A Novel Information Theoretic Metric for Labeled Trees

Master Thesis Presentation

Denys Lazarenko [1]

Thesis supervisor: Prof. Dr. Tobias Lasser [2]

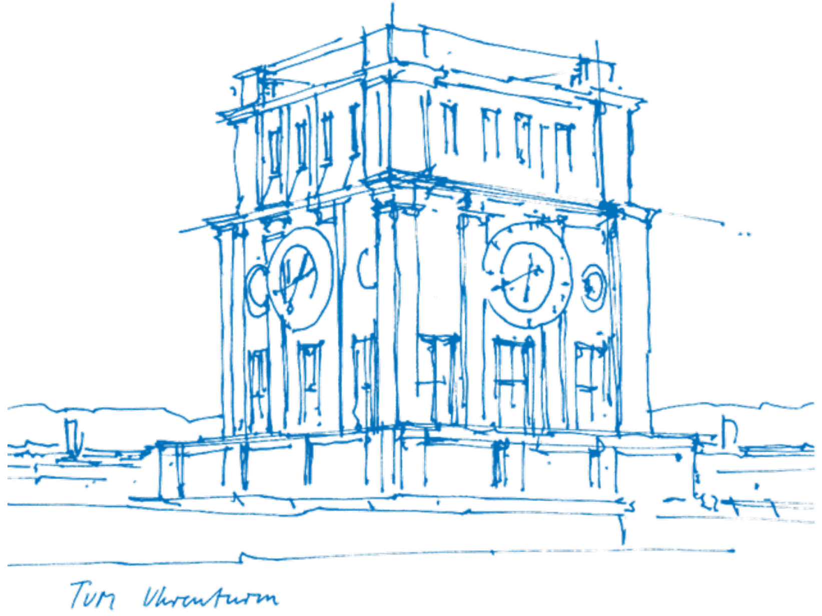
Principal Advisor: Prof. Thomas Bonald [3]

[1] Department of Mathematics, Technical University of Munich (TUM)

[2] Informatics, Technical University of Munich (TUM)

[3] Computer Science, Telecom Paris, Institut Polytechnique de Paris

21th of April 2021



A Novel Information Theoretic Metric for Labeled Trees
Master Thesis Presentation
Denys Lazarenko [1]
Thesis supervisor: Prof. Dr. Tobias Lasser [2]
Principal Advisor: Prof. Thomas Bonald [3]
[1] Department of Mathematics, Technical University of Munich (TUM)
[2] Informatics, Technical University of Munich (TUM)
[3] Computer Science, Telecom Paris, Institut Polytechnique de Paris
21th of April 2021

1. Good morning to everyone, my name is Denys Lazarenko and I am a student of Mathematics in Data Science Master Degree Program. Today I would like to present you my research, titled as **A novel Information Theoretic Metric for Labeled Trees**. It was conducted in cooperation with Data Intelligence and Graphs[DIG] team, a group of researchers at Télécom Paris, under the supervision of Professor Thomas Bonald. This work is based on two papers which are going to be published soon(for now they are in arxiv).

Outline

1 Problem Statement
2 Related Work
3 Methodology
4 Experiments
5 Results
6 Conclusions and Future Work

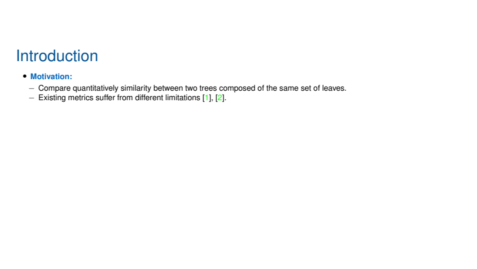
Outline

1 Problem Statement
2 Related Work
3 Methodology
4 Experiments
5 Results
6 Conclusions and Future Work

1. The outline is the following: we will talk about motivation and application of the current work, have a brief overview of existing metrics. Then I present a theoretic background of a novel metric and outcomes from experiments. At the end we are going to analyse the results and to talk about future work and possible improvements.

Introduction

- **Motivation:**
  - Compare quantitatively similarity between two trees composed of the same set of leaves.
  - Existing metrics suffer from different limitations [1], [2].



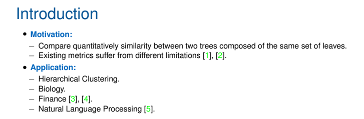
Introduction

- Motivation:
  - Compare quantitatively similarity between two trees composed of the same set of leaves.
  - Existing metrics suffer from different limitations [1], [2].

1. Trees are one of the most popular data structures in Computer Science because it is easy to implement and understand them. We know well how to analyse, traverse and use them as a base element of more complicated algorithms. Machine Learning is not an exception, 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 between two or more trees that have the same set of leaves but different topologies. To solve this task, numerous metrics were proposed, however, all these metrics either have high complexity or perform well only in a specific domain.

Introduction

- **Motivation:**
  - Compare quantitatively similarity between two trees composed of the same set of leaves.
  - Existing metrics suffer from different limitations [1], [2].
- **Application:**
  - Hierarchical Clustering.
  - Biology.
  - Finance [3], [4].
  - Natural Language Processing [5].



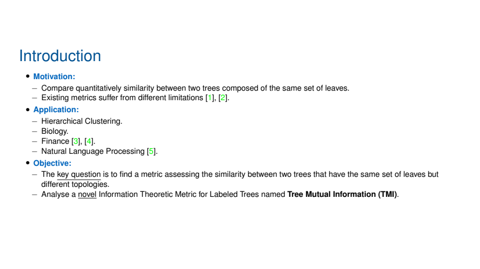
Introduction

- **Motivation:**
  - Compare quantitatively similarity between two trees composed of the same set of leaves.
  - Existing metrics suffer from different limitations [1], [2].
- **Application:**
  - Hierarchical Clustering.
  - Biology.
  - Finance [3], [4].
  - Natural Language Processing [5].

1. Indeed, the application area is vast. One of the most popular task of unsupervised learning is clustering. The subfamily of clustering algorithms, called hierarchical clustering, is often used in various tasks due to their explainability and adaptivity.
   There is a considerable demand for a good metric that would be able to evaluate performance of hierarchical clustering algorithms and adjust their hyperparameters.
   To tackle this issue, some new metrics were recently introduced: Dasgupta cost and Tree Sampling Divergence distance.
   Besides, other areas of application are Biology, Finance and Natural Language Processing.

Introduction

- **Motivation:**
  - Compare quantitatively similarity between two trees composed of the same set of leaves.
  - Existing metrics suffer from different limitations [1], [2].
- **Application:**
  - Hierarchical Clustering.
  - Biology.
  - Finance [3], [4].
  - Natural Language Processing [5].
- **Objective:**
  - The key question is to find a metric assessing the similarity between two trees that have the same set of leaves but different topologies.
  - Analyse a novel Information Theoretic Metric for Labeled Trees named **Tree Mutual Information (TMI)**.



Introduction

- Motivation:
  - Compare quantitatively similarity between two trees composed of the same set of leaves.
  - Existing metrics suffer from different limitations [1], [2].
- Application:
  - Hierarchical Clustering.
  - Biology.
  - Finance [3], [4].
  - Natural Language Processing [5].
- Objective:
  - The key question is to find a metric assessing the similarity between two trees that have the same set of leaves but different topologies.
  - Analyse a novel Information Theoretic Metric for Labeled Trees named Tree Mutual Information (TMI).

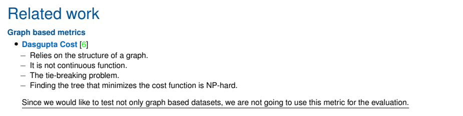
1. **Our Objective:** is to find a metric assessing the quality of trees. We therefore propose a novel Information Theoretic Metric for Trees named Tree Mutual Information (TMI).

Related work

Graph based metrics

- **Dasgupta Cost** [6]
  - Relies on the structure of a graph.
  - It is not continuous function.
  - The tie-breaking problem.
  - Finding the tree that minimizes the cost function is NP-hard.

Since we would like to test not only graph based datasets, we are not going to use this metric for the evaluation.



Related work

Graph based metrics

- Dasgupta Cost [6]
  - Relies on the structure of a graph.
  - It is not continuous function.
  - The tie-breaking problem.
  - Finding the tree that minimizes the cost function is NP-hard.

Since we would like to test not only graph based datasets, we are not going to use this metric for the evaluation.

1. Lets have a look on state of the art metrics, some of them are strictly applicable to graphs, while others are rather for a general usage. However, all these metrics have some limitations, lets analyse some of them:
   Dasgupta cost: Relies on the structure of a graph. It is not a continuous function, since slight changes modify the score significantly.
   Since we would like to test not only graph based datasets, we are not going to use this metric for the evalua
2. The following metrics are tree-based **CLICK!!!** :
   Robinson–Foulds is a symmetric distance between two trees, which measures the number of branch-splits present in one tree, but not in another, and scores 1 for each division that is not matched.
   RF has its well-known shortcomings: by moving a single node in a tree can result in a considerable jump of RF score, but in reality, these trees are almost identical
   Another metric to be considered is **ordered tree edit distance**, which basically optimizes number of operations which are necessary to transform one tree into another.
   There are 3 allowed operations: rename, delete or insert a node
   TED metric has a high time and space complexities. The algorithm's efficiency highly depends on the tree shape.

Related work

Graph based metrics

- **Dasgupta Cost** [6]
  - Relies on the structure of a graph.
  - It is not continuous function.
  - The tie-breaking problem.
  - Finding the tree that minimizes the cost function is NP-hard.

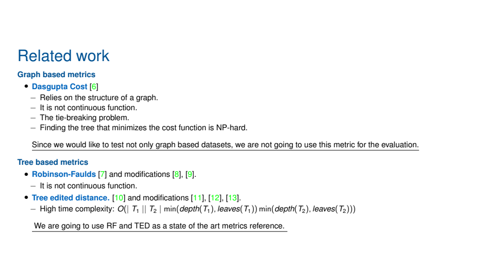
Since we would like to test not only graph based datasets, we are not going to use this metric for the evaluation.

Tree based metrics

- **Robinson-Faulds** [7] and modifications [8], [9].
  - It is not continuous function.
- **Tree edited distance.** [10] and modifications [11], [12], [13].
  - High time complexity: $O(|T_1||T_2| \min(depth(T_1), leaves(T_1)) \min(depth(T_2), leaves(T_2)))$

We are going to use RF and TED as a state of the art metrics reference.

└Related work



Related work

Graph based metrics

- Dasgupta Cost [6]
  - Relies on the structure of a graph.
  - It is not continuous function.
  - The tie-breaking problem.
  - Finding the tree that minimizes the cost function is NP-hard.

Since we would like to test not only graph based datasets, we are not going to use this metric for the evaluation.

Tree based metrics

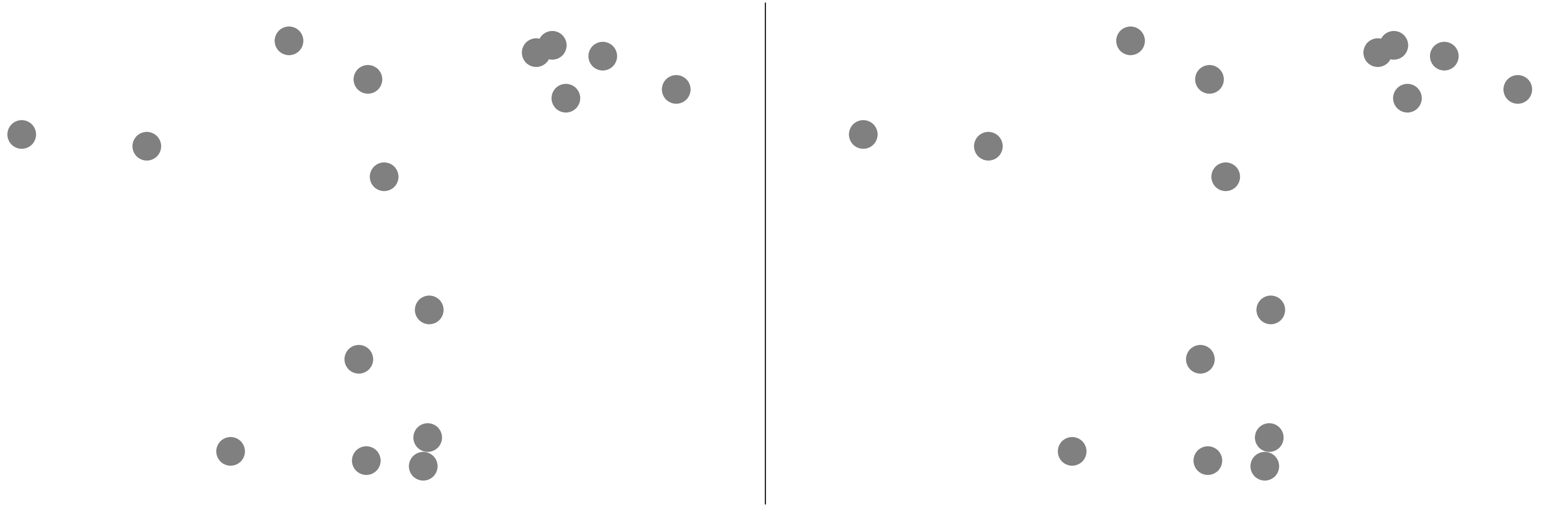
- Robinson-Faulds [7] and modifications [8], [9].
  - It is not continuous function.
- Tree edited distance. [10] and modifications [11], [12], [13].
  - High time complexity: $O(|T_1||T_2| \min(depth(T_1), leaves(T_1)) \min(depth(T_2), leaves(T_2)))$

We are going to use RF and TED as a state of the art metrics reference.

1. Lets have a look on state of the art metrics, some of them are strictly applicable to graphs, while others are rather for a general usage. However, all these metrics have some limitations, lets analyse some of them:
   Dasgupta cost: Relies on the structure of a graph. It is not a continuous function, since slight changes modify the score significantly.
   Since we would like to test not only graph based datasets, we are not going to use this metric for the evalua
2. The following metrics are tree-based **CLICK!!!** :
   Robinson–Foulds is a symmetric distance between two trees, which measures the number of branch-splits present in one tree, but not in another, and scores 1 for each division that is not matched.
   RF has its well-known shortcomings: by moving a single node in a tree can result in a considerable jump of RF score, but in reality, these trees are almost identical
   Another metric to be considered is **ordered tree edit distance**, which basically optimizes number of operations which are necessary to transform one tree into another.
   There are 3 allowed operations: rename, delete or insert a node
   TED metric has a high time and space complexities. The algorithm's efficiency highly depends on the tree shape.

Methodology

Clustering



What is the **best clustering**?

Methodology

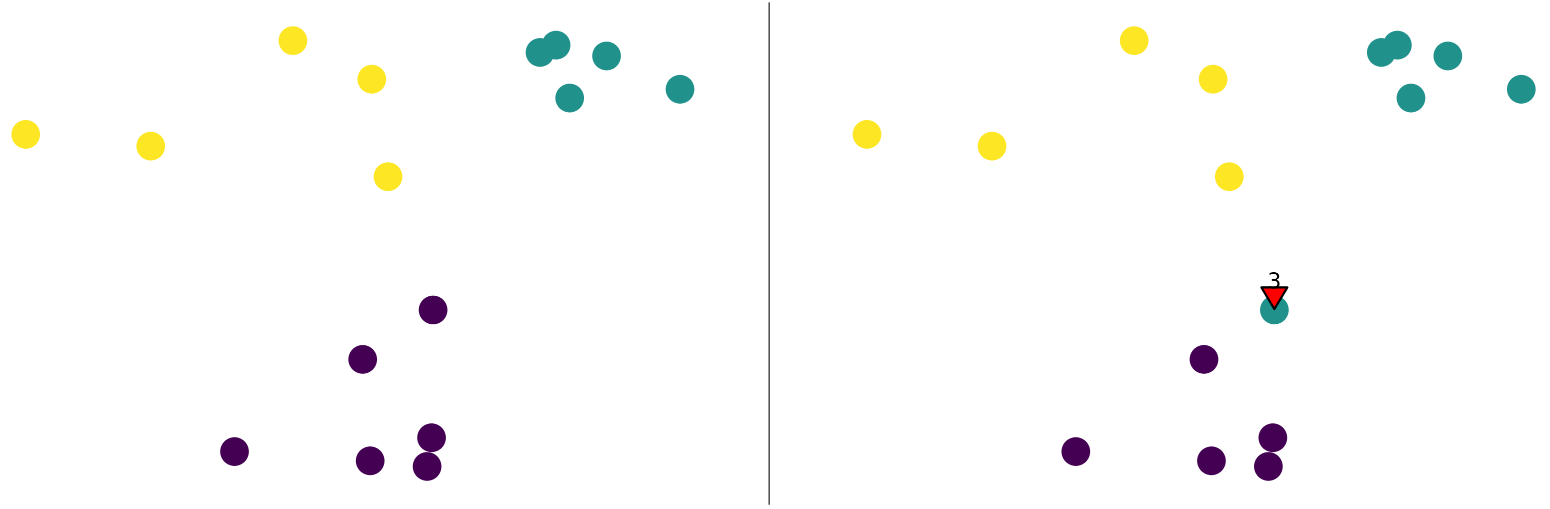
Clustering

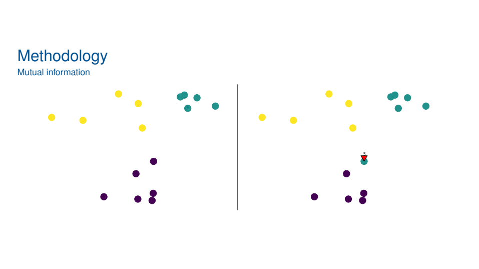
What is the **best clustering**?

1. In our new method we are trying to tackle these limitations. Let's get some feeling of methodology which lays underneath of it. Let's start from a very simple example: assume that we have a set of data points and we would like to partition it. Obviously, this figure represent a trivial example, where we have only one cluster with all points in it and it does not contain any information. The natural question which arises "What is the **best clustering**?"

Methodology

Mutual information

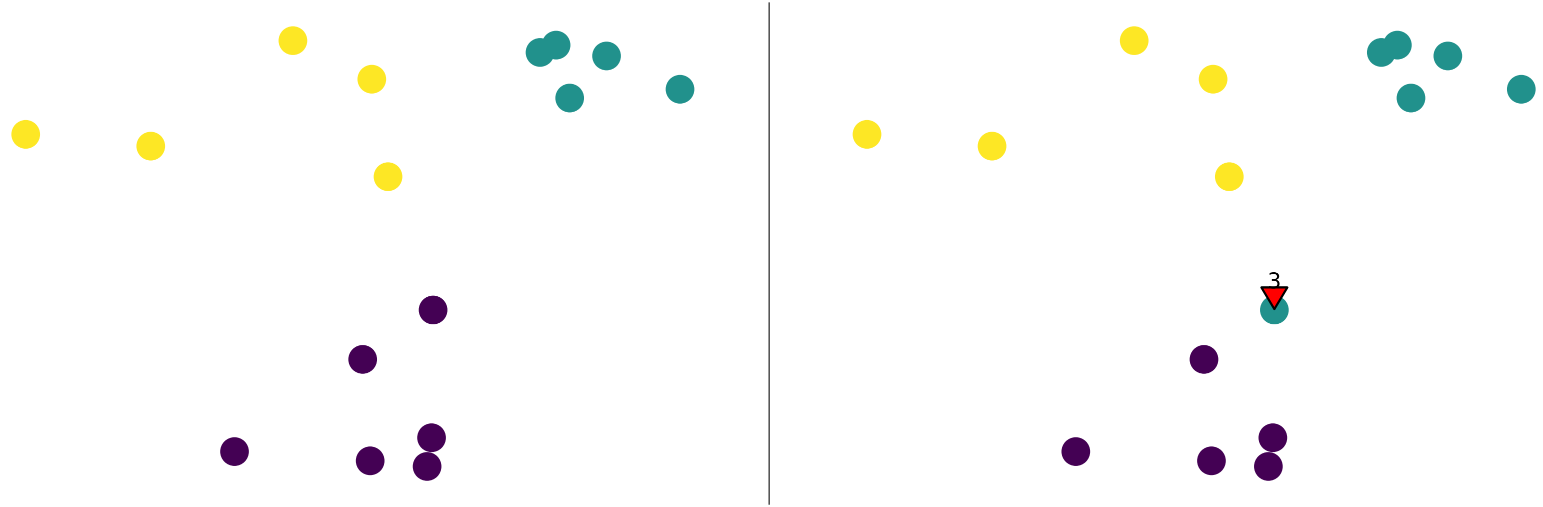




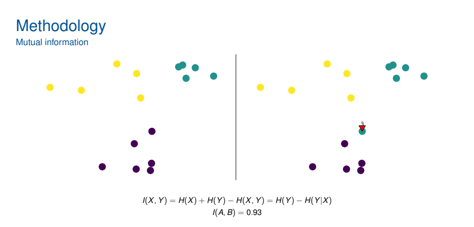
1. To better explain, assume that we partitioned data based on some criteria and we received 3 clusters from both sides which are marked yellow, green and purple respectively. The only difference between them is one data point which marked red. In left figure it belongs to purple cluster while in the right one to the green one.
2. Let's calculate a Mutual Information between these two clusterings. **[CLICK]** If two random variables *A* and *B* are strongly dependant then there is a high degree of Mutual Information.
3. In this case we have a value of 0.93, but since we are using a not normalised version of metric it can be greater than one.

Methodology

Mutual information



$$I(X,Y) = H(X) + H(Y) - H(X,Y) = H(Y) - H(Y|X)$$
$$I(A,B) = 0.93$$


Methodology

Mutual information

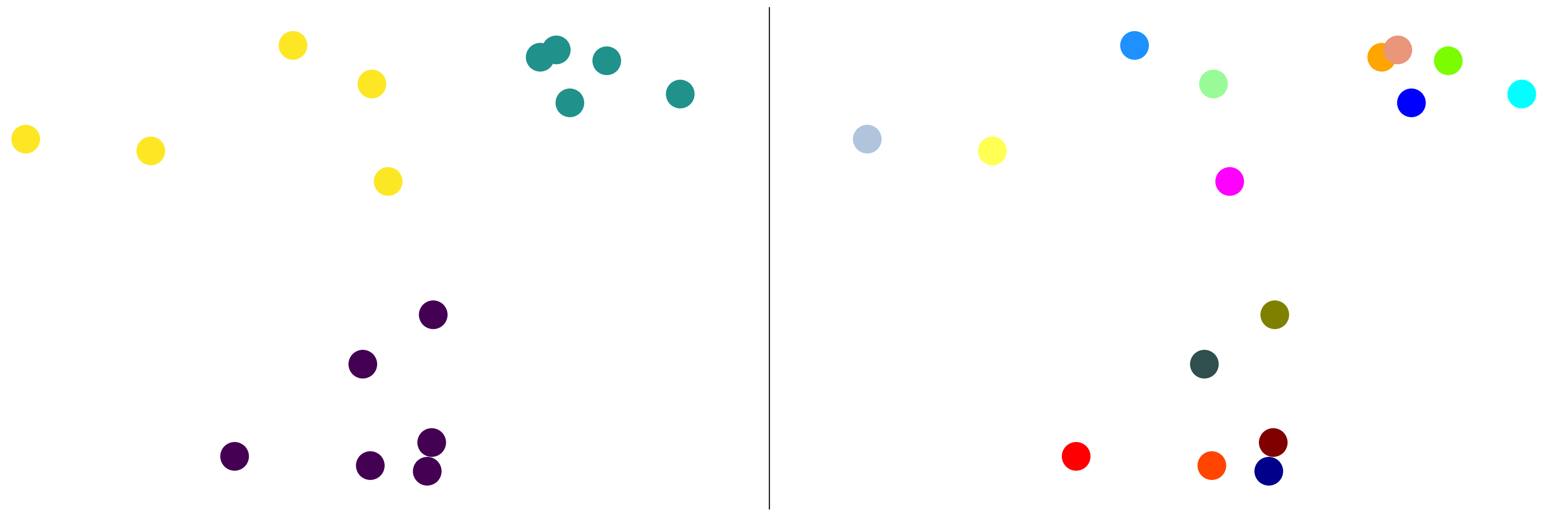
$I(X,Y) = H(X) + H(Y) - H(X,Y) = H(Y) - H(Y|X)$

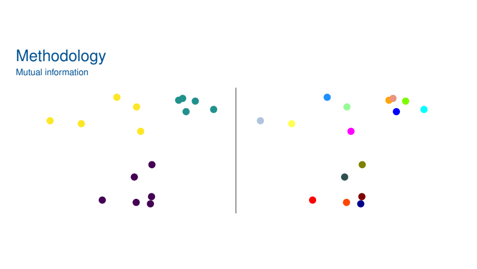
$I(A,B) = 0.93$

1. To better explain, assume that we partitioned data based on some criteria and we received 3 clusters from both sides which are marked yellow, green and purple respectively. The only difference between them is one data point which marked red. In left figure it belongs to purple cluster while in the right one to the green one.
2. Let's calculate a Mutual Information between these two clusterings. **[CLICK]** If two random variables *A* and *B* are strongly dependant then there is a high degree of Mutual Information.
3. In this case we have a value of 0.93, but since we are using a not normalised version of metric it can be greater than one.

Methodology

Mutual information



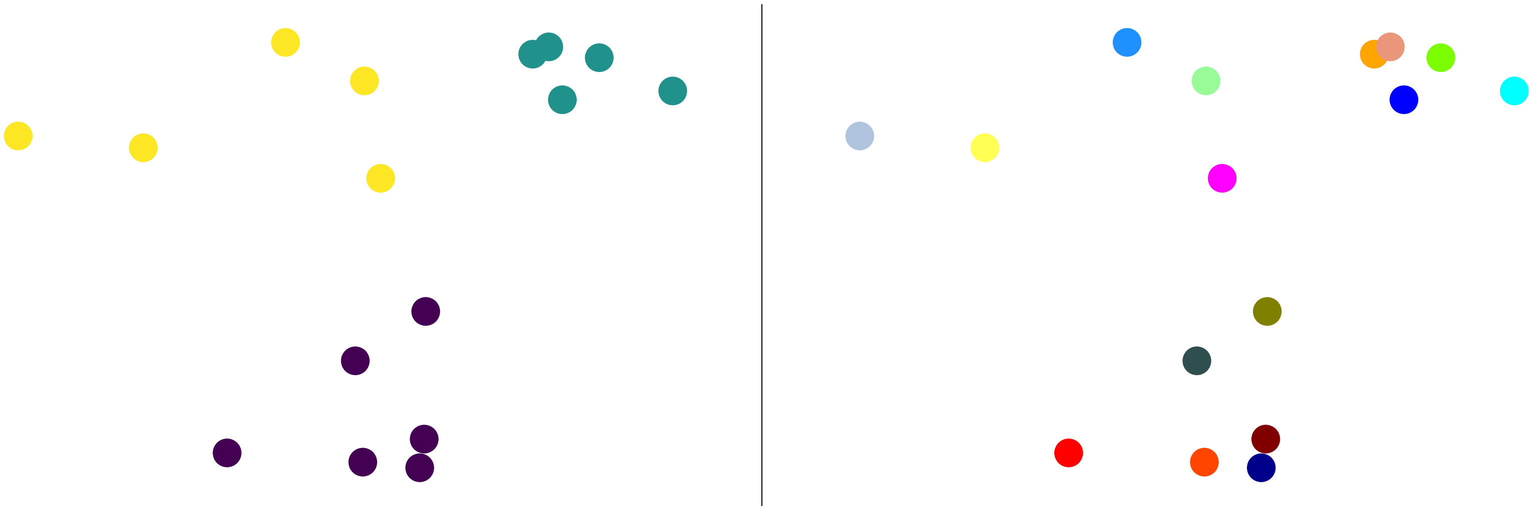


Methodology
Mutual information

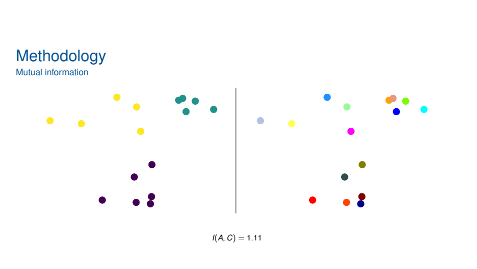
1. More in details: consider the following example: from left side we have the same partition as before, but from right side we have an edge case, when each data point represents a separate cluster. If we measure the mutual information between these two partitions **[CLICK]** then the value will be greater, meaning that we obtain more information, which is obviously does not represent the reality.
2. This metric is not adjusted for chance and will tend to increase as the number of clusters increases, regardless of the actual amount of information between the two distributions. That is why adjustment for chance is necessary.

Methodology

Mutual information



$I(A, C) = 1.11$

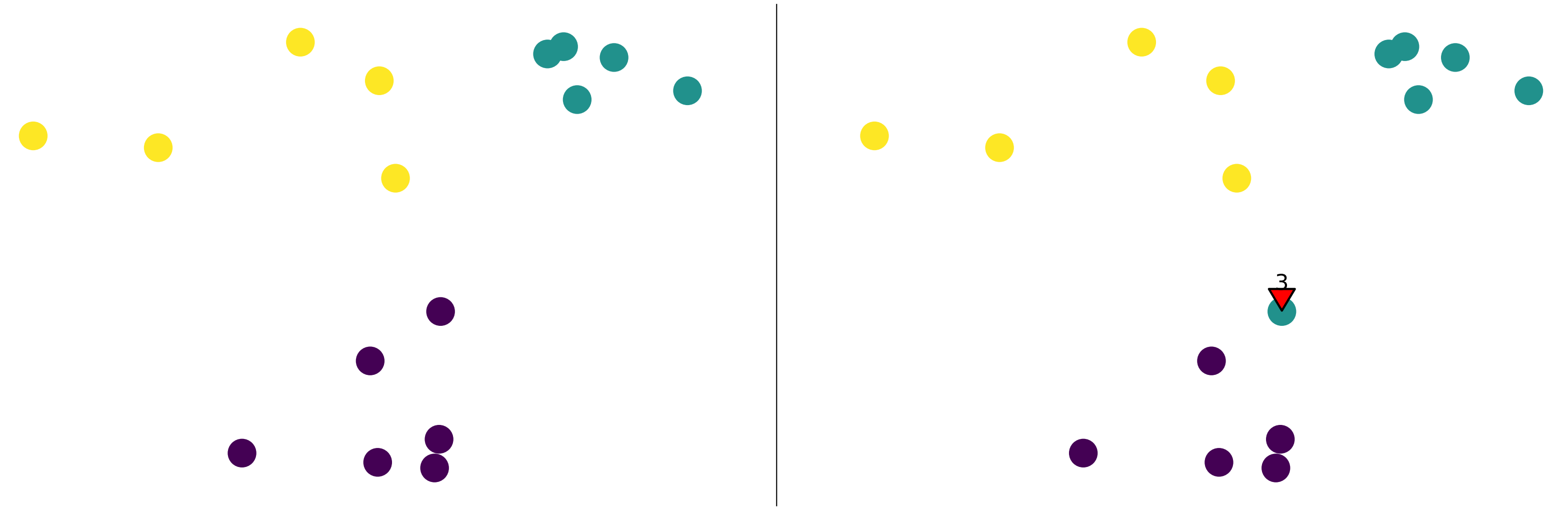


Methodology
Mutual information
$I(A, C) = 1.11$

1. More in details: consider the following example: from left side we have the same partition as before, but from right side we have an edge case, when each data point represents a separate cluster. If we measure the mutual information between these two partitions **[CLICK]** then the value will be greater, meaning that we obtain more information, which is obviously does not represent the reality.
2. This metric is not adjusted for chance and will tend to increase as the number of clusters increases, regardless of the actual amount of information between the two distributions. That is why adjustment for chance is necessary.

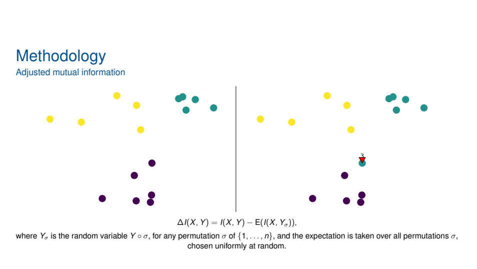
Methodology

Adjusted mutual information



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

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


Methodology

Adjusted mutual information

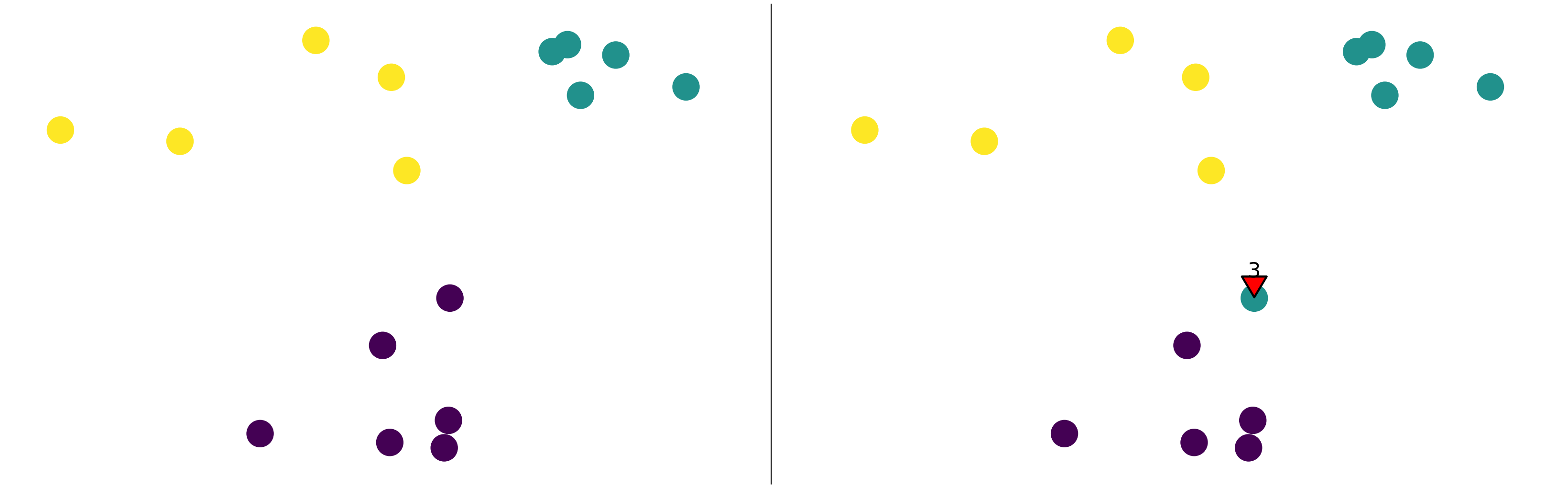
$\Delta I(X, Y) = I(X, Y) - \mathrm{E}(I(X, Y_\sigma)),$

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

1. **SLOW** The adjusted mutual information between X and Y is defined by following equation: Adjusted version equals to the Mutual Information subtracting the Expected value between X and Y sigma where Y_σ is the random variable Y permuted with σ (sigma), which is any permutation of n labels, chosen uniformly at random.

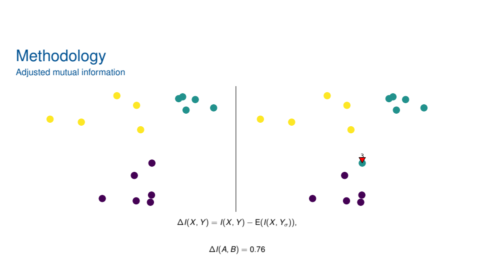
Methodology

Adjusted mutual information



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

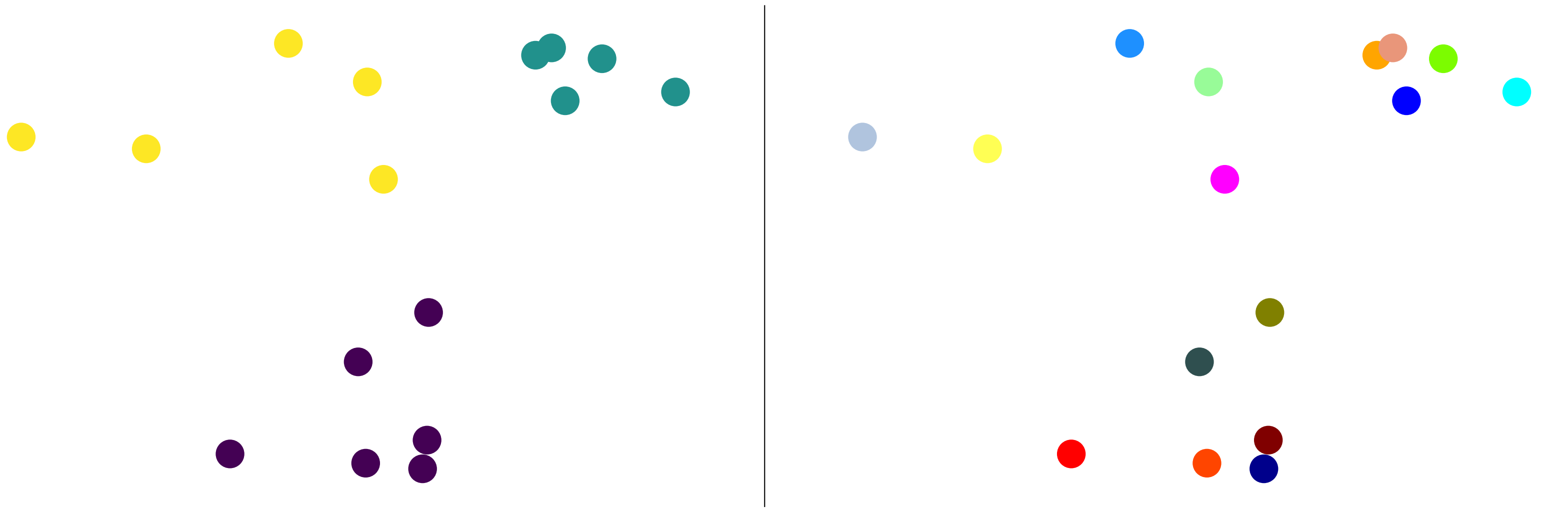
$$\Delta I(A, B) = 0.76$$



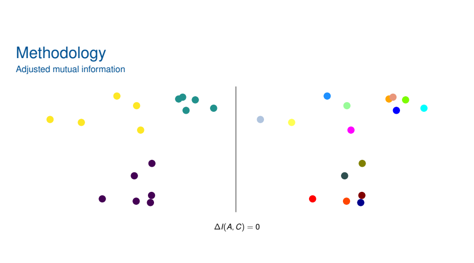
1. Following to this, we conduct the same experiment, but adjusting the metric against chance. We see that we already receive score which is lower than Mutual Information due to subtraction of Expected value.

Methodology

Adjusted mutual information



$\Delta I(A, C) = 0$



Methodology

Adjusted mutual information

$\Delta I(A, C) = 0$

1. And with the trivial clustering the score is 0, meaning that it is completely random and it does not contain any information.

Methodology

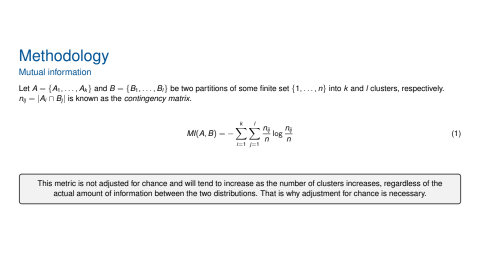
Mutual information

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. $n_{ij} = |A_i \cap B_j|$ is known as the *contingency matrix*.

$$MI(A, B) = -\sum_{i=1}^{k} \sum_{j=1}^{l} \frac{n_{ij}}{n} \log \frac{n_{ij}}{n} \tag{1}$$

This metric is not adjusted for chance and will tend to increase as the number of clusters increases, regardless of the actual amount of information between the two distributions. That is why adjustment for chance is necessary.

└Methodology



Methodology

Mutual information

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. $n_{ij} = |A_i \cap B_j|$ is known as the contingency matrix.

$$MI(A, B) = -\sum_{i=1}^{k} \sum_{j=1}^{l} \frac{n_{ij}}{n} \log \frac{n_{ij}}{n} \tag{1}$$

This metric is not adjusted for chance and will tend to increase as the number of clusters increases, regardless of the actual amount of information between the two distributions. That is why adjustment for chance is necessary.

1. More in depth: here we can see a straightforward formula how to calculate MI between two random variables. A and B are two partitions of some finite set into k and l clusters and $n_{ij} = |A_i \cap B_j|$ is the number of samples both in cluster A_i and cluster B_j, also known as *contingency matrix*.

Methodology

Adjusted Mutual Information

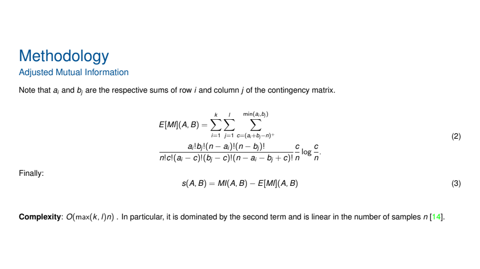
Note that a_i and b_j are the respective sums of row i and column j of the contingency matrix.

$$E[MI](A,B) = \sum_{i=1}^{k}\sum_{j=1}^{l}\sum_{c=(a_i+b_j-n)^+}^{\min(a_i,b_j)} \frac{a_i!b_j!(n-a_i)!(n-b_j)!}{n!c!(a_i-c)!(b_j-c)!(n-a_i-b_j+c)!}\frac{c}{n}\log\frac{c}{n}. \quad (2)$$

Finally:

$$s(A,B) = MI(A,B) - E[MI](A,B) \quad (3)$$

Complexity: $O(\max(k,l)n)$. In particular, it is dominated by the second term and is linear in the number of samples n [14].



Methodology

Adjusted Mutual Information

Note that a_i and b_j are the respective sums of row i and column j of the contingency matrix.

$$E[MI](A,B) = \sum_{i=1}^{k}\sum_{j=1}^{l}\sum_{c=(a_i+b_j-n)^+}^{\min(a_i,b_j)} \frac{a_i!b_j!(n-a_i)!(n-b_j)!}{n!c!(a_i-c)!(b_j-c)!(n-a_i-b_j+c)!}\frac{c}{n}\log\frac{c}{n}. \quad (2)$$

Finally:

$$s(A,B) = MI(A,B) - E[MI](A,B) \quad (3)$$

Complexity: $O(\max(k,l)n)$. In particular, it is dominated by the second term and is linear in the number of samples n [14].

1. In this slide you can see the formula of expected value of mutual information, which punishes randomness by subtracting expected value.
2. It is also frequent to normalize adjusted mutual information to get a score between 0 and 1. In this work, we only focus on the adjustment step.
3. its complexity is about $O(\max(k,l)n)$, which is quite slow for the large graphs.

Methodology

Contribution 1: Pairwise adjustment [15]

We consider permutations σ for which there exists $i, j \in \{1, \dots, n\}$ such that $\sigma(i) = j$ and $\sigma(j) = i$. We think about the set of such permutations σ where the samples i, j are drawn uniformly at random in the set $\{1, \dots, n\}$. We denote by σ_p such a random permutation.

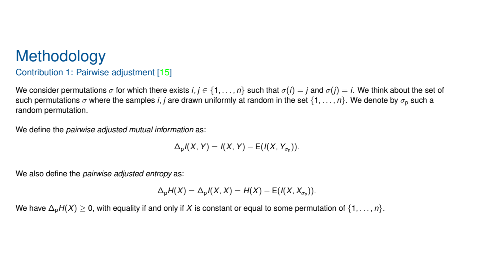
We define the *pairwise adjusted mutual information* as:

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

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 $\{1, \dots, n\}$.



Methodology

Contribution 1: Pairwise adjustment [15]

We consider permutations σ for which there exists $i, j \in \{1, \dots, n\}$ such that $\sigma(i) = j$ and $\sigma(j) = i$. We think about the set of such permutations σ where the samples i, j are drawn uniformly at random in the set $\{1, \dots, n\}$. We denote by σ_p such a random permutation.

We define the *pairwise adjusted mutual information* as:

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

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 $\{1, \dots, n\}$.

1. Having said that, we come up with an improved version of Adjusted Mutual Information. The main assumption: we consider permutations σ (sigma) for which there exists i, j from one to n such that $\sigma(i) = j$ and $\sigma(j) = i$. The samples i, j are drawn uniformly at random in the set one to n.
2. We define the *pairwise adjusted mutual information* and *pairwise adjusted entropy*. More detailed information can be find in our original paper.

Methodology

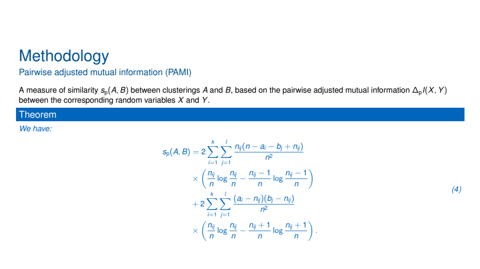
Pairwise adjusted mutual information (PAMI)

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.

Theorem

We have:

$$\begin{aligned} s_{\mathrm{p}}(A, B) &= 2 \sum_{i=1}^{k} \sum_{j=1}^{l} \frac{n_{ij}(n - a_i - b_j + n_{ij})}{n^2} \\ &\times \left(\frac{n_{ij}}{n} \log \frac{n_{ij}}{n} - \frac{n_{ij} - 1}{n} \log \frac{n_{ij} - 1}{n} \right) \\ &+ 2 \sum_{i=1}^{k} \sum_{j=1}^{l} \frac{(a_i - n_{ij})(b_j - n_{ij})}{n^2} \\ &\times \left(\frac{n_{ij}}{n} \log \frac{n_{ij}}{n} - \frac{n_{ij} + 1}{n} \log \frac{n_{ij} + 1}{n} \right). \end{aligned} \tag{4}$$



Methodology

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.

Theorem

We have:

$$\begin{aligned} s_{\mathrm{p}}(A, B) &= 2 \sum_{i=1}^{k} \sum_{j=1}^{l} \frac{n_{ij}(n - a_i - b_j + n_{ij})}{n^2} \\ &\times \left(\frac{n_{ij}}{n} \log \frac{n_{ij}}{n} - \frac{n_{ij} - 1}{n} \log \frac{n_{ij} - 1}{n} \right) \\ &+ 2 \sum_{i=1}^{k} \sum_{j=1}^{l} \frac{(a_i - n_{ij})(b_j - n_{ij})}{n^2} \\ &\times \left(\frac{n_{ij}}{n} \log \frac{n_{ij}}{n} - \frac{n_{ij} + 1}{n} \log \frac{n_{ij} + 1}{n} \right). \end{aligned} \tag{4}$$

1. We have an closed form expression for pairwise similarity.
   The time complexity of this formula is in $O(kl)$. It is independent of the number of samples n in contrast to the standard Adjusted Mutual information, where we summed over one additional index.
   As we can see, this metric is much faster then the standard Adjusted Mutual Information.

Methodology

Pairwise adjusted mutual information (PAMI)

For real data, we consider the 79 datasets of the benchmark suite [**clustering_benchmarks**] [1]. We apply to each dataset 10 clustering algorithms.

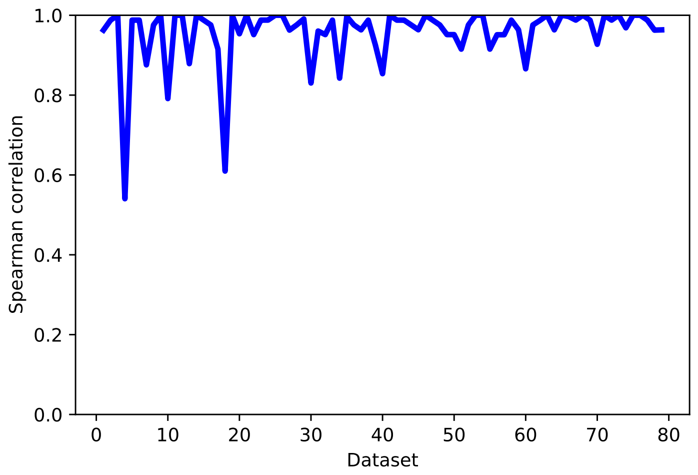
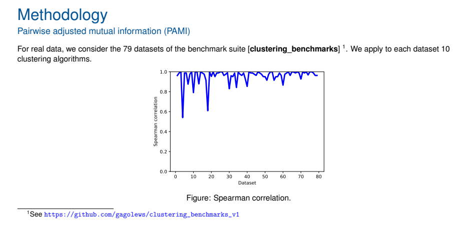


Figure: Spearman correlation.

[1] See https://github.com/gagolews/clustering_benchmarks_v1


Methodology
Pairwise adjusted mutual information (PAMI)
For real data, we consider the 79 datasets of the benchmark suite [clustering_benchmarks] [1]. We apply to each dataset 10 clustering algorithms.
Figure: Spearman correlation.
[1] See https://github.com/gagolews/clustering_benchmarks_v1

1. Next we demonstrate experiments which proves that the novel metric is actually behaves the same way as a standard one.
   For real data, we consider the 79 datasets. We apply to each dataset 10 common clustering algorithms from sckit-learn and measure similarity. We then compute the Spearman correlation of the corresponding similarities
2. We observe that the correlation is very high, suggesting again that both notions of adjusted mutual information tend to provide the same results. For 65 datasets among 79, the Spearman correlation is higher than 95%.

Methodology

Pairwise adjusted mutual information (PAMI)

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.

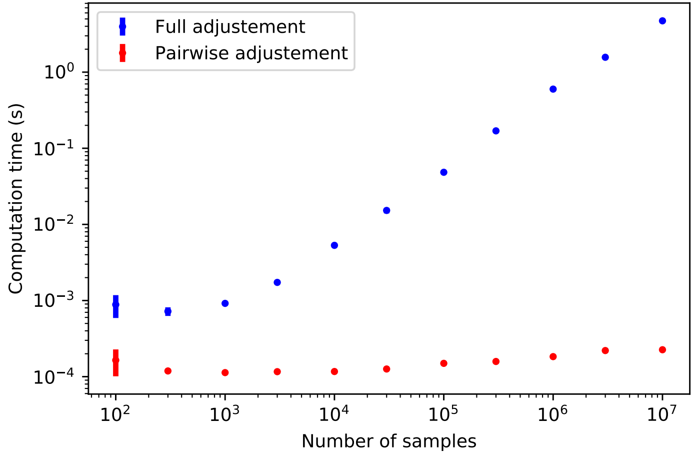
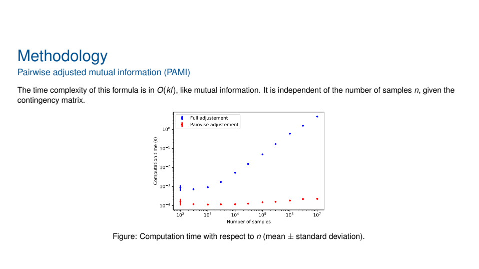


Figure: Computation time with respect to n (mean ± standard deviation).

└Methodology


Methodology

Pairwise adjusted mutual information (PAMI)

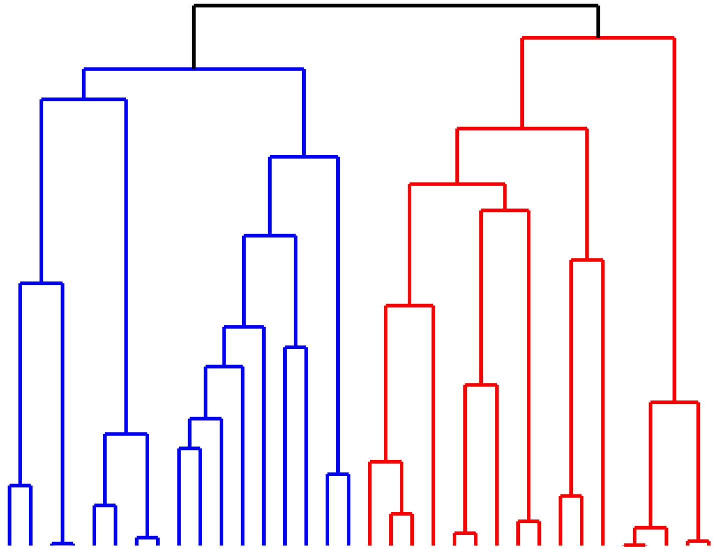
The time complexity of this formula is in $O(kl)$, like mutual information. It is independent of the number of samples n, given the contingency matrix.

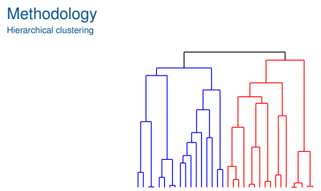
Figure: Computation time with respect to n (mean ± standard deviation).

1. Related to this, we compute run time of metric when the number of samples n grows from 10^2 (ten power of 2) to 10^7 (ten power of 7). The performance gain brought by pairwise adjustement is significant. In particular, the computation time becomes independent of the number of samples.

Methodology

Hierarchical clustering





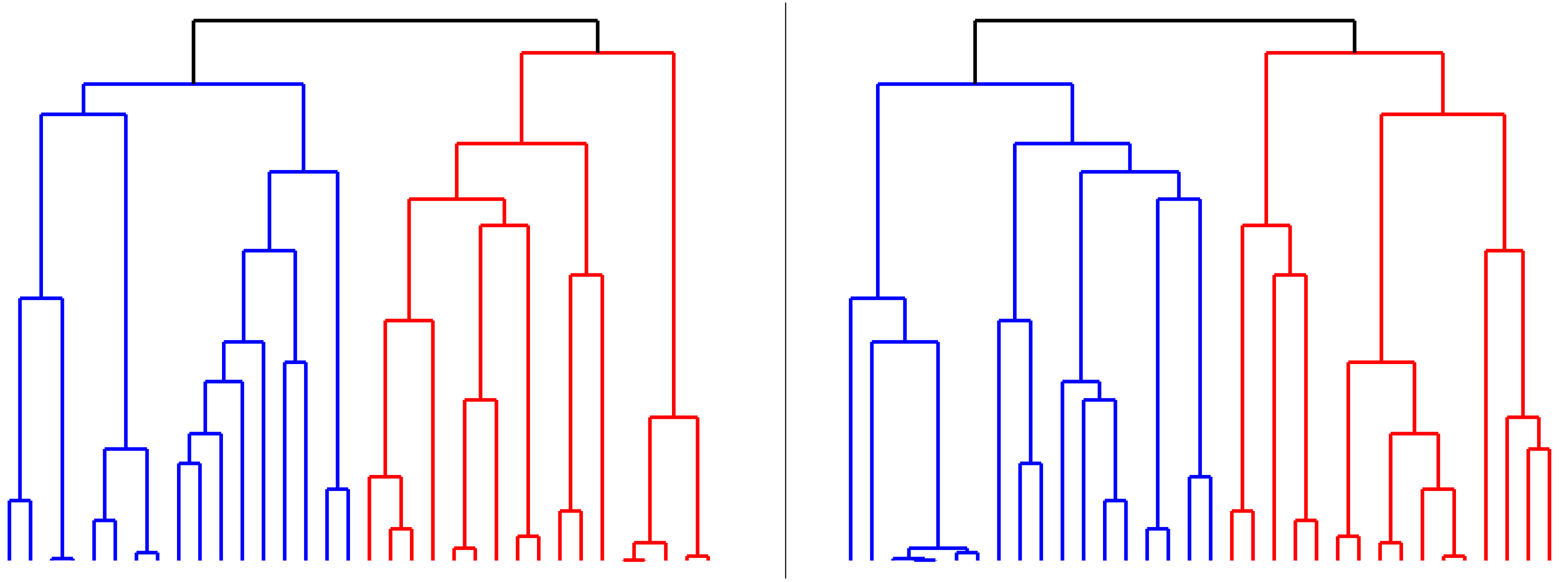
Methodology

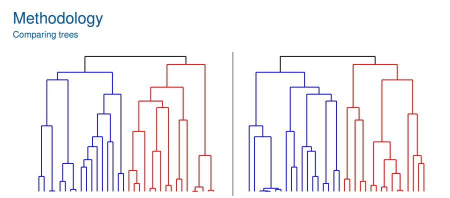
Hierarchical clustering

1. In this Figure we can see a typical output of hierarchical clustering algorithm - dendrogram, which is a compact representation of the hierarchical structure. The dendrogram *D* contains the pair of nodes merged through the run of the algorithm. Additionally, each branch is plotted at height *d*, thus all distances must be non-decreasing.

Methodology

Comparing trees

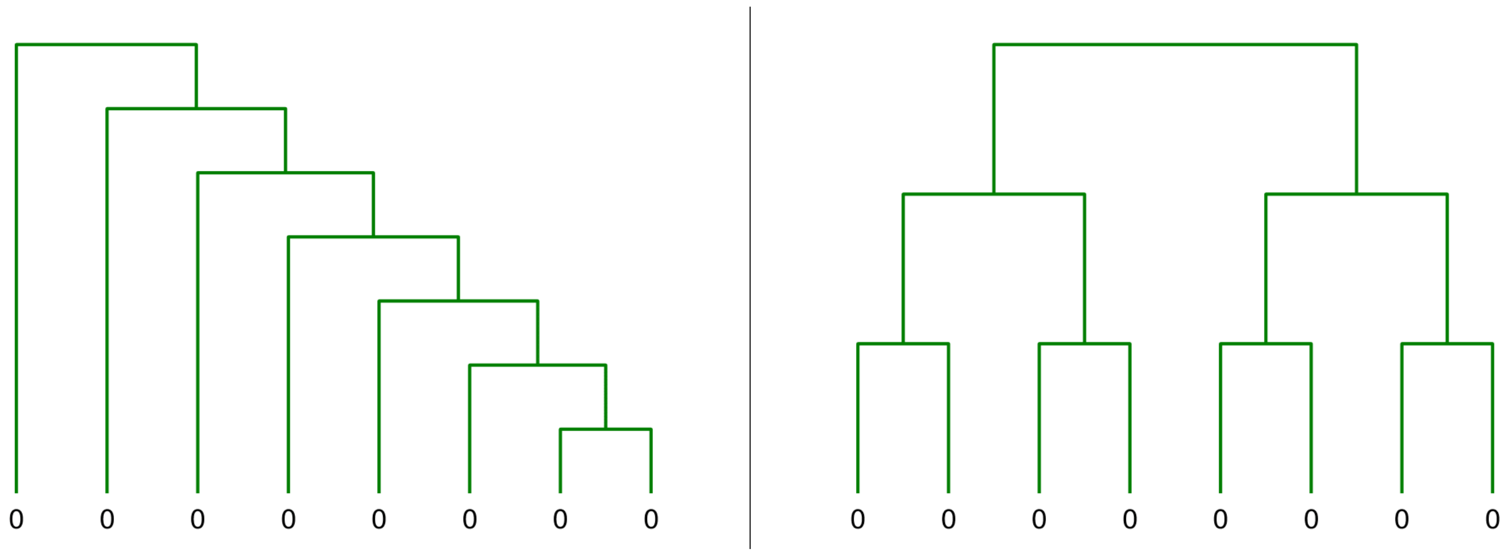


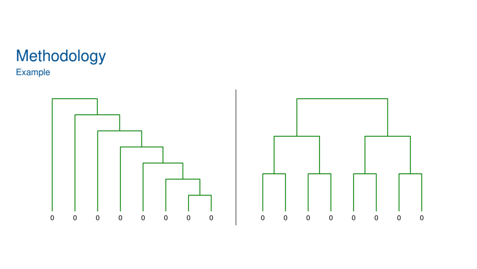


1. Frequently, we have two dendrograms as input and would like to understand how similar are they to each other.

Methodology

Example

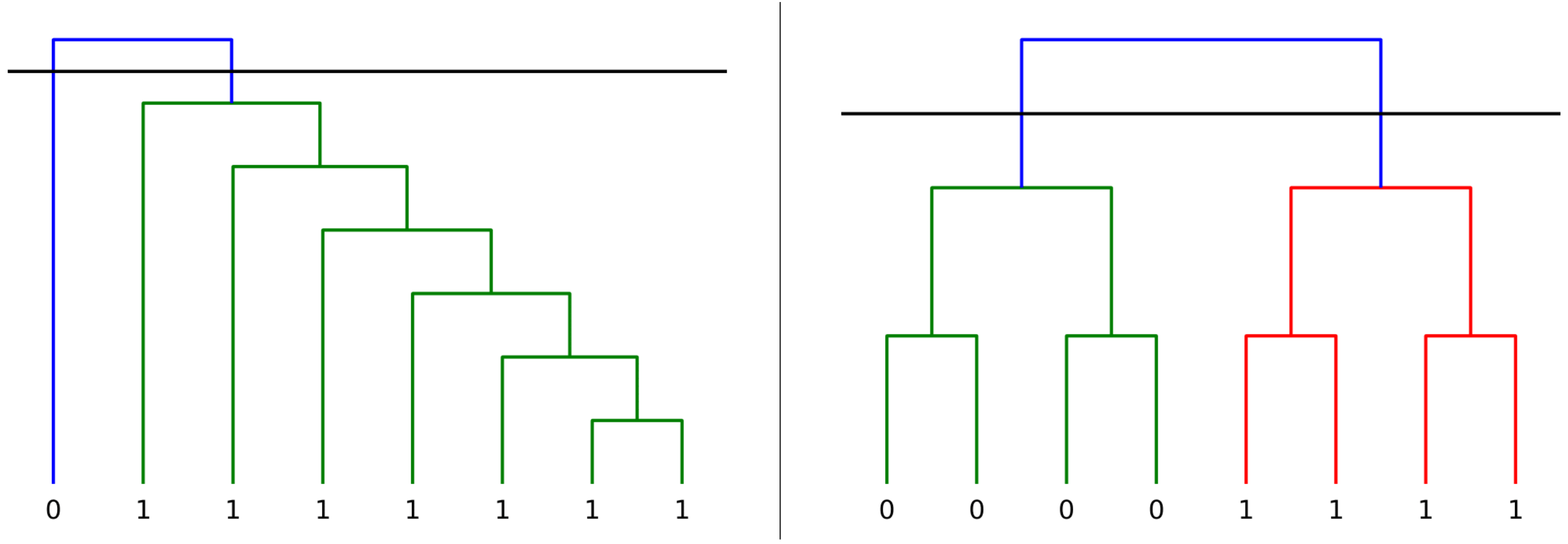




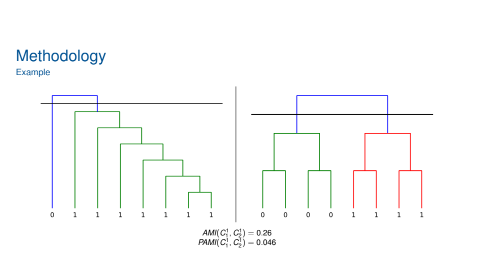
1. To better explain the second contribution to the current work, I propose to have a look on a simple example.
   We compare a caterpillar tree (left figure) with a fully binary tree (right figure). The leaves here represent partitioning. We start from the trivial clustering where all leaf nodes belong to the same cluster 0.
   Intuitively, the best clustering is achieved with 5 clusters in both trees.
   Let’s see whether our algorithm behaves the same way, as it is expected?

Methodology

Example



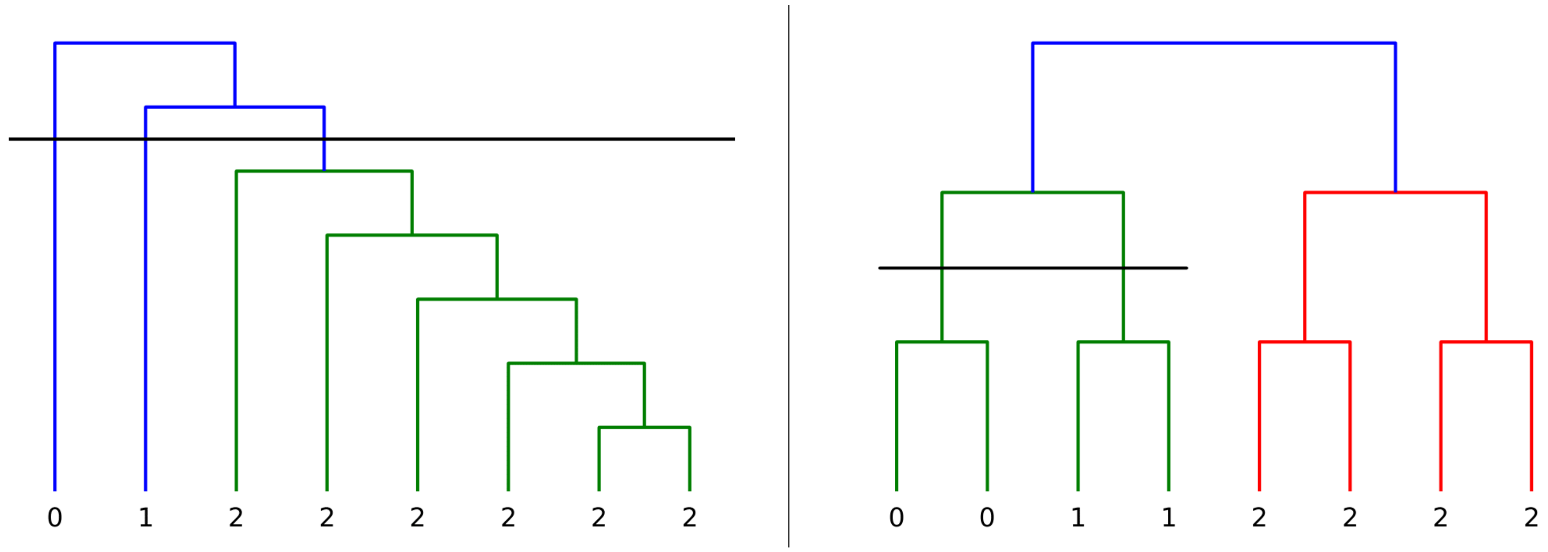
$$AMI(C_1^1, C_2^1) = 0.26$$
$$PAMI(C_1^1, C_2^1) = 0.046$$



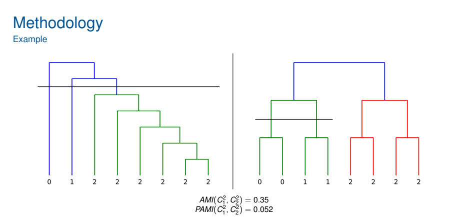
1. As a first step our algorithm make the only possible cut in both trees which results into score growth. AMI has score equal to 0.26 and PAMI is around zero point zero five. It is necessary to remember that both metrics are not normalized!

Methodology

Example



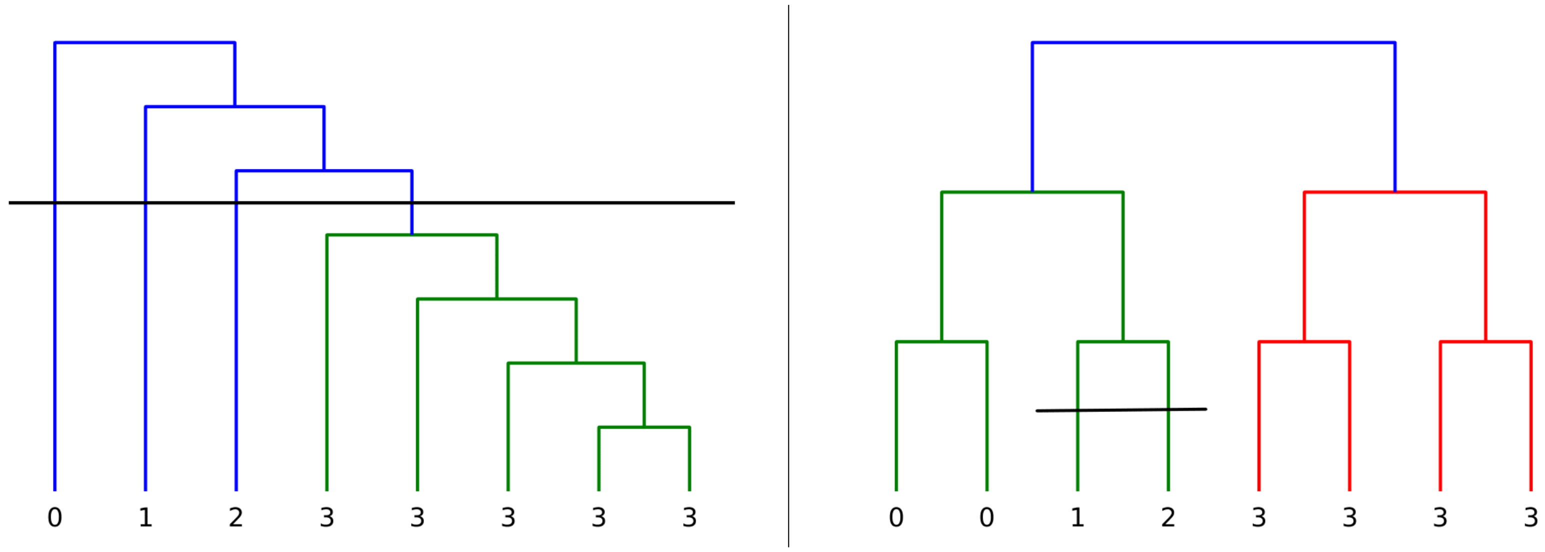
$$AMI(C_1^2, C_2^2) = 0.35$$
$$PAMI(C_1^2, C_2^2) = 0.052$$



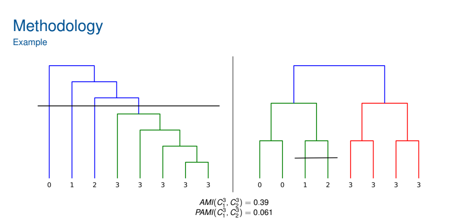
1. Following this, the next best cut for tree pairs is obtained by going into left subtree of binary tree and making an additional cut in caterpillar tree. It results into grows of total score for AMI and PAMI.

Methodology

Example



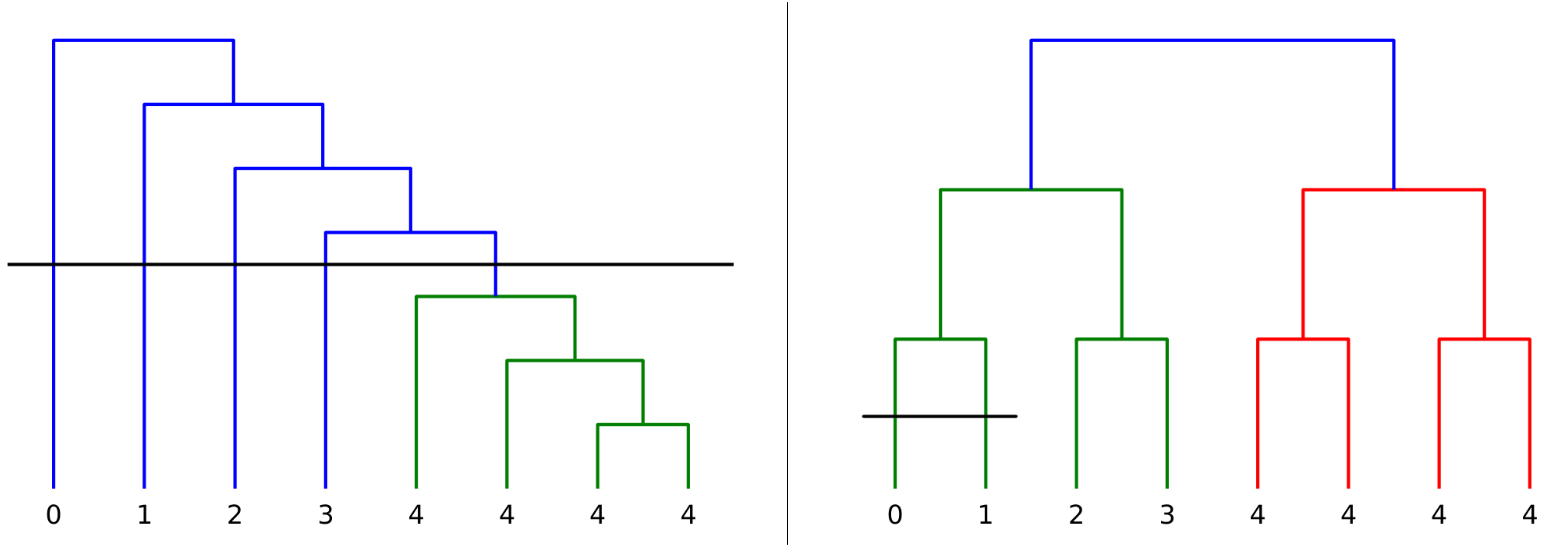
$$AMI(C_1^3, C_2^3) = 0.39$$
$$PAMI(C_1^3, C_2^3) = 0.061$$



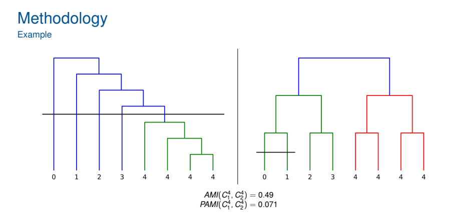
1. Continuing the process, we see that further cut in binary tree of the cluster with label "2" into 2 subclusters and an additional cut in caterpillar tree increase the score.

Methodology

Example



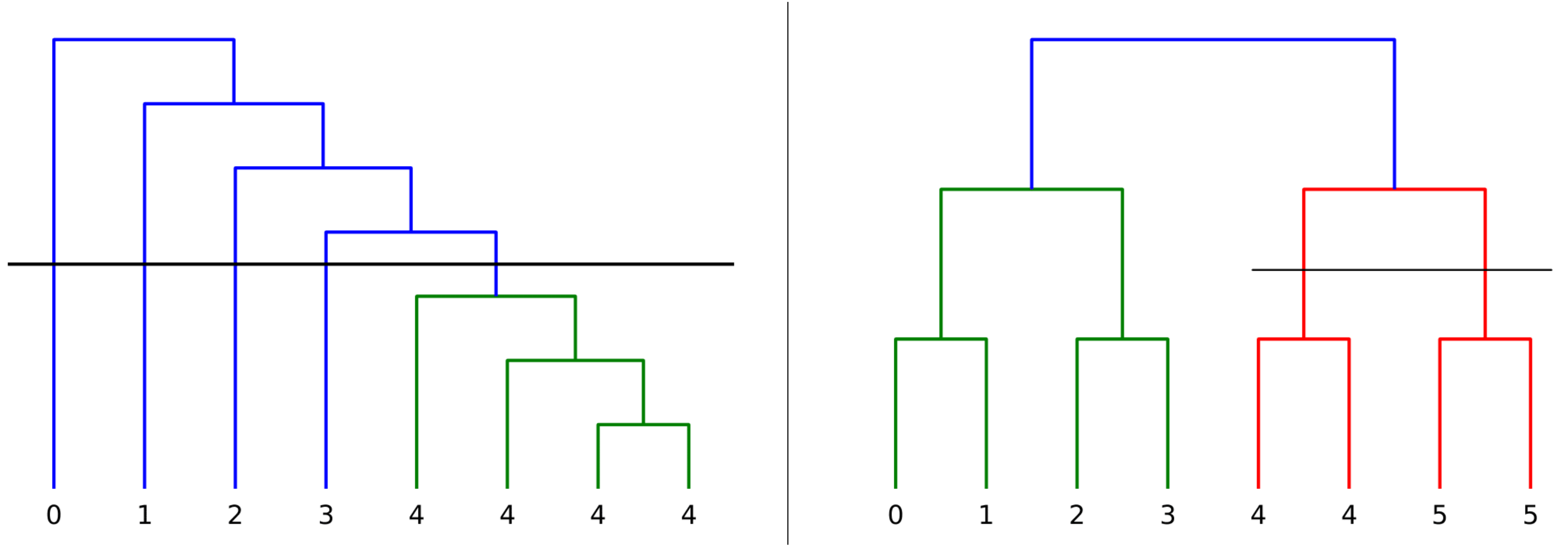
$$AMI(C_1^4, C_2^4) = 0.49$$
$$PAMI(C_1^4, C_2^4) = 0.071$$



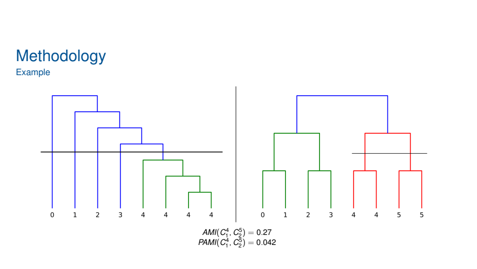
1. Finally, we reached the stage from our assumption - 5 clusters in both trees. It is indeed maximum value for both metrics AMI and PAMI.

Methodology

Example



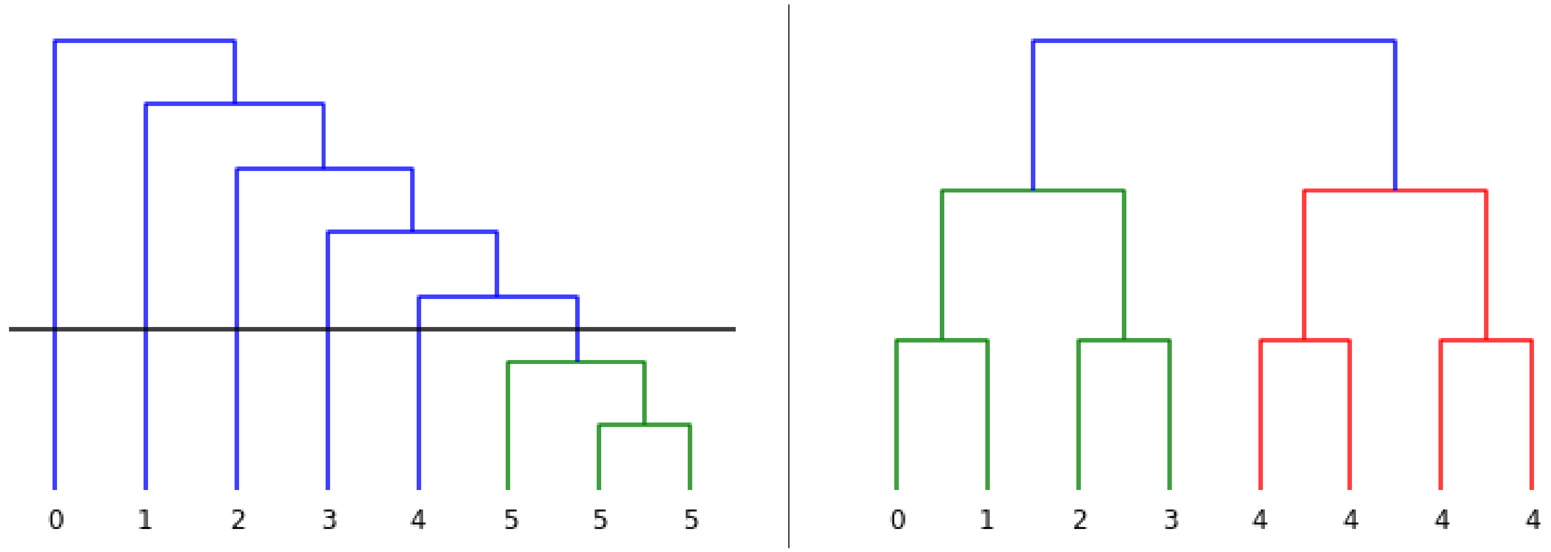
$$AMI(C_1^4, C_2^5) = 0.27$$
$$PAMI(C_1^4, C_2^5) = 0.042$$



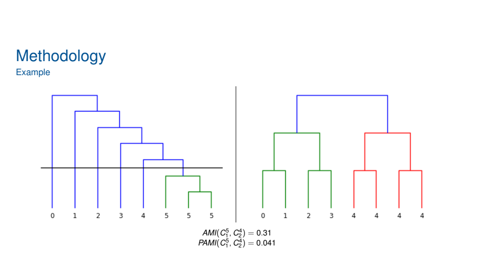
1. If we continue to do further cuts in any tree, it will lead to the degradation in the score. We can clearly see it in these examples. One additional cut in the binary tree reduces AMI and PAMI scores.

Methodology

Example



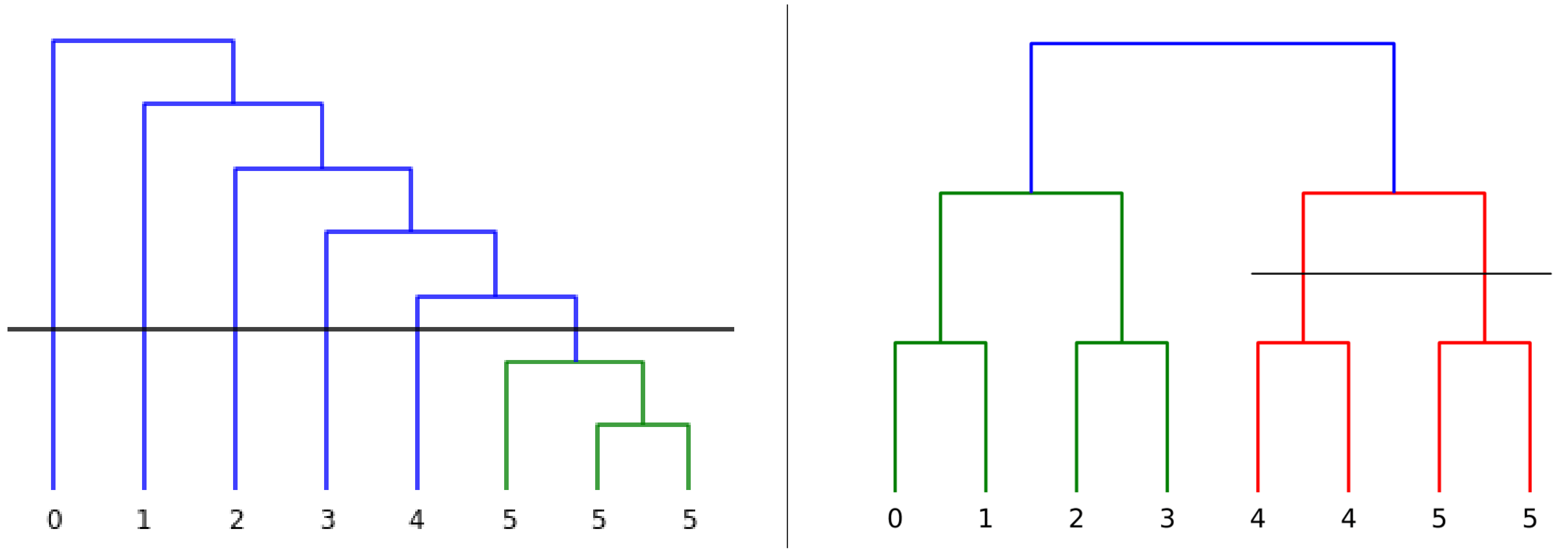
$$AMI(C_1^5, C_2^4) = 0.31$$
$$PAMI(C_1^5, C_2^4) = 0.041$$



1. If we make a cut in the caterpillar tree the value drops as well.

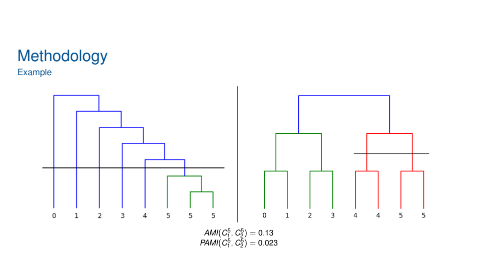
Methodology

Example



$$AMI(C_1^5, C_2^5) = 0.13$$

$$PAMI(C_1^5, C_2^5) = 0.023$$



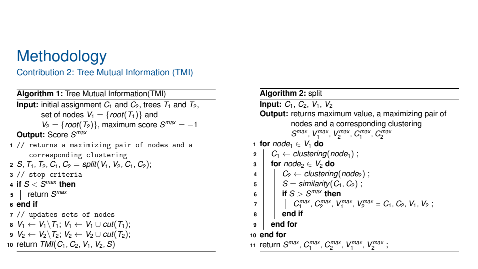
1. And the last possibility, to make a cut in both trees also results into degradation of similarity score. As a result, the maximum score is obtained with the total number of clusters equal to 5, as was expected in the beginning.

Methodology

Contribution 2: Tree Mutual Information (TMI)

```
Algorithm 1: 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
$S, T_1, T_2, C_1, C_2 = split(V_1, V_2, C_1, C_2)$;
// stop criteria
if $S < S^{max}$ then
    return $S^{max}$
end if
// updates sets of nodes
$V_1 \leftarrow V_1 \backslash T_1$; $V_1 \leftarrow V_1 \cup cut(T_1)$;
$V_2 \leftarrow V_2 \backslash T_2$; $V_2 \leftarrow V_2 \cup cut(T_2)$;
return $TMI(C_1, C_2, V_1, V_2, S)$
```

```
Algorithm 2: split
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 $node_1 \in V_1$ do
    $C_1 \leftarrow clustering(node_1)$ ;
    for $node_2 \in V_2$ do
        $C_2 \leftarrow clustering(node_2)$ ;
        $S = similarity(C_1, C_2)$ ;
        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 $S^{max}, C_1^{max}, C_2^{max}, V_1^{max}, V_2^{max}$ ;
```



Methodology
Contribution 2: Tree Mutual Information (TMI)

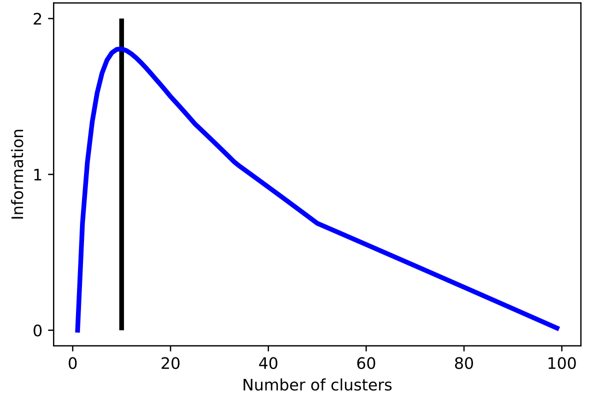
1. After we understand the logic of a new metric on the simple experiment let's move to the actual algorithm. The algorithm takes as input a pair of trees T_1 and T_2 in Newick format [16] with the same number of leaves n and performs the following steps:
   1.1 The TMI algorithm is initialized with 2 sets of nodes which we use to compare on every step: V_1 and V_2 for each of tree. Additionally, it takes as input maximum score S^{max} and clustering for each leaf set.
   1.2 Let's look into *split* algorithm on the right. We consider the clustering C_1 induced by the top level of tree T_1, and clustering C_2 for tree T_2 respectively. Compute similarity S between these clusters, where *similarity* is one of our metrics: AMI or PAMI. If newly computed score S is greater than maximum value S^{max} then we update maximum clusterings with new values.
   1.3 We repeat recursively the step above, meaning whenever score S increases, we change clustering going down in tree T and making cut.
   1.4 If on the step i no cut in trees T_1 and T_2 increases the score S^{max} then we hit the stop criteria and return maximum value S.

Methodology

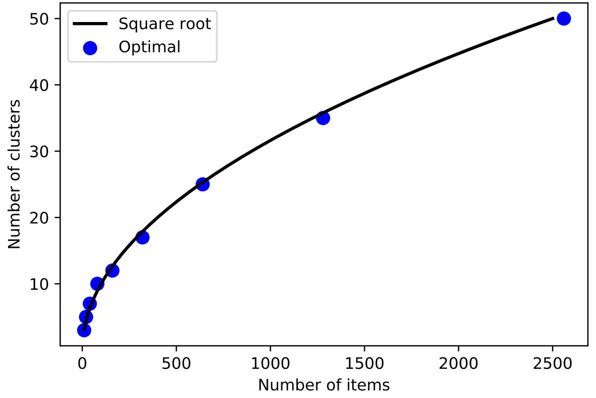
Optimal number of partitions

Conjecture

For a large number of items n, the adjusted entropy is maximized for $\sqrt{n}$ clusters of same size

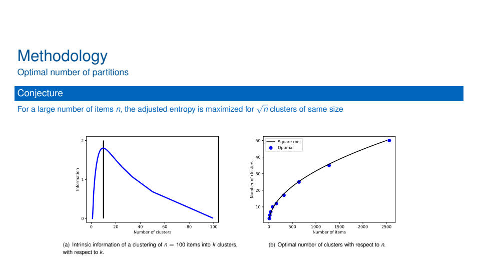


(a) Intrinsic information of a clustering of $n = 100$ items into k clusters, with respect to k.



(b) Optimal number of clusters with respect to n.

└Methodology


Methodology
Optimal number of partitions
Conjecture
For a large number of items n, the adjusted entropy is maximized for $\sqrt{n}$ clusters of same size
(a) Intrinsic information of a clustering of $n = 100$ items into k clusters, with respect to k.
(b) Optimal number of clusters with respect to n.

1. Following Conjecture suggests that for a large number of items n, the adjusted entropy is maximized for $\sqrt{n}$ clusters of same size.
   We justify the best cut property by following experiments.
   We take a randomly generated dataset of different size n. Then we partition it in K clusters in the range $\{1, \dots, n\}$. We measure adjusted entropy for each set of labels. In Figure 1, we can clearly see that the peak of shared information to the number of clusters equals 10. To identify an optimal number of clusters, we take the maximum index of all adjusted entropy scores. We can notice that in Figure 2 the optimal number of clusters and a squared root function have almost identical behaviour.
   We can extend this Conjecture further to the similarity s between clusterings A and B.

Methodology

Complexity

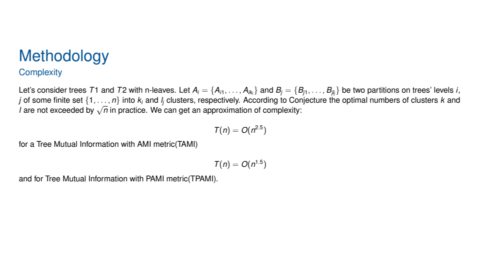
Let's consider trees $T1$ and $T2$ with n-leaves. Let $A_i = \{A_{i1}, \dots, A_{ik_i}\}$ and $B_j = \{B_{j1}, \dots, B_{jl_j}\}$ be two partitions on trees' levels i, j of some finite set $\{1, \dots, n\}$ into k_i and l_j clusters, respectively. According to Conjecture the optimal numbers of clusters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

$$T(n) = O(n^{2.5})$$

for a Tree Mutual Information with AMI metric(TAMI)

$$T(n) = O(n^{1.5})$$

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



Methodology

Complexity

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

$$T(n) = O(n^{2.5})$$

for a Tree Mutual Information with AMI metric(TAMI)

$$T(n) = O(n^{1.5})$$

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

1. To provide a closed form solution of algorithm complexity, we need to solve a discrete optimization problem which 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 and B_j be two partitions on trees' levels i, j into k_i and l_j clusters, respectively.
2. Complexity of *split* operation includes two nested loops V_1 and V_2 which on each step calculate metric AMI or PAMI. According to Conjecture 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:

   $$T(n) = O(n^{2.5})$$

   for a Tree Mutual Information with AMI metric

   $$T(n) = O(n^{1.5})$$

   for PAMI metric.

Experiments

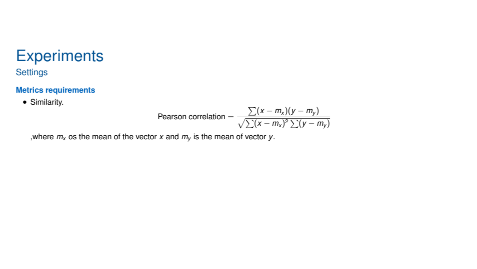
Settings

Metrics requirements

- Similarity.

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

,where m_x os the mean of the vector x and m_y is the mean of vector y.



Experiments
Settings
Metrics requirements
- Similarity.
$$\text{Pearson correlation} = \frac{\sum(x - m_x)(y - m_y)}{\sqrt{\sum(x - m_x)^2 \sum(y - m_y)}}$$
,where m_x os the mean of the vector x and m_y is the mean of vector y.

1. To justify effectiveness of proposed algorithm, we are moving to experiments. We test metrics in various scenarios and datasets: synthetic datasets give us a possibility to analyse the behaviour of metrics on simple examples. In contrast, real datasets help to understand the scaling and generalisation potential. We introduce the following requirements.
2. 1) To capture quantitatively behaviour of metrics in the syntactic experiments, we measure the Pearson correlation between values of corresponding metric and numbers of shuffled pairs. We expect this correlation to be high. **[CLICK]**
   2) Our metric should scale on large datasets. We would like to have a clear understanding on how it grows with the increasing number of samples. Therefore, we will measure time complexity and expect it to be low. **[CLICK]**
   3) Another critical aspect for the metric is to be explainable, therefore we analyse dependency of the optimal number of clusters to the similarity score. **[CLICK]**
   We will use following agglomerative clustering algorithms: Ward, Louvain and Paris. They all works with graph and vector data.

Experiments

Settings

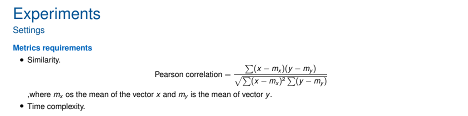
Metrics requirements

- Similarity.

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

,where m_x os the mean of the vector x and m_y is the mean of vector y.

- Time complexity.



Experiments

Settings

Metrics requirements

- Similarity.

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

,where m_x os the mean of the vector x and m_y is the mean of vector y.

- Time complexity.

1. To justify effectiveness of proposed algorithm, we are moving to experiments. We test metrics in various scenarios and datasets: synthetic datasets give us a possibility to analyse the behaviour of metrics on simple examples. In contrast, real datasets help to understand the scaling and generalisation potential. We introduce the following requirements.
2. 1) To capture quantitatively behaviour of metrics in the syntactic experiments, we measure the Pearson correlation between values of corresponding metric and numbers of shuffled pairs. We expect this correlation to be high. **[CLICK]**
   2) Our metric should scale on large datasets. We would like to have a clear understanding on how it grows with the increasing number of samples. Therefore, we will measure time complexity and expect it to be low. **[CLICK]**
   3) Another critical aspect for the metric is to be explainable, therefore we analyse dependency of the optimal number of clusters to the similarity score. **[CLICK]**
   We will use following agglomerative clustering algorithms: Ward, Louvain and Paris. They all works with graph and vector data.

Experiments

Settings

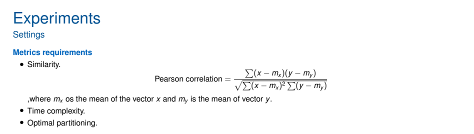
Metrics requirements

- Similarity.

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

,where m_x os the mean of the vector x and m_y is the mean of vector y.

- Time complexity.
- Optimal partitioning.



1. To justify effectiveness of proposed algorithm, we are moving to experiments. We test metrics in various scenarios and datasets: synthetic datasets give us a possibility to analyse the behaviour of metrics on simple examples. In contrast, real datasets help to understand the scaling and generalisation potential. We introduce the following requirements.
2. 1) To capture quantitatively behaviour of metrics in the syntactic experiments, we measure the Pearson correlation between values of corresponding metric and numbers of shuffled pairs. We expect this correlation to be high. **[CLICK]**
   2) Our metric should scale on large datasets. We would like to have a clear understanding on how it grows with the increasing number of samples. Therefore, we will measure time complexity and expect it to be low. **[CLICK]**
   3) Another critical aspect for the metric is to be explainable, therefore we analyse dependency of the optimal number of clusters to the similarity score. **[CLICK]**
   We will use following agglomerative clustering algorithms: Ward, Louvain and Paris. They all works with graph and vector data.

Experiments

Settings

Metrics requirements

- Similarity.

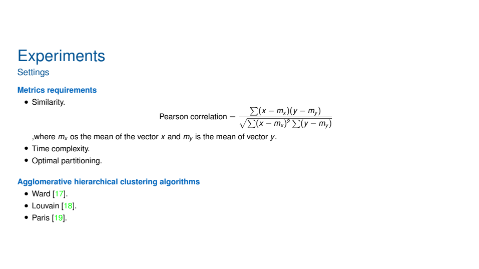
$$\text{Pearson correlation} = \frac{\sum(x-m_x)(y-m_y)}{\sqrt{\sum(x-m_x)^2\sum(y-m_y)}}$$

,where m_x os the mean of the vector x and m_y is the mean of vector y.

- Time complexity.
- Optimal partitioning.

Agglomerative hierarchical clustering algorithms

- Ward [17].
- Louvain [18].
- Paris [19].



Experiments
Settings
Metrics requirements
Similarity.
Pearson correlation = $\frac{\sum(x-m_x)(y-m_y)}{\sqrt{\sum(x-m_x)^2\sum(y-m_y)}}$
,where m_x os the mean of the vector x and m_y is the mean of vector y.
Time complexity.
Optimal partitioning.
Agglomerative hierarchical clustering algorithms
Ward [17].
Louvain [18].
Paris [19].

1. To justify effectiveness of proposed algorithm, we are moving to experiments. We test metrics in various scenarios and datasets: synthetic datasets give us a possibility to analyse the behaviour of metrics on simple examples. In contrast, real datasets help to understand the scaling and generalisation potential. We introduce the following requirements.
2. 1) To capture quantitatively behaviour of metrics in the syntactic experiments, we measure the Pearson correlation between values of corresponding metric and numbers of shuffled pairs. We expect this correlation to be high. **[CLICK]**
   2) Our metric should scale on large datasets. We would like to have a clear understanding on how it grows with the increasing number of samples. Therefore, we will measure time complexity and expect it to be low. **[CLICK]**
   3) Another critical aspect for the metric is to be explainable, therefore we analyse dependency of the optimal number of clusters to the similarity score. **[CLICK]**
   We will use following agglomerative clustering algorithms: Ward, Louvain and Paris. They all works with graph and vector data.

Experiments

Syntactic data: Binary Trees

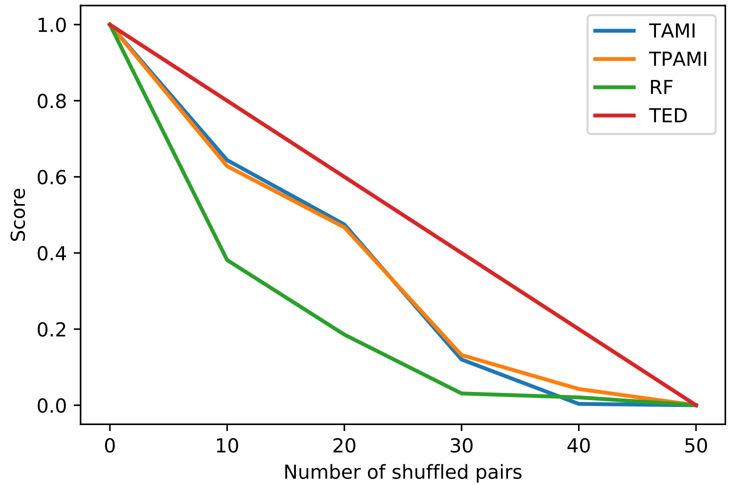


Figure: Binary trees with $n = 100$ leaves: dependence of the similarity score to the parameter k - shuffled leaves pairs.

Table: Pearson correlation between number of leaf shuffled pairs and values of corresponding metric.

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

.

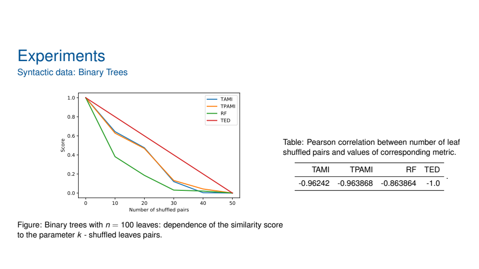

Experiments
Syntactic data: Binary Trees

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

Table: Pearson correlation between number of leaf shuffled pairs and values of corresponding metric.

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

1. Having said this, the first 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.
2. As we can see, all algorithms capture the trend correctly: with an increasing number of shuffled leaf pairs k, the similarity between trees decreases.
   To prove quantitatively the statement above, we measure the Pearson correlation. The correlation Table demonstrates that the TED metric perfectly correlates with noise, while TAMI and TPAMI have scores around -0.96 which is almost perfect. The worst performance has the RF metric.

Experiments

Syntactic data: Binary Trees

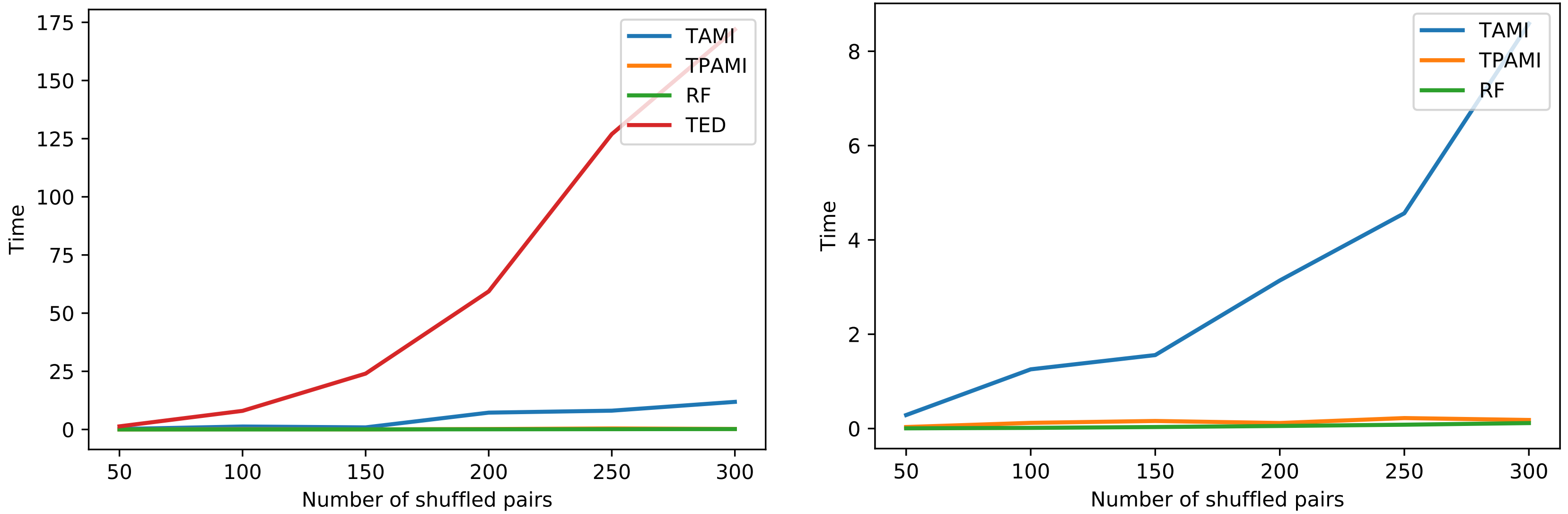
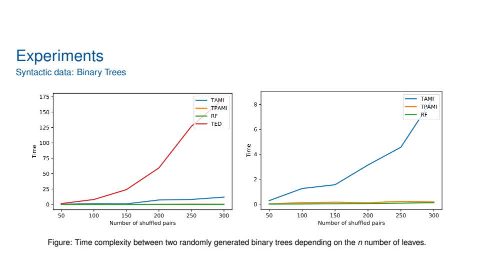


Figure: Time complexity between two randomly generated binary trees depending on the *n* number of leaves.


Experiments
Syntactic data: Binary Trees
Figure: Time complexity between two randomly generated binary trees depending on the n number of leaves.

1. Following to this, we would like to understand how time complexity changes with the leaves' number *n*. We randomly generate a pair of binary trees with *n* leaves and measure the time needed for the metrics to be calculated. The results are following:
   - TED metric has huge time complexity on relatively small trees -> left Figure.
   - While TAMI is significantly faster in comparison to TED, it is one order slower than TPAMI -> right Figure.
   - TPAMI and RF metrics have a similar performance.

Experiments

Syntactic data: Stochastic Block Model (SBM)

- **Input:** unlabeled graph $G = (V, E)$ (directed, undirected, bipartite) represented as adjacensy matrix A.
- **Output:** hierarchy of clusters represented as a dendrogram D.

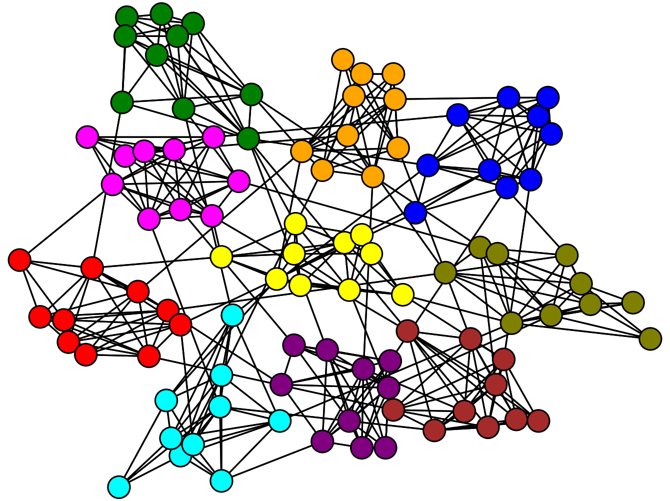
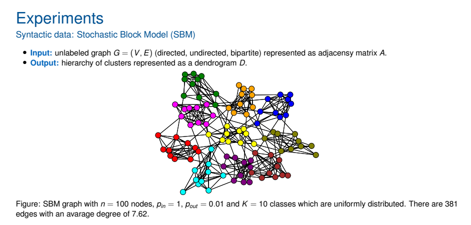


Figure: SBM graph with $n = 100$ nodes, $p_{in} = 1$, $p_{out} = 0.01$ and $K = 10$ classes which are uniformly distributed. There are 381 edges with an avarage degree of 7.62.



Experiments

Syntactic data: Stochastic Block Model (SBM)

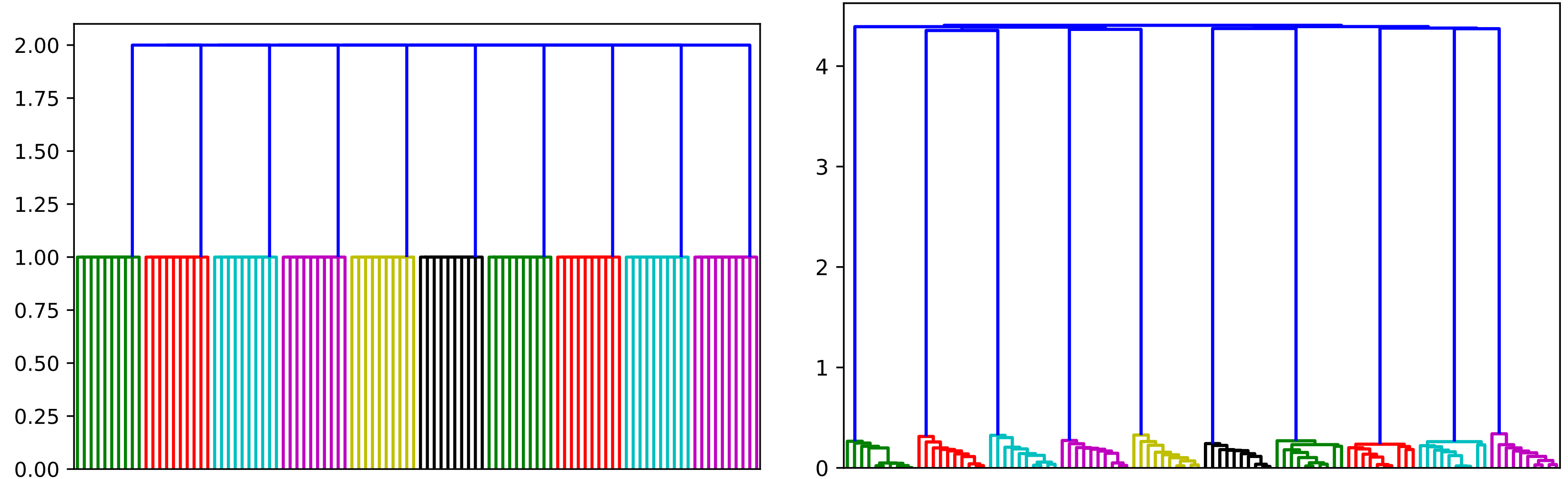
- **Input:** unlabeled graph $G = (V, E)$ (directed, undirected, bipartite) represented as adjacensy matrix A.
- **Output:** hierarchy of clusters represented as a dendrogram D.

Figure: 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.

1. The next experiment is based on Stochastic Block Model. SBM is a generative model that produces graphs with communities. It creates graphs with n nodes that are grouped into k sets.
2. So, we generate the graph G_{origin} with $n = 100$ nodes and $K = 10$ clusters.

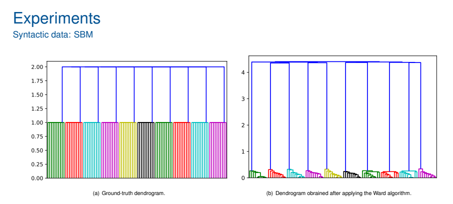
Experiments

Syntactic data: SBM



(a) Ground-truth dendrogram.

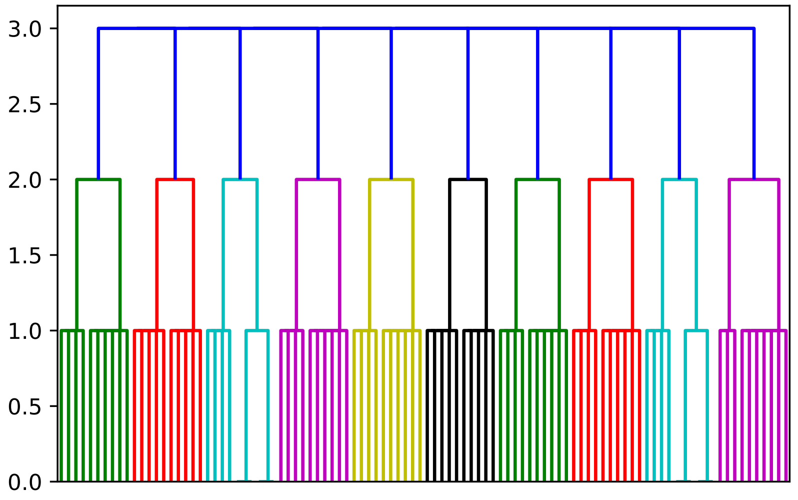
(b) Dendrogram obrained after applying the Ward algorithm.


Experiments
Syntactic data: SBM
(a) Ground-truth dendrogram.
(b) Dendrogram obrained after applying the Ward algorithm.

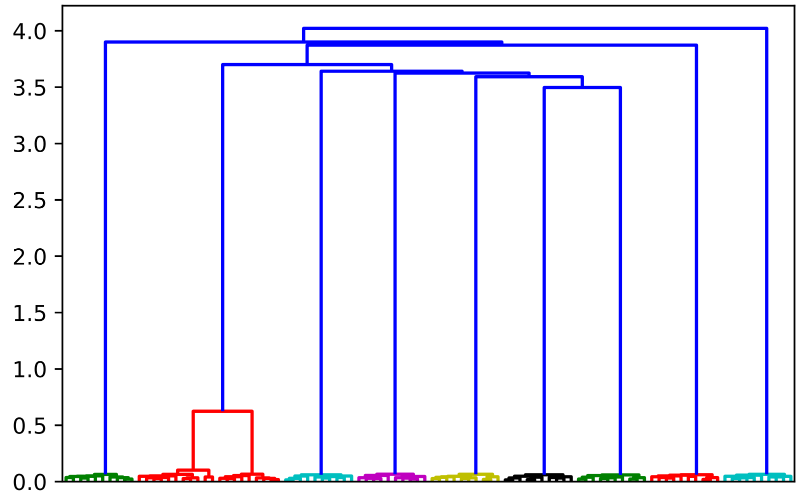
1. Then we construct a ground truth hierarchy of the given graph represented as a dendrogram D_{origin}, which for simplicity has only one level and 10 clusters with 10 nodes in each of them. Other 3 dendrograms are obtained by applying Ward, Louvain and Paris algorithms.

Experiments

Syntactic data: SBM

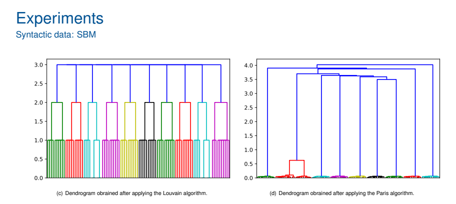


(c) Dendrogram obrained after applying the Louvain algorithm.



(d) Dendrogram obrained after applying the Paris algorithm.

└Experiments


Experiments

Syntactic data: SBM

Experiments

Syntactic data: SBM

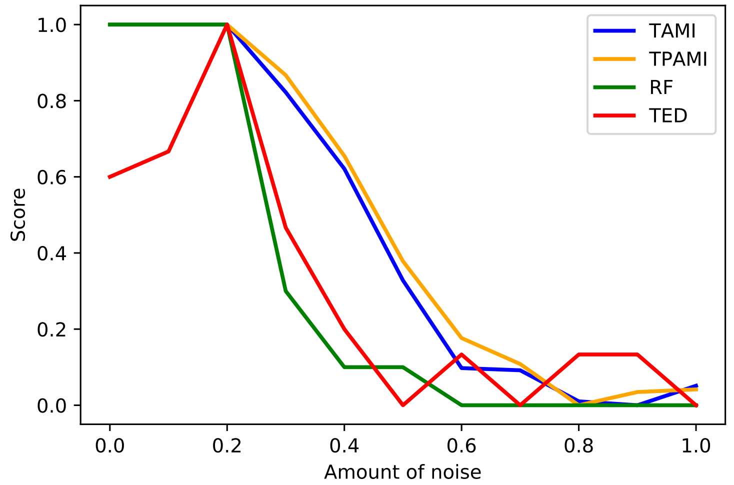
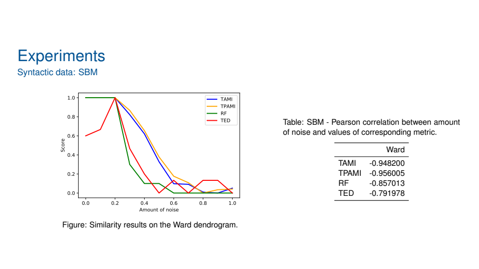


Figure: Similarity results on the Ward dendrogram.

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

	Ward
TAMI	-0.948200
TPAMI	-0.956005
RF	-0.857013
TED	-0.791978


Experiments
Syntactic data: SBM
Table: SBM - Pearson correlation between amount of noise and values of corresponding metric.
Figure: Similarity results on the Ward dendrogram.

1. We add noise $p_{shuffled}$ by randomly shuffling edges between nodes in the original graph G_{origin},After applying clustering algorithms: Ward, Louvain and Paris hierarchies of different qualities are obtained. Then we measure similarity.
2. TMI with AMI and PAMI behaves very similar in all experiments and outperforms two other metrics. These functions behave smoothly and coherent.
   In contrast, RF and TED behave very differently depending on the algorithm, which resembled in Pearson correlation score: both metrics show an unstable performance from one algorithm to another. For example, with the Ward clustering algorithm, they can capture similarity with a little amount of noise, but when it reaches the point $p_{shuffle} = 0.4$ both metrics drop to values around 0 and have some spikes when noise is high.

Experiments

Syntactic data: SBM

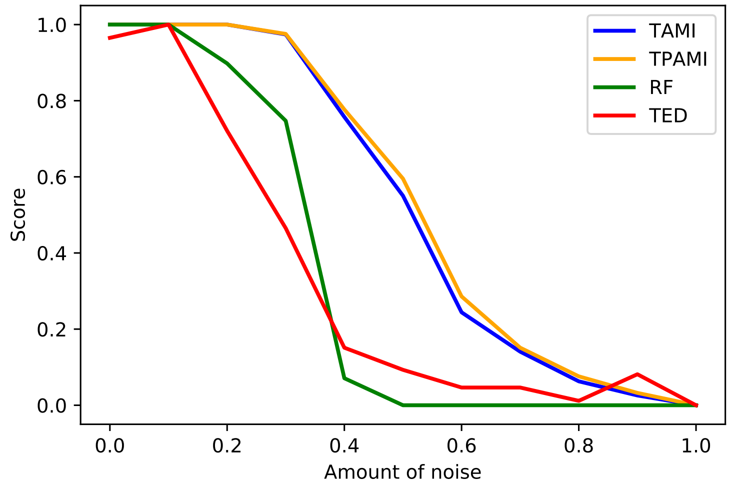
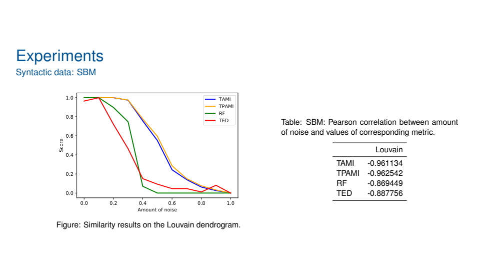


Figure: Similarity results on the Louvain dendrogram.

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

	Louvain
TAMI	-0.961134
TPAMI	-0.962542
RF	-0.869449
TED	-0.887756


Experiments
Syntactic data: SBM
Figure: Similarity results on the Louvain dendrogram.
Table: SBM: Pearson correlation between amount of noise and values of corresponding metric.

	Louvain
TAMI	-0.961134
TPAMI	-0.962542
RF	-0.869449
TED	-0.887756

1. Similar behaviour is observed for Lauvain algorithm.

Experiments

Syntactic data: SBM

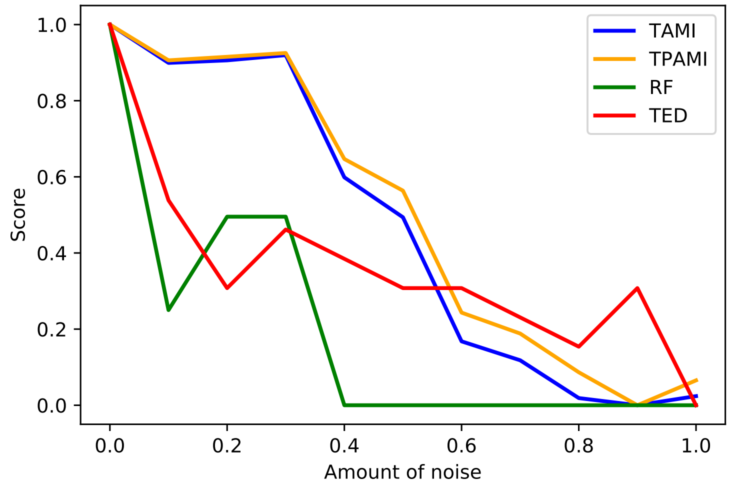
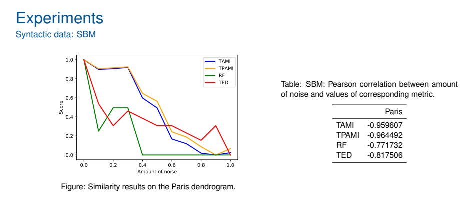


Figure: Similarity results on the Paris dendrogram.

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

	Paris
TAMI	-0.959607
TPAMI	-0.964492
RF	-0.771732
TED	-0.817506


Experiments
Syntactic data: SBM
Figure: Similarity results on the Paris dendrogram.
Table: SBM: Pearson correlation between amount of noise and values of corresponding metric.
Paris
TAMI -0.959607
TPAMI -0.964492
RF -0.771732
TED -0.817506

1. By using the Paris clustering algorithm, the final results are very unstable: RF shows almost always 0 similarity everywhere except for few spikes. Tree Edited Distance shows very high similarity even with a high amount of noise. These two metrics behave unstably and fail to capture actual dependence.

Experiments

Syntactic data: SBM

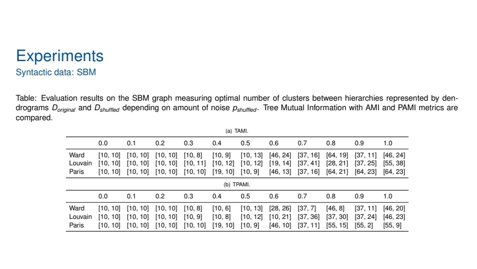
Table: Evaluation results on the SBM graph measuring optimal number of clusters between hierarchies represented by dendrograms $D_{original}$ and $D_{shuffled}$ depending on amount of noise $p_{shuffled}$. Tree Mutual Information with AMI and PAMI metrics are compared.

(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]


Experiments

Syntactic data: SBM

Table: Evaluation results on the SBM graph measuring optimal number of clusters between hierarchies represented by dendrograms $D_{original}$ and $D_{shuffled}$ depending on 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]

1. More in deep, in the Table (a) and (b) we can see dependency of optimal number of clusters for each tree, corresponding to the algorithm and the amount of noise $p_shuffled$. Now, lets analyse it:
   - TAMI and TPAMI identify the optimal number of clusters equal to 10 as in ground truth dedndrogram when the amount of noise is less than 0.3.
   - However, when noise is bigger than 0.5, fluctuation in the optimal number of clusters is quite significant for both metrics.

Experiments

Real datasets

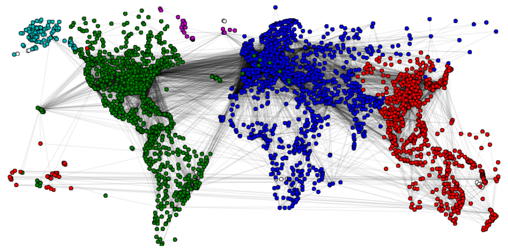


Figure: OpenFlights graph.

Table: Summary of the 2 datasets

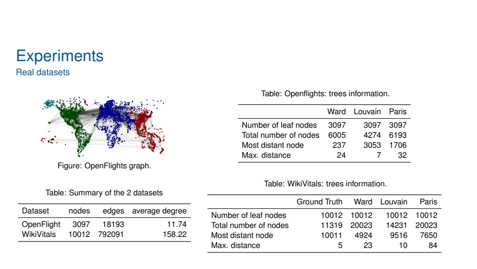
Dataset	nodes	edges	average degree
OpenFlight	3097	18193	11.74
WikiVitals	10012	792091	158.22

Table: 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

Table: 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



Experiments
Real datasets

Figure: OpenFlights graph.

Table: Summary of the 2 datasets

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

Table: 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

Table: 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

1. Next, we show the practical interest of TMI in terms of tree comparison. The experiments on real networks are performed on 2 datasets with various sizes and sparsity.
2. We conduct these experiments only with newly developed metrics. We disregard RF and TED metrics due to bad performance in the clustering scenario and inefficiency for large datasets.
3. OpenFlights is a weighted graph in which nodes represent airports and edges the number of flights. **We do not have a ground truth dendrogram for this graph.**
   The Wikivitals dataset is an unweighted graph where nodes represent Vital articles of Wikipedia . **We reconstruct the hierarchy from ground truth labels:** and use for evaluation.
4. We apply all three algorithms to the datasets and obtain trees with the following statistics, which you can observe in Tables on the right. We can see that the structure of these trees is very different.

Experiments

Real datasets: WikiVitals

Table: 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

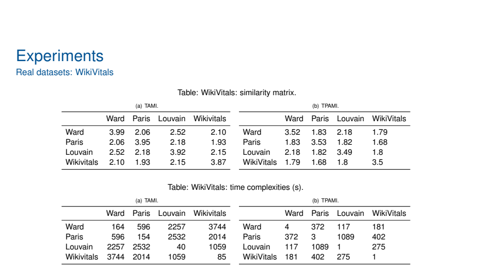
Table: WikiVitals: time complexities (s).

(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



Experiments

Real datasets: WikiVitals

Table: 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: 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

1. In order to spare the time I will only present results of WikiVitals dataset, where we know ground truth dendrogram. We compare all these dendrograms with each other. From the similarity matrix, we observe that both metrics TAMI and TPAMI identify the Louvain tree as the most similar to ground truth, while Ward takes the second spot and the Paris tree has the worst similarity rate. The outcome that the Louvain tree is the most similar to ground truth is not surprising, because they have alike structure: Louvain has only 7 levels of hierarchy and they are subdivided in the "general tree" way, meaning the number of clusters is not limited to be divided by 2. In the ground truth tree has 5 levels with similar properties. Two other algorithms: Ward and Paris tend to produce structure resembling binary tree.
2. The Table with time complexities presents us run time of each metric on certain dendrogram. TPAMI works much faster than TAMI metric, due to its smaller complexity by construction.

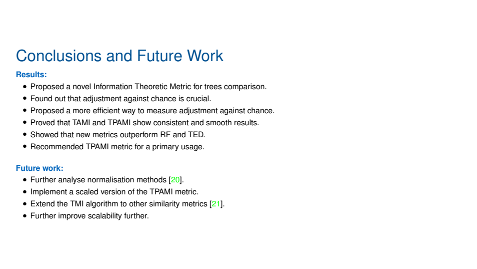
Conclusions and Future Work

Results:

- Proposed a novel Information Theoretic Metric for trees comparison.
- Found out that adjustment against chance is crucial.
- Proposed a more efficient way to measure adjustment against chance.
- Proved that TAMI and TPAMI show consistent and smooth results.
- Showed that new metrics outperform RF and TED.
- Recommended TPAMI metric for a primary usage.

Future work:

- Further analyse normalisation methods [20].
- Implement a scaled version of the TPAMI metric.
- Extend the TMI algorithm to other similarity metrics [21].
- Further improve scalability further.



Conclusions and Future Work

Results:

- Proposed a novel Information Theoretic Metric for trees comparison.
- Found out that adjustment against chance is crucial.
- Proposed a more efficient way to measure adjustment against chance.
- Proved that TAMI and TPAMI show consistent and smooth results.
- Showed that new metrics outperform RF and TED.
- Recommended TPAMI metric for a primary usage.

Future work:

- Further analyse normalisation methods [20].
- Implement a scaled version of the TPAMI metric.
- Extend the TMI algorithm to other similarity metrics [21].
- Further improve scalability further.

Conclusions:

- We proposed a novel Information Theoretic Metric and another way of adjusting mutual information against chance, that has a much lower complexity.
- Both TAMI and TPAMI show consistent and smooth results and outperform RF and TED metrics in all experiments.
- TPAMI tends to provide the same results as TAMI, but it involves much fewer computations and is therefore applicable to larger trees. Hence, we recommend it for primary usage.

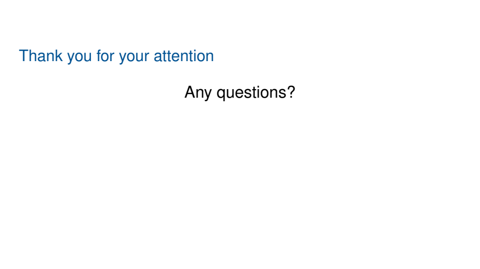
As a Future work:

- We plan to go deeper into normalisation studies and implement a scaled version of the TPAMI metric.
- We want to extend the TMI idea to other similarity metrics.
- It is necessary to test the novel metric on larger number of real datasets.

Thank you for your attention

Any questions?

└Thank you for your attention



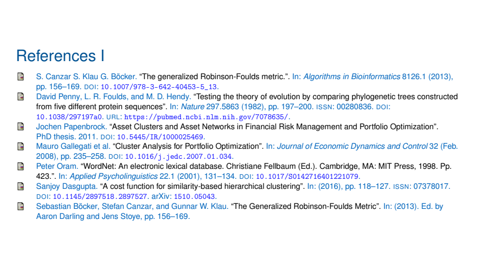
Thank you for your attention

Any questions?

References I

- 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`.
- 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/`.
- Jochen Papenbrock. “Asset Clusters and Asset Networks in Financial Risk Management and Portfolio Optimization”. PhD thesis. 2011. DOI: `10.5445/IR/1000025469`.
- 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`.
- 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`.
- 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`.
- 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.

└References



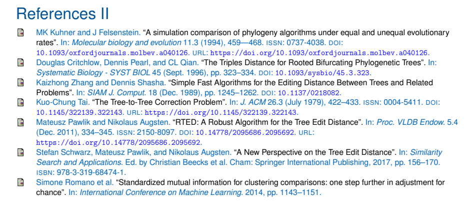
References I

- 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.
- 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/.
- Jochen Papenbrock. “Asset Clusters and Asset Networks in Financial Risk Management and Portfolio Optimization”. PhD thesis. 2011. DOI: 10.5445/IR/1000025469.
- 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.
- 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.
- 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.
- 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.

References II

- 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.
- 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.
- 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.
- 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.
- 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.
- 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.
- 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.

└References



References II

- 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.
- 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.
- 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.
- 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.
- 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.
- 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.
- 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.

References III

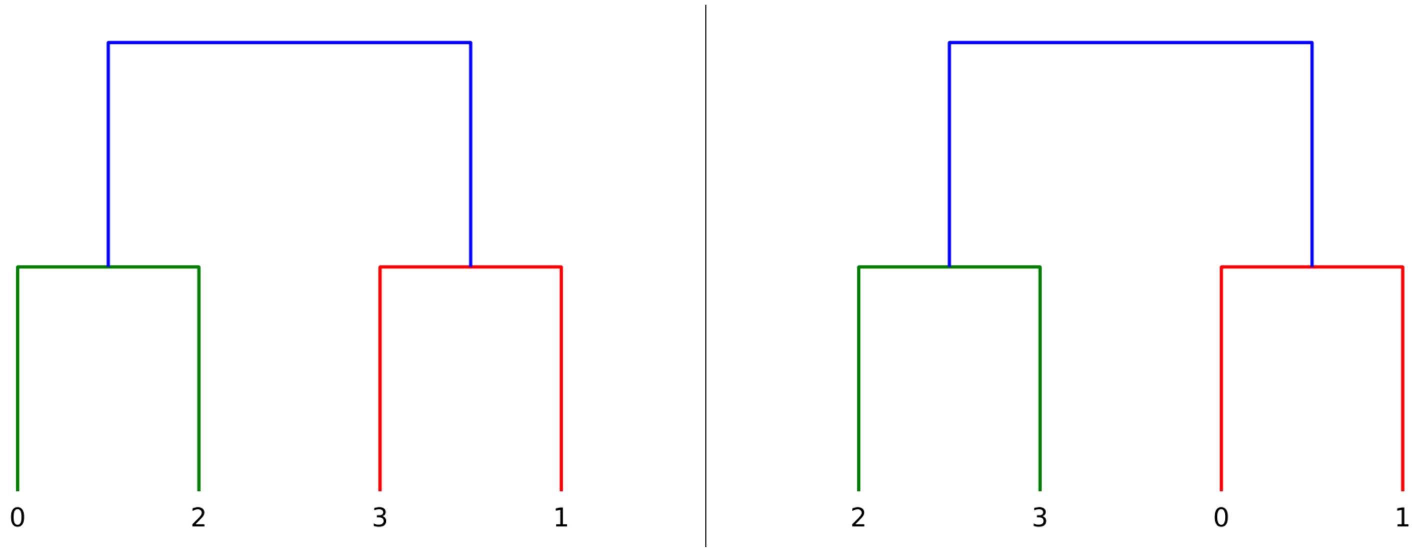
- Denys Lazarenko and Thomas Bonald. *Pairwise Adjusted Mutual Information*. 2021. arXiv: `2103.12641 [cs.LG]`.
- Olsen G. *Newick’s 8:45" Tree Format Standard*. Apr. 1990. URL: `http://evolution.genetics.washington.edu/phylip/newick_doc.html`.
- Joe H. Ward Jr. “Hierarchical Grouping to Optimize an 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`.
- 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`.
- Thomas Bonald et al. “Hierarchical Graph Clustering using Node Pair Sampling”. In: (2018). arXiv: `1806.01664`. URL: `http://arxiv.org/abs/1806.01664`.
- 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`.
- Simone Romano et al. “Adjusting for chance clustering comparison measures”. In: *The Journal of Machine Learning Research* 17.1 (2016), pp. 4635–4666.

References III

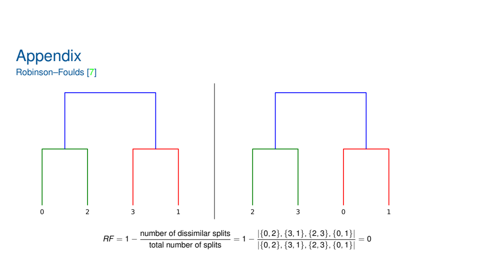
Denys Lazarenko and Thomas Bonald. *Pairwise Adjusted Mutual Information*. 2021. arXiv: 2103.12641 [cs.LG].
Olsen G. *Newick’s 8:45" Tree Format Standard*. Apr. 1990. URL: http://evolution.genetics.washington.edu/phylip/newick_doc.html.
Joe H. Ward Jr. “Hierarchical Grouping to Optimize an Objective 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.
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.
Thomas Bonald et al. “Hierarchical Graph Clustering using Node Pair Sampling”. In: (2018). arXiv: 1806.01664. URL: http://arxiv.org/abs/1806.01664.
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.
Simone Romano et al. “Adjusting for chance clustering comparison measures”. In: *The Journal of Machine Learning Research* 17.1 (2016), pp. 4635–4666.

Appendix

Robinson–Foulds [7]



$$RF = 1 - \frac{\text{number of dissimilar splits}}{\text{total number of splits}} = 1 - \frac{|\{0, 2\}, \{3, 1\}, \{2, 3\}, \{0, 1\}|}{|\{0, 2\}, \{3, 1\}, \{2, 3\}, \{0, 1\}|} = 0$$

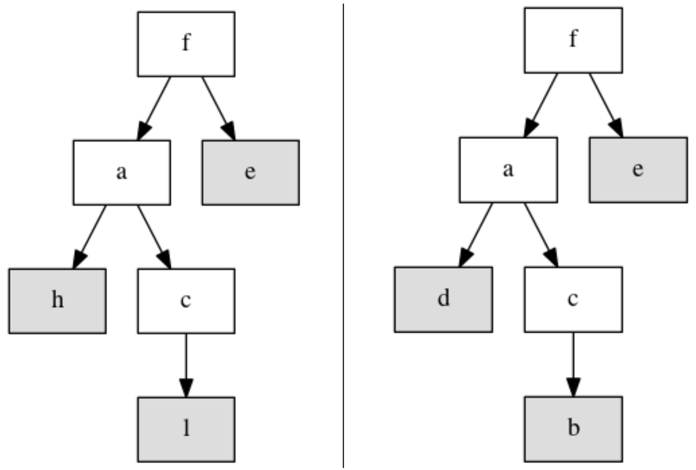

Appendix
Robinson–Foulds [7]

$RF = 1 - \frac{\text{number of dissimilar splits}}{\text{total number of splits}} = 1 - \frac{|\{0, 2\}, \{3, 1\}, \{2, 3\}, \{0, 1\}|}{|\{0, 2\}, \{3, 1\}, \{2, 3\}, \{0, 1\}|} = 0$

1.
   - Robinson–Foulds is a symmetric distance between two trees, which measures the number of branch splits present in one tree, but not in another and scores 1 for each division that is not matched.
   - Let's gain some understanding from a simple example: we can see that both trees has the same leaves, but differently partitioned. First tree has splits $(0, 2)$, $(3, 1)$ while second tree $(2, 3)$, $(0, 1)$. Despite the trees' same structure, we obtain 0 similarity, which is indeed a shortcoming in a clustering scenario.

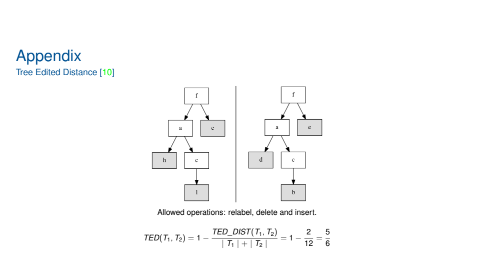
Appendix

Tree Edited Distance [10]



Allowed operations: relabel, delete and insert.

$$TED(T_1, T_2) = 1 - \frac{TED\_DIST(T_1, T_2)}{\mid T_1 \mid + \mid T_2 \mid} = 1 - \frac{2}{12} = \frac{5}{6}$$


Appendix
Tree Edited Distance [10]

Allowed operations: relabel, delete and insert.

$TED(T_1, T_2) = 1 - \frac{TED\_DIST(T_1, T_2)}{\mid T_1 \mid + \mid T_2 \mid} = 1 - \frac{2}{12} = \frac{5}{6}$

1. – Next concept which we are considering is **ordered tree edit distance**, which basically optimizes number of operations which are necessary to transform one tree into another.
   – There are 3 allowed operations: change one node label into another, delete or insert a node.
   – For example in these trees we see that leaves *h* and *l* in the left tree are in different positions than in the right one, but structurally trees are the same. Therefore, we can simply rename these nodes which results into distance equal to 2. To normalise this score, we divide it on the sum of total number of nodes in trees equal to 12. The total similarity is equal to 5 over 6.

Appendix

Agglomerative clustering algorithms

Ward. Ward seeks to reduce the number of squared disparities in all clusters. It is analogous to the objective function of K-means. Let $g(c)$ be the centroid of any cluster c

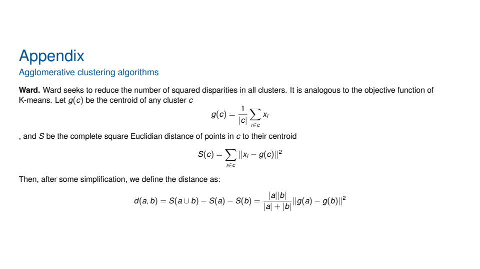
$$g(c) = \frac{1}{|c|}\sum_{i \in c} x_i$$

, and S be the complete square Euclidian 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:

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



Appendix

Agglomerative clustering algorithms

Ward. Ward seeks to reduce the number of squared disparities in all clusters. It is analogous to the objective function of K-means. 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 Euclidian 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:

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

1. It is typical agglomerative algorithm which uses Ward distance as a measure of proximity. ard seeks to reduce the number of squared disparities in all clusters. The distance is calculated as stated below and uses the function analogous to the objective function of K-means.

Appendix

Agglomerative clustering algorithms

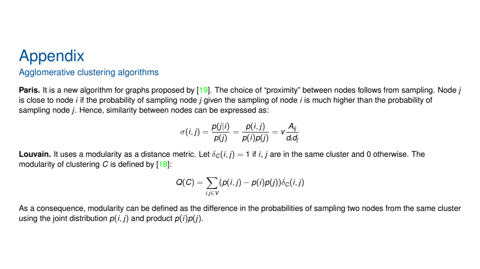
Paris. It is a new algorithm for graphs proposed by [19]. The choice of “proximity” between nodes follows from sampling. Node j is close to node i if the probability of sampling node j given the sampling of node i is much higher than the probability of sampling node j. Hence, similarity between nodes can be expressed as:

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

Louvain. It 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 [18]:

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

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 $p(i)p(j)$.



Appendix

Agglomerative clustering algorithms

Paris. It is a new algorithm for graphs proposed by [19]. 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:

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

Louvain. It 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 [18]:

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

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 $p(i)p(j)$.

1. **Paris.** It is a new algorithm for graphs where 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 by following formula:
   Louvain. It 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 Newman, look on the following formula. 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 $p(i)p(j)$.

Appendix

Proof of Theorem 1

Consider two items selected uniformly at random in $\{1, \dots, n\}$. Let A_{i_1}, B_{j_1} be the clusters of the first item, A_{i_2}, B_{j_2} be the clusters of the second item. In particular, these items belong respectively to the sets $A_{i_1} \cap B_{j_1}$ and $A_{i_2} \cap B_{j_2}$. The probability of this event is:

$$\frac{n_{i_1 j_1} n_{i_2 j_2}}{n^2}.$$

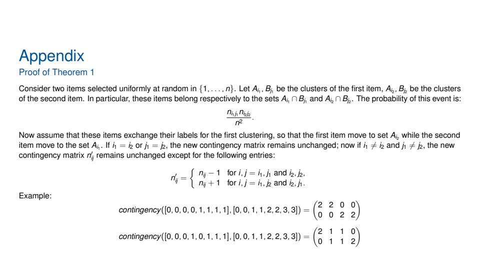
Now assume that these items exchange their labels for the first clustering, so that the first item move to set A_{i_2} while the second item move to the set A_{i_1}. If $i_1 = i_2$ or $j_1 = j_2$, the new contingency matrix remains unchanged; now if $i_1 \neq i_2$ and $j_1 \neq j_2$, the new contingency matrix n'_{ij} remains unchanged except for the following entries:

$$n'_{ij} = \begin{cases} n_{ij} - 1 & \text{for } i, j = i_1, j_1 \text{ and } i_2, j_2, \\ n_{ij} + 1 & \text{for } i, j = i_1, j_2 \text{ and } i_2, j_1. \end{cases}$$

Example:

$$\mathit{contingency}([0,0,0,0,1,1,1,1],[0,0,1,1,2,2,3,3]) = \begin{pmatrix} 2 & 2 & 0 & 0 \\ 0 & 0 & 2 & 2 \end{pmatrix}$$

$$\mathit{contingency}([0,0,0,1,0,1,1,1],[0,0,1,1,2,2,3,3]) = \begin{pmatrix} 2 & 1 & 1 & 0 \\ 0 & 1 & 1 & 2 \end{pmatrix}$$



Appendix

Proof of Theorem 1

Consider two items selected uniformly at random in $\{1, \dots, n\}$. Let A_{i_1}, B_{j_1} be the clusters of the first item, A_{i_2}, B_{j_2} be the clusters of the second item. In particular, these items belong respectively to the sets $A_{i_1} \cap B_{j_1}$ and $A_{i_2} \cap B_{j_2}$. The probability of this event is:

$$\frac{n_{i_1 j_1} n_{i_2 j_2}}{n^2}.$$

Now assume that these items exchange their labels for the first clustering, so that the first item move to set A_{i_2} while the second item move to the set A_{i_1}. If $i_1 = i_2$ or $j_1 = j_2$, the new contingency matrix remains unchanged; now if $i_1 \neq i_2$ and $j_1 \neq j_2$, the new contingency matrix n'_{ij} remains unchanged except for the following entries:

$$n'_{ij} = \begin{cases} n_{ij} - 1 & \text{for } i, j = i_1, j_1 \text{ and } i_2, j_2, \\ n_{ij} + 1 & \text{for } i, j = i_1, j_2 \text{ and } i_2, j_1. \end{cases}$$

Example:

$$\mathit{contingency}([0,0,0,0,1,1,1,1],[0,0,1,1,2,2,3,3]) = \begin{pmatrix} 2 & 2 & 0 & 0 \\ 0 & 0 & 2 & 2 \end{pmatrix}$$

$$\mathit{contingency}([0,0,0,1,0,1,1,1],[0,0,1,1,2,2,3,3]) = \begin{pmatrix} 2 & 1 & 1 & 0 \\ 0 & 1 & 1 & 2 \end{pmatrix}$$

1. Let's have a look into the Proof of Theorem. Distributions of clusters stays always the same we only shuffle their positions. We start from a simple example: we have two clusterings: cluster A has 2 different labels and cluster B has 4. We randomly select two items in clusters A and B, let in our case it be elements on position 4 and 5 and exchange their labels. As you can see the contingency matrix changes respectively: its elements decreases or increases by 1 or stay the same.
   More formally: we consider two items selected uniformly at random in $\{1, \dots, n\}$. Let A_{i_1}, B_{j_1} be the clusters of the first item, A_{i_2}, B_{j_2} be the clusters of the second item. In particular, these items belong respectively to the intersection of sets A_{i_1} and B_{j_1} and the same for A_{i_2} and B_{j_2}. The probability of this event is:

$$\frac{n_{i_1 j_1} n_{i_2 j_2}}{n^2}.$$

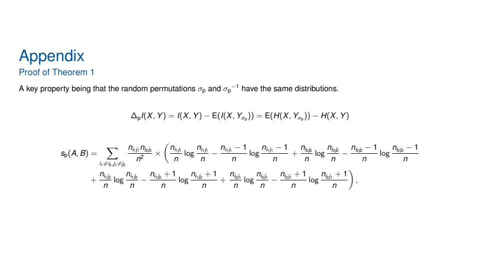
Appendix

Proof of Theorem 1

A key property being that the random permutations σ_{p} and ${\sigma_{\mathrm{p}}}^{-1}$ have the same distributions.

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

$$\begin{aligned} s_{\mathrm{p}}(A,B) = \sum_{i_1 \neq i_2, j_1 \neq j_2} \frac{n_{i_1 j_1} n_{i_2 j_2}}{n^2} \times \Bigg(& \frac{n_{i_1 j_1}}{n} \log \frac{n_{i_1 j_1}}{n} - \frac{n_{i_1 j_1} - 1}{n} \log \frac{n_{i_1 j_1} - 1}{n} + \frac{n_{i_2 j_2}}{n} \log \frac{n_{i_2 j_2}}{n} - \frac{n_{i_2 j_2} - 1}{n} \log \frac{n_{i_2 j_2} - 1}{n} \\ & + \frac{n_{i_1 j_2}}{n} \log \frac{n_{i_1 j_2}}{n} - \frac{n_{i_1 j_2} + 1}{n} \log \frac{n_{i_1 j_2} + 1}{n} + \frac{n_{i_2 j_1}}{n} \log \frac{n_{i_2 j_1}}{n} - \frac{n_{i_2 j_1} + 1}{n} \log \frac{n_{i_2 j_1} + 1}{n} \Bigg), \end{aligned}$$



1. Using formula of pairwise adjusted mutual information, which is exactly the same definition as the adjusted mutual information, except for the considered permutations σ_{p}. We obtain the similarity between clusterings A and B. We simply consider all options from the previous slide for the contingency matrix. Then we can group summation terms as follows.

Appendix

Proof of Theorem 1

where by convention, $x \log x = 0$ for any $x \leq 0$. Observing that for any given i_1, j_1,

$$\sum_{i_2 \neq i_1, j_2 \neq j_1} n_{i_1 j_1} n_{i_2 j_2} = n_{i_1 j_1}(n - a_{i_1} - b_{j_1} + n_{i_1 j_1}),$$

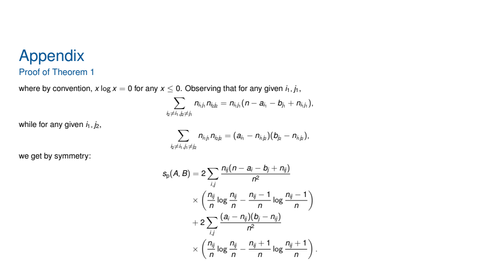
while for any given i_1, j_2,

$$\sum_{i_2 \neq i_1, j_1 \neq j_2} n_{i_1 j_1} n_{i_2 j_2} = (a_{i_1} - n_{i_1 j_2})(b_{j_2} - n_{i_1 j_2}),$$

we get by symmetry:

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

└Appendix



Appendix

Proof of Theorem 1

where by convention, $x \log x = 0$ for any $x \leq 0$. Observing that for any given i_1, j_1,

$$\sum_{i_2 \neq i_1, j_2 \neq j_1} n_{i_1 j_1} n_{i_2 j_2} = n_{i_1 j_1}(n - a_{i_1} - b_{j_1} + n_{i_1 j_1}),$$

while for any given i_1, j_2,

$$\sum_{i_2 \neq i_1, j_1 \neq j_2} n_{i_1 j_1} n_{i_2 j_2} = (a_{i_1} - n_{i_1 j_2})(b_{j_2} - n_{i_1 j_2}),$$

we get by symmetry:

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

1. Finally, the formula of pairwise similarity takes the form as was stated before.

Experiments

Real datasets: Openflights

Table: 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

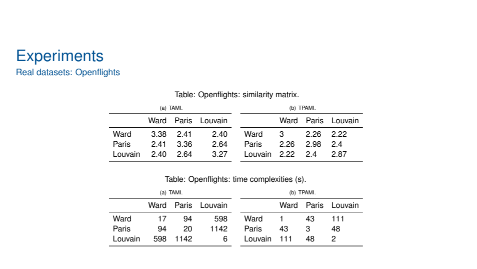
Table: Openflights: time complexities (s).

(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



Experiments
Real datasets: Openflights

Table: 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

Table: 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

1. We compare all these dendrograms with each other. We can see from similarity matrix that TAMI and TPAMI 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 and the last is Louvain. As was mentioned before, magnitude can vary because, we do not use normalisation.