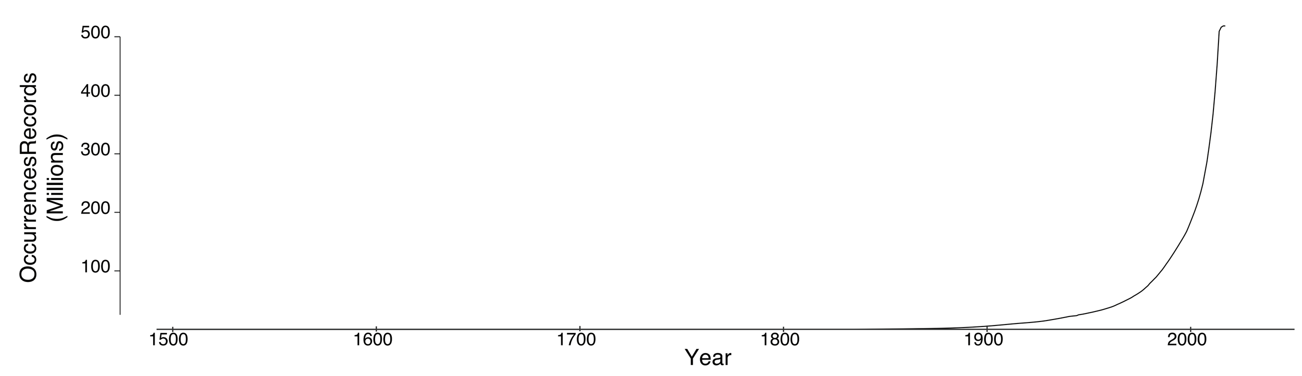
Predicting Execution Time of Climate-Driven Ecological Forecasting Models

Scott Farley[1] and John Williams[1,2]

[1]Department of Geography, University of Wisconsin, Madison, WI
[2]Center for Climatic Research, University of Wisconsin, Madison, WI

Abstract

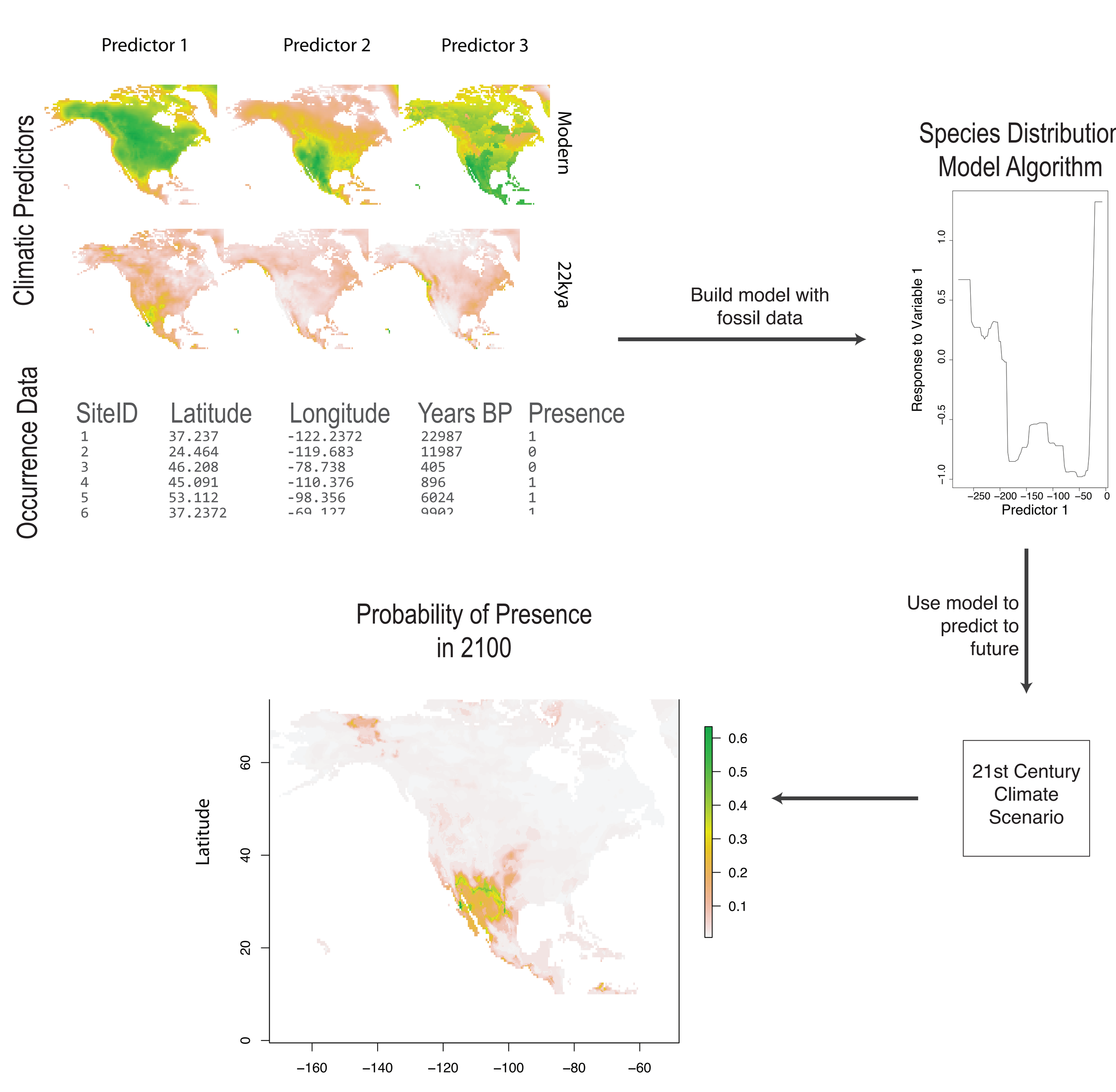
Species distribution models are climate-driven ecological forecasting tools that are widely used to predict species range shifts and ecological responses to 21st century climate change. As modern and fossil biodiversity databases expand rapdily and statistical methods become more computationally intensive, choosing the correct computing configuration on which to run these models becomes more important. We present a predictive model for estimating species distribution model execution time based on algorithm inputs and computing hardware. The model shows considerable predictive skill and can inform future resource provisioning strategies. Our models are extremely sensitive to the amount of data used in the fitting process and only minimally affected by the hardware they were run on. We also demonstrate a technique for predicting model accuracy that suggests that inclusion of training data from the fossil record can enhance the accuracy of distribution models. The ultimate goal of our project is to better understand limits to application of climate-driven forecasting models in order to improve large scale modeling of biodiversity shifts in response to climate change.



Trends in Global Biodiversity Information

Species Distribution Models

Species Distribution Models (SDMs) use statistical learning algorithms to estimate a species' response to climatic gradients. Response surfaces can be used to understand future ecological change. Increasingly, SDMs are using both contemporary and fossil data to model climate drivers of biodiversity across many states of the climate system. We evaluated three SDM algorithms: boosted regression trees (**GBM-BRT**), multivariate adaptive regression spines (**MARS**), and generalized additive models (**GAM**).



Methods

We systematically tested the accuracy and runtime of three popular SDM algorithms on four training set sizes, four spatial resolutions and 44 computing configurations (4xCPU, 11xRAM).

- All experiments were done using popular SDM packages in the R programming language
- **Fossil occurrences** from the spruce (*Picea*) genus over the last 21,000 years were obtained from the Neotoma Paleoecological Database
- 0.5º spatial resolution debiased and downscaled CCSM3 climate model output for North America was used to build a predictor feature vector for each occurrence
- **SDMs were projected onto 21st century** climates using HadCM3 climate model output
- Two models for algorithm runtime were fit to each SDM: a linear multiple regression and a boosted regression tree model
- Estimates of residual sum of squares and mean model error were used to evaluate each runtime model
- Models of SDM accuracy were fit using boosted regression trees using area under the receiver operator curve (AUC) as a classification error metric
- Runtime and accuracy models were tested with an **independent holdout set** of 20% of the total set of examples

Acknowledgments

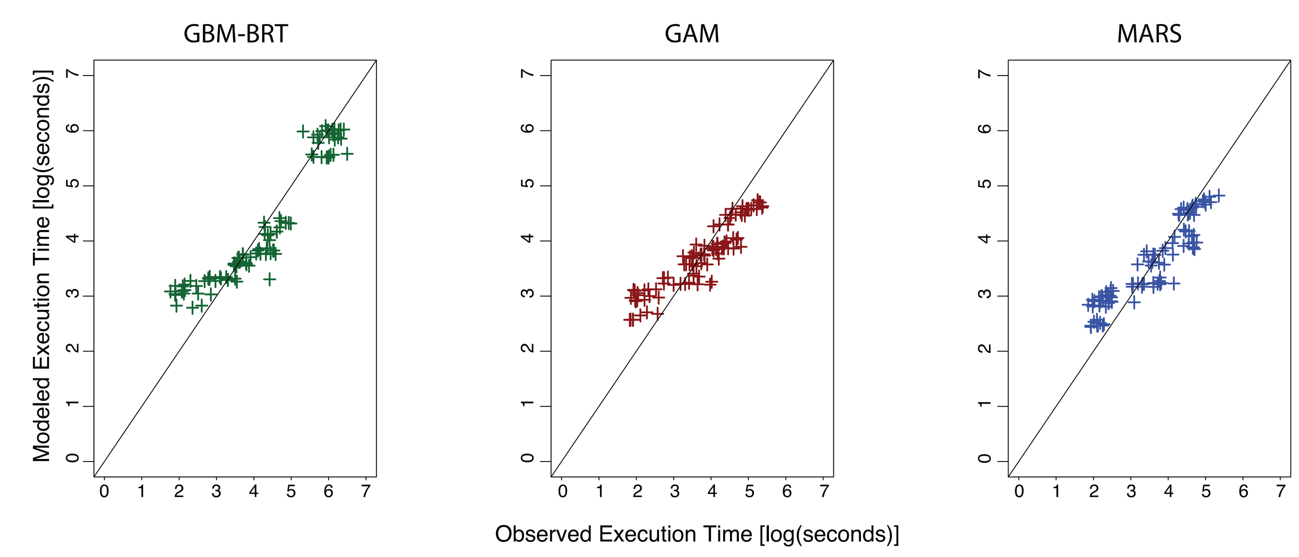
Funding for the authors was provided by University of Wisconsin-Madison Geography Department's Trewartha Research Award, the University of Wisconsin-Madison Vilas Research Trust, and the National Science Foundation (EAR-1550707).

Corresponding Author: Scott Farley • sfarley2@wisc.edu

Results

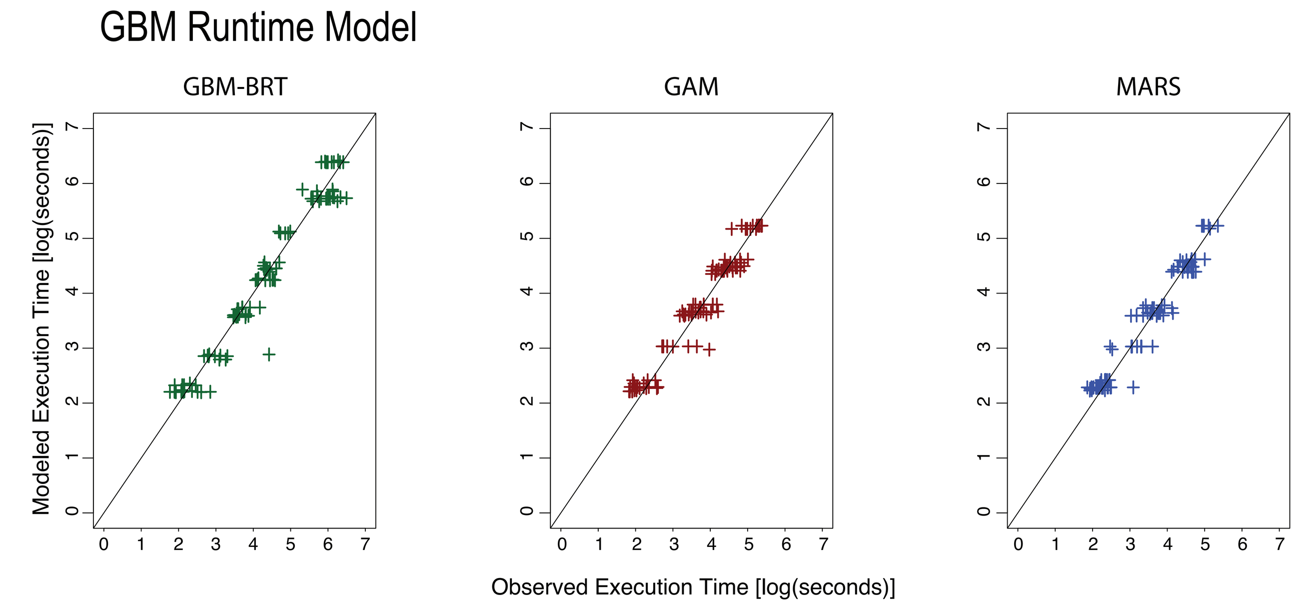
In general, boosted regression tree models outperformed linear models of runtime because they are better able to capture non-linearities in the empirical dataset. The best model was the regression tree model of the MARS algorithm. The mean prediction error across all models was 1.036 ± 1.353 seconds. Correlation between observed and predicted values was >0.8 for all models, though GBM models show lower residual deviance than the corresponding linear models.

Linear Runtime Model



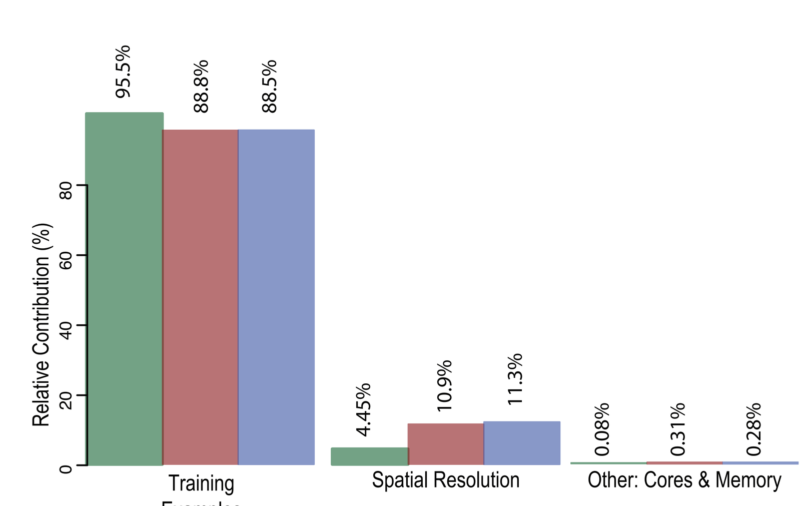
	Residual Sum of Squares	Mean Prediction Error
GBM-BRT	29.606	0.944 ± 1.944 s
GAM	27.127	1.065 ± 1.715 s
MARS	22.758	0.932 ± 1.495 s

GBM Runtime Model



	Residual Sum of Squares	Mean Prediction Error
GBM-BRT	9.711	0.982 ± 1.283 s
GAM	7.551	1.024 ± 1.310 s
MARS	5.771	0.974 ± 1.326 s

Relative Influence of Model Predictors

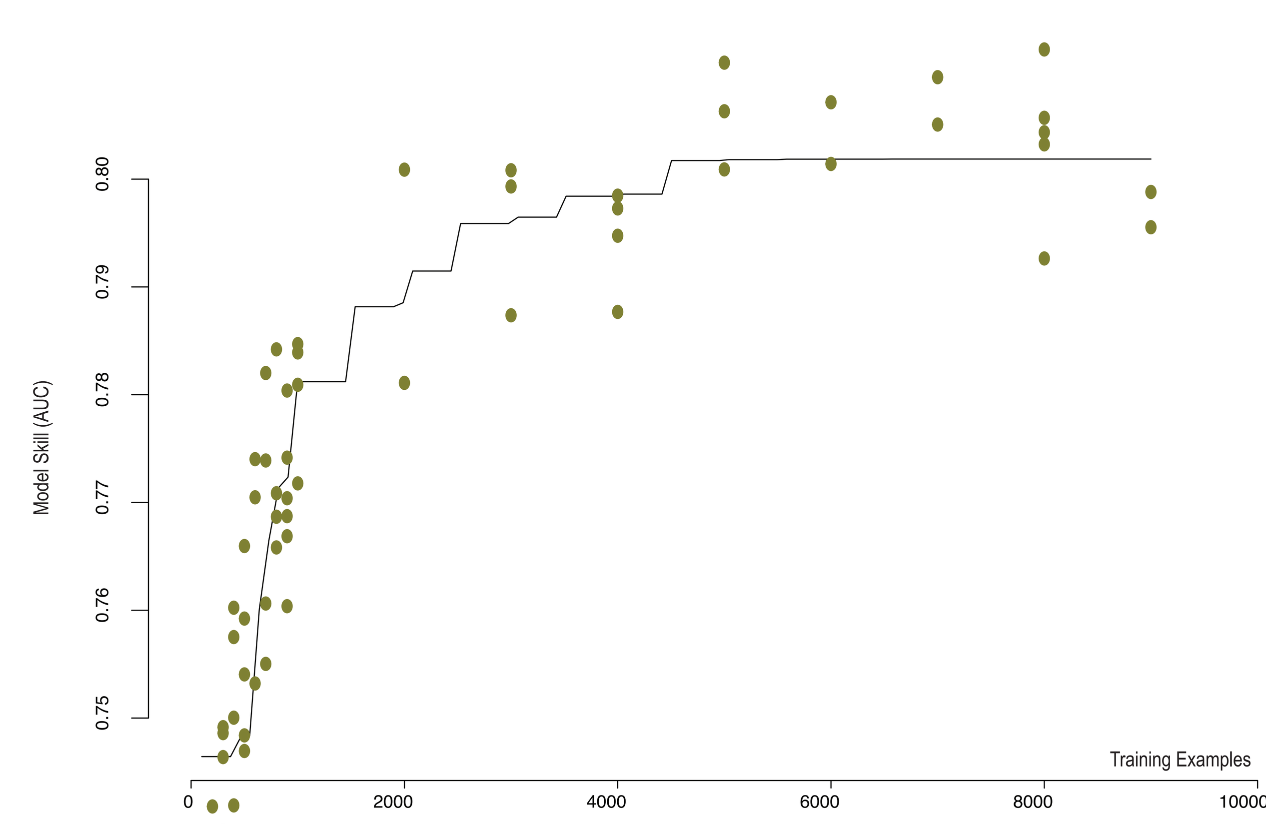


Model execution time is strongly dependent on the number of training examples using to fit the SDM. In all cases, the number of training examples and spatial resolution are **highly significant** ($p < 0.001$) predictors.

Computer hardware variables are not significant predictors of execution time for these SDMs. In some cases, additional memory was shown to reduce model speed, due to the increased overhead of memory management. Runtime logs suggest that these models are **CPU-bound**, demonstrating the need for parallel methods for SDM.

Modeling Accuracy

We also modeled the expected accuracy as a function of training examples used to fit the model, using the Area Under the Receiver Operator Curve (AUC) as an estimate of model skill. These models show significant accuracy gains are achieved by fitting models with more than 2000 training examples. All three SDM algorithms showed a positive, nonlinear relationship between training examples and model accuracy. The accuracy model demonstrates an observed-to-predicted correlation of 0.9 and a mean prediction error of 0.002 ± 0.014 AUC. Fossil data can supplement modern occurrence data by exposing the model to additional states of the climate system not present in the modern period.



Residual Sum of Squares	0.015
Mean Prediction Error	0.002 ± 0.014

Conclusion

Species Distribution Modeling algorithms are nearly all sequential, making them CPU-bound and unable to leverage parallel and distributed computing infrastructure. Typical modeling workflows do not currently require computing configurations larger than a modern desktop machine. Rapid growth trends in biodiversity databases suggest that model developers should direct attention towards new model implementations that take advantage of ensemble parallelism and can support millions of training examples.

Future Work

- Assess the potential for parallel ensemble methods
- Test runtime sensitivity to algorithm settings parameters
- Predict optimal computation configuration for given modeling scenario