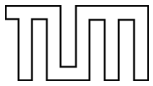


Technische Universität München

Department of Mathematics

Master's Thesis

A Novel Information Theoretic Metric for Labeled Trees

Denys Lazarenko

Supervisor: PD Dr. rer. nat. Tobias Lasser

Advisor: Prof. Thomas Bonald.

Submission Date: 29.04.2021

I hereby declare that this thesis is my own work and that no other sources have been used except those clearly indicated and referenced.

Garching,

Zusammenfassung

Häufig müssen Ähnlichkeiten von Bäumen, die aus dem gleichen Satz Blattknoten bestehen, jedoch unterschiedliche Topologien aufweisen, quantitativ verglichen werden. Obwohl diese Problematik bekannt ist und viele Metriken vorhanden sind, unterliegen diese verschiedenen Einschränkungen und sind nicht für größere Bäumen skalierbar. Diese Arbeit stellt die Basis einer neuartigen informationstheoretischen Metrik für Bäume, genannt Baum-Transinformation dar, die durch die von den betrachteten Bäumen geteilte Information ausgewertet werden kann. Die Metrik basiert auf der besten übereinstimmenden Teilung der von zwei Bäumen erzeugten Blattknoten eines Satzes. Diese Methode kann verwendet werden, um die Qualität des hierarchischen Clusterns zu bewerten und die Ergebnisse zu interpretieren.

Zusätzlich zu dieser neuartigen Metrik wird in dieser Arbeit eine neue Technik zur Bewertung der angepassten Transinformation, basierend auf paarweisen Vertauschungen, vorgestellt. Diese Berechnungsmethode ist wesentlich schneller und kann daher für den Vergleich größerer Bäume verwendet werden.

Alle Experimente wurden sowohl mit synthetischen als auch mit realen Datensätzen durchgeführt, um die Effizienz des Ansatzes in verschiedenen Situationen zu veranschaulichen. Die vorgeschlagene Metrik übertrifft bereits bestehende Metriken sowohl in der Qualität als auch in der Laufzeit.

Schlüsselwörter: Hierarchisches Clustering, Dendrogram, Bäume, angepasste Transinformation, Informationstheorie.

Abstract

It is frequently the case that we need to quantitatively compare the similarity or distance between trees composed of the same set of leaves but presenting different topologies. This problem is not new, and although there are many existing metrics, they suffer from numerous limitations and do not scale to large trees. This thesis provides a novel Information Theoretic Metric for Trees called Tree Mutual Information (TMI) which can be interpreted through the information shared by trees. The metric is based on the best-aligned partitions of the set of leaves induced by both trees. It can be used to evaluate the quality of hierarchical clustering and to interpret its results.

In addition to the novel metric, this thesis proposes a new technique for evaluating adjusted mutual information based on pairwise permutations. The computation is much faster, and can thus be used for comparing large trees. All the experiments were conducted both on synthetic and real datasets to illustrate the approach's efficiency in different settings. The proposed metric outperforms existing metrics both in quality and running time.

Keywords: Hierarchical clustering, Dendrogram, Tree, Adjusted Mutual Information, Information Theory.

Acknowledgement

This work was performed in cooperation with Télécom Paris. I would first like to thank my advisor Thomas Bonald at Télécom Paris who provided me with critical guidance during this research. I would also like to thank my thesis supervisor PD Dr. rer. nat. Tobias Lasser of the Informatics department at TUM who accompanied me throughout this Master Thesis, read it carefully and granted me precious comments on my work.

Contents

List of Figures

List of Tables

List of original publications

This thesis is based on the following original publication:

[1] Denys Lazarenko and Thomas Bonald. Pairwise Adjusted Mutual Information. 2021. arXiv: 2103.12641 [cs.LG].

ABBREVIATIONS

MI - Mutual Information
AMI - Adjusted Mutual Information
TMI - Tree Mutual Information
TAMI - Tree Adjusted Mutual Information
TPAMI - Tree Pairwise Adjusted Mutual Information
TED - Tree Edited Distance
RF - Robinson-Faulds

Chapter 1

Introduction

Trees are probably one of the most frequently used data structures in Computer Science due to their simplicity of implementation and understanding. It is well known how to analyse, traverse and use them as a base element of more complicated algorithms. This is not an exception for Machine Learning, where trees serve different purposes: Decision tree, Random Forest or Tree LSTMs. Additionally, it is frequently the case that we need to quantitatively compare similarity or calculate distance between two or more trees that have the same set of leaves but different topologies. To solve these tasks, numerous metrics were proposed [1], [2], [3]. However, all these metrics either have high complexity or perform well only in specific domains.

Indeed, the application area is vast. One of the most popular tasks of unsupervised learning is clustering. Its main goal is to make partitions of objects which are similar to each other. The subfamily of clustering algorithms, called hierarchical clustering, is often used in various tasks due to its explainability and adaptivity. There is a considerable demand for a good metric that would be able to evaluate performance of hierarchical clustering algorithms in order to compare them and adjust their hyperparameters.

While in supervised learning scenarios it is possible to compare ground truth labels with predicted values, in unsupervised learning this is not the case. To tackle this issue, some new metrics were introduced. These include Dasgupta cost [4] and Tree Sampling Divergence distance [5]. Both of these metrics were designed for graph structures, but data is not always available in this format.

Likewise, another area of application that relies on tree structures is biology. Phylogenetic trees have the same leaf labels, but different branching structures summarize information about historical evolutionary relationships, which are represented by sequenced genes. On the side, hierarchical algorithms have shown a great success in finance including portfolio diversification, optimization, risk management and market analysis [6], [7]. Last but not least, hierarchy is used in Linguistic and Natural Language Processing. For example, WordNet [8] - a well-known lexical collection where parts of speech are

grouped into sets of cognitive synonyms, exhibits a high quality. Nevertheless, it was collected manually requiring significant time and resources. In order to automatise this process, it is possible to take a big corpora of documents (i.e. English Wikipedia), apply hierarchical clustering algorithms, compare them with the existing ground truth of words grouped by themes from WordNet and develop an accurate algorithm which can then be scaled to other languages.

The performance evaluation of clustering algorithms is also a non-trivial task, requiring the evaluation metric not only to be independent of cluster labels, but also to partition data based on some membership criteria defined by a similarity metric. Let's consider some partitions of a set of n items. Trivial partitions such as a single set or n singletons convey little information about these items. We introduce the notion of adjusted entropy, which was derived from adjusted mutual information and quantifies the amount of information obtained by some partition. We present a novel Information Theoretic Metric for Labeled Trees called Tree Mutual Information (TMI), which can be in turn interpreted through adjusted mutual information. The metric is based on a maximum alignment between leaf sets in labeled trees. Further, it is equal to zero for the above trivial clustering and approximates optimal partitions with respect to the shared information score.

As an additional contribution to our primary problem solution, we introduce a new metric to quantify similarity between two clustering [9]. Hence, we propose an adjustment based on pairwise label permutations instead of full label permutations. Specifically, we consider permutations where only two samples, selected uniformly at random, exchange their labels. We show that the corresponding adjusted metric, which can be expressed explicitly, behaves similarly to the standard adjusted mutual information while having much lower time complexity. Lastly, both metrics are compared in terms of quality and performance on experiments utilising synthetic and real data.

1.1 Goals of the Thesis

In this thesis, we tackle the problem of measuring similarity between trees with same labels. Furthermore, we address the following research questions:

Q1 How to efficiently measure the similarity between two labeled trees with the same leaf sets but different topology?

Q2 How well does a novel information-theoretic metric assess the quality of hierarchical clustering of different data types? What is the optimal number of clusters that maximizes similarity between two dendrograms?

Q3 What are some pros and cons of the novel metric in comparison to state-of-the-art?

By the end of the thesis, all the research questions should be answered and a set of software artifacts produced. The artifacts can be used by the Chair of Mathematics and the general public in testing and evaluation.

1.2 Outline

This thesis is structured as follows:

Chapter 2 contains some background knowledge on data types, data structures, and common approaches to deal with the domain of tree comparison. We proceed with the main ideas of unsupervised learning and hierarchical clustering, on which we are mostly concentrated in this thesis.

In Chapter 3 we introduce existing metrics to evaluate performance of tree similarity measurement and its main components. We explain pros and cons of each metric and select one for future analysis. We then proceed with an evaluation strategy to compare different experiments that we introduce in Chapter 5.

In Chapter 4 we provide a full theoretical explanation which underlies the novel metric. We prove some crucial theorems and propositions as well as analyse the complexity of the metric.

In Chapter 5 we start with the experimental setup and some details about data choice for our experiments. We then describe the main baselines and our novel methodology. After that, we show the results of our experiments. We analyse the results for various hyperparameters of the models and conduct quantitative and qualitative comparisons of different approaches. We discuss the results to outline the main bottlenecks and limitations of designed approaches.

The thesis is concluded by Chapter 6, which contains a summary of the thesis and answers research questions. Lastly, we raise questions for further research and specify pathways for improvements.

Chapter 2

Background

In this chapter, we provide some background on topics, closely related to this thesis. In the first section, we explain for which kind of data types hierarchical clustering could be applied. We describe the motivation behind them and their purpose. In the second part, we examine different types of hierarchical clustering algorithms.

2.1 Data structures

There are numerous ways to categorize data in clustering problems. The most frequent definitions are: numerical, categorical, spatial, multivariate, and mixed. Depending on data type, different clustering algorithms are used. Besides the nature of data, there are different ways to represent the same data types. In this work, we consider data being represented as graphs or vectors.

2.1.1 Graphs

Here we display a typical example of an input to hierarchical clustering algorithm - graph, see the Figure 2.1: directed, undirected or bipartite represented as an adjacency matrix. If the graph is directed, then matrix A is symmetric, if the graph is weighted, then A_{ij} represents the weight between edge i and j. V describes the set of n vertices or nodes, and E describes the set of m edges [10].

Given a weighted, undirected, connected graph $G = (V, E)$ of n nodes and m edges without self-loops represented as adjacency matrix A. We denote by $d = A1$ the vector of degrees and by v the volume of the graph:

$$v = \sum_{i \in V} d_i = \sum_{i,j \in V} A_{ij}$$

.

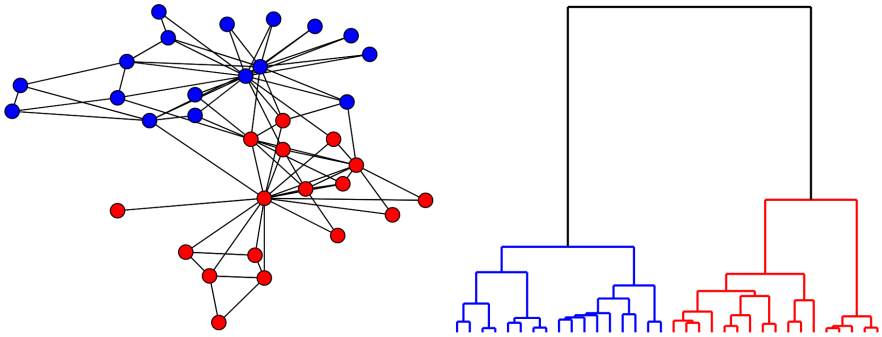


Figure 2.1: Graph and its representation as a dendrogram.

We introduce the definition of **sampling**. Under edge sampling, each node pair i, j is sampled with probability:

$$p(i,j) = \frac{A_{i,j}}{v}$$

The marginal distribution is calculated as:

$$p(i) = \sum_{j \in V} p(i,j) = \frac{d_i}{v}$$

If the sampling of node i is known, to calculate the probability of node j is possible through a conditional probability:

$$p(j|i) = \frac{p(i,j)}{p(i)} = \frac{A_i j}{d_i}$$

A number of hierarchical clustering algorithms have been developed specifically for graphs [11] [10] [12] [13] [14].

2.1.2 Vector Data

It is probably the most common to represent data. Usually, clustering techniques are applied to vector data. Several clustering algorithms have specifically been developed for this task, they do not directly apply to graphs unless the graph is embedded in some metric space. It is possible to transform vector data in graphs and vice versa. The choice of a distance is important for vector data. Common hierarchical clustering techniques are the linkage algorithm with different distance update formula and the nearest-neighbour clustering algorithm [15] [16].

2.2 Hierarchical clustering

Let's consider n points $x_1, \ldots, x_n \in R^d$, which we would like to cluster in hierarchical way that should capture the natural structure of a real dataset.

2.2.1 Divisive approach

The first class of algorithms to consider is *divisive*, which is a top-down method in which we begin from a single cluster and divide it sequentially before all objects are isolated or other stop criteria are used. Since there are $\mathcal{O}(2^n)$ ways to break each cluster, the solution is computationally costly. Therefore, all experiments in this work are conducted using *agglomerative algorithms*.

2.2.2 Agglomerative approach

In contrast to Divisive, the agglomerative method begins at individual clusters with one data point and merges clusters recursively. The *closest* clusters are merged depending on a distance d between them. The distance d is not necessarily a metric. It is only necessary that d be non-negative and symmetric. The algorithm is the following [17]:

1. Initialization: $K \leftarrow \{\{1\}, \ldots, \{n\}\}$
2. Agglomeration: For $t = 1, \ldots, n-1$,
   - $A, B \leftarrow argmin_{a,b \in K, a \neq b} d(a, b)$
   - $C \leftarrow A \cup B$
   - $K \leftarrow K \setminus \{A, B\}$
   - $K \leftarrow K \cup \{C\}$
   - Output $A, B, d(A, B)$

Ward Ward's distance [18] is a common distance for Agglomerative algorithms. Ward seeks to reduce the number of squared disparities in all clusters. It is analogous to the objective function of K-means [19]. Let $g(c)$ be the centroid of any cluster c

$$g(c) = \frac{1}{|c|} \sum_{i \in c} x_i$$

, and S be the complete square Euclidean distance of points in c to their centroid

$$S(c) = \sum_{i \in c} ||x_i - g(c)||^2$$

Then, after some simplification, we define the distance as:

$$d(a, b) = S(a \cup b) - S(a) - S(b) = \frac{|a||b|}{|a| + |b|} ||g(a) - g(b)||^2$$

Paris It is a new algorithm for graphs proposed by [10]. The choice of "proximity" between nodes follows from sampling. Node j is close to node i if the probability of sampling node j given the sampling of node i is much higher than the probability of

sampling node j. Hence, similarity between nodes can be expressed as:

$$\sigma(i,j) = \frac{p(j|i)}{p(j)} = \frac{p(i,j)}{p(i)p(j)} = v\frac{A_{ij}}{d_i d_j}$$

The general idea of Paris algorithm is:

Algorithm 1: Paris algorithm.

Input: G=(V,E)
Output: List of merges, L

```
for t = 1, ..., n − 1 do
    i, j ← argmax_{i,j∈V, i≠j} σ(i, j) ;
    append i, j to L merge i, j into node n + t update σ
end for
```

Algorithm 2: Hierarchical Louvain algorithm.

Input: G=(V,E)
Output: List of merges

```
clusters ← Louvain(G);
if | clusters |> 1 then
    graphs ← GetSubgraphs(G, clusters) ;
    return [HierarchicalLouvain(S) for S in graphs]
end if
else
    return [nodes(G)]
end if
```

Louvain Let us now look for the algorithm which uses a modularity as a distance metric. Let $\delta_C(i,j) = 1$ if i, j are in the same cluster and 0 otherwise. The modularity of clustering C is defined by [12]:

$$Q(C) = \sum_{i,j \in V} (p(i,j) - p(i)p(j))\delta_C(i,j)$$

As a consequence, modularity can be defined as the difference in the probabilities of sampling two nodes from the same cluster using the joint distribution $p(i,j)$ and product $p(i)p(j)$. The Louvain algorithm is composed of 4 steps:

1. **Initialization:** $C \leftarrow \{\{1\}, \ldots \{n\}\}$
2. **Iteration:** when modularity $Q(C)$ increases, update C by moving one node from one cluster to another.
3. **Aggregation:** merge all nodes belonging to the same cluster, update the weights and apply step 2.
4. Return C.

It is easy to move from the general version of the Louvain clustering algorithm to the hierarchical one, as shown in Algorithm 2.

2.2.3 Dendrograms

Regardless of which approach we choose to produce a hierarchical clustering, as an output, we will have a dendrogram Figure 2.1 as an output. The dendrogram D contains the pair of nodes merged through the run of the algorithm. By browsing the final dendrogram bottom-up, all partitions $P_0, \ldots, P_{n-1}$ can be retrieved. It is worth noting that the partition P_t is made of $n - t$ clusters. Additionally, a dendrogram includes the distance $d_t = d(i, j)$ and number of nodes within $n_{n+t} = n_i + n_j$ a cluster. Each branch is plotted at height d_t, thus all distances must be non-decreasing.

2.2.4 Trees

Once we obtain a dendrogram D, the next step is to analyze and work with it. A dendrogram's data structure is quite difficult to manage and has no convenient algorithms for it. However, dendrogram could be easily converted into a tree data structure. Formally, a tree is an acyclic and connected graph: $T = (V, E)$ with a set of vertices V and a set of edges E. Each node in a tree has zero or more child nodes, which lay under it in the tree. In this work, we mostly study rooted trees. Yet, it is possible to apply a new metric - explained in Chapter 4 - to unrooted trees as well. We store trees in a Newick format [20] which is a very flexible data representation. Therefore, we are going to use terms of dendrograms and trees interchangeably, assuming that we are able to very easily convert dendrograms to trees and backward.

Chapter 3

Clustering performance evaluation

The field of flat clustering algorithms is intensively addressed. Numerous metrics were created over the past years: Adjusted Rand Index(AMI) [21], Adjusted Mutual Information(AMI) [22], Homogeneity, Completeness and V-measure [23], Silhouette Coefficient [24] and many others. However, these general metrics are not directly applicable to the case of hierarchical clustering algorithms.

This chapter will present some of the most relevant hierarchical clustering metrics. In the following sections of this chapter, we analyse different hierarchical clustering metrics that vary in accordance to their baseline theory, computational complexity and explainability. Some of them are strictly applicable to graphs, while others are rather for a general usage. We compare their strength and weaknesses along with their limitations.

3.1 Graph based metrics

Let's consider a weighted, undirected, connected graph $G = (V, E)$ of n nodes, without self-loops. Let $w(i, j)$ be equal to the weight of edge i, j, if any, and to 0 otherwise [25]. We refer to the weight of node i as:

$$w(i) = \sum_{j \in V} w(i, j)$$

We denote by w the total weight of nodes:

$$w = \sum_{i \in V} w(i)$$

Similarly, for any sets A, $B \subset V$, let

$$w(A, B) = \sum_{i \in A, j \in B} w(i, j)$$

and

$$w(A) = \sum_{i \in A} w(i)$$

Cluster sampling. For every cluster pair we can calculate probability distribution as follows:

$$\forall A, B \in P, p(A, B) = \frac{w(A, B)}{w}$$

with marginal distribution

$$\forall A \in P, p(A) = \sum_{B \in P} p(A, B) = \frac{w(A)}{w}$$

Dasgupta Cost. Recently, a new metric to evaluate the quality of hierarchical clustering was introduced by Dasgupta [4] and was extended in [26], [27].

Assume that a given dendrogram represents the graph G, a rooted binary tree T whose leaves are the graph's nodes V. We denote by $\mathcal{I}$ the set of internal nodes of the tree T [25]. Dasgupta's cost function can be expressed as a sum over all internal tree's T nodes of joint sampling probabilities of two nodes multiplied by the sum of their marginal probabilities 3.1.

$$\sum_{A,B:(A,B) \in \mathcal{I}} p(A, B)(p(A) + p(B)) \tag{3.1}$$

The shortcomings of this method are:

- It necessarily relies on the structure of a graph.
- It is not a continuous function of A and B: slight changes modify the score significantly.
- Its main issue lies in a tie-breaking: since we have merges of equal heights in a dendrogram, it is not obvious to decide which pair of clusters to consider for merge first. Each will give a different value for Dasgupta's cost. Finding the tree that minimizes the cost function is NP-hard [4].

Since we would like to test beyond graph-based datasets, we discard this metric for the evaluation.

3.2 Tree based metrics

Robinson-Faulds. The Robinson-Foulds (RF) distance is one of the most frequently used metrics to measure the distance between trees [28]. It measures the number of branch partitions (splits) present only in one of the trees and scores 1 for each division that is not matched. In order to move from the distance calculation to similarity, we use the formula below.

$$\mathrm{RF} = 1 - \frac{\text{number of dissimilar splits}}{\text{total number of splits}}$$

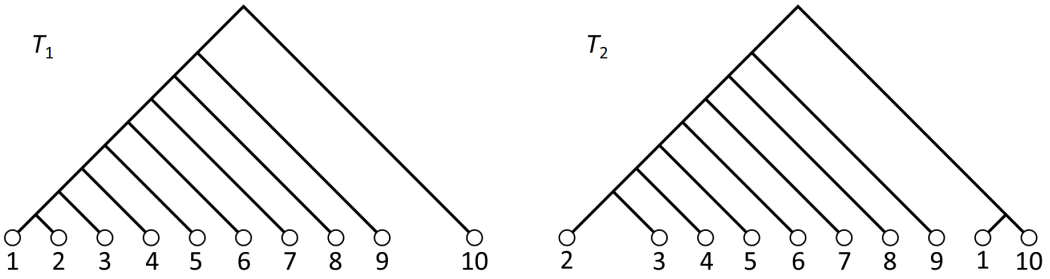


Figure 3.1: Two rooted phylogenetic trees. Despite their high similarity, the RF metric between these two trees is equal to $\frac{1}{9}$ [1].

RF has its well-known shortcomings. For instance, moving a single node in a tree can result in a considerable jump of RF score when, in reality, these trees are almost identical as shown in Figure 3.1. There are numerous modifications of this metric, which are trying to address this issue [29], [30]. The RF-like metrics are all based on the idea of shared clades, branches, or triplets [31] defined by belongings of exactly the same element. That is why this class of metrics is not robust and unstable for slight permutations. From an Information Theory point of view, RF metric treats each split as having the same amount of information, which is certainly a downside.

The standard RF has the complexity of $O(n)$ while computing its generalized version is NP-hard. Thus, we will only consider the standard Robinson-Faulds metric.

Tree edited distance. Another frequently used algorithm is based on the concept of string-to-string correction. The string-to-string correction problem minimizes the number of edit operations to transform one string into another. Normally, three main editing operations are defined: substitution, insertion and elimination of a character [32]. Nonetheless, it is necessary to adapt the algorithm to the tree's context [2]: change one node label, delete or insert a node. To move over the tree, the post-order traversal is used. We define TED similarity as suggested in [33]:

$$TED(T_1, T_2) = 1 - \frac{\mathrm{TED\_DIST}(T_1, T_2)}{\mid T_1 \mid + \mid T_2 \mid}$$

This algorithm is very well suitable for ordered labeled trees. It has complexity for 2 trees T_1 and T_2 equal to:

$$O(\mid T_1 \mid\mid T_2 \mid \min(\mathit{depth}(T_1), \mathit{leaves}(T_1)) \min(\mathit{depth}(T_2), \mathit{leaves}(T_2)))$$

The space complexity is $O(\mid T_1 \mid\mid T_2 \mid)$ [2]. There exist several modifications of this algorithm [34], [35], [36]. Due to its time and space complexities, it is difficult to apply

the algorithm for real-world problems. The algorithm's efficiency highly depends on the tree shape.

3.3 Metrics requirements

To measure tree comparison metrics' performance, we will rely on the three most fundamental criteria: similarity, time and optimal partitioning.

Similarity To capture quantitatively behaviour of metrics in the syntactic experiments Chapter 5, where we vary the amount of noise, we measure the Pearson correlation between metrics' results and noise.

$$corr = \frac{\sum(x - m_x)(y - m_y)}{\sqrt{\sum(x - m_x)^2 \sum(y - m_y)}}$$

,where m_x is the mean of the vector x and m_y is the mean of vector y [37].

Time complexity Since we expect our metric to scale on big datasets, we seek to have a clear understanding of how it grows with the increasing number of samples and the topology of trees. Therefore, we will measure time complexity not only analytically but practically as well.

Optimal partitioning Another critical feature for the metric is to be explainable and transparent, that is why we would like to observe how the optimal number of clusters correlate with the similarity score of the novel metric.

In the later chapters, we will use RF and TED metrics as the state of the art metrics and apply evaluation criteria in the evaluation experiments.

Chapter 4

Methodology

This Chapter presents a novel Information Theoretic Metric for Labeled Trees named Tree Mutual Information(TMI), which can be interpreted through the mutual information score and its expected value. The metric is based on a maximum alignment between leaves sets in labeled trees. We first explain the relation of our novel metric to information theory, show the application of adjusted mutual information in this context, and introduce a novel adjustment based on pairwise label permutations instead of full label permutations. We show that the corresponding adjusted metric can be expressed explicitly and behaves similarly to the standard adjusted mutual information for assessing the quality of clustering while having a much lower time complexity. Both metrics are adopted to be used as a construction element of a tree comparison solution.

Most of the proofs in this Chapter can be found in our publication [9].

4.1 Information Theory

Let P be the uniform probability measure on $\Omega = \{1, \ldots, n\}$, for some positive integer n. Let X, Y be random variables on the probability space (Ω, P). Without any loss of generality, we assume that X and Y are mapping from Ω to sets consisting of consecutive integers, starting from 1. Denoting by H the entropy, the mutual information between X and Y is defined by [38]:

$$I(X, Y) = H(X) + H(Y) - H(X, Y). \tag{4.1}$$

This is the information shared by X and Y, which is equal to 0 if X and Y are independent. A distance between X and Y can then be defined by:

$$d(X, Y) = H(X, Y) - I(X, Y) = H(X|Y) + H(Y|X).$$

This distance, known as the variation of information, is a metric in the quotient space of random variables under the equivalence relation $X \sim Y$ if and only if there is some bijection φ such that $X = \varphi(Y)$ [39].

Adjusted mutual information. The adjusted mutual information between X and Y, corresponding to the mutual information between X and Y *adjusted* against chance, is defined by:

$$\Delta I(X, Y) = I(X, Y) - \mathrm{E}(I(X, Y_\sigma)), \tag{4.2}$$

where Y_σ is the random variable $Y \circ \sigma$, for any permutation σ of $\{1, \ldots, n\}$, and the expectation is taken over all permutations σ, chosen uniformly at random.

Remark 1 (Normalization) *It is also frequent to normalize adjusted mutual information to get a score between 0 and 1 [22, 40]. In this work, we only focus on the adjustment step.*

Definition:

$$\begin{aligned} \Delta I(X, Y) &= \mathrm{E}(H(X, Y_\sigma)) - H(X, Y), \\ &= \frac{1}{2}(\mathrm{E}(d(X, Y_\sigma)) - d(X, Y)). \end{aligned} \tag{4.3}$$

This equivalence follows from Proposition 1 and the fact that the definition is symmetric in X and Y. All proofs could be found in the [9].

Proposition 1 *We have for any random variables X and Y:*

$$\begin{aligned} H(X) &= \mathrm{E}(H(X_\sigma)), \\ \mathrm{E}(H(X, Y_\sigma)) &= \mathrm{E}(H(X_\sigma, Y)), \\ \mathrm{E}(I(X, Y_\sigma)) &= \mathrm{E}(I(X_\sigma, Y)). \end{aligned}$$

In view of (4.3), we expect $\Delta I(X, Y)$ to be positive if X and Y share information, as X is expected to be closer to Y (for the distance d) than to Y_σ, a randomized version of Y. There are specific cases where $\Delta I(X, Y) = 0$, as stated in

Proposition 2 *We have $\Delta I(X, Y) = 0$ whenever Y (or X, by symmetry) is constant or equal to some permutation of $\{1, \ldots, n\}$.*

Adjusted entropy. Observing that $H(X) = I(X, X)$, we define similarly the adjusted entropy of X by:

$$\Delta H(X) = \Delta I(X, X) = H(X) - \mathrm{E}(I(X, X_\sigma)).$$

By (4.1), we get:

$$\Delta H(X) = \mathrm{E}(H(X, X_\sigma)) - H(X) = \frac{1}{2}\mathrm{E}(d(X, X_\sigma)). \tag{4.4}$$

Since d is a metric, this shows that the adjusted entropy of X is non-negative.

Proposition 3 *We have $\Delta H(X) = 0$ if and only if X is constant or equal to some permutation of $\{1, \ldots, n\}$.*

Proposition 3 characterizes random variables with zero adjusted entropy. Proposition results will be interpreted in terms of clustering in section 4.4.

4.2 Pairwise adjustment

In this section, pairwise adjusted mutual information is introduced. The definition is the same as adjusted mutual information, except that the permutation σ is now restricted to the set of pairwise permutations. Specifically, we consider permutations σ for which there exists $i, j \in \{1, \ldots, n\}$ such that $\sigma(i) = j$ and $\sigma(j) = i$, whereas $\sigma(t) = t$ for all $t \neq i, j$. We consider the set of such permutations σ where the samples i, j are drawn uniformly at random in the set $\{1, \ldots, n\}$. We denote by σ_{p} such a random permutation. Observe that σ_{p} is the identity with probability $1/n$ (the probability that $i = j$).

Pairwise adjusted mutual information. We define the *pairwise adjusted mutual information* as:

$$\Delta_{\mathrm{p}} I(X, Y) = I(X, Y) - \mathrm{E}(I(X, Y_{\sigma_{\mathrm{p}}})).$$

This is exactly the same definition as the adjusted mutual information, except for the considered permutations σ_{p}. It can be readily verified that the same properties apply, with the exact same proofs, a key property being that the random permutations σ_{p} and ${\sigma_{\mathrm{p}}}^{-1}$ have the same distributions.

In particular, we have the analogue of (4.3):

$$\begin{aligned} \Delta_{\mathrm{p}} I(X, Y) &= \mathrm{E}(H(X, Y_{\sigma_{\mathrm{p}}})) - H(X, Y), \\ &= \frac{1}{2}(\mathrm{E}(d(X, Y_{\sigma_{\mathrm{p}}})) - d(X, Y)). \end{aligned} \tag{4.5}$$

Moreover, $\Delta_{\mathrm{p}} I(X, Y) = 0$ whenever X or Y is constant or equal to some permutation of $\{1, \ldots, n\}$.

Pairwise adjusted entropy. We also define the *pairwise adjusted entropy* as:

$$\Delta_{\mathrm{p}} H(X) = \Delta_{\mathrm{p}} I(X, X) = H(X) - \mathrm{E}(I(X, X_{\sigma_{\mathrm{p}}})).$$

We have $\Delta_{\mathrm{p}} H(X) \geq 0$, with equality if and only if X is constant or equal to some permutation of $\{1, \ldots, n\}$.

4.3 Data processing inequality

The following results give two versions of the data processing inequality for the adjusted mutual information. We write $X \prec Y$ if $X = f(Y)$ for some mapping f. The proofs are deferred to the Appendix.

Theorem 1 *If $X \prec Y$, then*

$$\Delta I(X,Y) \leq \Delta I(X,X).$$

Moreover, the inequality is strict whenever $0 < H(X) < H(Y)$.

Proof of Theorem 1. Since $X \prec Y$, we have $H(X|Y) = 0$ and

$$I(X,Y) = H(X) - H(X|Y) = H(X) = I(X,X).$$

Now for any permutation σ of $\{1,\ldots,n\}$, we have $I(X_\sigma, Y) \geq I(X_\sigma, X)$ by the data processing inequality [38], which implies:

$$\Delta I(X,Y) \leq \Delta I(X,X).$$

Now assume that $0 < H(X) < H(Y)$. To prove that the inequality is strict, let σ be the permutation of $a, b \in \{1,\ldots,n\}$ with $X(a) \neq X(b)$. Let $i = X(a), j = X(b)$ and $A = \{\omega : X(\omega) = i\}, B = \{\omega : X(\omega) = j\}$. We get:

$$\mathrm{P}(X_\sigma = j|\, X = i) = \frac{1}{|A|}, \quad \mathrm{P}(X_\sigma = i|\, X = j) = \frac{1}{|B|}, \quad \mathrm{P}(X_\sigma = k|\, X = k) = 1 \quad \forall k \neq i, j,$$

so that:

$$H(X_\sigma|X) = \frac{|A|}{n} H\left(\frac{1}{|A|}\right) + \frac{|B|}{n} H\left(\frac{1}{|B|}\right),$$

with $H(p) = -p\log p - (1-p)\log(1-p)$ the entropy of a Bernoulli random variable with parameter p.

Now let $i' = Y(a), j' = Y(b)$ and $A' = \{\omega : Y(\omega) = i\}, B' = \{\omega : Y(\omega) = b\}$. Observe that $i = f(i')$ and $j = f(j')$, where f is the mapping such that $X = f(Y)$, so that $i' \neq j'$. Moreover, $|A'| \leq |A|$ and $|B'| \leq |B|$. Since $H(Y) > H(X)$, we can choose σ such that $|A'| < |A|$ or $|B'| < |B|$. We have:

$$\Pr(X_\sigma = j|\, Y = i') = \frac{1}{|A'|}, \quad \Pr(X_\sigma = i|\, Y = j') = \frac{1}{|B'|}, \quad \Pr(X_\sigma = f(k)|\, Y = k) = 1$$

so that:

$$H(X_\sigma|Y) = \frac{|A'|}{n} H\left(\frac{1}{|A'|}\right) + \frac{|B'|}{n} H\left(\frac{1}{|B'|}\right).$$

Since the mapping $t \mapsto tH\left(\frac{1}{t}\right)$ is increasing over $(1, +\infty)$, we get:

$$H(X_\sigma|Y) < H(X_\sigma|X)$$

and

$$I(X_\sigma, Y) = H(X_\sigma) - H(X_\sigma|Y) > H(X_\sigma) - H(X_\sigma|X) = I(X_\sigma, X).$$

In particular, $\mathrm{E}(I(X_\sigma, Y)) > \mathrm{E}(I(X_\sigma, X))$ and $\Delta I(X, Y) < \Delta I(X, X)$. □

Theorem 2 *If $X \prec Y$, $X \prec Z$ and the random variables Y and Z are independent conditionally to X, then*

$$\Delta I(Y, Z) \leq \Delta I(X, X).$$

Moreover, the inequality is strict whenever $0 < H(X) < \max(H(Y), H(Z))$.

Proof of Theorem 2. We have:

$$\begin{aligned} I(Y, Z) &= H(Y) + H(Z) - H(Y, Z), \\ &= H(X, Y) + H(X, Z) - H(X, Y, Z), \\ &= H(X) + H(Y|X) + H(X) + H(Z|X) - H(X) - H(Y, Z|X), \\ &= H(X) + I(Y, Z|X), \\ &= H(X) = I(X, X), \end{aligned}$$

where we used the fact that $X \prec Y$ and $X \prec Z$ for the second equality and the fact that $Y \perp Z|X$ for the last equality.

Now assume that $0 < H(X) < \max(H(Y), H(Z))$. For any permutation σ of $\{1, \ldots, n\}$, we have:

$$I(Y_\sigma, Z) \geq I(Y_\sigma, X) = I(Y, X_{\sigma^{-1}}) \geq I(X, X_{\sigma^{-1}}) = I(X, X_\sigma),$$

where both inequalities follows from the data processing inequality [38]. This implies:

$$\Delta I(Y, Z) \leq \Delta I(X, X).$$

A similar argument as that used in the proof of Theorem 1 shows that this inequality is strict. □

Using the same logic, we can prove theorems mentioned above for the Pairwise metric [9].

4.4 Application to clustering

Let $A = \{A_1, \ldots, A_k\}$ and $B = \{B_1, \ldots, B_l\}$ be two partitions of some finite set $\{1, \ldots, n\}$ into k and l clusters, respectively. Let $\Omega = \{1, \ldots, n\}$ and P be the uniform probability measure over Ω. Consider the random variables X and Y defined on (Ω, P) by $X^{-1}(i) = A_i$ for all $i = 1, \ldots, k$ and $Y^{-1}(j) = B_j$ for all $j = 1, \ldots, l$. Note that $X(\omega)$ and $Y(\omega)$ can be interpreted as the *labels* i and j of sample ω in clusterings A and B, for each $\omega \in \{1, \ldots, n\}$. We denote by $a_i = |A_i|$ the size of cluster A_i, by $b_j = |B_j|$ the size of cluster B_j, and by $n_{ij} = |A_i \cap B_j|$ the number of samples both in cluster A_i and cluster B_j, for all $i = 1, \ldots, k$ and $j = 1, \ldots, l$. The matrix $(n_{ij})_{1 \leq i \leq k, 1 \leq j \leq l}$ is known as the *contingency matrix*. Note that a_i and b_j are the respective sums of row i and column j of the contingency matrix.

Adjusted mutual information(AMI). A well-known metric for assessing the similarity $s(A, B)$ between clusterings A and B is the adjusted mutual information[1] $\Delta I(X, Y)$ between the corresponding random variables X and Y. In words, this is the common information shared by clusterings A and B not due to randomness.

By Proposition 2, we have $s(A, B) = 0$ whenever clustering A (or B, by symmetry) is trivial, that is, it consists of a single cluster or of n clusters (one per sample). This is a key property, showing the interest of the adjustment.

It is known that [22]:

$$s(A,B) = -\sum_{i=1}^{k}\sum_{j=1}^{l}\frac{n_{ij}}{n}\log\frac{n_{ij}}{n} + \sum_{i=1}^{k}\sum_{j=1}^{l}\sum_{c=(a_i+b_j-n)^+}^{\min(a_i,b_j)} \frac{a_i!b_j!(n-a_i)!(n-b_j)!}{n!c!(a_i-c)!(b_j-c)!(n-a_i-b_j+c)!}\frac{c}{n}\log\frac{c}{n}. \tag{4.6}$$

The time complexity of this formula, which is dominated by the second term, is in $O(\max(k, l)n)$ [40]. In particular, it is linear in the number of samples n.

Interestingly, we can similarly assess the quantity of information $q(A)$ contained in clustering A through the adjusted entropy $\Delta H(X)$ of the corresponding random variable X. This is the information contained in A not due to randomness. We have $q(A) \geq 0$ and, by Proposition 3, $q(A) = 0$ if and only if clustering A is trivial, that is, it consists of a single cluster or of n clusters (one per sample).

Since $q(A) = s(A, A)$, it follows from (4.6) that:

$$q(A) = -\sum_{i=1}^{k}\frac{a_i}{n}\log\frac{a_i}{n} + \sum_{i,j=1}^{K}\sum_{c=(a_i+a_j-n)^+}^{\min(a_i,a_j)} \frac{a_i!a_j!(n-a_i)!(n-a_j)!}{n!c!(a_i-c)!(a_j-c)!(n-a_i-a_j+k)!}\frac{c}{n}\log\frac{c}{n}. \tag{4.7}$$

This formula's time complexity, also dominated by the second term, is in $O(kn)$. Again, this complexity is linear in the number of samples n.

Pairwise adjusted mutual information(PAMI). A measure of similarity $s_{\mathrm{p}}(A, B)$ between clusterings A and B, based on the pairwise adjusted mutual information $\Delta_{\mathrm{p}} I(X, Y)$ between the corresponding random variables X and Y.

---

[1]Recall that we don't normalize the metric, see Remark 1.

$$
\begin{aligned}
s_{\mathrm{p}}(A,B) = {} & 2\sum_{i=1}^{k}\sum_{j=1}^{l}\frac{n_{ij}(n-a_i-b_j+n_{ij})}{n^2} \\
& \times\left(\frac{n_{ij}}{n}\log\frac{n_{ij}}{n}-\frac{n_{ij}-1}{n}\log\frac{n_{ij}-1}{n}\right) \\
& + 2\sum_{i=1}^{k}\sum_{j=1}^{l}\frac{(a_i-n_{ij})(b_j-n_{ij})}{n^2} \\
& \times\left(\frac{n_{ij}}{n}\log\frac{n_{ij}}{n}-\frac{n_{ij}+1}{n}\log\frac{n_{ij}+1}{n}\right).
\end{aligned}
\tag{4.8}
$$

The time complexity of this formula is in $O(kl)$, like mutual information. It is independent of the number of samples n, given the contingency matrix. The time complexity reduces to $O(m)$ the number of non-zero entries of the contingency matrix, provided the latter is stored in sparse format.

$$
\begin{aligned}
s_{\mathrm{p}}(A,B) = {} & 2\sum_{i,j:n_{ij}>0}\frac{n_{ij}(n-a_i-b_j+n_{ij})}{n^2} \\
& \times\left(\frac{n_{ij}}{n}\log\frac{n_{ij}}{n}-\frac{n_{ij}-1}{n}\log\frac{n_{ij}-1}{n}\right) \\
& + 2\sum_{i,j:n_{ij}>0}\frac{(a_i-n_{ij})(b_j-n_{ij})}{n^2} \\
& \times\left(\frac{n_{ij}}{n}\log\frac{n_{ij}}{n}-\frac{n_{ij}+1}{n}\log\frac{n_{ij}+1}{n}+\frac{1}{n}\log\frac{1}{n}\right) \\
& - 2\left(n^2-\sum_{i=1}^{k}a_i^2-\sum_{j=1}^{l}b_i^2+\sum_{i,j:n_{ij}>0}n_{ij}^2\right) \\
& \times\frac{1}{n}\log\frac{1}{n}.
\end{aligned}
$$

Similarly, we can define the quantity of information $q_{\mathrm{p}}(A)$ in clustering A through the pairwise adjusted entropy $\Delta_{\mathrm{p}}H(X)$ of the corresponding random variable X. Again, $q_{\mathrm{p}}(A) \geq 0$, with $q_{\mathrm{p}}(A) = 0$ if and only if clustering A is trivial.

$$
\begin{aligned}
q_{\mathrm{p}}(A) = {} & 2\sum_{i=1}^{k}\frac{a_i(n-a_i)}{n^2} \\
& \times\left(\frac{a_i}{n}\log\frac{a_i}{n}-\frac{a_i-1}{n}\log\frac{a_i-1}{n}-\frac{1}{n}\log\frac{1}{n}\right).
\end{aligned}
$$

Note that the time complexity of this formula in $O(k)$. It only depends on the number of clusters k, and not on the number of samples n.

4.4.1 Properties

The intrinsic information of clustering A, denoted by $h(A)$, is defined by $\Delta H(X)$. It is bounded by $\log K$. By Proposition 3, $h(A) \geq 0$, with equality if and only if $K = 1$ or $K = n$. We conjecture that for large n, the optimal clustering consists of $\sqrt{n}$ clusters of size $\sqrt{n}$.

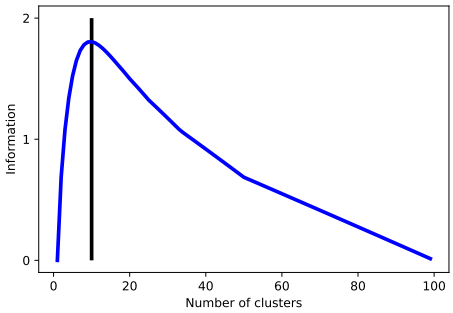


Figure 4.1: Intrinsic information of a clustering of $n = 100$ items into k clusters, with respect to k.

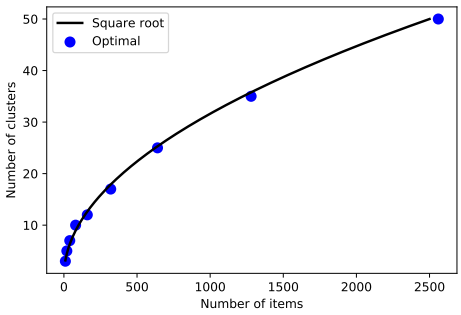


Figure 4.2: Optimal number of clusters with respect to n.

Best cuts. Find the cut that maximizes $h(A)$ for a given tree T. Find the best cuts, so that similarity $s(A, B)$ is maximum for two trees. Both benefit from the local change formulas 4.6 – 4.7. The trivial cuts leading to $K = 1$ or $K = n$ clusters are ruled out by the metric.

We justify the best cut property by the following experiments. Figure 4.1 represents the type of experiment in where a set of data points are generated at random with total size $n = 100$. Then we partition it in K clusters from the range $\{1, \ldots, 100\}$. We measure adjusted entropy for each set of labels and repeat the experiment 50 times with different randomisation and average results. Thereafter, it is clear that the peak of shared information to the number of clusters equals 10, a squared root of 100.

The second experiment is conducted similarly but, instead, we consider datasets of different size n. We partition data points in clusters $K \in \{1, \ldots, n\}$ and calculate adjusted entropy for each set of labels. To identify an optimal number of clusters, we take the maximum index of all adjusted entropy scores. Finally, we can see in Figure 4.2

that the optimal number of clusters and the squared root function have almost identical behaviour.

Similarity. The similarity $s(A, B)$ between clusterings A and B is defined by $\Delta I(X, Y)$. This is the common information shared by A and B not due to noise. It is bounded by $\min(\log K, \log L)$. It can be negative if A and B share less information than noise. Observe that $s(A, A) = h(A)$. By Theorem 1, refining a clustering cannot increase similarity: if $L > K > 1$ and each $B_j \in B$ is included in some $A_i \in A$, then $s(A, B) < s(A, A)$. By Theorem 2, two independent refinings of a clustering cannot increase similarity: if B' is another clustering with L' clusters, such that $\max(L, L') > K > 1$, each $B_j \in B$ and each $B'_j \in B'$ are included in some $A_i \in A$, and $B_i \cap B'_j \in \{\emptyset, B_i, B'_j\}$, then $s(B, B') < s(A, A)$.

4.5 Tree Mutual Information

4.5.1 Algorithm

Algorithm 3: Tree Mutual Information(TMI).

Input: initial assignment C_1 and C_2, trees T_1 and T_2, set of nodes $V_1 = \{root(T_1)\}$ and $V_2 = \{root(T_2)\}$, maximum score $S^{max} = -1$

Output: Score S^{max}

```
// returns a maximizing pair of nodes and a corresponding clustering
S, T1, T2, C1, C2 = split(V1, V2, C1, C2);
// stop criteria
if S < S^max then
    return S^max
end if
// updates sets of nodes
V1 ← V1\T1; V1 ← V1 ∪ cut(T1);
V2 ← V2\T2; V2 ← V2 ∪ cut(T2);
return TMI(C1, C2, V1, V2, S)
```

The algorithm takes as input a pair of dendrograms D_1 and D_2 with the same number of leaves n and performs the following steps:

1. Convert dendrograms D_1 and D_2 into a Newick format [20] - representation for trees T_1 and T_2 respectively.

2. Initialize 2 sets of nodes which we use to compare on every step: $V_1 = root(T_1)$ and $V_2 = root(T_2)$ for each of tree. Also initialize $S^{max} = -1$, C_1^{max} and C_2^{max} as trivial clustering.

3. Consider the clustering C_1 induced by the top level of T_1 and the clustering C_2 induced by the top level of T_2 (top cuts). Compute $S = similarity(C_1, C_2)$, where $similarity(C_1, C_2)$ is one of our metrics: AMI defined by Equation 4.6 or PAMI - Equation 4.8. If $S > S^{max}$ then update C_1^{max} and C_2^{max}.

Algorithm 4: Split operation.

Input: C_1, C_2, V_1, V_2

Output: returns maximum value, a maximizing pair of nodes and a corresponding clustering $S^{max}, V_1^{max}, V_2^{max}, C_1^{max}, C_2^{max}$

```
for node_1 ∈ V_1 do
    C_1 ← clustering(node_1) ;
    for node_2 ∈ V_2 do
        C_2 ← clustering(node_2) ;
        S = similarity(C_1, C_2) ;
        if S > S^max then
            C_1^max, C_2^max, V_1^max, V_2^max = C_1, C_2, V_1, V_2 ;
        end if
    end for
end for
return S^max, C_1^max, C_2^max, V_1^max, V_2^max ;
```

4. Recursively repeat: whenever S increases, change C_1 by going down in T_1 (lower cut). $C_1 = A \cup B$ where A and B are subclusters of C_1. Choose the best option from the set C_1, A, B which maximizes of S. Repeat the same for C_2: whenever S increases, change C_2 by going down in T_2.

5. Stop when S cannot be increased. Return S^{max}.

4.5.2 Complexity

To provide a closed form solution of algorithm complexity, we need to solve a discrete optimization problem that is not trivial. Hence, we are going to derive an approximated bound for the new metrics.

Let's consider trees $T1$ and $T2$ with n-leaves. Let $A_i = \{A_{i1}, \ldots, A_{ik_i}\}$ and $B_j = \{B_{j1}, \ldots, B_{jl_j}\}$ be two partitions on trees' levels i, j of some finite set $\{1, \ldots, n\}$ into k_i and l_j clusters, respectively. According to Property 4.4.1 the optimal numbers of clusters k and l are not exceeded by $\sqrt{n}$ in practice. We can get an approximation of complexity on the following parts:

1. Complexity of the metric: for AMI - $O(\max(k, l)n)$ and for PAMI - $O(kl)$.

2. Complexity of *split* operation includes two nested loops of total size $\mid V_1 \mid\mid V_2 \mid$ which on each step calculate metric AMI or PAMI. Now using Property 4.4.1 we can roughly estimate a maximum number of nodes considered in *split* operation. They are of size $\mid C_1 \mid < \sqrt{n}$,$\mid C_2 \mid < \sqrt{n}$ and that is why a combined complexity turns into $O(n)$. Therefore, depending on the metric, overall complexity of *split* operation equals to $O(\max(k, l)n^2) \approx O(n^2))$ for AMI and $O(kln) \approx O(n)$ for PAMI.

3. The recursion $T(\sqrt{n}) = T(\sqrt{n}-1)+O(split)$ goes no further than one level deeper per step into trees structure. Finally, it results in the overall complexity:

$$T(n) = O(n^{2.5}) \tag{4.9}$$

for a Tree Mutual Information with AMI metric (TAMI)

$$T(n) = O(n^{1.5}) \tag{4.10}$$

and for Tree Mutual Information with PAMI metric (TPAMI).

4.5.3 Implementation details

Obviously, the complexity of TAMI (4.9) is intractable for large graphs. Given this, we propose several options to reduce this complexity.

Sequential updates We can reduce AMI's complexity by one order n making sequential updates on every step of *split* algorithm instead of calculating it every time from scratch. Let's introduce sequential updates for the Mutual Information (MI) as well as for the Expected Mutual Information (EMI), which are defined in Formula 4.6 as first and second term respectively. This is possible because we know precisely that we move maximum one level down into the tree hierarchy per step, local updates are sufficient.

Variables X and Y are defined in the previous section and we introduce a new $Y^{'}$ random variable, which is obtained as an expanded version of Y by going one level down in the tree. Let's assume that cluster m was subdivided into $m^{'}$ and $m+1^{'}$ respectively. Consequently, in order to calculate probability for the new partitioning we should subtract $P(Y_m)log(P(Y_m))$ associated with cluster m and add probability for new sub-clusters $m^{'}$ and $m+1^{'}$. It results in the following updated equations:

$$H(Y^{'}) = H(Y) - P(Y_m)log(P(Y_m)) + P(Y_m^{'})log(P(Y_m^{'})) + P(Y_{m+1}^{'})log(P(Y_{m+1}^{'}))$$

$$\begin{aligned} H(X,Y^{'}) = H(X,Y) - \sum_{i=1}^{|X|} P(X_i,Y_m)log(P(X_i,Y_m)) + \\ + \sum_{i=1}^{|X|} P(X_i,Y_m^{'})log(P(X_i,Y_m^{'})) + \sum_{i=1}^{|X|} P(X_i,Y_{m+1}^{'})log(P(X_i,Y_{m+1}^{'})) \end{aligned}$$

$$MI(X,Y^{'}) = MI(X,Y) - P(Y_m)log(P(Y_m)) + P(Y^{'}_m)log(P(Y^{'}_m))+ \\ + P(Y^{'}_{m+1})log(P(Y^{'}_{m+1})) + \sum_{i=1}^{|X|} P(X_i,Y_m)log(P(X_i,Y_m))- \\ - \sum_{i=1}^{|X|} P(X_i,Y^{'}_m)log(P(X_i,Y^{'}_m)) - \sum_{i=1}^{|X|} P(X_i,Y^{'}_{m+1})log(P(X_i,Y^{'}_{m+1}))$$

Similarly, for EMI: $Y' = [Y_1, \dots Y^{'}_m, Y^{'}_{m+1}, \dots Y_l]$. In order to have a new score $EMI(X,Y^{'})$, we need to update following components: $b = [b_1, \dots b_m, \dots b_l]$ to $b = [b_1 \dots b^{'}_m, b^{'}_{m+1}, \dots b_l]$. Let's define this new notation:

$$f(b_m) = \sum_{i=1}^{k} \sum_{c_m=(a_i+b_m-n)^+}^{\min(a_i,b_m)} \frac{a_i!b_m!(n-a_i)!(n-b_m)!}{n!c_m!(a_i-c_m)!(b_m-c_m)!(n-a_i-b_m+c_m)!} \frac{c_m}{n} \log\left(\frac{c_m}{n}\right) \tag{4.11}$$

Then $EMI(C_1, C^{'}_2)$ can be calculated as:

$$EMI(C_1, C^{'}_2) = EMI(C_1, C_2) - f(b_m) + f(b^{'}_m) + f(b^{'}_{m+1}) \tag{4.12}$$

By the same logic, we can obtain updates for a.

Cashing Complexity can similarly be reduced by the use of cashing. Because of the construction of Equation 4.6 this can be efficiently performed. So as to calculate TAMI we employ Formula 4.6. The time complexity of this formula is dominated by the second term equal to $O(\max(k,l)n)$. We can rewrite Equation 4.6

$$\sum_{i=1}^{k} \sum_{i=j}^{l} f(a_i, b_j, n)$$

as stated in Equation 4.13. It only depends on parameters a_i, b_j and n. As a result, we discard term n in summation which reduces the overall complexity to $O(kl)$. This trick significantly speedups computation by summing up over cashed terms during execution.

$$f(a_i, b_j, n) = \sum_{c=(a_i+b_j-n)^+}^{\min(a_i,b_j)} \frac{a_i!b_j!(n-a_i)!(n-b_j)!}{n!c!(a_i-c)!(b_j-c)!(n-a_i-b_j+c)!} \frac{c}{n} \log \frac{c}{n}. \tag{4.13}$$

Chapter 5

Experiments and Evaluation

This section aims at evaluating the proposed metric in Chapter 4 against state of the art metrics for trees comparison mentioned in Chapter 3. We test these metrics in various scenarios and datasets: synthetic datasets give us a possibility to better understand the behaviour of metrics on simple examples and quantitatively measure the quality. In contrast, real datasets help to understand the scaling and generalisation potential of the metric regarding the requirements introduced in Chapter 3. All results are scaled to lay in the interval [0,1].

5.1 Experimental Setup

Hardware stack. The experiments are run on a computer equipped with an Intel Core i7 8-Core Processor and 16 GB of RAM, with a Ubuntu 20.04 OS.

Software stack. For the practical implementations of the algorithms, metrics, and experiments, the Python language was chosen. We also rely on open source libraries and datasets. For example, the ETE Toolkit [41] was used to encapsulate a tree structure. Other beneficial libraries which contain many metrics and visualisations implemented for clustering are scikit-learn [42] and scipy [43]. We used state of the art hierarchical clustering algorithms for graphs which are efficiently implemented in scikit-network package [44].

5.2 Syntactic data

Simple structure In this experiment, we generated base elements from more complex dendrograms and trees. The goal is to understand the behaviour of selected metrics on simple structures like the ones displayed in Figure 5.1 and apply them for analysis of more complex scenarios. In Figure 5.1(a) and 5.1(d) we can see a binary tree with 4 leaves, the only difference between these 2 trees is that we switched right and left subtrees. Figure 5.1(b) represents a trivial tree that does not contain any information

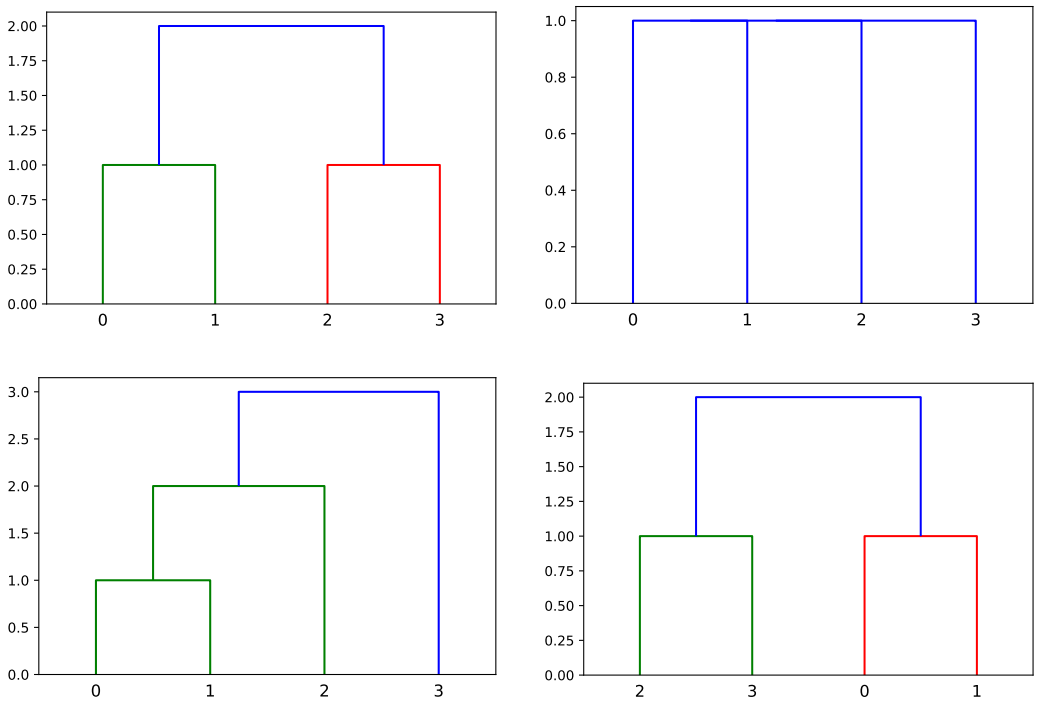


Figure 5.1: Simple Experiment - example dendrograms with different structure.

since each node either represents a separated cluster with one element or they all belong to one cluster with 4 elements. Finally, in Figure 5.1(c) we see a caterpillar tree that on every level adds one node to an already existing cluster.

Table 5.1: Simple experiment - similarity matrix.

(a) TAMI.

	a	b	c	d
a	0.46	-1	0.29	0.46
b	-1	-1	-1	-1
c	0.29	-1	0.36	0.29
d	0.46	-1	0.29	0.46

(b) TPAMI.

	a	b	c	d
a	0.17	-1	0.09	0.17
b	-1	-1	-1	-1
c	0.09	-1	0.09	0.09
d	0.17	-1	0.09	0.17

(c) RF.

	a	b	c	d
a	1	0	1	1
b	0	0	0	0
c	1	0	1	1
d	1	0	1	1

(d) ZSS.

	a	b	c	d
a	1	0.83	0.79	0.71
b	0.83	1	0.83	0.5
c	0.79	0.83	1	0.5
d	0.71	0.5	0.5	1

The experiment's results of comparing similarity in all-to-all settings can be seen in Table 5.1. As shown, TAMI and TPAMI metrics have very similar results, different only by a scale factor. Flat clustering Figure 5.1(b) receives 0 scores in all steps because it does not have any information in its structure. TAMI and TPAMI results in the same score to 5.1(a) and 5.1(d) as it was expected since they hold the same structure. The only noticeable contrast is the similarity between the caterpillar tree and binary tree: TAMI gives a lower score to the binary tree than to the caterpillar and identifies 3 clusters in both trees as an optimal number of clusters, shown in Table 5.2. According to TPAMI's results in Table 5.1(b), it tends to stop early and going less deep into the tree's structure.

RF results in Table 5.1(c) look very binary: we only see values 1 and 0. RF metric is not able to precisely capture differences in structures of the given trees.

The last metric used for consideration was TED: it shows better results than RF by identifying trees' structural differences. However, a noticeable drawback its disability to adjust for different cluster's orders. Besides, dendrograms 5.1(a) and 5.1(d) have distinct similarities. Also, this metric finds similarities between trivial clustering and other trees, which is not desired.

Table 5.2: Simple experiment - optimal number of clusters.

(a) TAMI.

	a	b	c	d
a	(2, 2)	(0, 0)	(3, 3)	(2, 2)
b	(0, 0)	(0, 0)	(0, 0)	(0, 0)
c	(3, 3)	(0, 0)	(2, 2)	(3, 3)
d	(2, 2)	(0, 0)	(3, 3)	(2, 2)

(b) TPAMI.

	a	b	c	d
a	(2, 2)	(0, 0)	(2, 3)	(2, 2)
b	(0, 0)	(0, 0)	(0, 0)	(0, 0)
c	(3, 2)	(0, 0)	(2, 2)	(3, 2)
d	(2, 2)	(0, 0)	(2, 3)	(2, 2)

The following two instances are intended to demonstrate the properties of TAMI and TPAMI metrics.

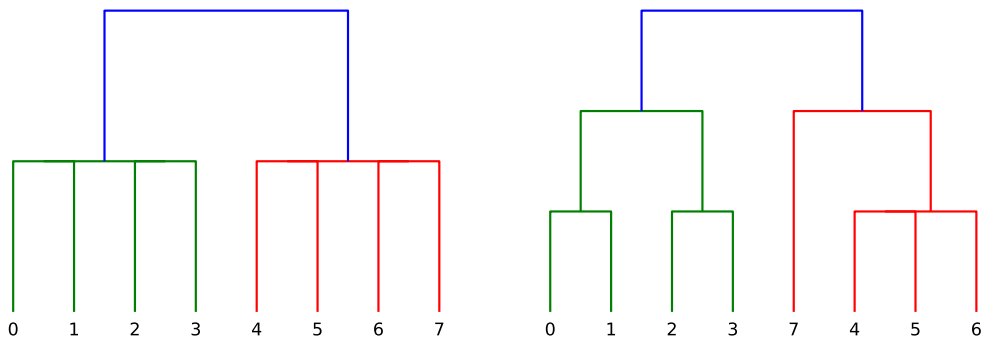


Figure 5.2: Simple Experiment - tree 5.2(a) is a subtree of 5.2(b).

First, let's consider that one tree is a subtree (sub clustering) of the second tree, in Figure 5.2 we see that if we collapse all subtrees in 5.2(b) then it will have the same structure as 5.2(a), so we can say that one tree is a subcase of another. According to Information Theory, level 2 in tree 5.2(a) contains all information. Going deeper in any subtrees will not contribute any additional information, which results in the optimal number of clusters equal to 2 for both trees. We observe this behaviour by using TAMI and TPAMI on these trees: both metrics show the clusters' optimal number of 2 and similarity 0.62 and 0.14, respectively. Although both scores are not normalised and give only a relative understanding of similarity, our metrics can correctly identify the optimal number of clusters.

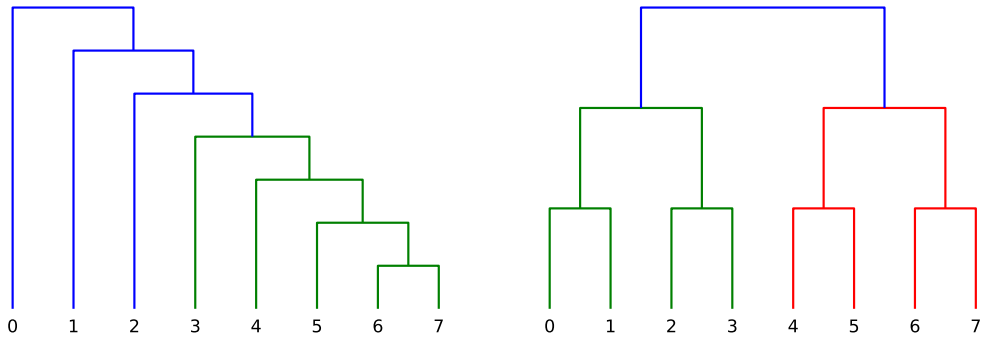


Figure 5.3: Simple Experiment - caterpillar and fully binary trees with 8 leaves.

Finally, we compare a caterpillar tree with a fully binary tree with 8 leaves. Intuitively, the best clustering is achieved by expanding all nodes in the left subtree of tree 5.3(b) and keeping the right(left) subtree as one complete cluster. It results in 5 clusters: 4 clusters with 1 element and one cluster of 4. Simultaneously, in the caterpillar tree 5.3(a) by sequentially going down till the 5th level, we obtain an identical distribution of nodes. After executing TAMI and TPAMI metrics, we obtain 5 clusters as the optimal number for both trees.

Clustering Now let's consider a bit more complex example. This experiment is intended to demonstrate the behaviour of the Mutual Information, Adjusted Mutual Information, Pairwise Adjusted Mutual Information, Expected Mutual Information and Entropy in clustering scenario to measure the similarity between ground truth partitioning and predicted one, depending on the parameter k in K-means clustering algorithm. For this experiment, we generate $M = 10$ isotropic Gaussian blobs of equal size $n = 50$, which overall results in $N = 500$ data samples. Then we apply the K-Means algorithm with parameter k ranging from 2 to 100 with step 8. We measure similarity between predicted and ground-truth clusterings. For generalisation purposes, we repeat the experiment 10 times and average the results.

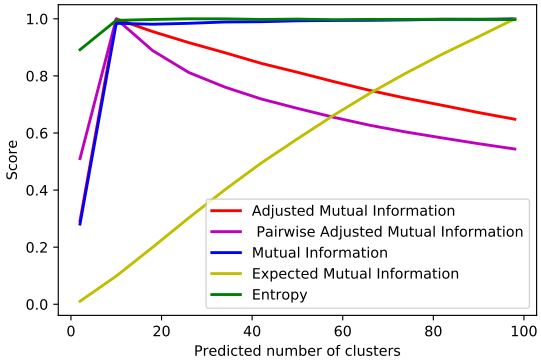


Figure 5.4: Evaluation of similarity score results on the Gaussian mixture model with $M = 10$ clusters and $n = 50$ elements in each of it.

Results in Figure 5.4 clearly show why adjustment against chance plays a crucial role for a correct measurement of information between assignments. It is worth to mention, both mutual information and its adjusted versions correctly identify an optimal number of clusters equals to 10. MI shows score similar to maximum everywhere, despite the growth of parameter k. In contrast, AMI and PAMI react correctly to the increase of noise and capture actual "information" between assignments.

Binary Trees This experiment is conducted in the following settings: a binary tree with 100 leaves is generated, and we introduce parameter k - a number of shuffled leaf pairs. Afterwards, we shuffle leaves and measure similarity between the original tree and permuted one. Additionally, the experiment is repeated $n_{repetitions} = 5$ times to generalise results better.

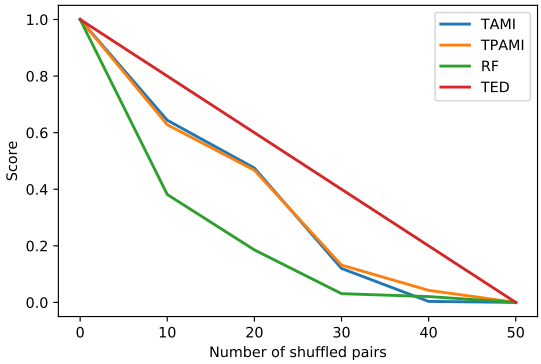


Figure 5.5: Binary trees with $n = 100$ leaves: dependence of the similarity score to the parameter k - shuffled leaf pairs.

Table 5.3: Binary trees - Pearson correlation between number of shuffled leaf pairs and values of the corresponding metric.

TAMI	TPAMI	RF	TED
-0.96242	-0.963868	-0.863864	-1.0

Similarity score results can be found in Figure 5.5. As we can see, all algorithms capture the trend correctly: with an increasing number of shuffled leaf pairs k, the similarity between trees decreases. It is critical to highlight that experiments are conducted on the same tree structure, only leaf names are shuffled. This influences explicitly TED metrics' behaviour, which in this scenario turns into a string-to-string problem and shows perfect performance. Later, we see that it is not the case when tree structure varies.

To prove quantitatively the statement above, we measure Pearson correlation between metrics' results and noise. Since the noise increases and we expect similarity to decrease it's growth, the correlation is negative see Table 5.3. It demonstrates that the TED metric perfectly correlates with noise, while TAMI and TPAMI have scores around -0.96 which is almost exact. RF has the worst performance with the value of -0.86.

The run time spent by each metric is depicted in Figure 5.6. The TED metric clearly has a considerably higher time complexity than others. The time complexities of TAMI and TPAMI are two and six times smaller than those of the TED metric, respectively.

The following experiment shows how time complexity changes with the number of leaves n. We randomly generate a pair of binary trees with n leaves and measure the time needed for the metrics to be calculated. To generalise the results, we repeat the experiment 10 times.

- TED metric has the highest time complexity on relatively small trees Figure 5.7(a). As explained in Chapter 3, it depends on the number of leaves and each tree's depth. Tree depth equals $log(n)$ for binary trees, which results in high complexity.
- While TAMI is significantly faster in comparison to TED, it is one order slower than TPAMI Figure 5.7(b).
- TPAMI and RF metrics have similar performance.

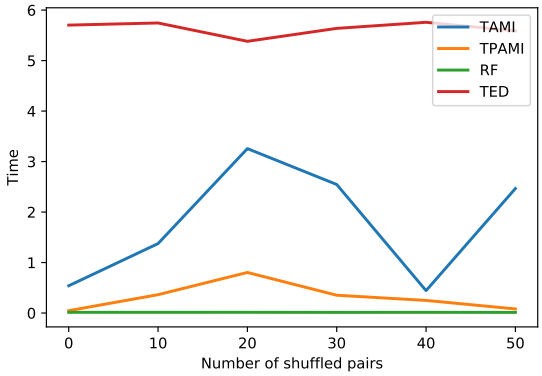


Figure 5.6: Binary trees with $n = 100$ leaves: dependence of the time complexity to the parameter k - shuffled leaf pairs.

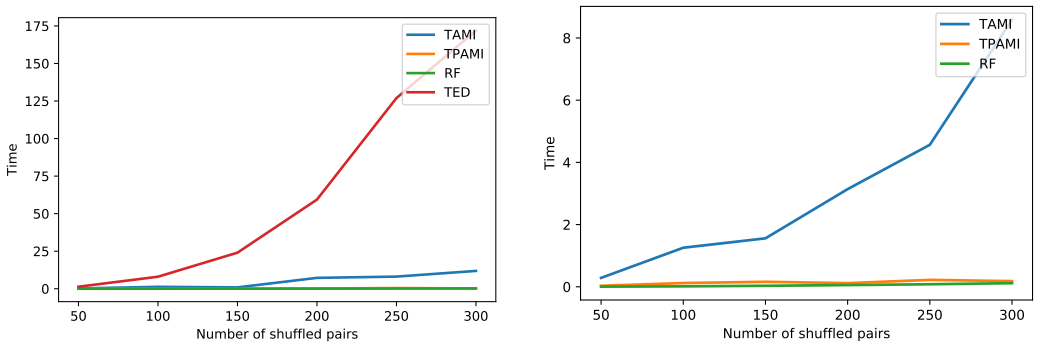


Figure 5.7: Time complexity between two randomly generated binary trees depending on the number of leaves n.

General Trees This is a similar experiment, but instead of binary trees, general trees are considered.

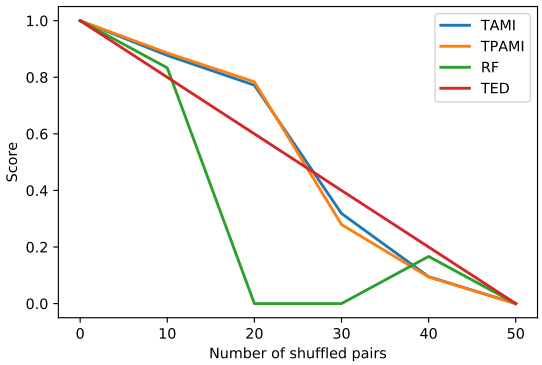


Figure 5.8: General trees with $n = 100$ leaves: dependence of the similarity score to the parameter k - shuffled leaf pairs.

Table 5.4: General trees - Pearson correlation between number of shuffled leaf pairs and values of the corresponding metric.

TAMI	TPAMI	RF	TED
-0.97592	-0.970041	-0.814345	-1.0

Results from Figure 5.8 and Table 5.4 gives us the following insights:

- TED, TAMI, and TPAMI metrics show high correlation scores, meaning that similarity consistently decreases with the growing number of permutations.
- In contrast to binary experiment, RF metric does not behave coherently: it drops to 0 with a small amount of noise and jumps when the number of permutations increases. Overall, its performance is unstable, indicating that RF fails to capture dependency between the original tree and the tree with noise.

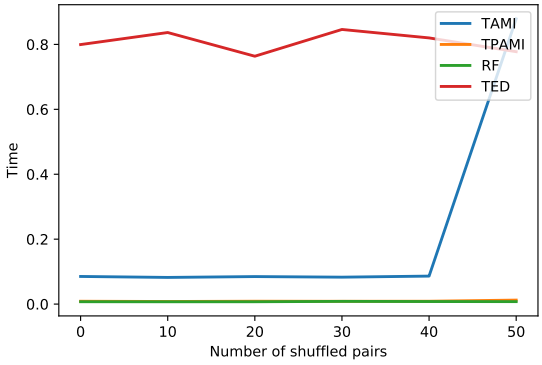


Figure 5.9: General trees with $n = 100$ leaves: dependence of the time complexity to the parameter k - shuffled leaf pairs.

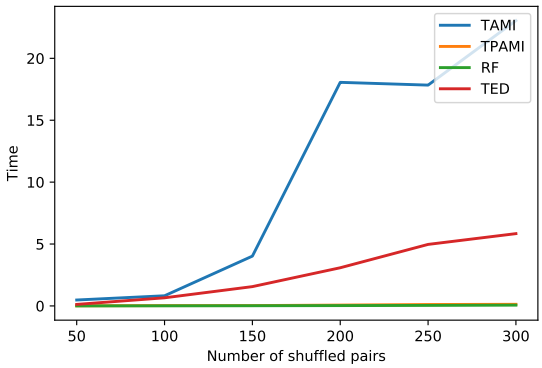


Figure 5.10: Time complexity between two randomly generated general trees depending on the number of leaves n.

Time complexity results show us the following:

- All metrics have lower time complexities on general trees Figure 5.10 when compared to binary trees Figure 5.7 .
- TAMI has the highest complexity and proves its theoretic estimation of $O(n^{2.5})$.
- Due to lower tree depth of general trees, time complexity for TED is better than in previous experiment and outperforms that of TAMI.
- TPAMI and RF have the lowest complexities in comparison to others.

Stohastic Block Model The Stochastic Block Model (SBM) is a generative model that produces graphs with communities. The SBM creates graphs with n nodes $1 \dots n$ that are grouped into k sets $\{C_k, \dots, C_k\}$. These graphs are generated from a symmetric matrix of edge probabilities $P = \{P_{rs}\}_{1 \leq r,s \leq k}$. A unitary weight with probability P_{rs} binds the nodes $i \in C_r$ and $j \in C_s$.

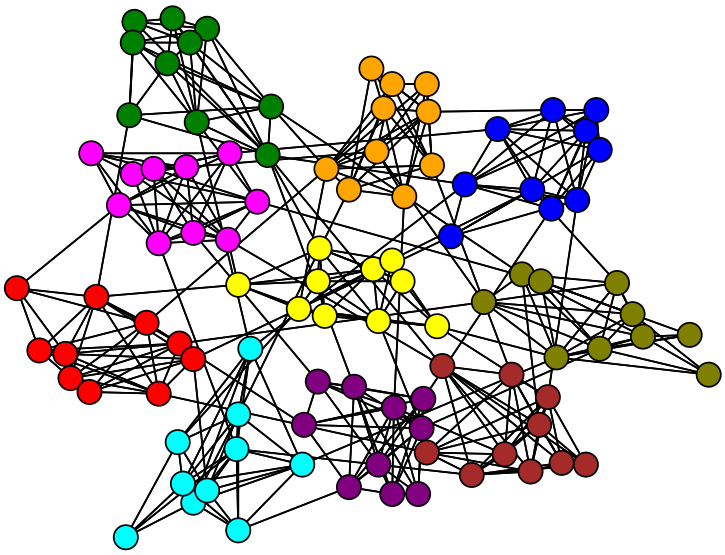
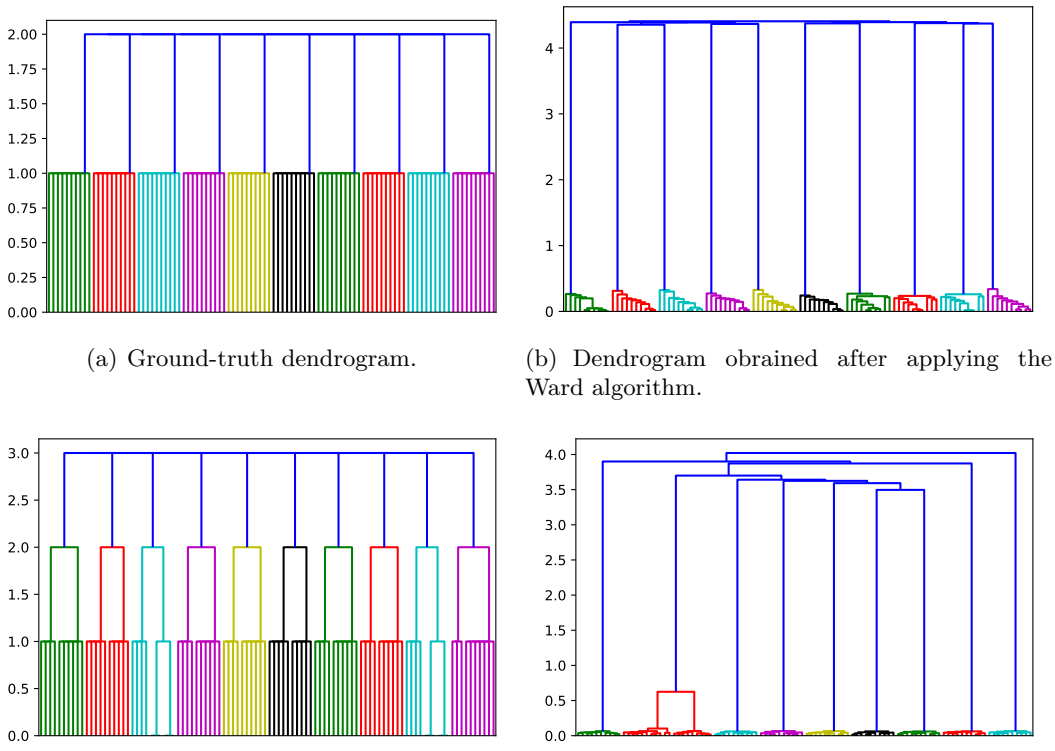


Figure 5.11: SBM graph with $n = 100$ nodes, $p_{in} = 1$, $p_{out} = 0.01$ and $K = 10$ classes which are uniformly distributed. There are 381 edges with an avarage degree of 7.62.

We want to apply hierarchical clustering algorithms discussed earlier in Section 2.2.2 and show how TMI can detect tree similarity with an increasing amount of noise. Firstly, we generate a graph G_{origin} with $n = 100$ nodes and $K = 10$ clusters with $p_{in} = 1$ probability of having edges inside the community and $p_{out} = 0.01$ to connect to other clusters. The resulting graph, has 381 edges and an average degree of 7.62. Visualisation can be found in Figure 5.11. Then we construct a ground truth hierarchy of the given graph represented as a dendrogram D_{origin}, which for simplicity only has one level and 10 clusters with 10 nodes in each of them. After applying clustering algorithms: Ward, Louvain and Paris to the graph G_{origin}, hierarchies of different qualities are obtained Figure 5.12. We measure similarity between dendrograms produced by these algorithms and the ground truth hierarchy using metrics discussed in Chapter 3. We add noise $p_{shuffled}$ by randomly shuffling edges between nodes in the original graph G_{origin}, which turns it into a shuffled graph $G_{shuffled}$. The higher the noise, the more difference between dendrograms D_{origin} and $D_{shuffled}$. In order to achieve better generalisation, the experiment is repeated $n_{repetitions} = 5$ times.

Figure 5.13 shows similarity scores between a ground truth dendrogram for the SBM graph and shuffled one obtained using each algorithm: Ward, Louvain and Paris. Results are scaled to lay in the range [0,1].

- TPAMI has the best performance in terms of Pearson correlation see Table 5.5.
- TMI with AMI and PAMI behaves very similarly in all experiments and outperforms two other metrics. There is high similarity between the ground truth tree and the one produced by different hierarchical algorithms when the amount of noise is low. When the amount of noise in shuffled graphs increases, similarity



(a) Ground-truth dendrogram.

(b) Dendrogram obrained after applying the Ward algorithm.

(c) Dendrogram obrained after applying the Louvain algorithm.

(d) Dendrogram obrained after applying the Paris algorithm.

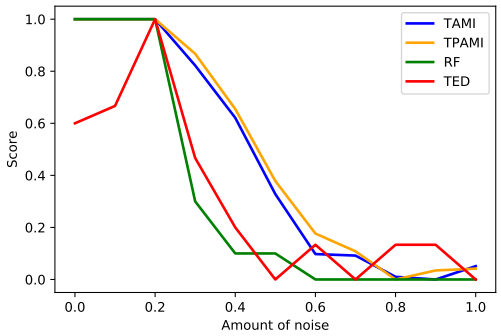
Figure 5.12: Dendrogram 5.12(a) is syntactically generated, while 5.12(b), 5.12(c), 5.12(d) obtained by applying clustering algorithms presented in Chapter 2.2.2 to the SBM graph.

gradually decreases. It shows that the newly proposed metric works well with both AMI and PAMI. These functions behave smoothly and coherently.

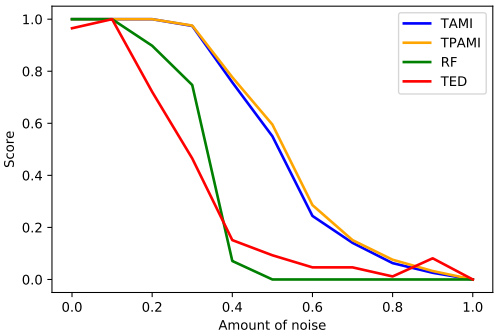
- In contrast, RF and TED behave very differently depending on the algorithm, which resembles correlation Table 5.5: both metrics show an unstable performance from one algorithm to another. For example, with the Ward clustering algorithm 5.13(a), they can capture similarity with a little amount of noise, but when noise reaches the point $p_{shuffle} = 0.4$ both metrics drop to values around 0. For the Louvain algorithm 5.13(b) Tree Edited Distance shows very high similarity even with a high amount of noise. By using the Paris clustering algorithm, the final results 5.13(c) are very unstable: RF and TED almost always show 0 similarity everywhere except for a few spikes.

In terms of time performance, we have the following results Figure 5.14:

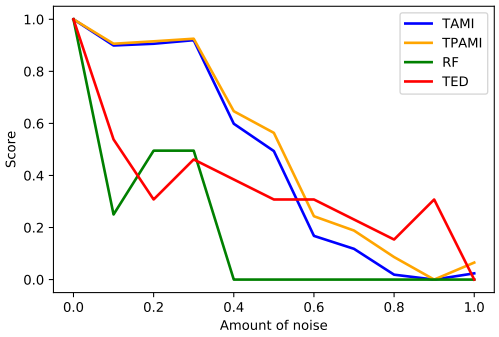
- TPAMI faster than TED and TAMI by a factor of 4 and 8 respectively.
- TAMI has the worst runtime results.



(a) Similarity results on the Ward dendrogram.



(b) Similarity results on the Louvain dendrogram.

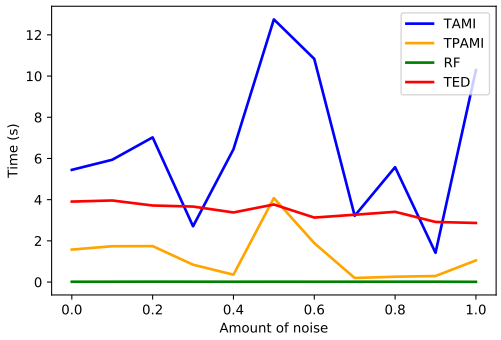


(c) Similarity results on the Paris dendrogram.

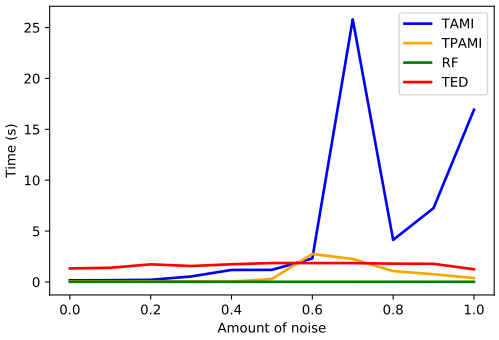
Figure 5.13: Evaluation results on the SBM graph measuring similarity between hierarchies represented by dendrograms $D_{original}$ and $D_{shuffled}$ depending on the amount of noise $p_{shuffled}$.

Table 5.5: SBM - Pearson correlation between amount of noise and values of the corresponding metric.

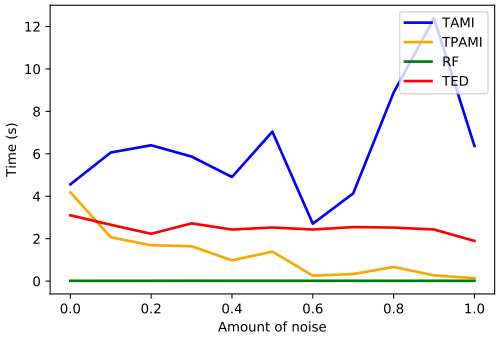
	Ward	Louvain	Paris
TAMI	-0.948200	-0.961134	-0.959607
TPAMI	-0.956005	-0.962542	-0.964492
RF	-0.857013	-0.869449	-0.771732
TED	-0.791978	-0.887756	-0.817506



(a) Runtime results on the Ward dendrogram.



(b) Runtime results on the Louvain dendrogram.



(c) Runtime results on the Paris dendrogram.

Figure 5.14: Evaluation results on the SBM graph measuring time complexity of each metric depending on the amount of noise $p_{shuffled}$.

Table 5.6: Evaluation results on the SBM graph measuring optimal number of clusters between hierarchies represented by dendrograms $D_{original}$ and $D_{shuffled}$ depending on the amount of noise $p_{shuffled}$. Tree Mutual Information with AMI and PAMI metrics are compared.

(a) TAMI.

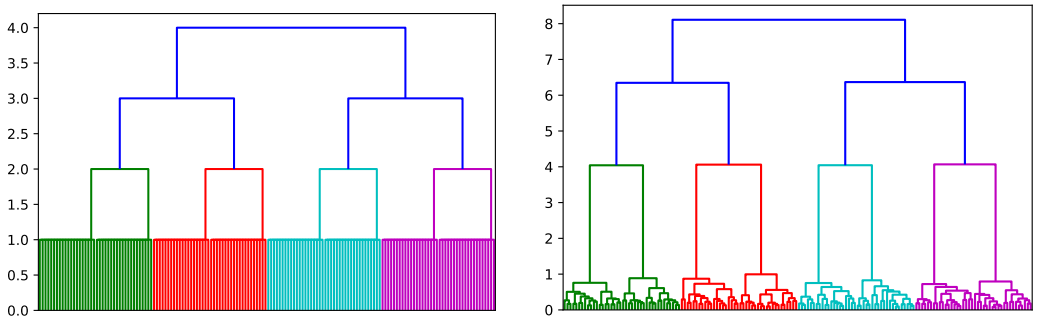
	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Ward	[10, 10]	[10, 10]	[10, 10]	[10, 8]	[10, 9]	[10, 13]	[46, 24]	[37, 16]	[64, 19]	[37, 11]	[46, 24]
Louvain	[10, 10]	[10, 10]	[10, 10]	[10, 11]	[10, 12]	[10, 12]	[19, 14]	[37, 41]	[28, 21]	[37, 25]	[55, 38]
Paris	[10, 10]	[10, 10]	[10, 10]	[10, 10]	[19, 10]	[10, 9]	[46, 13]	[37, 16]	[64, 21]	[64, 23]	[64, 23]

(b) TPAMI.

	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Ward	[10, 10]	[10, 10]	[10, 10]	[10, 8]	[10, 6]	[10, 13]	[28, 26]	[37, 7]	[46, 8]	[37, 11]	[46, 20]
Louvain	[10, 10]	[10, 10]	[10, 10]	[10, 9]	[10, 8]	[10, 12]	[10, 21]	[37, 36]	[37, 30]	[37, 24]	[46, 23]
Paris	[10, 10]	[10, 10]	[10, 10]	[10, 10]	[19, 10]	[10, 9]	[46, 10]	[37, 11]	[55, 15]	[55, 2]	[55, 9]

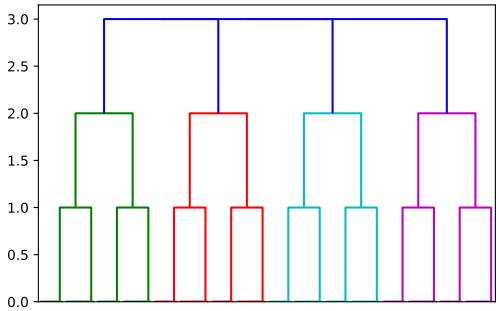
By analysing optimal number of clusters in Table 5.6 we see that:

- TAMI and TPAMI identify the optimal number of clusters correctly when the amount of noise is less than 0.3.
- With noise in a range of [0.3, 0.5] depending on algorithms metrics also shows a relatively correct number of clusters: for Ward, we see 10 vs 9 or 8 while for Louvain, it is 10 vs 12 or 11.
- When $p_{shuffled} > 0.5$, fluctuation in the optimal number of clusters is quite significant for both metrics.
- It is worth mentioning that TPAMI tends to show the smaller number and stop earlier without going too deep into a tree structure in contrast to the TAMI metric. This also highly correlates with time spent by each metric 5.14: TPAMI is significantly faster than the TAMI metric.

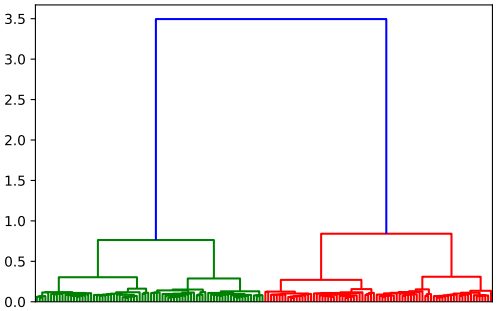


(a) Ground-truth.

(b) Ward.



(c) Louvain.

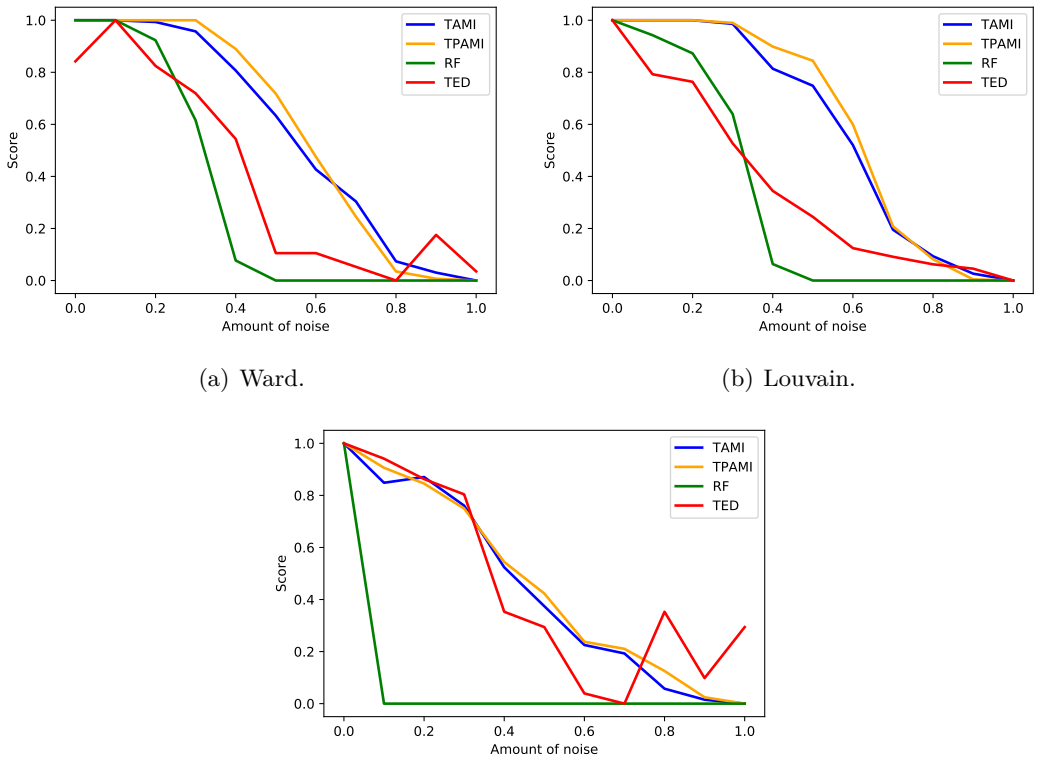


(d) Paris.

Figure 5.15: Dendrogram 5.15(a) is syntactically generated, while 5.15(b), 5.15(c), 5.15(d) obtained by applying clustering algorithms 2.2.2 to the HSBM graph.

Hierarchical Stochastic Block Model While it is effective to construct communities, the SBM 5.2 lacks a clear way to produce graphs with hierarchies. Furthermore, actual graphs often tend to have several relevant clustering scales, which motivates the concept of a hierarchical variant of SBM, as discussed in [45] and [46]. In this experiment, we consider a hierarchical stochastic block model (HSBM) with Poisson

distributed edge weights of 160 nodes and 3 levels of hierarchy (a binary tree with leaves corresponding to clusters of 20 nodes) with parameters: $decay\_factor = 0.3$, $p_{in} = 0.9$. The resulted graph has 2223 edges with an average degree of 27.6. The other settings for the experiment are the same as for 5.2. Visualisation of dendrograms can be seen in Figure 5.15.



(a) Ward.

(b) Louvain.

(c) Paris.

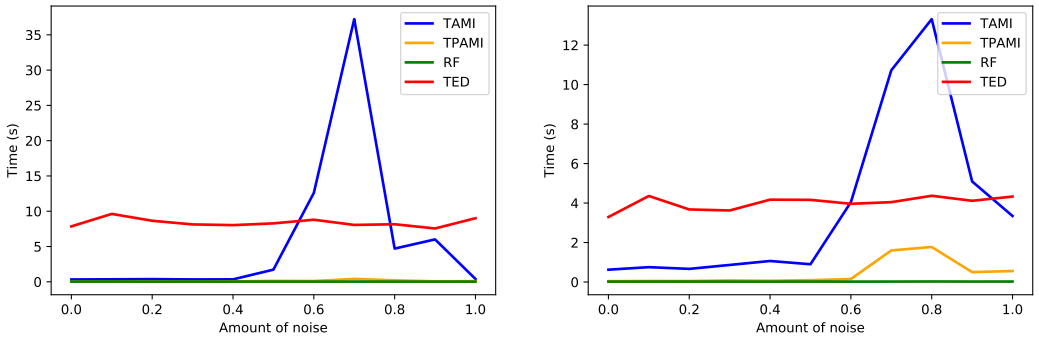
Figure 5.16: Evaluation results on HSBM graph measuring similarity between hierarchies represented by dendrograms $D_{original}$ and $D_{shuffled}$ depending on the amount of noise $p_{shuffled}$.

Table 5.7: HSBM - Pearson correlation between amount of noise and values of the corresponding metric.

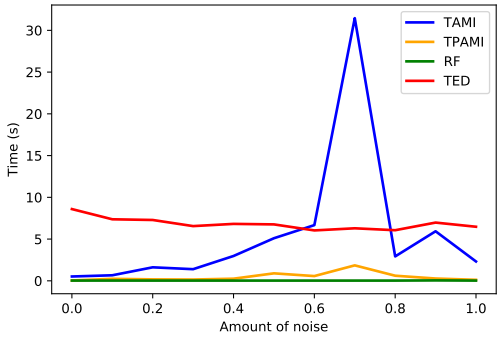
	Ward	Louvain	Paris
TAMI	-0.970748	-0.956428	-0.981013
TPAMI	-0.952583	-0.936338	-0.988256
RF	-0.871165	-0.871952	-0.500000
TED	-0.904393	-0.955369	-0.809923

The outcomes Figure 5.16 and Table 5.7 of the HSBM experiment resemble behaviour which was seen on SBM:

- Tree Mutual Information with AMI and PAMI performs much better than RF and TED. TAMI and TPAMI show correlation that is almost equal to 1.
- Unstable behaviour using Paris algorithm is seen for RF and TED: RF almost always shows 0 similarity score while TED has stepped view with the a very abrupt decline to 0 and little spikes at the end. Pearson correlation for the RF metric shows results in a range from -0.87 to -0.5, which is poor.



(a) Runtime results on the Ward dendrogram. (b) Runtime results on the Louvain dendrogram.



(c) Runtime results on the Paris dendrogram.

Figure 5.17: Evaluation results on HSBM graph measuring time complexity of each metric depending on the amount of noise $p_{shuffled}$.

Analysing time complexity results from Figure 5.17 we can see that TPAMI has nearly as good performance as the RF metric. TAMI has the worst outcomes while TED remains stable over different values on the [0,1].

Table 5.8: Evaluation results on the HSBM graph measuring optimal number of clusters between hierarchies represented by dendrograms $D_{original}$ and $D_{shuffled}$ depending on the amount of noise $p_{shuffled}$. Tree Mutual Information with AMI and PAMI metrics are compared.

(a) TAMI.

	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Ward	[8, 8]	[8, 8]	[8, 8]	[8, 8]	[7, 7]	[6, 12]	[24, 18]	[43, 40]	[33, 21]	[24, 16]	[23, 7]
Louvain	[8, 8]	[8, 8]	[8, 8]	[8, 8]	[8, 11]	[7, 13]	[15, 23]	[26, 41]	[34, 48]	[22, 47]	[13, 29]
Paris	[8, 9]	[8, 8]	[8, 11]	[8, 11]	[7, 14]	[6, 15]	[25, 16]	[17, 28]	[15, 12]	[23, 21]	[33, 15]

(b) TPAMI.

	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Ward	[4, 4]	[4, 4]	[4, 4]	[4, 4]	[5, 5]	[5, 7]	[6, 6]	[15, 12]	[23, 12]	[5, 7]	[23, 7]
Louvain	[4, 4]	[4, 4]	[4, 4]	[5, 5]	[4, 4]	[5, 5]	[5, 16]	[16, 39]	[5, 38]	[3, 31]	[4, 27]
Paris	[5, 5]	[5, 8]	[6, 8]	[6, 8]	[6, 10]	[6, 14]	[6, 11]	[17, 21]	[16, 16]	[14, 10]	[33, 8]

In terms of an optimal number of clusters Table 5.8 the case of HSBM is more difficult than SBM. There are different levels of hierarchy in ground-truth dendrogram 5.15(a): on the first level, we have 2 clusters which further are subdivided into 2 resulting in 4 clusters. On level 3, we have 8 clusters with 20 nodes each.

- TAMI and TPAMI identify an optimal number of clusters a bit differently: TAMI shows 8 clusters as optimal for Ward and Louvain when the amount of noise is less than 0.5, while TPAMI on the same settings gives 4 as an optimal number.
- We can see that TPAMI tends to early stop at level 2 of the ground truth dendrogram from Figure 5.15(a).
- However, when $p_{shuffled}$ is bigger than 0.5, fluctuation in the optimal number of clusters is quite significant for both metrics.
- It is worth mentioning that TPAMI tends to show smaller numbers and stop earlier without going too deep into a tree structure, in contrast to the TAMI metric. This highly correlates evaluation time of each metric Figure 5.14: TPAMI is significantly faster than the TAMI metric.
- The structure of the tree generated using the Paris algorithm is interpreted by both metrics differently than the one generated with Ward and Louvain. We see that TAMI predicts [8,9], while TPAMI predicts [5,5], [5,7] as optimal number of clusters.

These results may indicate that our decision to model the ground-truth dendrogram Figure 5.15(a) as a binary tree with 20 leaves in 4 levels was wrong. Therefore, we notice this instability during evaluation. This is another very useful feature of our new

metric, which can be used not only to test the quality of clustering algorithms but also to check the quality of ground-truth trees/dendrograms.

5.3 Real datasets

As stated in Section 4.5.3, we use cashing to speed up the TAMI metric computation on large datasets. To optimise its execution even more, part of the code was written in Cython [47] utilising built-in parallelisation. Additionally, in order to compute the first part of Formula 4.6 instead of the NumPy [48] version of Mutual Information, C based implementation from [49]. Utilising all these improvements we cut execution time by more than 50 times.

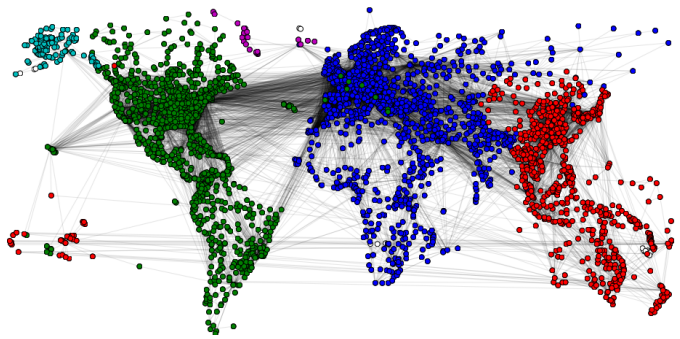
Next, we show the practical interest of TMI in terms of tree comparison. Experiments on real networks are performed on 2 datasets with various sizes and sparsity. Detailed information about these datasets can be found in Table 5.9. Both datasets are taken from [50].

Table 5.9: Summary of the 2 datasets.

Dataset	nodes	edges	average degree
OpenFlight	3097	18193	11.74
WikiVitals	10012	792091	158.22

We want to compute a similarity matrix $s(A, B)$ for trees obtained with Ward, Paris and Louvain algorithms on Openflights and WikiVitals datasets. The Ward algorithm has a GSVD embedding method with the number of components equal to 20 and regularisation of 0.1; for the Louvain algorithm we set depth equal to 20. We conduct these experiments only with metrics proposed in Chapter 4. We disregard RF and TED metrics due to bad performance in the clustering scenario and inefficiency when applied to large datasets.

Figure 5.18: OpenFlights graph [50].



Openflights OpenFlights is a weighted graph in which nodes represent airports worldwide and edge weights represent the number of flights between two airports. They dis-

play various regions around the world, like natural visual clusters including continents and countries.

Table 5.10: Openflights - trees information.

	Ward	Louvain	Paris
Number of leaf nodes	3097	3097	3097
Total number of nodes	6005	4274	6193
Most distant node	237	3053	1706
Max. distance	24	7	32

We apply all three algorithms to the dataset, convert the output to trees and obtain trees with the following statistics seen in Table 5.10. We can see that the structure of these trees is very different: they have distinct numbers of hierarchical layers varying from 7 to 32 and a different total number of nodes.

Table 5.11: Openflights - similarity matrix.

(a) TAMI.

	Ward	Paris	Louvain
Ward	3.38	2.41	2.40
Paris	2.41	3.36	2.64
Louvain	2.40	2.64	3.27

(b) TPAMI.

	Ward	Paris	Louvain
Ward	3	2.26	2.22
Paris	2.26	2.98	2.4
Louvain	2.22	2.4	2.87

We compare all these dendrograms with each other, and the results can be found in Table 5.11. We can see that TAMI Table 5.11(a) and TPAMI Table 5.11(b) identify the same similarity order between trees. For example, for the Ward tree both metrics show that the highest similarity is obtained with the identical tree, then the tree obtained by applying Paris algorithm which is more similar than Louvain. As was mentioned in Chapter 4, magnitude can vary because, we do not use normalisation.

Table 5.12: Openflights - optimal number of clusters.

(a) TAMI.

	Ward	Paris	Louvain
Ward	(64-64)	(94-68)	(120-149)
Paris	(68-94)	(65-65)	(82-130)
Louvain	(149-120)	(130-82)	(93-93)

(b) TPAMI.

	Ward	Paris	Louvain
Ward	(7, 7)	(48, 38)	(67, 72)
Paris	(38, 48)	(11, 11)	(25, 47)
Louvain	(72, 67)	(47, 25)	(35, 35)

In Table 5.12 we see the optimal number of clusters for each pair of trees. We observe behaviour to previous experiments: TPAMI tends to identify fewer clusters and stops earlier than TAMI.

Table 5.13: Openflights - time complexities (s).

(a) TAMI.

	Ward	Paris	Louvain
Ward	17	94	598
Paris	94	20	1142
Louvain	598	1142	6

(b) TPAMI.

	Ward	Paris	Louvain
Ward	1	43	111
Paris	43	3	48
Louvain	111	48	2

Table 5.13 presents time complexities of the experiment. TPAMI works much faster than the TAMI metric due to its lower complexity by construction and early stopping property.

Wikivitals The Wikivitals dataset is an unweighted graph where nodes represent Vital articles of Wikipedia with links between them and words used in summaries. We reconstruct the hierarchy from ground truth labels: there are 4 levels with the following number of clusters on each: 11, 109, 491, 1164. We convert the ground truth dendrogram into a tree and consider it as an additional tree in the experiment.

Table 5.14: Wikivitals - trees information.

	Ground Truth	Ward	Louvain	Paris
Number of leaf nodes	10012	10012	10012	10012
Total number of nodes	11319	20023	14231	20023
Most distant node	10011	4924	9516	7650
Max. distance	5	23	10	84

Table 5.14 presents statistics of each tree. In Table 5.15 we see the similarity matrix. It is most interesting to observe which algorithms produce hierarchical clustering most similar to the ground truth one.

Both metrics TAMI and TPAMI identify the Louvain tree as the most similar to ground truth, while Ward takes the second spot and Paris tree has the worst similarity rate. The outcome that Louvain tree is the most similar to ground truth is not surprising, because they have similar structure: Louvain only has 7 levels of hierarchy and they are subdivided similarly to the "general tree", meaning the number of clusters is not limited to be divided by 2 in every level. Both Ward and Paris tend to produce structure similar to a binary tree. Overall, results are pretty similar, the only noticeable difference is that TAMI finds the pair (Ward, Paris) less similar than (Ward, Wikivitals).

Table 5.15: Wikivitals - similarity matrix.

(a) TAMI.

	Ward	Paris	Louvain	Wikivitals
Ward	3.99	2.06	2.52	2.10
Paris	2.06	3.95	2.18	1.93
Louvain	2.52	2.18	3.92	2.15
Wikivitals	2.10	1.93	2.15	3.87

(b) TPAMI.

	Ward	Paris	Louvain	WikiVitals
Ward	3.52	1.83	2.18	1.79
Paris	1.83	3.53	1.82	1.68
Louvain	2.18	1.82	3.49	1.8
WikiVitals	1.79	1.68	1.8	3.5

Table 5.16: Wikivitals - optimal number of clusters.

(a) TAMI.

	Ward	Paris	Louvain	Wikivitals
Ward	(103-103)	(147-129)	(124-153)	(128-275)
Paris	(129-147)	(105-105)	(135-245)	(130-293)
Louvain	(153-124)	(245-135)	(129-129)	(173-300)
Wikivitals	(275-128)	(293-130)	(300-173)	(155-155)

(b) TPAMI.

	Ward	Paris	Louvain	WikiVitals
Ward	(10, 10)	(55, 96)	(44, 80)	(35, 226)
Paris	(96, 55)	(11, 11)	(96, 136)	(73, 206)
Louvain	(80, 44)	(136, 96)	(7, 7)	(72, 163)
WikiVitals	(226, 35)	(206, 73)	(163, 72)	(11, 11)

Table 5.16 shows the optimal number of clusters for each pair of trees. We see that TPAMI correctly identifies the number of clusters when we compare (Paris, Paris) and (WikiVitals, WikiVitals) pairs, which shows that the metric captures the higher-order structure of the trees correctly. Regarding TAMI, it happens to find an optimal number of clusters from deeper levels wherein the ground truth tree has 109 communities. In all cases, we see values that are close to the original ones.

Table 5.17 presents the time dimension of the experiment. We can see that TPAMI is 5 times faster than TAMI.

Table 5.17: Wikivitals - time complexities (s).

(a) TAMI.

	Ward	Paris	Louvain	Wikivitals
Ward	164	596	2257	3744
Paris	596	154	2532	2014
Louvain	2257	2532	40	1059
Wikivitals	3744	2014	1059	85

(b) TPAMI.

	Ward	Paris	Louvain	WikiVitals
Ward	4	372	117	181
Paris	372	3	1089	402
Louvain	117	1089	1	275
WikiVitals	181	402	275	1

Chapter 6

Conclusion

In this chapter, we conclude our work by answering the research questions posed in Chapter 1 and discussing possible improvements and future work.

6.1 Contribution and Future Work

We proposed a novel Information Theoretic Metric for labeled trees and another way of adjusting mutual information against chance through pairwise label permutations. It approximates optimal partitions with respect to the shared information score. As a sub-problem, we present an adjustment based on pairwise label permutations instead of full label permutations which has much lower complexity than the usual adjusted mutual information.

Experiments on synthetic and real data show that adjustment against chance plays a crucial role in correctly measuring information between partitions and, therefore, in measuring similarity between trees. Both TAMI and TPAMI show consistent and smooth results and outperform RF and TED metrics in all experiments. However, while TMI with the pairwise adjusted mutual information tends to provide the same results as with full adjusted mutual information, it involves much fewer computations and is therefore applicable to larger trees. Hence, we recommend it for primary usage.

Regarding future work and contributions, we plan to go deeper into normalisation studies [51] and implement a scaled version of the TPAMI metric. We also want to extend the TMI idea to other similarity metrics as those studied in [52].

6.2 Research Questions

Q1 How to efficiently measure the similarity between two labeled trees with the same leaf sets but different topology?

In this thesis, we considered existing state of the art metrics Chapter 3 and analyse their pros and cons and ran experiments to test them in different settings.

Q2 How well does a novel information-theoretic metric assess the quality of hierarchical clustering of different data types? What is the optimal number of clusters that maximizes similarity between two dendrograms?

Based on experimental evaluation, the TMI metric outperforms all other metrics. It is precise, efficient, and self-explanatory, making it very attractive for real-world problems. We prove Property 4.4.1 which explains an optimal number of clusters for a dendrogram from an information theoretic point of view, as well as conducts experiments that prove our theory in practice.

Q3 What are some pros and cons of the novel metric in comparison to state-of-the-art?

The TPAMI metric shows consistent and smooth results and outperforms RF and TED metrics in all experiments. It exhibits good time complexity and it is suitable for large datasets. An unsolved disadvantage, which was also mentioned as future work, is that the metric is not normalised.

Bibliography

[1] S. Canzar S. Klau G. Böcker. "The generalized Robinson-Foulds metric." In: *Algorithms in Bioinformatics* 8126.1 (2013), pp. 156–169. DOI: 10.1007/978-3-642-40453-5_13.

[2] Kaizhong Zhang and Dennis Shasha. "Simple Fast Algorithms for the Editing Distance Between Trees and Related Problems". In: *SIAM J. Comput.* 18 (Dec. 1989), pp. 1245–1262. DOI: 10.1137/0218082.

[3] David Penny, L. R. Foulds, and M. D. Hendy. "Testing the theory of evolution by comparing phylogenetic trees constructed from five different protein sequences". In: *Nature* 297.5863 (1982), pp. 197–200. ISSN: 00280836. DOI: 10.1038/297197a0. URL: https://pubmed.ncbi.nlm.nih.gov/7078635/.

[4] Sanjoy Dasgupta. "A cost function for similarity-based hierarchical clustering". In: (2016), pp. 118–127. ISSN: 07378017. DOI: 10.1145/2897518.2897527. arXiv: 1510.05043.

[5] Bertrand Charpentier and Thomas Bonald. "Tree Sampling Divergence: An Information-Theoretic Metric for Hierarchical Graph Clustering". In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, July 2019, pp. 2067–2073. DOI: 10.24963/ijcai.2019/286. URL: https://doi.org/10.24963/ijcai.2019/286.

[6] Jochen Papenbrock. "Asset Clusters and Asset Networks in Financial Risk Management and Portfolio Optimization". PhD thesis. 2011. DOI: 10.5445/IR/1000025469.

[7] Mauro Gallegati et al. "Cluster Analysis for Portfolio Optimization". In: *Journal of Economic Dynamics and Control* 32 (Feb. 2008), pp. 235–258. DOI: 10.1016/j.jedc.2007.01.034.

[8] Peter Oram. "WordNet: An electronic lexical database. Christiane Fellbaum (Ed.). Cambridge, MA: MIT Press, 1998. Pp. 423." In: *Applied Psycholinguistics* 22.1 (2001), 131–134. DOI: 10.1017/S0142716401221079.

[9] Denys Lazarenko and Thomas Bonald. *Pairwise Adjusted Mutual Information*. 2021. arXiv: 2103.12641 [cs.LG].

[10] Thomas Bonald et al. "Hierarchical Graph Clustering using Node Pair Sampling". In: (2018). arXiv: 1806.01664. URL: http://arxiv.org/abs/1806.01664.

[11] M. E. J. Newman and M. Girvan. "Finding and evaluating community structure in networks". In: *Phys. Rev. E* 69 (2 2004), p. 026113. DOI: 10.1103/PhysRevE.69.026113. URL: https://link.aps.org/doi/10.1103/PhysRevE.69.026113.

[12] M. E. J. Newman. "Fast algorithm for detecting community structure in networks". In: *Physical Review E* 69.6 (2004). ISSN: 1550-2376. DOI: 10.1103/physreve.69.066133. URL: http://dx.doi.org/10.1103/PhysRevE.69.066133.

[13] Pascal Pons and Matthieu Latapy. *Computing Communities in Large Networks Using Random Walks*. Ed. by pInar Yolum et al. Berlin, Heidelberg, 2005.

[14] N. Tremblay and P. Borgnat. "Graph Wavelets for Multiscale Community Mining". In: *IEEE Transactions on Signal Processing* 62.20 (2014), pp. 5227–5239. DOI: 10.1109/TSP.2014.2345355.

[15] J. Ward. "Hierarchical Grouping to Optimize an Objective Function". In: *Journal of the American Statistical Association* 58 (1963), pp. 236–244.

[16] Daniel Müllner. *Modern hierarchical, agglomerative clustering algorithms*. 2011. arXiv: 1109.2378 [stat.ML].

[17] Thomas Bonald. *Hierarchical clustering*. 2019.

[18] Joe H. Ward Jr. "Hierarchical Grouping to Optimize an Objective Function". In: *Journal of the American Statistical Association* 58.301 (1963), pp. 236–244. DOI: 10.1080/01621459.1963.10500845. eprint: https://www.tandfonline.com/doi/pdf/10.1080/01621459.1963.10500845. URL: https://www.tandfonline.com/doi/abs/10.1080/01621459.1963.10500845.

[19] J. Macqueen. "Some methods for classification and analysis of multivariate observations". In: *In 5-th Berkeley Symposium on Mathematical Statistics and Probability*. 1967, pp. 281–297.

[20] Olsen G. *Newick's 8:45" Tree Format Standard*. Apr. 1990. URL: http://evolution.genetics.washington.edu/phylip/newick_doc.html.

[21] Lawrence Hubert and Phipps Arabie. "Comparing partitions". In: *Journal of classification* 2.1 (1985), pp. 193–218.

[22] Nguyen Xuan Vinh, Julien Epps, and James Bailey. "Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance". In: *The Journal of Machine Learning Research* 11 (2010), pp. 2837–2854.

[23] Andrew Rosenberg and Julia Hirschberg. "V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure". In: *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 410–420. URL: https://www.aclweb.org/anthology/D07-1043.

[24] Peter J. Rousseeuw. "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis". In: *Journal of Computational and Applied Mathematics* 20 (1987), pp. 53 –65. ISSN: 0377-0427. DOI: https://doi.org/10.1016/0377-

0427(87)90125-7. URL: http://www.sciencedirect.com/science/article/pii/0377042787901257.

[25] Thomas Bonald and Bertrand Charpentier. "Learning Graph Representations by Dendrograms". In: (2018). arXiv: 1807.05087. URL: http://arxiv.org/abs/1807.05087.

[26] Vincent Cohen-Addad, Varun Kanade, and Frederik Mallmann-Trenn. "Hierarchical Clustering Beyond the Worst-Case". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017, pp. 6201–6209. URL: https://proceedings.neurips.cc/paper/2017/file/e8bf0f27d70d480d3ab793bb7619aaa5-Paper.pdf.

[27] Vincent Cohen-Addad et al. "Hierarchical clustering: Objective functions and algorithms". In: *Journal of the ACM* 66.4 (2019), pp. 1–42. ISSN: 1557735X. DOI: 10.1145/3321386. URL: https://dl.acm.org/doi/10.1145/3321386.

[28] Sebastian Böcker, Stefan Canzar, and Gunnar W. Klau. "The Generalized Robinson-Foulds Metric". In: (2013). Ed. by Aaron Darling and Jens Stoye, pp. 156–169.

[29] MK Kuhner and J Felsenstein. "A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates". In: *Molecular biology and evolution* 11.3 (1994), 459—468. ISSN: 0737-4038. DOI: 10.1093/oxfordjournals.molbev.a040126. URL: https://doi.org/10.1093/oxfordjournals.molbev.a040126.

[30] Douglas Critchlow, Dennis Pearl, and CL Qian. "The Triples Distance for Rooted Bifurcating Phylogenetic Trees". In: *Systematic Biology - SYST BIOL* 45 (Sept. 1996), pp. 323–334. DOI: 10.1093/sysbio/45.3.323.

[31] Mary Kuhner and Jon Yamato. "Practical Performance of Tree Comparison Metrics". In: *Systematic biology* 64 (Nov. 2014). DOI: 10.1093/sysbio/syu085.

[32] David T. Barnard, Gwen Clarke, and Nicolas Duncan. *Tree-to-tree Correction for Document Trees*. 1995.

[33] J. R. Rico-Juan and L. Micó. "Comparison of AESA and LAESA search algorithms using string and tree-edit-distances". In: *Pattern Recognit. Lett.* 24 (2003), pp. 1417–1426.

[34] Kuo-Chung Tai. "The Tree-to-Tree Correction Problem". In: *J. ACM* 26.3 (July 1979), 422–433. ISSN: 0004-5411. DOI: 10.1145/322139.322143. URL: https://doi.org/10.1145/322139.322143.

[35] Mateusz Pawlik and Nikolaus Augsten. "RTED: A Robust Algorithm for the Tree Edit Distance". In: *Proc. VLDB Endow.* 5.4 (Dec. 2011), 334–345. ISSN: 2150-8097. DOI: 10.14778/2095686.2095692. URL: https://doi.org/10.14778/2095686.2095692.

[36] Stefan Schwarz, Mateusz Pawlik, and Nikolaus Augsten. "A New Perspective on the Tree Edit Distance". In: *Similarity Search and Applications*. Ed. by Christian Beecks et al. Cham: Springer International Publishing, 2017, pp. 156–170. ISBN: 978-3-319-68474-1.

[37] "Pearson's Correlation Coefficient". In: *Encyclopedia of Public Health*. Ed. by Wilhelm Kirch. Dordrecht: Springer Netherlands, 2008, pp. 1090–1091. ISBN: 978-1-

4020-5614-7. DOI: 10.1007/978-1-4020-5614-7_2569. URL: https://doi.org/10.1007/978-1-4020-5614-7_2569.

[38] Thomas M Cover and Joy A Thomas. *Elements of Information Theory*. Wiley, 1991.

[39] Marina Meilă. "Comparing clusterings by the variation of information". In: *Learning theory and kernel machines*. Springer, 2003, pp. 173–187.

[40] Simone Romano et al. "Standardized mutual information for clustering comparisons: one step further in adjustment for chance". In: *International Conference on Machine Learning*. 2014, pp. 1143–1151.

[41] Jaime Huerta-Cepas, François Serra, and Peer Bork. "ETE 3: Reconstruction, Analysis, and Visualization of Phylogenomic Data". In: *Molecular Biology and Evolution* 33.6 (Feb. 2016), pp. 1635–1638. ISSN: 0737-4038. DOI: 10.1093/molbev/msw046. eprint: https://academic.oup.com/mbe/article-pdf/33/6/1635/7953632/msw046.pdf. URL: https://doi.org/10.1093/molbev/msw046.

[42] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[43] Pauli Virtanen et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* (2020).

[44] Thomas Bonald et al. "Scikit-network: Graph Analysis in Python". In: *Journal of Machine Learning Research* 21.185 (2020), pp. 1–6. URL: http://jmlr.org/papers/v21/20-412.html.

[45] Bertrand Charpentier. "Multi-scale clustering in graphs using modularity". Masters's Thesis. KTH Royal Institute of Technology, 2019.

[46] Vince Lyzinski et al. *Community Detection and Classification in Hierarchical Stochastic Blockmodels*. 2016. arXiv: 1503.02115 [stat.ML].

[47] Stefan Behnel et al. "Cython: The best of both worlds". In: *Computing in Science & Engineering* 13.2 (2011), pp. 31–39.

[48] Charles R. Harris et al. "Array programming with NumPy". In: *Nature* 585.7825 (Sept. 2020), pp. 357–362. DOI: 10.1038/s41586-020-2649-2. URL: https://doi.org/10.1038/s41586-020-2649-2.

[49] Gavin Brown et al. "Conditional Likelihood Maximisation: A Unifying Framework for Information Theoretic Feature Selection". In: *The Journal of Machine Learning Research* 13 (Feb. 2012), pp. 27–66.

[50] Thomas Bonald. *NetSet*. URL: https://netset.telecom-paris.fr/index.html (visited on 03/04/2021).

[51] Nguyen Xuan Vinh, Julien Epps, and James Bailey. "Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance". In: *Journal of Machine Learning Research* 11.95 (2010), pp. 2837–2854. URL: http://jmlr.org/papers/v11/vinh10a.html.

[52] Simone Romano et al. "Adjusting for chance clustering comparison measures". In: *The Journal of Machine Learning Research* 17.1 (2016), pp. 4635–4666.