Geometry- and Accuracy-Preserving Random Forest Proximities

Jake S. Rhodes ©, Adele Cutler, and Kevin R. Moon ©

Abstract—Random forests are considered one of the best out-of-the-box classification and regression algorithms due to their high level of predictive performance with relatively little tuning. Pairwise proximities can be computed from a trained random forest and measure the similarity between data points relative to the supervised task. Random forest proximities have been used in many applications including the identification of variable importance, data imputation, outlier detection, and data visualization. However, existing definitions of random forest proximities do not accurately reflect the data geometry learned by the random forest. In this paper, we introduce a novel definition of random forest proximities called Random Forest-Geometry- and Accuracy-Preserving proximities (RF-GAP). We prove that the proximity-weighted sum (regression) or majority vote (classification) using RF-GAP exactly matches the out-of-bag random forest prediction, thus capturing the data geometry learned by the random forest. We empirically show that this improved geometric representation outperforms traditional random forest proximities in tasks such as data imputation and provides outlier detection and visualization results consistent with the learned data geometry.

Index Terms—Proximities, random forests, supervised learning.

I. INTRODUCTION

Random forests [1] are well-known, powerful predictors comprised of an ensemble of binary recursive decision trees. Random forests are easily adapted for both classification and regression, are trivially parallelizable, can handle mixed variable types (continuous and categorical), are unaffected by monotonic transformations, are insensitive to outliers, scale to small and large datasets, handle missing values, are capable of modeling non-linear interactions, and are robust to noise variables [1], [2]. Random forests are simple to use, produce good results with little to no tuning, and can be applied to a wide variety of fields. Citing some recent examples, Benali et al. [3] demonstrated superior solar radiation prediction using random forests over neural networks. The work of [4] showed that random forests produced the best results with the lowest standard errors in classifying error types in rotating machinery when compared with more commonly used models in this application, such as the SVM and neural networks. Other recent successes include patient health prediction after exposure to COVID-19 [5], spatio-temporal COVID-19 case estimation [6], landslide susceptibility mapping [7], infectious diarrhea forecasting [8], cardiovascular disease prediction [9], deforestation rate prediction [10], nanofluid viscosity estimation [11], rural credit assessment [12], RNA pseudouridine site prediction [13], wearable-sensor activity classification [14], heavy metal distribution estimation in agricultural soils [15], and cell type classification via Raman spectroscopy [16].

In addition to their high predictive power, random forests have a natural extension to produce pair-wise proximity (similarity) measures determined by the partitioning space of the decision trees which comprise them. The random forest proximity measure between two observations was first defined by Leo Breiman as the proportion of trees in which the observations reside in the same terminal node [17]. As splitting variable values are found to optimize partitioning according to the supervised task, these proximities encode a supervised similarity measure.

Many machine learning methods depend on some measure of pairwise similarity (which is usually unsupervised) including dimensionality reduction methods [18], [19], [20], [21], [22], [23], [24], spectral clustering [25], and any method involving the kernel trick such as SVM [26] and kernel PCA [27]. Random forest proximities can be used to extend many unsupervised problems to a supervised setting and have been used for data visualization [28], [29], [30], [31], [32], outlier detection [31], [33], [34], [35], and data imputation [36], [37], [38], [39]. See Section II for further discussion of random forest proximity applications.

Unlike unsupervised similarity measures, random forest proximities incorporate variable importance relative to the supervised task as these variables are more likely to be used in determining splits in the decision trees [2]. Ideally, random forest proximities should define a data geometry that is consistent with the learned random forest; that is, the random forest predictive ability should be recoverable from the proximities. In this case, applications involving random forest proximities, such as data visualization, can lead to improved interpretability of the random forests specifically, and more generally the data geometry relative to the supervised task.

One way to test if the proximity-defined geometry is consistent with the learned random forest is to compute a proximity-weighted predictor where a data point’s predicted label consists of a proximity-weighted sum of the labels of all other points. This

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
predictor should match the random forest’s predictions if the proximities are capturing the random forest’s learning. In particular, matching the random forest’s out-of-bag predictions is a good indication that the proximities capture the random forest’s learned data geometry in an unbiased fashion as the out-of-bag samples were not used to construct the tree. Under Breiman’s original definition, the proximity-weighed predictions do not match the random forest’s out-of-bag predictions when applied to the training data (see Section V). Thus, this definition does not capture the data geometry learned by the random forest, limiting its potential for improved interpretability of the random forest. Indeed, this inconsistency negatively affects random forest proximity applications such as data visualization and imputation.

We define a new random forest proximity measure called Random Forest-Geometry- and Accuracy-Preserving proximities (RF-GAP) that defines a data geometry such that the proximity-weighted predictions exactly match those of the random forest for both regression and classification problems. Under our definition, an out-of-bag observation’s proximities are computed via in-bag (training) observations. That is, the sample used to generate a decision tree also generates the proximities of out-of-bag observations.

RF-GAP is purposefully designed to match the random forest predictions and follows the same schema for a classification or regression learning problem. Random forests use in-bag observations or “voting points” [40] within terminal nodes to make predictions. For our definition, the proximities between out-of-bag observations are weights generated using the counts of in-bag observations sharing the same terminal node. We prove the equivalence between the proximity-weighted predictions with those of the random forest and demonstrate this empirically. In this sense, RF-GAP proximities preserve the data geometry as viewed by the random forest’s learning. We compare RF-GAP proximities with existing random forest proximities in common applications: data imputation, visualization, and outlier detection. In these applications, RF-GAP outperforms existing definitions, empirically showing clear benefits from the improved random forest geometry representation.

II. RANDOM FOREST PROXIMITY APPLICATIONS

Random forest proximities have been used in many applications such as visualization via dimensionality reduction or clustering. Unsupervised random-forest clustering was introduced by Shi and Horvath in [41]. In this paper, they applied both classical and metric multi-dimensional scaling (MDS) to a number of datasets. Pouyan et al. used unsupervised random forest proximity matrices to generate 2D plots for visualization using t-SNE [28]. This approach was used to visualize two Mass cytometry data sets and provided clearer visual results over other compared distance metrics (euclidean, Chebyshev, and Canberra). Similarly, [30] used an unsupervised random forest for clustering tumor profilings and showed improvement over other common clustering algorithms with microarray data. Random forest proximities have also been used in the supervised setting for visualizing the data via dimensionality reduction, typically via MDS [31], [42], but other approaches have been used [32]. However, these approaches use a proximity definition that does not match the data geometry learned by the random forest. Thus the visualizations do not give an accurate representation of the random forest’s learning. Random forest proximities have also been used in outlier detection in a supervised setting. An observation’s outlier score is typically defined to be inversely proportional to its average within-class proximity measure (see Section VI-C for details). This approach was used to achieve better random forest error rates in both classification and regression in pathway analysis [31]. Nesa et al. demonstrated the superiority of random forests for detecting errors and events across the Internet of Things (IoT) device sensors over other multivariate outlier detection methods [43]. The random forest outlier detection algorithm has been effective in other contexts such as modeling species distribution [44], detecting food adulteration via infrared spectroscopy [45], predicting galaxy spectral measurements [46], and detecting network anomalies [35]. Another application of random forest proximities is data imputation. This is accomplished by replacing missing values of a given variable with a proximity-weighted sum of non-missing values. Pantanowitz and Marwala used this approach to impute missing data in the context of HIV seroprevalence [38]. They compared their results with five additional imputation methods, including neural networks, and random forest-neural network hybrids, and concluded that random forest imputation produced the most accurate results with the lowest standard errors. Shah et al. compared random forest imputation with multivariate imputation by chained equations (MICE [47]) on cardiovascular health records data [39]. They showed that random forest imputation methods typically produced more accurate results and that in some circumstances MICE gave biased results under default parameterization. In [36] it was similarly shown that random forests produced the most accurate imputation in a comprehensive metabolomics imputation study. Variable importance assessment is another application of random forest proximities. One approach is to use differences in proximity measures as a criterion for assessing variable permutation importance [48]. Variable selection criteria were compared across various data sets and some improvement over existing variable importance measures was achieved. In [49], feature contributions to the random forest decision space (defined by proximities) are explored. While many measures of variable importance are generally computed at a global level, the authors propose a bitwise, permutation-based feature importance which captures both the contribution (influence of the feature in the decision space) and closeness (position in the decision space relative to the in- and out-class), giving further insight to the contribution of each feature at the terminal node level.

Multi-modal/multiview problems can also be approached using random forest proximities. Gray et al. additively combined random forest proximities from different modalities or views (FDG-PET and MR imaging) of persons with Alzheimer’s disease or with mild cognitive impairment. They applied MDS to the combined proximities to create an embedding used for classification. Classification on the multi-modal embedding showed significantly better results than classification on both modes separately [29]. Cao, Bernard, Sabourin, and Heutte explored...
variations of proximity-based classification techniques in the context of multi-view radiomics problems [50]. They compared with the work from [29] and joined proximity matrices using linear combinations. The authors explored random forest parameters to determine the quality of the proximity matrices. They concluded that a large number of maximum-depth trees produced the best quality proximities, quantified using a one-nearest neighbor classifier. Using our proposed random forest proximity measure which accurately reflects the random forest predictions from each view may add to the success of this method, thus creating a truer forest ensemble for multi-view learning.

Seoane et al. proposed the use of random forest proximities to measure gene annotations with some improvement in precision over other existing methods [51]. Zhao et al. presented two matching methods for observational studies, one propensity-based and the other random forest-proximity-based [52]. For the random forest approach, subjects of different classes were iteratively matched according to the nearest proximity values. They showed that proximity-based matching was superior to propensity matching in addition to other existing matching techniques.

All of these applications used existing definitions of random forest proximities that do not match the data geometry learned by the random forest. In contrast, RF-GAP accurately reflects this geometry. Thus, the applications presented here should show improvement using our definition. Indeed, we show experimentally in Section V that using RG-GAP gives data visualizations that more accurately represent the geometry learned from the random forests, outlier scores that are more reflective of the random forest’s learning, and improved random forest imputations.

III. RANDOM FOREST PROXIMITIES

Here we provide several existing definitions of random forest proximities followed by RF-GAP which preserves the geometry learned by the random forest. Let \( \mathcal{M} = \{ (x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N) \} \) be the training data where each \( x_i \in \mathcal{X} \), is a \( d \)-dimensional vector of predictor variables with corresponding response, \( y_i \in \mathcal{Y} \). The strength of the random forest is highly dependent on the predictive power of the individual decision trees (base learners) and on the low correlation between the decision trees [17], [53]. To decrease the correlation between the decision trees, a bootstrap sample of the training data is used to train each decision tree \( t \). Each decision tree \( t \) in a random forest is grown by recursively partitioning (splitting) the bootstrap sample into nodes, where splits are determined across a subset of feature variables to maximize purity (classification) or minimize the mean squares of the residuals (regression) in the resulting nodes. This process repeats until a stopping criterion is met. For classification, splits are typically continued until nodes are pure (one class). For regression, a common stopping criterion is a predetermined minimum node size (e.g., 5). The trees in random forests are typically not pruned.

In addition to bootstrap sampling, further correlational decrease between trees is ensured by selecting a random subset of predictor variables at each node for split optimization. The number of variables to be considered is designated by the parameter \( \text{mtry} \) in some random forest packages (e.g., randomForest [54] and ranger [55]).

We will use the following notation to define random forest proximities (see Fig. 1 for a visual example):

- \( \mathcal{T} \) is the set of decision trees in a random forest trained on \( \mathcal{M} \) with \( |\mathcal{T}| = T \).
- \( B(t) \) is the multiset of indices in the bootstrap sample of the training data that is randomly selected to train the tree \( t \in \mathcal{T} \). Thus \( B(t) \) contains the indices of the in-bag observations.
- \( O(t) = \{ i = 1, \ldots, n \, | \, i \notin B(t) \} \). Thus \( O(t) \) is the set of indices of the training data that are not contained in \( B(t) \). \( O(t) \) is often referred to as the out-of-bag (OOB) sample.
- \( S_t = \{ t \in \mathcal{T} | i \in O(t) \} \). This is the set of trees in which the \( i \)th observation is OOB.
- \( v_1(t) \) contains the indices of all observations that end up in the same terminal node as \( x_i \) in tree \( t \).
- \( J_i(t) = v_1(t) \cap B(t) \). This is the set of indices in \( v_1(t) \) that correspond with the in-bag observations of \( t \). I.e., these are

\[
S_1 = \{ x_1 \text{ is out-of-bag} \}
\]
\[
|S_1| = 3
\]
\[
T = 8
\]
\[
8 \text{ trees in the forest}
\]
\[
O(t) = \{ 1, 7, 8 \}
\]
\[
B(t) = \{ 2, 3, 3, 4, 4, 5, 6, 6 \}
\]
\[
t \in S_1
\]
\[
(\text{out-of-bag indices})
\]
\[
J_1(t) = \{ 4, 5 \}
\]
\[
M_1(t) = \{ 4, 5 \}
\]
\[
\text{(in-bag indices)}
\]
\[
\text{(obs. sharing terminal node with } x_1)\]
\[
v_1(t) = \{ 1, 4, 5 \}
\]
\[
c_4(t) = 2
\]
\[
(\text{in-bag obs. in } v_1(t))
\]

Fig. 1. An example of a random forest and notation with regards to a particular observation \( x_1 \). The red-encircled trees are those in which \( x_1 \) is out of bag, making up the set of trees \( S_1 \). A particular tree in \( S_1 \) is exhibited. The out-of-bag indices for the tree are given in red \( (i \in O(t)) \), while the in-bag indices \( (i \in B(t)) \) are shown in black. The indices of observations residing in the same terminal node as \( x_1 \) are given by the set \( v_1(t) \). \( J_1(t) \) gives the in-bag observation indices in the terminal node \( v_1(t) \), while \( M_1(t) \) provides the corresponding multiset. \( c_4(t) \) denotes the number of repetitions of observation \( x_4 \) in the tree \( t \).
the observations that are in-bag and end up in the same
terminal node as \(x_i\).

- \(M_i(t)\) is the multiset of in-bag indices in the terminal
  node shared with the \(i\)th observation in tree \(t\) including
  multiplicities.
- \(c_j(t)\) is the in-bag multiplicity of the observation \(j\) in tree
  \(t\), \(c_j(t) := 0\) if observation \(j\) is OOB. Thus, \(\sum_{j=1}^{N} c_j(t) = N\).

The terminal nodes of a random forest partition the input
space \(\mathcal{X}\). This partition is often used in defining random forest
proximities as in Breiman’s original definition:

**Definition 1 (Original Random Forest Proximity [17]).** The
random forest proximity between observations \(i\) and \(j\) is given by

\[
p_{OR}(i, j) = \frac{1}{T} \sum_{t=1}^{T} I(j \in v_i(t)),
\]

where \(T\) is the number of trees in the forest, \(v_i(t)\) contains the
indices of observations that end up in the same terminal node as
\(x_i\) in tree \(t\), and \(I(.)\) is the indicator function. That is, the
proximity between observations \(i\) and \(j\) is the proportion of trees
in which they reside in the same terminal node, regardless of
bootstrap status.

Definition 1 (the original definition) does not capture the data
geometry learned by the random forest as it does not take an
observation’s bootstrap status (whether or not the observation
was used in the training of any particular tree) into account in
the proximity calculation: both in-bag and out-of-bag samples
are used and equally weighted. In-bag observations of different
classes will necessarily terminate in different nodes (as trees are
grown until pure). Thus this produces an over-exaggerated class
separation in the proximity values.

Despite these weaknesses, this definition has been used for
outlier detection, data imputation, and visualization. However,
these applications may produce misleading results as this defini-
tion tends to overfit the training data, quantified by low error rates
as proximity-weighted predictors. One attempt to overcome this
issue redefines the proximity measure between observations \(i\)
and \(j\) using only trees in which both observations are out-of-bag
(OOB proximities):

**Definition 2 (OOB Proximity [54], [56]).** The OOB proxim-
ity between observations with indices \(i\) and \(j\) is given by

\[
p_{OOB}(i, j) = \frac{\sum_{t \in S_j} I(j \in O(t) \cap v_i(t))}{\sum_{t \in S_i} I(j \in O(t))},
\]

where \(O(t)\) denotes the set of indices of the observations that
are out-of-bag in tree \(t\), and \(S_j\) is the set of trees for which observ-
ation \(j\) is out-of-bag. In other words, this proximity measures
the proportion of trees in which observations \(i\) and \(j\) reside in
the same terminal node, both being out-of-bag.

Definition 2 [56] is currently used in the randomForest
package by Liaw and Wiener in the R programming
language [57]. It may have been inadvertently used in papers
that used this package but, none made explicit mention of the
use of OOB observations in building proximities. However, we
find that this definition also does not characterize the random
forest predictions and generally produces higher error rates as
a proximity-weighted predictor when compared to the random
forest’s OOB error rate (see Section V). The reason that the
results do not match is two-fold. First, this definition does not
directly use in-bag observations. A classification or regression
forest trains on a training set (in-bag set) and predicts on an
unseen test set (OOB observations). Thus the OOB proximities
are missing a key factor in the random forest behavior. Second,
random forest predictions are made by voting or averaging
in-bag labels. In contrast, the OOB proximities are generated
without regard to the number of observations within a given
terminal node.

Several alternative random forest proximity measures be-
ond Definitions 1 and 2 have been proposed. In each case,
source code has not been provided. In [58], the authors define a
proximity-based kernel (PBK) that accounts for the number of
branches between observations in each decision tree, defining
the proximity between \(i\) and \(j\) as
\[p_{PBK}(i, j) = \frac{1}{T} \sum_{t=1}^{T} \frac{1}{1 + g_{ijt}},\]
where \(T\) is the number of trees in the forest, \(w\) is a user-defined
parameter, and \(g_{ijt}\) is the number of branches between observ-
ations \(i\) and \(j\) in tree \(t\). \(g\) is defined to be 0 if the observations
reside in the same terminal node. The proximity quality was
quantified using the classification accuracy when applied as
a kernel in a support-vector machine. This definition showed
some improvement over the original definition (Definition 1)
only when considering very small numbers of trees (5 or 10).
It should be noted that this kernel is an attempt to improve
random-forest-based similarity measures and not necessarily to
explain the random forest. However, PBK is computationally
expensive as all pair-wise branch distances must be computed
within each tree. This is not an issue for small numbers of trees,
but typically the number of trees in a random forest is measured
in hundreds (randomForest [54] has a default of 500 trees).
Additionally, this method adds a user-defined, tunable parameter
which adds to its complexity.

The authors in [59] describe an approach for computing ran-
dom forest proximities in the context of a larger class of Random
Partitioning Kernels. While most random forest proximities
are determined primarily through associations within terminal
nodes, this approach selects a random tree height and partitions
the data based on this higher-level splitting. The authors do
not compare with other proximity definitions nor do they frame
their work in the context of random forest proximities, but they
do compare this random forest kernel to other typical kernels
(linear, RBF, etc.) using a log-likelihood test. The random forest
kernel outperformed the others in most cases and the authors
visually demonstrated their kernel using 2D PCA plots. The
code for this approach is not publicly available.

Cao et al. introduced two random forest dissimilarity mea-
sures that were used in the context of multi-view classifica-
tion [60]. The first measure (denoted RFDisNC) weights the
proximity values by the proportion of correctly-classified obser-
vations within each node, accounting for both in- and out-of-bag
observations. The second (RFDisIH) is based on instance hard-
ness. Euclidean distances between observations at each terminal
node are calculated using only feature variables that were used
as splitting variables leading to the terminal node to avoid the
curse of dimensionality. These distances are then used as weights as a part of the dissimilarity measure. Given this distance, they use \( k \)-Disagreeing Neighbors (DN) in the formulation of the dissimilarity measure

\[
d_t(x_i, x_j) = \begin{cases} 
    kDN(x_j), & \text{if } v_i(t) = v_j(t) \\
    1, & \text{otherwise}
\end{cases},
\]

where

\[
kDN(x_i) = \left| \{x_j : x_j \in kNN(x_i), y_j \neq y_i \} \right|,
\]

where \( kNN(x_i) \) is the set of the \( k \)-nearest neighbors of \( x_i \) in the training data.

The use of \( k \)-DN gives a notion of difficulty in classifying a particular observation. In the multiview problem, the dissimilarities from different views are averaged before classification. The authors showed that RFDisIH performed better overall on classification tasks compared with other multi-view methods. However, RFDisIH was not compared with other random forest proximities in their commonly-used applications (e.g., visualization or imputation). A similarity measure can be constructed from RFDisIH as \( RFPXoI = 1 - \text{RFDisIH} \). RFProxIH (adapted from RFDisIH) is also not specifically intended to explain the random forest’s learning but uses the partitioning space to attempt to improve a forest-derived similarity measure.

While these alternative definitions have shown promise in their respective applications, their connection to the data geometry learned by the random forest is not clear. In contrast, we present a new definition of random forest proximities that exactly characterizes the random forest performance. In short, our new proximities provide a similarity measure that keeps the integrity of the forest.

**Definition 3 (Random Forest-Geometry- and Accuracy-Preserving Proximities (RF-GAP)).** Let \( B(t) \) be the multiset of (potentially repeated) indices of bootstrap (in-bag) observations. We define \( J_t(i) \) to be the set of in-bag observations which share the terminal node with observation \( i \) in tree \( t \), or \( J_t(i) = B(t) \cap v_i(t) \), and \( J_t(i) \subseteq M_t(i) \), which is the multiset including in-bag repetitions. Let \( c_j(t) \) be the multiplicity of the index \( j \) in the bootstrap sample. Then, for given observations \( i \) and \( j \), their proximity measure is defined as

\[
p_{\text{GAP}}(i, j) = \frac{1}{|S_i|} \sum_{t \in S_i} c_j(t) \cdot \frac{I(j \in J_t(i))}{|M_t(i)|}.
\]

That is, the proximity between an observation \( i \), and an observation \( j \) is constructed using a weighted average of the counts of \( j \) with the reciprocal of the number of in-bag observations in each shared terminal node serving as weights.

This proposed definition is, in part, inspired by the work of Lin and Jeon [40], who showed that random forests behave as nearest-neighbor predictors with an adaptive bandwidth. To see this, first consider that for kernel methods, such as the SVM, the prediction for the \( i \)th observation can be written as

\[
\hat{y}_i(p) = \sum_{j=1}^n p(i, j) y_j,
\]

where \( p(i, j) \) is some (potentially weighted) kernel function. A kernel function induces an inner product, and thus defines a similarity or proximity measure. Hence, Lin and Jeon proposed to use a proximity measure derived from the random forest for \( p \) as their kernel.

The three main proximity measures defined here (\( p_{\text{POO}} \), \( p_{\text{OOOB}} \), and \( p_{\text{GAP}} \)) can be obtained by weighting the in- and out-of-bag observations differently in the prediction of \( y_i \). \( p_{\text{POO}} \) equally weights in- and out-of-bag examples. Because the nodes in each tree of a random forest are typically grown to purity, the proximity measure \( p_{\text{POO}} \) overemphasizes class separation compared to the actual random forest when using the proximity weighted prediction \( \hat{y}_{p_{\text{POO}}} \). In this sense, this proximity-weighted prediction is a biased estimate of the random forest prediction function.

On the other hand, \( p_{\text{OOOB}} \) doesn’t use any of the in-bag observations, which are required to produce the random forest predictions. Thus the OOB proximities are also biased in the sense that the proximity-weighted prediction does not match the random forest prediction function. This is true even for large sample sizes (see Figs. A.1 and A.2, available in the online supplemental material).

In contrast, \( p_{\text{GAP}} \) appropriately weights both in-bag and OOB samples such that the proximity-weighted prediction matches the random forest OOB predictions for all sample sizes. Thus using \( p_{\text{GAP}} \) in the proximity-weighted prediction gives an unbiased estimate of the random forest prediction function.

OOB samples are commonly used to provide an unbiased estimate of the forest’s generalization error rate [1]. Thus, we use the forest’s OOB predictions to assess the quality of our proximities. We show in Section IV that the random forest OOB prediction (and thus the generalization error rate) is reproducible as a weighted sum (for regression) or a weighted-majority vote (for classification) using the proximities in Definition 3 as weights. Additionally, RF-GAP proximity-weighted predictions of test points match those of the random forest. Thus, this definition characterizes the random forest’s predictions, keeping intact the learned data geometry. Subsequently, applications using this proximity definition will provide results that are truer to the random forest from which the proximities are derived.

We note, however, that the RF-GAP proximities do not form a proper similarity measure as they are not symmetric. This is easily addressed in applications requiring symmetry by averaging \( p_{\text{GAP}}(i, j) \) and \( p_{\text{GAP}}(j, i) \). This was done in our experiments involving multi-dimensional scaling. In Fig. A.3, available in the online supplemental material, we also demonstrate that the RF-GAP proximities become more symmetric as the number of trees increases.

**IV. RANDOM FORESTS AS PROXIMITY-WEIGHTED PREDICTORS**

Here we show that the random forest OOB and test predictions are reproducible using RF-GAP proximities as weights in a weighted sum (for regression) or a weighted-majority vote (for classification). We first show that for a given observation, the proximities are non-negative and sum to one.
Note that for each observation, \( i \), the RF-GAP definition assigns the value \( p_{\text{GAP}}(i, i) = 0 \). This is important when using the proximities as a predictor as an observation should not be used in the prediction of its own label. However, as a similarity measure, setting \( p_{\text{GAP}}(i, i) = 0 \) is not meaningful. Thus, for other proximity-based applications, we construct the main diagonal entries of the RF-GAP proximity matrix by passing through the random forest a duplicate, identical observation assigned out-of-bag status.

**Proposition 1.** Defining \( p_{\text{GAP}}(i, i) = 0 \), the RF-GAP proximities are non-negative and \( \sum_{j=1}^{N} p_{\text{GAP}}(i, j) = 1 \).

**Proof.** It is clear from the definition that \( p_{\text{GAP}}(i, j) \geq 0 \) for all \( i, j \). The sum-to-one property falls directly from the definition

\[
\sum_{j=1}^{N} p_{\text{GAP}}(i, j) = \sum_{j=1}^{N} \frac{1}{|S_i|} \sum_{t \in S_i} c_j(t) \cdot I(j \in J_i(t)) \frac{1}{|M_i(t)|} = \frac{1}{|S_i|} \sum_{t \in S_i} \frac{1}{|M_i(t)|} \sum_{j=1}^{N} c_j(t) \cdot I(j \in J_i(t)) = \frac{1}{|S_i|} \sum_{t \in S_i} 1 = 1.
\]

This proposition allows us to directly use RF-GAP as weights for classification or regression. In contrast, other proximities must be row-normalized to serve as weights. We show that the RF-GAP proximity-weighted prediction matches the random forest OOB prediction, giving the same OOB error rate in both the classification and regression settings. The OOB error rate is typically used to estimate the forest’s generalization error and quantify its goodness of fit, this indicates that RF-GAP accurately represents the geometry learned by the random forest.

**Theorem 1 (Proximity-Weighted Regression).** For a given training data set \( S = \{(x_1, y_1), \ldots, (x_N, y_N)\} \), with \( y_i \in \{1, \ldots, K\} \) for all \( i \in \{1, \ldots, N\} \), the random forest OOB regression prediction can be reconstructed by a weighted-majority vote using RF-GAP proximities (Definition 3) as weights.

**Proof.** For a given tree, \( t \), and \( i \in O(t) \), the decision tree predictor of \( y_i \) is the mean response of the in-bag observations in the appropriate terminal node. That is

\[
\hat{y}_i(t) = \frac{1}{|M_i(t)|} \sum_{j \in J_i(t)} c_j(t) \cdot y_j
\]

The random forest prediction, \( \hat{y}_i \), is the mean response over all trees for which \( i \) is out of bag. That is

\[
\hat{y}_i = \frac{1}{|S_i|} \sum_{t \in S_i} \hat{y}_i(t) = \frac{1}{|S_i|} \sum_{t \in S_i} \frac{1}{|M_i(t)|} \sum_{j \in J_i(t)} c_j(t) \cdot y_j.
\]

The proximity-weighted predictor, \( \hat{y}_i^P \), is simply the weighted sum of responses, \( \{y_j\}_{j \neq i} \)

\[
\hat{y}_i^P = \sum_{j=1}^{N} p_{\text{GAP}}(i, j) y_j = \sum_{j=1}^{N} \left\{ \frac{1}{|S_i|} \sum_{t \in S_i} c_j(t) \cdot I(j \in J_i(t)) \right\} y_j = \sum_{j=1}^{N} \frac{1}{|S_i|} \sum_{t \in S_i} \frac{1}{|M_i(t)|} \sum_{j \in J_i(t)} c_j(t) \cdot I(j \in J_i(t)) y_j = \sum_{j=1}^{N} \frac{1}{|S_i|} \sum_{t \in S_i} \frac{1}{|M_i(t)|} \sum_{j \in J_i(t)} c_j(t) \cdot y_j = \hat{y}_i.
\]

**Theorem 2 (Proximity-Weighted Classification).** For a given training data set \( S = \{(x_1, y_1), \ldots, (x_N, y_N)\} \), with \( y_i \in \{1, \ldots, K\} \) for all \( i \in \{1, \ldots, N\} \), the random forest OOB classification prediction can be reconstructed by a weighted-majority vote using RF-GAP proximities (Definition 3) as weights.

**Proof.** Given the training set \( \{(x_1, y_1), \ldots, (x_N, y_N)\} \), with \( y_i \in \{1, \ldots, K\} \), for a given tree \( t \) and observation \( i \in O(t) \), the decision tree prediction of the label \( y_i \) is determined by the majority vote among in-bag observations within the shared terminal node

\[
\hat{y}_i(t) = \arg \max_{l=1, \ldots, K} \sum_{j \in J_i(t)} c_j(t) \cdot I(y_j = l).
\]

Thus, the random forest classification prediction, \( \hat{y}_i \), is the most popular predicted class across all \( t \in S_i \)

\[
\hat{y}_i = \arg \max_{k=1, \ldots, K} \sum_{t \in S_i} I(\hat{y}_i(t) = k) = \arg \max_{k=1, \ldots, K} \sum_{t \in S_i} I \left( \arg \max_{l=1, \ldots, K} \sum_{j \in J_i(t)} c_j(t) \cdot I(y_j = l) = k \right)
\]

We show equivalence with the proximity-weighted predictor. The proximity-weighted predictor predicts the class with the largest proximity-weighted vote:

\[
\hat{y}_i^P = \arg \max_{k=1, \ldots, K} \sum_{j=1}^{N} p_{\text{GAP}}(i, j) I(y_j = k) = \arg \max_{k=1, \ldots, K} \left\{ \sum_{j=1}^{N} \left( \frac{1}{|S_i|} \sum_{t \in S_i} c_j(t) \cdot I(j \in J_i(t)) \right) I(y_j = k) \right\}
\]
\[
\begin{align*}
\hat{y}_i(t) &= \arg\max_{l \in \{1, \ldots, K\}} \sum_{t \in S_i} \frac{1}{|S_i|} \sum_{j \in J_i(t)} c_j(t) \cdot I(y_j = l) \\
&= y_{i,t},
\end{align*}
\]

and the random forest predictor is
\[
\hat{y}_i = \arg\max_{k \in \{1, \ldots, K\}} \sum_{t \in S_i} I(y_{i,t} = k).
\]

The proximity-weighted predictor is thus
\[
\hat{y}^p_i = \arg\max_{k \in \{1, \ldots, K\}} \left\{ \sum_{t \in S_i} \frac{1}{|S_i|} \sum_{j \in J_i(t)} c_j(t) \cdot I(y_j = k) \right\}
\]
\[
= \arg\max_{k \in \{1, \ldots, K\}} \left\{ \sum_{t \in S_i} \frac{1}{|S_i|} \sum_{j \in J_i(t)} c_j(t) \cdot I(y_{i,t} = k) \right\}
\]
\[
= \arg\max_{k \in \{1, \ldots, K\}} \left\{ \sum_{t \in S_i} \frac{1}{|S_i|} |M_i(t)| I(y_{i,t} = k) \right\}
\]
\[
= \arg\max_{k \in \{1, \ldots, K\}} \sum_{t \in S_i} I(y_{i,t} = k)
\]
\[
= \hat{y}_i.
\]

We note that the proof for the proximity-weighted classification problem requires that the trees in the forest are grown until the terminal nodes are pure. We show empirically, however, that RF-GAP proximity predictions more closely match the random forest’s predictions than the other proximities even when trees are pruned. See Fig. 2 for a comparison. Here we compared proximity-weighted predictions using minimum node sizes of 1, 5, 10, 20, and 50. For regression problems, RF-GAP proximity predictions match those of the random forest perfectly, while this isn’t always the case for classification. The difference increases with node size as ties become more likely. We also note that the random forest and proximity-weighted predictions produce higher error rates with the increase in minimum node size (Fig. A.4, available in the online supplemental material).

V. EXPERIMENTAL VALIDATION OF PROXIMITY-WEIGHED PREDICTION

To demonstrate that the RF-GAP proximities preserve the random-forest learned data geometry, we empirically validate Theorems 1 and 2 from Section IV, demonstrating that the random forest predictions are preserved in the proximity construction. We also compare to the proximity-weighted predictor...
using the original random forest proximity definition (Definition 1), the OOB adaptation (Definition 2), PBK [58], and RFProxIH [60].

We compared proximity-prediction results on 24 datasets as described in Table B.1, available in the online supplemental material. Each dataset was randomly partitioned into training (70%) and test (30%) sets. For each dataset, the same trained random forest was used to produce all compared proximities. Tables A.1 and A.2, available in the online supplemental material, show the differences between the random forest predictions and the proximity-weighted predictions for test and training sets, respectively. The proximity-weighted error rates using RF-GAP proximities nearly exactly match the random forest OOB error rates; discrepancies are due to random tie-breaking. In contrast, the original definition, PBK, and RFProxIH typically produce much lower training error rates, suggesting overfitting. The OOB proximities (Definition 2) produce training error rates that are sometimes lower and sometimes higher than the random forest OOB error rate.

The RF-GAP proximities also generally produce the lowest test errors. This can be seen in Fig. 3, which plots the training versus test error rates using the different proximity measures. Table I gives the regression slope for each proximity definition. From here it is clear that the original proximities, PBK, and RFProxIH overfit the training data. This is corroborated in Fig. 4, which plots the difference between the random forest out-of-bag error rates and the proximity-weighted errors across the same datasets, demonstrating that the RF-GAP predictions nearly perfectly match those of the random forest for both training and tests sets. It is clear that the original definition, PBK, and RFProxIH generally overfit the training data in contrast. We re-emphasize that PBK and RFProxIH were not intended to demonstrate the forest’s learning but to use the forest as a means to construct a similarity measure.

It is well-established that increasing the number of trees increases the random forest generalization error rate for standard measures of error [1], [61], while the typical random forest default of 500 trees is generally sufficient [54], [55]. We compared proximity-based predictions for smaller numbers of trees (5, 10, 50, 100, 250) and demonstrated that RF-GAP predictions still most accurately match those of the random forest regardless of forest size. However, for classification problems, tie breaks are more often required with a small number of trees. Results can be found in Fig. A.5, available in the online supplemental material.

Table I

| Type        | Slope  |
|-------------|--------|
| RF-GAP      | 1.040  |
| Original    | 2.434  |
| OOB         | 1.543  |
| PBK         | 2.287  |
| RFProxIH    | 2.147  |

In addition to comparing proximity-weighted predictions to the random forest predictions, we compared predictions across the proximity definitions. Generally, RF-GAP predictions differed the most from the other proximity-weighted predictions, especially when using the training set. This is not surprising as the RF-GAP proximities are defined to match the random forest OOB predictions. See more details in Fig. A.6, available in the online supplemental material.

VI. COMPARISON OF COMMON PROXIMITY-BASED APPLICATIONS

In this section, we demonstrate that using RF-GAP in multiple applications leads to improved performance relative to the other random forest proximity measures. We compare scatterplots generated by applying MDS to the proximities for visualization in Section VI-A, data imputation in Section VI-B, and outlier detection in Section VI-C. Random forest proximities have already been established as successful in these applications (see Section II). Thus, we focus our comparisons in these applications on existing random forest proximity definitions to show that the improved representation of the random forest geometry in RF-GAP leads to improved performance. In each comparative experiment, we train the random forests using default hyperparameters. Each set of proximities is constructed using the same forest to ensure differences are attributed entirely to the proximity definition and not to the inherent randomness of the forest. Additional results beyond those presented here are given in Appendix A, available in the online supplemental material.
Fig. 4. These boxplots show the proportion of proximity-weighted OOB and test predictions which do not match the respective random forest predictions. Proximity-based training predictions were compared with the RF out-of-bag errors. RF-GAP proximity predictions most nearly match the random forest predictions for both the training (top) and test (bottom) data, thus, best preserving the geometry learned by the random forest. Datasets were split into 70% training and 30% test data. Errors were averaged across 5 runs. The dataset details can be found in Table B.1, available in the online supplemental material.

A. Visualization Using Multi-Dimensional Scaling

Leo Breiman [1], [17] first introduced the use of multi-dimensional scaling on random forest proximities to visualize the forest’s learning. Since then, MDS with random forest proximities has been used in many applications including tumor profiling [30], visualizing biomarkers [42], pathway analysis [31], multi-view learning [29], [50], and unsupervised learning [41]. Here, we compare visualizations using the five different proximities. In each case, metric MDS was applied to $\sqrt{1 - \text{prox}}$ to produce two-dimensional visualizations.

B. Imputation

In [36], experimental results showed that, out of nine compared imputation methods across seven different missing mechanisms (missing at random, missing completely at random, etc.), random forest-based imputation was generally the most accurate. This has been corroborated by [37], [38], [39]. Here we describe the algorithm for random forest imputation and compare results using various proximity definitions.

To impute missing values of variable $j$:

1) If variable $j$ is continuous, initialize the imputation with the median of the in-class values of variable $j$; otherwise, initialize it with the most frequent in-class value. In the regression context, the median or most frequent values are computed across all observations without missing values in variable $j$.

2) Train a random forest using the imputed dataset and

3) construct the proximity matrix from the forest.

4) If variable $j$ is continuous, replace the missing values with the proximity-weighted sum of the non-missing values. If categorical, replace the missing values with the proximity-weighted majority vote.

5) Repeat steps 2 - 4 as required. In many cases, a single iteration is sufficient.

We show empirically that random forest imputation is generally improved using the RF-GAP proximity definition. For our experiment, we selected datasets that are described in Table B.1, available in the online supplemental material, and for each dataset, we removed 5%, 10%, 25%, 50%, and 75% of values at random (that is, the values are missing completely at random, or MCAR), using the missMethods [63] R package. Two comparisons were made: 1) we computed the mean MSE across 100 repetitions using a single iteration, and 2) we computed the mean MSE (across 5 repetitions) at each of 10 iterations. An example using 10 iterations is given in Fig. 6.

A summary of performance rankings is given in Table II. Across all compared proximity definitions (RF-GAP, OOB, original, and RFProxIH), RF-GAP achieved the best imputation rankings across all percentages of missing values. For full results, see Table A.3, available in the online supplemental material, which gives the average MSE across 100 repetitions.
Fig. 5. Here is a comparison of MDS embeddings using different RF proximity definitions. Proximities were constructed from a random forest trained on the two-class Sonar dataset (208 observations of 60 variables) from the UCI repository which gave an OOB error rate of 14.9%. Multi-dimensional scaling (MDS) was applied to $\sqrt{1 - \text{prox}}$ using RF-GAP proximities, the original proximities, OOB proximities (Definition 2), PBK [58], and RFProxIH [60]. Using RF-GAP proximities, the visualization depicts a good representation of the forest’s classification problem. For correctly-classified points (dots), there are two clear groupings, while misclassified points (squares) are generally located between the groupings or found within the opposite class cluster, albeit closer to the decision boundary than not. The original definition, PBK, and RFProxIH over-exaggerate the separation between classes. This is apparent in examples in the figure as the two classes appear nearly linearly separable which does not accurately depict the random forest’s performance on the dataset. Using only OOB samples to generate the proximities improves upon those three but seems to add some noise to the visualization. There are still two major class clusters, but some correctly classified points are found farther inside the opposite class’ cluster compared to the RF-GAP visualization. More examples can be found in Appendix A.7, available in the online supplemental material.

Fig. 6. Imputation of values MCAR of the optical digits dataset [62]. The experiment was repeated over 5 runs and across 10 iterations. RF-GAP proximities produced the most accurate imputations.

| Group       | M, Correct | M, Incorrect | R, Correct | R, Incorrect |
|-------------|------------|--------------|------------|--------------|
| Original    |            |              |            |              |
| OOB         |            |              |            |              |
| PBK         |            |              |            |              |
| RF-GAP      |            |              |            |              |
| RFProxIH    |            |              |            |              |

TABLE II

| THE AVERAGE RANKS OF THE IMPUTATION SCORES ACROSS THE VARIOUS UCI DATASETS AND VARIOUS PERCENTAGES OF MISSING VALUES. EACH IMPUTATION EXPERIMENT WAS REPEATED 100 TIMES WITH DIFFERENT RANDOM INITIALIZATION. RF-GAP GENERALLY PRODUCES THE BEST IMPUTATIONS ACROSS ALL PERCENTAGES OF MISSING VALUES. SEE TABLE A.3, AVAILABLE IN THE ONLINE SUPPLEMENTAL MATERIAL, FOR THE FULL IMPUTATION RESULTS |
|-------------|------------|--------------|------------|--------------|
|             | 5%         | 10%          | 25%        | 50%          | 75%          |
| RF-GAP      | 1.10       | 1.20         | 1.15       | 1.35         | 1.65         |
| Original    | 2.55       | 2.70         | 2.65       | 2.60         | 2.35         |
| OOB         | 2.50       | 2.55         | 2.45       | 2.40         | 2.30         |
| RFProxIH    | 2.20       | 3.33         | 3.33       | 3.40         | 3.20         |

using a single iteration of the above-described algorithm. The number of observations and variables for each dataset are also provided in the table.

Figs. A.13, A.14, and A.15, available in the online supplemental material, compare imputation results across 17 datasets.

C. Outlier Detection

Random forests can be used to detect outliers in the data. In the classification setting, outliers may be described as observations with measurements that significantly differ from those of other observations within the same class. In some cases, these outliers may be similar to observations in a different class, or perhaps they may distinguish themselves from observations in all known large datasets were not used as RFProxIH is computationally inefficient. Also, datasets already missing values were not used.
Fig. 7. MDS applied to the random forest proximities computed from the Gene Expression Cancer dataset from UCI [62]. The point sizes are inversely proportional to the average proximity of a given observation to all other within-class observations. The misclassifications in RF-GAP, and OOB, are understandable based on the distance from their respective clusters. The original proximities and RFProxIH do not clearly account for the misclassified points. The outlier measure scaling in RF-GAP gives a clear reflection of the distance of points to their respective clusters.

Fig. 8. A sorted plot of the outlier measures for the MNIST dataset as provided by RF-GAP. The vertical axis is the outlier measure as described in Section VI-C. The top seven outlying images are labeled with an index and shown in Fig. 9.

Random forest proximity measures can be used to detect within-class outliers as outliers are likely to have small proximity measures with other observations within the same class. Thus, small average within-class proximity values may be used as an outlier measure. We describe the algorithm as follows:

1) For each observation, \(i\), compute the raw outlier measure score as \(\sum_{j \in \text{class}(i)} n_{\text{prox}}(i,j)\).

2) Within each class, determine the median and mean absolute deviation of the raw scores.

3) Subtract the median from the raw score and divide the result by the mean absolute deviation.

The outlier detection measure may be used in conjunction with MDS for visualization. See Fig. 7 for an example using the Gene Expression Cancer dataset from UCI [62] which has 5 classes across 801 observations and 20531 variables. Here, the point sizes of the scatter plot are proportionally scaled to the outlier measure. From the figure, it is clear that points outside of their respective class clusters have higher outlier measures. That the outlier measure is inversely proportional to the average proximity to within-class observations is clear in the case of RF-GAP proximities and is not very clear in the cases of the original definition and RFProxIH. This suggests that RF-GAP can be used to improve random forest outlier detection.

Figs. 8 shows the RF-GAP outlier scores for the MNIST dataset. From this plot, outliers may be visually identified as the datapoints with substantially higher outlier scores compared to their class. Fig. 9 shows some of the outlying images, some of which may be difficult for humans to distinguish. Appendix A.7, available in the online supplemental material, contains further experiments.

VII. CONCLUSION

In this paper, we presented a new definition of random forest proximities called RF-GAP that characterizes the random forest
out-of-bag prediction results using a weighted nearest neighbor predictor. We proved that the performance of the proximity-weighted predictor exactly matches the out-of-bag prediction results of the trained random forest and also demonstrated this relationship empirically. Thus, RF-GAP proximities capture the random forest-learned data geometry which provides empirical improvements over other existing definitions in applications such as proximity-weighted prediction, missing data imputation, outlier detection, and visualization.

Additional random forest proximity applications can be explored in future works, including quantifying outlier detection performance, comparing with non-tree-based methods, assessing variable importance, and applying RF-GAP to multi-view learning. This latter application shows promise as classification accuracy was greatly increased after combining proximities in [29], [50] using other definitions. An interesting visualization application was introduced in [32] which may see improvements using geometry-preserving proximities. Multi-view learning may also be paired with this approach in some domains to visualize and assess contributions from the various modes or to perform manifold alignment.

An additional area of improvement is scalability. Our approach is useful for datasets with a few thousand observations, but we can expand its capabilities by implementing a sparse version of RF-GAP. Further adaptations may make random forest proximity applications accessible for even larger datasets.

REFERENCES

[1] L. Breiman, “Random forests,” *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001.

[2] A. Cutler, D. R. Cutler, and J. R. Stevens, *Random Forests*. Boston, MA, USA: Springer US, 2012, pp. 157–175.

[3] L. Benali, G. Notton, A. Fouilloly, C. Voyant, and R. Dizene, “Solar radiation forecasting using artificial neural network and random forest methods: Application to normal beam, horizontal diffuse and global components,” *Renewable Energy*, vol. 132, pp. 871–884, 2019.

[4] T. Han, D. Jiang, Q. Zhao, L. Wang, and K. Xia, “Comparison of random forest, artificial neural networks and support vector machine for intelligent diagnosis of rotating machinery,” *Trans. Inst. Meas. Control*, vol. 40, pp. 2681–2693, 2018.

[5] C. Iwendi et al., “Covid-19 patient health prediction using boosted random forest algorithm,” *Front. Public Health*, vol. 8, 2020, Art. no. 357.

[6] C. M. Yeşilkınatan, “Spatial-temporal estimation of the daily cases of COVID-19 in worldwide using random forest machine learning algorithm,” *Chaos, Solitons Fractals*, vol. 140, 2020, Art. no. 110210.

[7] L. Yang et al., “Study of cardiovascular disease prediction model based on random forest in eastern China,” *Sci. Rep.*, vol. 10, no. 1, Mar. 2020, Art. no. 222.

[8] L. Yang et al., “Forecasting the incidence of infectious diarrhea using random forest in jiangsu province, China,” *BMC Infect. Dis.*, vol. 20, no. 1, Mar. 2020, Art. no. 421.

[9] L. Yang et al., “Study of cardiovascular disease prediction model based on random forest in eastern China,” *Sci. Rep.*, vol. 10, no. 1, Mar. 2020, Art. no. 5245.

[10] T. Han, D. Jiang, Q. Zhao, L. Wang, and K. Xia, “Comparison of random forest, artificial neural networks and support vector machine for intelligent diagnosis of rotating machinery,” *Trans. Inst. Meas. Control*, vol. 40, pp. 2681–2693, 2018.

[11] C. Iwendi et al., “Covid-19 patient health prediction using boosted random forest algorithm,” *Front. Public Health*, vol. 8, 2020, Art. no. 357.

[12] C. M. Yeşilkınatan, “Spatial-temporal estimation of the daily cases of COVID-19 in worldwide using random forest machine learning algorithm,” *Chaos, Solitons Fractals*, vol. 140, 2020, Art. no. 110210.

[13] Z. Lv, J. Zhang, H. Ding, and Q. Zou, “RF-PseU: A random forest predictor for RNA pseudouridine sites,” *Front. Bioeng. Biotechnol.*, vol. 8, pp. 134–134, Feb. 2020.

[14] S. Badar ud din, A. Tahir, and M. B. Bano, “Wearable sensors for activity analysis using SMO-based random forest over smart home and sports datasets,” in *Proc. IEEE 3rd Int. Conf. Adv. Comput. Sci.*, 2020, pp. 1–6.

[15] K. Tan, H. Wang, L. Chen, Q. Du, D. Fu, and C. Pan, “Estimation of the spatial distribution of heavy metal in agricultural soils using airborne hyperspectral imaging and random forest,” *J. Hazardous Mater.*, vol. 382, 2020, Art. no. 120987.

[16] W. Zhang et al., “Label-free discrimination and quantitative analysis of oxidative stress induced cytotoxicity and potential protection of antioxidants using raman micro-spectroscopy and machine learning,” *Anal. Chim. Acta*, vol. 1128, pp. 221–230, 2020. [Online]. Available: https://doi.org/10.1016/j.aca.2020.06.074.

[17] L. Breiman and A. Cutler, “Random forests,” Accessed: Oct. 1, 2021. [Online]. Available: https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm

[18] J. Kruskal and M. Wish, *Multidimensional Scaling*. Newbury Park, CA, USA: Sage Publications, 1978.

[19] L. van der Maaten and G. Hinton, “Visualizing data using t-SNE,” *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, 2008.

[20] K. R. Moon et al., “Visualizing structure and transitions in high-dimensional biological data,” *Nature Biotechnol.*, vol. 37, no. 12, pp. 1482–1492, Dec. 2019.

[21] A. F. Duque, G. Wolf, and K. R. Moon, “Visualizing high dimensional dynamical processes,” in *Proc. IEEE 29th Int. Workshop Mach. Learn. Signal Process.*, 2019, pp. 1–5.

[22] A. F. Duque, S. Morin, G. Wolf, and K. Moon, “Extendable and invertible manifold learning with geometry regularized autoencoders,” in *Proc. IEEE Int. Conf. Big Data*, 2020, pp. 5027–5036.

[23] J. B. Tenenbaum, V. Silva, and J. C. Langford, “A global geometric framework for nonlinear dimensionality reduction,” *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[24] L. McNees, J. Healy, and I. Melville, “UMAP: Uniform manifold approximation and projection for dimension reduction,” 2018, arXiv:1802.03426.

[25] A. Ng, M. Jordan, and Y. Weiss, “On spectral clustering: Analysis and an algorithm,” *Adv. Neural. Inf. Process. Syst.*, vol. 14, pp. 849–856, 2001.

[26] C. Cortes and V. Vapnik, “Support-vector networks,” *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995.

[27] B. Schölkopf, A. Smola, and K. Müller, “Kernel principal component analysis,” in *Advances in Kernel Methods - Support Vector Learning*. Cambridge, MA, USA: MIT Press, 1999, pp. 327–352.

[28] M. B. Pouyan, J. Birjandtalab, and M. Nourani, “Distance metric learning using random forest for cytotmetry data,” in *Proc. IEEE 38th Ann. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2016, pp. 2590–2590.

[29] K. R. Gray, P. Aljabar, R. A. Heckemann, A. Hammers, and D. Rueckert, “Random forest-based manifold learning for classification of imaging data in dementia,” in *Proc. Mach. Learn. Med. Imag.*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 159–166.

[30] T. Shi, D. Seligson, A. S. Belldegrun, A. Palote, and S. Horvath, “Tumor classification by tissue microarray profiling: Random forest clustering applied to renal cell carcinoma,” *Modern Pathol.*, vol. 18, no. 4, pp. 547–557, Apr. 2005.

[31] H. Pang et al., “Pathway analysis using random forests classification and regression,” *Bioinformatics*, vol. 22, no. 16, pp. 2028–2036, 06 2006.

[32] J. S. Rhodes, A. Cutler, G. Wolf, and K. R. Moon, “Random forest-based diffusion information geometry for supervised visualization and data exploration,” in *Proc. IEEE Statist. Signal Process. Workshop, 2021*, pp. 331–335.

[33] J. Zhang, M. Zulkernine, and A. Haque, “Random-forests-based network intrusion detection systems,” *IEEE Trans. Syst. Man Cybern., Part C (AppL Rev.),* vol. 38, no. 5, pp. 649–659, Sep. 2008.

[34] P. O. Gislason, J. A. Benediktsson, and J. R. Sveinsson, “Random forests for land cover classification,” *Pattern Recognit. Lett.*, vol. 27, no. 4, pp. 294–300, 2006.

[35] V. L. Narayana, A. P. Gopi, S. R. Khadherphi, and V. Pavanvi, “Accurate identification and detection of outliers in networks using group random forest methodology,” *J. Crit. Rev.*, vol. 7, no. 6, pp. 381–384, 2020.

[36] M. Kolkka, J. Virtanen, M. Kohlmaainen, J. Paananen, and K. Hahninen, “Random forest-based imputation outperforms other methods for imputing lc-ms metabolomics data: A comparative study,” *BMC Bioinf.*, vol. 20, no. 1, Oct. 2019, Art. no. 492.
Jake S. Rhodes received the BA degree in finance and economics from Utah State University, in 2015, the BS degree in applied mathematics from Weber State University, in 2017, and the PhD degree from Utah State University (USU), in 2022. He is an assistant professor with Idaho State University. During the last year of his doctoral program, he participated in the National Security Internship Program (NSIP), Pacific Northwest National Laboratory, where he worked in areas of vision and robust deep-learning models. His research interests include machine learning (supervised and unsupervised), deep learning, and dimensionality reduction.

Adele Cutler received the BSc (Hons) degree in mathematics from the University of Auckland, and the MS and PhD degrees in statistics from the University of California, Berkeley. She is a professor emerita with the Department of Mathematics and Statistics, Utah State University (USU). She is one of the pioneering researchers in the development of random forests and archetypal analysis. Her research interests are in statistical learning, statistical computing, random forests, and archetypal analysis.

Kevin R. Moon received the BS and MS degrees in electrical engineering from Brigham Young University, the MS degree in mathematics and the PhD degree in electrical engineering from the University of Michigan, respectively. He is an assistant professor with the Department of Mathematics and Statistics, Utah State University (USU). Prior to joining USU in 2018, he was a postdoctoral scholar (2016–2018) with the Genetics Department and the Applied Mathematics Program, Yale University. His research interests are in the development of theory and applications in machine learning, information theory, deep learning, and manifold learning. More specifically, he is interested in dimensionality reduction, data visualization, ensemble methods, nonparametric estimation, information-theoretic functionals, neural networks, and applications in biomedical data, finance, engineering, and ecology.