A machine learning method for parameter estimation and sensitivity analysis

Marcella Torres[1][0000−0002−7974−1631]

University of Richmond, Richmond VA 23173, USA
mtorres@richmond.edu

Abstract. We discuss the application of a supervised machine learning method, random forest algorithm (RF), to perform parameter space exploration and sensitivity analysis on ordinary differential equation models. Decision trees can provide complex decision boundaries and can help visualize decision rules in an easily digested format that can aid in understanding the predictive structure of a dynamic model and the relationship between input parameters and model output. We study a simplified process for model parameter tuning and sensitivity analysis that can be used in the early stages of model development.

Keywords: Parameter estimation · Machine learning · Sensitivity analysis · Ordinary differential equations · Random forest.

1 Introduction

Machine learning, a sub-field of artificial intelligence, is most commonly used to produce an accurate classifier given data. For example, given a data set containing various predictors such as blood pressure and age, a machine learning model could be used to predict whether a given patient has diabetes. This type of application requires adequate experimental data to produce a good classifier, and then the function of such a machine learning model is to accurately classify a new case given predictors. Dynamical systems models, in contrast, offer greater flexibility and more information about the time course of a system rather than just an outcome. Yet these models contain parameters that can be hard to estimate given limited data and model complexity, and the effectiveness of a model depends on the quality of the connection between model parameters and output. In deterministic models such as the ordinary differential equations model presented here, the only model inputs are the model parameters and initial conditions and so these are the sources of any uncertainty in model prediction. Before such a model can be deemed useful for prediction, this uncertainty must be quantified by performing sensitivity analysis. This is a critical step in the model building process, since knowledge of influential parameters will guide experimental design, data assimilation, parameter estimation, and model refinement in the form of complexity reduction.

Machine learning techniques have previously been applied to the problem of parameter estimation in ordinary differential equations, and to identifying

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_26

multicollinearity among parameters. For example, support vector clustering (a combination of support vector machine and clustering) has been used for model parameter estimation [24], and clustering (an unsupervised learning method) has also been used to reduce the number of parameters that need to be estimated by identifying collinearities in pairwise groupings of parameters [6, 23]. Neural networks have also been combined with differential equations to produce models with large data sets [18, 19], and in the case of limited data, sparse identification of nonlinear dynamical systems has been used to discover model structure [5] . Since machine learning uses pattern recognition methods to construct classifiers for data sets, it can also be used to uncover the predictive structure of a model, i.e. the strength of the connection between inputs and outputs. If the goal of developing a good dynamical model is that the model can produce an expected range of biologically reasonable outcomes (epidemic or endemic in SIR models, for example, or survival versus extinction in a predator prey model), then we already have some idea of how output could be classified. When we know the class of each observation in the data set, we can use a supervised learning method to classify future observations. Decision tree algorithms, also called classification and regression tree algorithms (CART), developed by Breiman [3], are one example of many available supervised learning methods, and this is the method we apply here. Decision trees are ideal for parameter space exploration and global sensitivity analysis because the complex relationships between parameters can be easily visualized and decision rules can guide parameter subset selection.

By combining uncertainty analysis (UA) using Latin Hypercube Sampling, as developed by Marino et al. [12], and sensitivity analysis using decision trees, intuition can be gained about how partitions in parameter space produce different model behaviors. In this way we can gain early insight into what behaviors the model is capable of producing and which parameters are driving outcomes. This preliminary exploration can also serve several practical purposes:

- **Visual communication of the significance of model parameters to non-mathematicians.** The tree format allows interactions between multiple parameters and the associated output to be represented simultaneously. In this way it can be made clear that the same behavior can result from different combinations of parameters. Most other learning-based methods of parameter exploration are not easily visualized or are restricted to pairwise parameter plots.
- **Sensitivity analysis.** The first step in estimating model parameters is to identify sensitive parameters - the parameters that impact model output [23]. This is easily obtained from decision tree algorithms as a feature importance measure. In general, this a very quick way of getting a first look at the key parameters that drive model behavior. We can change class definitions dependant on the behavior for which we want to identify these important parameters; for example, instead of endemic versus epidemic we can consider stable versus unstable.
- **Decision rules can be used to find representative sets for simulation.** Collections of parameters that serve as input to a model can represent

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_26

different hypothetical individuals and sub-populations in a biological model. To model these individuals or groups specifically, we can choose parameter sets from value ranges restricted to those suggested by a decision tree.
- **Decision rules can be used to set bounds on parameter space for fitting.** Given experimental data, we know the ultimate class of the observation. Setting bounds on the parameter space and choosing initial parameter values for data fitting is nontrivial and can be time consuming. Generation of a decision tree from simulated data is relatively fast and straightforward. We can then refer to the decision rules on parameters that result in the same class the observation is in to initiate parameter estimation.
- **Decision rules can be used to restrict parameters to those that produce biologically reasonable behavior**. When representative subsets of parameters found using decision rules are investigated and found to produce non-physiological behavior, these parameter sets can be excluded or constraints can be identified. This could also be done by labeling output as either biologically viable or nonviable in the data set as the class label.

To perform sensitivity analysis using decision tree algorithms, we first perform uncertainty analysis by sampling model output over a range of parameter values. Next, a data set is created by appending associated model output to the matrix containing all sampled parameter sets. Each parameter set, together with its output, is then given a binary class label that characterizes the behavior of interest which will serve as the class in the decision tree classifier. Since model output is continuously defined, this is done by defining a criteria on output values which transforms each output value to one of the two class labels. The decision tree is then trained, pruned, and tested on the data. Given acceptable classification accuracy, the tree, or an ensemble of trees, can then be used to analyze parameter importance and identify subsets of parameters that produce particular model outcomes.

2 Methods

First, we perform uncertainty analysis using Latin Hypercube Sampling (LHS), the most efficient of the Monte Carlo methods. LHS is a stratified sampling without replacement method developed by McKay et al. [14, 12, 8, 10] in which each parameter is independently sampled from a statistical distribution in order to efficiently create an unbiased collection of parameter sets that can each be used to generate model output, thus simulating a variety of responses. The choice of statistical distribution from which to sample will be determined by knowledge of the modeled phenomena or an examination of available data. In the absence of information about the underlying distribution, the usual choice is to sample from a uniform distribution, with maximum and minimum values for each parameter determined by physiological constraints or through experimentation.

Next, we combine the matrix of parameter vectors with a vector (or vectors) of model output sampled at a particular time or times where we expect behavior to differ for different outcomes. Each row is labeled with a binary classifier

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_26

according to a criterion on model output values which transforms each output value to a class label. A simplified table showing the representation of data is given in Figure 1, where the classes are labeled "0" or "1".

In statistical learning, measurements are first made on an observation. The goal is to predict which class the observation is in based on the measurements. In this case, the measurements are the parameters $\vec{\theta}$ sampled using LHS which serve as input to the model for simulation and generation of model output. Each observation is a vector of m measurements $\vec{\theta}^{(i)} = \left[\theta_1^{(i)}, \theta_2^{(i)}, \cdots, \theta_m^{(i)}\right]$ which are the parameter sets that are used in combination as model input, and Ω is the parameter space of all possible parameter vectors which is defined by the LHS sample space.

Next, we employ a feature importance calculation with random forest algorithm (RF) to determine the sensitive parameters. A RF is composed of a large number of individual decision trees which sample from the data set with replacement [9, 4, 1]. Finally, we examine an individual decision tree to obtain the rules on parameters which lead to each of the states of interest. Using these rules, we obtain subsets of parameters that may exhibit different behavior.

3 Simple Benchmark model example

Here we illustrate the process of using decision trees to identify parameter subsets and significant parameters, decision tree sensitivity analysis). Ultimately, we will apply this method to more complex, large-scale systems which can benefit from its simplicity, but here we aim to compare the method to more traditional approaches, including solution of the sensitivity equations, so a simple model was chosen. We test the method by comparing results to global sensitivity analysis performed using PRCC as well as the magnitude of computed local sensitivities integrated over time for a well studied model of HIV infection.

3.1 Decision tree sensitivity analysis example: an HIV infection model

Perelson et al. developed a simplified model of HIV interaction with T cells consisting of four differential equations with eight parameters that model concentration of uninfected (T), latently infected (T^*), and actively infected (T^{**}) $CD4^+$ cells and free infectious virus particles [17]:

$$\frac{dT}{dt} = s - \mu_T T + rT\left(1 - \frac{T + T^* + T^{**}}{T_{\max}}\right) - k_1 VT \tag{1}$$

$$\frac{dT^*}{dt} = k_1 VT - \mu_T T^* - k_2 T^* \tag{2}$$

$$\frac{dT^{**}}{dt} = k_2 T^* - \mu_b T^{**} \tag{3}$$

$$\frac{dV}{dt} = N\mu_b T^{**} - k_1 VT - \mu_V V. \tag{4}$$

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_26

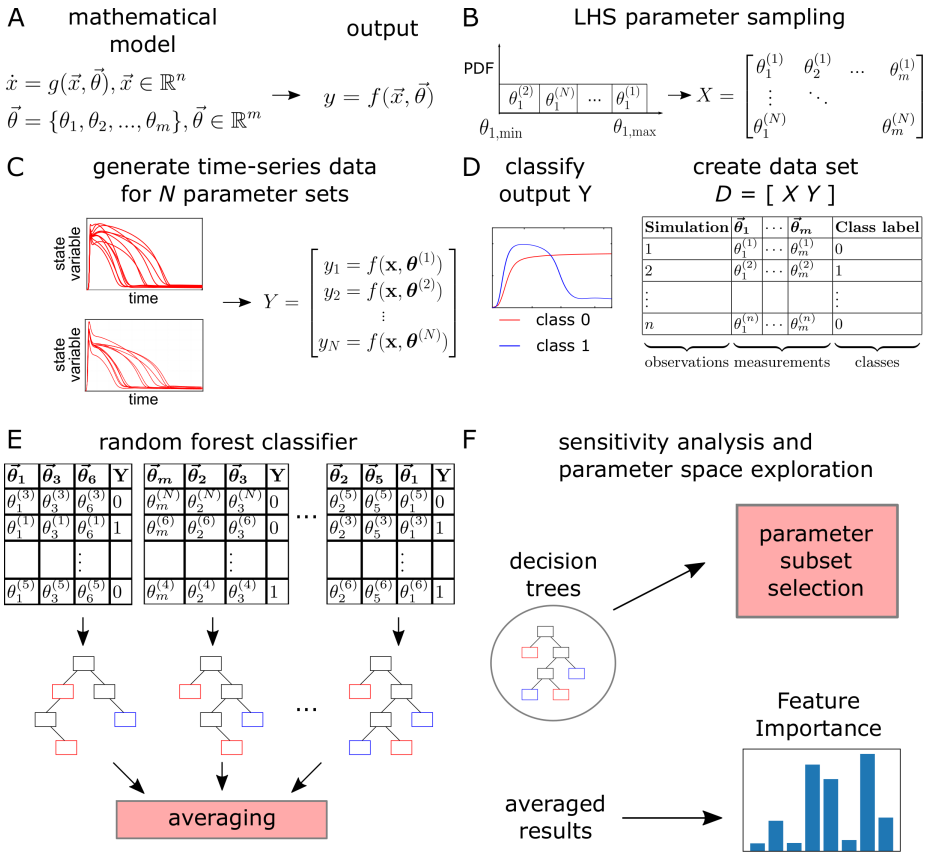

Fig. 1: **Process overview.** (A) An example mathematical model, an n-dimensional system of ordinary differential equations, where $\vec{x}$ represents a vector of state variables and $\vec{\theta}$ is the parameter vector with m model parameters. Model output y depends on state variables and parameters. (B) LHS sampling is performed over all non-fixed parameters, by random sampling of each parameter without replacement from a specified probability density function (a uniform distribution is pictured) over N equally-size bins. This creates N parameter sets, each containing all m model parameters, to form LHS matrix X. (C) N sets of time-series data is produced using all parameter sets. A vector Y of model output sampled at a specified time is selected as the response variable. (D) Assign class label to each output y and create data set D, the set of parameter sets and their associated labeled output. (E) Train random forest classifier with bootstrapping. (F) Obtain feature or permutation importance measure on averaged results as a parameter sensitivity measure. Examine decision trees individually to obtain parameter subsets for parameter space exploration.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_26

Parameters and constants that Perelson et al. defined are in Table 1. Initial conditions used in the model are $T(0) = 1000$ mm^{-3}, $T^*(0) = T^{**}(0) = 0$, and $V(0) = 10^{-3}$ mm^{-3}. Marino et al. performed uncertainty analysis using LHS and sensitivity analysis using PRCCs, and the range used for sampling is also given in Table 1. We will use the same ranges on parameters for LHS sampling to create a data set for constructing decision trees, for comparison to PRCC results.

Two steady states are possible in this model: an uninfected state E_B with no virus, and an endemically infected state E_P with a constant level of the virus [12, 17].

Table 1: Parameters and constants from Perelson et al. [17] and parameter ranges used for LHS sampling by Marino et al. [12].

Parameter	Description [17]	Default [17]	LHS range [12]
s	Rate of supply of CD4$^+$ T cells from precursors	10 day^{-1} mm^{-3}	$[10^{-2}, 50]$
r	Rate of growth of CD4$^+$	0.03 day^{-1}	$[10^{-4}, 0.2]$
μ_T	Death rate of infected and latently infected CD4$^+$ cells	0.02 day^{-1}	$[10^{-2}, 50]$
$T_{\max}$	Maximum CD4$^+$ population level	1500 mm^{-3}	1500
k_1	Rate at which CD4$^+$ cells become infected	2.4×10^{-5} mm^3 day^{-1}	$[10^{-7}, 10^{-3}]$
k_2	Rate at which latently infected CD4$^+$ cells become actively infected	3×10^{-3} day^{-1}	$[10^{-5}, 10^{-2}]$
μ_b	Death rate of actively infected CD4$^+$ cells	0.24 day^{-1}	$[10^{-1}, 0.4]$
N	Number of free virus produced by lysing a CD4$^+$ cell	Not fixed	$[1, 2^3]$
μ_V	Death rate of free virus	2.4 day^{-1}	$[10^{-1}, 10]$

Perelson et al. discovered that a criterion for achieving the uninfected steady state E_B is

$$N < \frac{(k_2 + \mu_T)\,\mu_V + k_1 T_0}{k_1 k_2 T_0}, \tag{5}$$

and showed further showed analytically that parameters contained in this critical value for N, defined as N_{crit}, are bifurcation parameters.

Generating the data set LHS was performed in MATLAB with sample size $N = 1000$, using code provided by Marino et al. [13]. A data matrix was created such that each row of the matrix contained a parameter set consisting of parameters s, r, μ_T, k_1, k_2, μ_b, μ_V, and N was sampled from a uniform distribution. The model given in Equations (1)-(4) tracks four variables that are possible model outputs to define as a classifier in decision tree sensitivity analysis. We

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_26

chose concentration of free virus particles, V, as a binary classifier, sampled at time $t = 4000$ days, since this output should characterize the two steady states of biological interest: uninfected (E_B) or endemically infected (E_P). We labeled model output as uninfected if its value was near zero ($V < 1 \times 10^{-6}$), and endemically infected otherwise. Finally, the data set was balanced such that each class was equally represented, resulting in a data set of size $N = 434$ in the form given in Figure 1.

Random forest for the model of HIV infection In the RF algorithm, individual decision trees in the forest are built using bootstrap samples selected with replacement from the total data set, and the predictions of all trees are averaged in the final step. In this way, variance is reduced. For each tree, observations not used for fitting (out-of-bag or OOB observations) can then be used for validation using each of the individual decision tree models for which it was not contained in the bootstrap sample. The out-of-bag score (OOB score) is then the number of correctly predicted observations from the out of bag sample.

There are many RF implementations available. In our example, we use RandomForestClassifier in the Python *scikit-learn* machine learning package [16]. A brief overview of the general RF algorithm, with reference to important arguments that can be used to tune the model, is:

1. Draw a same-size bootstrap sample from the original data set (argument: number of samples to draw, `max_sample`). Observations left out of the sample (OOB data) is used as an unbiased accuracy measure (out-of-bag score) and in computing feature importance.
2. Train a tree on the bootstrapped sample by randomly selecting features without replacement at each node (argument: number of features to consider at each split, `max_features`), then splitting according to the feature choice that optimizes the objective function (argument: splitting criterion, `criterion`).
3. Repeat (argument: number of trees in forest, `n_estimators`).
4. Aggregate the predictions of all trees by majority voting. The OOB score of the RF is then computed as the averaged correct predictions of OOB observations using trees for which they are OOB.
5. The feature importance, used here as a measure of parameter sensitivity, is averaged over all trees.

In Figure 2, we show the impact of tuning several of the RF algorithm parameters on prediction accuracy for a RF trained on the LHS-generated data for the HIV dynamical system, with the eight model input parameters defined in Table 1 as features and model output variable free virus concentration after 4000 days defined as class. In this example, limiting the number of model input parameters to consider as candidate predictors at each split does not impact model performance. A minimum number of trees in the forest to achieve a high accuracy as measured by OOB score could be as low as 50 trees, however for a data set of this size a larger forest is also a reasonable choice given the negligible difference in computation time. Large decision trees can suffer from overfitting,

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_26

so it is best to choose the smallest tree size with sufficient classification accuracy. Here it appears that a maximum tree depth of four, or even three, is sufficient. Boulesteix et al. [2] contains an excellent practical guide to RF tuning parameters important in computational biology applications.

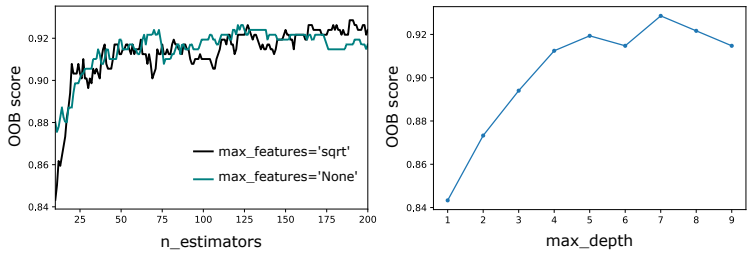

Fig. 2: **Dependence of OOB score on RF tuning parameters.** On the left, out-of-bag accuracy score is computed for varying number of trees in the forest (`n_estimators`), for two different values of number of features (model parameters) to consider at each split (`max_features`). On the right, out-of-bag accuracy score is computed for varying tree size ((`max_depth`).

Feature importance: comparison to PRCC and local sensitivity analysis. A comparison between RF feature importance, PRCCs, and sensitivities as measures of parameter importance is shown in in Figure 3, with sensitive parameters as determined by each method appearing with (*). PRCCs (Partial Rank Correlation Coefficients) uses partial rank correlation to first rank transform the vectors containing sample parameters and associated outputs, and then calculates the correlation between each parameter and output after discounting the effects of the remaining parameters [12, 7]. Thus this global method apportions variability in model output to variability in parameters, allowing us to determine how each parameter effects model output (sensitivity analysis). Traditional, local sensitivity analysis methods, unlike RF feature importance or PRCC methods, neglect relationships between parameters and consider only the impact of individual parameters on model output by holding all others fixed. These values are obtained by integrating the partial differential equations for each parameter with respect to free virus V over the whole time course, something that is computationally intensive or impossible for a larger model.

In comparison, feature importance is calculated as part of the RF classification algorithm in *scikit-learn* [16]; this is the total amount that the selection criterion decreases with each split on the given feature (the selection criterion used here is Gini impurity). Since we are splitting on parameters with model output defined as class, this is a parameter importance measure that indicates which parameters contribute most to determining model outcomes which implicitly takes into account interactions between model parameters (unlike tradi-

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_26

tional, local methods) [2]. Unlike PRCCs, these are values that range from 0 to 1 and are normalized to sum to 1. Averaged results for a random forest of 100 decision trees are shown in Figure 3, and we expect these results to be unbiased since parameters are continuous variables [21]. With a feature importance cutoff of 5%, we identify the same important parameters identified analytically by Perelson et al. as bifurcation parameters, which in turn are the same parameters identified by PRCC and traditional methods.

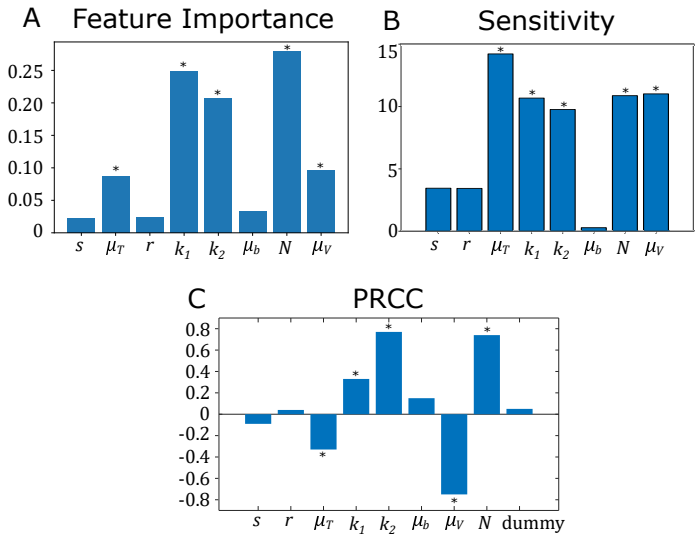

Fig. 3: Comparison of parameter importance measures. Parameters denoted sensitive by each measure are denoted with (*). (A) Averaged feature importance estimates for a random forest of 100 decision trees. The parameter importance cutoff was set to 5%, with parameters contributing more than 5% to reduction in the selection criterion considered sensitive. (B) Magnitude of the computed sensitivities of free virus V to parameters integrated over time, obtained by solving the partial differential sensitivity equations. (C) PRCC results from Marino et al. [12] computed by sampling HIV model output at time $t = 4000$ days with sample size of $N = 200$. Additional PRCC values, including time $t = 2000$ days and varying sample sizes, are given in Supplement D, Table D.1 in [12].

Parameter subset selection A unique benefit of decision tree classification algorithms is visualization of complex relationships among features. While we used a random forest of trees to check accuracy and determine parameter importance, for visualization and interpretation of decision rules individual trees are examined.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_26

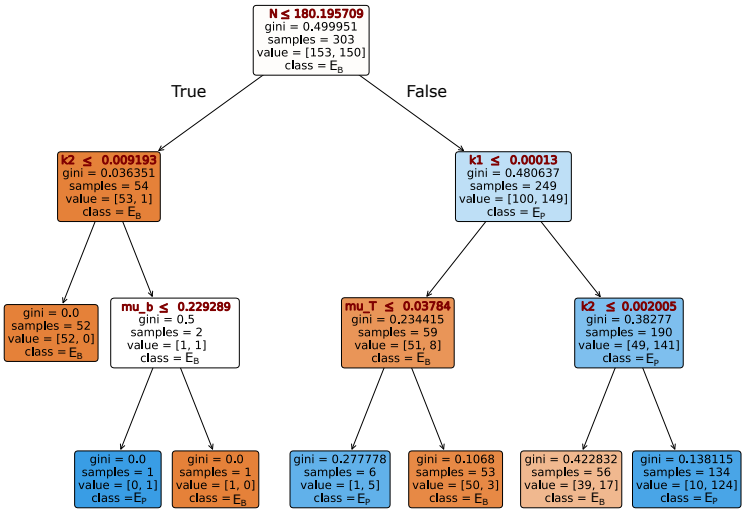

Fig. 4: **Decision tree for the HIV model.** Classes are labeled as uninfected (E_B) or endemically infected (E_P). In the tree, each node has a decision rule based on a parameter that splits the data. If evaluation of the rule has a true result, the data will be sorted to the left in the split. A class is designated at each node depending on which class is most represented (the number of data points in each class is given in "value".) For example, from the leftmost branch of the tree, we see if $117.9 \leq N \leq 180.2$ and $k_2 \leq 0.009$, the predicted model steady state is uninfected.

For example, the decision tree shown in Figure 4 can be interpreted as collections of parameter sets that are input into the model that produce particular steady states. Classes are labeled as uninfected (E_B, no virus remains in the long term) or endemically infected (E_P, a constant level of virus remains in the long term). In the tree, each node has a decision rule based on one of the parameters that splits the data. If evaluation of the rule has a true result, the data will be sorted to the left in the split. A class is designated at each node depending on which class is most represented (the number of data points in each class is given in "value".) From the leftmost branch of the tree, we see if $117.9 \leq N \leq 180.2$ and $k_2 \leq 0.009$, where N is the number of free virus produced by lysing a $CD4^+$ cell and k_2 is the transition rate of $CD4^+$ cells from latently to actively infected, the predicted model steady state is uninfected unless $\mu_b \leq 0.229$.

A decision tree can exported in text format that is easily translated to code for setting parameter ranges in running model simulation. For example, using the `export_text` function in Python *scikit-learn* for the decision tree in Figure 4 results in the output given in Listing 1.1 [16].

Listing 1.1: Text output of decision rules from `export_text` function in Python *sklearn* for the decision tree in Figure 4

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_26

```
|--- N <= 180.195709
|    |--- k2 <= 0.009193
|    |    |--- class: 0
|    |--- k2 >  0.009193
|    |    |--- mu_b <= 0.229289
|    |    |    |--- class: 1
|    |    |--- mu_b >  0.229289
|    |    |    |--- class: 0
|--- N >  180.195709
|    |--- k1 <= 0.000130
|    |    |--- mu_T <= 0.037840
|    |    |    |--- class: 1
|    |    |--- mu_T >  0.037840
|    |    |    |--- class: 0
|    |--- k1 >  0.000130
|    |    |--- k2 <= 0.002005
|    |    |    |--- class: 0
|    |    |--- k2 >  0.002005
|    |    |    |--- class: 1
```

These rules on parameters can be used to guide simulation and parameter fitting to get a sense of the different paths to the same outcomes. For example, given the tree in Fig 4 and the decision rules in Listing 1.1, we can choose parameter sets that satisfy decision rules

1. $N > 180.19571, k_1 \leq 0.00013, \mu_T \leq 0.03784$ (parameter inputs sampled from these ranges pictured in Figure 5A)
2. $N > 180.19571, k_1 > 0.00013, k_2 > 0.00201$ (parameter inputs sampled from these ranges pictured in Figure 5B)
3. $N \leq 180.19571, k_2 > 0.00919, \mu_b \leq 0.22929$ (parameter inputs sampled from these ranges pictured in Figure 5C).

The rules allow us to select collections of parameters that, when input in the model, ultimately lead to the endemically infected steady state yet may exhibit different behavior. Figure 5 depicts free virus versus time, with parameters sampled from three regions of parameter space determined by each of the three decision rules discovered with decision tree sensitivity analysis.

Parameter sets sampled from ranges set by decision rules produce different behavior across groups even though most of trajectories end up in the endemically infected state as predicted. As compared to Figure 5A, the twenty simulations run using the Rule 2 inputs shown in Figure 5B have an earlier peak, and some exhibit an early spike in free virus concentration. In contrast to simulations run using Rule 1 and Rule 2, simulations run using Rule 3 shown in Figure 5C all have much lower magnitude of free virus concentration over the entire time course of infection. This is not meant to be an exhaustive demonstration of how decision rules could be used and is intended to show that we gain insight from this simple approach if what are interested in is primarily an exploration of parameter space. We could also consider fixing all parameters that were not found

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_26

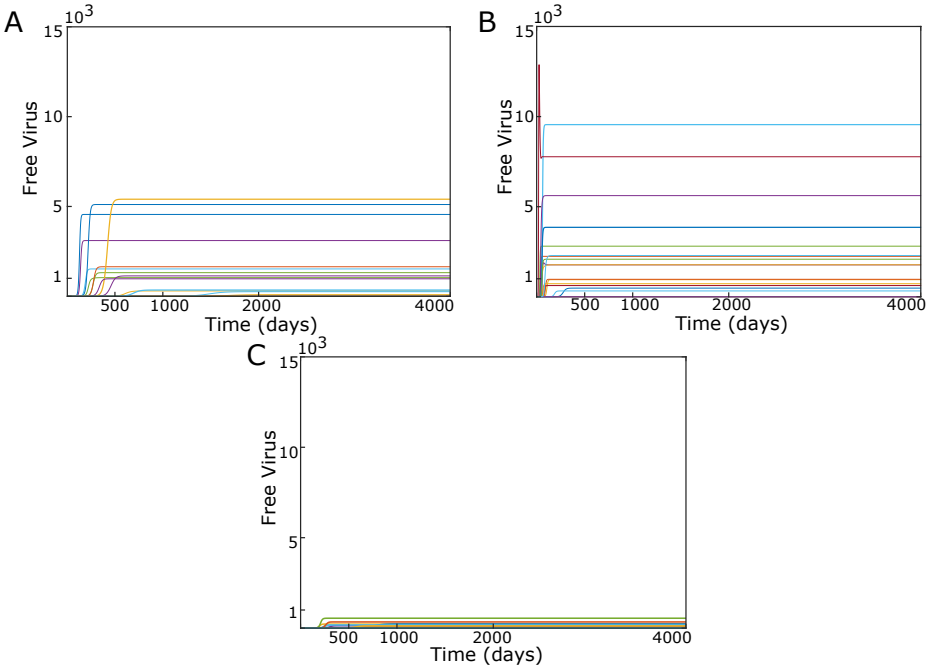

Fig. 5: **Free virus (model output) versus time in days.** The three decision tree paths that lead to the endemically infected steady state are used to set parameter ranges for sampling. In (A), parameter inputs to the model are sampled from ranges set by Rule 1 $=N > 180.19571, k_1 \leq 0.00013, \mu_T \leq 0.03784$. In (B), parameter inputs to the model are sampled from ranges set by Rule 2$=N > 180.19571, k_1 > 0.00013, k_2 > 0.00201$. In (C), parameter inputs to the model are sampled from ranges set by Rule 3$=N \leq 180.19571, k_2 > 0.00919, \mu_b \leq 0.22929$.

sensitive by the feature importance metric before allowing only the sensitive parameters to vary within boundaries set by decision rules, for example.

4 Discussion and conclusion

We have demonstrated a process for qualitative analysis of dynamical model behavior using decision trees. The main advantages of this method are the clarity in the connection between the model outcomes of interest and the parameters, its ease of implementation, the simultaneous attainment of both important parameters (sensitivity analysis) and subsets of parameters linked to particular outcomes (parameter subset selection), and the visualization of parameter relationships.

This method employs simulated data, which is a limitation that allows early investigation of model behavior that can be expanded into a larger investigation of parameter space as experimental data becomes available. In addition, as with

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_26

most sampling-based global sensitivity methods, experimentation is required to find an adequately sized data set to produce a good classifier. A benefit of this method is that RF is known to implicitly account for interactions between features, an important consideration where features are dynamical systems model parameters [2, 20]; however, no information about the nature of interactions is provided that can be obtained without sorting through tree decision rules. Methods for systematically analyzing RF results to obtain more information about interactions have, however, been proposed in bioinformatics applications [22, 25, 11, 15].

In future work, we propose to further investigating how interactions among parameters could be analyzed by comparing splitting attributes on parameters across decision trees in a random forest, for more complex models with known interactions. We will apply this process to such a larger-scale ordinary differential equations model with many parameters next, as well as to an agent-based model, for which traditional methods of sensitivity analysis are more difficult to conduct and results are hard to interpret.

References

1. Amit, Y., Geman, D.: Shape quantization and recognition with randomized trees. Neural computation **9**(7), 1545–1588 (1997)
2. Boulesteix, A.L., Janitza, S., Kruppa, J., König, I.R.: Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **2**(6), 493–507 (2012)
3. Breiman, L.: Classification and regression trees. Wadsworth statistics/probability series, Wadsworth International Group (1984), https://books.google.com/books?id=uxPvAAAAMAAJ
4. Breiman, L.: Random forests. Machine learning **45**(1), 5–32 (2001)
5. Brunton, S.L., Proctor, J.L., Kutz, J.N.: Discovering governing equations from data by sparse identification of nonlinear dynamical systems. Proceedings of the national academy of sciences **113**(15), 3932–3937 (2016)
6. Chu, Y., Hahn, J.: Parameter set selection via clustering of parameters into pairwise indistinguishable groups of parameters. Industrial & Engineering Chemistry Research **48**(13), 6000–6009 (2009)
7. Conover, W.J., Iman, R.L.: Rank transformations as a bridge between parametric and nonparametric statistics. The American Statistician **35**(3), 124–129 (1981)
8. Helton, J.C., Davis, F.J.: Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems. Reliability Engineering & System Safety **81**(1), 23–69 (2003)
9. Ho, T.K.: The random subspace method for constructing decision forests. IEEE transactions on pattern analysis and machine intelligence **20**(8), 832–844 (1998)
10. Iman, R.L., Conover, W.J.: The use of the rank transform in regression. Technometrics **21**(4), 499–509 (1979)
11. Jiang, R., Tang, W., Wu, X., Fu, W.: A random forest approach to the detection of epistatic interactions in case-control studies. BMC bioinformatics **10**(1), 1–12 (2009)

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_26

12. Marino, S., Hogue, I.B., Ray, C.J., Kirschner, D.E.: A methodology for performing global uncertainty and sensitivity analysis in systems biology. Journal of Theoretical Biology **254**(1), 178–196 (2008). https://doi.org/10.1016/j.jtbi.2008.04.011
13. Marino, Simeone and Hogue, Ian B. and Ray, Christian J. and Kirschner, Denise E.: Uncertainty and sensitivity functions and implementation (matlab functions for prcc and efast), http://malthus.micro.med.umich.edu/lab/usanalysis.html
14. McKay, M.: Latin hypercube sampling as a tool in uncertainity analysis of computer models. Proceedings of the 1992 Winter Simulation Conference (1992). https://doi.org/10.1145/167293.167637
15. Meng, Y., Yang, Q., Cuenco, K.T., Cupples, L.A., DeStefano, A.L., Lunetta, K.L.: Two-stage approach for identifying single-nucleotide polymorphisms associated with rheumatoid arthritis using random forests and bayesian networks. In: BMC proceedings. vol. 1, pp. 1–6. BioMed Central (2007)
16. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
17. Perelson, A.S., Kirschner, D.E., De Boer, R.: Dynamics of hiv infection of cd4+ t cells. Mathematical biosciences **114**(1), 81–125 (1993)
18. Rackauckas, C., Ma, Y., Martensen, J., Warner, C., Zubov, K., Supekar, R., Skinner, D., Ramadhan, A., Edelman, A.: Universal differential equations for scientific machine learning. arXiv preprint arXiv:2001.04385 (2020)
19. Raissi, M., Perdikaris, P., Karniadakis, G.E.: Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational Physics **378**, 686–707 (2019)
20. Rodenburg, W., Heidema, A.G., Boer, J.M., Bovee-Oudenhoven, I.M., Feskens, E.J., Mariman, E.C., Keijer, J.: A framework to identify physiological responses in microarray-based gene expression studies: selection and interpretation of biologically relevant genes. Physiological genomics **33**(1), 78–90 (2008)
21. Strobl, C., Boulesteix, A.L., Zeileis, A., Hothorn, T.: Bias in random forest variable importance measures: Illustrations, sources and a solution. BMC bioinformatics **8**(1), 1–21 (2007)
22. Tang, R., Sinnwell, J.P., Li, J., Rider, D.N., de Andrade, M., Biernacka, J.M.: Identification of genes and haplotypes that predict rheumatoid arthritis using random forests. In: BMC proceedings. vol. 3, pp. 1–5. BioMed Central (2009)
23. Torres, M., Wang, J., Yannie, P.J., Ghosh, S., Segal, R.A., Reynolds, A.M.: Identifying important parameters in the inflammatory process with a mathematical model of immune cell influx and macrophage polarization. PLoS computational biology **15**(7), e1007172 (2019)
24. Yılmaz, Ö., Achenie, L.E., Srivastava, R.: Systematic tuning of parameters in support vector clustering. Mathematical Biosciences **205**(2), 252–270 (2007)
25. Yoshida, M., Koike, A.: Snpinterforest: a new method for detecting epistatic interactions. BMC bioinformatics **12**(1), 1–10 (2011)

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_26