Extremely randomized trees for clustering complex data

Forêts d’arbres aleatoires pour le clustering de données complexes

AFIA-SFC: Recent advances on unsupervised learning

Kevin Dalleau*, Miguel Couceiro, Malika Smail-Tabbone
21 of September, 2021

Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
*Project RHU Fight-HF: https://anr.fr/ProjetIA-15-RHUS-0004
Outline

1. Unsupervised Extremely randomized Trees
2. Empirical evaluation
3. Application: Graph clustering
4. Graph-Trees (GT)
5. Experiments
6. Discussion
Unsupervised classification, a.k.a clustering:

- **Goal**: find homogeneous groups of unlabeled instances.
- Active field, with multiple types of approaches: centroid-based (k-means), density-based (DBSCAN), hierarchical clustering (HAC), etc.

Many algorithms rely on a distance metric between instances

- Large number of distances in the literature\(^1\)

---

Motivations and goals

**Motivation I.** Set of relevant and available distances depends on:

- characteristics of the data: continuous, categorical, ordinal, etc.
- chosen algorithm

**Goal:** Similarity measure agnostic to data types.

**Motivation II.** Preprocessing burden in many practical cases:

- scaling issues
- correlation between variables
- missing values

**Goal:** Reduce the preprocessing burden.
Motivations and goals

**Motivation I.** Set of relevant and available distances depends on:

- characteristics of the data: continuous, categorical, ordinal, etc.
- chosen algorithm

**Goal:** Similarity measure agnostic to data types.

**Motivation II.** Preprocessing burden in many practical cases:

- scaling issues
- correlation between variables
- missing values

**Goal:** Reduce the preprocessing burden.
Unsupervised Extremely randomized Trees
Shi et al.\textsuperscript{2}: method to compute a \textit{similarity} in unsupervised settings.

- Method based on RF: Unsupervised Random Forest (URF).
- RF: popular tree-based algorithm, extensively used.
- Ensemble method, combining decision trees in order to obtain better results in supervised learning tasks.

Idea: once the forest constructed, run the training data down each tree.

1. All instances in the same leaf are considered similar.
2. Similarity measure: if two instances \( i \) and \( j \) are in the same leaf of a tree, the overall similarity between the two instances is increased by one.

Normalization: all values lie in [0, 1].
How to build a decision-tree in an unsupervised setting?
Answer: generation of synthetic instances.
Two procedures to generate synthetic instances are presented in Shi et al.\textsuperscript{3}

- **addCl1**: the synthetic instances are obtained by a random sampling from the observed distributions of variables.
- **addCl2**: random sampling in the hyper rectangle containing the observed instances.

### addCl1: an example

<table>
<thead>
<tr>
<th>Instance</th>
<th>Feature #1</th>
<th>Feature #2</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5.1</td>
<td>3.5</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>7.0</td>
<td>3.2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>6.4</td>
<td>2.8</td>
<td>1</td>
</tr>
</tbody>
</table>
### addCl1: an example

<table>
<thead>
<tr>
<th>Instance</th>
<th>Feature #1</th>
<th>Feature #2</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5.1</td>
<td>3.5</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>7.0</td>
<td>3.2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>6.4</td>
<td>2.8</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>5.1</td>
<td>3.2</td>
<td>0</td>
</tr>
</tbody>
</table>
**addCl1: an example**

<table>
<thead>
<tr>
<th>Instance</th>
<th>Feature #1</th>
<th>Feature #2</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5.1</td>
<td>3.5</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>7.0</td>
<td>3.2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>6.4</td>
<td>2.8</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>5.1</td>
<td>3.2</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>6.4</td>
<td>3.5</td>
<td>0</td>
</tr>
</tbody>
</table>
### addCl1: an example

<table>
<thead>
<tr>
<th>Instance</th>
<th>Feature #1</th>
<th>Feature #2</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5.1</td>
<td>3.5</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>7.0</td>
<td>3.2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>6.4</td>
<td>2.8</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>5.1</td>
<td>3.2</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>6.4</td>
<td>3.5</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>5.1</td>
<td>2.8</td>
<td>0</td>
</tr>
</tbody>
</table>
addCl2: an example

<table>
<thead>
<tr>
<th>Instance</th>
<th>Feature #1</th>
<th>Feature #2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5.1</td>
<td>3.5</td>
</tr>
<tr>
<td>2</td>
<td>7.0</td>
<td>3.2</td>
</tr>
<tr>
<td>3</td>
<td>6.4</td>
<td>2.8</td>
</tr>
</tbody>
</table>

Feature #1: [5.1, 7.0]
Feature #2: [2.8, 3.5]

<table>
<thead>
<tr>
<th>Instance</th>
<th>Feature #1</th>
<th>Feature #2</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5.1</td>
<td>3.5</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>7.0</td>
<td>3.2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>6.4</td>
<td>2.8</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>5.5</td>
<td>2.9</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>6.7</td>
<td>3.1</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>5.9</td>
<td>3.4</td>
<td>0</td>
</tr>
</tbody>
</table>
Successfully used in fields such as biology or image processing.

However: The method presents some limitations:

- The generation step is not computationally efficient.
- Bias induced by the generated instances.
- It is necessary to construct many forests with different synthetic instances and average their results.
P. Geurts et al.: Extremely Randomized Trees (ET)\textsuperscript{4}

- Very similar to RF.
- Another randomization: split threshold selected partially/totally at random

Two important parameters:

1. $K$, the number of attributes to be randomly selected at each node.
2. $n_{\text{min}}$ (smoothing strength), the minimum instance size for node split.

Following the tracks of Shi et al. of URF, we propose to use ET.

- Novel approach where the generation of synthetic cases is not necessary.
- **addCl3**: a method to generate synthetic labels and associate them to the observed instances.

**Result**: Unsupervised Extremely randomized Trees (UET)\textsuperscript{5}

**Randomization**: numerical/ordinal or categorical variables

\textsuperscript{5}K. Dalleau, M. Couceiro, M. Smaïl-Tabbone: Unsupervised Extremely Randomized Trees. PAKDD (3) 2018: 478-489
### addCl3: example

<table>
<thead>
<tr>
<th>Instance</th>
<th>Feature #1</th>
<th>Feature #2</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5.1</td>
<td>3.5</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>7.0</td>
<td>3.2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>6.4</td>
<td>2.8</td>
<td>0</td>
</tr>
</tbody>
</table>

**Table 1: addCl3**
Algorithme 1 : Unsupervised Extremely Randomized Trees

Données : Observations $O$

Résultat : Similarity matrix $S$

1. $K, n_{\text{min}}, n_{\text{trees}}$

2. $D \leftarrow \text{addCl3}(O)$;

3. $T \leftarrow \text{Build\_an\_extra\_tree\_ensemble}(D)$  // Here $K = 1$;

4. $S = 0_{n_{\text{obs}}, n_{\text{obs}}}$  // Initialization of a zero matrix of size $n_{\text{obs}}$;

5. pour $d_i \in D$ faire

6.     pour $d_j \in D$ faire

7.         $S_{i,j} = \text{number of times the instances } d_i \text{ and } d_j \text{ fall in the same leaf node in each tree of } T = \{t_1, t_2, \ldots, t_M\}$;

8.     fin

9. fin

10. $S_{i,j} = \frac{S_{i,j}}{M}$;
Empirical evaluation
The procedure goes as follows:

1. A similarity matrix is constructed using UET.
2. This similarity matrix is transformed into a dissimilarity matrix using\textsuperscript{6}:

\[ DIS_{ij} = \sqrt{1 - SIM_{ij}} \]

3. An hierarchical agglomerative clustering (with average linkage) is performed using this distance matrix, with the relevant number of clusters for the labeled dataset.

This procedure is ran 10 times: For each clustering, Adjusted Rand Indices (ARI) are computed, and are compared using the Kruskal-Wallis test.

First we evaluate the influence of the parameters on the results of UET:

- The number of trees (averaging strength) $n_{trees}$.
- The minimum number of instances to split $n_{min}$.

<table>
<thead>
<tr>
<th>Dataset</th>
<th># instances</th>
<th># features</th>
<th># labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>150</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Wine</td>
<td>178</td>
<td>13</td>
<td>3</td>
</tr>
<tr>
<td>Wisconsin</td>
<td>699</td>
<td>9</td>
<td>2</td>
</tr>
</tbody>
</table>

**Table 2**: Properties of used datasets
Observations:

- $n_{trees}$: no significant diff. in ARI for $n_{trees} > 50$ ($p > 0.1$ for all datasets)
- $n_{min}$: values between 20% and 30% of the number of instances seems to lead to the best results.
- UET fails with small values of $n_{min}$

Explanation: larger values of $n_{min}$ are necessary with noisy data

---

Cluster discrimination

**Question:** is UET able to discriminate instances from different clusters?

- 3 generated datasets of 1000 instances: two without any cluster structure (NoC4 and NoC5), and one with a cluster structure (C4)
- 20 runs of UET: 20 similarity matrices
- Comparison of the mean difference $\bar{\Delta}$ between
  1. the mean intracluster similarity $\mu_{\text{intra}}$
  2. the mean intercluster similarity $\mu_{\text{inter}}$

<table>
<thead>
<tr>
<th>Dataset</th>
<th>$\bar{\Delta}$</th>
<th>$\sigma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>NoC4</td>
<td>0.00042</td>
<td>0.00003</td>
</tr>
<tr>
<td>NoC50</td>
<td>0.00007</td>
<td>0.00003</td>
</tr>
<tr>
<td>C4</td>
<td>0.68417</td>
<td>0.00341</td>
</tr>
</tbody>
</table>

**Table 3:** Mean difference between intercluster and intracluster similarities in different settings, on synthetic datasets.
Question: is UET able to discriminate instances from different clusters?

- 3 generated datasets of 1000 instances: two without any cluster structure (\textit{NoC4} and \textit{NoC5}), and one with a cluster structure (\textit{C4})
- 20 runs of UET: 20 similarity matrices
- Comparison of the mean difference $\bar{\Delta}$ between
  1. the mean intracluster similarity $\mu_{\text{intra}}$
  2. the mean intercluster similarity $\mu_{\text{inter}}$

<table>
<thead>
<tr>
<th>Dataset</th>
<th>$\bar{\Delta}$</th>
<th>$\sigma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>NoC4</td>
<td>0.00042</td>
<td>0.00003</td>
</tr>
<tr>
<td>NoC50</td>
<td>0.00007</td>
<td>0.00003</td>
</tr>
<tr>
<td>C4</td>
<td>0.68417</td>
<td>0.00341</td>
</tr>
</tbody>
</table>

\textbf{Table 3:} Mean difference between intercluster and intracluster similarities in different settings, on synthetic datasets.
We then assessed UET on benchmark datasets:

- Comparison of Normalized Mutual Information (NMI) scores with the values presented in H. Elghazel et al.\(^8\).
- Comparison of ARI obtained with UET and URF.
- UET computed with \( n_{trees} = 50 \), \( n_{min} = \lceil \frac{n_{instances}}{3} \rceil \).

\(^8\)H. Elghazel and A. Aussem, Feature selection for unsupervised learning using random cluster ensembles, Data Mining, 2010
### Datasets used for comparison

<table>
<thead>
<tr>
<th>Dataset</th>
<th># instances</th>
<th># features</th>
<th># labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>150</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Wine</td>
<td>178</td>
<td>13</td>
<td>3</td>
</tr>
<tr>
<td>Wisconsin</td>
<td>699</td>
<td>9</td>
<td>2</td>
</tr>
<tr>
<td>Lung</td>
<td>32</td>
<td>56</td>
<td>3</td>
</tr>
<tr>
<td>Breast tissue</td>
<td>106</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>Isolet</td>
<td>1559</td>
<td>617</td>
<td>26</td>
</tr>
<tr>
<td>Pima</td>
<td>768</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>Parkinson</td>
<td>195</td>
<td>22</td>
<td>2</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>351</td>
<td>34</td>
<td>2</td>
</tr>
<tr>
<td>Segmentation</td>
<td>2310</td>
<td>19</td>
<td>7</td>
</tr>
</tbody>
</table>

**Table 4**: Datasets used for comparison
Comparative evaluation with the results from Elghazel et al. ⁹.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>UET - NMI</th>
<th>Literature - NMI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wisconsin</td>
<td>78.33 ± 3.25</td>
<td>73.61 ± 0.00</td>
</tr>
<tr>
<td>Lung</td>
<td>29.98 ± 6.17</td>
<td>22.51 ± 5.58</td>
</tr>
<tr>
<td>Breast tissue</td>
<td>74.48 ± 2.92</td>
<td>51.18 ± 1.38</td>
</tr>
<tr>
<td>Isolet</td>
<td>61.22 ± 1.47</td>
<td>69.83 ± 1.74</td>
</tr>
<tr>
<td>Parkinson</td>
<td>25.50 ± 6.14</td>
<td>23.35 ± 0.19</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>13.47 ± 1.11</td>
<td>12.62 ± 2.37</td>
</tr>
<tr>
<td>Segmentation</td>
<td>69.62 ± 2.14</td>
<td>60.73 ± 1.71</td>
</tr>
</tbody>
</table>

⁹Feature selection for unsupervised learning using random cluster ensembles, Data Mining (ICDM), 2010
## Benchmarking

<table>
<thead>
<tr>
<th>Dataset</th>
<th>UET (ARI - Time (s))</th>
<th>URF (ARI - Time (s))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wisconsin</td>
<td>87.13 - 128.42 s</td>
<td>82.92 - 968.71 s</td>
</tr>
<tr>
<td>Lung</td>
<td>23.24 - 5.23 s</td>
<td>6.52 - 86.93 s</td>
</tr>
<tr>
<td>Breast tissue</td>
<td>58.85 - 9.15 s</td>
<td>39.05 - 99.40 s</td>
</tr>
<tr>
<td>Isolet</td>
<td>28.04 - 692.82 s</td>
<td>* - * s</td>
</tr>
<tr>
<td>Parkinson</td>
<td>25.21 - 16.27 s</td>
<td>12.68 - 279.30 s</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>6.04 - 39.13 s</td>
<td>7.28 - 727.30 s</td>
</tr>
</tbody>
</table>

**Table 5**: Comparative evaluation between URF and UET
What about the preprocessing tasks we mentioned earlier?
Datasets

Here we used two datasets freely available in *Scikit-learn*

- **blob500**: 500 instances, 5 features and 3 blob shaped clusters
- **moon500**: 500 instances, 2 features, 2 moon-shaped clusters

**Question**: robustness to variable transformations & correlations
Robustness to monotone transformation of variables

Why:\textsuperscript{10}

- Robustness to change in scales
- Robustness to outliers

Procedure:

- computation of $\bar{\Delta}$ on the original data
- multiplication or addition of $n$ column of the dataset by a scalar (drawn from $\mathcal{U}(2, 100)$)
- computation of new $\bar{\Delta}$

Robustness to monotone transformation of variables

<table>
<thead>
<tr>
<th>Operation</th>
<th>Number of variables</th>
<th>$\tilde{\Delta}$</th>
<th>$\sigma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiplication</td>
<td>0</td>
<td>0.2981</td>
<td>0.0044</td>
</tr>
<tr>
<td>Multiplication</td>
<td>1</td>
<td>0.2991</td>
<td>0.0029</td>
</tr>
<tr>
<td>Multiplication</td>
<td>2</td>
<td>0.2992</td>
<td>0.0036</td>
</tr>
<tr>
<td>Addition</td>
<td>0</td>
<td>0.2987</td>
<td>0.0037</td>
</tr>
<tr>
<td>Addition</td>
<td>1</td>
<td>0.2976</td>
<td>0.0045</td>
</tr>
<tr>
<td>Addition</td>
<td>2</td>
<td>0.2970</td>
<td>0.0035</td>
</tr>
</tbody>
</table>

**Table 6:** Influence of a multiplication or addition by a scalar on $\tilde{\Delta}$ (moon500)
Resistance to monotone transformation of variables

<table>
<thead>
<tr>
<th>Operation</th>
<th>Number of variables</th>
<th>$\Delta$</th>
<th>$\sigma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiplication</td>
<td>0</td>
<td>0.3283</td>
<td>0.0072</td>
</tr>
<tr>
<td>Multiplication</td>
<td>1</td>
<td>0.3297</td>
<td>0.0060</td>
</tr>
<tr>
<td>Multiplication</td>
<td>2</td>
<td>0.3285</td>
<td>0.0067</td>
</tr>
<tr>
<td>Addition</td>
<td>0</td>
<td>0.3250</td>
<td>0.0053</td>
</tr>
<tr>
<td>Addition</td>
<td>1</td>
<td>0.3296</td>
<td>0.0046</td>
</tr>
<tr>
<td>Addition</td>
<td>2</td>
<td>0.3267</td>
<td>0.0059</td>
</tr>
</tbody>
</table>

**Table 7**: Influence of a multiplication or addition by a scalar on $\Delta$ (blob500)
Behaviour w.r.t correlated variables

Procedure:

- blob500 dataset
- replacement of each column by a random linear combination of another
- $\bar{\Delta}$ and $\sigma$ computation.
Behviour w.r.t correlated variables

Figure 1: Change of difference between mean intracluster and mean intercluster similarities when (i) changing features by linear combinations of other features and (ii) changing features by random values. The x axis represents the number of features modified by the procedure.
Conclusion

What has been presented:

- A novel stochastic method to compute similarities using decision trees.
- Extension of URF by using extremely randomized trees as a base estimator.
- With no need for instance generation.

Conclusion:

- Essentially one parameter influenced the results: $n_{min}$ (smoothing).
- Explanation: higher values $n_{min}$ give better results under noise.
Advantages of UET:

1. Synthetic data generation is no longer necessary.
2. 1.5 to more than 10 times faster than URF in our experiments.
3. Adaptability to complex data: attributed graphs
Application: Graph clustering
Graph clustering

What is a graph?

- $G = (V, E)$, $V$ set of vertices and $E$ a set of edges (pairs of vertices).
- Graphs can be attributed: vertices/edges endowed with an attribute tuple.

Goal of graph clustering:

- Two types: between and within graphs.
- Within graphs: find a partition of sets of related vertices in a graph.
- Related: connected by many edges w.r.t. vertices from other clusters.
- Vertex-attributed graphs: attribute homogeneity taken into account

Application of UET?

- A tree-based method for computing vertex (dis)similarities.
- Bridging the gap between random decision trees and graph clustering.
- Handles vertex attributes by building forests with different tree types.
Graph clustering

What is a graph?
- \( G = (V, E) \), \( V \) set of vertices and \( E \) a set of edges (pairs of vertices).
- Graphs can be attributed: vertices/edges endowed with an attribute tuple.

Goal of graph clustering:
- Two types: between and within graphs.
- Within graphs: find a partition of sets of related vertices in a graph.
- Related: connected by many edges w.r.t. vertices from other clusters.
- Vertex-attributed graphs: attribute homogeneity taken into account

Application of UET?
- A tree-based method for computing vertex (dis)similarities.
- Bridging the gap between random decision trees and graph clustering.
- Handles vertex attributes by building forests with different tree types.
Graph-Trees (GT)
Tree-based distances: field with recent interesting developments.

- Shi et al. 11: method to compute distances between samples in unsupervised settings.
- Dalleau et al. 12: extension using Extremely Randomized Trees, with better performance.
- Ting et al. 13: mass-based distance using isolation forests.

12 Unsupervised Extra Trees: a stochastic approach to compute similarities in heterogeneous data. International Journal of Data Science and Analytics, Springer Verlag, 2020
idea: use a hierarchical partitioning of the original space into non-overlapping and non-empty regions $H_i$'s

- $R(x, y|H_i)$ be the smallest local region covering $x$ and $y$ w.r.t. $H$.

Mass-based dissimilarity: estimated by a number $t$ of models is

$$m_e(x, y|D) = \frac{1}{t} \sum_{i=1}^{t} \tilde{P}(R(x, y|H_i))$$

where $\tilde{P}(R) = \frac{1}{|D|} \sum_{z \in D} 1(z \in R)$. 
Figure 2: Ex. of partitioning of 8 instances in non-overlapping non-empty regions using a random tree structure: $m_e(1, 4) = \frac{1}{8}(2) = 0.25$, and $m_e(1, 8) = \frac{1}{8}(8) = 1$. 
Idea of Graph Trees (GT)\textsuperscript{14}:

1. Compute several partitions of the vertices using random trees,
2. Compute a dissimilarity measure between the vertices using the partitions.

How are the partitions of vertices obtained?

- The root node of each tree contains all the vertices of the graph.
- At each node, a split is performed. Split:
  1. A vertex $v_1$ is randomly sampled from that node
  2. Each vertex $v_k$ that share an edge with $v_1$ form the left child node
  3. While all other vertices from the parent node form the right child node
- The growth is stopped when a stopping criterion is met.

\textsuperscript{14}https://gitlab.inria.fr/kdalleau/graphtrees/
Proposed approach: Graph Trees

Idea of Graph Trees (GT)\textsuperscript{14}:

1. Compute several partitions of the vertices using random trees,
2. Compute a dissimilarity measure between the vertices using the partitions.

How are the partitions of vertices obtained?

- The root node of each tree contains all the vertices of the graph.
- At each node, a split is performed. Split:
  1. A vertex $v_1$ is randomly sampled from that node
  2. Each vertex $v_k$ that share an edge with $v_1$ form the left child node
  3. While all other vertices from the parent node form the right child node
- The growth is stopped when a stopping criterion is met.

\textsuperscript{14}https://gitlab.inria.fr/kdalleau/graphtrees/
It is possible to build forests with different types of trees (*Graph forests*):

1. Graph trees that specialize on the graph structure
2. Trees that specialize on the attribute space.

**In our case:** Unsupervised Extremely randomized Trees (UET).

**Aggregation** of the (dis)similarities obtained with the different types of trees.
Experiments
Experiments on simple graphs

First evaluation: simple graphs with no attributes

1. Distance matrices using GT, with $n_{trees} = 200$
2. $k$-means on the points obtained using t-SNE $^{15}$ on the distance matrix
3. $\rightarrow$ NMI $^{16}$

The process repeated 20 times.

$^{15}$t-Distributed Stochastic Neighbor Embedding
$^{16}$Normalized Mutual Information
## Experiments on simple graphs

<table>
<thead>
<tr>
<th>Dataset</th>
<th># vertices</th>
<th># edges</th>
<th>Average degree</th>
<th># clusters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Football</td>
<td>115</td>
<td>1226</td>
<td>10.66</td>
<td>10</td>
</tr>
<tr>
<td>Email-Eu-Core</td>
<td>1005</td>
<td>25571</td>
<td>33.24</td>
<td>42</td>
</tr>
<tr>
<td>Polbooks</td>
<td>105</td>
<td>441</td>
<td>8.40</td>
<td>3</td>
</tr>
<tr>
<td>SBM3</td>
<td>450</td>
<td>65994</td>
<td>293.307</td>
<td>3</td>
</tr>
</tbody>
</table>

**Table 8:** Datasets used for the evaluation of GT clustering on simple graphs

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Graph-trees</th>
<th>Louvain\textsuperscript{17}</th>
<th>MCL\textsuperscript{18}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Football</td>
<td>0.923 (0.007)</td>
<td>0.924 (0.000)</td>
<td>0.879 (0.015)</td>
</tr>
<tr>
<td>Email-Eu-Core</td>
<td>0.649 (0.008)</td>
<td>0.428 (0.000)</td>
<td>0.589 (0.012)</td>
</tr>
<tr>
<td>Polbooks</td>
<td>0.524 (0.012)</td>
<td>0.521 (0.000)</td>
<td>0.544 (0.02)</td>
</tr>
<tr>
<td>SBM3</td>
<td>0.998 (0.005)</td>
<td>0.684 (0.000)</td>
<td>0.846 (0.000)</td>
</tr>
</tbody>
</table>

**Table 9:** Comparison of NMI on benchmark graph datasets. Best in boldface

\textsuperscript{17}Blondel et al.. Fast unfolding of communities in large networks. J. statistical mechanics : theory and experiment, 2008(10) :P10008, 2008

Experiments on attributed graphs

<table>
<thead>
<tr>
<th>Dataset</th>
<th># vertices</th>
<th># edges</th>
<th># attributes</th>
<th># clusters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parliament</td>
<td>451</td>
<td>11646</td>
<td>108</td>
<td>7</td>
</tr>
<tr>
<td>HVR</td>
<td>307</td>
<td>6526</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>Lawyers</td>
<td>71</td>
<td>575</td>
<td>70</td>
<td>2</td>
</tr>
<tr>
<td>WebKB</td>
<td>877</td>
<td>1480</td>
<td>1703</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 10: Datasets used for the evaluation of GT clustering on attributed graphs

<table>
<thead>
<tr>
<th>Dataset</th>
<th>NMI GT+UET</th>
<th>NMI Literature</th>
</tr>
</thead>
<tbody>
<tr>
<td>HVR</td>
<td>1.00 (0.000)</td>
<td>0.89</td>
</tr>
<tr>
<td>Parliament</td>
<td>0.65 (0.039)</td>
<td>0.78</td>
</tr>
<tr>
<td>Lawyers</td>
<td>0.12</td>
<td>0.66</td>
</tr>
<tr>
<td>WebKB</td>
<td>0.999 (0.002)</td>
<td>0.995 (0.002)</td>
</tr>
</tbody>
</table>

Table 11: Comparison of clusterings using GT. Best results from Bojchevski et al., and Maekawa et al. on WebKB. Best results are indicated in boldface.

19Bojchevski et al. Bayesian Robust Attributed Graph Clustering: Joint Learning of Partial Anomalies and Group Structure, 2018
20Maekawa et al.: Non-linear Attributed Graph Clustering by Symmetric NMF with PU Learn.2018
### Table 12: Results using the dissimilarities from UET and the labels (ground truth).

Best results are indicated in boldface.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>GT</th>
<th>GT+UET</th>
<th>Ground truth</th>
</tr>
</thead>
<tbody>
<tr>
<td>HVR</td>
<td>0.15</td>
<td>0.15</td>
<td>0.15</td>
</tr>
<tr>
<td>Parliament</td>
<td>0.46</td>
<td>0.15</td>
<td>0.20</td>
</tr>
<tr>
<td>Lawyers</td>
<td>0.27</td>
<td>0.23</td>
<td>0.26</td>
</tr>
<tr>
<td>WebKB</td>
<td>0.74</td>
<td>0.70</td>
<td>0.74</td>
</tr>
</tbody>
</table>
Discussion
• Method based on the construction of random trees to compute similarities between graph vertices.

• Competitive with state of the art methods in terms of quality of clustering on non-attributed graphs.

• Computing forests of GT and other trees that specialize in other types of input data: possible to compute dissimilarities between vertices in attributed graphs.
Graph forests using UET for the attribute trees seems promising:

- Less preprocessing, can manage mixed types attributes *out of the box*.
- **Some control**: importance of the vertex attributes, choice of aggregation method between the graph trees and the attribute trees
- **Real life application**: Project RHU Fight-HF\(^2^1\) \(^2^2\)

However

- **Empirical evaluation**: quality that varies greatly between the datasets.
- Choice to consider the attribute space: guided by the distribution of the variables or a visualization of the embeddings?

\(^2^1\) [https://anr.fr/ProjetIA-15-RHUS-0004](https://anr.fr/ProjetIA-15-RHUS-0004)

Merci de votre attention!

Questions?