEP dataset.

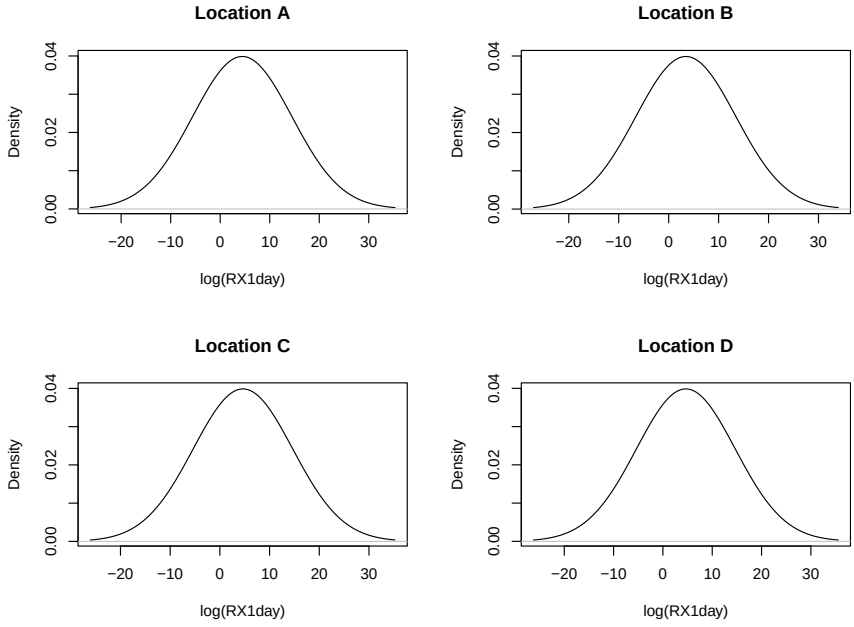


Figure 4.7: Kernel density estimates of log transformed Annual RX1day values for 4 locations from the MSWEP dataset.

4.4 Data Preparation

Lastly, before we can move onto modelling it is important we prepare our data thoroughly for our aims.

Fortunately, as such data has already been preprocessed up to a research standard due to previous work done by Steptoe and Economou [2023], there requires no cleaning of the data. However, there still exists work to be done on the specific contents of our data, with two main aspects needed to be considered.

Firstly, we may wish to exclude any entries which come from the 242 locations outside Nepal that are present in our data, in order to reduce computation time and the complexity of our models. Secondly, we may wish to constrain all of our data to a common time span to ensure consistency in our modelling.

At different stages of modelling, such alterations may or may not be relevant due to a desire to investigate different features of both the data and BART models. However, at some point both alterations will be used and so it is important to prepare for such before training our first model.

Chapter 5

Previous Work

Providing the main motivation for our research, before modelling with BART, in this chapter we first discuss in more detail previous work done by Steptoe and Economou [2023] on the data described in the previous chapter.

5.1 An additive solution

With a number of datasets offering differing estimates to choose from and no clear optimal solution, the problem is therefore well established. To solve this then, as previously mentioned, Steptoe and Economou [2023] consider using a data blending framework based on Generalized Additive Models.

In particular, with $Y_{s,t,m}$ representing the RX1day precipitation maximum for grid s, year t and dataset m, they model $Y_{s,t,m}$ with a Generalized Extreme Value Distribution (GEV) where:

$$Y_{s,t,m} \sim GEV(\mu_{s,t,m}, \sigma_{s,t,m}, \xi_m)$$

$$\begin{aligned}
\mu_{s,t,m} &= \beta_0 + \mathscr{F}(year_t) + \mathscr{G}(lon_s, lat_s) + \mathscr{H}(lon_s, lat_s, u_m^{(\mu)}) + \epsilon_i \\
\log(\sigma_{s,t,m}) &= \gamma_0 + \mathscr{F}(year_t) + \mathscr{G}(lon_s, lat_s) + \mathscr{H}(lon_s, lat_s, u_m^{(\sigma)}) + \epsilon_i \\
\log(\xi_m) &= \delta_0 + u_m^{(\xi)} + \epsilon_i
\end{aligned}$$

Here $\mathscr{F}(\cdot)$, $\mathscr{G}(\cdot)$ and $\mathscr{H}(\cdot)$ are smooth functions estimated in model fitting and β_0, γ_0 and δ_0 are the intercepts for the model.

Analysing such, we note the intercept terms and the functions $\mathscr{F}(\cdot)$, $\mathscr{G}(\cdot)$ are common for each dataset and hence incorporate the data blending element of the model with such elements representing the global relationships present in the data. On the flip side, the terms $u_m^{(\cdot)}$, are dataset specific parameters and allow the individual variability for each dataset to be incorporated in our model.

Importantly for their model, such dataset specific parameters can be integrated out for each GEV parameter such that we may attain a distribution only dependent on the blended elements of the model. Specifically this is done by attaining $\mu_{s,t}$, $\sigma_{s,t}$ and ξ through

$$\mu_{s,t} = \int_{u_{s,m}^{(\mu)}} \mu_{s,t,m} \, du_m^{(\mu)}$$
$$\sigma_{s,t} = \int_{u_{s,m}^{(\sigma)}} \sigma_{s,t,m} \, du_m^{(\sigma)}$$
$$\xi = \int_{u_m^{(\xi)}} \xi_m \, du_m^{(\xi)}$$

such that we may model $Y_{s,t}$ as

$$Y_{s,t} \sim GEV(\mu_{s,t}, \sigma_{s,t}, \xi)$$

Hence, the final GEV distribution can be thought of providing a summary of all the information captured by all the datasets whilst avoiding any singular dataset dominating the results as the variability of each individual dataset has been removed.

5.2 Results

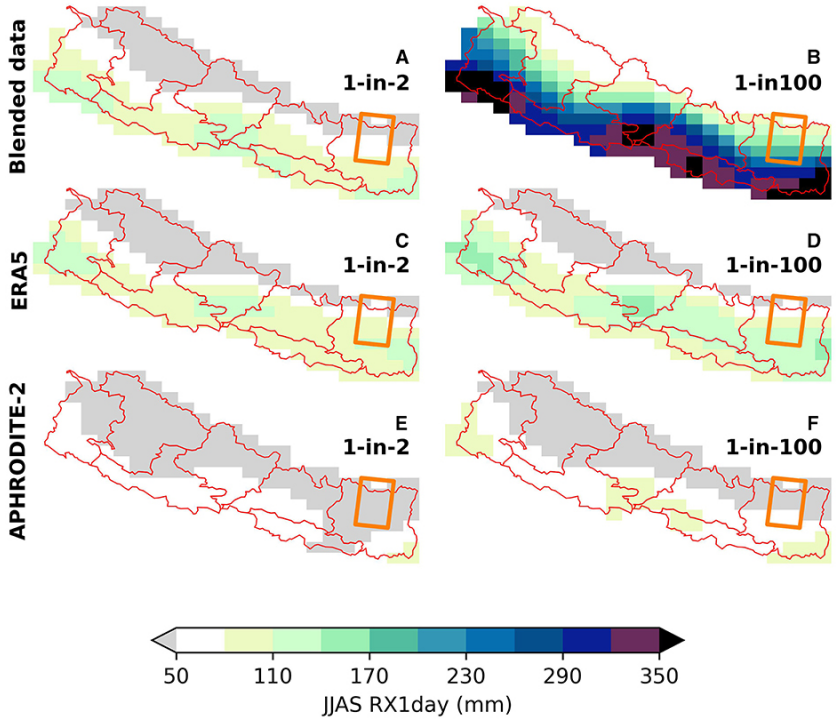


Figure 5.1: From Steptoe and Economou [2023], plots showing the estimates of 1-in-2 and 1-in-100 year RX1day events for the blended model and baseline datasets.

To exemplify the suitability of such an approach, the authors fit this model to the MSWEP, GloSea5, IMDAA and HAR datasets, whilst leaving the APHRODITE-2 and ERA5 as baseline datasets to evaluate their results to. Plotted in Figure 5.1 we see the estimates for 1-in-2 and 1-in-100 year RX1day precipitation events for the blended model and baseline datasets, which detail RX1day values which are estimated to occur every 2 years and every 100 years respectively from the blended model.

Interestingly, through such plots the authors note how estimates from a single dataset can seemingly misrepresent extreme precipitation events when compared to estimates that factor in information from several different sources. This is especially prevalent for more extreme events, as for the 1-in-100 predictions, which, as mentioned previously, is particularly concerning for new enterprises such as building a new dam as such extreme events are the incidents that need to be most accounted for.

Furthermore, in order to see how the model incorporates each component dataset, the authors also include a predictive distribution of the blended model against the input datasets as shown in Figure 5.2.

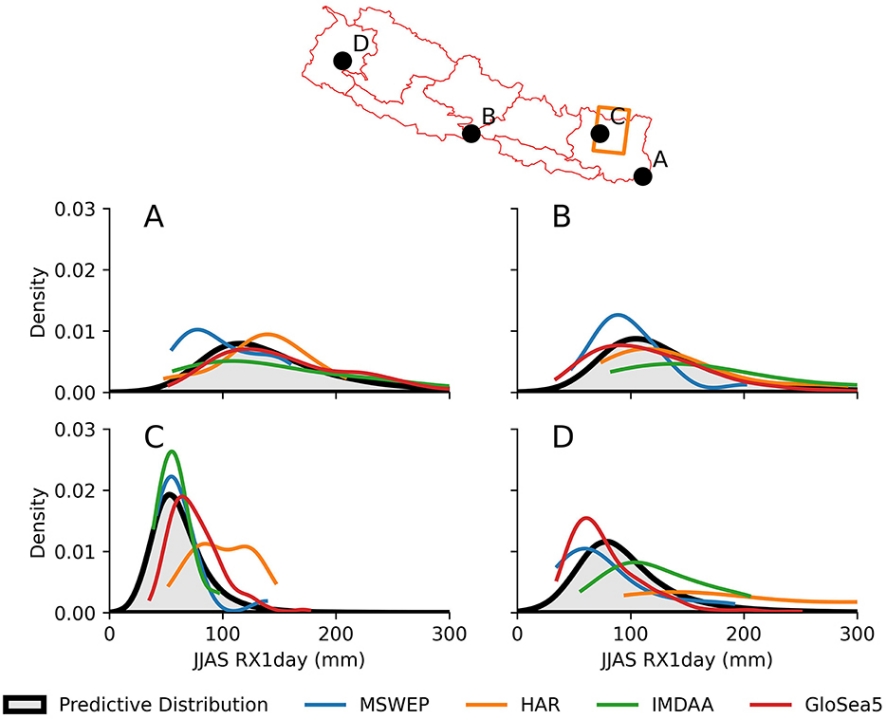


Figure 5.2: From Steptoe and Economou [2023], plots of the predictive distribution of the blended model with the distributions of its component datasets for 4 locations across Nepal as shown.

As we can see, such a model adequately incorporates all uncertainty present in each dataset, with it providing realistic values for each location. This highlights how the blending framework is working as hoped, with no single data source dominating the results.

5.3 Evaluation

Evaluating such work, it is important to note that such study was not necessarily done to find a new 'best' dataset. Rather, when a heuristic measure to identify a best dataset doesn't exist, such modelling offers a blending framework in order to summarise and incorporate all uncertainties presented by multiple data sources in a way which can influence reliable decision making. From this mindset, as mentioned by the authors, without the comparison of the model to the individual datasets, the model is no different than any other data source. Hence, it is the models output in addition to its comparison to its constituent datasets that is its biggest strength, with this being well suited to aid decision making. It is important to note also all this is done with the aim of producing a spatially and temporally consistent model, which is not restricted to the logistical requirements of observed data.

This is not the first time Generalized Additive Models have been used to model a combination of climate data, with them proving suitable for many related tasks. Notably, Economou et al. [2023] considered a probabilistic modelling framework by using Generalized Additive Models to integrate climate model reanalysis data with observational data in a way that can produce accurate results at any spatial resolution. Applying the problem specifically to temperature data from Cyprus and Morocco, they found that such an integration can allow imputation of missing observed values from relevant reanalysis data with strong in and out of sample results. However, although the methods used were similar to Steptoe and Economou [2023], such work differs in its aims, with less of a focus on providing a blended summary of numerous different datasets and more of a focus on utilising reanalysis data to help extend the use of observational data to provide more complete data records at a higher resolution.

Chapter 6

Initial Modelling

Having discussed the results of previous work, in this chapter we now present initial results of testing a BART model on each of the MSWEP, HAR, IMDAA and GloSea5 datasets separately, to see how such a model performs in understanding the spread of extreme precipitation across Nepal for any resolution.

6.1 Implementation within R

First of all, it is important to discuss our implementation of such a model within R.

Due to the models high performance, there exists numerous different packages which we may use to test BART upon our data, most notably **BayesTree** supplied by Chipman et al. [2010] in their paper introducing such a model. However, such a package has since been surpassed in numerous aspects such as computation times, and can often prove cumbersome due to an inability for models to be used along with the `predict` function.

Improving on such a package, the two main alternatives are given by the **bartMachine** and **BART** packages, introduced by Kapelner and Bleich [2013] and Sparapani et al. [2021] respectively. Both offer slight variations of the model introduced by Chipman et al. [2010], with the **bartMachine** package for example removing the SWAP step from the MCMC algorithm used to draw tree models from the posterior. They also differ in implementation language with **bartMachine** using Java and **BART** using C++. In fact, due to a preference working with the latter, we choose to use the **BART** for such work.

For such a package, it is important to note that when fitting a BART model it uses default parameters $(3, 0.90, 2, 200) = (\nu, q, k, m)$ unless otherwise specified. Hence, in early stages of model fitting, we shall use such values for ease of implementation.

6.2 First Models

Initially then, we will consider each dataset individually and evaluate if BART can provide a spatially consistent and accurate estimation of extreme precipitation across Nepal, according to each source. This will give us an early indication on the suitability of BART for our data, and help us decide upon specific modelling details used for our data blending framework later on.

For such work, due to the investigation in Chapter 4, we will use *Latitude* and *Longitude* as predictor variables and restrict entries to locations within Nepal in order to reduce computation times. As previously mentioned, we will also initially use default parameters given by the **BART** package.

6.2.1 Transforming the data

Before seeing the results of such models, we recall we had considered transforming our data in Chapter 4, due to a violation of the normality assumption needed for BART. To investigate this further, we fit our first BART model on the MSWEP dataset and consider plots of the residuals in Figure 6.1.

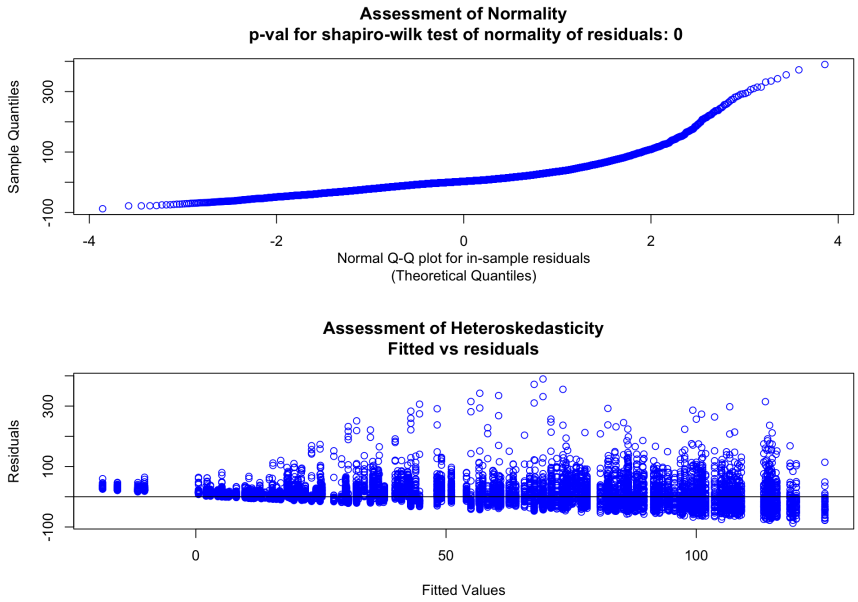


Figure 6.1: Plots checking model assumptions of the fitted BART model on the MSWEP data.

As we can see, through the Q-Q (quantile-quantile) plot there is a clear violation of normality with a lack of a straight line, indicative of such processes. Furthermore, such plots suggest heteroscedasticity of our errors due to the funnel shape of the second plot, again violating assumptions of our model.

As such assumptions aren't being met, the model is likely not capturing the given relationships as we'd have hope. Therefore, to address this issue and to stabilize this changing variance, it

might be wise to first consider a log transformation of our RX1day values before fitting such a model, as previously suggested.

As such behaviour is seen for all other datasets, for all future models we will perform log transformations of our response variable before fitting from now on.

6.2.2 Initial estimates

Using such transformed data, in Figure 6.2 we plot point estimates of each of our 4 BART models trained on each individual dataset, for each location. Such point estimates are derived from the average of the after burn-in samples $f^*(x)$, as given in Equation 3.13.

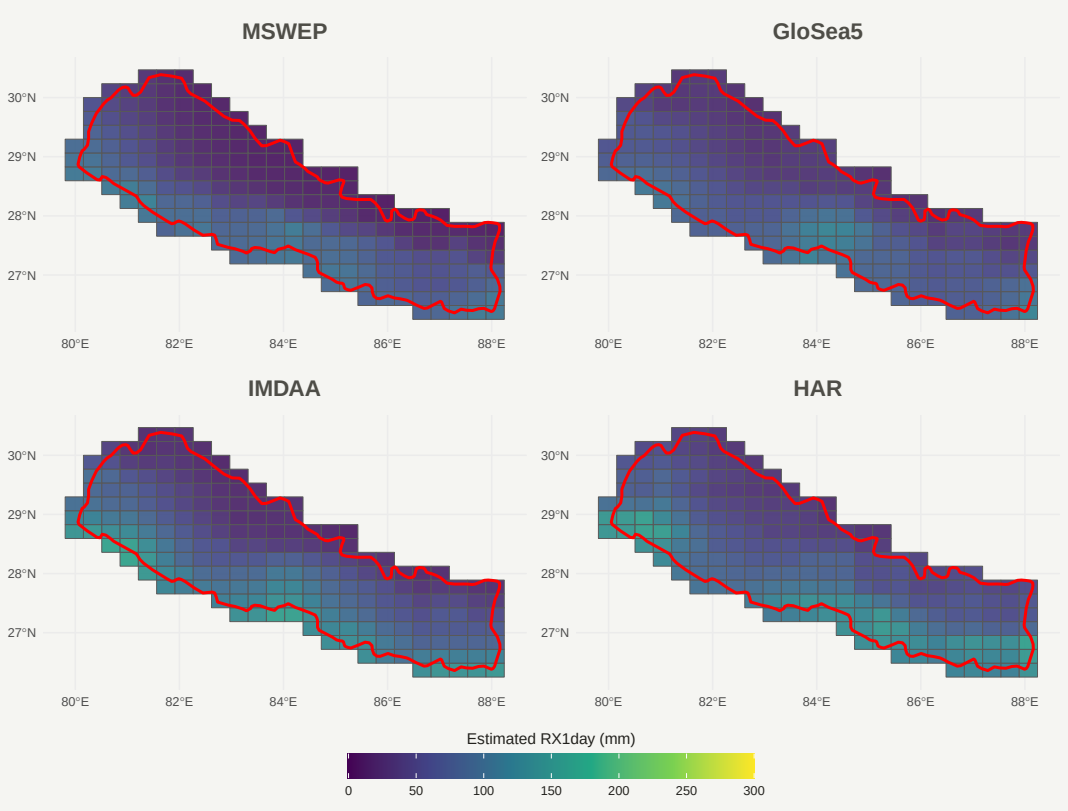


Figure 6.2: Predicted RX1day values from BART models fitted with transformed data.

Analysing this, we see our BART models seem to working reasonably well, with our results adequately representing Nepalese rainfall trends, and mimicking the uniqueness of each dataset seen last chapter in Figure 4.4, with the IMDAA and HAR datasets in particular offering more extreme estimates.

Publishing in-sample Root Mean Squared Error (RMSE) scores for each model in Table 6.1, we also note how each model seems to capture relationships in the data similarly well, with the exception of the MSWEP dataset which appears to perform slightly better. This is likely as a result of a difference in the recording of such a dataset.

Dataset	MSWEP	IMDAA	HAR	GloSea5
RMSE	27.678	49.364	47.594	43.116

Table 6.1: In-sample RMSE scores for each of the 4 models.

It is also important to check our transformation is working as hoped, and so in Figure 6.3 we again consider plots of the residuals of our model fitted on the MSWEP dataset.

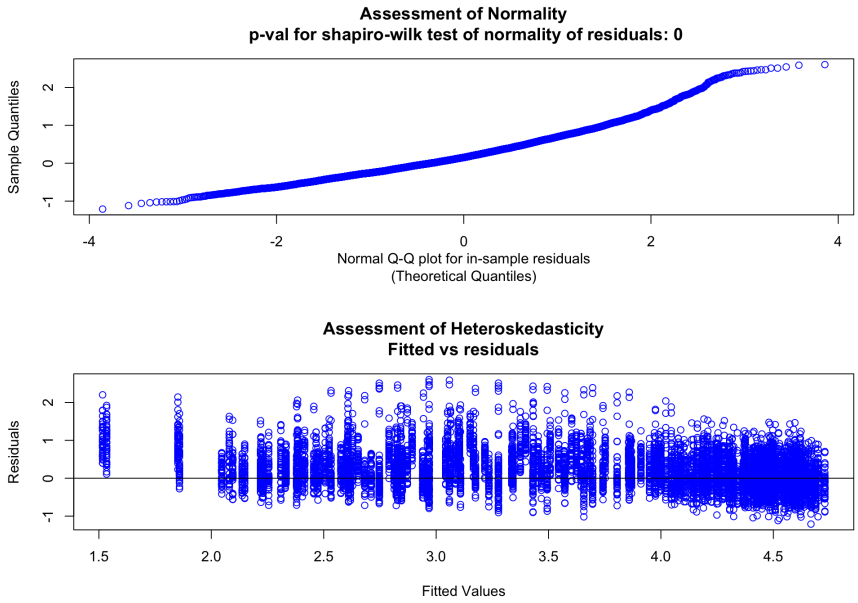


Figure 6.3: Plots checking the assumptions of the BART model fitted with log-transformed MSWEP data.

Analysing as such, the transformation seems to have worked with both assumptions appearing more plausible. This is most notably seen through the second plot, with the residuals now showing little signs of heteroscedasticity. Due to this, we consider no further transformations of our data.

6.3 Out-of-Sample Performance

As of yet, we have only considered the performance of such models in relation to values that the model itself has been trained on. This can lead to dangers related to overfitting as the model may appear to be performing better than in reality. Hence, it is also important to consider the performance of our models on unseen data by considering splitting into suitable training and testing subsets.

To do this, we perform 10 independent test/train splits on each dataset and consider the out-of-sample performance of our models through RMSE scores in Table 6.2. In detail, such splits are

made such that for every latitude-longitude pairing, we perform a random 80:20 split to divide our data such that 80% of the recorded years for each location are present in the training subset, and 20% are located in the test subset. This split is done differently for each location, to ensure the same years aren't systematically left out for all our training data, as this may induce our model to learn some unwanted temporal effects which may impact our results.

Dataset	**MSWEP**	**IMDAA**	**HAR**	**GloSea5**
In-sample RMSE	27.660	49.095	47.334	42.971
Out-of-sample RMSE	28.197	50.929	49.869	44.170

Table 6.2: In-sample and Out-of-sample RMSE scores for each of the 4 models.

As we can see, there is signs of slight overfitting with each model performing worse for unseen data rather than for in-sample results.

6.3.1 Cross-Validation

Thus far, for all of our models we have been using default values given in the **BART** package, $(\nu, q, k, m) = (3, 0.90, 2, 200)$. However, due to signs of overfitting seen in Table 6.2 and since we do not know if such values are optimal for our specific problem, it would be wise to perform a thorough evaluation of these hyperparameters. This will be done through cross-validation and will help us determine suitable values for use in future modelling.

Choice of m

The most influential of these hyperparameters is given through the number of trees, m, which can greatly increase the complexity and computation time of our model. Hence, before considering the other parameter we first consider m on its own.

Keeping the remaining hyperparameters as default and constant, in Figure 6.4 we plot the average RMSE given through a 4-fold cross-validation for each dataset and for a number of different values of m.

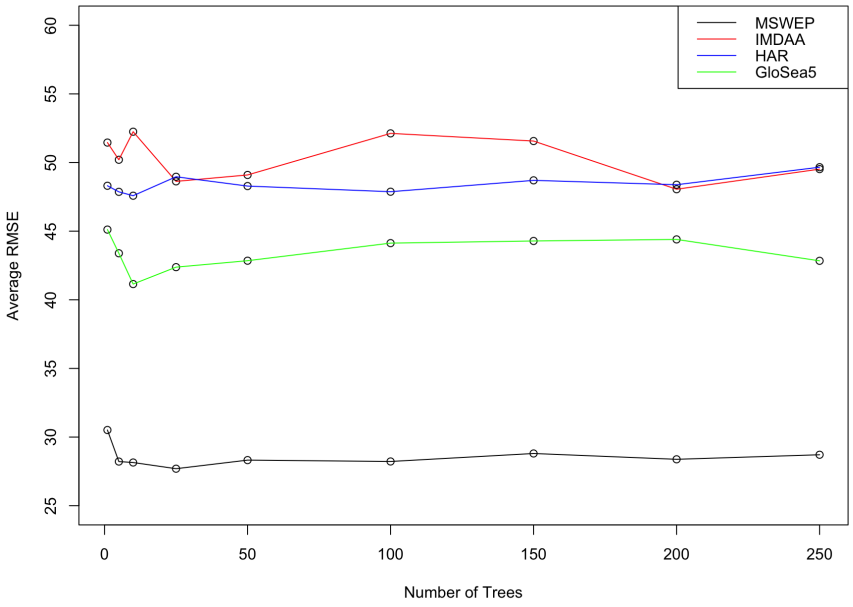


Figure 6.4: Average RMSE of each model from 4-fold cross validation with different values of m.

As we can see, for each model, the RMSE score decreases quickly at first before levelling off for subsequent values. The specific nature of this is different for each model, but a wise choice seems to be $m = 50$ as it offers near-optimal performance for each model whilst still avoiding the complexity and heavy computation times that come with larger values of m. Such a choice is also suggested by Kapelner and Bleich [2013] who found that BART models rarely offer increased performance for m larger than 50.

Grid-Search

To optimize the remaining parameters, we also perform a grid-search on the set of potential values, using cross-validation to assess performance. We avoid detail on this procedure but state that the optimal values chosen were $(3, 0.90, 2, 200) = (\nu, q, k, m)$ and hence such values will be used from now.

6.3.2 Comparison to other models

So that we may put some meaning behind such performance metrics, it is important to compare the performance of BART against other popular models. To this aim therefore, we consider 3 further modelling techniques through linear regression, gradient boosting (Friedman [2001]) and random forests (Breiman [2001]). These are implemented in R through the `lm`, `gbm` (Ridgeway and Ridgeway [2004]) and `randomforest` packages respectively.

For each dataset, we perform 20 independent test/train splits and compute the average out-of-sample performance for each model shown in Table 6.3. Both of the Random Forest and gradient boosting models use hyperparameters chosen through 4-fold cross-validation.

Dataset	BART	Random Forest	Gradient Boosting	Linear Regression
MSWEP	28.661	29.484	29.247	37.147
IMDAA	50.255	51.167	50.548	55.377
HAR	49.277	49.450	49.660	56.396
GloSea5	43.691	44.177	44.017	47.289

Table 6.3: Average out-of-sample RMSE results from each model tested on each dataset with 20 independent test/train splits.

As we can see, BART performs best for each dataset, with Random Forest and Gradient boosting coming next ahead of linear regression which as expected fails to capture the complex relationships in the data. This highlights the strength of our BART model, with it consistently out performing other well reputed models for each dataset.

For more detail, we also include box plots of the results of such testing on the MSWEP dataset in Figure 6.5, which highlights the superiority of BART in more detail.

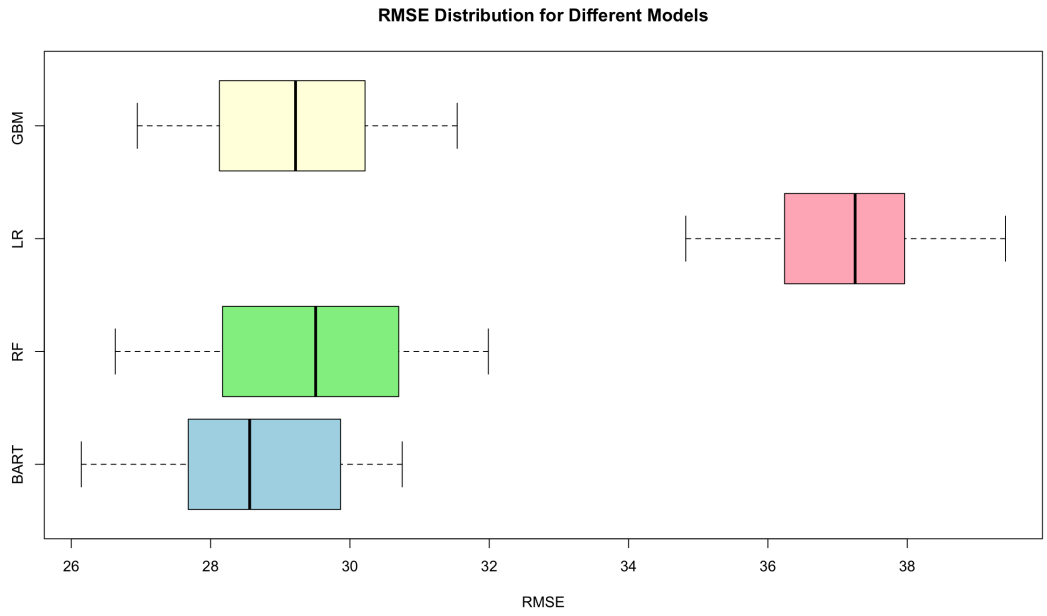


Figure 6.5: Box plots showing the out-of-sample performance of BART, Linear Regression (LR), Random Forest (RF) and gradient boosting (GBM) tested on the MSWEP dataset with 20 independent test/train splits.

6.4 Posterior Inference

As well as its high performance, one of the main advantages of using a BART model is the offer of full posterior inference on each of our predictions. Hence, to take advantage of this, using the locations specified in Figure 4.5 we consider the posterior distribution of our estimates.

In particular, in Figure 6.6 we include the plots of the estimated posterior distribution of $f(x)$ given through the kernel density estimate of the 1000 after burn-in samples $f^*(x)$, for each of the locations and for each dataset. This highlights the ability of each model to learn the individual relationships given in each dataset, with each fitted model offering extremely different estimates for each location.

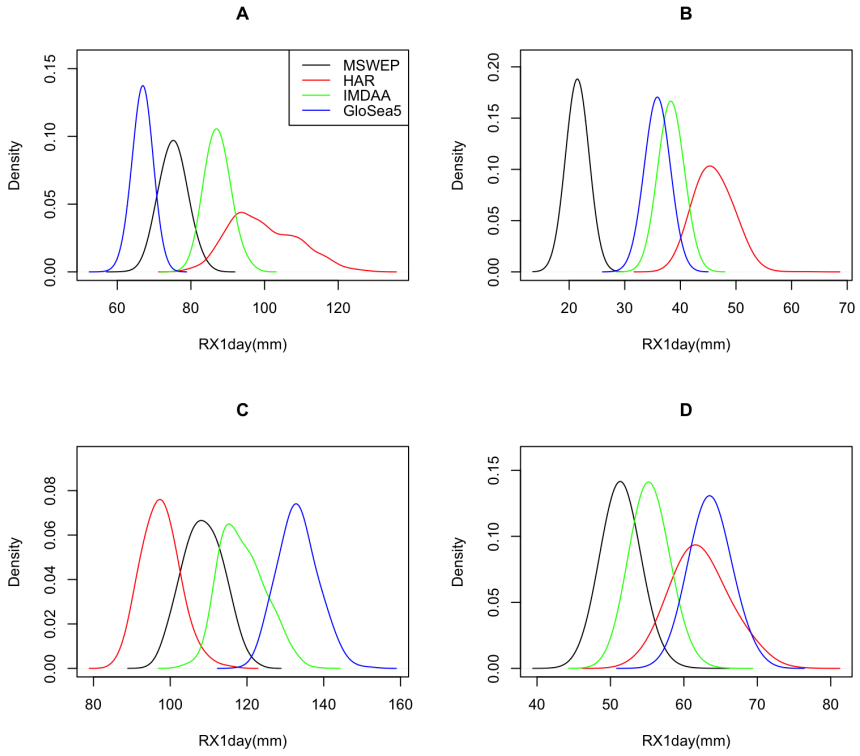


Figure 6.6: Plots of the estimated posterior distribution of annual RX1day values given for 4 locations from the BART model trained on each dataset.

It would also be nice to consider such posterior distributions in relation to the observed values at each location given for each dataset, to see if our model is adequately representing the uncertainty at each point. To this aim therefore, taking the fitted BART model trained on the MSWEP dataset, we consider as such in Figure 6.7.

As we can see, although both our posterior and kernel density of the sample values seem to peak at the same points, they differ greatly in the spread of the distribution, with the posterior being much more sharply peaked. This is especially an issue for our problem as under representing the uncertainty of extreme precipitation across Nepal can have huge effects for policymakers.

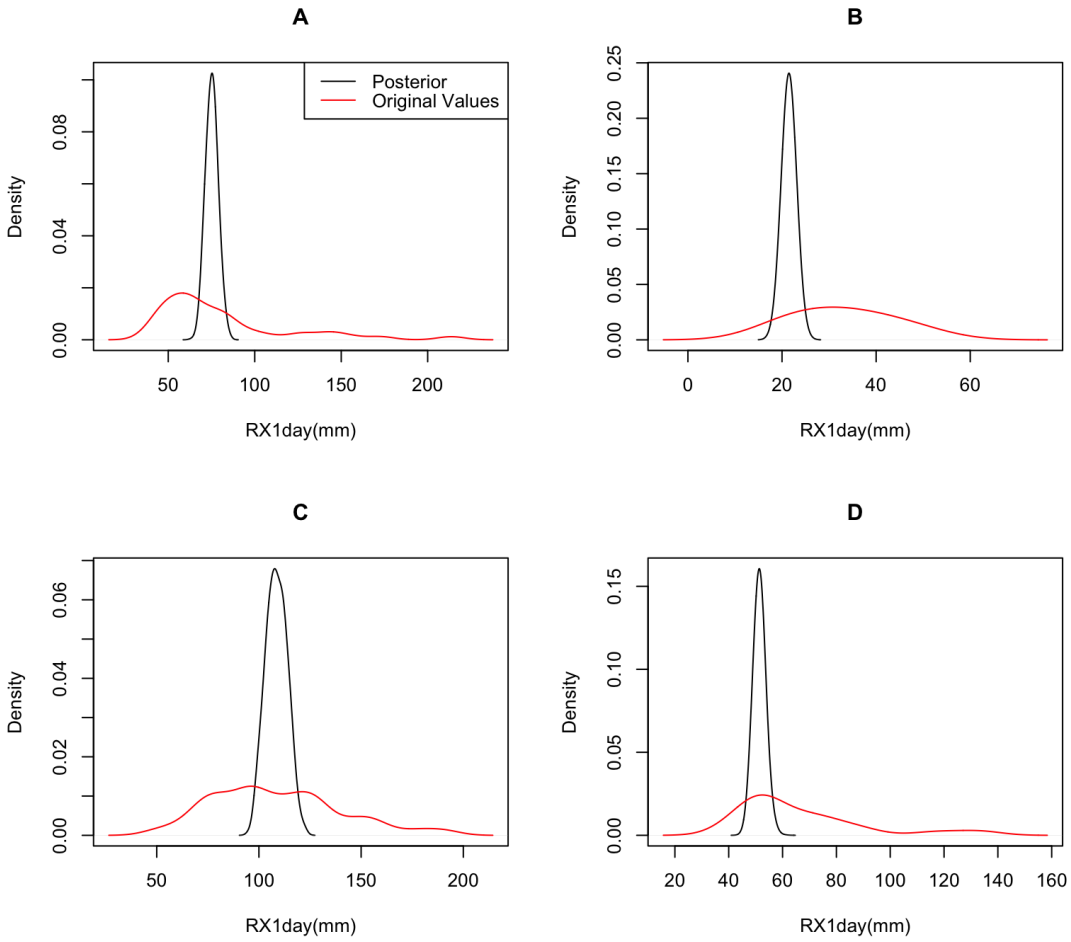


Figure 6.7: The estimated posterior distribution of 4 locations from the BART model trained on the MSWEP dataset, plotted along with the kde from actual observations from these locations from the MSWEP dataset.

Chapter 7

Data Blending

Having performed initial evaluation of BART's suitability on individual datasets, in this chapter we will now consider how we may use BART to adequately blend all of our data to produce a flexible model which can summarise all uncertainty present.

7.1 Aims

Recalling the situation presented in Chapter 1, and taking inspiration from Steptoe and Economou [2023], the main aim for our data blending framework is to produce a model which:

- Produces spatially consistent and reliable estimates that can be evaluated at any resolution.
- Incorporates uncertainty of all datasets in order to give a blended solution, not dominated by any single source.

Due to the sensitivity of the problem estimating extreme values, it would also be nice if our blended model were to over represent rather than under represent the uncertainty of extreme values in order to account for faults in the model itself.

To do this then, we need to consider how we may use BART to mix our data. One such option is given by Yannotty et al. [2024a] who use BART to weight the outputs of several models according to their performance, in order to give a combined estimate. More specifically, such model mixing has also been extended to climate models, with Yannotty et al. [2024b] using a slight variation of BART to do similar work on the output of 4 different climate models detailing surface temperature in the Northern hemisphere.

However, essential to such models is a 'ground-truth' dataset, used in training to determine the success of the other datasets in order to influence the weights. This proves a limitation for our needs therefore since, as mentioned previously, between our different observational datasets, such a dataset does not exist.

Hence, without such a 'ground-truth' dataset we are limited in the ways we may perform such a data blending framework. To this aim therefore, for our modelling we simply combine all data

into one merged dataset and train our model similarly to that done in chapter 6, with just Latitude and Longitude acting as predictor variables. By combining all data in such a way it is hoped that the overall uncertainty in the data may be learnt by our model, rather than being dominated by any individual data source which may occur if we were to split our data in alternative ways.

It is also important to note that before training with this merged dataset, we ensure each constituent data source is confined to a common recording period of the years 2000-2016, to ensure no unwanted temporal relationships are learnt by our model. To prevent the GloSea5 dataset from dominating our results due to its size, we also only include data from one realisation of the Met Office climate prediction model. As a result of this, our final merged dataset is made up of 3536 entries from each of the 4 different datasets.

7.2 Initial Results

Using such blended data, we therefore fit a BART model using all of the combined data. Although this is a new model, we use hyperparameters decided upon last chapter for ease, with such optimal values unlikely to change greatly. Evaluating such a model, shown in Figure 7.1, we plot the point estimates for each observational location.

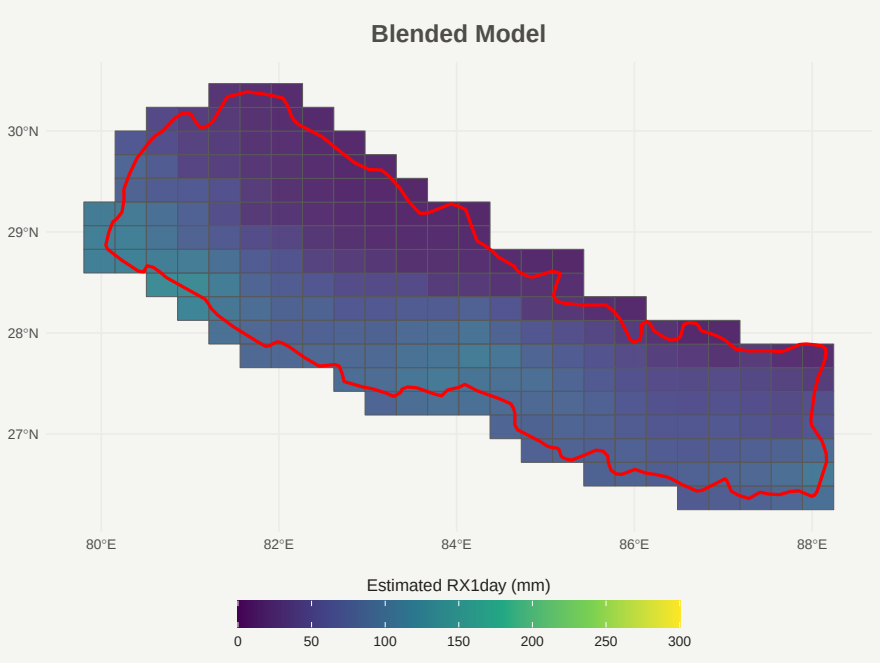


Figure 7.1: Predictions from the BART model trained on the blended data.

As we can see, such results are spatially consistent as desired, and also represent known Nepalese rainfall trends as previously seen, with higher values given in the South and lower values in the North. The estimates also seem to adequately summarise values seen from each dataset as shown in Figure 4.4, with our results not being too similar to any one source.

It would also be useful to see how our model compares to other methods, and so just as before we make 20 independent test/train splits and test BART along with linear regression, gradient boosting and random forests models. To ensure no one dataset dominates any model, such a split is done in a way so that there exists an equal number of entries from each data source. Doing as such, we show results from this in Figure 7.2.

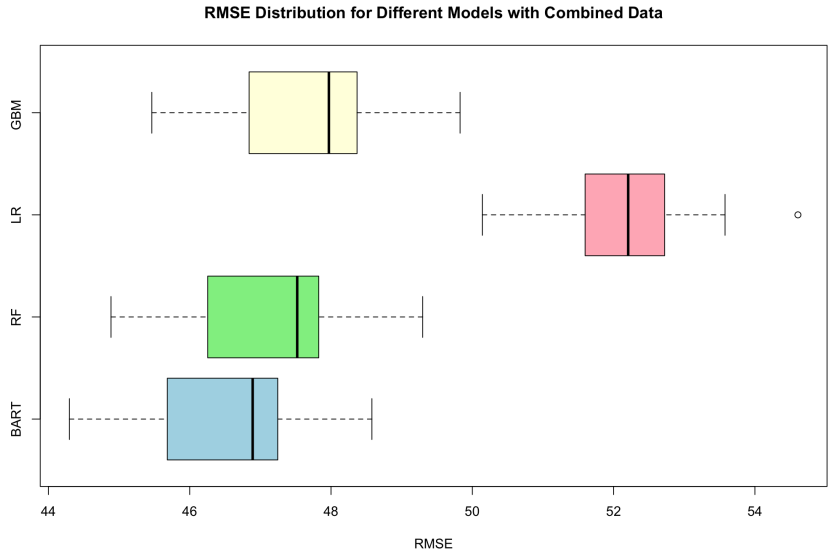


Figure 7.2: Box plots of RMSE values from each test/train split on our blended dataset for fitted BART, Random Forest (RF), Linear Regression (LR) and Gradient Boosting (GBM) models.

As before, we see the superior performance of BART, with the average RMSE being lowest for such a model. This highlights its suitability to our data blending framework and shows it is learning the complex relationships within our data better than other competing models.

7.3 Uncertainty

Having shown that our blended model produces accurate and spatially consistent estimates, we must now investigate the other important feature of our blended model on whether it adequately encapsulates the uncertainty of each dataset. This is where the true strength in our BART model hopefully will come through through its embodiment of the Bayesian framework.

7.3.1 Model Uncertainty

To do this, let us first consider the uncertainty given by the BART model of the 4 locations specified in Figure 4.5. This is done by considering the kernel density estimate (KDE) of the 1000 after burn-in samples of $f^*(x)$ for each location.

Plotted in Figure 7.3, for comparison we also include the the KDE of raw values from each of the individual datasets for each location, with values cut off at their data limits.

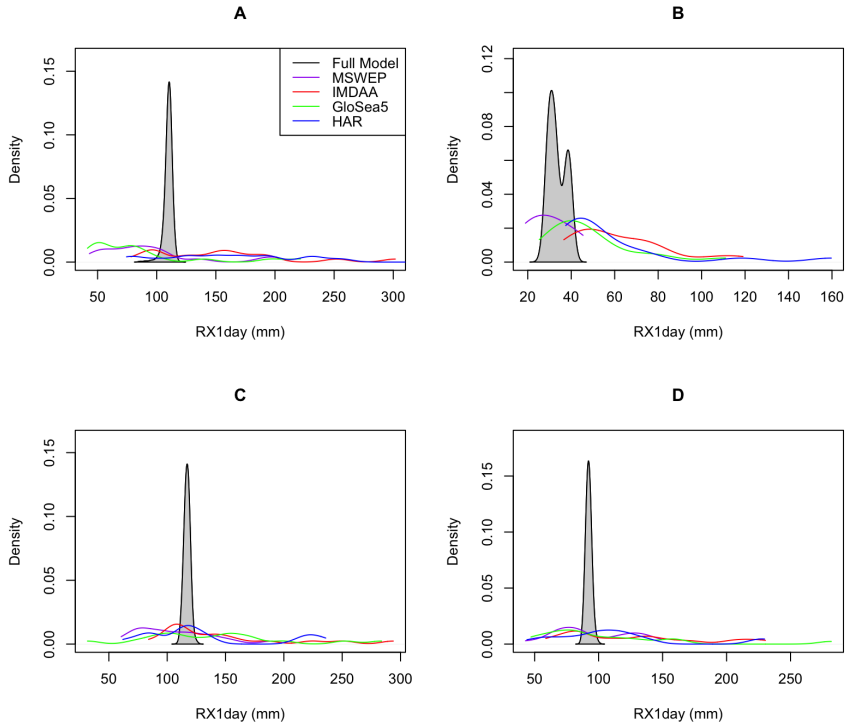


Figure 7.3: Uncertainty of $f(x)$ for 4 locations given by our blended model. KDEs from raw values from each dataset are also plotted, with values cut off at their data limits

Interestingly, we note that the peaks of our blended model estimates all seem to lie in realistic locations related to the plot from each individual dataset, with the blended model seemingly not favouring any particular source. Unfortunately though, such estimates are sharply peaked, giving a lack of uncertainty for the value at each point. This is extremely undesirable for our model, with this misrepresentation of the uncertainty proving dangerous for the contextual nature of our problem.

To solve this, we first consider using similar methods to bagging (Breiman [1996]) by training several different BART models on different subsets of the blended data and aggregating the results. Through this, each model will capture relationships in different parts of the data, hopefully resulting in some variation in our results which will cause such posterior estimates to be less sharply peaked.

Specifically, we fit 20 models each trained on a different sub sample of the data and combine posterior estimates from all models. We then compute kernel density estimates from such values and plot accordingly in Figure 7.4 for the same locations as before. Such sub samples of the data are created using similar methods to our test/train splits, by ensuring each individual dataset is represented equally.

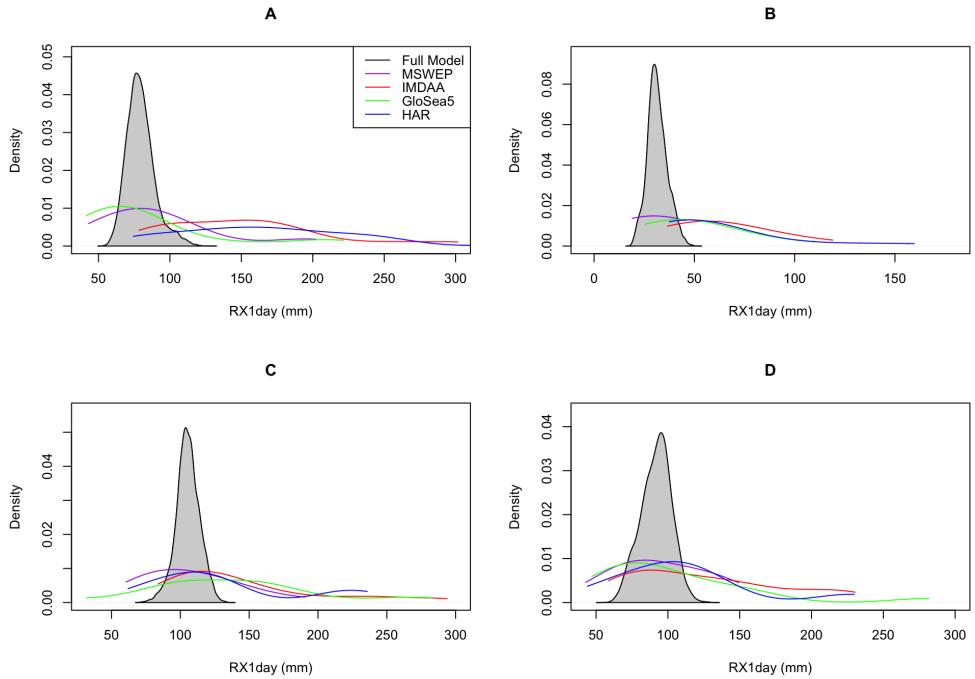


Figure 7.4: Uncertainty plots for 4 locations given through multiple BART models trained on different parts of our blended dataset. KDEs from raw values from each dataset are also plotted, with values cut off at their data limits

As we can see, such methods improves our uncertainty estimates, with our new framework now given less sharply peaked estimates as hoped. However, the problem has still not been fully removed, with such high-peaked distributions still sub-optimal for our aims.

7.3.2 Full Uncertainty

As we recall, in our BART framework we look to model

$$y = f(x) + \epsilon \qquad \epsilon \sim N(0, \sigma^2) \tag{7.1}$$

However, plots in Figure 7.3 and Figure 7.4 only show the uncertainty of $f(x)$, which is estimated from the sequence of after burn-in samples $f^*(x)$. This represents draws from the posterior distribution of $f(x)$, rather than the predictive distribution of y and so, in other words, up till now we have only considered the uncertainty of the **mean** of y, given through $f(x)$.

Hence, such plots are only showing the uncertainty of y produced from the uncertainty of the BART model itself, denoted as the **Epistemic** uncertainty. To understand the true predictive uncertainty of y therefore, we must also include the added uncertainty given by ϵ representing the natural uncertainty or inherent noise in the data. This is instead denoted as the **Aleatoric** uncertainty of y.

To do this, after each burn-in sample $f^*(x)$ we also add a sample drawn from $\epsilon \sim N(0, \sigma^2)$, to create a new sequence of samples that incorporates both the Epistemic and Aleatoric uncertainty of y. As the sequence $f^*(x)$ converge to the true posterior distribution of $f(x)$ as discussed in Chapter 3, and as repeated sampling of ϵ converges to $N(0, \sigma^2)$ due to Monte Carlo methods (Robert et al. [1999]), such a sequence adequately represents the full uncertainty of y given by our modelling for each prediction.

Such different types of uncertainty is discussed in detail by Valdenegro-Toro and Mori [2022] who states that 'these uncertainties are usually combined and predicted as a single value, called predictive uncertainty'. Taking Bayesian Linear Regression as an example, as described by Korbak [2024], the nature of the model produces a predictive distribution which naturally incorporates both aleatoric and epistemic uncertainty. However, due to the nature of our BART model, such a predictive distribution is analytically unfeasible, with only the posterior distribution of $f(x)$ being able to be sampled from and so we may proceed as previously described.

Updated Results

Using such methods therefore, in Figure 7.5 we include plots of the full uncertainty of y for each point.

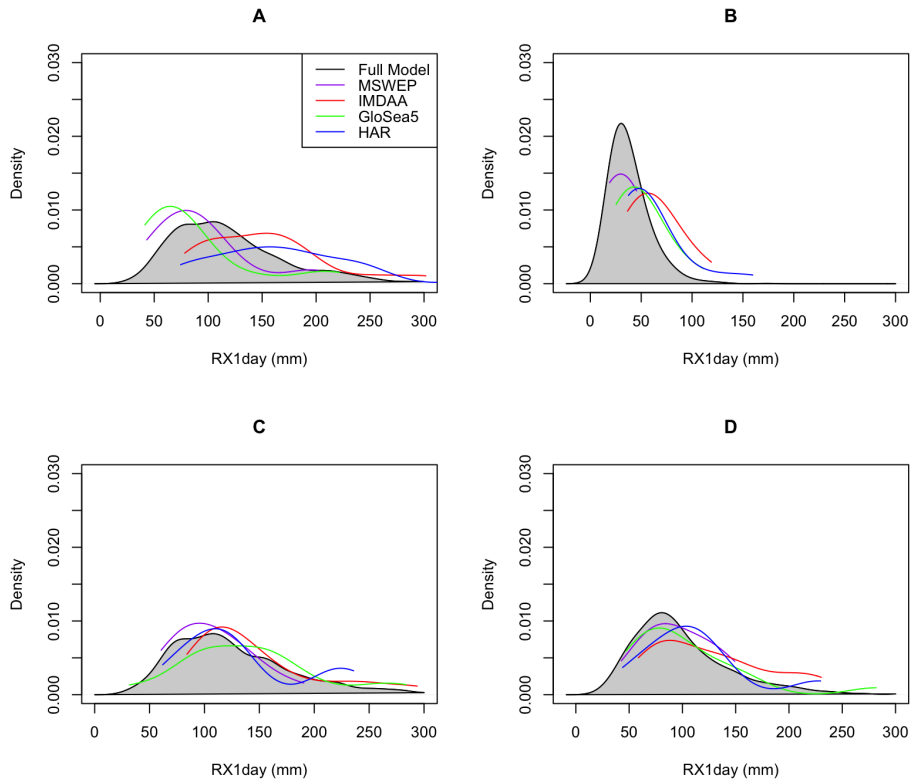


Figure 7.5: Full uncertainty of RX1day(mm) estimates for 4 locations from our blended model. KDEs from raw values from each dataset are also plotted, with values cut off at their data limits

As we can see, such plots are much better, with the added uncertainty from ϵ leading to less sharply peaked distributions from the blended model. In particular, for locations A, C and D the blended model gives a predictive distribution which mediates all individual datasets, whereas at location B, the model seemingly favours a left-skewed distribution influenced by the MSWEP dataset. This highlights how the blended model produces estimates which favour different models at different locations, showing it can adapt to the different relationships seen across Nepal to give a summary of the uncertainty present in each dataset.

7.3.3 Changes in Uncertainty

It is also important to understand how such uncertainty in our estimates change throughout Nepal, so in Figure 7.6 we plot the standard deviation of each estimate given in Figure 7.1 for each location. This is calculated through the standard deviation of the 1000 after burn-in samples, $f^*(x)$, for each location.

Interestingly, we note that uncertainty is highest in the most Western regions of Nepal, where estimates of extreme precipitation are highest, and lowest in the North where such estimates are lowest. As such a feature seems logical this would suggest are model is behaving in a sensible way.

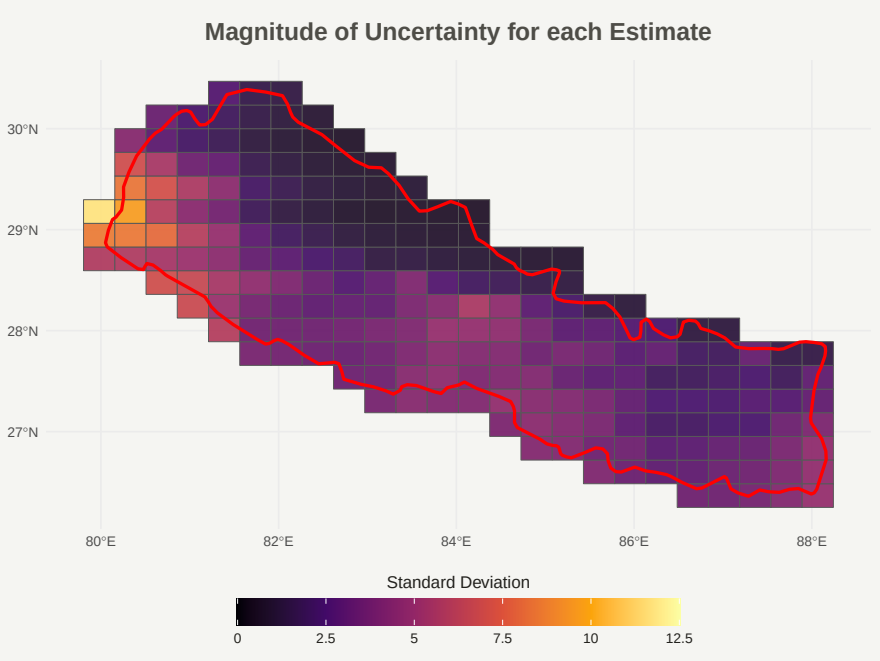


Figure 7.6: Standard Deviation of estimates for each location in Nepal from our blended model.

7.4 Comparison to Previous Work

With our blended model now adequately summarising the uncertainty of each dataset, our attention now turns to comparing our results to that of previous work done by Steptoe and Economou [2023]. To do this, in Figure 7.7 we first consider uncertainty estimates of the 4 locations for both our blended model and the blended model of Steptoe and Economou [2023] based on Generalized Additive Models (GAMs).

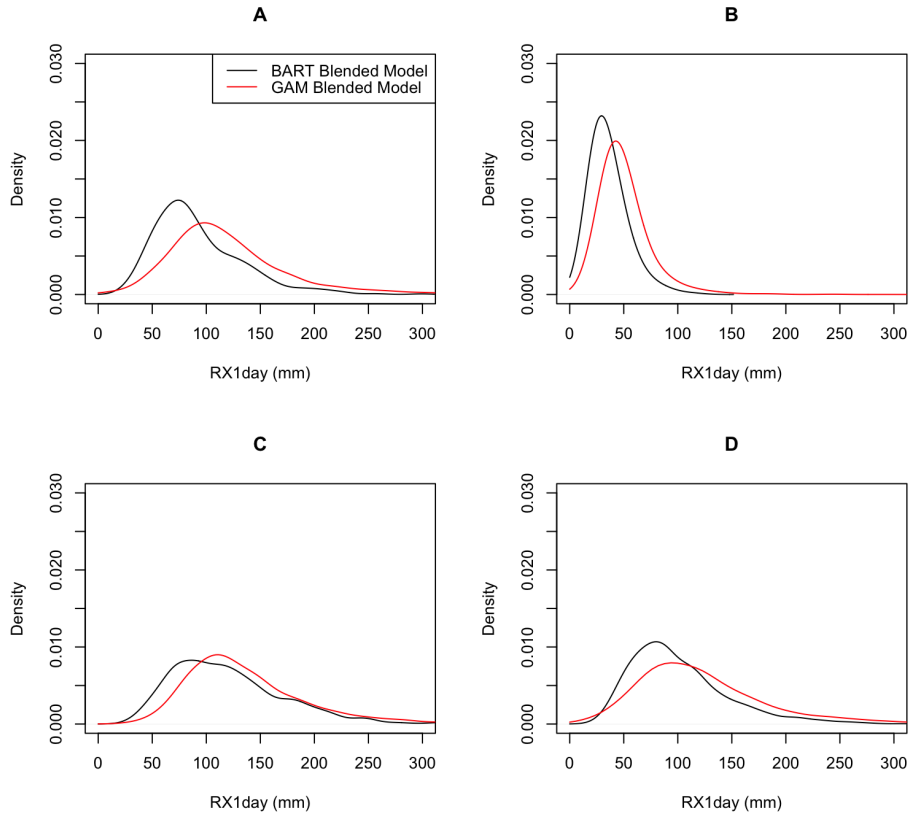


Figure 7.7: Uncertainty estimates from our blended model and the blended model of Steptoe and Economou [2023] for 4 locations.

Interestingly, we note the similarity in such uncertainty estimates, albeit with our blended model often giving slightly higher peaked distributions. This highlights the suitability of our BART model with it producing similar results, but also the difficulty of the problem, with no clear way of determining which of these blended models are optimal.

To further such comparisons, we also look to consider 1-in-2 and 1-in-100 year RX1day estimates from our blended model. As previously described in Chapter 5, such estimates provide estimates of values likely to happen every 2 or every 100 years from the predictive distribution given from the blended model and are compared to that of a further baseline dataset, APHRODITE-2.

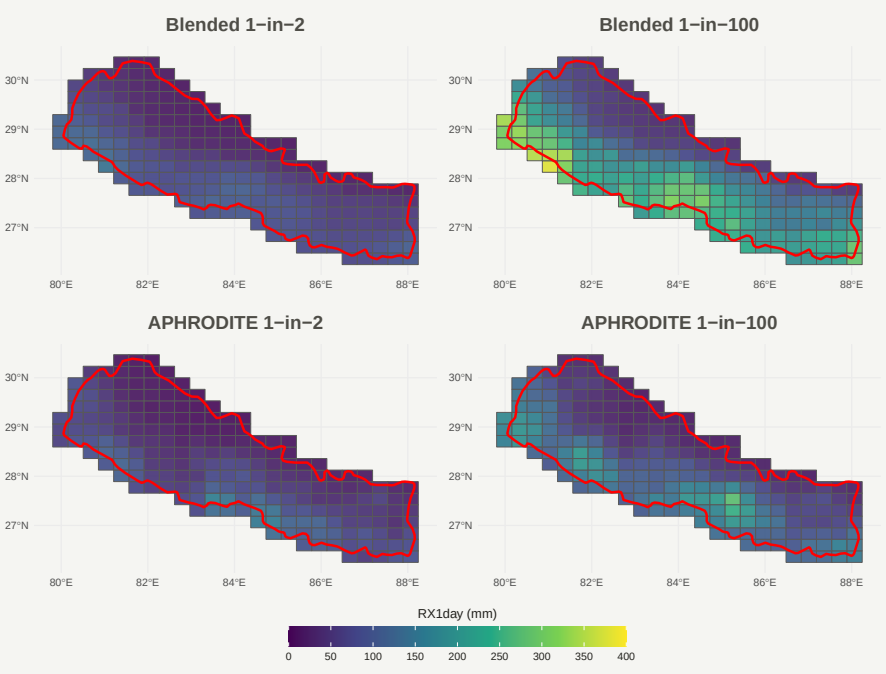


Figure 7.8: 1-in-2 and 1-in-100 year RX1day estimates from our blended model compared to the APHRODITE dataset. (Note the fill of the chloropleth has been altered to account for a larger range of values.)

Plotted in Figure 7.8 we clearly see that the APHRODITE-2 dataset hugely under represents high precipitation events compared to our blended estimate, especially for more extreme events. Such results are similar to those given by Steptoe and Economou [2023] in Figure 5.1, and again highlight the adequacy of our BART model with it produces similar results to those already seen.

Hence, as well as highlighting the suitability of BART for such a data blending framework, such similar results further reiterates the dangers for policymakers of only using a single dataset compared to using a blended solution composed by many different sources, with it giving much different estimates of extreme precipitation throughout Nepal.

Chapter 8

Conclusion

Concluding then, in this paper we have investigated the suitability of BART to provide a data blending solution to the problem of a proliferation of atmospheric datasets throughout Nepal.

Taking inspiration from Steptoe and Economou [2023], we have specifically looked to provide a solution to the issue of predicting extreme precipitation events when numerous competing datasets are available, aiming to provide a blended model which is spatially consistent and able to suitably incorporate all information available from constituent datasets.

Shown in Chapter 7, such aims were met with BART providing a blended model which gave spatially consistent and accurate annual RX1day estimates throughout Nepal. Offering optimal out-of-sample performance compared to competing models, once both the Epistemic and Aleatoric uncertainty had been included, our blended model also produced uncertainty quantifications which adequately summarised all uncertainty available, with no one single dataset dominating our results.

Interestingly, for both estimations of 1-in-x year events and the uncertainty of specific locations, our blended model also produced similar results to that of Steptoe and Economou [2023], highlighting both the suitability of our model and the dangers of the problem. In particular, such work re-iterated the need for policymakers to incorporate numerous datasets in order to make reliable decisions, as in Figure 7.8 we saw how extreme events can be hugely underrepresented unless we consider a data blending solution.

However, there still exists a great deal of further investigation needed into the problem to check the suitability of using BART in such an environment. Most notably, by choosing one of the datasets as 'ground-truth', it would be interesting to investigate further work such as by Yannotty et al. [2024b], in order to evaluate more sophisticated data blending techniques through model mixing methods. Furthermore, additional evaluation of the different types of uncertainty present in our final blended model is needed to help understand such results in more detail.

Bibliography

Hylke E Beck, Eric F Wood, Ming Pan, Colby K Fisher, Diego G Miralles, Albert IJM Van Dijk, Tim R McVicar, and Robert F Adler. Mswep v2 global 3-hourly 0.1 precipitation: methodology and quantitative assessment. *Bulletin of the American Meteorological Society*, 100(3): 473–500, 2019.

George EP Box and David R Cox. An analysis of transformations. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 26(2):211–243, 1964.

L. Breiman, J. Friedman, R.A. Olshen, and C.J. Stone. *Classification and Regression Trees*. Chapman and Hall/CRC, 1st edition, 1984. doi: 10.1201/9781315139470.

Leo Breiman. Bagging predictors. *Machine learning*, 24:123–140, 1996.

Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

Wray Buntine. Learning classification trees. *Statistics and Computing*, 2(2):63–73, 1992.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

Hugh A Chipman, Edward I George, and Robert E McCulloch. Bayesian cart model search. *Journal of the American Statistical Association*, 93(443):935–948, 1998.

Hugh A Chipman, Edward I George, and Robert E McCulloch. Bart: Bayesian additive regression trees. 2010.

David GT Denison, Bani K Mallick, and Adrian FM Smith. A bayesian cart algorithm. *Biometrika*, 85(2):363–377, 1998.

Theo Economou, Georgia Lazoglou, Anna Tzyrkalli, Katiana Constantinidou, and Jos Lelieveld. A data integration framework for spatial interpolation of temperature observations using climate model data. *PeerJ*, 11:e14519, 2023.

Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

Trevor Hastie and Robert Tibshirani. Bayesian backfitting (with comments and a rejoinder by the authors. *Statistical Science*, 15(3):196–223, 2000.

W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. 1970.

Hans Hersbach, Bill Bell, Paul Berrisford, Shoji Hirahara, András Horányi, Joaquín Muñoz-Sabater, Julien Nicolas, Carole Peubey, Raluca Radu, Dinand Schepers, et al. The era5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, 146(730):1999–2049, 2020.

Jennifer Hill, Antonio Linero, and Jared Murray. Bayesian additive regression trees: A review and look forward. *Annual Review of Statistics and Its Application*, 7(1):251–278, 2020.

Adam Kapelner and Justin Bleich. Bartmachine: Machine learning with bayesian additive regression trees. *arXiv preprint arXiv:1312.2171*, 2013.

Tomek Korbak. Interpreting uncertainty in bayesian linear regression. https://tomekkorbak.com/2020/05/29/interpreting-uncertainty-in-bayesian-linear-regression/: :text=Aleatoric

Antonio R Linero and Yun Yang. Bayesian regression tree ensembles that adapt to smoothness and sparsity. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 80 (5):1087–1110, 2018.

Yang Liu, Mikhail Traskin, Scott A Lorch, Edward I George, and Dylan Small. Ensemble of trees approaches to risk adjustment for evaluating a hospital's performance. *Health care management science*, 18:58–66, 2015.

C MacLachlan, Alberto Arribas, K Andrew Peterson, A Maidens, D Fereday, AA Scaife, M Gordon, M Vellinga, A Williams, RE Comer, et al. Global seasonal forecast system version 5 (glosea5): A high-resolution seasonal forecast system. *Quarterly Journal of the Royal Meteorological Society*, 141(689):1072–1084, 2015.

Amal Saki Malehi and Mina Jahangiri. Classic and bayesian tree-based methods. In Petrică Vizureanu, editor, *Enhanced Expert Systems*, chapter 3. IntechOpen, Rijeka, 2019. doi: 10.5772/intechopen.83380. URL `https://doi.org/10.5772/intechopen.83380`.

Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

Peter Müller, Ya-Chen Tina Shih, and Song Zhang. A spatially-adjusted bayesian additive regression tree model to merge two datasets. 2007.

Jonathan J Oliver and David J Hand. On pruning and averaging decision trees. In *Machine Learning Proceedings 1995*, pages 430–437. Elsevier, 1995.

S Indira Rani, T Arulalan, John P George, EN Rajagopal, Richard Renshaw, Adam Maycock, Dale M Barker, and M Rajeevan. Imdaa: High-resolution satellite-era reanalysis for the indian monsoon region. *Journal of Climate*, 34(12):5109–5133, 2021.

RI Rhodes, LC Shaffrey, and SL Gray. Can reanalyses represent extreme precipitation over england and wales? *Quarterly Journal of the Royal Meteorological Society*, 141(689):1114–1120, 2015.

Greg Ridgeway and Maintainer Greg Ridgeway. The gbm package. *R Foundation for Statistical Computing, Vienna, Austria*, 5(3), 2004.

Christian P Robert, George Casella, and George Casella. *Monte Carlo statistical methods*, volume 2. Springer, 1999.

Rodney Sparapani, Charles Spanbauer, and Robert McCulloch. Nonparametric machine learning and efficient computation with bayesian additive regression trees: The bart r package. *Journal of Statistical Software*, 97:1–66, 2021.

Hamish Steptoe and Theo Economou. Proliferation of atmospheric datasets can hinder policy making: a data blending technique offers a solution. *Frontiers in big Data*, 6:1198097, 2023.

Luke Tierney. Markov chains for exploring posterior distributions. *the Annals of Statistics*, pages 1701–1728, 1994.

Matias Valdenegro-Toro and Daniel Saromo Mori. A deeper look into aleatoric and epistemic uncertainty disentanglement. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1508–1516. IEEE, 2022.

Xun Wang, Vanessa Tolksdorf, Marco Otto, and Dieter Scherer. Wrf-based dynamical downscaling of era5 reanalysis data for high mountain asia: Towards a new version of the high asia refined analysis. *International Journal of Climatology*, 2020.

Yuhong Wu, Håkon Tjelmeland, and Mike West. Bayesian cart: Prior specification and posterior simulation. *Journal of Computational and Graphical Statistics*, 16(1):44–66, 2007.

John C Yannotty, Thomas J Santner, Richard J Furnstahl, and Matthew T Pratola. Model mixing using bayesian additive regression trees. *Technometrics*, 66(2):196–207, 2024a.

John C Yannotty, Thomas J Santner, Bo Li, and Matthew T Pratola. Combining climate models using bayesian regression trees and random paths. *arXiv preprint arXiv:2407.13169*, 2024b.

Akiyo Yatagai, Kenji Kamiguchi, Osamu Arakawa, Atsushi Hamada, Natsuko Yasutomi, and Akio Kitoh. Aphrodite: Constructing a long-term daily gridded precipitation dataset for asia based on a dense network of rain gauges. *Bulletin of the American Meteorological Society*, 93(9):1401–1415, 2012.

Junni L Zhang and Wolfgang K Härdle. The bayesian additive classification tree applied to credit risk modelling. *Computational Statistics & Data Analysis*, 54(5):1197–1205, 2010.