Using Machine Learning Algorithms to Better Bound The Uncertainty Ranges of Parameters in GCMs

Alexander Vukadinovic

University of Waterloo, 2017
Department of Applied Mathematics

Acknowledgements

I would like to extend my deepest thank you and appreciation to my supervisor Dr. Chris Fletcher for his guidance and support during my time as a graduate student. I am grateful to have been given access to your wealth of knowledge, which has been invaluable to my academic career. It has been a pleasure to work in such an interdisciplinary environment, and I feel I have benefited greatly as a result. A special thank you to Dr. Adam Kolkiewicz for contributing his time and expertise to reviewing this paper and for providing helpful feedback.

Contents

1 Introduction

Complex systems such as global climate models require large computational resources to run simulations of the physical systems they model. Generally these models take input parameters x_j, $j = 1, 2, ..., d$, to produce responses, or output variables $\mathbf{y} = f(\mathbf{x})$. Varying the input parameters to test the response of the output variables can be a feasible approach for certain problems, depending on how much computation time is needed to complete the experiments. Understanding the relationships between the input parameters of the model and the output variables can be difficult to discern analytically due to the highly complex mathematical form of the system. Where the computation time required is is not feasible, a simpler statistical model, called a meta-model, can be used as an approximation to predict variable responses for a large number of input parameter combinations. These statistical models can in tern be used to determine which input parameters have significant impacts on which output variables. Generalized linear models (GLMs) are commonly used for this task largely due to their simplicity. Previous work such as [22] employs a GLM, specifically a multiple linear regression (MLR), and variance-based sensitivity analysis to determine influential input parameters on top of atmosphere net radiative fluxes in the Community Atmosphere Model 5 (CAM5). Similarly, [17] uses this method to assess the influence of parameters on quantitative cloud and aerosol processes in CAM5. The employed GLM method usually assumes an appropriate approximation to be a function of a linear combination of input parameters and their interactions, which may introduce biases in the estimation of parameter importance. An alternative method is to use nonlinear models and the importance measure suggested in [9]:

$$\hat{VI}_j = 1 - \frac{\hat{Var}[E[\hat{f}(\mathbf{x}|\mathbf{x}_{-j})]]}{\hat{Var}(\hat{f}(\mathbf{x}))} \tag{1}$$

for parameter j[1]. The drawback of this approach is the need to employ Monte Carlo (MC) or quasi-Monte Carlo (QMC) sampling to estimate the expected values and variances. In (4), $\hat{f}(\cdot)$ is the meta-model of $f(\cdot)$. In the case of climate models, usually many output variables are of interest, which means assessing parameter importance for M variables requires $d \times M$ values of VI_j. The computational costs can be quite high as a result. The paper also considers a step-wise approach where $f(\mathbf{x})$ is evaluated by adding a parameter x_j at each step, but this also may not be ideal since complex relationships between parameters may result in importance biases with this approach as well.

The first part of this paper suggests the use of the random forest (RF) permutation importance measure to be applied to this type of problem for climate models. This avoids much of the computational costs and provides unbiased estimates of parameter importance. The second goal of this paper is to outline a process for constraining parameter uncertainty ranges for a given global climate profile. For a given vector $\mathbf{y}$ of climate model output variables, which could contain, for example, global mean temperature, radiative fluxes, high cloud percentage etc., there may be many combinations of input parameters $\mathbf{x}$ which produce a similar climate profile to $\mathbf{y}$. Determining these plausible ranges of $\mathbf{x}$ involves sampling this parameter space extensively and finding which values of $\mathbf{x}$ produce a climate similar to that described by $\mathbf{y}$. Evaluating $f(\mathbf{x})$ for each $\mathbf{x}$ is not computationally feasible and instead a

[1]The $-j$ denotes all components of $\mathbf{x}$ excluding component j

meta-model $\hat{f}(\mathbf{x})$ is used to estimate climate model output values. Proposed here is the use of ϵ-Support Vector Regression (ϵ-SV) model for $\hat{f}(\cdot)$. The method is compared to MLR models which are commonly employed as $\hat{f}(\mathbf{x})$ in climate related literature. Due to the highly non linear relationships between the input parameters and the output variables in climate models, ϵ-SV models should provide much better meta-models than the MLR method. In a related study [16], the ϵ-SV model is shown to over-fit on the training data when used to forecast northern polar stratospheric variability and the author recommends using MLR for forecasting. It is however noted by the author that the range of tuning parameters explored for ϵ-SV may not have been optimal. In this paper, a wider range of parameters are explored and chosen using $80-20$ MC cross validation. The mean squared error (MSE) of the optimal models are compared with the MSE of a MLR model with variable selection. Results indicate that ϵ-SV is better adapted to be used as $\hat{f}(\cdot)$ compared to MLR.

The RF, MLR and ϵ-SV models are applied to data obtained from 350 ensemble CAM4 runs. Parameters related to black carbon (BC) and sulfate (SO4) aerosols are perturbed in order to assess their impact on 14 output variables describing the climate profile. In current GCMs, typical resolutions of the atmosphere component are about 100 to 200km in the horizontal and 100 to 1000m in the vertical. As such, small scale interactions such as aerosol radiative forcing, microphysics and impacts on cloud formation -which have large variability within these grid boxes- must all be parameterized to be included in the models. Including these effects in GCMs is important as they affect many aspects of the model's simulation including Hadley circulation, precipitation patterns, and tropical variability. Unfortunately these affects are not well understood and are a large source of uncertainty in GCMs. The aerosol BC which is emitted from combustion of fossil fuels, biofuels and biomass, has large mass absorption in the shortwave causing warming in the adjacent atmosphere. SO4 can increase this absorption when deposited on BC, hence modeling their interactions is important to consider. BC can also act as a cloud condensation nuclei (CCN), meaning water droplets can condense on the particulate causing an increase in cloud formation which can in turn cause cooling by reflecting sunlight. Aerosols such as BC and SO4 have short lifespans in the troposphere -about one week- but through their affect on radiative absorption and cloud formation, can have a large impact on the climate. For more information on BC aerosol modeling and it's climate impacts, the reader is refereed to [11] and [1] respectively.

Comparisons between observational data and models shows that models tend to underestimate BC mass concentrations. In order to address this, this paper applies the ϵ-SV model as mentioned above to estimate CAM4 climate profiles for 2^{20} parameter combinations. The predictions are compared to find the parameter combinations which produce a climate profile closest to the default CAM4. The default CAM4 is used since it has been extensively tested and compared to observations to be considered an accurate model of the current climate. We expect that the ranges obtained from the parameter combinations which produce climates similar to the default CAM4 could serve as estimates for the actual distribution and physical properties of BC and SO4 in the atmosphere.

The layout of this paper consist of an outline of the model data and statistical methods in section 2. The section starts with a somewhat detailed explanation and derivations for the techniques as well as comparisons to other methods used in climate sensitivity analysis. Also, a step by step overview is outlined for determining plausible parameter ranges. Lastly, the section describes how model selection was performed and provides descriptions of the

input parameters and output variables. Section 3 contains the results of the methods as applied to 350 CAM4 experiments with perturbed parameter input values related to BC and SO4. The first part of the section outlines results with regards to parameter importance while the latter part provides results related to constraining uncertainty ranges for these parameters in CAM4.

2 Data and Methods

The gathered data comes from an ensemble of CAM4 runs of 350 cases constituting 350 parameter combinations. A total of 9 parameters were considered relating to BC and SO4 concentrations and distributions in the atmosphere, including those related to cloud formation. The latter were considered since aerosols such as BC and SO4 can have indirect effects on cloud formation, yet these effects are not currently well modeled in GCMs. These two aerosols can have indirect effects on a climate system such as acting to reduce cloud fraction as well as act as cloud condensation nuclei. Varying parameters relating to these indirect effects means we can account for changes in cloud formation. Compared are output values of 14 variables concerning radiative fluxes, precipitation and cloud distribution.

2.1 Parameter Importance

A very versatile yet simple method for determining feature importance for statistical modeling is to employ random forests. The RF method has become increasingly popular in the sciences for its ability to model highly complex interactions and deal with highly correlated predictor variables. When using the method for model building, few assumptions are made about functional form or data types and indeed, this is one of the large benefits of RF. On the other hand, when using RF for determining variable importance, major biases may result and hence care should be taken when using this method. High correlation of predictor variables may produce unreliable results when determining the importance of these variables [20]. Also, when predictor variables are of different types, i.e. categorical an continuous, or the categorical variables have differing numbers of classes, variable importance may again be unreliable [6]. These problems can be overcome and methods are outlined in the two previous papers. For the work in this paper, the predictor variables (input parameters of the model) are all continuous and independent so the importance results can be seen as reliable. A quick outline of RF is given as well as its application to variable importance.

Decision Trees

The general concept of decision trees can be applied to both regression and classification problems, the former will be the focus going forward in this paper. An example is given in figure 1 with two predictor variables x_1 and x_2. First assume the tree is given and the goal is to predict the value for an observation with predication variable values $x_1 = 5.4$ and $x_2 = 1.1$. Since $x_1 \geq 5$, move down the right *branch*, and so on for x_2 and x_1 again, the predicted response value is at the $leaf$ or *node* labeled 1.5. These *split points* partition the $x1, x2$ plane into rectangles, which assign a value to each region.

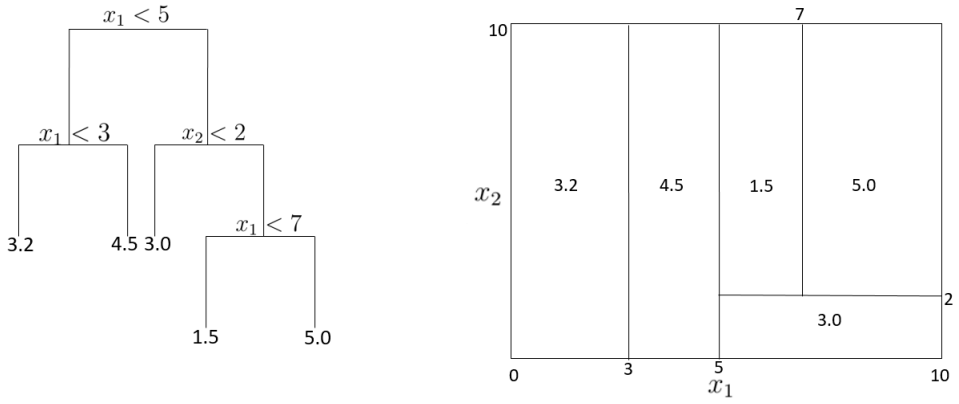


Figure 1: Left: Example decision tree. Right: Feature space partitioned by the decision tree.

To construct a tree, consider the training observations denoted as $\{y^i, x^i\} \in \mathbb{R} \times \mathbb{R}^d$, $i = 1, 2, ..., N$. The goal is to partition $\mathbb{R}^d$ and assign the prediction value to be the average of the response variables whose predictors are located in the partitioned region. To illustrate what this means, suppose in the previous example, only two training observations lye in the left most region in figure 1 and have values $\{3.4, 2, 5\}$ and $\{3.0, 1, 3.5\}$. If we have an observation with $x_1^i = 2$, $x_2^i = 8$ the predicted value would be $y^i = 3.2$. For regression trees, the aim is to find the partition of the space which produces the lowest sum of squared residuals (RSS) over the training data. There are clearly many ways the space can be partitioned and comparing all possible combinations is not computationally feasible. Instead, a *recursive binary splitting* approach is applied where at each level in the tree, the partition is split into two boxes $R_1(j, s) = \{x^i | x_j^i < s\}$ and $R_1(j, s) = \{x^i | x_j^i \geq s\}$ for $s \in \mathbb{R}$. In the two dimensional example, the first step has $j = 1$ since the split is done on the first predictor, and $s = 5$ since this is the value found to be optimal to split on. The boxes $R_1(1, 5)$ and $R_2(1, 5)$ are the left and right halves of the plane at $x_1 = 5$ respectively. So at each step we seek the values of j and s which minimize:

$$\sum_{i:x^i \in R_1(j,s)} (y^i - \hat{y}_{R_1})^2 + \sum_{i:x^i \in R_2(j,s)} (y^i - \hat{y}_{R_2})^2 \tag{2}$$

The number of steps depends on a tuning parameter fixing the number of training observations per leaf. The algorithm can be executed efficiently, but may not be the most optimal solution since the optimization is performed separately at each node instead of simultaneously minimizing the RSS across all predictors. Though decision trees are very interpretable, they do not by themselves prove to be very good models in real life situations. Instead, bootstrapping is applied across an ensemble of trees which produces the RF algorithm.

Random Forest

Decision trees suffer from high variance, meaning that splitting the data into different test and training sets can produce very different results. Bootstrapping works exceptionally well in reducing the variance of predictions when applied to decision trees. Samples are

randomly drawn from the data with replacement to create B bootstrapped samples. This is demonstrated in figure 2 with 3 bootstrapped samples of $N = 6$ observations. A decision tree is trained over each bootstrapped sample to produce B trees.

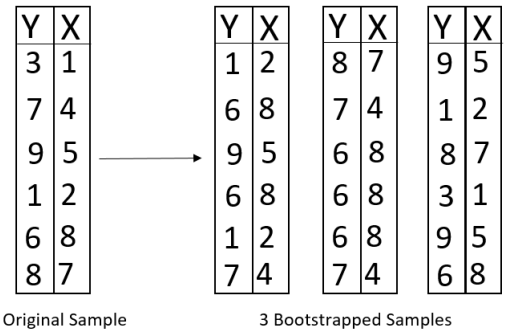


Figure 2: Three bootstrap samples of an original sample of size six.

Denote the b^{th} tree as $f_b(x)$ for $b = 1, 2, ..., B$. The predicted value of the response is simply the average predicted values over the B trees. Hence the variance of the prediction is σ^2/B where σ^2 is the variance of the individual trees. Further, it can be shown [13] that the variance of the prediction across B trees can be decomposed as follows:

$$Var\left(\frac{1}{B}\sum_{b=1}^{B} f_b(x)\right) = \rho\sigma^2 + \sigma^2\frac{1-\rho}{B} \tag{3}$$

where ρ is the correlation between trees. For large enough B, the second term in (3) is insignificant, while the first term can be reduced by reducing ρ. To achieve a low correlation between trees, at each split in a tree, a random[2] subset of p predictors are chosen out of the total d predictors. The predictor to be split on is chosen from this subset instead of all available predictors. This is repeated at each split, usually until each leaf has a predetermined number of training values.

Permutation Importance

The Random Forest algorithm inherently produces an easy way to measure variable importance. Referring to figure 2, the last bootstrapped sample does not contain the case $(7, 4)$, while the first two samples do. On average, about one third of the bootstrapped samples will not contain a particular observation, these are the out of bag (OOB) samples. This is clear since the probability of a particular bootstrapped sample not containing a particular observation (y^i, x^i) is $(1 - 1/N)^N$ where N is the total number of observations. For large N, say > 20 , $(1 - 1/N)^N \approx e^{-1}$ which is about $1/3$. As a result, splitting the data into training and test samples is not required. Instead, for all trees $f_b(x)$, $b = 1, 2, ..., B$, we take all OOB cases for tree b and produce a prediction using $f_b(x)$ and then calculate the MSE.

[2]This paper uses the word random but it should be understood that pseudo-random is meant when referring to numbers generated by computer algorithms.

OOB samples can be used further for determining variable importance by way of the permutation importance measure. The idea is that by randomly permuting the predictor variable x_j for all samples $i = 1, 2, ..., N$, it should no longer be useful in estimating the response y. If the procedure for calculating the MSE is again repeated with this permuted set, then the MSE should rise substantially if x_j is an important predictor. The average increase in MSE over all B trees is taken and scaled by the standard deviation of this difference in MSE, denoted σ_j, to produce a measure of importance for predictor j. By default the randomForest package [3] in R outputs the scaled variable importance which is convenient for plotting and other visualization. On the other hand, [6] shows that using the scaled importance for the purposes of statistical testing should not be done. Instead setting '$scale = FALSE$' to produce the raw importance scores will allow for reliable statistical testing of significance. Using the RF permutation importance algorithm has been shown to be unbiased when the predictors are continuous [6] and uncorrelated [13], as is the case for the parameters considered in this paper. This RF algorithm does not assume a linear model and is able to better model complex relationships between GCM model inputs and outputs, and provide a more reliable measure of parameter importance.

Let the set containing all OOB cases for tree b be denoted as Ω_b, $|\Omega_b|$ be the number of cases in Ω_b and $x^i_{\pi_j}$ be observation i after permuting predictor j across all observations. Then the raw variable importance VI_j for predictor j is calculated as:

$$VI_j = \frac{1}{B}\sum_{b=1}^{B}\left[\sum_{i\in\Omega_b}\frac{[(y^i - f(x^i_{\pi_j})) - (y_i - f(x^i))]}{|\Omega_b|}\right] \tag{4}$$

VI_j is a random variable even for a fixed sample since the bootstrapped samples as well as the choice of subset to split at each branch is random. From (4), the null hypothesis that variable j is unimportant as a predictor of the response variable can be stated as:

$$VI_j \overset{a.s.}{\sim} N(0, \frac{\sigma_j^2}{B}) \tag{5}$$

by the Central Limit Theorem. A simple z-test can be used to determine significance levels. A main advantage of the random forest permutation importance measure is that as compared to univariate importance methods, the permutation test accounts for the impact of the predictor both individually as well as in multivariate interactions.

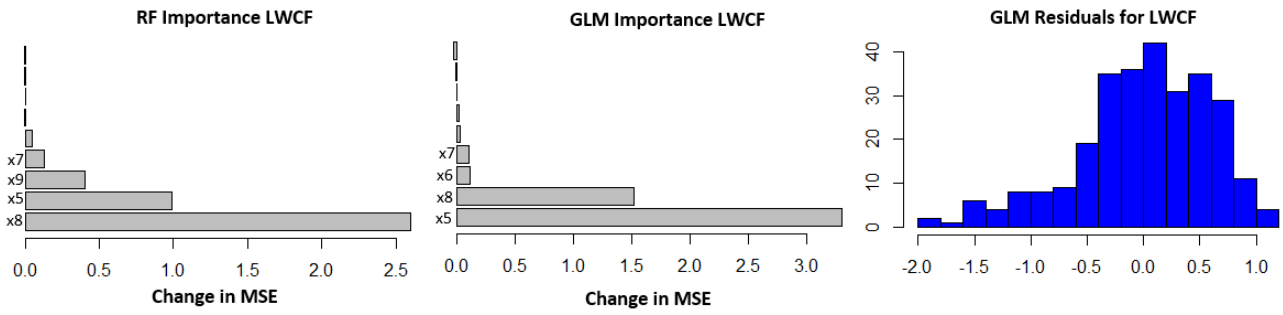


Figure 3: Comparison of GLM (MLR) and RF importance predictions for global mean LWCF difference between perturbed and default CAM4

For example, using the data in this paper, the RF permutation method identified some predictors as important which were not identified in an ANOVA analysis of a recursively fitted GLM. Figure 3 compares importance results for global mean LWCF difference (perturbed-default). The RF algorithm is seen to identify different parameters compared to the MLR method. The non-normal distribution of the residuals shows that a linear model is not optimal. Indeed RF is likely the correct importance measure since no assumptions of linearity are made, and the predictors are uncorrelated and continuous. The MLR method has clear limitations in measuring importance and so the RF permutation algorithm is used here.

2.2 Quasi-Monte Carlo Sampling

In order to efficiently sample the parameter space, a natural method to employ is QMC or low discrepancy points. Intuitively, by discrepancy what is meant is the size of the gaps left between sampling points in the sampling space. The goal is to distribute the sampled points along the d-dimensional unit hypercube $[0,1)^d$ as uniformly as possible and in a computationally efficient way. Note that these points can be scaled to fit any desired interval. More formally, discrepancy is defined as [7]:

$$D(A,N) = \left| \frac{\#\{x_i \in A\}}{N} - vol(A) \right|$$

where $A \subseteq [0,1)^d$ is a box, $\#\{x_i \in A\}$ is the number of sample points contained in A and $vol(A)$ is the volume of A. As an example, Figure 4 is an arbitrary sample of 8 points in a 2 dimensional space where $D(A,8) = 4/8 - 1/2 = 0$.

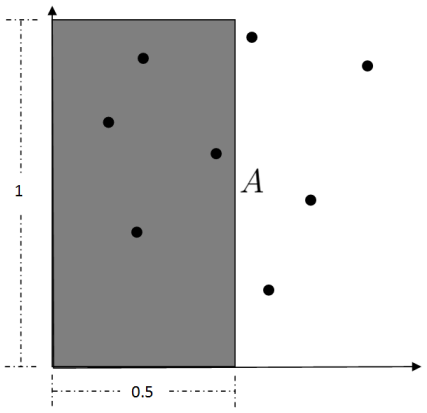


Figure 4: Measuring discrepancy for a sample of 8 points in a 2D space.

The discrepancy is dependent on the choice of A and the worst case discrepancy is defined as:

$$D^*(N) = \sup_A |D(A, N)| \tag{6}$$

called *star discrepancy*. Star discrepancy can be seen as a metric for a how well distributed a set of points are in high dimensional space. It can be shown that a simple grid sampling technique using the Cartesian product along d dimensions is usually not ideal for low discrepancy sampling. Large rectangular gaps are created and points overlap when projected into lower dimensions, meaning some sampled points are essentially wasted. The three dimensional Cartesian product of 2^3 points in Figure 5 shows the sampled points are separated onto planes by this method. Their projection onto any two dimensional subspace produces only 2^6 unique points as seen in Figure 6.

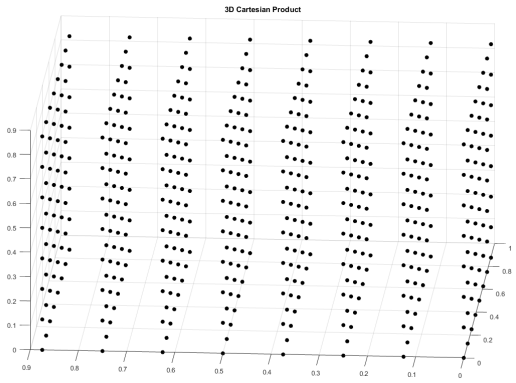


Figure 5: Cartesian product of 2^3 points in 3D space

If for example, along each dimension, 2^k points are sampled, the Cartesian product produces $N = 2^{kd}$ grid points. Adding a single parameter to the space increases the number of required sampling points by a factor of 2^k. For a large number of dimensions in the

parameter space, this can quickly drain computational resources and become impossible to sample sufficiently.

Other methods such as MC or pseudo-random sampling and Latin Hypercube (LHS) techniques are shown to have higher discrepancy for large N as compared to QMC [14]. Referring to Figure 6, the four sampling techniques are demonstrated in two dimensions.

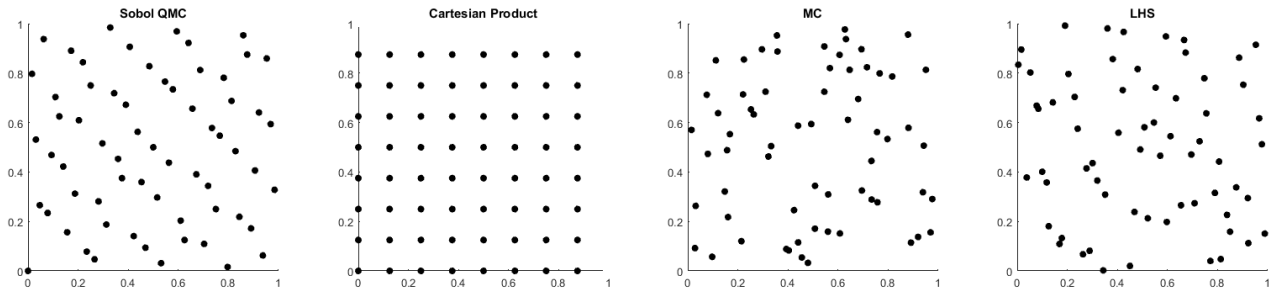


Figure 6: Sampling methods in 2D space. Left to Right: quasi-Monte Carlo, Cartesian Product, Pseudo-Random Numbers, Latin Hypercube.

Since QMC sampling refers to any deterministic sampling of points, there are many methods for generating samples. This paper uses the Sobol sequence to generate these points since this is what is used in [14], which provides a comparison of the sampling methods. The MC and LHS generated points tend to cause clumping and large gaps in the space in Figure 6. The QMC method on the other hand generates a better distribution and also does well when extended to higher dimensions.

2.3 Support Vector Regression

To predict the output variables from ensemble runs of the chosen climate models, an ϵ-SV regression model is trained on values obtained from 350 runs of CAM4. The ϵ-SV method in [21] is adapted from the Support Vector Machine classifier first introduced in 1992 [2]. For a given set of training data of size N given by $\{(y_1, x_1)^T, (y_2, x_2)^T, ..., (y_N, x_N)^T\}$ where $(y_i, x_i)^T \in \mathbb{R} \times \mathbb{R}^n$, the aim is to define a function $y = f(x)$ which estimates the training response values within an error of at most $\epsilon > 0$ for all training values. To illustrate the method, the linear case

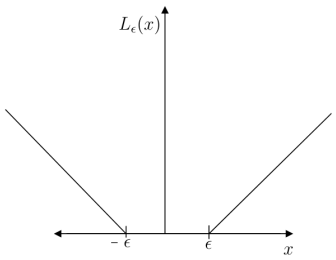
$$f(x) = \beta^T x + \beta_o \tag{7}$$

is considered, followed by the generalization to the non-linear case. The restriction of the error being at most ϵ implies $|y_i - \beta^T x_i - \beta_o| \leq \epsilon$ for $i = 1, 2, ..., N$. Note that for a given ϵ, it is not guaranteed that there exists such a function satisfying the error constraint. To allow for some observations to have error greater than ϵ and hence guarantee a solution, introduce a convex loss functions $L_{\epsilon i}$. In the case of ϵ-SV, the loss function is given by:

$$L_{\epsilon i}(\beta, \beta_o) = \begin{cases} 0, & |y_i - \beta^T x_i - \beta_o| \leq \epsilon \\ |y_i - \beta^T x_i - \beta_o| - \epsilon, & |y_i - \beta^T x_i - \beta_o| > \epsilon \end{cases}$$

The function is convex since it is the composition of a convex function:

$$L_\epsilon(x) = \begin{cases} 0, & |x| \leq \epsilon \\ |x| - \epsilon, & |x| > \epsilon \end{cases}$$



with an affine function $(y_i - \beta^T x_i - \beta_o)$. $L_{\epsilon i}$ corresponds to the amount by which the training observation x_i is outside of the tube of width 2ϵ around $f(x)$. Figure 7 depicts this visually on a one dimensional feature space with ξ denoting the value of the loss function for the observation with error greater than ϵ.

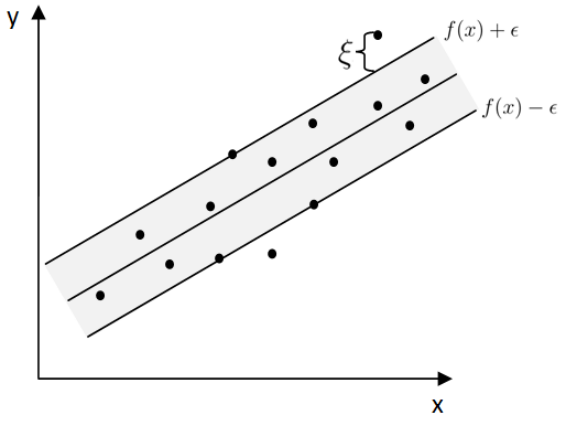


Figure 7: ϵ-SV with a single feature and one observation error exceeding ϵ by ξ.

Optimization

In this section, derivations of the $\epsilon-$SV method are based on those in [2], [8], [18] and [21]. This paper provides justifications and explications for each step beyond what is given in the above sources. Sources for these 'filled in' gaps can be found in standard texts on optimization.

Note that a function given by (7), which satisfies the given error constraint with the ϵ-intensive loss function, need not be unique. A desirable property of (7) which can be imposed is to minimize $||\beta||$. If x is drawn i.i.d., then by minimizing $||\beta||$, the variance of the predictions from (7) is clearly minimized. In other words, given the choice between functions satisfying the above constraints, preference is given to the one which is less sensitive to noise in the data, i.e. perturbing the feature space slightly should lead to an as small as possible change in the predicted value. Finally, the notation used in [21], which will be used here, is to represent the ϵ-intensive loss function in terms of the variables ξ_i, ξ_i^*, denoting the amount of slack given above and below the tube to observation i respectively. This optimization

problem can be stated as

$$\begin{aligned} \text{minimize} \quad & \mathbf{F}(\beta, \xi, \xi^*) = \frac{1}{2}||\beta||^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) \\ \text{subject to} \quad & \begin{cases} y_i - \beta^T x_i - \beta_o \leq \epsilon + \xi_i \\ \beta^T x_i + \beta_o - y_i \leq \epsilon + \xi_i^* \\ \xi_i, \xi_i^{(*)} \geq 0 \end{cases} \end{aligned} \tag{8}$$

where C is a constant parameter which controls how harshly errors impact the minimization of the objective function. For instance, larger values of C imply larger penalties for residuals greater than ϵ and the scheme therefore produces a steeper (less flat) function. The optimization problem (8) is convex (since $\mathbf{F}$ and the constraints are the sum of convex functions in β and β_o) and hence can be efficiently solved. The Lagrangian of (8) is given by:

$$\begin{aligned} \mathbf{L} = \frac{1}{2}||\beta||^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) + \sum_{i=1}^{N}\lambda_i(y_i - \beta^T x_i - \beta_o - \epsilon - \xi_i) \\ + \sum_{i=1}^{N}\lambda_i^*(\beta^T x_i + \beta_o - y_i - \epsilon - \xi_i^*) + \sum_{i=1}^{N}(\nu_i\xi_i + \nu_i^*\xi_i^*) \end{aligned} \tag{9}$$

where $\lambda^{(*)}$ ($\lambda^{(*)} = \lambda, \lambda^*$) and $\nu^{(*)}$ and the Lagrange multipliers. In most cases, to solve (8) it is more efficient to formulate the problem in terms of the dual problem. The dual function is defined as the infimum of the Lagrangian over the feasible β, β_o and $\xi^{(*)}$:

$$\mathbf{G}(\lambda^{(*)}, \nu^{(*)}) = \inf_{\beta, \beta_o, \xi^{(*)}}(\mathbf{L}(\beta, \beta_o, \xi^{(*)}, \lambda_i^{(*)}, \nu_i^{(*)})) \tag{10}$$

This leads to the standard dual problem:

$$\begin{aligned} \text{maximize} \quad & \mathbf{G}(\lambda^{(*)}, \nu^{(*)}) \\ \text{subject to} \quad & \lambda_i^{(*)}, \nu_i^{(*)} \geq 0 \end{aligned} \tag{11}$$

Since we require the Lagrange value to be positive by (11), (9) is a linear combination of convex function, and hence convex under this constraint. It follows that a sufficient condition for finding (10) is to set $\nabla\mathbf{L} = 0$. The result of this calculation yields:

$$\partial_{\beta_o}\mathbf{L} = \sum_{i=1}^{N}(\lambda_i - \lambda_i^*) = 0 \tag{12}$$

$$\partial_{\beta}\mathbf{L} = \beta - \sum_{i=1}^{N}(\lambda_i - \lambda_i^*)x_i = 0 \tag{13}$$

$$\partial_{\xi_i^{(*)}}\mathbf{L} = C - \lambda_i^{(*)} - \nu_i^{(*)} = 0 \tag{14}$$

Plugging (12),(13) and (14) into (9), the $\xi_i^{(*)}$ are canceled and β_o drops from the equation due to (12). From (14), since $\lambda_i^{(*)}$ and $\nu_i^{(*)}$ are non negative, this implies the constraint $0 \leq \lambda_i^{(*)} \leq C$. The dual problem is hence given by:

$$\begin{aligned} &\text{maximize} \;\; \mathbf{G}(\lambda^{(*)}) = -\frac{1}{2}\sum_{i,j=1}^{N}(\lambda_i - \lambda_i^*)(\lambda_j - \lambda_j^*)x_i^T x_j - \epsilon\sum_{i=1}^{N}(\lambda_i + \lambda_i^*) + \sum_{i=1}^{N} y_i(\lambda_i - \lambda_i^*) \\ &\text{subject to} \;\; \begin{cases} 0 \leq \lambda_i^{(*)} \leq C \\ \sum_{i=1}^{N}(\lambda_i - \lambda_i^*) = 0 \end{cases} \end{aligned} \tag{15}$$

More generally, minimizing a differentiable $\mathbf{F}(z)$ with respect to constraints $f_i(z) \leq 0$ for $i = 1, 2, ...n$ and denoting its dual by $\mathbf{G}(\lambda)$, it is clear that $\mathbf{G}(\lambda) \leq \mathbf{F}(z)$ for any feasible z and λ since:

$$\mathbf{G}(\lambda) = \inf_z \mathbf{L}(z, \lambda) = \inf_z(\mathbf{F}(z) + \sum_{i=1}^{n}\lambda_i f_i(z)) \leq (\mathbf{F}(z) + \sum_{i=1}^{n}\lambda_i f_i(z)) \tag{16}$$

Since $\lambda_i f_i(z) \leq 0$, $\mathbf{G}(\lambda)$ is a lower bound for $\mathbf{F}(z)$, in particular for optimal $\tilde{z}$ and $\tilde{\lambda}$: $\mathbf{G}(\tilde{\lambda}) \leq \mathbf{F}(\tilde{z})$. At the optimal values, to achieve equality $\mathbf{G}(\tilde{\lambda}) = \mathbf{F}(\tilde{z})$, i.e. a zero duality gap, from (16) it is clear that we require $\tilde{\lambda}_i f_i(\tilde{z}) = 0 \; \forall \; i$. The sufficient conditions derived above, which produce a zero duality gap for a convex optimization problem, are generally stated as the Karush-Kuhn-Tucker (KKN) conditions:

1) $f_i(z) \leq 0$ for $i = 1, 2, ..., n$
2) $\lambda_i \geq 0$ for $i = 1, 2, ..., n$
3) $\lambda_i f_i(z) = 0$ for $i = 1, 2, ..., n$
4) $\nabla \mathbf{L}(z, \lambda) = 0$

It is also easy to see from (16) that if $\mathbf{G}(\tilde{\lambda}) = \mathbf{F}(\tilde{z})$ for feasible $\tilde{z}$ and $\tilde{\lambda}$, then $\tilde{\lambda}_i f_i(\tilde{z}) = 0$ for all i.

In the $\epsilon-$SV case it is required that:

$$\begin{aligned} \lambda_i^{(*)}(y_i - \beta^T x_i - \beta_o - \epsilon) &= 0 \\ \nu_i^{(*)}\xi_i^{(*)} &= 0 \end{aligned} \tag{17}$$

for all i. For the first condition in (17) either $\lambda_i^{(*)} = 0$ or $y_i = \beta^T x_i + \beta_o \pm \epsilon$, i.e. the response variable y_i is on the boundary of the envelope depicted in Figure 7. Substituting (13) into (7):

$$f(x) = \sum_{i=1}^{N}(\lambda_i - \lambda_i^*)x_i^T x + \beta_o \tag{18}$$

Since most $\lambda_i^{(*)}$ are zero, the final solution is a linear combination of training points in the feature space which lye on the boundary of the margin, these are termed the *support vectors*. In Figure 7 for example, there are three support vectors, one on the upper and two on the lower margin boundary.

Substituting (14) into (17) implies $(C-\lambda_i^{(*)})\xi_i^{(*)} = 0$ for all i and in particular, $\lambda_i^{(*)} = C$ for values outside of the tube around $f(x)$. Therefore the constant β_o can be found from a constraint in (8):

$$\beta_o = y_i - \beta^T x_i + \epsilon, \qquad 0 < \lambda_i < C$$

Kernels

The extension of (7) to the non linear case is accomplished by mapping the feature space to a higher dimensional space through a non linear mapping: $h(x_i) : \mathbb{R}^n \to \mathbb{R}^D$ where $D \geq N$ and possibly $D = \infty$. The goal is to choose a map $h(\cdot)$ such that the relationship between the features x_i and response y_i is close to linear. The high (possibly infinite) dimensionality of the new feature space in $\mathbb{R}^D$ may seem computationally unfeasible, but by employing the so called kernel trick, the problem can be solved efficiently. Before explaining this process, first suppose the relationship between $y_i \in \mathbb{R}$ and $x_i \in \mathbb{R}^n$ is best described by a 2^{nd} degree polynomial. The map $h(x) : \mathbb{R}^n \to \mathbb{R}^{\frac{n(n+3)}{2}}$:

$$x_i = \begin{bmatrix} x_{1i} \\ x_{2i} \\ \vdots \\ x_{ni} \end{bmatrix} \quad \to \quad h(x_i) = \begin{bmatrix} x_{1i} \\ x_{2i} \\ \vdots \\ x_{1i}x_{2i} \\ \vdots \\ x_{n-1i}x_{ni} \\ x_{1i}^2 \\ \vdots \\ x_{ni}^2 \end{bmatrix} \tag{19}$$

transforms the feature space into one which is larger in dimension by a factor of $(n+3)/2$. More generally the dimension of the new feature space using a polynomial of degree d is

$$dim(h(x)) = \sum_{d=1}^{D} \binom{n}{d} \tag{20}$$

This type of transformation is typical in many algorithms in order to reduce the problem of model fitting to a linear problem. It is clear that the dimension of the new feature space increases rapidly with both n and D, increasing computation time. Instead, consider (18) in the larger feature space for a general $h(x)$:

$$f(x) = \sum_{i=1}^{N} (\lambda_i - \lambda_i^*) h(x_i)^T h(x) + \beta_o \tag{21}$$

here the explicit form of $h(x)$ is is not required, instead knowing the form of the dot product $h(x_i)^T h(x)$ is sufficient. Denote $\mathbf{K}(u,v) = h(u)^T h(v)$ the *kernel* of the transformation $h(\cdot)$. In general, a kernel is a continuous real valued function such that $\mathbf{K}(u,v) = \mathbf{K}(v,u)$. If also $\mathbf{K}(u,v)$ is square integrable, then it can be expanded in the form:

$$\mathbf{K}(u,v) = \sum_{i=1}^{\infty} \alpha_i \phi_i(u)^T \phi_i(v) \tag{22}$$

where α_i and $\phi(\cdot)_i$ are eigenvalues and eigenfunctions of the integral operator:

$$\int \mathbf{K}(u,v)\phi_i(u)du = \alpha_i \phi_i(v)$$

The kernel in (22) defines an inner product over some complete Hilbert space if $\alpha_i \geq 0$ for all i. This can be guaranteed by ensuring Mercer's condition holds:

$$\int\int \mathbf{K}(u,x)g(u)g(v)dudv > 0$$

for all square integrable $g(\cdot)$. Some examples of commonly used kernels which satisfy Mercer's condition are:

1) d^{th} degree polynomial: $\mathbf{K}(u,v) = (1 + u^T v)^d$
2) Radial basis: $\mathbf{K}(u,v) = exp(-\gamma||u-v||^2)$
3) Neural network: $\mathbf{K}(u,v) = tanh(c_1 u^T v + c_2)$

The Radial kernel is an example of an inner product corresponding to an infinite dimensional space. To see this, let $u, v \in \mathbb{R}^n$ and set $n = 2$ to simplify notation, but it is possible to generalize the following results. Noting that:

$$-\gamma||u-v||^2 = -\gamma \sum_{i=1}^{2} u_i^2 - \gamma \sum_{i=1}^{2} v_i^2 + 2\gamma u_i^T v_i$$

then taking the exponent and expanding the last term using the Taylor series gives:

$$exp(-\gamma||u-v||^2) = exp(-\gamma \sum_{i=1}^{2} u_i^2) exp(-\gamma \sum_{i=1}^{2} v_i^2) \sum_{k=0}^{\infty} \frac{2^k \gamma^k}{k!} \left(u_i^T v_i\right)^k \tag{23}$$

Hence (23) is an inner product on the map $h(\cdot)$ whose components are given by:

$$h(x) = exp(-\gamma \sum_{i=1}^{2} x_i^2)(1, \sqrt{\frac{2^1\gamma^1}{1!}}x_1, \sqrt{\frac{2^1\gamma^1}{1!}}x_2, \sqrt{\frac{2^2\gamma^2}{2!}}x_1^2, \sqrt{\frac{2^2\gamma^2}{2!}}x_2^2, \sqrt{\frac{2^2\gamma^2}{2!}}2x_1x_2, ...)^T$$

So $h(x) : \mathbb{R}^2 \to \mathbb{R}^\infty$ and the radial kernel is the inner product of this mapping. Clearly computing the inner product $h(u)^T h(v)$ explicitly is not possible, but using the exponential representation it can be done with comparatively low computational cost. In the next section, the radial kernel is used as it was found to work best through cross validation experiments.

2.4 Implementation of Procedures

This paper essentially does two things: determine parameter importance for 14 CAM4 variables, and produce plausible bounds on parameter perturbation values for combinations which produce similar climates to the default CAM4. The first is accomplished using RF permutation importance algorithm. For this the randomForest package [3] was used in R. First, 100,000 trees are trained on the same amount of bootstrapped samples on the 350 cases ran on CAM4. This large amount of trees is required for determining variable importance but not generally needed for prediction. The trees are fully grown, and splits are performed on 3 variables for each branch. For regression trees with p predictors, $p/3$ are recommended [8] to be used at each branch split. The RF permutation importance method gives a relative measure of how important each parameter is in increasing the accuracy of predicting the response variable values. Some parameters have importance measures close to zero, but some are statistically significant.

The second goal of the paper is to reduce uncertainty ranges for the parameter values above. A separate SVR model is trained on each of the variables using the 350 cases. The radial kernel is used which has 3 tuning values: C, γ, and ϵ. The package e1071 [15] which is used in this paper for SVR automatically scales the parameters and variables to mean zero and unit variance. This means that value of ϵ -which controls the maximum size of allowable prediction error without penalty- does not need to be chosen separately for each variable and is left to the default of $\epsilon = 0.1$. The other two tuning values are chosen using a grid search method where every combination of $C \in \{2^{-5}, 2^{-3}, ..., 2^{15}\}$ and $\gamma \in \{2^{-15}, 2^{-13}, ..., 2^{3}\}$ are used. These grid values are as recommended in [10]. To ensure robustness, a $80 - 20$ Monte Carlo (MC) cross validation method is used over 10 samples (10 is used as a trade-off between robustness and computation time). That is, for each grid point, the SVR is trained on a random 80% sub-sample of the 350 cases and the remaining 20% is used as the test sample. The average mean squared error (MSE) is taken, and the combination of γ and C which produce the lowest average MSE are chosen.

QMC is used to sample the hypercube defined by the parameter ranges in table 1. As a compromise between sampling the space with a sufficient resolution and computation time, 2^{20} points are taken in the 9 dimensional space. The SVR models are used to predict the values of the output variables, which are then ranked using a skill score. For the differences in global mean values, we first normalize the absolute value of these differences from 0 to 1, denoted m_j, $j = 1, 2, ..., 14$. For each case, the 14 values from 0 to 1 are summed to produce a number from 0 to 14:

$$\bar{M}_i = \sum_{j=1}^{14} m_j, \qquad i = 1, 2, ..., 2^{20}$$

To bound this number from 0 to 1, divide by $\max_i(\bar{M}_i)$ to produce the skill score for the mean difference of case i:

$$M_i = 1 - \frac{\bar{M}_i}{\max_{\mathrm{i}}(\bar{M}_i)}$$

Here a value close to 1 means the SVR model predicts global mean values closer to the default CAM4 for parameter combination i, when compared to skill scores closer to 0. A

skill score for the global variances is created by taking the distance of the variance ratios from 1, denoted v_i. These are summed over the 14 variables to produce:

$$\bar{V}_i = \sum_{j=1}^{14} v_j$$

To produce a value between 0 and 1 for each case i, this value is divided by the maximum over all i denoted $\max_i(\bar{V}_i)$. The skill score for the variance is then:

$$V_i = 1 - \frac{\bar{V}_i}{\max_i(\bar{V}_i)}$$

Again, a value closer to 1 implies a spatial distribution which closer matches that of the default, as compared to a value closer to 0. The skill score for the correlation, denoted C_i is left unchanged since this value is already from 0 to 1, where values closer to 1 imply better matching to the default. The final skill score, SS_i, is the average of the three, producing a number between 0 and 1. For important parameters (defined in the next section), bounds can be determined by analyzing the ranges of these parameters for the cases producing the highest skill scores. The technique for bounding parameter values for GCMs can be summarized as follows:

1) Sample the parameter space using QMC.
2) Using the previous sampled values, run GCM experiments to obtain global means, variance correlation values.
3) Train SVR model for each variable using some form of cross validation method.
4) Obtain large sample of the parameter space using QMC.
5) Use SVR models to produce predicted values.
6) Rank the best combinations using a skill score to determine plausible parameter ranges.

Parameter and Variable Data

The 9 parameter values used are listed in table 1 with their description and sample ranges. The values in these ranges are considered equally likely and are hence sampled uniformly. The 350 combinations for the ensemble runs were chosen using LHS, but QMC would also work.

Table 1: Parameter Descriptions and Sampling Ranges

Label	CAM4 Name	Description	Range
x1	x	Fraction of sulfate mass that is hydrophilic	0 to 1
x2	gamma	Spatial dist. of BC: 0-confined to land, 1-globally uniform.	0 to 1
x3	delta	Scaling factor on total mass of BC.	0 to 40
x4	altitude	Altitude dist. of BC.	0 to 39
x5	cldfrc_rhminl	Min. relative humidity for low stable cloud formation.	0.8 to 0.99
x6	cldopt_rliqocean	Liquid drop size over ocean.	8.4 to 19.6
x7	hkconv_cmftau	Time scale for consumption rate of shallow CAPE.	900 to 14,400
x8	cldfrc_rhminh	Min. relative humidity for high stable cloud formation.	0.65 to 0.85
x9	zmconv_tau	Time scale for consumption rate of deep CAPE.	1,800 to 28,800

The variables listed in table 2 constitute the output variables used as a measure of model response to varying the input parameters.

Table 2: Variable Descriptions

Label	CAM4 Name	Description
1	FNET	Net flux= shortwave - longwave.
2	QRS	Shortwave heating rate.
3	QRL	Longwave heating rate.
4	SWCF	Shortwave cloud forcing.
5	LWCF	Long wave cloud forcing.
6	CLDH	Percent of high cloud.
7	CLDM	Percent of medium cloud.
8	CLDL	Percent of low cloud.
9	FSNT	Net solar flux at TOA.
10	FLNT	Net longwave flux at TOA.
11	SHFLX	Surface sensible heat flux.
12	LHFLX	Surface latent heat flux.
13	PRECT	Tropical precipitation.
14	AEROD_V	Aerosol optical depth.

For the global mean values, the data is in the form of the ensemble mean of the difference

between default and perturbed cases: perturbed-default. Variance data is a ratio given as the ensemble mean of:

$$\frac{\text{global mean variance of perturbed}}{\text{global mean variance of default}}$$

Lastly, the correlation data is the ensemble mean of the global correlation between perturbed and default cases. Naturally, a relative measure of how similar the perturbed and default models are would be to compare which perturbed model outputs produce ensemble means closest to zero and variance ratios and correlation values closest to 1 across all 14 variables. This is quantified using a skill score descried in the next subsection.

3 Results

The techniques outlined in the preceding sections are implemented here to solve the problem of parameter importance and for finding sets of parameter perturbation combinations which produce climates models similar to the default CAM4. It is found that some parameters may not impact climate output variables in all three areas of measure (global means, variance and correlation). This underscores the need to conduct importance testing of variables across all three measures in order to accurately determine the impact of input model parameters on model output variables. To determine the set of viable parameter combinations which produce realistic climates (default CAM4), from the 2^{20} combinations, the top 1,000 which produce the skill score closest to 1 are used to narrow the range of realistic parameter values. For the 2^{20} predictions for each variable, the min, max and medians are listed in table 6. In addition, for each variable, the significant parameters (95% confidence) are listed. Since not all parameters have a large impact on the model output variables, for some parameters, this paper was not able to decrease their uncertainty ranges. Nevertheless, this section presents bounds on most parameter values which are narrower than previously used in other work.

3.1 Parameter Importance Results

The global mean values are taken as the difference between the 350 ensemble runs and the default model values over the 14 variables. The goal is to find the parameter combinations which produce values as close to zero as possible across all 14 variables. The bar plots, figures 8, 9 and 10, depict the relative importance of of each parameter for all 14 variables. The higher the proportion of VI_j the parameter accounts for, the more explanatory power it has for variable j, while the total value of VI_j is not important on its own. In figure 8, parameter x5 shows a significant impact on all 14 variables except 6 and 7. This is expected since x5 relates to low cloud formation and variables 6 and 7 are the percentage of high and medium cloud formation. This shows that there is no significant -in magnitude- indirect effect of minimum relative humidity for low stable cloud formation on higher altitude clouds in CAM4. Instead, variables 6 and 7 are mostly impacted by the minimum relative humidity for high stable cloud formation parameter (x8). Variable 5 and 10 are also highly impacted by x8 which is also expected since medium and high clouds readily absorb outgoing longwave radiation.

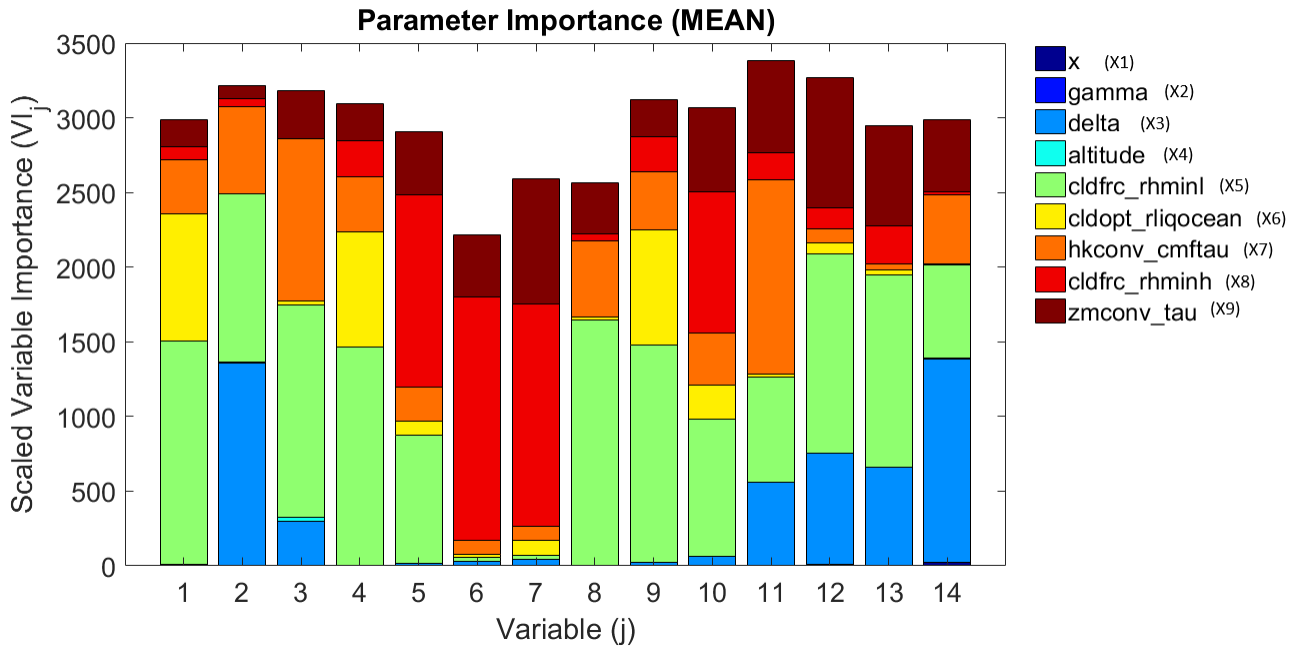


Figure 8: Parameter importance for mean values

Of the parameters directly related to BC (x2-x4), x3 is the only parameter which has a significant impact on any global mean values. This parameter controls the total mass of BC in the atmosphere and has a large measure of importance for variables 2 and 11-14. BC is known to absorbs shortwave radiation and heat the surrounding atmosphere. Variables 2 and 11 describe this type of heating and hence should be impacted by x3. Similarly, variable 14, describing aerosol optical depth, is highly impacted by the perturbing of BC mass in the atmosphere. In contrast to other aerosols which typically absorb more blue light than red light, BC strongly absorbs light at all visible wavelengths. Hence BC acting as the largest factor affecting global mean aerosol optical depth, relative to the other parameters considered in this paper, is consistent with what is expected from what is known about BC. Lastly, BC has a relativity large importance measure for variable 13, tropical precipitation. Generally, studies have shown that atmospheric heating by BC can have significant impacts causing global reductions in global precipitation. For more detailed information see [1] and [12] for comprehensive reports on the global impacts of BC.

Figure 9 is a bar plot of the importance measures for the mean global variances of the same 14 CAM4 model output variables. The variance ratio is a measure of how well the global distributions of of the perturbed model matches that of the default CAM4. This is an important measure since it is possible for example, for mean global high cloud percentages to be the same for two climate profiles, but for the high clouds to be distributed much differently in the two models.

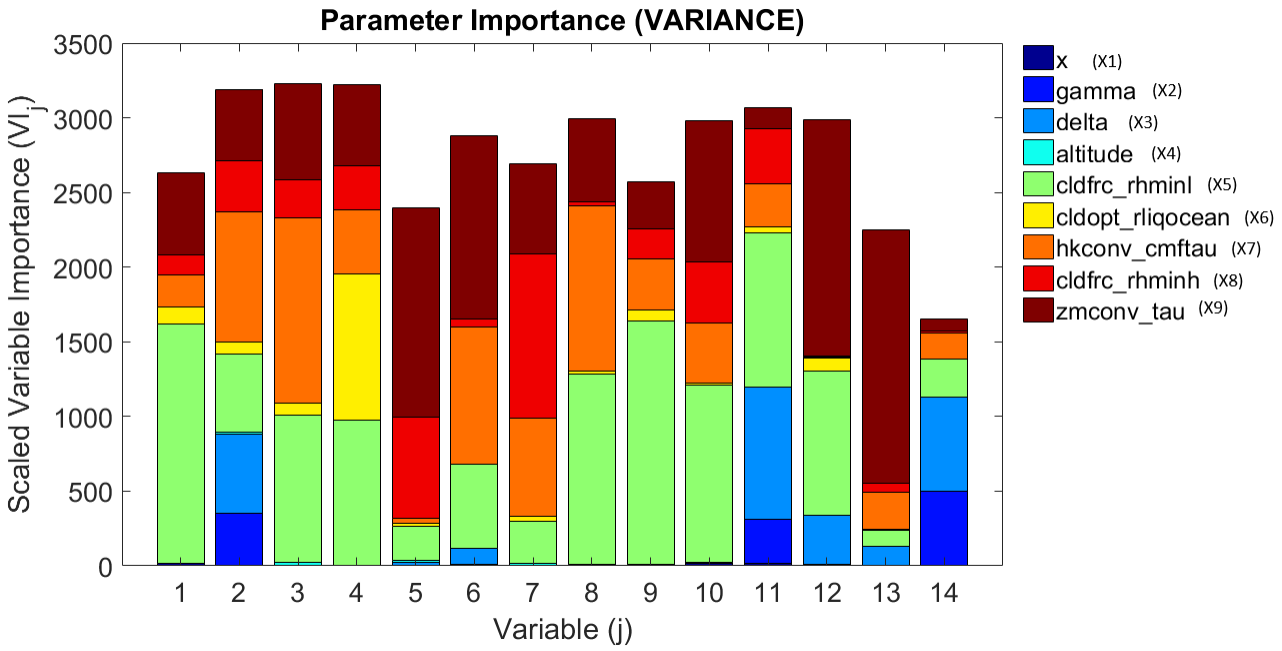


Figure 9: Parameter importance for variance values

In contrast to figure 8, parameter x2, the spatial distribution of BC has a relatively high importance for some variables. The spatial distribution of BC affects variables 2, 11 and 14 in a relatively significant proportion. The distribution of BC affects where solar radiation is absorbed and where the atmosphere is heated, hence the distribution of variable 2, the shortwave heating rate, is also affected. Variable 11, the surface sensible heat flux, which is the transfer of heat energy from the Earth's surface to the atmosphere, is impacted by whether BC is distributed mostly over the land or water since absorption and reflection rates of solar radiation vary over these two surfaces. Finally, the aerosol optical depth, variable 14, also varies significantly depending on the BC distribution since BC greatly absorbs solar light.

Measuring correlation is also an important metric for how similar the perturbed and default CAM4 models are. This mean global correlation is used to determine whether the global mean variances of the output variables are of the same sign more often or less often across regions of the globe. In figure 10, with regards to the parameter affecting BC in the atmosphere, again x2 and x3 are important for variables 2, 11 and 14. Parameter x9 has the largest impact across all variables, even more so than for the global variances. It should be noted that parameters x1 and x4 do not have an important impact on the mean, variance or correlation values of any of the 14 variables. This was not an expected outcome and should be explored further. They do however show a statistically significant impact on the variance and correlation of some variables (see appendix).

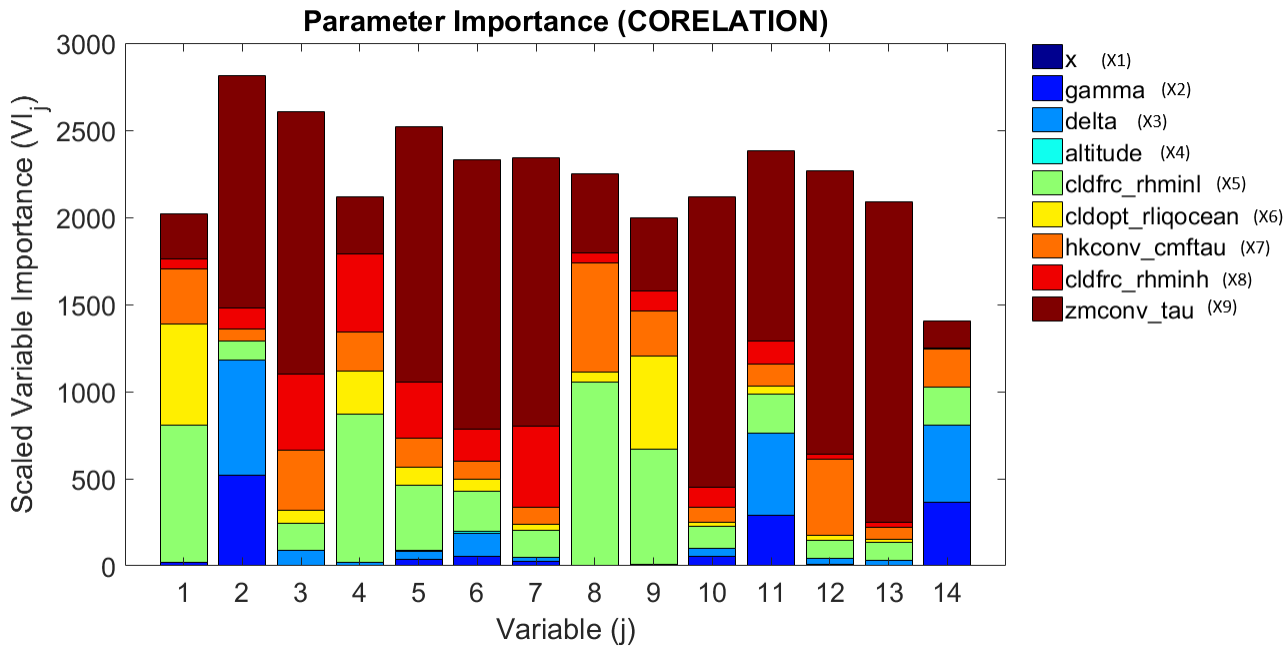


Figure 10: Parameter importance for correlation values

3.2 Plausible Ranges of Parameters

The 350 ensemble runs of CAM4 serve as the training sets for the ϵ-SV models of the 14 output variables. From the 2^{20} parameter combinations, a predicted values for each variable is given by the SVR model for the global mean, variance and correlation. Summary statistics are given in tables 6, 7 and 8 in the appendix. This produces 3×14 ϵ-SV models which are chosen by a parameter grid search for C and γ through MC cross validation. Using the skill score, the top 1000 cases are ranked according to which combinations of parameter values produce global means close to the default CAM4, variance ratios close to 1 and correlations close to 1.

The histograms of the parameter values for these best 1000 cases are given in figure 11. For parameters x1,x2 and x4, the distribution is uniform across the entire range they were sampled. This is an expected result for x1 and x4 since the permutation importance method showed them to be unimportant features for any of the output variables. Their uniform distribution also suggests that they are unimportant. For the parameter x2, the permutation importance algorithm identifies the parameter as important for certain cases, as an example, for the variance of variable 14. It is possible that for any value of x2, the other parameters can be chosen in such a way as to offset its impact on the climate. This is just one possibility and should be investigated further. An interesting result from figure 11 is the distribution of x3.

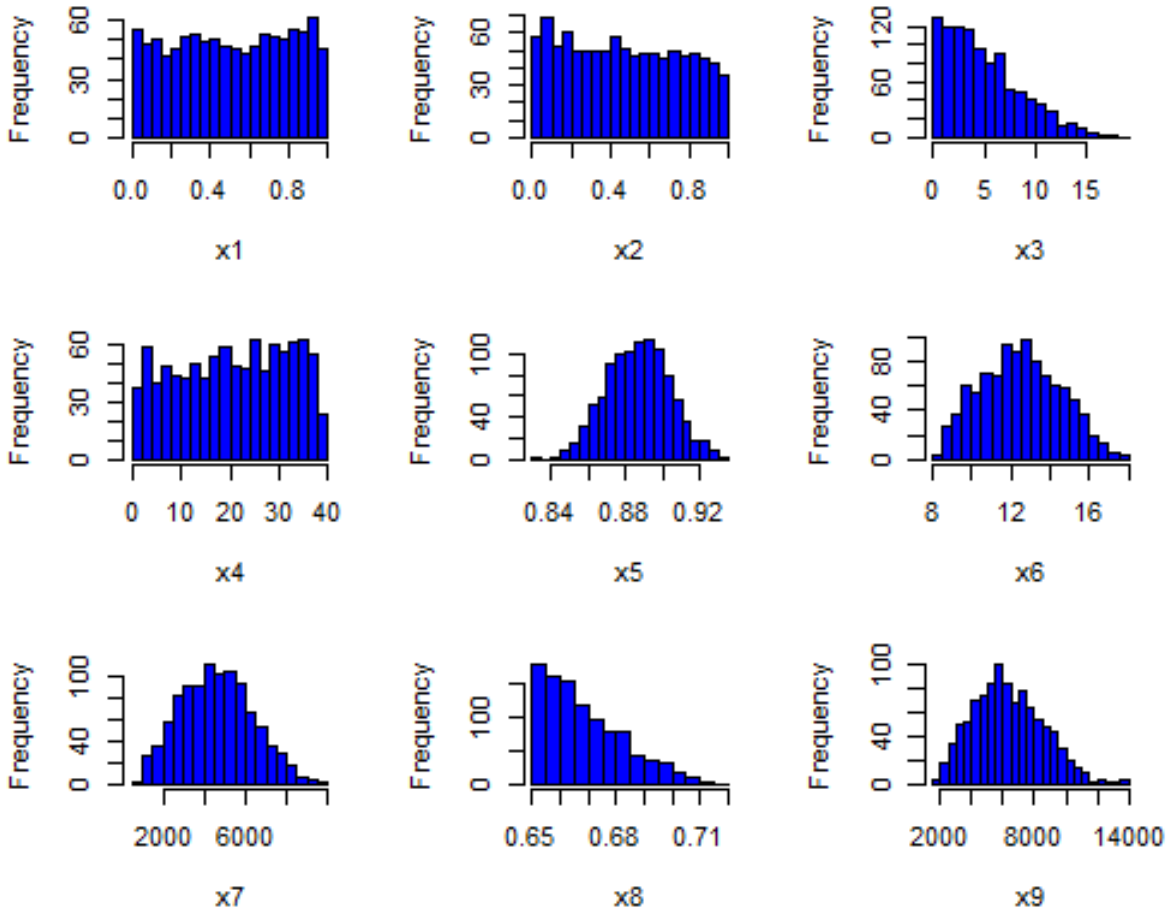


Figure 11: Histogram plots of parameter ranges for top 1,000 parameter combinations with highest skill scores.

It suggests that lower levels of BC mass are more likely to produce the current climate profile in CAM4. This may give an estimate of the plausible amount of BC currently in the atmosphere. Other parameter values show considerable reduction of the plausible ranges. For example, parameter x5 takes on values from 0.8 to 0.94 in our CAM4 runs, but the top 1000 cases are restricted between about 0.84 to 0.94. The distribution of x8 may suggest that the uncertainty range suggested in the literature may be too narrow, and values of x8 should be taken below 0.65. The other parameters seems to be fairly normally distributed, but their ranges are also constrained relative to the initial sampled values. Figure 11 provides a range of plausible parameter values which can be used to run CAM4 simulations with aerosol parameters. The next step in this research is to choose a sub sample of the 1000 parameter combinations and actually run CAM4 to verify that the GCMs produce climates similar to the default CAM4.

4 Conclusion

The techniques outlined together could be used as a standard approach for bounding parameter uncertainty ranges in GCMs. Currently, GLM is widely used in the literature [4], [17], [22] to achieve measures of parameter importance. This paper suggests using the RF permutation method over the other importance measures compared in this paper. We have found that using ANOVA method with MLR can produce different results when compared using RF for parameter importance. From [6] and [20], RF permutation importance should provide a reliable measure of importance. In addition, considering the assumption of linearity is not required for RF, the RF method is likely to be the correct measure of importance for a highly nonlinear system such as most climate models.

The systematic procedure outlined in this paper for constraining parameter uncertainty ranges is argued to be well adapted to the problem. The number of sampling points can be controlled easily using QMC and generally have lower discrepancy as compared to other methods such as LHS, MC and the Cartesian product. The paper has shown that the ϵ-SV method generally produces lower MSE when compared with the commonly used MLR technique, refer to figures 3, 4 and 5 in the appendix. The computational complexity has been quite manageable, even for 2^{20} samples, and its improvement in MSE is likely worth the increase in computation time for most future studies in the field of climate modeling. A large amount of predicted values can be produced and ranked relatively quickly, which is ideal for the problem explored in this paper. Further study should be done to examine the impact of the choice of skill score on the parameter ranges.

The work in this paper also suggests that BC mass in the atmosphere is likely near the lower bound of current estimates. More accurate physical measurements are needed but the results here can be used as a guide for plausible BC related parameter ranges for CAM4. Interestingly, the paper was not able to constrain the altitude distribution ranges of BC in any way. This should also be analyzed further. Since SO4 is known to impact the CCN properties of BC, other parameters related to SO4 may also need to be explored. If one wishes to reproduce the current climate in CAM4 along with BC related parameters, using the parameter values listed in table 9 in the appendix should provide such a climate profile.

5 Appendix

Table 3: ϵ-SV Tuning Values (MEAN)

	C	γ	ϵ-SV MSE	MLR MSE
AEROD_V	128	0.007813	1.40E-06	8.25E-06
FNET	2048	0.00195	0.4540698	0.8165151
QRS	32	0.00781	1.67E-06	4.68E-06
QRL	32	0.03125	1.22E-05	2.16E-05
SWCF	32768	0.00195	0.7672817	2.514795
LWCF	2048	0.00781	0.118944	0.4437511
CLDH	8192	0.00195	0.1235529	0.2576635
CLDM	2048	0.00781	0.163513	0.5466679
CLDL	8192	0.00195	0.2132523	0.8729098
FSNT	32768	0.00195	0.8197897	2.554843
FLNT	2768	0.00781	0.2960727	0.9688627
SHFLX	32	0.00781	0.026706	0.0431452
LHFLX	32	0.03125	0.1391119	0.2508728
PRECT	32	0.03125	0.0007351	0.0014376

Table 4: ϵ-SV Tuning Values (VAR)

	C	γ	ϵ-SV MSE	MLR MSE
AEROD_V	512	0.00781	0.00105	0.00956
FNET	512	0.00781	0.00032	0.0007
QRS	8192	0.00195	0.00014	0.00041
QRL	32	0.03125	0.00053	0.00079
SWCF	8192	0.00195	0.00243	0.01048
LWCF	32768	0.00781	0.00224	0.00605
CLDH	32	0.03125	0.00051	0.00165
CLDM	512	0.00781	0.00094	0.00187
CLDL	128	0.00781	0.00066	0.00156
FSNT	512	0.00781	0.00018	0.00056
FLNT	32768	0.00049	0.00013	0.00023
SHFLX	32	0.00781	0.00032	0.00033
LHFLX	2	0.03125	0.00032	0.00035
PRECT	512	0.00195	0.00188	0.00351

Table 5: ϵ-SV Tuning Values (COR)

	C	γ	ϵ-SV MSE	MLR MSE
AEROD_V	32	0.03125	3.73E-05	0.00031139
FNET	8192	0.00781	3.34E-06	0.00021213
QRS	128	0.00781	6.87E-06	1.11E-05
QRL	32	0.00781	1.22E-05	2.05E-05
SWCF	32	0.03125	0.00010142	0.00038073
LWCF	512	0.00781	3.94E-05	4.25E-05
CLDH	32	0.00781	2.14E-05	1.99E-05
CLDM	32	0.00781	1.85E-05	2.13E-05
CLDL	512	0.00781	2.76E-05	0.00014093
FSNT	32	0.03125	1.34E-06	4.58E-06
FLNT	8	0.03125	3.72E-06	4.33E-06
SHFLX	2048	0.00195	7.51E-06	1.11E-05
LHFLX	8	0.03125	3.89E-06	4.13E-06
PRECT	2	0.03125	0.00011658	0.00010568

Table 6: Variable Ranges (MEAN)

#	Name	Min	Max	Median	Sig.
1	FNET	-29.2335	22.8389	0.8425	x2, x5-x9
2	QRS	-0.02053	0.06834	0.02264	x2-x3, x5-x9
3	QRL	-0.08019	0.02962	-0.02636	x3, x5-x9
4	SWCF	-32.466	29.457	1.997	x3, x5-x9
5	LWCF	-8.920	-3.038	0.9329	x2-x9
6	CLDH	-7.9094	-0.4732	-3.6714	x2-x9
7	CLDM	-9.8212	0.6915	-3.5459	x2-x3, x5-x9
8	CLDL	-7.378	16.453	1.182	x5-x9
9	FSNT	-32.878	30.886	2.518	x2, x5-x9
10	FLNT	-4.1383	9.1242	1.6874	x2-x3, x5-x9
11	SHFLX	-3.70067	4.06073	-0.07291	x2-x3, x5-x9
12	LHFLX	-5.2260	6.2894	-0.3162	x2-x3, x5-x9
13	PRECT	-0.25613	0.28468	-0.03245	x2-x3, x5-x9
14	AEROD_V	-0.001432	0.048329	0.020163	x2-x3, x5, x7-x9

Table 7: Variable Ranges (VAR)

#	Name	Min	Max	Median	Sig.
1	FNET	0.5128	1.4132	0.9741	x1, x5-x9
2	QRS	0.8362	1.2606	0.9988	x2-x9
3	QRL	0.7451	1.2889	0.9657	x4-x9
4	SWCF	0.3965	2.1765	1.0101	x5-x9
5	LWCF	0.3694	1.2423	0.8247	x3-x9
6	CLDH	0.7503	1.2014	1.0220	x2-x5, x7-x9
7	CLDM	0.5481	1.0462	0.8482	x3-x9
8	CLDL	0.850	1.554	1.147	x4-x9
9	FSNT	0.6552	1.3207	0.9998	x1, x5-x9
10	FLNT	0.8394	1.1601	0.9804	x1, x4-x9
11	SHFLX	0.8357	1.2874	1.0353	x1-x9
12	LHFLX	0.8227	1.3049	1.0370	x1,x3, x5-x9
13	PRECT	0.7823	1.5535	1.2540	x3, x5-x9
14	AEROD_V	0.9528	1.5126	1.1114	x2-x3, x5, x7-x9

Table 8: Variable Ranges (COR)

#	Name	Min	Max	Median	Sig.
1	FNET	0.9214	1.0008	0.9877	x2, x5-x9
2	QRS	0.9124	0.9872	0.9559	x2-x3-x5-x9
3	QRL	0.8698	0.9663	0.9190	x3, x5-x9
4	SWCF	0.7919	0.9704	0.9125	x3, x5-x9
5	LWCF	0.8562	0.9800	0.9329	x2-x9
6	CLDH	0.9117	0.9788	0.9460	x2-x9
7	CLDM	0.8840	0.9721	0.9300	x2-x3, x5-x9
8	CLDL	0.8157	0.9916	0.9513	x5-x9
9	FSNT	0.9674	0.9979	0.9911	x2,x5-x9
10	FLNT	0.9582	0.9916	0.9780	x2, x3, x5-x9
11	SHFLX	0.9257	0.9821	0.9609	x2-x3, x5-x9
12	LHFLX	0.9575	0.9960	0.9807	x2-x3, x5-x9
13	PRECT	0.7934	0.9986	0.8914	x2-x3, x5-x9
14	AEROD_V	0.9269	1.0066	0.9834	x2-x3, x5,x8-x9

Table 9: Top 20 Parameter Combinations Ranked by Skill Score (SS)

SS	x1	x2	x3	x4	x5	x6	x7	x8	x9
0.928185	0.07755	0.029574	0.165367	19.42736	0.888602	13.76669	6517.692	0.652453	5416.862
0.926481	0.889292	0.240847	1.54808	27.27764	0.881276	12.77101	5099.567	0.658623	4197.509
0.923623	0.616049	0.284719	2.536392	26.42562	0.888721	12.41507	5155.649	0.651135	6926.616
0.923513	0.727741	0.287045	2.28714	31.84684	0.890337	13.34337	5196.461	0.656445	7822.533
0.922438	0.042838	0.232568	1.043129	36.79269	0.90677	11.53768	6071.329	0.659189	5719.621
0.921143	0.176015	0.226825	3.782921	21.63099	0.896409	13.14958	6165.648	0.651998	6509.865
0.921011	0.526978	0.493789	1.167412	5.570775	0.875204	13.5256	5387.662	0.655942	3825.458
0.919839	0.050089	0.492586	7.7285	32.618	0.889944	12.63873	4955.783	0.653771	6287.932
0.919759	0.463301	0.26689	2.440262	35.35885	0.894771	13.14318	6818.738	0.66226	5994.082
0.919643	0.131081	0.693092	2.379456	4.423622	0.883988	11.75072	3368.319	0.662584	4980.748
0.918503	0.918325	0.204733	4.412613	2.454088	0.877931	10.49314	2485.559	0.652545	2962.474
0.918347	0.591505	0.176344	0.915604	17.75482	0.865912	14.97419	5350.004	0.656126	5958.651
0.918068	0.306308	0.235387	5.233345	9.441184	0.894897	12.47323	5519.421	0.650821	5862.066
0.918059	0.629153	0.499761	4.043159	13.54851	0.879734	12.51527	5373.165	0.661594	4617.04
0.917815	0.773938	0.213389	1.181908	30.5547	0.902319	12.38677	4450.854	0.659608	8124.649
0.917603	0.683937	0.15464	0.450745	8.103676	0.857756	13.83842	3716.036	0.656798	4199.62
0.917556	0.47887	0.609035	2.088165	26.88733	0.888986	12.09	6032.383	0.66115	4336.297
0.916847	0.894547	0.175766	6.87439	14.0186	0.874181	12.45115	4123.595	0.668338	3945.218
0.916092	0.20578	0.323456	1.617432	30.98886	0.896316	11.62485	3811.514	0.669574	5861.371
0.915761	0.161561	0.326003	0.486259	36.76078	0.883519	15.20108	5686.765	0.663425	7956.867

6 References

[1] Bond T.C. et al., 2013. Bounding the role of black carbon in the climate system: A scientific assessment. *Journal of Geophysical Research: Atmospheres*. 10.1002/jgrd.50171.
[2] Boser B.E., et al. 1992. A Training Algorithm for Optimal Margin Classifiers. Proceedings of the Fifth Annual Workshop on Computational Learning Theory 5 144-152, Pittsburgh, PA.
[3] Brieman L. et al., 2015. randomForest Package for R. Repository: CRAN.
[4] Covey C., et al., 2013. Efficient screening of climate model sensitivity to a large number of perturbed input parameters. *Journal of Advances in Modelling Earth Systems*. Vol. 5 598610.
[5] Dalal I.L. et al., 2008. Low Discrepancy Sequences for Monte Carlo Simulations on Reconfigurable Platforms. Report of the Cooper Union for the Advancement of Science and Art, New York, NY.
[6] Diaz-Uriate R. et al., 2006. Gene selection and classification of microarray data using random forest. BMC Bioinformatics 10.1186/1471-2105-7-3
[7] Glasserman, P., 2010. Monte Carlo Methods in Financial Engineering. Springer.
[8] Hastie, T. et al., 2008. The Elements of Statistical Learning (2nd ed.). Springer.
[9] Homma T. et al., 1996. Importance measures in global sensitivity analysis of nonlinear models. Reliability Engineering and System Safety Vol. 52 1-17
[10] Hsu, C. et al., 2016. A Practical Guide to Support Vector Classification. National Taiwan University.
[11]Flato, G., J. Marotzke, B. Abiodun, P. Braconnot, S.C. Chou, W. Collins, P. Cox, F. Driouech, S. Emori, V. Eyring, C. Forest, P. Gleckler, E. Guilyardi, C. Jakob, V. Kattsov, C. Reason and M. Rummukainen, 2013: Evaluation of Climate Models. In: Climate Change 2013: The Physical Science Basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change [Stocker, T.F., D. Qin, G.-K. Plattner, M. Tignor, S.K. Allen, J. Boschung, A. Nauels, Y. Xia, V. Bex and P.M. Midgley (eds.)]. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA.
[12]Boucher, O., D. Randall, P. Artaxo, C. Bretherton, G. Feingold, P. Forster, V.-M. Kerminen, Y. Kondo, H. Liao, U. Lohmann, P. Rasch, S.K. Satheesh, S. Sherwood, B. Stevens and X.Y. Zhang, 2013: Clouds and Aerosols. In: Climate Change 2013: The Physical Science Basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change [Stocker, T.F., D. Qin, G.-K. Plattner, M. Tignor, S.K. Allen, J. Boschung, A. Nauels, Y. Xia, V. Bex and P.M. Midgley (eds.)]. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA.
[13] Kalisch M., 2012. Random Forest, Applied Multivariate Statistics Spring 2012 Slides. Swiss Federal Institute Of Technology Zurich.
[14] Kucherenko1 S. et al., 2015. Exploring multi-dimensional spaces: a Comparison of LHS and QMC Sampling Techniques. CPSE, Imperial College London.
[15] Meyer D. et. al., 2017 e1071 Package for R. Repository: CRAN.
[16]Minokhin I. 2015. Forecasting northern polar stratospheric variability using a hierarchy of statistical models. University of Waterloo MSc Geography Thesis
[17] Qian Y. el al., 2015. Parametric sensitivity analysis of precipitation at global and local

scales in the Community Atmosphere Model CAM5. *Journal of Advances in Modeling Earth Systems*. DOI: 10.1002/2014MS000354.
[18] Smola A.J., et al. 2003. A tutorial on support vector regression. NeuroCOLT Technical Report NC-TR-98-030, Royal Holloway College, University of London, UK.
[19] Storlie C. et al., 2009. Implementation and evaluation of nonparametric regression procedures for sensitivity analysis of computationally demanding models. Reliability Engineering and System Safety, Vol. 94 Is. 11 1735
[20] Strobl C. et al., 2008. Conditional variable importance for random forests. BMC Bioinformatics, 10.1186/1471-2105-9-307
[21] Vapnik V., 1995. The Nature of Statistical Learning Theory. Springer.
[22] Zhao C. et al., 2013. A sensitivity study of radiative fluxes at the top of atmosphere to cloud-microphysics and aerosol parameters in the community atmosphere model CAM5. Atmos. Chem. Phys., 13, 10969-10987