SPORF - Summary

September 2019

Decision forests can be random forests, gradient boosting trees, or SPORF (where very sparse random projections are used). In random forests decision trees are the base learners whose constrution progresses with the estimation of Gini impurity. XGBoost is a gradient boosted decision tree and is popular for its speed and scalability. SPORF is an ensemble technique that has the desirable properties of RF and oblique forest methods - robustness, computational efficiency, scalability, and interpretability. But given SPORF searched for splits over sparse random projection, it can not find splits when a signal is contained in a dense linear combination of features or non-linear combination of features.

What makes SPORF desirable are its properties that incluse random search for splits which helps in avoiding overfitting, flexible sparsity, ease of tuning, data insight which is the assessibility of relative contribution of each feature to the model, expediency, and scalability.

In each algorithm 500 trees are used where the split objective is the reduction in Gini impurity. The classification trees are fully grown unpruned and the hyperparameters one of which is the average sparsity of univariate projections sampled at each split node, are tuned via out-of-bag error.

On comparing algorithms using the same implementation, SPORF's performance falls in between that of F-RC and RF which is the slowest, but it improves in accuracy with additional training.

SPORF and other oblique forests are more consistent than random forests. Also, while RF cannot learn new features, SPORF learns important features. In addition to combining the best of existing axis-aligned and axis-oblique methods, SPORF is robust to hyperparameter selection. SPORF also identifies default hyperparameter settings which can be of great value for those who use it out of the box. It is also robust to high-dimensional noise. SPORF was compared against other techniques as well in terms of training and prediction times. SPORF performed well upon strong-scaling (relative increase in speed of using multiple cores over taht of using single core) and forest packing (reorganizing and restructuring a forest to reduce prediction latency).