An Extension of Random Forest-Clustering Schemes Which Works with Partition-Level Constraints

Manuele Bicego(✉) and Hafiz Ahmad Hassan

Computer Science Department, University of Verona, Strada le Grazie 15,
37135 Verona, Italy
manuele.bicego@univr.it

Abstract. Many classical clustering algorithms, like K-Means, spectral clustering, or hierarchical approaches, have been adapted to work with constraints; surprisingly, the literature completely lacks constrained versions of Random Forest Clustering (RFC) schemes, a class of methods whose usefulness has been shown in different scenarios. In this paper, we take one step to fill this gap, proposing a simple extension of RFC which works in the presence of partition-level constraints. In particular, the proposed approach exploits the modularity of RFC schemes, which all start from a Random Forest (RF) trained on available (unlabelled) data, by integrating in this first step the a priori knowledge given by the constraints, leaving the remaining part of the pipeline unchanged. We show the feasibility of our simple extension on three different RFC schemes, employing 18 datasets of small and moderate size. We also positively compare the obtained constrained RFCs with respect to some literature alternatives.

Keywords: Random Forest Clustering · Constrained Clustering · Decision Trees

1 Introduction

Clustering represents a widely investigated and applied exploratory data tool whose usefulness has been assessed in different scenarios (e.g., [21–23]). Clustering is definitely a challenging problem due to its unsupervised nature: actually, in clustering, no labels are available, and the natural groups (i.e., clusters) should be extracted directly from data. In some practical scenarios, however, some extra information is available, which can be used to derive better results. One example is *constrained clustering* [10,14,18], a paradigm which belongs to the widely investigated family of approaches for semi-supervised learning ([45]).

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78383-8_7.

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15324, pp. 93–108, 2025.
https://doi.org/10.1007/978-3-031-78383-8_7

In constrained clustering, the a priori information is provided in the form of constraints. Such constraints can be of different types and work at different levels, like constraints on clusters (e.g., imposing a minimum size on clusters), on pairs of instances (e.g., knowing pairs of objects that must or must not be in the same cluster), or directly on partition (e.g., having a set of objects for which labels are known) – for more info see e.g., [14].

Due to the scientific relevance and the practical usefulness of constrained clustering solutions, many classical clustering algorithms have been adapted to the constrained case: the most striking example is the K-means algorithm [28], for which several constrained versions have been proposed, starting from the COP-Kmeans approach of [48] up to more recent versions [9,19,47]. Other examples include constrained extensions of spectral clustering [36,49], density-based clustering [24], and hierarchical clustering [11].

Surprisingly, the literature completely lacks constrained versions for Random Forest Clustering (RFC) schemes, a clustering paradigm which exploits Random Forests (RFs) [7], typically employed in supervised settings, to perform clustering [3,4,6,16,29,34,38,39,41,50,52]. This paper takes one step to fill this gap, presenting a simple extension of RFC schemes that work in the presence of partition-level constraints. We focus on the most important class of RFC approaches, which are often simply referred to as *the* Random Forest Clustering approach [3,6,16,38–40,52]. Within this paradigm, the clustering is obtained in three steps: i) a RF is created to unsupervisedly model the data; ii) a distance between objects is derived through the learned RF and iii) the final clustering is obtained through a standard distance-based algorithm, such as hierarchical clustering or spectral clustering – the differences among various RFC schemes typically lie in the definition of the RF-distance. Our starting intuition is that it is possible to straightforwardly embed the constraints in the first step of the pipeline, i.e., in the learning of the forest. Please note that in RFC the unsupervised learning of the forest is still an unsolved issue (see e.g., discussions in [3]), since no labels are available: the standard solutions imply generating a synthetic negative class, and to train a classification Forest (done e.g., in [6,39,52]), or to use Extremely Randomized Trees (ERT – [15]), which can be trained without labels (done e.g., in [3,29,41]), and have shown to be very effective in deriving powerful distances [5]. In this paper, we describe a constrained extension of these RFC methods, focusing on partition-level constraints, in which the constraints are given in terms of a subset of labels for the objects [14]. This represents a classic scenario [27], very often also used to derive instance-level constraints (i.e., Must-Link and Cannot-Link constraints, which are derived from a subset of labelled objects) – for some discussions on the different forms of constraints interested readers can refer to [10,31,35].

The proposed extension is very simple and suggests replacing the unsupervised training of the forest with a *supervised* training, done with the labelled objects, discarding all the unlabelled data. Our idea starts from the observation that, in RFC schemes, the training of RFs is problematic (no labels [3]); however, the RF is not used to directly perform clustering, but only to define the tests used to compute the distance (i.e., as a *feature extractor* – [6]); therefore an RF encoding the constraints, even if not describing all the objects of our problem, would be definitely descriptive, providing more focused tests able to characterize the different clusters.

The presented approach has been thoroughly evaluated with 18 datasets, employing 3 different versions of RFC: the original one of Shi and Horvath [39], the approach of Zhu and colleagues [52], and the very recent RatioRF method [6]. Results show that our simple extension is drastically beneficial, especially for datasets of moderate size. We also present a comparison of constrained RFCs with some literature approaches, showing that the proposed methods represent a viable alternative to standard as well as advanced constrained clustering methods.

The remainder of the paper is organized as follows: in section 2, we summarize the RFC scheme, while in section 3, we introduce the proposed extension; section 4 contains the empirical evaluation, whereas section 5 concludes the paper.

2 Random Forests and Random Forest Clustering

In this section, we will provide a brief overview of the class of RFC which we consider, starting from Decision Trees (DT) and RF – mainly to set up the notation. In the more general formulation of [8], given a vectorial representation of d features, a DT t is a *complete* binary tree, where each internal node j has associated a test $\theta_j = (\nu_j, f_j)$, with ν_j being a threshold on a feature f_j; the two children represent the two possible results of the binary test $\theta_j = (\nu_j, f_j)$: more in detail, an object $\mathbf{x} = [x_1, .., x_d]$ goes to the left child if $x_{f_j} < \nu_j$, to the right one otherwise. Given an object $\mathbf{x}$, and a tree t, let us denote as $\ell_t(\mathbf{x})$ the leaf of the tree where the object $\mathbf{x}$ falls, and as $P_t(\mathbf{x})$ denotes the set of tests on the path $\mathbf{x}$ is taking from the root to the leaf $\ell_t(\mathbf{x})$.

Typically, DTs are learned starting from a training set $\mathbf{X}$, used to determine, for all nodes j, the optimal tests $\theta_j = (\nu_j, f_j)$. More in detail, the training follows a recursive procedure: in a given node, i) the best pair (ν_j, f_j) is found according to an optimality criterion evaluated using the objects arrived at that node, and ii) the objects are propagated to the left or the right node according to the result of the test. This recursive procedure starts from the root, where all objects are used, and is recursively iterated until nodes contain a single object or a maximum depth is reached. The optimality criterion used inside a node depends on the task: for example, in classification DTs [8], where labels are available, the best rule is the one that maximizes the separability of the classes of the objects reaching that node, such as the Gini criterion. RFs [7] represent a robust ensemble of Decision Trees, obtained with a randomization mechanism in the learning of the different trees: in the typical scenario, M different Decision Trees are built starting from random subsamples of the problem training set, and the final decision is taken by averaging decisions of the different trees.

2.1 Random Forest Clustering

In recent years, a great interest has arisen in the exploitation of RFs beyond the classical regression and classification scenarios, especially in unsupervised contexts such as outlier detection [25,26] or clustering. In this last scenario, there are some approaches that exploit RFs (or RF-like schemes) to directly devise clustering algorithms, such as [4,29,34,41,50]. However, as said in the introduction, the most important trend is the so-called distance-based RF clustering (often simply referred to as *the* Random Forest Clustering method [3,6,16,38–40,52]), which exploits the data description abilities of RF to derive a *distance* between points, to be employed with classic distance-based clustering algorithm, such as spectral clustering or hierarchical clustering. In this class of approaches, clustering is performed in three steps, briefly summarized in the following.

Step 1. Training of the Random Forest. A forest is trained on the available data. Since labels are not available (clustering), unsupervised methods should be derived. The typical options are two:

- **Negative Sampling**: in this option, we train a classical classification forest, e,g, based on CART [8], in which the set of points to be clustered represents the positive class, while the negative class is synthetically generated. This last step is typically performed by random sampling from the product of empirical marginal distributions of the dataset, in order to create a negative class of the same size of the dataset. This training scheme represents the first and most employed option in RFC – e.g. used in [6,16,38–40,52].
- **Extremely Randomized Trees**: in this option, less investigated than the previous one, a forest of Extremely Randomized Trees [15] is used. Within these trees, randomization is taken to the extreme: in every node, the rule is found by randomly selecting a feature, and then by selecting the threshold by uniformly sampling a value from the domain of the objects of the training set arrived at that node. Examples of RFC methods using this option can be found in [3,29,41].

Step 2. Distance computation. In this step, a distance between all pairs of objects to be clustered is computed through the learned RF. The different RFC schemes typically differ in the way this distance is computed, exploiting different concepts like path overlaps [39,52], probability masses [1,42] or axiomatic definitions of similarity [6]. However, in all cases, the idea is that a good measure of similarity between two objects can be defined by comparing the way they answer to the tests of the trees of the forest. The typical scheme is to define a similarity on a single tree, to be aggregated and transformed to distance at the Forest level. In more detail, the general formulation of a RF-distance, defined on a forest with T trees, is:

$$d^{RF}(\mathbf{x}, \mathbf{y}) = \sqrt{1 - \sum_{t=1}^{T} s^t(\mathbf{x}, \mathbf{y})} \tag{1}$$

where $s^t(\mathbf{x}, \mathbf{y})$ defines the tree-similarity between $\mathbf{x}$ and $\mathbf{y}$. Here we studied the following tree similarities, which lead to different RFC schemes:

- **RFC-Shi**. In this case the similarity between two objects $\mathbf{x}$ and $\mathbf{y}$ in a tree t of the forest is defined as:

$$s^t_{Shi}(\mathbf{x},\mathbf{y}) = \begin{cases} 1 & \text{if} \ell_t(\mathbf{x}) = \ell_t(\mathbf{y}) \\ 0 & \text{if} \ell_t(\mathbf{x}) \neq \ell_t(\mathbf{y}) \end{cases} \tag{2}$$

  This similarity, aggregated at the forest level, measures the number of times, over the total number of trees of the forest, the two objects reach the same leaf – i.e., they provide the same answers to all questions in the path. This represents the original measure proposed by [7], firstly used for clustering in [39].
- **RFC-Zhu**. This represents a tree-similarity introduced in [52][1]:

$$s^t_{Zhu}(\mathbf{x},\mathbf{y}) = \frac{\mathrm{depth}(\mathrm{lca}(\ell_t(\mathbf{x}), \ell_t(\mathbf{y}))}{\max\{\mathrm{depth}(\ell_t(\mathbf{x})), \mathrm{depth}(\ell_t(\mathbf{y}))\}} \tag{3}$$

  where $\mathrm{lca}(\ell_t(\mathbf{x}), \ell_t(\mathbf{y}))$ is the *least common ancestor* of $\ell_t(\mathbf{x})$ and $\ell_t(\mathbf{y})$. In this case, the idea is that the larger the overlap between the two paths $\mathbf{x}$ and $\mathbf{y}$ are following in the tree t, the larger their similarity.
- **RFC-RatioRF**. This represents the very recent RatioRF measure, introduced in [6] by following the axiomatic definition of similarity given by Tversky [44]. This approach has been shown to outperform all the alternatives in the clustering scenario [6], also in the presence of missing data [37]. In this case:

$$s^t_{RatioRF}(\mathbf{x},\mathbf{y}) = \frac{|P_t(\mathbf{x}) \capdot P_t(\mathbf{y})|}{|P_t(\mathbf{x}) \capdot P_t(\mathbf{y})| + |P_t(\mathbf{x}) \dotminus P_t(\mathbf{y})| + |P_t(\mathbf{y}) \dotminus P_t(\mathbf{x})|} \tag{4}$$

  where i) $P_t(\mathbf{x})$ denotes the set of tests on the path of $\mathbf{x}$, $P_t(\mathbf{x}) \capdot P_t(\mathbf{y})$ is the set of tests on which $\mathbf{x}$ and $\mathbf{y}$ agree, among the tests in $P_t(\mathbf{x}) \cup P_t(\mathbf{y})$, and ii) $P_t(\mathbf{x}) \dotminus P_t(\mathbf{y})$ (or, equivalently, $P_t(\mathbf{y}) \dotminus P_t(\mathbf{x})$) is the set of tests on the path of $\mathbf{x}$ ($\mathbf{y}$) on which $\mathbf{y}$ ($\mathbf{x}$) disagrees.

Step 3. Clustering. The final clustering is then obtained using any distance-based clustering algorithm, such as Spectral Clustering.

Let us conclude this section with a couple of very general considerations on the complexity of the RFC scheme. The computational load of the first step mainly depends on the size of the dataset, which, within the Negative Sampling, is also doubled; however, this represents a classic and widely studied issue with classification RFs, and very fast options have been proposed – e.g., [43]. When using ERTs, however, training is very fast, since no optimization is required. The second step (i.e. the computation of the RF-similarities) typically represents the bottleneck of RFC schemes, since the similarity should be determined for all pairs of objects to be clustered, and this quadratic complexity often makes the scaling

[1] Among the three variants proposed in [52], we employed here the second one, representing the best compromise between clustering accuracy and computational requirements, as also suggested in [3].

to very large datasets too computationally demanding. For what concerns the computation of a single similarity, the complexity mainly depends on the depth of the tree (since objects have to traverse all trees): however, experiments in [2] have shown that very short trees (with max depth 7 or 8) already permit to get excellent results. Given the distance, a standard distance-based method is used in the third step; the optimization of such methods is again a widely studied topic, with many optimized variants already present in the literature – e.g. [51].

3 The proposed approach

In this section, we introduce the variant of the RFC schemes described in the previous section, which permits the integration of partition-level constraints. Let us start with the definition of the problem, which follows [27]. Given a dataset $\mathbf{X} = \{\mathbf{x}_1, \cdots \mathbf{x}_N\}$ ($\mathbf{x}_i \in \Re^d$) to be clustered in K clusters, and a value $p \in (0, 1)$, let us define the p-partition level constraints as a set of labels from 1 to K, assigned by an expert, to pN objects of $\mathbf{X}$. Let us define as $\mathbf{X}^C \subset \mathbf{X}$ these objects, Y^C the corresponding labels, and $\mathbf{X}^U \subset \mathbf{X}$ the remaining objects. Clearly $|\mathbf{X}^C| = pN, |\mathbf{X}^U| = (1-p)N$, and $\mathbf{X}^U \cup \mathbf{X}^C = \mathbf{X}$. The goal is to cluster the objects in $\mathbf{X}$ in K clusters, starting from the representation $\mathbf{Z} = \{\mathbf{X}^U, \mathbf{X}^C, Y^C\}$.

In our constrained modification of the RFC, the main intuition is that constraints can be easily integrated into the first phase of the pipeline, namely in the learning phase. Our modification is extremely simple, but permits us to get promising results in the experiments. We start from two observations. The first is that the trained forest is not used for clustering, but simply to derive the distance measure for clustering: actually, as described in the previous section, the distance is computed by letting all objects in $\mathbf{X}$ traverse trees of the RF, subsequently comparing their paths. In this sense, the forest can also be trained with a subset of objects, or, in principle, also with objects from a hold-out set. Second observation: if we consider the basic version of the distance [39], where two objects are similar if they end in the same leaf in several trees of the forest, it is clear that a good forest is composed of trees in which objects belonging to different clusters follow different paths, ending in different leaves (by the way, this is true also for other RF measures). But this is actually what is looked for when building a RF with classification trees like CART [8], i.e. to get trees in which objects of different classes are grouped in different parts of the trees. Putting these two facts together, we have our simple approach to constrained RFC: to train a classification RF using only $\{\mathbf{X}^C, Y^C\}$, and to leave steps 2 and 3 of the RFC pipeline unchanged. The proposed approach is sketched in Fig. 1: in red the novel ingredients with respect to the standard RFC.

Let us stress that working on the learning stage of the RFC pipeline has different advantages: i) we can work on the weakest part of the RFC pipeline, for which a well-established option is still missing; ii) it seems reasonable that inserting the a priori information in the early stage of the pipeline would be more beneficial than inserting in later stages: actually, with better forests –

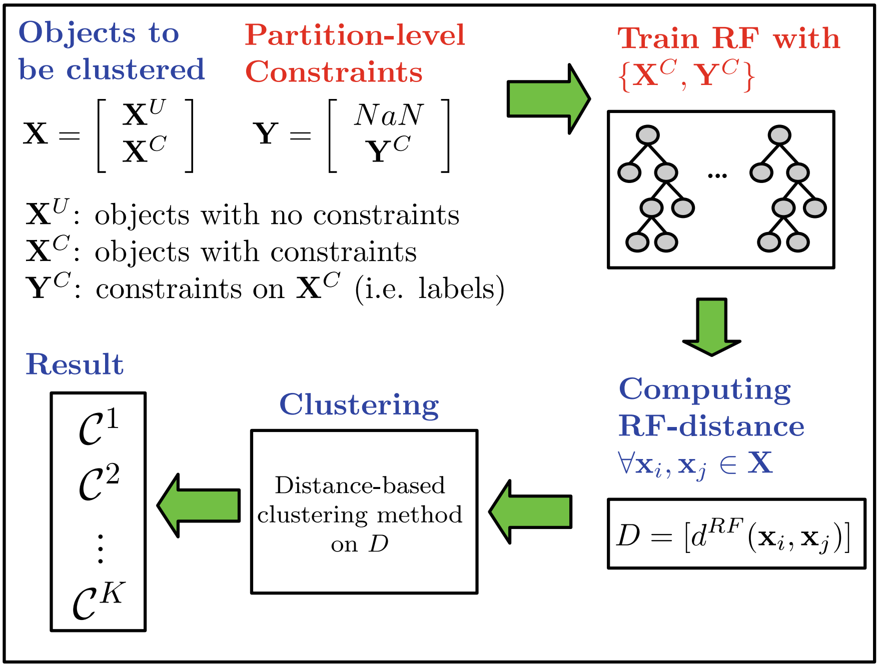

Objects to be clustered
Partition-level Constraints
Train RF with $\{\mathbf{X}^C, \mathbf{Y}^C\}$
$\mathbf{X} = \begin{bmatrix} \mathbf{X}^U \\ \mathbf{X}^C \end{bmatrix}$
$\mathbf{Y} = \begin{bmatrix} NaN \\ \mathbf{Y}^C \end{bmatrix}$
...
$\mathbf{X}^U$: objects with no constraints
$\mathbf{X}^C$: objects with constraints
$\mathbf{Y}^C$: constraints on $\mathbf{X}^C$ (i.e. labels)
Result
$\mathcal{C}^1$
$\mathcal{C}^2$
$\vdots$
$\mathcal{C}^K$
Clustering
Distance-based clustering method on D
Computing RF-distance
$\forall \mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}$
$D = [d^{RF}(\mathbf{x}_i, \mathbf{x}_j)]$

Fig. 1. The proposed approach. In red the modification introduced with respect to the standard RFC scheme.

built exploiting the constraints – we would have better distances, which would lead to a better clustering result; iii) the computational overhead is very limited, and related only to training; indeed, with our approach, we are actually *reducing* the computational overhead with respect to the unconstrained case of "Negative Sampling", since we are training a classification forest with a reduced number of points (only pN objects, whereas for Negative Sampling we need $2N$ objects, since typically the generated negative class has the same cardinality of the dataset).

4 Experimental evaluation

In this section, the proposed approach is evaluated and compared with some alternatives. The evaluation is based on 18 datasets, which are listed in Tab. 1. The full description of these datasets, together with the source and the preprocessing, is provided in the Section B of the supplementary material. We considered datasets of both small and moderate size (these latter marked with an "(*)" in Tab. 1), which represents the situation where tree-based approaches have been shown to be most useful [17]. As done in most of the clustering experiments, the evaluation is done by comparing the cluster assignment with the true labels. In particular, we used the standard Adjusted Rand Index (ARI – [20]), which employs a contingency table between the obtained grouping and the true one to quantify the quality of the result (the higher this index, the better the clustering), also performing a correction to consider the chance of the formation of the clusters.

Table 1. Datasets used for evaluation. The set of datasets denoted with "(*)" represents the "moderate size" datasets.

Name	Objects	Features	Clusters
Iris	150	4	3
Glass	214	9	4
Ecoli	336	7	8
Seeds	210	7	3
Cryotherapy	90	6	2
AutoMpg	398	6	2
Imox	192	8	4
StoneFlakes	70	8	3
Pima	768	8	2
Nose	358	128	5
Microarray	90	100	5
Flickr(*)	1000	87	5
Isolet(*)	1200	617	4
EEGEyeState(*)	1280	14	2
UAVIntDet-1(*)	1100	54	2
UAVIntDet-2(*)	1100	54	2
UAVIntDet-3(*)	1100	54	2
UAVIntDet-4(*)	1100	18	2

Concerning RFC approaches, we considered the three different schemes described in Section 2.1, namely RFC-Shi, RFC-Zhu, and RFC-RatioRF, evaluating their extension to the constrained case. In all experiments, we used forests with 100 trees, sampling 50% of features in each node, and building each tree using a random 50% of the training set – for datasets of moderate size, we followed suggestions in [6], using 128 random objects for each tree. Each tree was built until its maximum depth; finally, for classification trees we used the Gini Criterion. Given the distance, the final clustering is obtained with Spectral Clustering using the Ng-Jordan-Weiss normalized version [46], and repeating the inner k-means 20 times, as done in most of the RFC schemes [3,6,52]. The Matlab code of the proposed approach is available at https://profs.scienze.univr.it/~bicego/code.html.

Constrained clustering experiments were performed following the classic protocol for partition level constraints [27]: for a given level of supervision p, we

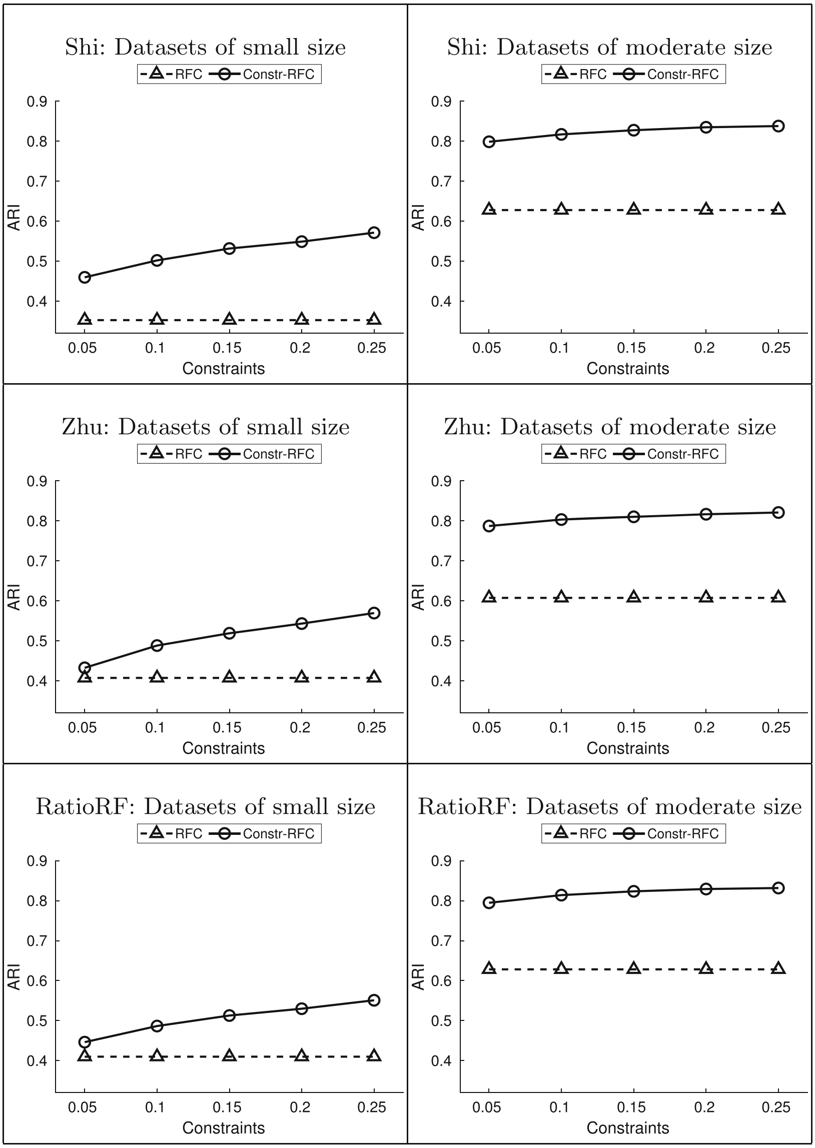


Fig. 2. Comparison between unconstrained and constrained strategies for different RF Clustering schemes: first row: average over datasets of small size, second row: average over datasets of moderate size.

randomly sampled pN objects from the dataset, considering their labels as the constraints. We investigated different levels of p from 0.05 to 0.25, with step 0.05, and we repeated the whole pipeline 100 times. In each of the 100 runs, the constraints set was kept fixed and used as input to all the constrained versions.

In the following, after reporting the comparison of the constrained versions of RFC with the corresponding unconstrained versions, we compare the proposed schemes with some literature alternatives in terms of both the ARI index and the Negative Effects of Constraints (NEC) ratio [12].

4.1 Comparison with the unconstrained RFC

The comparison with the unconstrained RFC is reported in Fig. 2. Due to lack of space, we reported here only the averages over the datasets of small size (first row) and of moderate size (second row), leaving the results dataset per dataset to the Section C of the supplementary material. In the figure, the three columns represent the results with the RFC-Shi, RFC-Zhu, and RFC-RatioRF, respectively. In every plot, we report the average of the ARI criterion (the higher, the better) over the 100 repetitions and over the group of datasets.

From the plots, it is evident that the exploitation of the constraints is, on average, very beneficial for the RFC schemes, especially for datasets of moderate size. To assess the statistical significance of the comparison reported in the Figure, we performed a paired t-test, comparing, for each RF scheme, and each level of constraints, all the repetitions over the different groups of datasets of the constrained version with the corresponding unconstrained scheme. All tests reported a statistically significant difference, according to a level of 0.05.

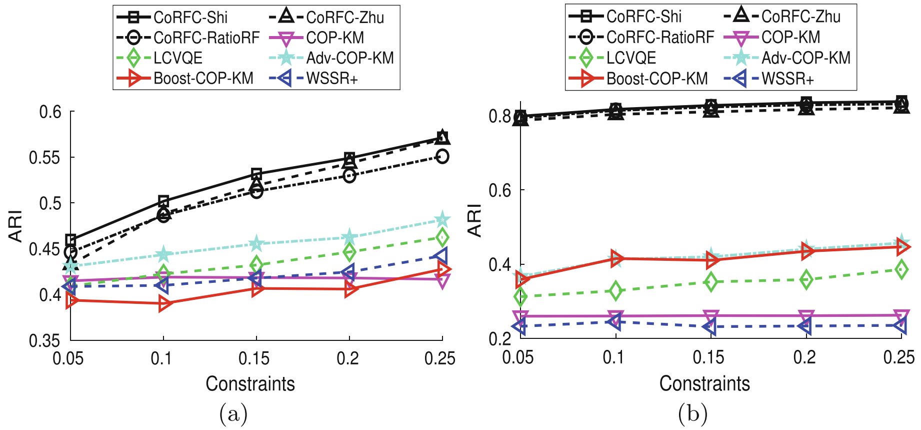


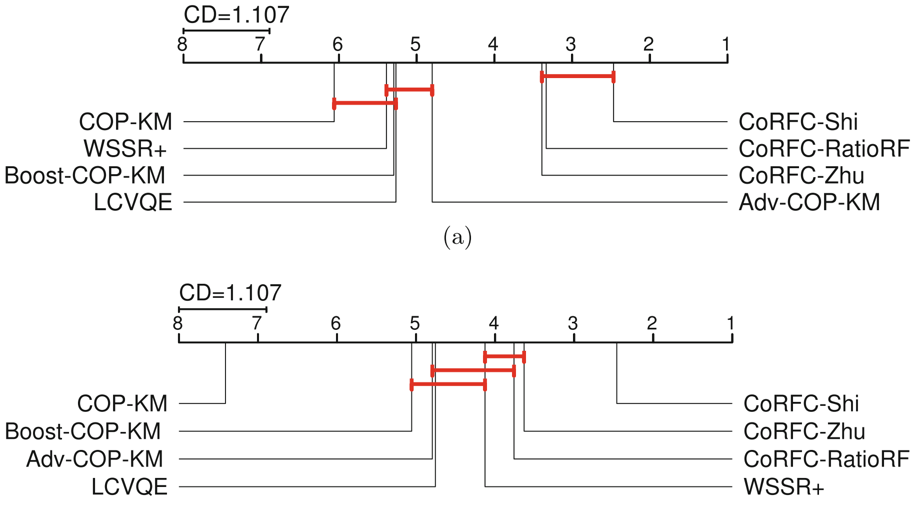
Fig. 3. Comparison between the proposed approach and some alternatives in terms of ARI for different levels of constraints: a) average over small datasets, b) average over datasets of moderate size.

4.2 Comparison with other standard constrained clustering methods

To get an idea of the potentialities of the proposed approach, we compare our constrained RFC schemes (RFC-Shi, RFC-Zhu, and RFC-RatioRF) with a few standard literature alternatives. In particular, we first considered some standard and widely used algorithms: two versions of the COP-Kmeans algorithm, namely **COP-KM**, the original extension of [48] and **Adv-COP-KM**, a more advanced one proposed in [30]); another standard approach, i.e., the **LCVQE** method of [32]; then we considered some more advanced and recent approaches, having some far commonalities with our approach: i) **Boost-COP-KM**, a recent algorithm based on ensemble learning (as RF), introduced in [30]; ii) **WSSR+**, a very recent constrained spectral clustering algorithm, proposed in [33], which learns a similarity measure from the constraints (as in RFC), to be clustered with spectral clustering (which we used in our experiments). We evaluate all these methods with the same protocol, i.e., computing the ARI values; to have a fair comparison, in all repetitions, the starting point was the same: the dataset and the same constraints used for our constrained RFC methods. The description of these methods, together with the implementation details, is provided in the Section D of the supplementary material. Comparative results are shown in Figure 3. Also in this case we show results averaged over datasets of small (part (a) of fig. 3) and moderate (part (b)) size, whereas the complete results, dataset per dataset, are reported in section D of the supplementary material.

It is interesting to observe that the constrained RFC-schems, on average, outperform alternatives, especially when working with large datasets and with a large level of supervision. This is somehow expected since the labels of the constraints are used to train the initial forest, which is then used to derive the distance used for clustering: the larger this Set, the better the forest is trained. In order to assess the statistical significance of the comparison, we employ a Friedman test followed by a post-hoc Nemenyi test. The Friedman test represents a common non-parametric test, employed when comparing more than two approaches [13]. The test is based on the ranking of the compared approaches, and permits the assessment of the presence of a significant difference among them; if the null hypothesis is rejected, a Nemenyi test is applied, assessing via a critical value which pairs of methods are different with a statistical significance. The critical value represents the required minimum difference between the ranks of the approaches. The final output of the whole procedure can be represented via a critical diagram [13], which shows the ranks (from highest to lowest): a line connecting two or more methods indicates that there is not a statistically significant difference between them. We applied this procedure to the comparison between the constrained RFC schemes and the literature alternatives, using as significance level $\alpha = 0.05$. The obtained critical diagram is shown in Fig. 4(a), from which it can be observed that the three constrained RFC schemes outperform all the alternatives with a statistical significance.

To have a better understanding of the proposed approach, also in comparison with literature alternatives, we provide here an analysis of the so-called negative effects of constraints (NEC). As firstly hypothesized by [12], it may be the case that the introduction of constraints does not lead to an improvement of the clustering, reducing the actual accuracy of the clustering results. Following [12], we quantified such effect by counting how many times, among the 100 repetitions,



(a)

(b)

Fig. 4. Statistical analysis of the comparison between the proposed approach and some alternatives: a) critical diagram for ARI, b) critical diagram for NEC.

using the constraints does not improve the clustering accuracy with respect to not using them. This measure, which we refer to as *NEC*, is aimed at measuring how effective are the different constrained clustering methods in exploiting the constraints so that there is an increase in the clustering accuracies.

The results are shown in Figure 5, for our RFC schemes and for the literature alternatives. Please note that the lower this measure, the better the exploitation of the constraints. Also in this case we show results averaged over datasets of small (part (a) of fig. 5) and moderate (part (b)) size, whereas detailed results dataset per dataset are reported in section D of the supplementary material. Also in this case, we can observe that the proposed approach compares reasonably well with literature alternatives, especially for large levels of supervision and for datasets of small size.

Also in this case we analyzed the statistical significance of the results via a critical diagram, shown in part (b) of Fig. 4. CoRFC-Shi is better than all alternatives with a statistical significance, whereas CoRFC-RatioRF and CoRFC-Zhu are equivalent to the very recent WSSR+. It is interesting to note that the best constrained variant is obtained when using the oldest RFC scheme (with the Shi-Breiman distance); in the unconstrained case, RatioRF and Zhu outperform Shi, as also confirmed in other works [6,52]. Probably, when a better RF, trained by exploiting the constraints, is provided, the simple Shi-Breiman distance with the 0/1 mechanism is very adequate, and more sophisticated strategies, like RatioRF or Zhu, may introduce overtraining in the whole process. On the contrary, without constraints, worse RFs are derived, and more clever strategies should be used. This is confirmed by the theoretical characterization of [5], which shows

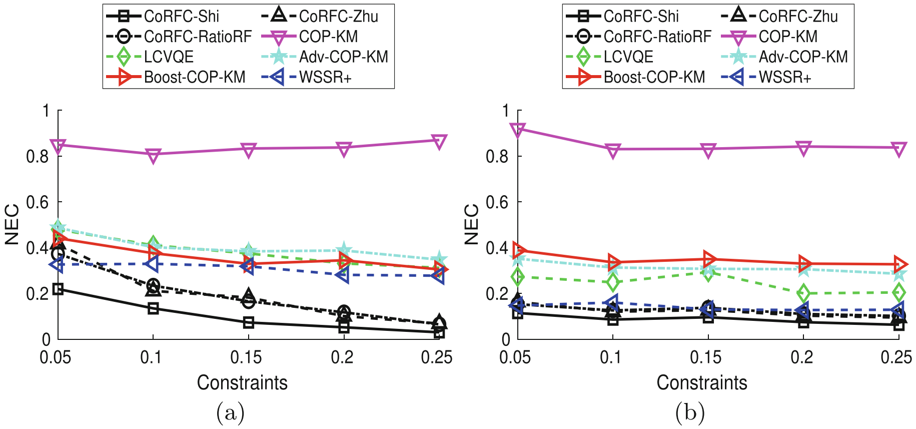


Fig. 5. Comparison between the proposed approach and some alternatives in terms of NEC ratio for different levels of constraints: a) average over small datasets, b) average over datasets of moderate size.

that, when starting from Forest built with completely random trees (Extremely Randomized Trees), the RatioRF distance has a better theoretical behaviour (i.e., better bounds) than the Shi-Breiman distance.

5 Conclusions

In this paper, we presented a simple approach to extend RFC schemes to work with partition-level constraints. The approach is very straightforward, simply modifying the first (and weakest) step of the RFC scheme. Obtained results are promising, showing that i) constraints can be easily and fruitfully integrated into different RFC schemes and ii) the obtained constrained RFC schemes represent a valid alternative to standard as well advanced constrained clustering approaches. Future works will include in the experimental evaluation larger real-world datasets and applications, in order to show the practical relevance of the proposed method.

References

1. Aryal, S., Ting, K., 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 (2020)
2. Bicego, M., Cicalese, F., Mensi, A.: RatioRF: A novel measure for random forest clustering based on the tversky's ratio model. IEEE Tr. on Knowledge and Data Engineering **35**(1), 830–841 (2023)
3. Bicego, M., Escolano, F.: On learning random forests for random forest-clustering. In: Proc. Int. Conf. on Pattern Recognition. pp. 3451–3458. IEEE (2021)

4. Bicego, M.: K-random forests: a k-means style algorithm for random forest clustering. In: Proc. Int. Joint Conf. on Neural Networks. pp. 1–8. IEEE (2019)
5. Bicego, M., Cicalese, F.: On the good behaviour of extremely randomized trees in random forest-distance computation. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 645–660. Springer (2023)
6. 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. **35**(1), 830–841 (2023)
7. Breiman, L.: Random forests. Mach. Learn. **45**, 5–32 (2001)
8. Breiman, L., Friedman, J., Olshen, R., Stone, C.: Classification and Regression Trees. Wadsworth (1984)
9. Covoes, T.F., Hruschka, E.R., Ghosh, J.: A study of k-means-based algorithms for constrained clustering. Intelligent Data Analysis **17**(3), 485–505 (2013)
10. Davidson, I., Basu, S.: A survey of clustering with instance level constraints. ACM Trans. on Knowledge Discovery from data **1**(1-41), 2–42 (2007)
11. Davidson, I., Ravi, S.: Agglomerative hierarchical clustering with constraints: Theoretical and empirical results. In: Proc. Europ. Conf. on Principles of Data Mining and Knowledge Discovery. pp. 59–70 (2005)
12. Davidson, I., Wagstaff, K.L., Basu, S.: Measuring constraint-set utility for partitional clustering algorithms. In: Proc. Europ. Conf. on Principles of Data Mining and Knowledge Discovery. pp. 115–126 (2006)
13. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. The Journal of Machine Learning Research **7**, 1–30 (2006)
14. Gançarski, P., Dao, T.B.H., Crémilleux, B., Forestier, G., Lampert, T.: Constrained clustering: Current and new trends. A Guided Tour of Artificial Intelligence Research: Volume II: AI Algorithms pp. 447–484 (2020)
15. Geurts, P., Ernst, D., Wehenkel, L.: Extremely randomized trees. Mach. Learn. **63**(1), 3–42 (2006)
16. Gray, K.R., Aljabar, P., Heckemann, R.A., Hammers, A., Rueckert, D.: Random forest-based similarity measures for multi-modal classification of alzheimer's disease. Neuroimage **65**, 167–175 (2013)
17. Grinsztajn, L., Oyallon, E., Varoquaux, G.: Why do tree-based models still outperform deep learning on typical tabular data? Adv. Neural. Inf. Process. Syst. **35**, 507–520 (2022)
18. Grossi, V., Romei, A., Turini, F.: Survey on using constraints in data mining. Data Min. Knowl. Disc. **31**, 424–464 (2017)
19. Hong, Y., Kwong, S.: Learning assignment order of instances for the constrained k-means clustering algorithm. IEEE Trans. on Systems, Man, and Cybernetics, Part B (Cybernetics) **39**(2), 568–574 (2008)
20. Hubert, L., Arabie, P.: Comparing partitions. Journal of Classification pp. 193–218 (1985)
21. Jain, A.K.: Data clustering: 50 years beyond k-means. Pattern Recogn. Lett. **31**(8), 651–666 (2010)
22. Jain, A.K., Dubes, R.C.: Algorithms for clustering data. Prentice-Hall, Inc. (1988)
23. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM computing surveys (CSUR) **31**(3), 264–323 (1999)
24. Lelis, L., Sander, J.: Semi-supervised density-based clustering. In: Proc. Int. Conf. on Data Mining. pp. 842–847 (2009)
25. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation forest. In: Proc. Int. Conf. on Data Mining. pp. 413–422 (2008)

26. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation-based anomaly detection. ACM Trans. on Knowledge Discovery from Data (TKDD) **6**(1), 1–39 (2012)
27. Liu, H., Fu, Y.: Clustering with partition level side information. In: Proc. Int. Conf. on Data Mining. pp. 877–882 (2015)
28. Lloyd, S.: Least squares quantization in pcm. IEEE Trans. on information theory **28**, 129–137 (1982)
29. Moosmann, F., Triggs, B., Jurie, F.: Fast discriminative visual codebooks using randomized clustering forests. In: Advances in neural information processing systems. pp. 985–992 (2006)
30. Okabe, M., Yamada, S.: Clustering using boosted constrained k-means algorithm. Frontiers in Robotics and AI **5**, 18 (2018)
31. Pei, Y., Fern, X.Z., Tjahja, T.V., Rosales, R.: Comparing clustering with pairwise and relative constraints: A unified framework. ACM Trans. on Knowledge Discovery from Data **11**(2), 1–26 (2016)
32. Pelleg, D., Baras, D.: K-means with large and noisy constraint sets. In: Proc. European Conference on Machine Learning. pp. 674–682 (2007)
33. Peng, H., Pavlidis, N.G.: Weighted sparse simplex representation: a unified framework for subspace clustering, constrained clustering, and active learning. Data Min. Knowl. Disc. **36**(3), 958–986 (2022)
34. Perbet, F., Stenger, B., Maki, A.: Random forest clustering and application to video segmentation. In: Proc. of British Machine Vision Conference. pp. 1–10 (2009)
35. Qian, P., Jiang, Y., Wang, S., Su, K.H., Wang, J., Hu, L., Muzic, R.F.: Affinity and penalty jointly constrained spectral clustering with all-compatibility, flexibility, and robustness. IEEE Trans. on neural networks and learning systems **28**(5), 1123–1138 (2016)
36. Rangapuram, S.S., Hein, M.: Constrained 1-spectral clustering. In: Artificial Intelligence and Statistics. pp. 1143–1151. PMLR (2012)
37. Raniero, M., Bicego, M., Cicalese, F.: Distance-based random forest clustering with missing data. In: Proc. Int. Conf. on Image Analysis and Processing. pp. 121–132. Springer (2022)
38. Rennard, S.I., Locantore, N., Delafont, B., Tal-Singer, R., Silverman, E.K., Vestbo, J., Miller, B.E., Bakke, P., Celli, B., Calverley, P.M., et al.: Identification of five chronic obstructive pulmonary disease subgroups with different prognoses in the eclipse cohort using cluster analysis. Ann. Am. Thorac. Soc. **12**(3), 303–312 (2015)
39. Shi, T., Horvath, S.: Unsupervised learning with random forest predictors. J. Comput. Graph. Stat. **15**(1), 118–138 (2006)
40. Shi, T., Seligson, D., Belldegrun, A., Palotie, A., Horvath, S.: Tumor classification by tissue microarray profiling: Random forest clustering applied to renal cell carcinoma. Mod. Pathol. **18**, 547–557 (2005)
41. Shotton, J., Johnson, M., Cipolla, R.: Semantic texton forests for image categorization and segmentation. In: Proc. Int. Conf. on Computer Vision and Pattern Recognition. pp. 1–8 (2008)
42. 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: Proc. Int. Conf. on Knowledge Discovery and Data mining. pp. 1205–1214 (2016)
43. Tiwari, M., Kang, R., Lee, J., Piech, C., Shomorony, I., Thrun, S., Zhang, M.J.: Mabsplit: Faster forest training using multi-armed bandits. Adv. Neural. Inf. Process. Syst. **35**, 1223–1237 (2022)
44. Tversky, A.: Features of similarity. Psychol. Rev. **84**(4), 327 (1977)

45. Van Engelen, J.E., Hoos, H.H.: A survey on semi-supervised learning. Mach. Learn. **109**(2), 373–440 (2020)
46. von Luxburg, U.: A tutorial on spectral clustering. Stat. Comput. **17**(4), 395–416 (2007)
47. Vouros, A., Vasilaki, E.: A semi-supervised sparse k-means algorithm. Pattern Recogn. Lett. **142**, 65–71 (2021)
48. Wagstaff, K., Cardie, C., Rogers, S., Schrödl, S.: Constrained k-means clustering with background knowledge. In: Proc. Int. Conf. on Machine Learning. vol. 1, pp. 577–584 (2001)
49. Wang, X., Qian, B., Davidson, I.: On constrained spectral clustering and its applications. Data Min. Knowl. Disc. **28**, 1–30 (2014)
50. Yan, D., Chen, A., Jordan, M.: Cluster forests. Computational Statistics & Data Analysis **66**, 178–192 (2013)
51. Zhu, W., Nie, F., Li, X.: Fast spectral clustering with efficient large graph construction. In: 2017 IEEE international conference on acoustics, speech and signal processing (ICASSP). pp. 2492–2496. IEEE (2017)
52. Zhu, X., Loy, C., Gong, S.: Constructing robust affinity graphs for spectral clustering. In: Proc. Int. Conf. on Computer Vision and Pattern Recognition. pp. 1450–1457 (2014)