2020 25th International Conference on Pattern Recognition (ICPR)
Milan, Italy, Jan 10-15, 2021

On learning Random Forests for Random Forest-clustering

Manuele Bicego
Università degli Studi di Verona, Verona (Italy)

Francisco Escolano
Universidad de Alicante, Alicante (Spain)

***Abstract*—In this paper we study the poorly investigated problem of learning Random Forests for distance-based Random Forest clustering. We studied both classic schemes as well as alternative approaches, novel in this context. In particular, we investigated the suitability of Gaussian Density Forests [1], Random Forests specifically designed for density estimation. Further, we introduce a novel variant of Random Forest, based on an effective non parametric by-pass estimator of the Rényi entropy, which can be useful when the parametric assumption is too strict. An empirical evaluation involving different datasets and different RF-clustering strategies confirms that the learning step is crucial for RF-clustering. We also present a set of practical guidelines useful to determine the most suitable variant of RF-clustering according to the problem under examination.**

I. Introduction

Random Forests (RFs) [2], [1] represent a widely known pattern recognition tool, whose usefulness has been largely shown in many different fields. From a general perspective, Random Forest approaches have been almost always studied for regression and classification: in these contexts they represent state-of-the-art approaches, able to compete with the most effective and established approaches like SVM or Neural Networks. In other pattern recognition scenarios, such as clustering, RFs have received less attention: even if some excellent RF-based approaches have been proposed, their exploitation in the clustering scenario is far from been as mature as in classification/regression. Among the different approaches proposed to RF-clustering, a particularly important trend – called *distance-based RF clustering* [2], [3], [4], [5] – follows the idea of exploiting the capabilities of RF in describing data to derive a meaningful *similarity measure* between points, to be exploited in a classic distance-based clustering algorithm, like hierarchical clustering or spectral clustering [6]. In these approaches, the first step is to learn a Random Forest, which is then used to derive the similarity: the main problem, here, is that labels are not available, and classic supervised strategies cannot be used. Even if crucial – proper RFs are fundamental to derive good similarities for clustering– this step has received poor attention by researchers, whose efforts have been mainly devoted to the definition of the similarity. This paper is aimed at filling this gap, and deals with the problem of learning proper Random Forests for distance-based RF clustering.

In literature, the most common approach consists in training a standard classification forest which discriminates between the original data and a synthetically generated negative class [3], [4]. Typically, the negative class is obtained by sampling points from the product of empirical marginal distributions of the observed data: in this way the dependency structure of the original data is removed. Other options, less investigated, rely on exploiting Extremely Randomized Trees [7], which can be built without labels, since they are based on random splits. For example, in [5], a general purpose – i.e. not specifically designed for clustering – RF-based similarity measure has been introduced, defined on the basis of Isolation Forests [8], a particular type of Random Forests designed for one-class classification which describe data using Extremely Randomized Trees.

In this paper we thoroughly study this problem of learning the Random Forest for distance-based RF clustering, and analyse and compare classic as well as alternative approaches. In particular, we investigated four strategies. The first strategy is the standard "sampling-negatives + classification RF", described above, which represents the baseline. The second strategy is based on Extremely Randomized Trees [7]: even if this strategy has been already employed, its utility in this context has not been sufficiently assessed, especially within an explicit comparison with the classic scheme. The third investigated strategy is based on Density Forests, i.e. Random Forests which perform density estimation [1], [9]; these tools, never used in this context, seem to be particularly suited, for two different reasons: i) they can be trained without labels, ii) they can provide a rich description of the data. Here we investigate the Gaussian variant introduced in [1], which assumes Gaussianity in each node. The last investigated strategy starts from the following two observations: i) Density Forests may suffer from classic and known problems of density estimation procedures (e.g. computation of partition function); ii) with parametric methods, we can get bad estimates if data do not follow the model assumption – the Gaussianity in each node in this case. We therefore introduced a novel variant, which we call *Rényi Random Forest*, in which, inside each tree, each split is decided on the basis of an effective non parametric estimation of Rényi entropy belonging to the class of bypass entropy-estimators [10]. The approach does not directly estimate the density, thus avoiding problems in density estimation, and is non parametric, thus working well when parametric assumptions do not hold.

These four strategies have been empirically tested with different datasets, different forest parametrizations, different RF-based distances and different distance-based clustering algorithms. Results show that the learning of forests is crucial

978-1-7281-8808-9/20/$31.00 ©2020 IEEE

in RF clustering, and that results obtained with the standard approach can be almost always improved with alternative strategies. The empirical analysis permitted also to derive a set of practical guidelines, useful to determine the most suitable variant according to the problem at hand.

Summarizing, the main contributions of the manuscript are:

- we investigated the problem of learning the Random Forest in Random Forest clustering: this was missing in the literature, since every approach just uses one scheme;
- we investigated the use of Gaussian Density forests in a context in which they were never used;
- we introduced a novel learning scheme, based on Rényi entropy estimators, specifically designed for this task;
- we provided an empirical comparison of the 4 different learning schemes, analysing different aspects, such as different Forest parametrizations, distances and clustering algorithms. This permitted also to derive a set of guidelines useful to choose, given a particular problem, the most suitable parametrization and configuration for RF-clustering.

The remainder of the paper is organized as follows: in Sect. II the needed background on Random Forests and Random Forest clustering is provided; the different learning schemes are presented in Sect. III; Sect. IV contains the experimental evaluation; finally, Sect. V concludes the paper.

II. Background

A. Trees and Forests

In the more general formulation, given a problem in a d-dimensional space, a decision tree is a *complete* binary tree T in which each internal node j is associated to a threshold θ_j and a feature f_j; the two edges connecting the node to its children are associated with the two possible results of performing the binary test defined by the pair (θ_j, f_j): an object $\mathbf{x} = [x_1, .., x_d]$ follows the left path if $x_{f_j} < \theta_j$, the right path otherwise. Trees can be learnt by exploiting a training set S, which is used to determine, at each node j, the optimal pair (θ_j, f_j). Typically, this is done by finding the pair (θ_j^*, f_j^*) which maximizes the information gain $I_{(\theta_j, f_j)}$,

$$\theta_j^*, f_j^* = \arg\max_{\theta_j, f_j} I_{(\theta_j, f_j)} \tag{1}$$

The information gain is often defined in terms of the entropy of the training points which arrive at node j; given S_j, the set of objects of the training set S reaching a particular node j, the information gain is defined as:

$$I_{(\theta_j, f_j)} = n_j H(S_j) - \sum_{i \in \{L,R\}} n_j^i H(S_j^i) \tag{2}$$

where $H(\cdot)$ represents an entropy measure, S_j^L, S_j^R are the set of objects of S_j which go through the left and the right paths, respectively, as obtained with the splitting pair (θ_j, f_j) (we remove (θ_j, f_j) from S_j^L and S_j^R for readability). The variables n_j, n_j^L and n_j^R denote the cardinality of S_j, S_j^L and S_j^R, respectively. For classification, the entropy $H(S)$ measures the purity of the classes. Other solutions are possible, as seen in the following.

Random Forests [2], [1] realize a robust ensemble of decision trees, exploiting a randomization mechanism in the learning of the different trees. The randomization level may vary a lot, ranging from the simplest exploitation of different random subset of samples to build each tree up to the extreme case of designing completely random trees, i.e. the so-called Extremely Randomized Trees [7]. The different trees are then aggregated to get the final model. Breiman in [2] shows that this aggregation exhibits different interesting theoretical properties – he derives an upper bound on the generalization error, in terms of the strengths of individual trees and their correlation. The different Forests used in this paper will be presented in Sect. III.

B. Random Forest clustering

As stated in the introduction, Random Forests have not been largely investigated for clustering, and their potentialities in this context have not been completely exploited. Here we focus of a specific class of RF clustering approaches, called distance-based RF-clustering [2], [3], [4], [5], which exploit the description capabilities of Random Forests to derive a similarity measure, to be used with a distance-based clustering algorithm such as Spectral Clustering or Hierarchical Clustering. These approaches are based on three steps:

1. Learning of the Forest. A RF is trained on the points to be clustered. Clearly, the main issue is that labels are not available.

2. Deriving pairwise similarities from the Forest. The trained RF is used to derive a similarity measure between points: in the simplest approach, introduced in [2], [3], the idea is to consider that two objects are similar if they end up in the same leaf of a given tree, since they have answered in the same way to all tests in their path. Given the forest, the similarity measure is thus represented by the number of times – over the whole set of trees – that two objects end up in the same leaf. Despite its simplicity, this similarity has shown to be very useful in many different clustering applications [11], [12], [13]. This reasoning can be extended in different ways, leading to more complex measures such as [4], [5]: more details can be found in the experimental session.

3. Clustering via a distance-based approach. Given the similarity measure, any distance-based clustering algorithm can be used, such as Hierarchical Clustering or Spectral Clustering [6].

In this paper we focus our attention on the first step, by analysing some different schemes, described in the following section.

III. The learning schemes

We investigated four different approaches: the first two have been already employed in the RF-clustering domain; the third is taken from the density estimation field but is new in the context of RF-clustering; finally the last represents a novel scheme, specifically designed for this task.

A. Classification Random Forests

The first option, which represents the baseline, is to use the classic Random Forest for classification, trained with randomly generated negative points [3], [4]. Standard classification decision trees, such as CART or C4.5 [14], are employed, trained with two classes: the positive class, which contains the points to be clustered, and a synthetically generated negative class, of the same size of the positive one. The negative class is obtained by random sampling from the product of empirical marginal distributions of the observed data: in this way the dependency between features is removed.

B. Extremely Randomized Random Forests

The second option is to use Extremely Randomized Trees [7], a class of trees which includes different levels of randomization in the random forests building: for example, we can randomly choose the feature on which to perform the split, or we can select it from a random subset; the effects of the possible randomization levels have been largely investigated by Geurts and colleagues in [7]. Here we employed the most extreme version, in which, at each node, the feature to be used to perform the split is chosen randomly, together with the splitting threshold. This version seems suitable in this context, since it does not need any supervision – i.e. no labels.

C. Gaussian Density Random Forests

Since the goal is to provide a description of the data, a possibility we investigate here is to employ Random Forests which perform density estimation [1], [9], which can be trained without labels. This possibility has never been investigated in the distance-based RF-clustering context. In particular here we considered the Density Forests introduced in [1], in which each tree provides a hierarchical description of the data, assuming that every node is distributed as a Gaussian. This assumption is useful: actually, in the learning of a tree, the best split at every node is the one which maximizes the Entropy gain: for Gaussians, the entropy can be explicitly computed, given the covariance which can be estimated with the points falling in that node. One needed clarification: obviously, density forests can be also directly exploited to get the final clustering, e.g. via mode seeking; in such cases, however, we have to face with known problems in density estimation. For example, we have to compute a proper partition function, in order to ensure probabilistic normalization: even if good numerical approximations exist [15], a closed form solution can not be derived. In our case, however, we do not have this issue, since we do not need to compute the density, but only the entropy gain, to be used when building the trees.

Going into details, the learning of a tree in this Gaussian Density Forests is performed again by choosing, at each node j, the pair (θ_j, f_j) which maximizes the information gain defined in eq. (2). In this case, the assumption is to have a multivariate Gaussian distribution at every node: therefore the entropy $H(S)$ is defined as the entropy of the d-dimensional Gaussian

$$H(S) = \frac{1}{2}\log((2\pi e)^d \det(\Sigma_S)) \quad (3)$$

where Σ_S is the covariance matrix of the set of points in S, and $\det(\cdot)$ indicates the determinant of a matrix.

In this regard, the Gibbs theorem ensures that Gaussian variables have the maximum entropy among all the variables with equal variance. This fact is exploited in [16] to use the Gaussian hypothesis as an upper bound of those related to the (unknown) empirical distribution. In this paper, the Gibbs theorem allows us to define Gaussianity as a first level of statistical refinement.

Thus, under the Gaussian hypothesis, the information gain in eq. (2) reduces to:

$$I^G_{(\theta_j, f_j)} = n_j \log(\det(\Sigma_{S_j})) - \sum_{i \in \{L,R\}} n_j^i \log(\det(\Sigma_{S_j^i})) \quad (4)$$

D. Rényi Random Forests

Gaussian Density Forests may be very adequate, since they provide effective Entropy estimation also in the parts of the trees where few points are present. However, there may be situations in which the Gaussianity assumption is too strict: to face these cases, here we propose a novel class of Random Forests, based on decision trees in which, similarly to Gaussian density forests of [1], each split is found by optimizing the entropy gain. However, differently than [1], we do not make any assumption on the shape of the data, and use a non parametric bypass entropy estimator of the Rényi entropy [10]. This method starts from the following observation: to get a non parametric estimate of the entropy, the simplest way is to get the density via non parametric approaches such as Parzen's windows, and then to estimate the entropy from the density – the so called plug-in estimators. However, when the number of features is somehow large, direct plug-in estimators cannot be used, suffering from the curse of dimensionality. In these cases, better estimates of the Entropy can be obtained using the so-called *bypass estimators* [16], which only rely on samples and do not need the density estimation to get the entropy. In our case, since we only need the entropy to choose the best split, these approaches seem to be very promising. Here we adopted the bypass entropy estimator proposed in [10], which – instead of estimating the classic Shannon Entropy, which estimation may suffer from some problems, see [10] – proposes a method to estimate the more general Rényi entropy. The Rényi entropy, for a random variable $\mathbf{x}$ taking values in $\Re^d$, is defined as:

$$H_\alpha(\mathbf{x}) = \frac{1}{1-\alpha}\log \int_{\Re^d} p(\mathbf{x})^\alpha d\mathbf{x} \quad (5)$$

The definition is given for $\alpha \neq 1$, while the limit $H_1 = \lim_{\alpha \to 1} H_\alpha$ represents the Shannon entropy. Given a set of n points $\mathbf{X}$, in [10] the estimation starts by building a generalized Nearest Neighbor graph of the samples in $\mathbf{X}$. In such graph, the nodes $\mathcal{V}$ represent the samples, whereas the edges $\mathcal{E}$ are defined on the basis of a set of integers $\mathcal{K}$, and contains, for every node, only the connections to its $\mathcal{K}$ neighbors – e.g., if $\mathcal{K} = [1, 3, 7]$, each node has a connection only with its first, third and seventh nearest neighbors. Proximities between

nodes are computed using Euclidean distances between corresponding samples. Given the graph, the estimator $\hat{H}_\alpha(\mathbf{x})$ is computed as:

$$\hat{H}_\alpha(\mathbf{X}) = \frac{1}{1-\alpha} \log \frac{L_p(\mathbf{X})}{\gamma n^{1-p/d}} \tag{6}$$

where $p = d(1-\alpha)$ and

$$L_p(\mathbf{X}) = \sum_{(\mathbf{x},\mathbf{y})\in\mathcal{E}} \|\mathbf{x}-\mathbf{y}\|^p$$

Finally, γ is a constant which can be estimated by generating a large sample $\mathbf{G}$ of M points in $[0,1]^d$ and setting

$$\gamma = \frac{L_p(\mathbf{G})}{M^{1-p/d}}$$

Typically [16], α is set to be as nearest to 1 as possible, in order to approach the Shannon Entropy. Even if some heuristic schemes to find the proper value of α have been proposed [16], in our experiments we fixed its value to 0.999999.

In our novel Rényi tree, we propose to use this entropy to estimate the best split to be chosen inside a node. In particular, we plug equation (6) into equation (2); after some mathematical manipulations, we have that the best split at each node j is the one which maximizes:

$$\begin{aligned} I^R_{(\theta_j,f_j)} =& n_j \left[\log(L_p(S_j)) - (1-\frac{p}{d})\log(n_j)\right] \\ &- \sum_{i\in\{L,R\}} n^i_j \left[\log(L_p(S^i_j)) - (1-\frac{p}{d})\log(n^i_j)\right] \end{aligned} \tag{7}$$

This entropy estimation, being non parametric, may permit better splits when distributions inside a node are not Gaussian. This is confirmed in our experimental evaluation. Just to provide an intuition here, please consider the data distribution in Fig. 1(top), and suppose that the optimal split should be chosen along the x-axis. As can be observed there are two clouds of points, each one composed by two parts: a denser left part together with a less dense right part. When estimating the best split using the Gaussian Entropy, the less dense part of the left group is considered as the left tail of the right group: the split is therefore chosen in the middle of the left distribution (vertical red dashed line). This is consistent with the maximum entropy of Gaussian variables. On the contrary, in the non parametric density estimation of the Rényi entropy we do not assume any underlying distribution, and a more reasonable split can be found (vertical blue dotted line). Thus, Rényi entropy acts as a statistical refinement of Gaussianity when needed.

IV. Experimental evaluation

In this section we present the experimental evaluation: after introducing the experimental details in section IV-A, we will present results and comments in IV-B. Subsequently, in section IV-C, we present some general guidelines useful to select proper versions and parameters for Random Forest Clustering.

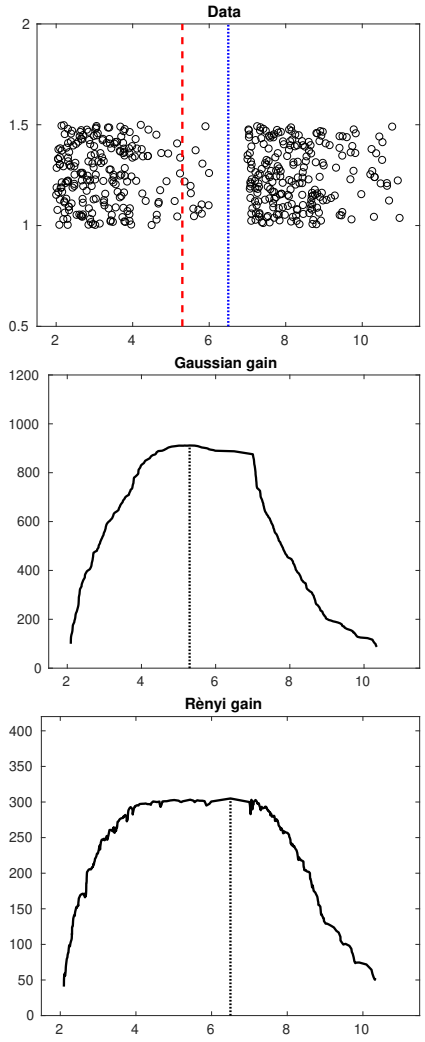

Fig. 1. Example of splitting.

A. Experimental Details

Training details. In the classic scheme with classification Random Forests ("Class-RF"), we used the Gini criterion to decide the best split. When learning each tree, the splitting process is stopped only when a node contains one element or only objects with the same label. For randomized Random Forests ("Random-RF") we stop the growing of the tree when reaching the max depth of 50 or when a node contains one element. For the Gaussian Density Random Forest ("GaussDens-RF"), for a proper estimation of the covariance matrix we stop the growing of the tree when a node contains less than 10 objects. We used full covariances, which, as typically done, are regularized by adding to the diagonal a small value (10^{-7}). Finally, for Random Forests based on Rényi entropy ("Rényi-RF"), we used $\mathcal{K} = \{3\}$, again stopping the growing of the tree when a node contains less than 10 objects.

In all cases every tree of every forest is built by randomly selecting 80% of dataset and by selecting the best split among 50% or 100% of the features. In our experiments we used

forests with 50 and 100 trees. For every configuration (number of trees - features), Random Forests have been trained 20 times, each one representing the starting point from which to compute the similarities.

Distances. We used four different distance measures. The "Shi" distance represents the first version of a Random Forest-based distance used for clustering, as defined in [2], [3]. As explained in section II, this measure is obtained by defining the similarity between x and y as the number of trees where x and y fall in the same leaf, divided by the total number of trees; the dissimilarity is obtained as the squared root of 1 minus the similarity. Then, we also used two measures introduced in [4]: the first ("Zhu2") represents the second variant[1] introduced in [4], which extends the approach in [2], [3] by defining the similarity as proportional to the averaged length of the path two objects have in common in their traversal down to the leaves. The third analysed measure ("Zhu3", the third variant introduced in [4]) weights every node in the common path – the weight of a node is computed as the inverse of the number of points which reach such node. Finally, we consider the distance defined in [5] ("Ting"), another Random Forest-based similarity which defines the distance between two points x and y as the ratio of points of the training set reaching the LCA (Lowest Common Ancestor) of x and y.

Clustering schemes and evaluation. Given the similarity/dissimilarity, clustering is performed with three different methods: i) Spectral Clustering [6], a typical choice in more recent RF-clustering works (e.g., [4]), using the Ng-Jordan-Weiss normalized version [6], and repeating the inner k-means 20 times[2]; ii) Affinity Propagation [17], a widely known distance-based clustering approach[3], and iii) a Hierarchical Clustering scheme (the Ward-Link variant, Matlab implementation). We evaluate the clustering procedures with supervised datasets, removing labels and comparing the obtained clusters to the original labelling using the classical *adjusted Rand index* (ARI – [18]) index – the higher the better. We tested the different learning schemes, parametrizations, measures, and clustering schemes using the 8 classic datasets described in table I, all available from the UCI Machine learning repository.

B. Results

The set of experiments encompasses a wide range of aspects: for every dataset and every learning scheme, we have different RF parametrizations (number of trees and features subsampling), different RF-measures, and different clustering algorithms. Considering that these last two aspects are the most interesting for RF clustering, we present in table II results aggregated for the four distances and the three clustering procedures, for every dataset (rows) and every learning scheme (column): for each entry of each table, the best (in average) parametrization of the Random Forest has been chosen (an analysis of Random Forest parametrization is reported in section IV-C). Reported numbers are averages over the 20 repetitions. A bold value in a line indicates that the accuracy of the corresponding learning scheme is better than all the other learning schemes, on the given dataset, with a statistically significant level according to a classic t-test with 0.05 of significance level.

TABLE I
DETAILS OF THE DATASETS EMPLOYED FOR TESTING.

Name	#objects	#features	#cluster	#obj per cluster
Iris	150	4	3	50,50,50
Wine	178	13	3	59,71,48
Glass	214	9	4	70,76,17,51
BTissue	106	9	6	21,15,18,16,14,22
Heart	297	13	2	160,137
Lung	32	54	3	9,13,10
Parkinsons	195	22	2	48,147
Auto-mpg	398	6	2	229,169

Different information can be evinced from the results. The first, and most important, is that the classic learning scheme (Class-RF) is hardly the best solution: in the 96 results reported in the table, only in two occasions it outperformed the alternatives with a statistical significance. This confirms the intuition behind this paper, i.e. that Random Forest clustering can really benefit from alternative learning schemes. The second observation is that Random Forests based on data entropy (GaussDens-RF and Rényi-RF) seem to be a really valid option in this context: in 52 cases over 96 they are significantly better than the alternatives; Random-RF were the best in 16 cases, whereas in the remaining 20 there are no statistically significant differences in the accuracies. However, it seems evident that the optimal learning scheme drastically depends on the dataset employed. In Section IV-C we will try to derive some general guidelines useful to choose the optimal version in a given case. For what concerns the distance, Zhu2, Zhu3 and Ting distances work equally well, with Zhu3 partially showing slightly superior results (37 over 96). Interestingly, there are different cases in which the Shi distance works very well, being better than the alternatives in 25 cases over 96. In the others, its behaviour is drastically worst than those of the alternatives. Finally, the Spectral Clustering algorithm seems to be the most adequate clustering scheme (59 over 128) – this confirms the intuitions in [4].

C. Guidelines for Random Forests clustering

In this section we propose some guidelines useful to choose the most suitable version: first, we provide suggestions for the best parametrization of the Forest, the best distance and the best clustering method; after this, we discuss guidelines for choosing, according to the given case, the most suitable RF learning scheme, which was the core of this paper.

Choice of the parametrization. For what concerns the first point, in Table III we present a comparative analysis of

[1]The first variant in [4] coincides with Shi.

[2]https://github.com/areslp/matlab/blob/master/spectral_clustering/ SpectralClustering.m

[3]The version we used allows setting the number of clusters, see http://www.psi.toronto.edu

TABLE II
RESULTS

Spectral Clustering

	Shi				Zhu2			
Problem	Class-RF	Random-RF	GaussDens-RF	Rényi-RF	Class-RF	Random-RF	GaussDens-RF	Rényi-RF
Iris	0.5270	0.3536	**0.8423**	0.7354	0.7166	0.7390	**0.8894**	0.6768
Wine	0.8561	**0.8914**	0.8667	0.4251	0.7871	**0.8834**	0.8234	0.4075
glass	0.1522	0.1781	0.2119	**0.2358**	0.1997	0.2237	0.1361	**0.3069**
BTissue	0.3977	0.3877	**0.4304**	0.1993	0.3778	0.3801	**0.4365**	0.3075
heart	0.2835	0.2706	0.2816	0.1581	0.2354	**0.3796**	0.1583	0.0195
Lung	0.1327	0.1863	0.1670	0.1793	0.1009	0.1842	0.1478	0.1708
Parkinsons	0.1619	0.1836	0.1771	**0.3868**	0.1695	0.1547	0.0964	0.0447
Auto-mpg	0.1202	0.2090	**0.4651**	0.1910	0.4080	0.4806	0.4177	**0.5224**

	Zhu3				Ting			
Problem	Class-RF	Random-RF	GaussDens-RF	Rényi-RF	Class-RF	Random-RF	GaussDens-RF	Rényi-RF
Iris	0.7357	0.7680	**0.8929**	0.7207	0.6979	0.7177	**0.8894**	0.6183
Wine	0.8588	**0.8973**	0.8363	0.4297	0.6416	**0.8690**	0.8173	0.3993
glass	0.2100	0.2387	0.1360	**0.3120**	0.1913	0.2272	0.1689	**0.3029**
BTissue	0.4037	0.4237	0.4360	0.2502	0.3471	0.3924	**0.4375**	0.3203
heart	0.2904	**0.3433**	0.2612	0.0454	0.1781	**0.3797**	0.1318	0.0197
Lung	0.1372	0.1970	0.1855	0.1726	0.1065	0.1834	0.1845	0.1706
Parkinsons	0.1701	0.1810	0.1068	0.0447	0.1685	0.1549	0.0940	0.0449
Auto-mpg	0.3971	0.3212	0.4617	0.4360	0.3854	0.4823	0.3307	**0.5224**

Affinity Propagation

	Shi				Zhu2			
Problem	Class-RF	Random-RF	GaussDens-RF	Rényi-RF	Class-RF	Random-RF	GaussDens-RF	Rényi-RF
Iris	0.5535	0.1399	0.5584	**0.6137**	0.6844	0.7251	**0.8845**	0.5956
Wine	0.7350	0.1472	**0.8183**	0.3286	0.6839	0.8153	0.8427	0.4660
glass	**0.2119**	0.1691	0.1631	0.1726	0.2033	0.2275	0.1386	**0.2983**
BTissue	0.3794	0.1813	**0.4464**	0.1987	0.3861	0.4030	**0.4459**	0.2657
heart	0.2673	0.3280	0.3171	0.0406	0.2679	**0.3765**	0.2714	0.0252
Lung	0.0976	0.0804	**0.1853**	0.1661	0.0833	0.1117	**0.2212**	0.1900
Parkinsons	**0.1777**	0.0370	0.0279	0.0505	0.1718	0.1655	0.1060	0.0447
Auto-mpg	0.2568	0.2794	0.2738	**0.3906**	0.3879	0.4436	0.3904	**0.5224**

	Zhu3				Ting			
Problem	Class-RF	Random-RF	GaussDens-RF	Rényi-RF	Class-RF	Random-RF	GaussDens-RF	Rényi-RF
Iris	0.6770	0.7113	**0.8787**	0.6313	0.6699	0.7157	**0.8940**	0.5846
Wine	0.7054	0.7626	**0.8700**	0.4769	0.6004	0.7842	0.8049	0.4165
glass	0.1928	0.2345	0.1374	**0.2977**	0.1806	0.2216	0.1673	**0.2975**
BTissue	0.3798	0.4037	**0.4536**	0.2394	0.3690	0.4018	**0.4460**	0.3148
heart	0.3050	**0.3603**	0.3199	0.0451	0.2344	**0.3732**	0.2017	0.0281
Lung	0.1278	0.1037	0.2212	0.1927	0.0863	0.1152	**0.2212**	0.1611
Parkinsons	0.1896	0.1681	0.1546	0.0419	0.1684	0.1609	0.0816	0.0447
Auto-mpg	0.4128	0.3885	0.3480	**0.5224**	0.3795	0.4775	0.3694	**0.5224**

Hierachical Clustering

	Shi				Zhu2			
Problem	Class-RF	Random-RF	GaussDens-RF	Rényi-RF	Class-RF	Random-RF	GaussDens-RF	Rényi-RF
Iris	0.4356	0.3402	0.7236	0.6993	0.6872	0.7426	**0.8965**	0.6246
Wine	0.8416	0.8140	0.6951	0.3395	0.7791	**0.8957**	0.6443	0.3947
glass	0.1337	0.0919	**0.2045**	0.1627	0.1896	0.1986	0.1504	**0.3253**
BTissue	0.3820	0.3454	**0.4209**	0.1968	0.4002	0.3762	**0.4334**	0.2525
heart	0.0895	**0.1816**	0.1241	0.0327	0.1199	**0.2088**	0.0690	0.0210
Lung	0.1362	0.1755	**0.2212**	0.1991	0.1203	0.1780	**0.2213**	0.1851
Parkinsons	0.1785	0.1655	0.1777	0.1519	0.1547	0.1795	0.0816	0.0447
Auto-mpg	0.0961	0.1388	**0.3190**	0.0857	0.3800	0.4121	0.2657	**0.5224**

	Zhu3				Ting			
Problem	Class-RF	Random-RF	GaussDens-RF	Rényi-RF	Class-RF	Random-RF	GaussDens-RF	Rényi-RF
Iris	0.6816	0.7227	**0.9019**	0.6328	0.6592	0.7325	**0.8857**	0.5604
Wine	0.8330	**0.8802**	0.6801	0.3944	0.5869	**0.8649**	0.6444	0.3949
glass	0.1683	0.1235	0.1689	**0.3152**	0.1702	0.2445	0.1689	**0.3219**
BTissue	0.4119	0.4061	0.4336	0.2172	0.3857	0.3774	**0.4458**	0.3158
heart	0.1622	0.1931	0.0703	0.0218	0.1185	**0.2007**	0.0690	0.0196
Lung	0.1122	0.1785	**0.2213**	0.1969	0.1149	0.1694	**0.2213**	0.1888
Parkinsons	0.1673	0.1975	0.0892	0.0447	0.1785	0.1540	0.0816	0.0447
Auto-mpg	0.2948	0.1748	0.3134	0.3677	0.3478	0.4245	0.2657	**0.5224**

the different options relative to each considered aspect (RF parametrization, distance, and clustering): for each aspect we compute the average of the ARI values of the different alternatives by varying all other aspects: for example, when analysing the distance (Table III(b)), we compute the average of all results obtained with Shi, Zhu2, Zhu3 and Ting, for all different RF-parametrizations, training schemes, clustering methods, and repetitions, thus resulting, for each dataset, in 960 values (4 parametrizations $\times$ 4 trainings $\times$ 3 clusterings $\times$ 20 repetitions). In the table, for each dataset, a bold value, if present, indicates the best option which has a statistical significant difference with respect to the others, according to an unpaired t-test with significance 0.05. More than one bold value represent equivalent options which are significantly better than non-bold values.

By analysing the Table different information can be derived: first, we can observe that the subsampling of the features is almost always beneficial (except in one case), leading to better performances. This is somehow expected, since it permits to increase the diversity in the trees composing the forest, this being crucial for proper generalization. For what concerns the number of trees, apparently there is no significant differences: our suggestion is to use 50 trees, since this permits faster computations. Regarding the distance, it can be observed that Zhu2, Zhu3, and Ting are almost always performing all reasonably well, with two exceptions: the Wine dataset, when the Ting distance is drastically worst that the other two, and the Auto-mpg, where the worst is Zhu3. For this reasons, our suggestion is to use the Zhu2 distance, which has also the larger average accuracy. Finally, for clustering the most reasonable choice seems to be to use the Spectral Clustering: in many cases it is better than the alternative, and when this does not happen, it represents the second best choice. Summarizing, here is our first set of guidelines: train forests with a reduced number of trees (50 may be enough), perform subsampling of features, extract the Zhu2 distance and get the final result via Spectral Clustering.

Choice of the learning scheme. For what concerns the learning scheme, if we focus our attention to the upper right part of Table II (Spectral Clustering and Zhu2 distance) we can observe that there is not a single best learning scheme, but that this varies according to the dataset. Considering the characteristics of the problems listed in Table I, we can observe that entropy-based methods (GaussDens-RF and Rényi-RF) are more adequate when input datasets are of reduced dimensionality (Iris, Gauss, BTissue and Auto-mpg). This is reasonable, since with low dimensional data entropy estimation can be easier. For datasets of higher dimensionality (Wine, Heart, Lung, Parkinsons) it seems more adequate to use the Random scheme, which does not need any estimation, being based on random choices: the only exception is the Parkinsons dataset, in which however the Random training represents the second best choice. For the entropy-based methods, it is however needed to decide if using Gaussian Entropy or Rényi entropy; to do that we can use the intuition which leaded us to the introduction of the latter: when the Gaussian assumption does not hold, then the use of Rényi-RF should be more reasonable. To confirm this, we restrict our attention to the four low-dimensional datasets (Iris, Gauss, BTissue and Auto-mpg), and we made a test for Gaussianity on the clusters contained in such datasets. To test Gaussianity we used the Royston's Multivariate Normality Test [19][4]: it represents the multivariate extension of the well known Shapiro-Wilk test [20], considered to be one of the best test of univariate normality [21]. According to this test, for Glass and Auto-mpg datasets all clusters are non-Gaussian, with a p-value below the Matlab precision value. For Iris and BTissue, most of the clusters are Gaussian, and the others have a p-value which is near the acceptance threshold (0.01). To provide numbers, after making the test for each cluster in a dataset we average the obtained p-values – please note that the lower the p-value the stronger is the rejection of the hypothesis of Gaussianity– : they are 0.0509 (Iris), 1.11e-07 (Glass), 0.122 (BTissue), 0 (Auto-mpg). Looking at the averaged accuracies in Table II, we can have a confirmation of our intuition: when clusters are mainly Gaussian (Iris and BTissue), the GaussDens-RF is the best choice, whereas in other cases it is better to use Rényi-RF. A further step is needed, since clusters are not known in advance: our proposal is to use the GaussDens-RF learning strategy, which represents a fast and effective scheme, and to test the Gaussianity of the obtained clusters: if all clusters are non-Gaussian, then we should train the Forest with the Rényi-RF, otherwise we can keep the forest trained with GaussDens-RF.

Summary. Summarizing, here are the guidelines for Random Forest Clustering: i) Number of Trees: 50; ii) Feature subsampling: 50%; iii) Learning: if the problem is high dimensional (e.g. dimensionality larger than 10), then use the Random-RF training; in the other cases use the GaussDens-RF strategy, and check the Gaussianity of the resulting clusters using the Royston's test; if all clusters are non-Gaussian, then train the forest with Rényi-RF; iv) Distance: Zhu2; v) Clustering: Spectral clustering. To confirm these guidelines, in Table IV we report, for each dataset, the ARI obtained with these guidelines, together with the average ARI and the ARI obtained with the best configuration: we can observe that the version with the Guidelines performs adequately well, with values which are always well above the average accuracy. Comparing with the maximum value, we can observe that the differences are almost always quite low, in the order of few points of percentage. The only exception is the Parkinsons problem, for which the difference is quite high. The motivations are still under investigation, but we should consider that this represents a very difficult case (the case with the lowest averaged accuracy), and a finer parameter tuning can be necessary to get adequate performances.

[4]We used the code of Trujillo-Ortiz et al., available at https://it.mathworks.com/matlabcentral/fileexchange/17811-roystest

TABLE III

ANALYSIS OF DIFFERENT ASPECTS: (A) RF NUMBER OF TREES AND FEATURE SAMPLING; (B) DISTANCE; (C) CLUSTERING METHOD. WITH "TOTAL" WE INDICATE THE NUMBER OF EXPERIMENTS OVER WHICH THE AVERAGE IS TAKEN.

	Trees and Feature sampling *(Total: 960)*				Distance *(Total: 960)*				Clustering method *(Total: 1280)*		
Problem	50-0.5	50-1	100-0.5	100-1	Shi	Zhu2	Zhu3	Ting	SC	AP	HC
Iris	**0.6629**	0.5668	**0.6731**	0.5664	0.3791	**0.6988**	**0.7001**	**0.6912**	**0.6469**	0.5961	0.6089
Wine	**0.6521**	0.4497	**0.6680**	0.4721	0.4956	**0.5941**	**0.6133**	0.5389	**0.6126**	0.5175	0.5514
glass	0.1802	**0.1970**	0.1804	**0.1986**	0.1554	**0.2024**	0.1934	**0.2050**	**0.1982**	**0.1920**	0.1769
BTissue	**0.3556**	0.3194	**0.3572**	0.3234	0.3083	**0.3471**	**0.3502**	**0.3499**	**0.3503**	0.3318	0.3346
heart	**0.1704**	0.1266	**0.1811**	0.1291	0.1448	**0.1558**	**0.1663**	0.1403	0.1750	**0.1931**	0.0873
Lung	0.1445	0.1342	0.1485	0.1450	0.1432	0.1410	0.1473	0.1407	0.1457	0.1310	**0.1525**
Parkinsons	**0.1132**	0.0836	**0.1164**	0.0879	0.0874	**0.1054**	**0.1078**	0.1006	**0.1133**	0.0909	0.0966
Auto-mpg	**0.3312**	0.2712	**0.3334**	0.2792	0.1605	**0.3980**	0.2599	**0.3965**	0.3155	**0.3371**	0.2587
Average	0.3263	0.2686	0.3323	0.2752	0.2343	0.3303	0.3173	0.3204	0.3197	0.2987	0.2833
	(a)				(b)				(c)		

TABLE IV

RESULTS OBTAINED WITH GUIDELINES.

Dataset	Guidelines	Average	Best
Iris	0.8893	0.6173	0.9019 (Gauss,100,0.5,Zhu3,HC)
Wine	0.8426	0.5605	0.8973 (Rand,100,0.5,Zhu3,SC)
glass	0.2430	0.1890	0.3253 (Rényi,100,1,Zhu2,HC)
BTissue	0.4365	0.3389	0.4536 (Gauss,100,0.5,Zhu3,AP)
heart	0.3796	0.1518	0.3797 (Rand,100,0.5,Ting,SC)
Lung	0.1831	0.1430	0.2213 (Gauss,50,1,Zhu2,HC)
Parkinsons	0.1547	0.1003	0.3868 (Rényi,100,0.5,Shi, SC)
Auto-mpg	0.4919	0.3037	0.5224 (Rényi,50,1,Zhu2,SC)

V. CONCLUSIONS

In this paper we investigated the problem of learning a Random Forest for distance-based Random Forest clustering. We analysed four different schemes, based on classic classification RF, extremely randomized RF, Gaussian Density Estimation Forest and on a novel variant of RF, based on a non parametric by-pass estimator of the Rényi entropy. A large empirical evaluation confirmed that the proper learning of the Random Forest is crucial in RF clustering, and that the classic scheme can be very often improved by more advanced schemes.

ACKNOWLEDGEMENTS

M. Bicego was partially supported by the University of Verona through the "Bando di Ateneo per la Ricerca di Base 2015" and "Internazionalizzazione di Ateneo 2018". F. Escolano is funded by the project RTI2018-096223-B-I00 of the Spanish Government.

REFERENCES

[1] A. Criminisi, J. Shotton, and E. Konukoglu, "Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning," *Foundations and Trends in Computer Graphics and Vision*, vol. 7, no. 2-3, pp. 81–227, 2012.

[2] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[3] T. Shi and S. Horvath, "Unsupervised learning with random forest predictors," *Journal of Computational and Graphical Statistics*, vol. 15, no. 1, pp. 118–138, 2006.

[4] X. Zhu, C. Loy, and S. Gong, "Constructing robust affinity graphs for spectral clustering," in *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014*, 2014, pp. 1450–1457.

[5] K. 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," in *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 1205–1214.

[6] U. von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.

[7] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine Learning*, vol. 63, no. 1, pp. 3–42, 2006.

[8] F. Liu, K. Ting, and Z. Zhou, "Isolation forest," in *Proc. of Int. Conf. on Data Mining*, 2008, pp. 413–422.

[9] H. Liu, M. Xu, H. Gu, A. Gupta, J. Lafferty, and L. Wasserman, "Forest density estimation," *Journal of Machine Learning Research*, vol. 12, pp. 907–951, 2011.

[10] D. Pál, B. Póczos, and C. Szepesvári, "Estimation of rényi entropy and mutual information based on generalized nearest-neighbor graphs," in *Advances in Neural Information Processing Systems 23*, 2010, pp. 1849–1857.

[11] K. R. Gray, P. Aljabar, R. A. Heckemann, A. Hammers, and D. Rueckert, "Random forest-based similarity measures for multi-modal classification of alzheimer's disease," *NeuroImage*, vol. 65, pp. 167 – 175, 2013.

[12] T. Shi, D. Seligson, A. Belldegrun, A. Palotie, and S. Horvath, "Tumor classification by tissue microarray profiling: Random forest clustering applied to renal cell carcinoma," *Modern Pathology*, vol. 18, pp. 547–557, 2005.

[13] S. I. Rennard, N. Locantore, B. Delafont, R. Tal-Singer, E. K. Silverman, J. Vestbo, B. E. Miller, P. Bakke, B. Celli, P. M. Calverley *et al.*, "Identification of five chronic obstructive pulmonary disease subgroups with different prognoses in the eclipse cohort using cluster analysis," *Annals of the American Thoracic Society*, vol. 12, no. 3, pp. 303–312, 2015.

[14] J. Quinlan, *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., 1993.

[15] S. Gupta, "Probability integrals of multivariate normal and multivariate t," *Annals of Mathematical Statistics*, vol. 34, no. 3, 1963.

[16] P. Benavent, F. Escolano Ruiz, and J. Saez, "Learning gaussian mixture models with entropy-based criteria," *IEEE Trans. on Neural Networks*, vol. 20, no. 11, pp. 1756–1771, 2009.

[17] B. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, p. 972976, 2007.

[18] L. Hubert and P. Arabie, "Comparing partitions," *Journal of Classification*, pp. 193–218, 1985.

[19] J. Royston, "An extension of shapiro and wilk's test for normality to large samples," *Applied Statistics*, vol. 31, no. 2, pp. 115–124, 1982.

[20] S. Shapiro and M. Wilk, "An analysis of variance test for normality," *Biometrika*, vol. 52, pp. 591–611, 1965.

[21] C. Mecklin and D. Mundfrom, "A monte carlo comparison of the type i and type ii error rates of tests of multivariate normality," *Journal of Statistical Computation and Simulation*, vol. 75, pp. 93–107, 2005.