Team Tealeaf: Time Series Forecasting with Manifold Forests

Vivek Gopalakrishnan[1], Morgan Sanchez[1], Suyeon Ju[1], and Celina Shih[1]

[1]Department of Biomedical Engineering, Johns Hopkins University

1 Description of the project and the problem to be solved

Ensemble methods like Random Forests (RF) are among the most accurate and computationally efficient learners for a variety of machine learning tasks. Recent extensions to RF developed in the NeuroData ecosystem like Sparse Projection Oblique Randomer Forests (SPORF) and Manifold Forests (MORF) have further improved these powerful algorithms by allowing them to more intelligently split the feature space and utilize inherent structure in data. Despite its numerous advantages, SPORF has two main drawbacks that prevent it from being widely used in many applied machine learning settings: first, it is difficult to install and second, it is currently not capable of performing regression. Our yearlong goal is to both improve the usability of SPORF by helping to integrate it into scikit-learn, the leading package for machine learning in Python, and by extend its functionality to settings such as univariate/multivariate regression and time series forecasting.

We will help integrate SPORF into scikit-learn by evaluating the algorithm through extensive benchmarking tasks. Benchmarking is the practice of comparing machine learning models on metrics like computational efficiency and accuracy. We will benchmark SPORF by measuring its performance on a comprehensive suite of classification data sets relative to other popular algorithms. These data sets will come from OpenML, an open science platform for machine learning.

Our second goal is to develop the methodology for regression using SPORF. Random forest regression naturally generalizes to SPORF when the output variables are univariate, but in higher dimensions, extensive experiments into possible split criteria have not been conducted. This year, we will develop an ideal split criteria for multivariate regression with SPORF, implement our algorithm, and PR this extension into scikit-learn. Lastly, we will leverage MORF's ability to work with structured data and our multivariate regression algorithm to develop a SPORF-based time series forecasting algorithm.

2 Timeline and tasks for each team member

2.1 Vivek

2.1.1 Sprint 1

- **Develop a theory for multivariate regression using SPORF.** Create multiple split criteria and evaluate them through a series of experiments. The final deliverable is a set of clear Jupyter Notebooks and a summary of results in a single Overleaf document.
- **Implement our SPORF-based multivariate regression algorithm.** The final deliverable is a PR into NeuroDataDesign fork of scikit-learn.
- **Develop a theory for time series forecasting using MORF.** The final deliverable is a summary of proposed methods in a single Overleaf document.
- **Implement our MORF-based time series forecasting algorithm.** The final deliverable is a PR into the NeuroDataDesign fork of scikit-learn.

2.1.2 Sprint 2

In Sprint 2, I want to add functionality to the regression trees developed in Sprint 1. Specifically, I want to add the following tools:

- **Automatic hyperparameter selection using BOHB.** I will leverage an existing Python implementation of BOHB to automatically select optimal parameters for regression trees [1].
- **Automatic selection of number of trees.** Given a data set, automatically select the number of trees that guarantees convergence to the minimal mean squared error (MSE) value. To do this, I will implement the algorithm

proposed in [2].
• **Quantile estimates at the leaf nodes.** The typical prediction returned by a Random Forest is the average of the y's in the leaf nodes of each tree. Instead, I will implement a quantile regression forest which returns confidence estimates of the conditional mean $\mathbb{E}[Y|X]$ based on [3].

The final deliverable for each of these tasks is a PR into the NeuroDataDesign fork of scikit-learn.

2.1.3 Sprint 3

• **Write paper(s) about extensions for multivariate regression and time series forecasting.** SPORF-based regression would most likely work best as an addition to the SPORF paper and time series forecasting adaptations would most likely work best as its own paper. Submit to Journal of Machine Learning (JMLR) or more appropriate journal based on advice from Jovo.

2.2 Morgan

2.2.1 Sprint 1

Write a bestSplitter class using scikit-learn framework for MORF. The final deliverable is a pull request into the NeuroDataDesign fork of scikit-learn.

Steps to Achieve This:

• Learn about SPORF, MORF, univariate regression, and multivariate regression. Read papers and other online sources to learn theory behind these topics.
• Get acquainted with the MORF code in the SPORF repository and convert MORF code to Cython using the scikit-learn framework. This will involve learning C++ and writing a bestSplitter class.
• Extend MORF to regression and write this using the scikit-learn framework. The final deliverable will be a PR along with a short demo notebook.

2.2.2 Sprint 2

Use MORF-based time-series forecasting to build a structured data imputation tool. The final deliverable will be a Jupyter Notebook demonstrating data imputation on EEG signals with missing and/or noisy data.

Steps to Achieve This:

• Extend MORF to multivariate regression.
• Develop a plan to use MORF-based time series forecasting to fill in missing or noisy EEG data. One possible approach is to forecast the signal forwards using data on the left side of the missing timesteps and backwards using data on the right side of the missing timesteps, and then average the two predictions.
• Write a demo notebook that performs imputation on EEG data that has been altered to contain missing data, and compare predictions to actual removed data visually.

2.2.3 Sprint 3

• **Help write paper(s) about extensions for multivariate regression and time series forecasting.** SPORF/MORF-based regression would most likely work best as an addition to the SPORF/MORF paper and time series forecasting adaptations would most likely work best as its own paper. Submit to Journal of Machine Learning (JMLR) or more appropriate journal based on advice from Jovo.

2.3 Celina

2.3.1 Sprint 1

• **Optimize hyperparameters using BOHB method for benchmarking SPORF.** The final deliverable is a figure showing median rank of hyperparameters to justify the best hyperparameters.
• **Implement our MORF-based time series forecasting algorithm.** The final deliverable is a PR into either scikit-learn or Jesse's fork of scikit-learn.

2.3.2 Sprint 2

• **Use MORF-based time series forecasting to predict changes to connectomes over time.** Using data provided by collaborators at Duke (Al Johnson), I will predict changes to the connectomes of mouse models of autism over time. This approach will be used to find the Regions of Interest (ROIs) that change most significantly over time within and between different strains of mice.

2.3.3 Sprint 3

• **Help write paper(s) about extensions for multivariate regression and time series forecasting.** SPORF-based regression would most likely work best as an addition to the SPORF paper and time series forecasting adaptations would most likely work best as its own paper. Submit to Journal of Machine Learning (JMLR) or more appropriate journal based on advice from Jovo.

2.4 Suyeon

2.4.1 Sprint 1

• **Perform benchmarks the current, available SPORF classification algorithm using default parameters.** Write benchmarking code given the current, available SPORF algorithm. The final deliverable is a set of Jupyter Notebooks and technical reports.
• **Perform benchmarks the current, available SPORF classification algorithm using optimized hyperparameters.** Write benchmarking code given the current, available SPORF algorithm. The final deliverable is a set of Jupyter Notebooks and technical reports.
• **Implement our MORF-based time series forecasting algorithm.** The final deliverable is a PR into either scikit-learn or Jesse's fork of scikit-learn.

2.4.2 Sprint 2

• **Predict the onset of seizures using SPORF/MORF algorithm.** A natural application of the SPORF algorithm to EEG data is the task of seizure detection. Using an EEG data set provided by Adam, I will classify pre-seizure states using MORF. The final deliverable is a set of Jupyter Notebooks and a technical report.

2.4.3 Sprint 3

• **Write paper(s) about extensions for multivariate regression and time series forecasting.** SPORF-based regression would most likely work best as an addition to the SPORF paper and time series forecasting adaptations would most likely work best as its own paper. Submit to Journal of Machine Learning (JMLR) or more appropriate journal based on advice from Jovo.

References

[1] Stefan Falkner, Aaron Klein, and Frank Hutter. BOHB: Robust and Efficient Hyperparameter Optimization at Scale. page 10. 1

[2] Miles E. Lopes, Suofei Wu, and Thomas C. M. Lee. Measuring the Algorithmic Convergence of Randomized Ensembles: The Regression Setting. *arXiv:1908.01251 [cs, math, stat]*, August 2019. 2

[3] Nicolai Meinshausen. Quantile Regression Forests. page 17. 2