

## RESEARCH ARTICLE

# A general iterative clustering algorithm

Ziqiang Lin<sup>1</sup>  | Eugene Laska<sup>1,2</sup> | Carole Siegel<sup>1,2</sup>

<sup>1</sup>Department of Psychiatry, New York University Langone School of Medicine, New York, NY, USA

<sup>2</sup>Department of Population Health, Division of Biostatistics, New York University Langone School of Medicine, New York, NY, USA

**Correspondence**

Eugene Laska, Department of Psychiatry, New York University, One Park Ave, New York, NY, 10016, USA.  
Email: [eugene.laska@nyulangone.org](mailto:eugene.laska@nyulangone.org)

**Present Address**

One Park Avenue, New York, NY, 10016, USA

**Funding information**

National Institute on Alcohol Abuse and Alcoholism, Grant/Award Number: PO1AA027057-01

**Abstract**

The quality of a cluster analysis of unlabeled units depends on the quality of the between units dissimilarity measures. Data-dependent dissimilarity is more objective than data independent geometric measures such as Euclidean distance. As suggested by Breiman, many data driven approaches are based on decision tree ensembles, such as a random forest (RF), that produce a proximity matrix that can easily be transformed into a dissimilarity matrix. An RF can be obtained using labels that distinguish units with real data from units with synthetic data. The resulting dissimilarity matrix is input to a clustering program and units are assigned labels corresponding to cluster membership. We introduce a general iterative cluster (GIC) algorithm that improves the proximity matrix and clusters of the base RF. The cluster labels are used to grow a new RF yielding an updated proximity matrix, which is entered into the clustering program. The process is repeated until convergence. The same procedure can be used with many base procedures such as the extremely randomized tree ensemble. We evaluate the performance of the GIC algorithm using benchmark and simulated data sets. The properties measured by the Silhouette score are substantially superior to the base clustering algorithm. The GIC package has been released in R: <https://cran.r-project.org/web/packages/GIC/index.html>.

**KEYWORDS**

clustering, extremely randomized tree, iterative RF clustering, proximity, random forest

## 1 | INTRODUCTION

Finding distinct homogeneous clusters of a sample of units, each with many attributes or features, can clarify complicated heterogeneous relationships. For example, in medicine, an apparent heterogeneous disorder may actually be a combination of several subtype disorders with specific clinical and/or biological features. These features may indicate specific treatment with better outcomes and their identification through cluster analysis fulfills a primary goal of precision medicine. Complex illnesses

such as schizophrenia, alcohol use disorder, and PTSD will benefit from the modern ability to handle the large number of biological features that are now collected across multiple domains with increasing scientific and technological sophistication.

A *cluster* is a group of units that are close to each other and far from units in other clusters, where distance, *proximity* or *dissimilarity* is a function of the input attributes or features. There are two critical elements in a cluster analysis. The underlying distance between all pairs of units is a core ingredient. The second is a structured algorithm

for finding a partition of the units into separate groups that maximizes an objective function of dissimilarities. The overarching goal of the methods we introduce here is to improve cluster analysis, but the vehicle to accomplish it is to improve methods for obtaining distance measures. Historically, these were defined by data independent geometric measures such as Euclidean distance. However, experience has taught that defining similarity metrics based on formulaic assumption on data structures whose complexities are not well understood can lead to meaningless results. Data-dependent dissimilarity (DDD) [22], in contrast, provides a more principled measure of dissimilarity than does data independent geometric models. Much work has been done to develop and refine DDD measures.

In this communication, we introduce an approach that iterates between DDDs obtained from decision tree based supervised classifiers and the resulting new clusters. These in turn are used to obtain new estimates of DDD leading to new clusters until convergence. After providing background information in the Methods section, we describe our new approach, the general iterative cluster (GIC) algorithm. We illustrate the method using Breiman's random forest [6] and Geurts, Ernst, and Wehenkel's extremely randomized trees (ERT) [17] as base DDDs and partitioning around medoids (PAM) [21, 31] as the classification algorithm. The data sets on which the clustering algorithm are compared include real-world data sets, which are part of the set of benchmarks commonly used in cluster analysis research, as well as randomly generated simulation data.

## 2 | METHODS

### 2.1 | Background

A machine learning/data mining clustering task consists of identifying clusters from a data set of unlabeled units and their features. There is no ground truth. It is usually assumed that the training data set is comprised of random and independent samples from a fixed and unknown probability distribution over the set of all possible feature vectors. The most common cluster objective function is the average dissimilarity between each unit in the cluster and the center of the cluster. The widely used *k-means algorithm* utilizes the centroid and the *k-medoids algorithm* [21, 31] utilizes the medoid as the center. The medoid is the member of a cluster whose mean dissimilarity from all other members in the cluster is a minimum. It is generally believed that it is easier to interpret clusters determined by the *k-medoids* centers than those arising from a *k-means* analysis. Because it is based on means, the *k-means* approach is more vulnerable to outliers than

is *k-medoids*. The input to the *k-medoids* algorithm is an arbitrary dissimilarity matrix, whose elements are not required to satisfy the geometric distance metric conditions. Commonly used formula-based *dissimilarity* functions are Euclidean, Manhattan, Angular Distance, Hamming, Cosine Similarity, and the Huffman Code. Milligan [26] conducted a Monte Carlo study of 30 internal criterion measures for cluster analysis, and Hubalek [19] used 20 of the 43 similarity measures he collected for cluster analyses on mushroom data.

Recognizing the limitations of formula defined dissimilarity functions such as those listed above, a considerable number of methods have been proposed to produce DDD measures. One important way is based on a simple but elegant idea. Split the data into subsets using a specific decision rule at each node in a decision tree. Pairs of units that follow the same pathway down the tree to the terminal nodes or leaves are similar with respect to the decision rule. The proximity of a pair of units is the fraction of times a common path, defined in many possible ways, is followed. Then the square root of one minus the proximity is a dissimilarity measure.

#### 2.1.1 | Random forest

Many approaches to obtain DDD are variants or derivatives of the random forest (RF) algorithm introduced by Breiman [6], who built on Amit and Geman's [2] contributions on geometric feature selection, Ho's [19] work on random methods, and Dietterich's [13] random split selection approach. An RF is an ensemble of individual trees [6, 42] grown to obtain a classifier based on bootstrapped samples of labeled data on a sample of units. In the process a data driven proximity matrix is produced. A decision tree is grown from a bootstrap sample of units. At each node, a random subset of features is selected and an optimal splitting threshold determined, based on a criterion that maximizes a measure of node purity such as the degree to which units in the child nodes belong to a single class. A widely used criterion is the Gini impurity index [8]. The splitting process continues until an unpruned tree is grown. Replicate trees are grown following the same rules on independent bootstrap samples. Breiman [6] proposed that the proximity between units is the fraction of trees in which both members are in the same terminal node. Bicego and Escolano [5] performed an empirical evaluation of four RF learning schemes examining alternative forest parametrizations, distances, and clustering algorithms.

Our approach applies to any classification algorithm based on an ensemble of decision trees that utilize labels at nodes and produce a proximity matrix.

### 2.1.2 | Generalized RF

An ensemble of trees grown to obtain a classifier is a generalized RF that includes

- (A) an initialization label from which the classification process begins, for example, artificial or random labels for sample units;
- (B) rules at each node for growing a decision tree and a stopping rule for deciding when to discontinue splitting;
- (C) a rule for defining similarity between units.

The similarity matrix is turned into a dissimilarity matrix and used in a clustering program, such as k-medoids.

### 2.1.3 | Example initialization approaches for start-up labels

For decision trees ensembles designed for classification, unit labels are required to grow a tree. There are several ways of starting the process. One is to introduce an auxiliary sample. Labels identify whether a unit's data are from the original or the auxiliary sample. Breiman [6] and Breiman and Cutler's [7] approach, which we call RFC, is to randomly produce synthetic feature data from a reference distribution obtained by sampling from the product of empirical marginal distributions of the sample data. The motive is to reduce the between tree dependency. Shi and Horvath [33] proposed alternatives. In *AddCl1*, synthetic data are generated by randomly sampling from the product of empirical marginal distributions of the variables. In *AddCl2*, synthetic data are obtained by randomly sampling from the hyper-rectangle that contains the observed data. Siegel and colleagues [35] proposed "purposeful" clustering in which the auxiliary data are a sample from a separate population related to the purpose of the clustering. In their search for subtypes of PTSD for war fighters, they used an auxiliary set of data of healthy controls who were war fighters. In the initial iteration step, an RF was grown to distinguish these individuals from individuals with PTSD. Another approach is to assign labels to the units by any strategy. Dalleau and colleagues [11] proposed *AddCl3*, which randomly assigns a label for each unit. Yet another strategy is to apply a geometric measure such as Euclidean distance on the part of the feature vector that is numeric, enter the resulting between unit distances into a cluster program and use the resulting cluster membership as labels for the initial iteration of the decision tree ensemble.

### 2.1.4 | Example approaches to forming a decision tree with labels

A decision tree can include all subjects or a randomly chosen bootstrap sample. All available features can be considered for determining a splitting rule at each node or a random subset can be selected. Among candidate features at a node, the threshold on which to perform a split is usually chosen so as to optimize a criterion such as the Gini impurity index. Another approach is to find the linear combination of features that optimize the impurity index at each node. In classical RF, a random set of features is selected and the one with a threshold that produces the best Gini impurity index is used. At each node in an extremely random tree (ERT) introduced by Geurts et al. [17], a splitting threshold is randomly selected for each of a randomly selected subset of features at each node. The feature used is the one whose split produces the best value of the Gini impurity index.

### 2.1.5 | Example decision trees without a label based splitting criteria

In the extreme, forests can be grown that do not require labels at any stage. For example, Breiman [6] and Cutler and Zhao [10] introduced *extreme random splitting*, calling the resulting ensemble a *purely random forest*; both the feature and the location of the cut-point at every node in every tree in the ensemble is randomly chosen. Fernando and Webb [15] proposed the Centered Forest, which is an unsupervised stochastic forest. At each node in a tree, units are divided into two equal subsets by splitting at the median value of a randomly chosen feature. Recursive splits do not depend on labels. These kinds of forests cannot be used in the GIT we describe in Section 2.2 except for the initial run.

### 2.1.6 | Example definitions of similarity

Many contributions in the literature have been devoted to obtaining better distance measures for input to cluster programs. Breiman [6] used the fraction of trees two units are in the same terminal node. Aryal and colleagues [3] carried out a comparative study of data-dependent approaches without learning in measuring similarities of data objects. One alternative measure is called Zhu2, [43], which defines similarity as proportional to the average length of the path two units share in their travel down the tree to the terminal node. Another is "Zhu3" [43], which utilizes node weights, defined as the inverse of the number of units that reach the node at every node along the shared path. Ting and colleagues [38] define similarity between a pair of units as the ratio of units in the

training set reaching the lowest common ancestor of the pair. These authors used isolation forests (iForest) [25] to obtain pairwise similarity, defined as the probability mass of the smallest region in feature space covering the pair in a hierarchical partitioning of the space into non-overlapping and non-empty regions. The dissimilarity between two units across an iForest is the average probability mass in the deepest shared node in a collection of trees.

### 2.1.7 | Some examples of generalized RF distance ensembles for clustering

Kulkarni and Sinha [23] presented a taxonomy of various versions of random forest. Similar in spirit to a probability mixture distribution approach to clustering, Bicego [4] proposed a method based on a set of RFs, each one devoted to modeling one cluster. The RFs are iteratively updated using a k-means-like clustering algorithm. Yan et al. [41] proposed a method that randomly probes the vector space of features to detect locally “good” clusters that are subsequently aggregated by spectral clustering [39] to produce what they call cluster forests.

Several recent applications of ERTs have appeared in the domain of brain tumor segmentation [18, 29, 36]. Other applications include content-based image classification [27], image categorization and segmentation [34], and video segmentation [28].

### 2.1.8 | Cluster ensembles

Strehl and Ghosh [37] considered the problem of combining multiple partitions of a set of units into a single consolidated cluster set without reference to their source. For example, using Breiman’s approach to obtaining synthetic data, multiple RF runs, called the “ensemble constructor” over the same data set, or a single run over different data sets, are used to produce sets of clusters. Each one is called an “ensemble member” and collectively they are referred to as the “base clusters.” A “consensus function” combines the base clusters to produce an overall consolidated cluster. Alhusain and Hafe [1] used this method to determine underlying population structure based on genetic data. They call their method the random forest cluster ensemble (RFcluE). Clusters are found using k-means operating on the dissimilarity output of the RF based on Breiman’s algorithm after multidimensional scaling [14] is used for transformation to Euclidean space. The overall definition of similarity between any two units is the proportion of times in the ensemble that the pair are assigned to the same cluster. The so-called co-association matrix is input to an agglomerative hierarchical clustering algorithm to obtain the final cluster.

## 2.2 | The general iterative cluster algorithm

The GIC algorithm is very general and can be applied to DDD-based clustering approaches described above except in the purely random case. The simple idea is that new proximity matrices and clusters are obtained iteratively. The GIC algorithm begins by running the underlying or base classification method using an initialization procedure as required to obtain a proximity matrix, followed by running the selected cluster algorithm. Thereafter, each iteration uses the same base classifier followed by the same clustering algorithm. At each step, units are labeled according to the cluster to which they were assigned in the cluster algorithm in the previous iteration. The process continues until convergence. Convergence occurs when the assignment of units to clusters does not change, which corresponds to a proximity matrix that does not change. Techniques such as *AddCl1* or *AddCl2* can be used to generate synthetic data when called for by the base algorithm, but only for the first iteration. Because the results are potentially dependent on the random assignment or the random seed, the procedure is repeated many times to obtain a balanced data set. The approach can be conceptualized as providing improved estimates of the dissimilarity measures, which in turn produces improved clusters.

One example that can be used as the base classifier is the approach of Shi and Horvath [33]. After the first iteration, the synthetic data are not used again. Another example is to use ERT as the underlying classifier and PAM as the clustering method. We denote this by IERT. In the first and only the first step, either *AddCl1*, *AddCl2*, or *AddCl3* is used to assign labels. In one version of ERT, the square root of the number of features are randomly selected as candidates at each node and cut-points are randomly selected for each feature. The cut-point and feature with the best purity index are chosen for the split. The proximity matrix resulting from this ERT is converted to a dissimilarity matrix, which is input to PAM to obtain clusters. Thereafter, at each successive iteration, the ERT ensemble is grown based on unit labels corresponding to the cluster to which the units were assigned in the previous iteration. The process is repeated until convergence.

As a third example, in the RFcluE method for cluster ensembles [1], after the proximity matrix is first obtained, clusters are found using the GIC algorithm iteration process. The resulting proximity matrices are used by RFcluE as before. Multidimensional scaling is used to transform each one to Euclidean space, which is then passed to a k-means clustering algorithm.

## 2.3 | Real-world data sets for comparisons of base DDD and ICAs

### 2.3.1 | Evaluation using the iris data

The iris flower data set described by R. A. Fisher in 1936 [16] contains 50 examples of flowers from each of three iris species, *setosa*, *virginica*, and *versicolor*. It is considered one of the standard benchmark data sets for cluster analysis research and perhaps the best-known database to be found in the pattern recognition literature. Fisher's paper has been cited nearly 20,000 times. Four measures were taken for each flower, sepals length, sepals width, petals length, and petals width. Detailed information about these data can be found at <https://archive.ics.uci.edu/ml/datasets/iris>.

### 2.3.2 | Evaluation using a heart disease data set

The heart disease data set [12] comes from patients undergoing angiography in a multisite study conducted at the Cleveland Clinic in Cleveland, the Hungarian Institute of Cardiology in Budapest, the Veterans Administration Medical Center in Long Beach, and University Hospitals in Zurich and Basel. It too is considered one of the standard benchmark data sets for cluster analysis research. It is comprised of 120 individuals who have heart disease and 150 who do not. Although many measures were taken on each individual, 13 are considered the "standard" data set including age, gender, chest pain type, resting blood pressure, serum cholesterol, fasting blood sugar, resting electrocardiographic results, maximum heart rate achieved during exercise, exercise-induced angina, exercise-induced ST depression, the slope of the peak exercise ST segment, number of major vessels, and thal. Detailed information can be found at [https://archive.ics.uci.edu/ml/datasets/Statlog+\(Heart\)](https://archive.ics.uci.edu/ml/datasets/Statlog+(Heart)).

### 2.3.3 | Evaluation using standard real-world data sets

Dalleau et al. [11] studied ERT starting the clustering algorithm with AddCl3. We used the same real-world data sets they used for empirical evaluations. The size of the sample, the number of features, and the number of labels for each study are given in Table 5. The data are available on the UCI website <https://archive.ics.uci.edu/ml/index.php>.

## 2.4 | Evaluations using simulated data

Data sets were simulated for a variety of cases. We used 9 and 49 continuous features and 2, 5, and 10 clusters. A multivariate normal distribution was assumed with means equal to (0.5, -0.5) for two clusters, (0.5, -0.5, 1, -1, 0) for 5 clusters and (0.5, -0.5, 1, -1, 0, 2, -2, 3, -3, 5) for 10 clusters. For two clusters, the sample size was 200, for 5 clusters it was 500, and for 10 clusters it was 1000, with 100 units in each cluster. Random vectors were simultaneously generated with the specified marginal means, and the between-feature correlations were randomly generated from partial correlation. These are derived from specified eigenvalues of the covariance matrices with lower bounds set equal to one. [10, 20, 24]. A second set of data were produced using the same procedures, but in this simulation, an independent three-level categorical feature was added to the list of features. The simulations were performed using the r program NORTA [9] (<https://rdrr.io/github/superdesolator/NORTARA/>).

## 2.5 | Example base DDD ensemble algorithms

Two base DDD ensemble algorithms and their corresponding GIC algorithms were chosen to illustrate the method. PAM was used to obtain cluster results for RFC and ERT, and their GIC counterparts were labeled IRFC and IERT. It is not our purpose to contrast the base DDD methods but to investigate the degree of improvement in the proximity matrices and the resulting improvement in clusters that are obtained using the iteration process. The number of trees for RFC and IRFC runs was set to 1000, and for ERT and IERT runs it was set to 10,000. All other tunable parameters were set to their default values; the number of features randomly selected at each node for all four methods is the square root of the total number of features. The max depth for RFC and IRFC is reached when all leaves are pure or when all leaves have less than 2 units. The max depth for ERT and IERT is reached when the number of units in a node is one-third of the number of units in the sample. RFC and IRFC used a bootstrap sample. ERT and IERT used all subjects at each iteration. Since results produced by clustering algorithms are affected by initial values and the random seed, each of the four approaches was run 500 times for the iris and heart disease data. The results shown for these two data sets in Table 1 are the average of these runs and their standard deviations. These taught us how stable are the averages. As a consequence, for the remaining data sets shown in the table, the number of runs for ERT and ERTI was reduced from 500 to 10 based on the extensive computation time required for each run

and the standard deviations of the Silhouette scores and the Jaccard Indices used to appraise the GIC algorithm. For the iris and heart disease data, there were little to be gained by additional replication.

There are many implementations of the PAM clustering algorithm. We used the one in the “cluster” package in R [30]. For ERT, we used the Python packages numpy, pandas, and sklearn, and made modifications so that the proximity matrix was able to be accessed.

## 2.6 | Indices for appraising the clustering algorithms

There are many indices in the literature for appraising how good are the result of applying a clustering algorithm. In this section, we describe the two indices we used. Let  $C = \{C_1, C_2, \dots, C_m\}$  be the set of clusters obtained by applying a clustering method where  $m$  is the number of clusters. Denote by  $n_i$  the number of units in cluster  $C_i$ ,  $i = 1, 2, \dots, m$ . Then  $N = \sum_{i=1}^m n_i$  is the number of units in the entire sample.

### 2.6.1 | The Silhouette score

The Silhouette score [32] is a measure that indicates how close unit  $i$  is to members of its own cluster compared with how close it is to units of its nearest neighbor cluster. Suppose there are  $m$  clusters. For any data point  $i$  in  $C_i$ , let  $a(i) = \frac{1}{|C_i|-1} \sum_{j \in C_i, i \neq j} d(i, j)$  be the average distance between  $i$  and every other point in the same cluster, where  $d(i, j)$  is the distance between data point  $i$  and data point  $j$ . Let  $b(i) = \min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i, j)$  be the smallest average distance between  $i$  and all of the data points in each of the other clusters.  $b(i)$  is the average distance between  $i$  and members of the nearest cluster. Then the Silhouette score for unit  $i$  is  $s(i) = \frac{b(i)-a(i)}{\max(a(i), b(i))}$ . A Silhouette score takes values between  $-1$  and  $1$ , and as the value increases, the nearer the unit is to other units in its own cluster and the further it is from units in its nearest neighbor cluster.

### 2.6.2 | The Jaccard index

The Jaccard index [40], sometimes called the Jaccard similarity coefficient, is a measure of the similarity of two partitions,  $P_1$  and  $P_2$  in terms of the proportion of units that are in both partitions. Its formal definition is the number of units in the intersection of  $P_1$  and  $P_2$  divided by the number of units in the union of  $P_1$  and  $P_2$ . Here we will use it to compare a partition based on the true labels to a partition based on a clustering algorithm of the same data set.

The Jaccard index is defined to be  $\frac{|P_1 \cap P_2|}{|P_1 \cup P_2|}$ . A larger value of the index indicates greater similarity between  $P_1$  and  $P_2$ .

## 3 | RESULTS

### 3.1 | Evaluations using real-world data

Table 1 displays the Silhouette scores and the Jaccard indices for the RFC, ERT, and their counterpart, IRFC and IERT iterative clustering method for nine real-world data sets. For the iris data, the Silhouette score for the IRFC was the highest by far, improving from 0.17 for RFC to a remarkable 0.83. Starting at 0.23, the Silhouette score for ERT also had a very large improvement to 0.44. Figure 1 is a plot in Cartesian coordinates of petal length versus petal width, the top two features found in the RFC and IRFC list of important variables in the RF. It can be seen that for RFC, many virginica (green) dots were labeled setosa (red), while for ERT, many versicolor (blue) dots were labeled virginica (green). The iterations tended to correct these erroneous labels.

For the heart disease data set, the IRFC improvement over the RFC (0.41 compared with 0.2) as measured by the Silhouette score, just as for the iris data, was remarkable. The same was true but to a lesser extent for ERT (0.01) compared with IERT (0.19). The Jaccard index of the base classifier and the corresponding GIC algorithm were quite similar for both methods. A scatterplot of maximum heart rate achieved and exercise induced ST depression, two of the top three features found in the RF of the RFC and IRFC list of important variables, is displayed in Figure 2. It can be seen that there was an excess of normal controls with the base RFC, which was corrected by the IRFC method. The ratio of heart disease subjects to normal controls was close to the ground truth using IRFC. As for ERT and IERT, the ratio of heart disease subjects to normal controls is similar to that of IRFC.

Looking at all of the data sets as a whole, the Silhouette scores appear to be relatively low for both base approaches. RFC and ERT were relatively close to each other, with a large difference only once, for the Wisconsin data. The GIC improved the Silhouette scores for every data set for both RFC and ERT. The IRFC was larger in eight of the nine data sets and substantially larger four times. The overall effect of the GIC on the Jaccard index was small with one exception: the iris data changed from 0.29 for ERT to 0.52 for IERT.

### 3.2 | Evaluation using simulated data sets

Table 2 displays the Silhouette scores and the Jaccard indices for the RFC, ERT, and their counterparts, IRFC

**TABLE 1** Silhouette score and Jaccard index for RFC, IRFC, ERT and IERT clustering methods for real-world data sets

Dataset <sup>a</sup>	Silhouette score				Jaccard Index			
	RFC	IRFC	ERT	IERT	RFC	IRFC	ERT	IERT
Mean value and standard deviation								
Iris	0.169	0.834	0.023	0.437	0.648	0.706	0.287	0.517
(150, 4, 3)	(0.004)	(0.014)	(0.003)	(0.041)	(0.031)	(0.025)	(0.194)	(0.235)
Heart disease	0.022	0.407	0.009	0.192	0.474	0.414	0.177	0.179
(270, 13, 2)	(0.002)	(0.068)	(0.001)	(0.053)	(0.037)	(0.026)	(0.057)	(0.063)
Mean value								
Wisconsin	0.109	0.761	0.505	0.604	0.666	0.796	0.942	0.894
(699, 9, 2)								
Lung	0.055	0.301	0.094	0.274	0.268	0.261	0.095	0.182
(32, 56, 3)								
Breast tissue	0.224	0.709	0.282	0.409	0.331	0.355	0.427	0.431
(106, 9, 6)								
Isolet	−0.004	0.187	0.063	0.244	0.156	0.192	0.016	0.039
(1559, 617, 26)								
Parkinson	0.254	0.814	0.254	0.447	0.451	0.446	0.168	0.189
(768, 8, 2)								
Ionosphere	0.122	0.594	0.298	0.496	0.440	0.404	0.513	0.519
(351, 34, 2)								
Segmentation	0.245	0.603	0.295	0.528	0.405	0.386	0.179	0.137
(2310, 19, 7)								

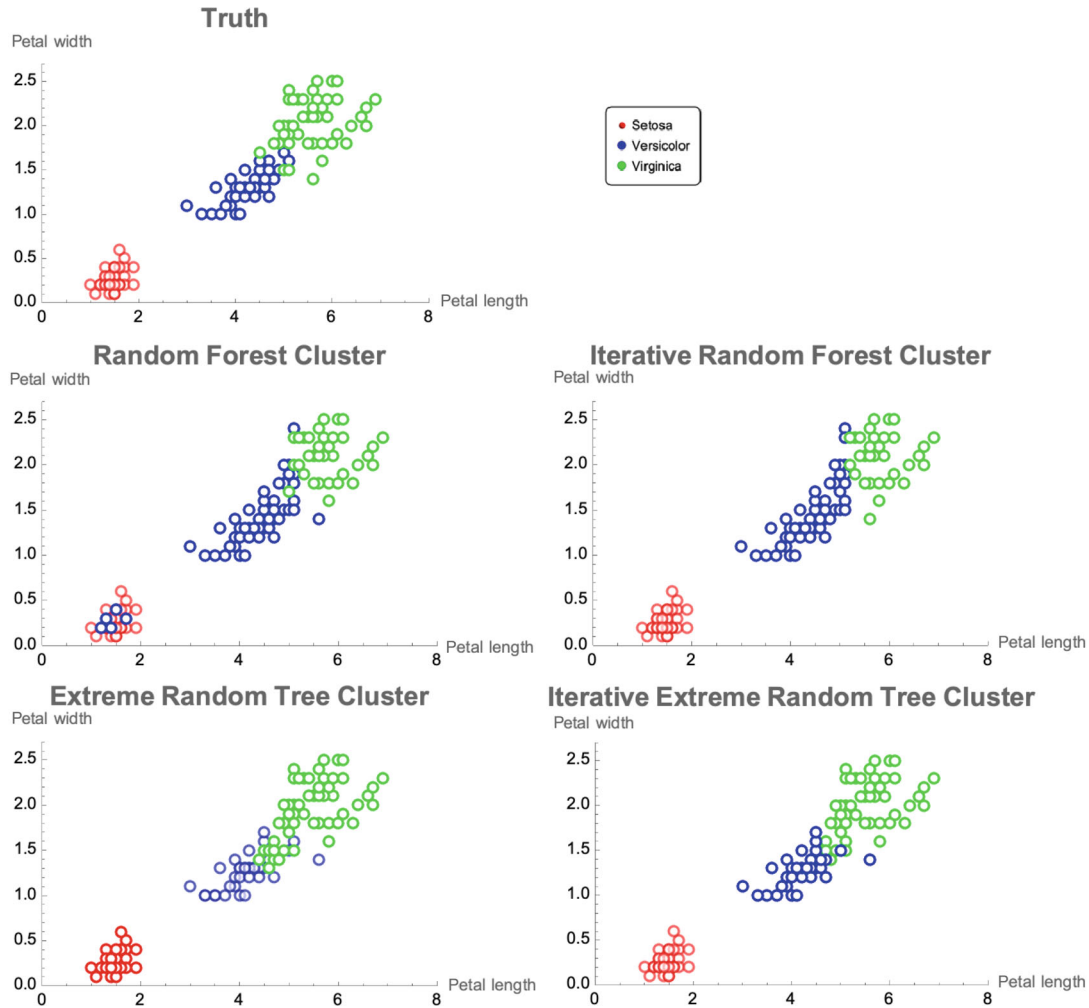
<sup>a</sup>Size of sample, number of features, number of labels.

**TABLE 2** Silhouette score and Jaccard index for RFC, IRFC, ERT and IERT clustering methods in simulated data with 9 and 49 continuous features

Number of clusters	Silhouette score				Jaccard index			
	RFC	IRFC	ERT	IERT	RFC	IRFC	ERT	IERT
9 continuous features								
2	0.016	0.539	0.110	0.182	0.376	0.366	0.493	0.487
5	0.008	0.356	0.073	0.113	0.143	0.134	0.081	0.068
10	−0.007	0.323	0.074	0.105	0.165	0.132	0.033	0.037
49 continuous features								
2	0.005	0.337	0.032	0.107	0.372	0.374	0.575	0.688
5	0.003	0.141	0.011	0.048	0.196	0.160	0.111	0.096
10	−0.018	0.105	0.039	0.097	0.325	0.211	0.036	0.028

and IERT iterative clustering methods for simulations with 9 and 49 continuous features. Table 3 shows the same information for simulations with 9 and 49 continuous features and 1 independent categorical feature. In every

case, the base method produced clusters with very poor Silhouette scores. For both methods, the GIC produced substantial improvements; the largest increment, as for the real-world data, accrued to RFC. The Jaccard indices, as



**FIGURE 1** Scatterplots of petal length versus petal width features for the iris data for ground truth and 4 clustering methods. Ground truth clusters are setosa, versicolor and virginica shown in the upper left plot

for the real-world data, did not have a consistent pattern of change, which in any case, was small.

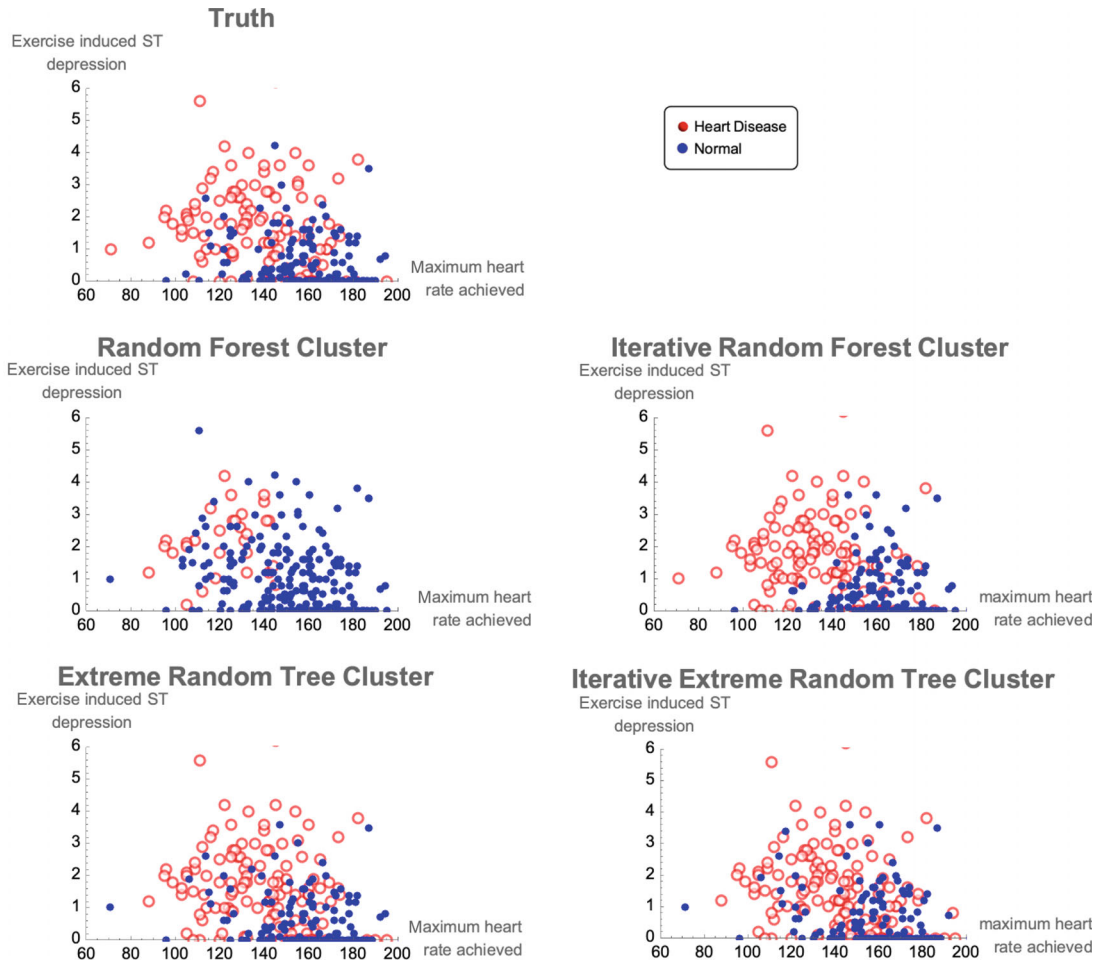
### 3.3 | Convergence of the GIC

It is natural to ask about rate of convergence of the GIC algorithms. Convergence occurs for PAM at an iteration where the medoids and cluster labels for each unit are the same as in the previous iteration. Tables 4 and 5 display the mean value at each iteration and the incremental change from one iteration to the next of the average between units distance respectively for the iris and heart disease data. This is the average of the entries in the proximity matrix. In the iris data, it can be seen that the absolute value of the differences is monotonically decreasing and close to zero by the seventh iteration for the IRFC and almost immediately for the IERT. For the heart disease data, the same pattern of monotonic decrease in the absolute value of the difference also converges to zero.

### 3.4 | Choice of number of clusters

In most applications, the number of clusters is unknown and a common issue is choosing a value to use in the clustering algorithm. Unfortunately, there is no completely satisfactory answer, and as a result, a variety of heuristics have evolved. Clearly, it is desirable to minimize distances between units in a cluster and maximize distances between units in different clusters. A common strategy is to create an objective function that balances the compactness and separation goals, and to choose the number of clusters that provides the maximum over a range of reasonable candidates. We used the Silhouette score as the function and found the maximum over a range of clusters from 2 to 11 in the iris data set. Of particular interest is the relationship between the value of the underlying DDD-based RF clustering method and its corresponding GIC. We found the maximum for the RFC occurred at six clusters with a Silhouette score of 0.178 and the maximum for the IRFC occurred at 2 clusters with a Silhouette score





**FIGURE 2** Scatterplots of maximum heart rate achieved versus exercise-induced ST depression for the heart disease data, for ground truth and 4 clustering methods

**TABLE 3** Silhouette score and Jaccard index for RFC, IRFC, ERT and IERT clustering methods in simulated data with 9 and 49 continuous features and one categorical feature

Number of clusters	Silhouette score				Jaccard index			
	RFC	IRFC	ERT	IERT	RFC	IRFC	ERT	IERT
9 continuous features +1 categorical feature								
2	0.021	0.500	0.143	0.339	0.385	0.360	0.347	0.370
5	0.006	0.314	0.109	0.231	0.135	0.128	0.063	0.045
10	-0.007	0.201	0.177	0.224	0.137	0.103	0.048	0.025
49 continuous features +1 categorical feature								
2	0.009	0.291	0.040	0.217	0.543	0.411	0.594	0.384
5	0.002	0.136	0.017	0.135	0.220	0.169	0.065	0.272
10	-0.011	0.100	0.079	0.187	0.168	0.219	0.027	0.042

of 0.966. The Silhouette scores over the range are shown in Figure 3). Notice that the scale of the ordinates are different in the two plots because of the sizeable difference in the ranges. The ground truth has three clusters with

a Silhouette score of 0.76. Figure 4 shows scatter plots of petal width versus petal length using 6 and 2 as input for the number of clusters for the RFC and IRFC respectively. The plot for the ground truth with 3 clusters is shown in

**TABLE 4** Mean pairwise proximity and change in mean pairwise proximity and standard deviation over iterations of IRFC and IERT for the iris data

Iteration	IRFC		IERT	
	Mean pairwise distance	Mean iteration change in pairwise distance	Mean pairwise distance	Mean iteration change in pairwise distance
1	0.717 (0.39)	-	0.658 (0.004)	-
2	0.702 (0.40)	-0.015 (0.08)	0.651 (<0.001)	-0.007 (0.004)
3	0.694 (0.41)	-0.008 (0.05)	0.651 (<0.001)	<0.001 (<0.001)
4	0.702 (0.70)	0.007 (0.04)	0.651 (<0.001)	<0.001 (<0.001)
5	0.696 (0.41)	-0.006 (0.04)	0.651 (<0.001)	<0.001 (<0.001)
6	0.698 (0.41)	0.002 (0.04)	0.651 (<0.001)	<0.001 (<0.001)
7	0.696 (0.41)	-0.002 (0.04)	0.651 (<0.001)	<0.001 (<0.001)
8	0.696 (0.41)	<0.001 (0.04)	0.651 (<0.001)	<0.001 (<0.001)

**TABLE 5** Mean pairwise proximity and change in mean pairwise proximity and standard deviation over iterations of IRFC and IERT for the heart disease data

Iteration	IRFC		IERT	
	Mean pairwise distance	Mean iteration change in pairwise proximity	Mean pairwise proximity	Mean iteration change in pairwise proximity
1	0.80 (0.22)	-	0.64(0.01)	-
2	0.87 (0.17)	0.07(0.11)	0.62 (0.01)	-0.02 (0.01)
3	0.86 (0.18)	-0.01 (0.09)	0.62 (0.01)	0.004 (0.001)
4	0.84 (0.20)	-0.02 (0.08)	0.62 (0.004)	0.003 (0.004)
5	0.82 (0.22)	-0.02 (0.08)	0.63 (0.004)	0.001 (0.001)
6	0.78 (0.24)	-0.04 (0.08)	0.63 (0.004)	0.001 (0.001)
7	0.77 (0.24)	-0.01 (0.08)	0.63 (0.004)	<0.001 (0.001)
8	0.77 (0.24)	-0.01 (0.07)	0.63 (0.004)	<0.001 (0.001)

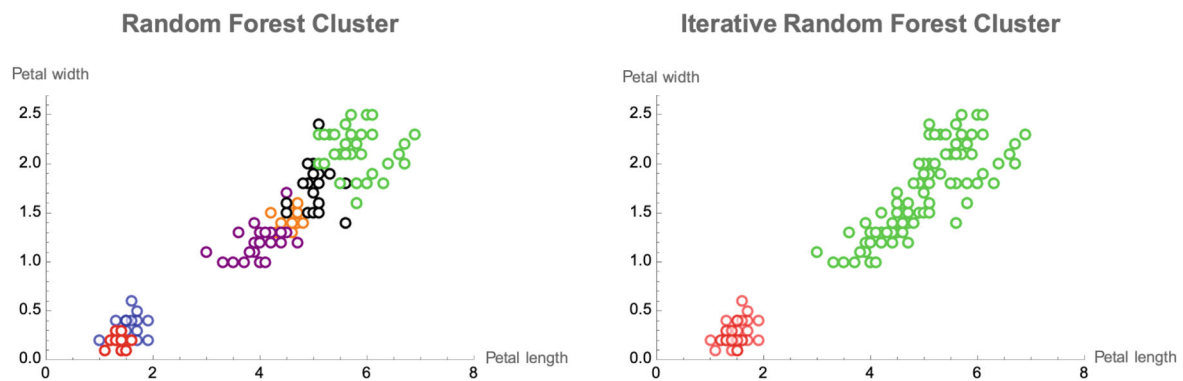
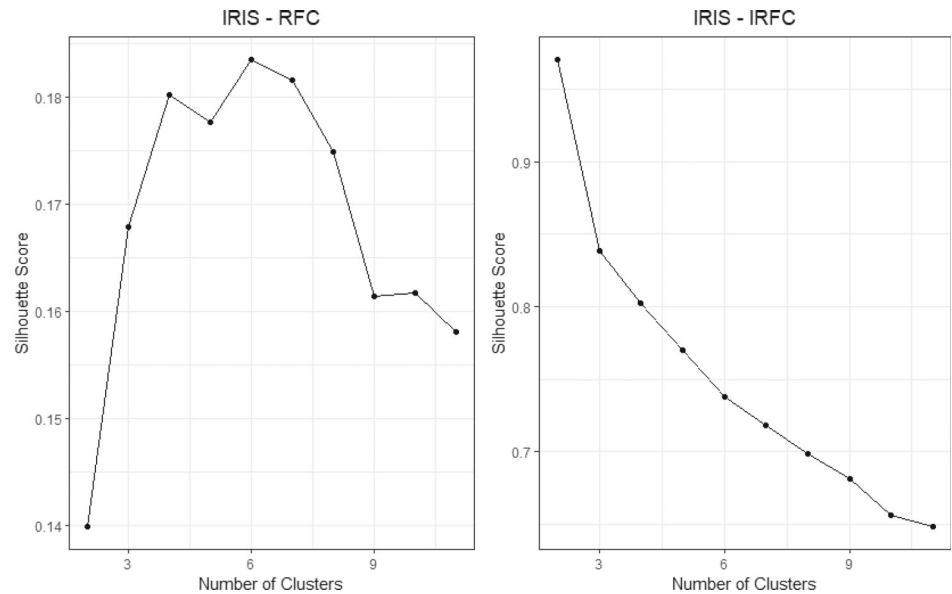
the upper left corner of Figure 1. From the plots, it appears that all three of the candidate number of clusters are visually plausible. If the choice were based entirely on the properties of the Silhouette score, we should use the proximity measures of the IRFC with 2 clusters. It is the subject matter expertise of the botanist that is required to declare that 3 is the true number of clusters.

### 3.5 | Choice of initialization labels

To start the iteration process, unit labels are required to grow the forest. We used the iris data assuming three clusters with three different initialization approaches followed by IRFC to iterate to the final clusters. In the first, synthetic feature data were randomly produced from a reference distribution obtained by sampling from the product of empirical marginal distributions of the sample

data [7]. The second method was “purposeful clustering” [35], which used the ground truth as the initial labels. The third method was *AddCl3* [11], which is just a random assignment of labels. Table 6 displays the number of flowers in each cluster and the Silhouette scores for each of the three label initialization approaches and the ground truth. Figure 5 displays the scatter plots of petal width versus petal length for these approaches in Cartesian coordinates. From the plots, it appears that all but *AddCl3* are plausible. This strategy had difficulty distinguishing members of the *sartosa* and *versicolor* clusters. The other two approaches were in full agreement with the ground truth identifying the same 50 *sartosa* irises. Not surprisingly, the purposeful label assignment produced the best Silhouette scores and the random assignment produced the worst. Although the difference in the Silhouette scores is small, it is visual inspection that informs the analyst that *AddCl3* produces an unsatisfactory clustering. These examples suggest that

**FIGURE 3** Silhouette scores over the range possible cluster numbers for the iris data. Note that the range of the ordinates in the two plots are not the same. The maximum for RFC is 6 and for IRFC is 2



**FIGURE 4** Scatterplots of petal length versus petal width for the iris data for the number of clusters that maximized the Silhouette scores, 6 for RFC and 2 for IRFC. Ground truth with 3 clusters are shown in the upper left plot in Figure 1

the data analyst should carefully consider the method to use to initialize to labels. The more known information about the ultimate clusters that can be used, the better.

## 4 | DISCUSSION

The real-world and the simulation examples demonstrate that, at least for these data sets, the new GIC algorithm produces clusters that have superior properties compared with the base method, as measured by substantially higher Silhouette scores. The Jaccard index values from the GIC algorithm were about the same as those from the base method. Though the pattern was replicated for all of the examples we considered, we provide no proof that this will be the case for every data set. Nevertheless, the evidence suggests that an analyst will likely obtain better results in a clustering application by using the GIC algorithm. There are many modifications of RF that have been proposed for

estimating similarities. We believe that except for purely random forests that make no use of labels, all of them can be improved by iterating as we have described at the appropriate point in the procedure. How large the improvement will be depends on the base procedure. In the data sets we examined, the degree of improvement in the Silhouette score for RFC was usually considerably greater than for ERT.

If the distance between every pair of units converges, it follows that the PAM results will converge too. In Tables 4 and 5, we see that the absolute value of the mean difference between successive iterations decreases monotonically. A monotonically decreasing series of positive values bounded below must converge. We have run the GIC algorithm on many simulated data sets for RFC and ERT in addition to those reported here and all have converged monotonically in absolute value by the eight iteration or so. But it is possible that the algorithm is trapped in a local minimum. Notice that the actual change in the mean

TABLE 6 Number of units in clusters and the Silhouette score for IRFC for different initial labels for the iris data

Initial label method	Setosa	Versicolor	Virginica	Silhouette score
Ground truth	50	50	50	0.759
Breiman and Cutler	50	66	34	0.834
Purposeful clustering	50	54	46	0.861
AddCl3	34	74	42	0.753

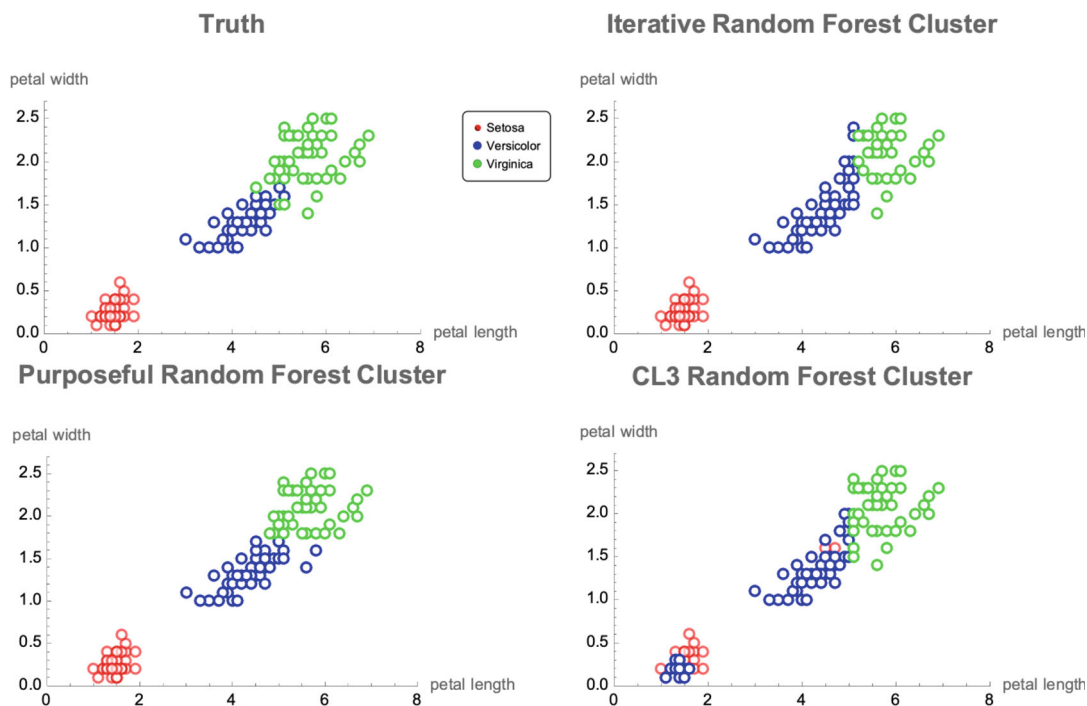


FIGURE 5 Scatterplots of petal length versus petal width for the iris data for different initial labeling strategies all assuming 3 clusters. Ground truth clusters are shown in the upper left

pairwise dissimilarity is relatively small, 0.021, 0.007, 0.03, and 0.009, as seen in Tables 4 and 5. This is likely why the Jaccard index does not change much.

In many applications, it is necessary to have a way to place a new unit into one of the discovered clusters. A new unit may be classified by running its feature vector down the final forest in the iteration. The proximities between the new unit and the medoids are equal to the fraction of terminal nodes they reside in together, or whatever way proximity is measured. The unit is assigned to the class corresponding to the largest of these proximities.

RF and its many versions are efficient algorithms with considerable capability for handling high-dimensional data. The iteration method provides an improvement in the generation of similarities and the clusters they produce. There are many properties of the GIC algorithm still to be learned for different data types. The GIC package has been released in R: <https://cran.r-project.org/web/packages/GIC/index.html>.

## ACKNOWLEDGMENTS

The authors would like to thank Dr. Charles Marmar for the many stimulating discussions about clustering and classification in mental health applications.


## CONFLICT OF INTEREST

The authors have no conflicts to disclose.

## DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available in <https://archive.ics.uci.edu/ml/index.php>.

## ORCID

Ziqiang Lin  <https://orcid.org/0000-0003-1990-6788>

## REFERENCES

1. L. Alhusain and A. M. Hafez, *Cluster ensemble based on random forests for genetic data*, *BioData Min.* 10 (2017), no. 1, 1–25.

2. Y. Amit and D. Geman, *Randomized inquiries about shape: An application to handwritten digit recognition*, Chicago Univ IL Dept of Statistics, 1994.
3. S. Aryal, K. M. Ting, T. Washio, and G. Haffari, *A comparative study of data-dependent approaches without learning in measuring similarities of data objects*, *Data Min. Knowl. Disc.* 34 (2020), no. 1, 124–162.
4. M. Bicego, “K-random forests: A k-means style algorithm for random forest clustering,” *2019 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2019, pp. 1–8.
5. M. Bicego and F. Escolano, “On learning random forests for random forest-clustering,” *2020 25th International Conference on Pattern Recognition (ICPR)*, IEEE, 2021, pp. 3451–3458.
6. L. Breiman, *Random forests*, *Mach. Learn.* 45 (2001), no. 1, 5–32.
7. L. Breiman and A. Cutler, *RfTools-for predicting and understanding data*, Interface’04 Workshop, 2004.
8. L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and regression trees*, Routledge, 2017.
9. H. Chen, *Initialization for NORTA: Generation of random vectors with specified marginals and correlations*, *INFORMS J. Comput.* 13 (2001), no. 4, 312–331.
10. A. Cutler and G. Zhao, *Pert-perfect random tree ensembles*, *Comput. Sci. Stat.* 33 (2001), 490–497.
11. K. Dalleau, M. Couceiro, and M. Smail-Tabbone, “Unsupervised extremely randomized trees,” *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Springer, 2018, pp. 478–489.
12. R. Detrano, A. Janosi, W. Steinbrunn, M. Pfisterer, J.-J. Schmid, S. Sandhu, K. H. Guppy, S. Lee, and V. Froelicher, *International application of a new probability algorithm for the diagnosis of coronary artery disease*, *Am. J. Cardiol.* 64 (1989), no. 5, 304–310.
13. T. G. Dietterich, *An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization*, *Mach. Learn.* 40 (2000), no. 2, 139–157.
14. R. P. Duin and E. Pekalska, *The dissimilarity representation for pattern recognition: A tutorial*. Technical Report, 2009.
15. T. L. Fernando and G. I. Webb, *SimUSF: An efficient and effective similarity measure that is invariant to violations of the interval scale assumption*, *Data Min. Knowl. Disc.* 31 (2017), no. 1, 264–286.
16. R. A. Fisher, *The use of multiple measurements in taxonomic problems*, *Ann. Eugenics* 7 (1936), no. 2, 179–188.
17. P. Geurts, D. Ernst, and L. Wehenkel, *Extremely randomized trees*, *Mach. Learn.* 63 (2006), no. 1, 3–42.
18. M. Goetz, C. Weber, J. Bloecher, B. Stieltjes, H.-P. Meinzer, and K. Maier-Hein, *Extremely randomized trees based brain tumor segmentation*, *Proc. BRATS Challenge-MICCAI*, 2014, pp. 006–011.
19. T. K. Ho, “Random decision forests,” *Proceedings of 3rd International Conference on Document Analysis and Recognition*, Vol 1, IEEE, 1995, pp. 278–282.
20. H. Joe, *Generating random correlation matrices based on partial correlations*, *J. Multivar. Anal.* 97 (2006), no. 10, 2177–2189.
21. L. Kaufman and P. J. Rousseeuw, *Finding groups in data: An introduction to cluster analysis*, Vol 344, John Wiley & Sons, 2009.
22. C. L. Krumhansl, *Concerning the applicability of geometric models to similarity data: The interrelationship between similarity and spatial density*, *Psychol. Rev.* 85 (1978), no. 5, 445–463.
23. V. Kulkarni and P. Sinha, *Random forest classifier: A survey and future research directions*, *Int. J. Adv. Comput.* 36 (2013), no. 1, 1144–1156.
24. D. Kurowicka and R. M. Cooke, *Uncertainty analysis with high dimensional dependence modelling*, John Wiley & Sons, Chichester, West Sussex, 2006.
25. F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation forest,” *2008 Eighth IEEE International Conference on Data Mining*, IEEE, 2008, pp. 413–422.
26. G. W. Milligan, *A Monte Carlo study of thirty internal criterion measures for cluster analysis*, *Psychometrika* 46 (1981), no. 2, 187–199.
27. F. Moosmann, B. Triggs, and F. Jurie, “Fast discriminative visual codebooks using randomized clustering forests,” *Twentieth Annual Conference on Neural Information Processing Systems (NIPS’06)*, MIT Press, 2006, pp. 985–992.
28. F. Perbet, B. Stenger, and A. Maki, *Random forest clustering and application to video segmentation*. BMVC, Citeseer, 2009, pp. 1–10.
29. A. Pinto, S. Pereira, H. Correia, J. Oliveira, D. M. Rasteiro, and C. A. Silva, “Brain tumour segmentation based on extremely randomized forest with high-level features,” *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, IEEE, 2015, pp. 3037–3040.
30. R Core Team, *R: A language and environment for statistical computing*. <https://www.R-project.org/>, 2021.
31. L. Rousseeuw and P. Kaufman, *Clustering by means of medoids*, *Proc. Stat. Data Anal. Based on the L1 Norm Conf.*, Neuchatel, Switzerland, 1987, pp. 405–416.
32. P. J. Rousseeuw, *Silhouettes: A graphical aid to the interpretation and validation of cluster analysis*, *J. Comput. Appl. Math.* 20 (1987), 53–65.
33. T. Shi and S. Horvath, *Unsupervised learning with random forest predictors*, *J. Comput. Graph. Stat.* 15 (2006), no. 1, 118–138.
34. J. Shotton, M. Johnson, and R. Cipolla, “Semantic texton forests for image categorization and segmentation,” *2008 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2008, pp. 1–8.
35. C. E. Siegel, E. M. Laska, Z. Lin, M. Xu, D. Abu-Amara, M. K. Jeffers, M. Qian, N. Milton, J. D. Flory, R. Hammamieh, B. J. Daigle Jr., A. Gautam, K. R. Dean, V. I. Reus, O. M. Wolkowitz, S. H. Mellon, K. J. Ressler, R. Yehuda, K. Wang, L. Hood, F. J. Doyle III, M. Jett, and C. R. Marmar, *Utilization of machine learning for identifying symptom severity military-related PTSD subtypes and their biological correlates*, *Transl. Psychiatry* 11 (2021), no. 1, 1–12.
36. M. Soltaninejad, G. Yang, T. Lambrou, N. Allinson, T. L. Jones, T. R. Barrick, F. A. Howe, and X. Ye, *Automated brain tumour detection and segmentation using superpixel-based extremely randomized trees in FLAIR MRI*, *Int. J. Comput. Assist. Radiol. Surg.* 12 (2017), no. 2, 183–203.
37. A. Strehl and J. Ghosh, *Cluster ensembles—A knowledge reuse framework for combining multiple partitions*, *J. Mach. Learn. Res.* 3, no. Dec (2002), 583–617.
38. K. M. Ting, Y. Zhu, M. Carman, Y. Zhu, and Z.-H. Zhou, “Overcoming key weaknesses of distance-based neighbourhood methods using a data dependent dissimilarity measure,” *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 1205–1214.
39. U. Von Luxburg, *A tutorial on spectral clustering*, *Stat. Comput.* 17 (2007), no. 4, 395–416.
40. X. L. Xie and G. Beni, *A validity measure for fuzzy clustering*, *IEEE Trans. Pattern Anal. Mach. Intell.* 13 (1991), no. 8, 841–847.

41. D. Yan, A. Chen, and M. I. Jordan, *Cluster forests*, *Comput. Stat. Data Anal.* 66 (2013), 178–192.
42. Z.-H. Zhou, *Ensemble methods: Foundations and algorithms*, Chapman and Hall/CRC, 2019.
43. X. Zhu, C. Change Loy, and S. Gong, “Constructing robust affinity graphs for spectral clustering,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1450–1457.

**How to cite this article:** Z. Lin, E. Laska, and C. Siegel, *A general iterative clustering algorithm*, *Stat. Anal. Data Min.: ASA Data Sci. J.* **15** (2022), 433–446. <https://doi.org/10.1002/sam.11573>