Atmos. Meas. Tech. Discuss.,
doi:10.5194/amt-2019-308-AC1, 2020

Atmospheric Measurement Techniques
Discussions

Interactive comment on "Gradient Boosting Machine Learning to Improve Satellite-Derived Column Water Vapor Measurement Error" *by* Allan C. Just et al.

Allan C. Just et al.

allan.just@mssm.edu

Received and published: 12 March 2020

Response to RC1: We thank the reviewer for their helpful and specific comments. We completely agree that revisions contributing to more accessible methods and discussion sections will increase the impact of our work. We have added plain language descriptions that explain the meaning behind specialized terms that are commonly used in machine learning. More importantly, our more accessible language is to be taken in conjunction with the reproducible data + open source R code we have archived in the Zenodo open-access digital repository ((https://doi.org/10.5281/zenodo.3568449), which will ease the integration of the methods and ideas that we employ into other

atmospheric measurement applications. We have substantively revised the language we use to describe methods within the introduction, methods, and discussion sections. We add additional details, define terms coming from the field of machine learning, and seek to clarify the points of confusion raised by the reviewer. We have also included references to two strong introductory articles to guide readers that are seeking additional information on the inner workings of the machine-learning methods that we describe. We hope that the referee also sees the improved clarity and accessibility of the revised manuscript which we consider substantially improved. We include a point-by-point response to comments below:

1. - L64-65. What do you mean by "weak predictor"? And "binary partitioning" of what?

Response: We have added additional information to clarify this sentence in our introduction which previously read, "Gradient boosting involves fitting a large number of tree-based models where each subsequent tree is made a weak predictor of the error from the previous trees." The revised section now reads, "XGBoost involves fitting a large number of tree-based models. Each subsequent tree is fit to the error from the previous trees and the predictions of all the trees are added together. Each tree's prediction is multiplied by a shrinkage factor (or "learning rate") η, a number between 0 and 1. By adding successive trees, XGBoost descends the gradient of the loss function. The component trees use a recursive binary partitioning of the predictors that accommodates varying types and scales of predictor variables and is robust to outliers (Elith et al., 2008)." To further clarify, both the shrinkage factor η and the use of random dropout (we explain the DART method elsewhere), are used to decrease overfitting that occurs when a model is directly applied to residual error. Our revisions have clarified that the binary partitioning used in regression trees is a splitting of the predictor variables.

2. - L113-122. The problem with this paragraph is similar to the one I highlighted during the access review, namely that it uses many specialized terms without defining them. Therefore, this paragraph is not very informative to a reader who does not have a

background in this type of methods (a condition that is probably not uncommon among the readership of AMT), and probably is also not very informative to a reader who does.

Response: We have added additional detail to explain specialized terms and have added a reference to an accessible introduction to regression trees. Below we include specific examples of how we have clarified points that were not clear to the reviewer.

3. In particular, the following aspects are not clear to me. Let us suppose that we have a number of predictors (e.g. solar zenith angle, viewing zenith angle, AOD, etc.). When you say that the model "specifies a few recursive binary splits of predictors etc.", do you mean that it defines a threshold for each predictor and returns a different output depending on whether the predictor is above or below the threshold? And does the next level of the tree apply similar operations to the result of this first thresholding, and so on? If so, make this point clearer in your discussion. I had to look inside the references to understand this, but such a basic level of detail should be already understandable from your paper, without forcing the reader to peruse the references.

Response: Following the reviewer's suggestion, we have revised this section to explain in greater detail how tree-based regression models operate. While avoiding jargon, we also see our paper as an opportunity to inform a new readership on the terminology of machine learning and how it can be applied in earth sciences. It now reads: "For an introduction to regression trees, see Strobl et al. (2009). A regression tree is a model that specifies recursive binary splits of predictors and assigns a constant value to all cases that end up in the same terminal node (namely, their mean on the dependent variable). The algorithm chooses the splits across all predictors that minimize the variance of the residuals. The maximum number of splits within each tree (also known as the maximum depth) can be set as a hyperparameter. A set of multiple trees can be used for prediction by combining the outputs of the individual trees for each case. Such a set of trees can accommodate complex relationships including non-linearities and interactions while being robust to outliers. Boosting is a method of fitting a series of models iteratively, with each model fit on the residuals of the previous models. While each tree

may individually perform relatively poorly at predicting the outcome (and thus is known as a "weak learner"), the combination of many trees can collectively describe complex relationships and account for the impact of many predictors. Further, because boosting includes sequentially learning by combining many iteratively fit trees that address the error in previous trees, this technique performs well, achieving low testing error. The XGBoost package is a scalable gradient boosting implementation with additional features including penalties to avoid overfitting and optimized computational speed (Chen and Guestrin, 2016)." We believe our revisions achieve a reasonable balance of including sufficient descriptive information and useful references without swamping the reader with excessive detail.

4. Who decides which predictors should be split and whether they should be split independently or according to certain logical combinations (AND, OR, etc.)? Is it the user or is it the training algorithm that makes this decision? In addition, if this is up to the training algorithm, how is the system trained? How is the cost function defined and how are the system parameters adjusted?

Response: As clarified in our revised Statistical Methods section, the regression tree algorithm selects at each step the split across all predictors that minimizes the variance of the residuals. The selection of splits is done recursively and the number of splits (depth of the tree) is a hyperparameter that is tuned by the analyst (we discuss our hyperparameter tuning approach below). While a split upon another split of a tree constructs a logical AND statement, the addition of sufficient splits can approximate an OR statement. The user does not select split points.

5. Again, what do you mean by "weak learner"? How are multiple learners combined? Who decides what weight should be given to each learner, and how?

Response: the "weak learner" term was explained in the revised manuscript. It now reads: "While each tree may individually perform relatively poorly at predicting the outcome (and thus is known as a "weak learner"), the combination of many trees can collectively describe complex relationships and account for the impact of many predictors. “ Instead of fitting a single large decision tree to all the data with many splits, which will perform poorly in prediction (having low bias but very high variance), the boosting approach learns slowly by fitting a smaller decision tree to the residuals of the model and slowly improving the model in areas where it does not perform well. In general, statistical learning approaches that learn slowly tend to perform well by producing both low bias and low variance.

6. What is the role of "gradient" in gradient boosting? Gradient of what with respect to what?

Response: The gradient in question is the residuals (the observed values minus the predicted values), which are the gradient of squared-error loss with respect to the residuals. A gradient-boosting model adds trees in order to minimize this gradient.

7. - L130. Please define the “several hyperparameters related to the desired size and complexity of the model”. Plus, why "hyperparameters" and not simply "parameters"?

Response: In machine learning, a hyperparameter is a configuration set before the learning process begins and that takes values that cannot be directly estimated from the data. Parameters, such as regression coefficients or split points in tree-based models, are estimated from the data. Simple algorithms like linear regression don’t require hyperparameters while more complex algorithms may have several hyperparameters. For these more complex machine-learning algorithms, hyperparameters need to be predefined by researchers within a certain range of values. Often a set of appropriate hyperparameter values that result in improved performance are selected through a cross-validation process. We have added the following sentences to clarify, “XGBoost has several hyperparameters related to the desired size and complexity of the model that need to be set in training for each dataset. We had a priori selected to tune our XGBoost models with DART using six hyperparameters (Supplementary Table S2), while using default values for other potential hyperparameters based on previous modelling

experience.”

8. - L132. What do you mean by “nested comparison”? In particular, in what sense “nested”?

Response: To avoid overfitting, it is important that all learning, including the selection of hyperparameter values, occurs within the training dataset. However, the evaluation of performance should still be done on data that was not used in algorithm training and thus requires cross-validation within the training dataset. This leads to nested cross-validation, where the training data is being further split in half in order to evaluate the performance of different hyperparameter values. Our revised section now reads, “Our tuning and evaluation approach used two-level (nested) cross-validation. Within each training fold for our outer cross-validation, we further randomly split the training data in half and performed a 2-fold cross-validation to compare the performance of XGBoost models using 50 random sets of potential hyperparameters selected with Latin hypercube sampling (Stein, 1987) to be well-spaced across the range of potential hyperparameter values.”

9. - L133. Could you provide a reference for Latin hypercube sampling, and possibly summarize what it essentially does?

Response: We use Latin Hypercube Sampling (LHS) to generate random combinations of parameter values. It is based on the Latin square design, which has a single sample in each row and column. One-dimensional LHS involves dividing your cumulative density function into n equal partitions; and then choosing a random data point in each partition. We are using a multi-dimensional LHS because we have six hyperparameters to tune across simultaneously. LHS helps to ensure that samples are representative of the real variability in the distribution. We have added additional explanation of the purpose of the Latin hypercube sampling and our section now reads, “Within each training fold for our outer cross-validation, we further randomly split the training data in half and performed a 2-fold cross-validation to compare the performance of XGBoost models using 50 random sets of potential hyperparameters selected with Latin hypercube sampling (Stein, 1987) to be well-spaced across the range of potential hyperparameter values. While this is more similar to a random search than a grid search, it is expected to more efficiently find well performing sets of hyperparameters than random search, because it decreases the likelihood of checking combinations that are trivially different or leaving unexplored regions in the six-dimensional space, which has too many combinations to effectively cover with a grid search. We selected the set of hyperparameters that minimized the RMSE within the withheld portion of the training data before refitting with all training data."

10. - In general, the fundamental question I have about Section 3 is: if I want to replicate your study or apply your method to another problem - e.g., by writing my own code - what do I actually need to do? What are the computational steps involved?

Response: We have taken several steps to assist our readers with replication. We have added substantially to the detailed description of our methods, and as we have listed in our manuscript, we have already placed full reproducible R code, including all computational steps, and our datasets in an Open Access Zenodo repository (DOI 10.5281/zenodo.3266058) enabling anyone to rerun our full code and regenerate all of our results in the manuscript. This makes checking the quality and reproducibility of our work fully accessible and will assist readers in replicating this study in new datasets or in applying these methods to other problems in atmospheric measurement science.

11. - L146. I think "Shapely" should actually read "Shapley"

Response: we have corrected this in the revised manuscript.

12. - L442. Some details of the reference appear to be missing.

Response: we have corrected the reference which appeared to be missing journal information.