A Novel Personalized Random Forest Algorithm for Clinical Outcome Prediction

Adriana Johnson[a], Gregory F. Cooper[a], Shyam Visweswaran[a]

[a]Department of Biomedical Informatics, University of Pittsburgh, Pittsburgh, Pennsylvania, United States of America

Abstract

Machine learning algorithms that derive predictive models are useful in predicting patient outcomes under uncertainty. These are often "population" algorithms which optimize a static model to predict well on average for individuals in the population; however, population models may predict poorly for individuals that differ from the average. Personalized machine learning algorithms seek to optimize predictive performance for every patient by tailoring a patient-specific model to each individual.

Ensembles of decision trees often outperform single decision tree models, but ensembles of personalized models like decision paths have received little investigation. We present a novel personalized ensemble, called Lazy Random Forest (LazyRF), which consists of bagged randomized decision paths optimized for the individual for whom a prediction will be made.

LazyRF outperformed single and bagged decision paths and demonstrated comparable predictive performance to a population random forest method in terms of discrimination on clinical and genomic data while also producing simpler models than the population random forest.

Keywords:

Machine Learning, Decision Trees, Algorithms

Introduction

Personalized medicine calls for care that is tailored to each individual, and predictive models can be useful for supporting such care [19]. Using machine learning algorithms, predictive models can be trained on large biomedical datasets, and these models can then be applied using patient information to perform inference and make predictions for individuals [14].

Most algorithms fit a model (or ensemble of models) using a training dataset in an "eager" fashion, constructing the model prior to encountering a new individual for whom a prediction will be made [18]. The model is optimized to predict well on average for any future member of the population represented by the training data. However, important but uncommon features may not be captured by a "population" model derived using an eager machine learning algorithm, and this may result in lower predictive performance for certain subgroups of the population. The "average best choice" may not be the best choice for an individual [2]. This is a major shortcoming when using predictive models to support personalized medicine, where the goal is to achieve optimal care for each individual [4].

An alternative paradigm is to fit a model using information from the individual for whom a prediction will be made [17]. This "lazy" approach delays model fitting until information regarding the features present in the individual of interest are known, and a personalized predictive model is tailored to that individual. This personalized model is optimized to predict well for the individual, rather than the population on average. When that individual is a patient, the personalized model is called a patient-specific model.

Prior Work

Personalized algorithms have been developed that produce patient-specific ensembles in several ways. Random forest is an ensemble population machine learning algorithm that produces randomized bagged decision trees [3], and one personalized random forest approach uses personalized bootstrap datasets to derive randomized decision trees [12,21]. Another approach derives patient-specific base models where each model in the ensemble is a patient-specific decision tree tailored to the individual of interest. The Lazy Decision Tree (LazyDT) method was the first algorithm that derived personalized decision tree models using features present in the the individual of interest [9]. Since the personalized decision tree is a single path, we refer to the personalized decision tree model as a decision path model.

Several decision path algorithms have been described in the literature and have been shown to have superior predictive performance compared to population decision tree algorithms [7,9,10,15,18]. However, little work has been done on ensembles of patient-specific decision paths. Fern et al. introduced a boosting algorithm to construct boosted ensembles of LazyDTs, and found that this algorithm resulted in higher accuracy than single and bagged LazyDTs [6]. Margineantu et al. used a bagging algorithm with lazy option trees which resulted in better calibrated probability estimates than corresponding population algorithms [13]. We are interested in using other ensemble approaches to improve the performance of decision path models, and to our knowledge no personalized random forest algorithm has been described that derives ensembles of patient-specific decision paths.

Hypothesis

We hypothesized that a personalized random forest consisting of randomized patient-specific decision paths would outperform single decision paths and population random forest models in terms of discrimination and would also produce simpler models.

LazyRF (*D*, *Test, B*)
Inputs: *D* is a training dataset consisting of *m* training cases and *n* variables and a target variable *T*
Test is the test case (patient of interest) for which a prediction will be made
B is the number of models in the ensemble

```
1  F ← empty forest of paths
2  P ← empty list of predictions
3  Generate B bootstrap datasets by randomly drawing m training cases from D with replacement
4  FOR each bootstrap dataset:
5     S ← empty path
6     LOOP:
7        BestScore ← 0; BestFeature ← ∅
8        Randomly select √n candidate features from Test
9        FOR each randomly selected candidate feature X = x:
10          S'← add X = x to S
11          D_S' ← training cases in bootstrap dataset that satisfy S'
12          IF D_S' ≠ ∅:
13             Score ← EntScore (S) - EntScore(S') calculated using Equation 3
14             IF Score > BestScore: BestFeature ← X = x
15       IF BestFeature ≠ ∅:
16          Add BestFeature to S
17       ELSE: Exit from LOOP
18    F ← add S to F
19    P ← add P_S(T | Test) which is estimated using Equation 1 from training data that satisfy S
```

Output: Forest of paths ***F*** and $P_F(T \mid \textbf{\textit{Test}})$ calculated as the mean of the predictions in ***P***

Figure 1 – Pseudocode of LazyRF algorithm.

Methods

Our novel algorithm is called Lazy Random Forest (LazyRF). In this section, we first provide details of LazyRF and the algorithms used for comparison. We then describe the experimental approach used to evaluate the algorithms' performance.

Algorithmic Methods

The base method used in LazyRF is a randomized patient-specific decision path that optimizes an entropy score. The randomized decision path (RDP) algorithm differs from a regular decision path in terms of the search, as RDP only evaluates random subsets of features from the patient of interest (known as the test case) when constructing a decision path.

The pseudocode of the LazyRF algorithm is given in Figure 1. First, we describe the structure of a decision path model, then we provide details on the search and score used by the RDP algorithm. We then explain how LazyRF trains RDPs on bootstrap datasets to produce the patient-specific ensemble. We conclude with descriptions of the comparison algorithms.

Decision Path – Model Structure

A decision path model M is represented as $M = (S, \theta)$, where S is a path and θ are the parameters of the probability distributions over the target variable T. Let $\boldsymbol{V} = (X_1, X_2, \ldots, X_i, \ldots, X_n)$ be a list of the n variables describing training dataset D. Let a feature $X_i = x_i$ be defined as a variable-value pair. A decision path S consists of a conjunction of q features such that $S = (X_a = x_a \wedge X_b = x_b \wedge \ldots \wedge X_j = x_j \wedge \ldots \wedge X_q = x_q)$. The variable list $\boldsymbol{V_S} = (X_a, X_b, \ldots, X_j, \ldots, X_q)$ is a subset of $\boldsymbol{V}$, and the value list $\boldsymbol{v_S} = (x_a, x_b, \ldots, x_j, \ldots, x_q)$ consists of the values of test case (patient of interest) for the variables in $\boldsymbol{V_S}$. The target variable T can take r possible values, denoted by $(t_1, t_2, \ldots, t_k, \ldots, t_r)$. The parameter list $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_k, \ldots, \theta_r)$ denotes the r probabilities for the distribution $P(T \mid \boldsymbol{V_S} = \boldsymbol{v_S})$ over T. The values of those probabilities are estimated from D_S, which contains the training cases in D for which $\boldsymbol{V_S} = \boldsymbol{v_S}$. We use the maximum likelihood estimator to calculate the probabilities $\boldsymbol{\theta}$ associated with a path S. The estimate for probability θ_k is given by

$$\theta_k \equiv P(T = t_k | V_S = v_S) = \frac{N_k}{N}, \quad (1)$$

where N is the number of training cases in D_S, and N_k is the number of those cases in D_S for which $T = t_k$.

Randomized Decision Path – Model Search

An RDP model for a test case is constructed by performing a greedy hill-climbing search, adding features from the test case one at a time that optimize an information gain score, relative to a bootstrapped dataset. The score is calculated using D_S, the set of training cases that share all the features in path S.

Starting with an empty path S, the algorithm randomly samples $\sqrt{n}$ features that are present in the test case. For each candidate feature $X_i = x_i$, the feature is temporarily appended to path S to produce candidate path S'. The subset of training cases that share all the features in S' is referred to as $D_{S'}$. If $D_{S'}$ is empty (i.e., no training cases satisfy S'), the algorithm proceeds to the next feature. If $D_{S'}$ is not empty, the algorithm calculates the entropy of S' with $D_{S'}$ and determines the difference in entropy between S and S'. This difference is the information gain score of candidate path S' containing candidate feature $X_i = x_i$.

The highest scoring candidate feature is added to S, and a new random subset of $\sqrt{n}$ features is sampled. The search stops when the score cannot be improved, and a new bootstrapped dataset is considered (see below).

Randomized Decision Path – Model Score

For a path S and data D_S that contains the training cases that satisfy the path S, the entropy is given by:

$$H(S) = -\sum_{k=1}^{r} P(T = t_k \mid \boldsymbol{V_S} = \boldsymbol{v_S}) \log_2 P(T = t_k \mid \boldsymbol{V_S} = \boldsymbol{v_S}), \quad (2)$$

where $P(T = t_k \mid \boldsymbol{V_S} = \boldsymbol{v_S})$ is the proportion of the dataset D_S that includes training cases that have the value t_k for T. Using

Equation 1 to calculate these probabilities gives us the following entropy score:

$$EntScore(S) = -\frac{1}{N}\left(\sum_{k=1}^{r} N_k \log_2 \frac{N_k}{N}\right), \quad (3)$$

where N is the number of training cases in D_S, r is the number of values the target variable T can take, and N_k is the number of training cases in D_S that take the k^{th} value of T. The algorithm selects the candidate feature corresponding to candidate path S' that results in the greatest reduction of entropy (which is equivalent to the greatest information gain) for the current path S.

Lazy Random Forest

LazyRF produces an ensemble of RDPs by training multiple patient-specific models using bootstrap datasets. A bootstrap dataset is produced from a training dataset of m training cases by randomly sampling with replacement m times from the training dataset. The result is a bootstrap dataset containing on average 63% of the training cases from the original training dataset, with some of those cases appearing multiple times [5].

LazyRF constructs 25 bootstrap datasets, and an RDP is trained on each bootstrap. The ensemble prediction of LazyRF is calculated as the average of the individual path predictions.

Comparison Algorithms

We compared LazyRF to three algorithms: a single non-randomized decision path, a bagged ensemble of 25 non-randomized decision paths, and a standard population random forest of randomized decision trees.

The single non-randomized decision path (DP) differs from the LazyRF algorithm in three ways: it builds a single decision path model (rather than an ensemble), it uses the entire training dataset (rather than bootstrap datasets), and it considers all available features (rather than a random subset of features, described in line 8 of Figure 1). The structure, search, and score of DP are otherwise unaltered.

The ensemble of bagged non-randomized decision paths (BagDP) differs from the LazyRF algorithm in only one way: it considers all available features (rather than a random subset of features, described in line 8 of Figure 1). Like LazyRF, BagDP trains decision path models on 25 bootstrap datasets, and the decision path structure, search, and score are otherwise unaltered. Like LazyRF, the prediction of BagDP is calculated as the average of the individual path predictions.

The population random forest (RF) consists of randomized decision trees (which are population models) trained on 25 bootstrap datasets. A randomized decision tree that optimizes information gain in terms of entropy is trained on each bootstrap dataset. Like LazyRF, the ensemble prediction of RF is calculated as the average of the individual tree predictions.

Experimental Methods

Datasets

Six clinical and genomic datasets were used for evaluation. These datasets consist of real patient information collected for research purposes. Details regarding the datasets are described in Table 1. All datasets had binary target variables and were divided into approximately 80%/20% train-test splits with similar proportions of positive cases (i.e., individuals who have disease) in training and test datasets.

The chronic pancreatitis dataset is comprised of single nucleotide variants from individuals with chronic pancreatitis and from healthy controls, and the target is presence of chronic pancreatitis [20]. The pneumonia dataset is comprised of clinical, laboratory, and radiographic findings of patients admitted with community acquired pneumonia, and the target is whether a patient experienced a dire outcome (defined as death within 30 days of presentation, intensive care unit admission, or another severe complication) [8]. The sepsis datasets are comprised of demographics, clinical findings, and genetic and inflammatory markers of patients admitted with community acquired pneumonia [11]. The target of the sepsis-d dataset is death within 90 days of enrollment, and the target of the sepsis-s dataset is development of severe sepsis during hospitalization. The heart failure dataset is comprised of demographics and clinical, laboratory, radiographic, and electrocardiographic findings of patients admitted with heart failure [1]. The target of heart-failure-d is death during hospitalization, and the target of heart-failure-c is development of one or more serious complications (including death) during hospitalization.

Table 1 – Brief descriptions of datasets.

Dataset	# Vars	# Cases	# Positive (%)
chronic-pancreatitis	142	2201	980 (44.5%)
pneumonia	156	2287	261 (11.4%)
sepsis-d	19	1673	189 (11.3%)
sepsis-s	18	1673	478 (28.6%)
heart-failure-d	17	11,178	500 (4.47%)
heart-failure-c	20	11,178	1255 (11.2%)

Experimental Protocols

We evaluated the algorithms in terms of discrimination as measured by area under the receiver operating characteristic curve (AUROC) and model complexity as measured by mean predictive path length, defined as the number of features in the path used for prediction.

For each dataset, we compared AUROCs of LazyRF vs. DP, LazyRF vs. BagDP, and Lazy RF vs. RF using DeLong's test. We compared mean AUROCs and mean path lengths of LazyRF vs. DP, LazyRF vs. BagDP, and Lazy RF vs. RF on the six datasets using the Wilcoxon signed-rank test. We implemented the algorithms with Python (version 3.7) and performed analysis of results in R using "wilcox.test" and "pROC" [16]. We performed all experiments on a MacBook Pro with a 3.3 GHz Dual-Core Intel Core i5 processor and 16GM of RAM, running the 64-bit macOS Catalina operating system.

Results

In this section, we present the AUROCs and mean path lengths of the algorithms on the six datasets.

Discrimination

The mean AUROCs were 0.733 for LazyRF, 0.583 for DP, 0.680 for BagDP, and 0.767 for RF. AUROCs for each algorithm and dataset are shown in Table 2. When the performance of the personalized algorithms was compared pairwise using DeLong's test, LazyRF had statistically significantly higher AUROCs than DP for five out of six datasets (indicated in bold in Table 2), and LazyRF had statistically significantly higher AUROCs than BagDP for three out of six datasets (indicated in italics in Table 2). When the performance of the random forest algorithms were compared pairwise using DeLong's test, RF had a statistically significantly higher AUROC than LazyRF for one out of six datasets (indicated by an asterisk in Table 2).

Table 2 – AUROCs of DP, BagDP, LazyRF and RF algorithms. Statistically significantly better performance indicated by: bold for Lazy RF vs. DP, italics for LazyRF vs. BagDP, and asterisk for LazyRF vs. RF.

Dataset	DP	BagDP	LazyRF	RF
chronic-pancreatitis	0.736	0.814	**0.847**	0.814
pneumonia	0.512	0.546	0.558	0.822*
sepsis-d	0.641	0.748	***0.843***	0.844
sepsis-s	0.551	0.743	**0.772**	0.736
heart-failure-d	0.515	0.624	***0.665***	0.656
heart-failure-c	0.545	0.605	***0.712***	0.735
mean	0.583	0.680	***0.733***	0.767

Applying the Wilcoxon signed-rank test to all six datasets, we found that LazyRF had a statistically significantly higher mean AUROC than DP ($p = 0.03$) and BagDP ($p = 0.03$) at the 0.05 level.

When mean AUROCs of LazyRF and RF were compared with the Wilcoxon signed-rank test, there was no statistically significant difference at the 0.05 level.

Model Complexity

The mean path lengths were 4.20 for LazyRF, 4.21 for DP, 3.75 for BagDP, and 6.83 for RF. Mean path lengths for each algorithm and dataset are shown in Table 3. When mean path lengths were compared with the Wilcoxon signed-rank test, LazyRF had statistically significantly shorter mean path lengths than RF at the 0.05 level ($p = 0.03$). LazyRF did not have statistically significantly different mean path lengths than DP or BagDP at the 0.05 level.

Table 3 – Mean path lengths of DP, BagDP, LazyRF and RF algorithms.

Dataset	DP	BagDP	LazyRF	RF
chronic-pancreatitis	2.44	2.31	3.51	5.22
pneumonia	2.36	2.12	4.26	5.47
sepsis-d	3.15	2.79	3.21	5.92
sepsis-s	3.95	3.56	3.12	5.68
heart-failure-d	5.96	5.17	6.05	9.68
heart-failure-c	7.42	6.54	5.06	8.99
mean	4.21	3.75	4.20	6.83

Discussion

LazyRF had the highest predictive performance of the personalized algorithms as measured by AUROC. Ensemble methods like bagging and random forest have been found to improve predictive performance over single models like decision trees by reducing variance without increasing bias, resulting in lower predictive error. As the LazyRF algorithm trains multiple models on bootstrap datasets and incorporates randomized feature selection, the models in LazyRF are possibly more diverse and may result in lower variance than DP or BagDP.

LazyRF and RF did not have statistically significantly different predictive performance as measured by AUROC. Previous work has demonstrated that decision paths can result in higher accuracies and AUROCs than population decision trees, indicating that personalization can improve predictive performance of models. In the case of random forest, however, we did not find that personalization improved predictive performance over the population algorithm. We did find that personalization resulted in more concise models than RF. By producing simpler models, the personalized ensemble may be easier to comprehend than the population ensemble, but this requires further experimental exploration.

This finding is consistent with the results of Fern et al [6]. They found that boosting decision paths resulted in higher predictive performance than bagged and single decision paths. When compared to boosted decision trees, however, boosted decision paths resulted in simpler models with comparable predictive performance. It is an open question why the improvements from personalization over population models that have been found in single model paradigms may not extend to ensembles in terms of predictive performance.

One limitation of this work is that the algorithms were only evaluated on data collected for research purposes. It is possible that the algorithms perform differently on electronic health record data or other real-world data. The datasets were limited in number and scope, and they are not open access. Additionally, the current version of LazyRF can only handle discrete data, so any continuous data requires discretization prior to use with LazyRF. Further evaluation on a wider range of data would enhance the generalizability of our findings.

In addition to testing performance on a broader set of more heterogenous data, future work includes incorporating other types of decision paths, such as those that use Bayesian scoring, into LazyRF methodology. Further inquiry is also needed to determine why ensembles of decision paths like LazyRF and boosted LazyDTs do not improve performance metrics like accuracy and AUROC over corresponding population ensembles.

Conclusions

In conclusion, applying a random forest ensemble approach to personalized decision paths is associated with improvements in predictive performance over single and bagged decision paths. Personalization of random forest through the use of patient-specific decision paths is associated with comparable predictive performance and simpler models when compared to a population random forest approach. These findings provide support for the potential use of personalized random forest algorithms for patient-specific prediction.

Acknowledgements

This work was supported by the National Library of Medicine of the National Institutes of Health under award number T15 LM007059, and by the National Institute of General Medical Sciences of the National Institutes of Health under award number T32 GM008208.

References

[1] T.E. Auble, M. Hsieh, W. Gardner, G.F. Cooper, R.A. Stone, J.B. McCausland, and D.M. Yealy, A prediction rule to identify low-risk patients with heart failure, *Acad. Emerg. Med.* **12** (2005) 514–521.

[2] R. Bellazzi, F. Ferrazzi, and L. Sacchi, Predictive data mining in clinical medicine: a focus on selected methods and applications, *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **1** (2011) 416–430.

[3] L. Breiman, Random forests, *Mach. Learn.* **45** (2001) 5–32.

[4] M.Z.I. Chowdhury, and T.C. Turin, Precision health through prediction modelling: factors to consider before implementing a prediction model in clinical practice, *J. Prim. Health Care*. **12** (2020) 3.

[5] P. Domingos, Why does bagging work? A Bayesian account and its implications, *Proc. Third Int. Conf. Knowl. Discov. Data Min.* (1997) 2–5.

[6] X.Z. Fern, and C.E. Brodley, Boosting lazy decision trees, in: Proc. Twent. Int. Conf. Mach. Learn., Washington DC, 2003.

[7] A. Ferreira, G.F. Cooper, and S. Visweswaran, Decision path models for patient-specific modeling of patient outcomes, in: AMIA Annu. Symp. Proc., Washington, DC, 2013: pp. 413–421.

[8] M.J. Fine, R.A. Stone, D.E. Singer, C.M. Coley, T.J. Marrie, J.R. Lave, L.J. Hough, D.S. Obrosky, R. Schulz, E.M. Ricci, J.C. Rogers, and W.N. Kapoor, Processes and outcomes of care for patients with community-acquired pneumonia: results from the Pneumonia Patient Outcomes Research Team (PORT) cohort study, *Arch. Intern. Med.* **159** (1999) 970–980.

[9] J.H. Friedman, R. Kohavi, and Y. Yun, Lazy decision trees, *AAAI-96 Proc.* **1** (1996).

[10] A. Johnson, G.F. Cooper, and S. Visweswaran, Patient specific modeling with personalized decision paths, in: AMIA Annu. Symp. Proc., Virtual, 2020.

[11] J.A. Kellum, L. Kong, M.P. Fink, L.A. Weissfeld, D.M. Yealy, M.R. Pinsky, J. Fine, A. Krichevsky, R.L. Delude, and D.C. Angus, Understanding the inflammatory cytokine response in pneumonia and sepsis: results of the Genetic and Inflammatory Markers of Sepsis (GenIMS) study, *Arch. Intern. Med.* **167** (2007) 1655–1663.

[12] J. Lee, Patient-specific predictive modeling using random forests: an observational study for the critically ill, *JMIR Med. Informatics*. **5** (2017) e3.

[13] D.D. Margineantu, and T.G. Dietterich, Improved class probability estimates from decision tree models, in: Nonlinear Estim. Classif., 2002: p. pp 173-188.

[14] A. Rajkomar, J. Dean, and I. Kohane, Machine learning in medicine, *N. Engl. J. Med.* **380** (2019) 1347–58.

[15] G.A.S. Ribeiro, A.C.M. de Oliveira, A.L.S. Ferreira, S. Visweswaran, and G.F. Cooper, Patient-specific modeling of medical data, in: P. Perner (Ed.), Mach. Learn. Data Min. Pattern Recognit., Springer International Publishing, 2015: pp. 415–424.

[16] X. Robin, N. Turck, A. Hainard, N. Tiberti, F. Lisacek, J.-C. Sanchez, and M. Müller, pROC: an open-source package for R and S+ to analyze and compare ROC curves, *BMC Bioinformatics*. **12** (2011).

[17] A. Sharafoddini, J.A. Dubin, and J. Lee, Patient similarity in prediction models based on health data: a scoping review, *JMIR Med Inf.* **5** (2017) e7.

[18] S. Visweswaran, A. Ferreira, G.A. Ribeiro, A.C. Oliveira, and G.F. Cooper, Personalized modeling for prediction with decision-path models, *PLoS One*. **10** (2015) e0131022.

[19] A.K. Waljee, P.D.R. Higgins, and A.G. Singal, A primer on predictive models, *Clin. Transl. Gastroenterol.* **5** (2014) e44.

[20] D.C. Whitcomb, D. Yadav, S. Adam, R.H. Hawes, R.E. Brand, M.A. Anderson, M.E. Money, P.A. Banks, M.D. Bishop, J. Baillie, S. Sherman, J. Disario, F.R. Burton, T.B. Gardner, S.T. Amann, A. Gelrud, S.K. Lo, M.T. Demeo, W.M. Steinberg, M.L. Kochman, B. Etemad, C.E. Forsmark, B. Elinoff, J.B. Greer, M. O'connell, J. Lamb, and M. Michael Barmada, Multicenter approach to recurrent acute and chronic pancreatitis in the United States: The North American Pancreatitis Study 2 (NAPS2), *Pancreatology*. **8** (2008) 520–531.

[21] R. Xu, D. Nettleton, and D.J. Nordman, Case-specific random forests, *J. Comput. Graph. Stat.* **25** (2016) 49–65.

Address for correspondence

Adriana Johnson, adrianajohnson@pitt.edu