Adaptive Locally Weighted Random Forest (AWL-RF)

Akansha Bagga, Ruhani Chhabra, Lucas Libelo
Thomas Jefferson High School for Science and Technology
Advanced Machine Learning 1 Quarter 2 Project
Dr. Yilmaz
February 2, 2025

Abstract

The Adaptive Locally Weighted Random Forest (ALW-RF) algorithm enhances the traditional random forest methodology by addressing key limitations related to uniform feature treatment and computational redundancy. Unlike conventional random forest algorithms that assume equal predictive power across all features, AWL-RF assigns differential weights to attributes based on their predictive significance, calculated using Information Gain, Gain Ratio, and Pearson Correlation. This mechanism influences tree weighting during the ensemble voting process, with weights being assigned adaptively in accordance with the K nearest neighbors of the test point. The result is more efficient and accurate, and reduces the impact of irrelevant features, improves computational efficiency, and achieves higher predictive accuracy. We tested the AWL-RF algorithm with 3 different attribute selection methods in addition to the standard random forest algorithm and concluded that the AWL-RF algorithm outperformed the standard random forest algorithm in almost every case.

Introduction

Traditional random forest (RF) algorithms are powerful, but suffer from two critical limitations: (1) Uniform feature treatment, in which they assume that all attributes have equal predictive power, ignoring the inherent differences in feature importance, and (2) Computational redundancy, by which they often include irrelevant attributes, increasing computational costs without improving accuracy.

To address these limitations, AWL-RF integrates feature selection directly into the forest's structure by weighting trees based on the aggregate local predictive power of their selected attributes. This approach leverages metrics like Information Gain, Gain Ratio, and Pearson Correlation to guide both feature selection and tree weighting, aiming to reduce redundancy and enhance accuracy.

Related Work

The integration of feature selection and ensemble learning has been extensively explored to address the limitations of traditional random forests. Early work by Cuong Nguyen et al. [1] demonstrated that preprocessing datasets with filter-based feature selection (e.g., Information Gain) before training a random forest reduces computational costs. However, their method treats feature selection as a static step, decoupling it from tree construction and ignoring dynamic feature interactions during training. Similarly, Díaz-Uriarte & Alvarez de Andrés [2] proposed GeneSrF, which recursively eliminates low-importance features during forest training using Gini importance. While dynamic, their reliance on a single metric overlooks complementary criteria like Gain Ratio or correlation, limiting robustness in heterogeneous datasets like medical risk prediction.

Weighted ensemble approaches have also sought to prioritize high-performing trees. Hong et al. [3] weighted trees by their out-of-bag (OOB) error, but this method treats trees as black boxes, failing to account for why a tree performs well—such as its use of predictive features. Wang et al. [4] introduced Regularized Random Forest (RRF), which penalizes redundant features during splits using a fixed regularization parameter. Though effective, RRF's static regularization cannot adapt to varying feature importance across datasets. More recently, Khan et al. [5] correlated feature-class relationships to weight trees but assumed linear interactions, struggling with non-linear dependencies common in medical data (e.g., age and cholesterol).

Hybrid methods bridging feature selection and weighting have emerged but remain incomplete. Amalia Utamima et al. [6] combined Information Gain with random forests for diabetes prediction, ranking features statically and training forests on top subsets. Like Nguyen's work, their approach lacks

dynamic adaptation during training. Chen et al. [7] advanced this with Dynamic Feature Selection Forests (DFS-Forest), reselecting features at each node using entropy thresholds. However, DFS-Forest does not propagate feature importance to tree weights, missing opportunities to prioritize trees with high-impact feature subsets.

Our work, Adaptive Locally Weighted Random Forest (AWL-RF), addresses these gaps by unifying dynamic feature selection and tree weighting, while adding a localized aspect to increase minority classification accuracy. Unlike static preprocessing or global regularization, AWL-RF adaptively assigns tree weights based on the collective predictive power of their feature subsets.

Methods

This algorithm involves several stages: feature evaluation, tree construction, K nearest neighbors calculation, and ensemble classification. Data is initially loaded from CSV files into dictionaries, with numerical and categorical features identified through type inference. Feature evaluation metrics, including Information Gain, Gain Ratio, and Pearson Correlation are computed to assess each attribute's predictive power.

In the construction phase, each decision tree within the forest is constructed from a bootstrap sample of the training data, with a randomly selected subset of features. The weights are not assigned until the classification phase begins.

Before the classification phase, we run a k-nn algorithm on the training data to determine the K nearest neighbors of all test points. When classifying test points, tree weights are dynamically assigned using attribute selection algorithms that are calculated with the nearest neighbors of the test point. Tree weights are determined based on the aggregate predictive power of their selected attributes, computed using Information Gain, GainRatio, or Pearson Correlation from the K nearest neighbors. This ensures that trees relying on more relevant features contribute significantly to the final prediction.

During the classification phase, each tree votes on the class label for a given test sample. The voting process incorporates tree weights, amplifying the impact of more reliable trees. Specifically, the classification result from each tree is repeated in the voting pool proportional to its weight, and the final prediction is determined by majority vote. Performance evaluation is conducted using accuracy metrics and confusion matrices.

Dataset and Preprocessing

Our dataset is on student depression analysis (Opeyemi). It is a dataset that provides academic, geographic, and other information about students. We will be classifying whether or not a student has depression, based on factors such as age, gender, and academic/work pressure, profession, and job satisfaction. We decided to use this dataset because of the large number of instances, ~27,900, and its application toward today's mental health landscape.

Attribute	Description
ID	ID of student
Gender	Gender of student
Age	Age of student
City	What city the student lives in

Profession	What job the student holds
Academic Pressure	Level of academic pressure on a scale from 1.0 to 5.0
Work Pressure	Level of job pressure on a scale from 1.0 to 5.0
CGPA	Cumulative grade point average, on a 10.0 scale
Study Satisfaction	Satisfaction of academic performance on a scale from 1.0 to 5.0
Job Satisfaction	Satisfaction of work-related performance on a scale from 1.0 to 5.0
Sleep Duration	How many hours the student sleeps per night on average
Dietary Habits	Health of student's nutrition, ranked by low, moderate, high
Degree	What degree the student is working towards (BE, PhD, etc.)
Suicidal Thoughts	Whether the student has had recent suicidal thoughts
Work/Study Hours	How many hours the student works/studies per day
Financial Stress	Stress related to student's finances, on a scale from 1.0 to 5.0
Family History	Whether the student has family history of mental illness
Depression	Class variable, whether the student has depression or not

Discretization was only necessary for the class variable to convert to binary, as both categorical data and numerical data were able to be passed through our implementation of random forest. Attribute selection would further impede on our later use of tree weighting based on attribute correlation. Numerical values were normalized using min-max normalization. We used scikit-learn to separate our dataset into train and test files, using a 70-30 split.

Experiments

To measure the performance of the weighted random forest algorithm, we combined it with the following attribute selection methods to provide weights: Pearson Correlation, InfoGain, and GainRatio. We also compared our results to a standard random forest. Each forest we created used the same number of trees, and number of instances and attributes used to train each tree. Due to the size of our dataset, running the random forest algorithm with a large number of trees took an unreasonable amount of time, so we chose to use 100 trees per forest. Each tree has a sample size of 6510 (⅓ of the number of instances in the dataset), and each tree was trained on four attributes (log2 of the number of attributes). We experimented with different k-values and conducted three experiments with each attribute evaluation method using the best k-value.

Results

Accuracy

Attribute evaluation method	Trial 1	Trial 2	Trial 3	Average

Standard RF	0.7528	0.7679	0.7337	75.15%
Correlation	0.8108	0.8011	0.8036	80.52%
InfoGain	0.8028	0.8077	0.8077	80.61%
GainRatio	0.7894	0.7922	0.7900	79.22%

Precision

Attribute evaluation method	Trial 1	Trial 2	Trial 3	Average
Standard RF	0.8211	0.8300	0.8141	82.17%
Correlation	0.8333	0.8083	0.8183	82 %
InfoGain	0.8085	0.8328	0.8207	82.07%
GainRatio	0.7903	0.8037	0.7920	79.53%

Recall

Attribute evaluation method	Trial 1	Trial 2	Trial 3	Average
Standard RF	0.7108	0.7287	0.6874	70.90%
Correlation	0.7861	0.7823	0.7813	78.32%
InfoGain	0.7851	0.7818	0.7865	78.45%
GainRatio	0.7739	0.7784	0.7736	77.53%

Specificity

Attribute evaluation method	Trial 1	Trial 2	Trial 3	Average
Standard RF	0.4441	0.4801	0.3936	43.93%
Correlation	0.6293	0.6634	0.6398	64.42%
InfoGain	0.6727	0.6174	0.6523	64.75%
GainRatio	0.6758	0.6597	0.6699	66.85%

Confusion Matrices

Attribute evaluation method	Trial 1	Trial 2	Trial 3
Standard RF			

	0	1

	0	1

	0	1

	0 1566 1960 1 109 4736	0 1693 1833 1 110 4735	0 1388 2138 1 91 4754
Correlation	| | 0 | 1 | | 0 | 2219 | 1307 | | 1 | 277 | 4568 |	| | 0 | 1 | | 0 | 2339 | 1187 | | 1 | 478 | 4367 |	| | 0 | 1 | | 0 | 2256 | 1270 | | 1 | 374 | 4471 |
InfoGain	| | 0 | 1 | | 0 | 2372 | 1154 | | 1 | 497 | 4348 |	| | 0 | 1 | | 0 | 2177 | 1349 | | 1 | 261 | 4584 |	| | 0 | 1 | | 0 | 2300 | 1226 | | 1 | 384 | 4461 |
GainRatio	| | 0 | 1 | | 0 | 2383 | 1143 | | 1 | 620 | 4225 |	| | 0 | 1 | | 0 | 2326 | 1200 | | 1 | 498 | 4347 |	| | 0 | 1 | | 0 | 2362 | 1164 | | 1 | 594 | 4251 |

Discussion

In almost every case, locally adaptive weighted random forest algorithms performed better than the standard random forests. Pearson Correlation, InfoGain, and GainRatio all had an average accuracy around 5% higher than that of the standard RF. Variations in confusion matrices depended on the attributes selected for the trees in each forest. Random forest often has difficulties performing well on skewed data. Recall, and specificity were also around 7% and 21% higher for the adaptive weighted forests than the standard random forest. Given the distribution of our dataset, the higher specificity values suggest that our algorithm is better at handling skewed data than the RF algorithm. The average precision of our algorithm with every attribute selection method was slightly lower than the precision of the standard RF, however, this difference in precision is extremely small, suggesting our algorithm only decreases the precision very slightly. The differences in recall, specificity, and precision were almost the same for every attribute selection method used with the adaptive weighted forests.

The attribute selection algorithm with the best accuracy was InfoGain, with an average accuracy of 80.61%, 5.46% better than that of the standard RF, but only 0.09% better than correlation, the next best algorithm. The variations in performance of different attribute selection methods are likely due to the nature of the dataset and the attribute selection methods, not of our algorithm. Precision and recall were

also very similar among the different attribute selection methods, with differences of 0.07% and 0.13% between the best and second best algorithms, respectively. The adaptive weighted forests using InfoGain had the best specificity with a larger margin, 2.1% better than Infogain, the next best algorithm.

One potential improvement of our algorithm in the future is combining multiple attribute selection methods in a single tree. This might be difficult because each method has a different range of values given to attributes. If we were to perform this study again with more computing power, we would create larger forests, explore higher values of K, and test our algorithm on various datasets to ensure it improves upon random forest in many different cases.

Contributions

The allocation of work among the three group members is as follows:

Akansha:

- Came up with KNN idea
- Background research
- Found dataset
- Code for the random forest and KNN
- Abstract, Introduction, Related Works, and Methods sections

Ruhani:

- Background research
- Code for attribute selection
- Ran the experiments
- Created presentation
- Methods, and Dataset and Preprocessing sections

Lucas:

- Came up with weighted RF idea
- Background research
- Code for the weighted random forest
- Created presentation
- Abstract, Experiments, Discussion, and References sections

References

Lin, R., He, X., Feng, J., Zalmout, N., Liang, Y., Xiong, L., & Dong, X. L. (2021). *PAM: Understanding product images in cross product category attribute extraction* [Paper presentation]. 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. https://doi.org/10.1145/3447548.3467164

Menze, B. H., Kelm, B. M., Masuch, R., Himmelreich, U., Bachert, P., Petrich, W., & Hamprecht, F. A. (2009). A comparison of random forest and its gini importance with standard chemometric methods for the feature selection and classification of spectral data. *BMC Bioinformatics*, *10*(1). https://doi.org/10.1186/1471-2105-10-213

Peker, M., Arslan, A., Şen, B., Çelebi, F. V., & But, A. (2015). A novel hybrid method for determining the depth of anesthesia level: Combining ReliefF feature selection and random forest algorithm (ReliefF+RF). 2015 International Symposium on Innovations in Intelligent Systems and Applications (INISTA), 1–8. https://doi.org/10.1109/INISTA.2015.7276737

Nguyen, C., Wang, Y., & Nguyen, H. N. (2013). Random forest classifier combined with feature selection for breast cancer diagnosis and prognostic. *Journal of Biomedical Science and Engineering*, *06*(05), 551-560. https://doi.org/10.4236/jbise.2013.65070

Reif, D. M., Motsinger, A. A., McKinney, B. A., Crowe, J. E., & Moore, J. H. (2006). Feature selection using a random forests classifier for the integrated analysis of multiple data types. 2006 IEEE Symposium on Computational Intelligence and Bioinformatics and Computational Biology, 1–8. https://doi.org/10.1109/CIBCB.2006.330987

R., U., & Ghosh, S. (2019). Maxima-turn-switching strategy of sensor-equipped UAVs for target localization in 2-D and 3-D environments. 2019 IEEE 15th International Conference on Control and Automation (ICCA), 610–615. https://doi.org/10.1109/ICCA.2019.8900007

Uduagbamen, P. K., Sanusi, M., Udom, O. B., Salami, O. F., Adebajo, A. D., & Alao, O. J. (2020). Metabolic acidosis in the surgical intensive care unit: Risk factors, clinical correlates and outcome. findings from a high dependency heart and vascular surgical center in nigeria. *World Journal of Cardiovascular Surgery*, *10*(11), 226-241. https://doi.org/10.4236/wjcs.2020.1011025

Opeyemi, S. (2025). Student depression dataset. Kaggle. https://www.kaggle.com/datasets/hopesb/student-depression-dataset