Distance-Based Random Forest Clustering with Missing Data

Matteo Raniero, Manuele Bicego(✉), and Ferdinando Cicalese

Computer Science Department, University of Verona, Verona, Italy
{manuele.bicego,ferdinando.cicalese}@univr.it

Abstract. In recent years there has been an increased interest in clustering methods based on Random Forests, due to their flexibility and their capability in describing data. One problem of current RF-clustering approaches is that they are not able to directly deal with missing data, a common scenario in many application fields (e.g. Bioinformatics): the usual solution in this case is to pre-impute incomplete data before running standard clustering methods. In this paper we present the first Random Forest clustering approach able to *directly* deal with missing data. We start from the very recent RatioRF distance for clustering [3], which has shown to outperform all other distance-based RF clustering schemes, extending the framework in two directions, which allow the integration of missing data mechanisms directly inside the clustering pipeline. Experimental results, based on 6 standard UCI ML datasets, are promising, also in comparison with some literature alternatives.

Keywords: Random Forest clustering · Missing data · Ratio RF distance

1 Introduction

Random Forests (RFs) [6,8] represent a widely and successfully applied model for Pattern Recognition and Machine Learning. RFs are ensembles of decision trees [19], models which define, in their basic version, a hierarchical splitting of the feature space. Generally speaking, Random Forests have been mostly studied for regression and classification, whereas in alternative scenarios, such as clustering, their potentialities have not been fully exploited yet. When considering clustering, methods based on Random Forest can be broadly divided into two classes: in the first RFs (or RF-like schemes) are directly used to perform clustering [2,16,17,23,32]; in the second class [1,3,6,22,26,33] RFs are employed to derive a meaningful dissimilarity measure, to be used with a standard distance-based clustering method, such as Hierarchical clustering or Spectral clustering. In this second line, which we call *distance-based RF-clustering*, different measures have been proposed, ranging from the simplest and most employed one defined by Breiman [6,22] up to more recent and complex dissimilarities [1,3,26,33].

One problem of all these RF-clustering approaches is that they are not able to deal with missing data [18], i.e. problems where some variables do not have

S. Sclaroff et al. (Eds.): ICIAP 2022, LNCS 13233, pp. 121–132, 2022.
https://doi.org/10.1007/978-3-031-06433-3_11

a value. These scenarios are very common, especially in the biomedical field [25], in which subjects involved in a clinical study may skip some exams [15], or high-throughput sequencing technologies may return incomplete data [27]. In general, in the clustering case, the typical solutions to this problem are [31]: i) to ignore objects with missing values, or, better, ii) to complete the data with imputation methods [18]. Imputation methods, to be performed before the analysis, replace a missing value with a new one. The simplest example, called Strawman imputation, replaces a missing value in a variable with the median of all non missing values for the same variable. Since these approaches do not explicitly consider the final task (clustering), they can have some limitations, as shown in some scenarios (see e.g. [5]). Therefore a more sophisticated and recent trend appeared, which proposes to face the missing data problem *directly* inside the clustering process. In this perspective, some methods have been proposed which extend known clustering techniques (e.g. K-means) [5,7,9,13,30]. However, to the best of our knowledge, such extensions for Random Forest clustering are completely missing. In this paper we make one contribution to fill this gap, proposing the first RF-clustering method able to directly deal with incomplete data. It is important to observe that, even if RFs have been employed in the missing data context (e.g. Missforests [24]), there are no RF-clustering methods able to directly work with incomplete data, and this represents the main contribution of this paper.

We start from the very recent RatioRF distance for clustering [3], which has shown to outperform all other distance-based RF clustering schemes [1,6,22,26,33]: this measure, defined on a set-based interpretation of the Tversky definition of similarity [28], determines the similarity between two objects by comparing their answers to a carefully selected subset of tests of the decision trees composing the forest. In this paper we propose two extensions of this framework to deal with missing data, both starting from the following observation: in the RatioRF framework a missing value represents a problem only when it is implied in a test of a node of the decision tree; in such case it is not possible to provide an answer to the binary test. To cope with this we can i) use a random decision (yes or no) or ii) keep both answers (yes/no) in an agnostic way. Due to the set-based formulation of the RatioRF distance both options can be easily integrated in the framework, as detailed in the paper. We evaluate the proposed scheme with some clustering experiments involving 6 UCI ML datasets, showing that: i) performances of RF-clustering did not degrade too much with moderate levels of missingness; ii) the two Ratio-RF modifications are equally reasonable, having different behaviours in different datasets; iii) the proposed distances compare very favourably with alternative classic distances for missing data.

The rest of the paper is organized as follows: in Sect. 2 we review the RatioRF approach, fixing the notation and introducing the basic concepts. The proposed approach is then fully presented in Sect. 3, and evaluated in Sect. 4. Finally, Sect. 5 concludes the paper.

2 Random Forest Clustering with RatioRF

In this section we will briefly introduce the starting point of our work, i.e. the very recent Random Forest clustering scheme using the RatioRF dissimilarity measure [3]. After introducing the RatioRF distance, we will briefly summarize the complete clustering scheme.

2.1 The RatioRF Distance

Assume we have set of objects/points U and a set of binary tests A (for attributes) defined over the whole set U, i.e., for each object $x \in U$ and each test $\theta \in T$ there is a unique value $\theta(x) \in \{yes, no\}$. A decision tree on a ground set of objects/points U and test set A is a binary tree T where: (i) each internal node ν is associated to a binary test $\theta_\nu \in A$; (ii) the two edges connecting the node to its children are associated with the two possible results—denoted Y for *yes* and N for *no*—of performing test θ_ν on an object from U. Further, ν_Y (resp. ν_N) denotes the child of ν connected to ν via the edge associated with Y (resp. N); $r(T)$ denotes the root of T. Let ν be a node of T at level $h+1$ and $\theta_1, b_1, \theta_2, b_2, \ldots, \theta_h, b_h$ be the sequence of nodes (tests) and edges (results), encountered on the unique path from $r(T)$ to ν. Then, it is possible to associate to ν the set of objects $S_\nu = \{x \in U \mid \theta_i(x) = b_i, i = 1, \ldots, h\}$. In words, a node ν is representative of (or it *contains*) all the objects that, when tested according to the adaptive strategy represented by the decision tree T, follow the path from the root to ν.

For each object x there is a single leaf containing it denoted as by $\ell(x)$. Let $P_T(x)$ be the set of pairs $(test, result)$ associated to x by the strategy/tree T

$$P_T(x) = \{(\theta, b_x^\theta) \mid \theta \text{ is a test on the path from the root } r(T) \text{ to the leaf } \ell(x) \text{ and } b_x^\theta = \theta(x)\}.$$

Let θ be a test and $b \in \{Y, N\}$. It is possible to say that x agrees with (θ, b) if $\theta(x) = b$. Similarly, objects x and y agree on test θ if $\theta(x) = \theta(y)$.

A decision tree T can be used to select the set of features Φ relevant for the assessment of similarity between pairs of objects from the universe U. In particular, in [3] authors define $\Phi = \{(\theta_\nu, b) \mid \nu \text{ is a node of } T,\ b \in \{Y, N\}\}$, as the set of possible outcomes of the tests used by the decision tree. For an object x its feature set $X = P_T(x)$ is defined as a set of test results on the path from $r(T)$ to the leaf $\ell(x)$ associated to x by the decision tree. These are the features from Φ that are most relevant for x, in the sense of being sufficient to identify x.

Now, assume we want to compare objects x, y represented by the set of features $X = P_T(x)$, and $Y = P_T(y)$, respectively. In [3] authors define

$$X \doteq Y = \{(\theta, b) \mid (\theta, b) \in X \text{ and } \theta(y) \neq b\} \tag{1}$$

to be the set of features that are relevant for x and on which y disagrees. Symmetrically the set of features relevant for y and on which x disagrees are given by the set

$$Y \dot{-} X = \{(\theta, b) \mid (\theta, b) \in Y \text{ and } \theta(x) \neq b\} \tag{2}$$

They also define

$$X \cap\!\!\!\cdot\, Y = \{(\theta, b) \in X \cup Y \mid \theta(x) = \theta(y)\} \tag{3}$$

to be the set of features on which x and y agree, among the features in $P_T(x) \cup P_T(y)$, which are those relevant for describing them (i.e., for identifying one or the other). The *Ratio-DecisionTree* similarity measure $\mathrm{RatioDT}(\cdot, \cdot)$ is defined by [3]

$$\mathrm{RatioDT}(x, y) = \frac{|X \cap\!\!\!\cdot\, Y|}{|X \cap\!\!\!\cdot\, Y| + |X \dot{-} Y| + |Y \dot{-} X|}, \tag{4}$$

As observed in [3], this similarity measure is symmetric and the corresponding dissimilarity obtained as $\sqrt{1 - \mathrm{RatioDT}(x, y)}$ is a metric.

Remark 1. In [3], the approach above was derived following an axiomatic definition of similarity measures given by Tversky [28]. An alternative perspective on such similarity measure computation is the following: let $\Phi(XY) = P_T(x) \cup P_T(y)$ denote the set of features restricted to those employed by the tree to describe x and y. Then let $X_{\Phi(XY)}$ (resp. $Y_{\Phi(XY)}$) be the element of $\Phi(XY)$ on which x (resp. y) agrees. Then, $\mathrm{RatioDT}(x, y) = |X_{\Phi(XY)} \cap Y_{\Phi(XY)}| / |X_{\Phi(XY)} \cup Y_{\Phi(XY)}|$, i.e., the Jaccard distance computed on the restricted set of features, that the tree selected for x and y.

The RatioDT similarity measure is straightforwardly generalized to Random Forests by averaging the decision tree distance in Eq. (4) over all the trees in the forest. More precisely, given a trained RF whose trees are $T_1, \ldots, T_m$, fix a pair of points $x, y \in U$ and let $\mathrm{RatioDT}_\mathrm{t}(x, y)$ be the similarity computed according to (4) from the decision tree T_t. Then, the Random Forest similarity measure $\mathrm{RatioRF}(x, y)$ is defined by averaging over all decision trees, i.e.

$$\mathrm{RatioRF}(x, y) = \frac{1}{m} \sum_{t=1}^{m} \mathrm{RatioDT}_\mathrm{t}(x, y). \tag{5}$$

If the clustering algorithm needs in input a dissimilarity, it is possible to transform the similarity into a dissimilarity using $\sqrt{1 - \mathrm{RatioRF}(x, y)}$, as done in [22].

2.2 The Complete Random Forest Clustering Procedure

The clustering is obtained with the following procedure:

1. **RF training**. In this step a Random Forest is trained on the data to be clustered. The main issue is that labels are not available: to face this issue it is possible to use Extremely Randomized Trees [12], i.e. trees in which the split feature and the threshold are chosen randomly – this representing a common and reasonably good solution for RF-clustering [4].

2. **Distance computation**. In this second step the RatioRF distance is computed from the trained forest, as explained in Sect. 2.1.
3. **Clustering**. Starting from the similarity, the final clustering is then obtained via any distance-based clustering algorithm, such as Hierarchical Clustering or Spectral Clustering [29].

3 Dealing with Missing Data

The RF-clustering scheme defined in the previous section requires all values for all features of the objects involved in the clustering. The presence of missing data impacts the first (RF training) and the second step (RatioRF distance computation). In the following we will introduce first how to derive the RatioRF distance with missing data, since this represents the most problematic part of the approach.

3.1 Computing RatioRF with Missing Data

The above definition of RatioDT assumes that all tests are defined on every objects. The presence of missing data in a data set is equivalent to the situation in which for some object x and test θ the value $\theta(x)$ is not defined, which is typically indicated by $\theta(x) = NAN$. There are two issues that need to be addressed if we want to employ the RatioDT(x, y) also in the presence of missing data. When an object x reaches a node ν such that $\theta_\nu(x) = NAN$:

1. should the pair (θ_ν, NAN) be part of the set of features X describing x?
2. what is the next node/test to consider for x between ν_Y and ν_N, i.e., how should we complete the partial root-to-leaf path for x beyond ν? How should we decide, considering that the test result $\theta_\nu(x)$ doesn't say which of the edges Y or N to follow?

Regarding point 1, our choice is not to consider such a node as part of the set X, as this would imply *unfounded* dissimilarity of x with any other objects y for which test θ_ν is defined. By *unfounded* we mean that we do not know whether the missing value of x on test θ_ν agrees or not with $\theta_\nu(y)$, hence it would not be fair to assume it is different.

Regarding point 2, we actually analyse two possibilities: (i) choosing at random whether to continue the root-to-leaf path for x on ν_Y or ν_N; (ii) extending the path in both directions, i.e., having $P_T(x)$ be a collection of root-to-leaf paths parting from one another at some node associated to a test where x is not defined. We call (i) the SinglePath approach and (ii) the MultiPath approach. Accordingly, we denote by $X^{SP} = P_T^{SP}(x)$ (resp. $X^{MP} = P_T^{MP}(x)$) the set of features (pairs of test and results) selected by the SinglePath (resp. MultiPath) approach. A simple pseudocode describing a recursive construction of such sets is given in Algorithms 1 2. Employing such procedures we assign $P_T^{SP}(x) =$SinglePath$(x, T, root(T))$ and $P_T^{MP}(x) =$MultiPath$(x, T, root(T))$.

Therefore, in the presence of missing data, we can compute RatioDT(x, y) like in (4) by substituting X, Y with X^{SP}, Y^{SP} (resp. X^{MP}, Y^{MP}).

Algorithm 1: SinglePath(x, T, v)

Input: A decision tree T; an object x; and a node ν of T
Output: a set X of relevant feature (pairs $(\theta_w, \theta_w(x))$ over some path from ν to a leaf of T.
if ν is a leaf **return** $\emptyset$;
if $\theta_\nu(x) = Y$ **then**
  return $\textsc{SinglePath}(x, T, \nu_Y) \cup \{(\theta_\nu, Y)\}$
if $\theta_\nu(x) = N$ **then**
  return $\textsc{SinglePath}(x, T, \nu_N) \cup \{(\theta_\nu, N)\}$
if $\theta_\nu(x) = NAN$ **then**
  choose ν_{next} randomly between ν_Y and ν_N;
  return $\textsc{SinglePath}(x, T, \nu_{next})$

Algorithm 2: MultiPath(x, T, v)

Input: A decision tree T; an object x; and a node ν of T
Output: a set X of relevant feature (pairs $(\theta_w, \theta_w(x))$ over some collection of paths T starting at ν and reaching a leaf.
if ν is a leaf **return** $\emptyset$;
if $\theta_\nu(x) = Y$ **then**
  return $\textsc{MultiPath}(x, T, \nu_Y) \cup \{(\theta_\nu, Y)\}$
if $\theta_\nu(x) = N$ **then**
  return $\textsc{MultiPath}(x, T, \nu_N) \cup \{(\theta_\nu, N)\}$
if $\theta_\nu(x) = NAN$ **then**
  return $\textsc{MultiPath}(x, T, \nu_Y) \cup \textsc{MultiPath}(x, T, \nu_N)$

3.2 Training Trees with Missing Data

In the training phase we also need to deal with the presence of missing data: in particular, we need to decide how an object x used in the procedure for building a tree is moved down (to the right or the left child?) after a split associated to a test for which x's value is missing/not known. Our choice is to have x continue on both child nodes. This appears to be well in the spirit of not assigning an arbitrarily imputed value to x for the test (since any choice would be unfounded as observed in the description of the testing phase)[1].

4 Experimental Evaluation

This section contains the empirical evaluation of the proposed approach. First we introduce the experimental details, then we present the results and discussion. A comparative analysis with literature alternatives concludes the section.

[1] Some experiments, not reported here, showed that empirical results would not change too much if we randomly choose one of the two paths.

4.1 Experimental Details

In order to evaluate our methods we consider some public datasets from the UCI ML Repository [10], whose details are reported in Table 1. As commonly done in clustering, we use supervised problems, remove labels, compute the clustering result and then compare it with the original labelling. In particular, we quantify the performance results for clustering quality considering the classic adjusted Rand index (ARI) [14].

Table 1. Details of the datasets used in the analysis.

Dataset	#objects	#features	#of clusters
Iris	150	4	3
Btissue	106	9	6
Wine	178	13	3
Glass	214	9	4
Leaf	340	15	30
Libras	390	90	15

To simulate missingness, data were artificially removed from these datasets using the MCAR (Missing Completely At Random) protocol [20]. This protocol consists in removing data completely at random, without taking into account any relationship between features. We only considered one constraint: no objects with all missing features can be considered. We considered 4 levels of missingness, i.e. removing 5%, 10%, 20% and 30% of the data. For every problem and each level of missingness we generated 20 datasets.

For what concerns the proposed RF-clustering scheme, in all experiments we trained RFs using the strategy described in Sect. 3.2: we used 100 trees in each forest, with $\log(n)$ for the maximum depth of each tree (with n the number of objects in the dataset). Once the RF distance is computed, the clustering is obtained using three classic approaches: spectral clustering, using the Ng-Jordan-Weiss normalized version [29], repeating the inner k-means 20 times, Affinity Propagation [11] and Hierarchical clustering, in the Ward-Link version.

4.2 Results and Discussion

In this section we compare the result of the proposed approach with the complete case, i.e. with the result obtained with the original RatioRF scheme on the complete matrix. The main goal of this analysis is to measure the impact of the missingness on the performance results for clustering quality. The results are reported in Table 2, for the different clustering methods and missingness values. In detail, the column "No Missing" contains the results with the original RF-Ratio scheme (we averaged the ARI among 20 repetitions); the other columns

contain the mean ARI of the two approaches (Single Path and Multi Path), averaged over the 20 generated datasets with missing data. In order to have a statistically significant comparison, for every missing level, we perform a paired t-test ($\alpha = 0.05$) with the complete case. Bold values in table represent those cases for which there is no statistical difference between the results with and without missing data (i.e. situations in which missing data does not impact the clustering performances). From the table it is evident that the proposed approach is robust in dealing with missing data; both versions are very robust with moderate levels of missingness: in fact, the performance results for clustering quality do not degrade too much with respect to the complete case, especially for 5–10 and in some cases 20% of missing data. Please note that, when using Affinity Propagation, we have robustness also in three datasets for a remarkable 30% of missing data.

Table 2. Results for the proposed approach, in comparison with the No Missing case.

Spectral clustering

Dataset	No Missing	Missing 5%		Missing 10%		Missing 20%		Missing 30%	
		SP	MP	SP	MP	SP	MP	SP	MP
Iris	0.6903	**0.7002**	**0.6936**	**0.6914**	**0.6781**	**0.6791**	**0.6757**	0.6633	0.5855
Breast	0.4169	**0.4109**	**0.4090**	0.3860	0.3893	0.3411	0.3696	0.3138	0.3566
Libras	0.2969	**0.2890**	**0.2882**	**0.2907**	**0.2910**	0.2774	0.2833	0.2527	0.2859
Wine	0.8623	**0.8666**	**0.8689**	**0.8669**	**0.8784**	0.8339	0.8365	0.7967	0.8060
Leaf	0.3891	0.3583	0.3723	0.3227	0.3607	0.2294	0.3082	0.1640	0.2396
Glass	0.1976	**0.1904**	**0.1928**	0.1782	0.1815	0.1767	0.1785	0.1683	0.1672

Affinity Propagation

Dataset	No Missing	Missing 5%		Missing 10%		Missing 20%		Missing 30%	
		SP	MP	SP	MP	SP	MP	SP	MP
Iris	0.7021	**0.7150**	**0.6827**	**0.7216**	0.6284	**0.6942**	0.4522	**0.6698**	0.3875
Breast	0.3800	0.3483	**0.3625**	0.3552	**0.3694**	0.3382	0.3281	0.3012	0.2933
Libras	0.2125	**0.2151**	0.2235	**0.2073**	**0.2232**	0.1876	0.2287	0.1613	**0.2098**
Wine	0.6882	**0.6983**	**0.6950**	**0.7231**	**0.7207**	**0.6600**	**0.6561**	0.6047	0.4972
Leaf	0.3540	**0.3472**	**0.3612**	0.3265	**0.3488**	0.2656	0.3291	0.2173	0.2812
Glass	0.1592	**0.1553**	**0.1526**	**0.1496**	**0.1480**	**0.1460**	0.1335	**0.1449**	0.1022

Hierarchical Clustering

Dataset	No Missing	Missing 5%		Missing 10%		Missing 20%		Missing 30%	
		SP	MP	SP	MP	SP	MP	SP	MP
Iris	0.7374	0.6951	**0.7132**	0.6808	0.6701	0.6558	0.6229	0.6147	0.5576
Breast	0.3673	**0.3562**	**0.3562**	**0.3551**	**0.3539**	**0.3448**	**0.3624**	0.3198	0.3419
Libras	0.2880	**0.2849**	**0.2905**	**0.2893**	**0.2876**	**0.2846**	**0.2876**	0.2697	0.2885
Wine	0.8868	**0.8617**	**0.8583**	**0.8370**	**0.8567**	0.8010	0.8018	0.6962	0.7610
Leaf	0.3931	0.3730	**0.3888**	0.3650	**0.3872**	**0.3207**	0.3731	0.2619	0.3293
Glass	0.2250	**0.2447**	**0.2483**	**0.2325**	**0.2399**	**0.2128**	**0.1941**	0.1641	0.1862

In order to have a better comparison between the SinglePath and the Multi-Path approach, we present in Fig. 1 the performance results for clustering quality of the two methods for the different datasets, averaging them among the three different clustering methodologies. In the figure, a filled mark denoted a situation in which one of the two approaches outperforms the other with a statistically significant difference (again according to a paired t-test with $\alpha = 0.05$). From the plots we can notice that there is not a clear best strategy, and the optimal solution highly depends on the datasets: it seems that MultiPath is preferable with datasets with several features (as Leaf and Libras), whereas SinglePath is more appropriate for low dimensional problems (as Iris).

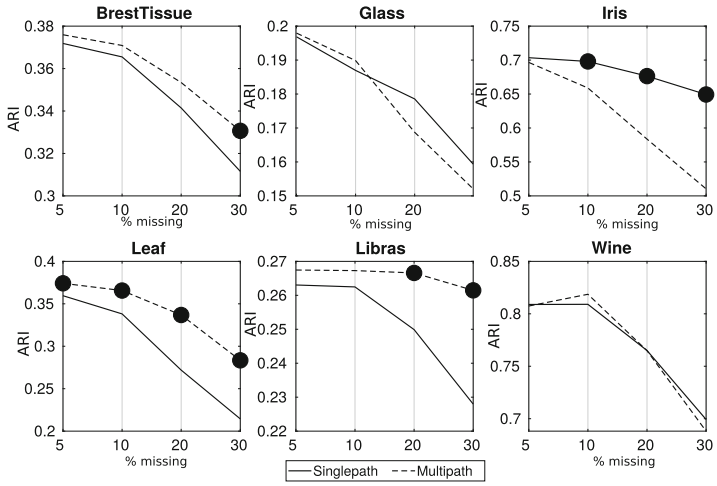

Fig. 1. SinglePath vs MultiPath

Finally, to check the overall validity of our approach, here we present a comparative analysis with some alternative distances used to deal with missing data, as described in the recent [21]. In particular we employed the Heterogeneous Euclidean-Overlap Metric (HEOM), the Heterogeneous Value Difference Metric (HVDM) and two redefinitions of these two, namely HEOM-REDEF and HVDM-REDEF. These distances can be computed with missing data (see [21] for a comparison between them), and are used in our framework as input for the three clustering procedures described before (Spectral clustering, Affinity Propagation and Hierarchical Clustering). As a further comparison, we also compute the results with the standard RatioRF pipeline on data pre-imputed with the simple Strawman method. This last comparison would permit to measure the benefits of including the management of the missing data inside the clustering procedure with respect to the pre-imputation solution.

In Fig. 2 results for the tree clustering schemes are averaged and presented using bar plots. From results, it is clear that our proposal largely outperforms

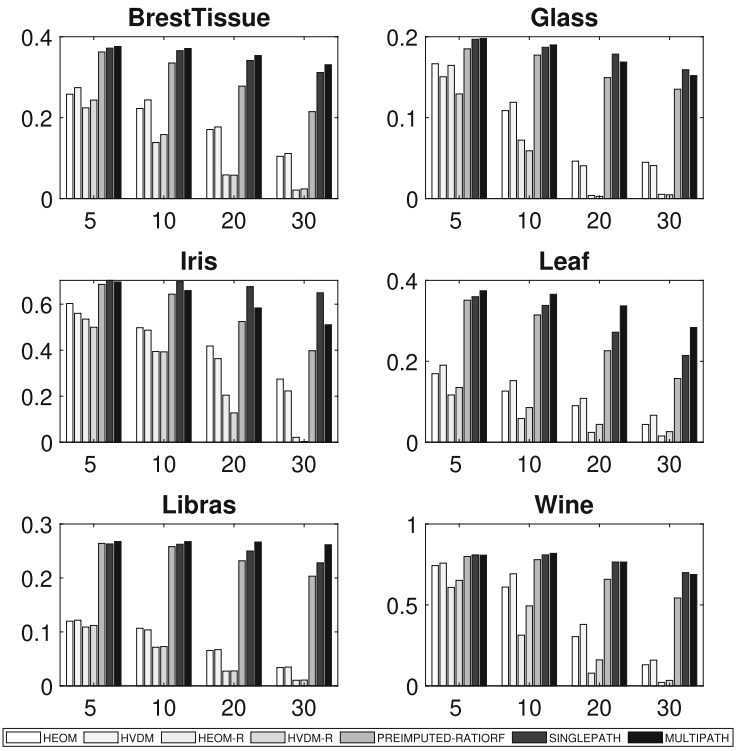

Fig. 2. Comparisons with other distances.

the alternative distances in almost all the cases, with improvements which are very relevant for large levels of missingness. Interestingly, the two proposed approaches also improve over the standard RatioRF pipeline applied on pre-imputed data, thus providing a further confirmation that it is more beneficial to deal with the missing data directly inside the clustering scheme, as shown for other clustering strategies in [5,7,9,13,30].

5 Conclusions

In this paper we presented an extension of the Random Forest clustering approach able to deal with missing data, based on two extensions of the recent RatioRF framework. An empirical evaluation confirms the robustness of the proposed strategies, both with respect to the results obtained with the complete data as well as in comparison with literature alternatives. In our future work we plan to add more empirical comparisons, in particular following two different directions: from one side we will enalrge the number of analysed datasets, in order to determine if there exists a correlation between the accuracies and the different aspects of a given dataset (its missingness nature, number of features/objects,

number of clusters); on the other side we will include in the analysis more comparisons with classic as well advanced approaches to deal with missing data, like imputation (using more sophisticated approaches like MICE or knn-imputation) or marginalization.

Acknowledgements. Authors would like to thank the anonymous reviewers for providing helpful comments and suggestions.

References

1. Aryal, S., Ting, K.M., Washio, T., Haffari, G.: A comparative study of data-dependent approaches without learning in measuring similarities of data objects. Data Min. Knowl. Disc. **34**(1), 124–162 (2019). https://doi.org/10.1007/s10618-019-00660-0
2. Bicego, M.: K-random forests: a K-means style algorithm for random forest clustering. In: Proceedings of International Joint Conference on Neural Networks (IJCNN 2019) (2019)
3. Bicego, M., Cicalese, F., Mensi, A.: RatioRF: a novel measure for random forest clustering based on the Tversky's ratio model. IEEE Trans. Knowl. Data Eng. (2022, in press). https://doi.org/10.1109/TKDE.2021.3086147, https://ieeexplore.ieee.org/document/9446631
4. Bicego, M., Escolano, F.: On learning random forests for random forest clustering. In: Proceedings of International Conference on Pattern Recognition, pp. 3451–3458 (2020)
5. Boluki, S., Dadaneh, S., Qian, X., Dougherty, E.: Optimal clustering with missing values. BMC Bioinform. **20**(Suppl. 12), 321 (2019)
6. Breiman, L.: Random forests. Mach. Learn. **45**, 5–32 (2001)
7. Chi, J., Chi, E., Baraniuk, R.: k-POD: a method for k-means clustering of missing data. Am. Stat. **70**(1), 91–99 (2016)
8. Criminisi, A., Shotton, J., Konukoglu, E.: Decision forests: a unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. Found. Trends Comput. Graph. Vis. **7**(2–3), 81–227 (2012)
9. Datta, S., Bhattacharjee, S., Das, S.: Clustering with missing features: a penalized dissimilarity measure based approach. Mach. Learn. **107**(12), 1987–2025 (2018). https://doi.org/10.1007/s10994-018-5722-4
10. Dua, D., Graff, C.: UCI machine learning repository (2017). http://archive.ics.uci.edu/ml
11. Frey, B.J., Dueck, D.: Clustering by passing messages between data points. Science **315**(5814), 972–976 (2007)
12. Geurts, P., Ernst, D., Wehenkel, L.: Extremely randomized trees. Mach. Learn. **63**(1), 3–42 (2006)
13. Hathaway, R., Bezdek, J.: Fuzzy c-means clustering of incomplete data. IEEE Trans. Syst. Man Cybern. B (Cybern.) **31**(5), 735–44 (2001)
14. Hubert, L., Arabie, P.: Comparing partitions. J. Classif. **2**(1), 193–218 (1985)
15. Jakobsen, J., Gluud, C., Wetterslev, J., Winkel, P.: When and how should multiple imputation be used for handling missing data in randomised clinical trials - a practical guide with flowcharts. BMC Med. Res. Methodol. **17**, 162 (2017)
16. Moosmann, F., Triggs, B., Jurie, F.: Fast discriminative visual codebooks using randomized clustering forests. In: Advances in Neural Information Processing Systems 19, pp. 985–992 (2006)

17. Perbet, F., Stenger, B., Maki, A.: Random forest clustering and application to video segmentation. In: Proceedings of British Machine Vision Conference, BMVC 2009, pp. 1–10 (2009)
18. Pigott, T.: A review of methods for missing data. Educ. Res. Eval. **7**(4), 353–383 (2001)
19. Quinlan, J.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc., Burlington (1993)
20. Rubin, D.B.: Inference and missing data. Biometrika **63**(3), 581–592 (1976)
21. Santos, M., Abreu, P., Wilk, S., Santos, J.: How distance metrics influence missing data imputation with k-nearest neighbours. Pattern Recogn. Lett. **136**, 111–119 (2020)
22. Shi, T., Horvath, S.: Unsupervised learning with random forest predictors. J. Comput. Graph. Stat. **15**(1), 118–138 (2006)
23. Shotton, J., Johnson, M., Cipolla, R.: Semantic texton forests for image categorization and segmentation. In: Proceedings of International Conference on Computer Vision and Pattern Recognition (CVPR 2008) (2008)
24. Stekhoven, D., Buhlmann, P.: Missforest: non-parametric missing value imputation for mixed-type data. Bioinformatics **28**(1), 112–118 (2011)
25. Sterne, J., et al.: Multiple imputation for missing data in epidemiological and clinical research: potential and pitfalls. BMJ **338**, b2393 (2009)
26. Ting, K., Zhu, Y., Carman, M., Zhu, Y., Zhou, Z.H.: Overcoming key weaknesses of distance-based neighbourhood methods using a data dependent dissimilarity measure. In: Proceedings of International Conference on Knowledge Discovery and Data Mining, pp. 1205–1214 (2016)
27. Troyanskaya, O., et al.: Missing value estimation methods for DNA microarrays. Bioinformatics **17**(6), 520–525 (2001)
28. Tversky, A.: Features of similarity. Psychol. Rev. **84**(4), 327 (1977)
29. von Luxburg, U.: A tutorial on spectral clustering. Stat. Comput. **17**(4), 395–416 (2007)
30. Wagstaff, K.: Clustering with missing values: no imputation required. In: Classification, Clustering, and Data Mining Applications, pp. 649–658 (2004)
31. Wagstaff, K.: Clustering with missing values: no imputation required. In: Banks, D., McMorris, F.R., Arabie, P., Gaul, W. (eds.) Classification, Clustering, and Data Mining Applications, pp. 649–658. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-642-17103-1_61
32. Yan, D., Chen, A., Jordan, M.: Cluster forests. Comput. Stat. Data Anal. **66**, 178–192 (2013)
33. Zhu, X., Loy, C., Gong, S.: Constructing robust affinity graphs for spectral clustering. In: Proceedings of International Conference on Computer Vision and Pattern Recognition, CVPR 2014, pp. 1450–1457 (2014)