UCLouvain

École polytechnique de Louvain

Exploratory analysis of network data based on the Bag-of-Paths framework

Clustering & Embedding

Author: **Emeline Christoph**
Supervisor: **Marco Saerens**
Readers: **Sylvain Courtain, Cyril De Bodt**
Academic year 2021–2022
Master [120] in Computer Science and Engineering

First of all, I would like to thank my supervisor, Pr. Marco Saerens, for his involvement, his advice, his availability and his precious help throughout this year at each stage of the elaboration of this work.

I would also like to thank Sylvain Courtain for his availability, for answering my questions and for his help throughout the experimental part.

Furthermore, I am grateful to Lauranne for her proofreading, advice and support during the writing of this thesis.

Last but not least, I would like to thank all my family, especially my parents, for their unconditional support and love over the past five years, in particular this one, but also for their involvement in every stage of my life.

Contents

1 Introduction

The study of graphs is a field that has seen a growing of interest in recent years. Nowadays, it is a rapidly advancing field that spread through a wide variety of sectors (technological, social, biological ...) and is useful for various real-world scenarios (World Wide Web, User interest ...). The analysis of these graphs through theoretical tools in constant development has made it possible to extract the hidden knowledge they contain and to capture patterns of interactions between their nodes [49].

Many applications related to the data mining and machine learning fields make good use of this extracted information. One such application is network visualization, which computes a representation of the nodes in a two- or three-dimensional space and draws this representation on a two- or three-dimensional graph where each node is coloured according to the cluster to which it belongs. Others are link prediction and network reconstruction which predict which links in the graph may appear in the future or are missing. Another widely used application is node classification, which assigns a label to each unlabelled node based on the rules learned by training a prediction model on the labelled nodes. Node recommendation is a task that provides a user with the most interesting nodes for him based on similarity criteria with the nodes he has rated well. The last example is node clustering presented in Section 5.1, which groups nodes into communities such that nodes in the same group are more similar to each other than to nodes in other groups. An often used clustering algorithm is the k-means algorithm described in Subsection 5.1.1. Clustering provides insight into the organisation of the graph [3, 40].

To analyze a graph, a well known and efficient technique is graph embedding. It allows converting a graph into its representation in a low dimensional space while preserving as much as possible the properties and structure of the graph. The embedding is used to extract most of the characteristics of a node and represent it by a feature vector containing this condensed information. This reduces the initial volume of data to retain only the essential features of the original data and provides a convenient way to manipulate those data for analysis [3, 40]. The graph embedding technique is related to two traditional research problems, namely the Graph Drawing and the Graph Representation Learning [25]. It is one of the hot topics in the fields of data mining and machine learning in recent years. Evaluating the quality of a given embedding is a complex task that strongly depends on the properties of the graphs that are to be preserved. However, it is expected that the embedding representation of the nodes captures and preserves the structure and semantic information of the graph.

There are numerous embedding techniques in the literature, such as Matrix factorization based-methods, Deep Learning based-methods or kernels based-methods [3]. Matrix factorization based-methods represent the properties of the graph as a matrix and factor it to obtain the embedding representation. They can be separated into two categories, the Graph Laplacian eigenmaps [2] and the node proximity matrices, such as HOPE [50]. Deep Learning based-methods which apply deep learning models to a graph can also be divided into two categories. The first one uses random-walks on the graph, such as DeepWalk [51] and node2vec [24], while the second does not use random-walks, such as GCN [31] and GNN [56]. Kernel based-methods, such as the Bag-of-paths framework described in Chapter 3, define distance measures between nodes through a kernel from which embedding is computed.

The objective of this thesis will be to analyze the performance of a new distance measure, i.e. the Poisson surprisal distance, compared to other well-defined distance measures when used to produce an embedding either in a low-dimensional space with 5% of the total dimensionality or in three-dimensional space. Two methods will be considered to compute the embedding from the distance matrix, namely the classical multidimensional scaling and the t-Distributed Stochastic Neighbor Embedding. The use of this latter method is based on the work done by A. M. Safi for his Master's thesis [54]. Three techniques for evaluating the quality of an embedding will be used to answer the following questions

1.a. *Which combinations (distance measure, embedding method) provide the best results in a three-dimensional node embedding task and, in this context, how does the introduced Poisson surprisal distance perform?*

1.b. *Which distances provide the best results in a low-dimensional (5%) node embedding task and, in this context, how does the introduced Poisson surprisal distance perform?*

The second main research question that will be addressed in this thesis aims to compare the community detection made on the kernel computed on a graph to the community detection made on the embedding computed on this graph. It will answer the following question

2. *Which combinations (distance measure, clustering method) provide the best results in a node clustering task and, in this context, how does the introduced Poisson surprisal distance perform?*

This thesis will also introduce a new embedding technique, i.e. the bag-of-paths embedding, that, as the name implies, is defined on the bag-of-paths framework and will use a gradient descent to improve the initial embedding produced. It will use the same embedding evaluation techniques used in Questions 1.a. and 1.b. to answer the following questions

3.a. *Which embedding provides the best results on a three-dimensional node embedding task and, in this context, how does the introduced bag-of-paths embedding perform?*

3.b. *Which embedding provides the best results on a low-dimensional (5%) node embedding task and, in this context, how does the introduced bag-of-paths embedding perform?*

To answer these questions, 17 datasets for which a ground-truth partition in communities is known will be used. The code used to perform the experiments can be found in the following toolbox[1]. The code whose provenance is not referenced in the corresponding theoretical part was either developed by the research group led by M. Saerens or written especially for these experiments.

This thesis will first present an overview of relevant graph theory by explaining what a graph is and by defining some useful matrices on a graph. Chapter 2 will also introduce the concept of similarity and kernel matrices along with well-defined distances on a graph. Chapter 3 will introduce the bag-of-paths framework and give an overview of some well-defined distances in this framework, but also a formal definition of the new investigated Poisson Surprisal distance. Chapter 4 will define the two methods used to compute an embedding on a distance matrix as well as the new bag-of-paths embedding method studied while Chapter 5 will explain the three techniques used to evaluate the quality of an embedding. Chapter 6 will first define the quality measures used to evaluate the quality of community detection and the statistical tests used on the scores computed by each technique of the fifth chapter, and will then describe the experimental methodology used. The following chapter, Chapter 7, will present the results obtained for each research questions and discuss them. Finally, a conclusion will be presented to summarise the main results obtained and describe the limitations of the current work.

---

[1] This repository is private. Access is provided via the GitHub account *thesis-access* with the password *echristoph_thesis_2022* and the access token *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*

Graphs and Networks

2

This first chapter aims to introduce the concept of graphs along with some useful matrices and notations that will be used in the following chapters of this thesis. For the interested readers, a more comprehensive introduction to the graph theory is presented in [16, 49] from which this chapter is inspired. This chapter will then explain the notion of similarity and dissimilarity matrices between nodes as well as the concept of kernel matrix. It will then introduce some distance matrices used as a reference for the new distance described in Chapter 3.

2.1 Basic Concepts

A network, also called a graph, G is a structure composed of a finite non-empty set of nodes (or vertices) $\mathcal{V}$ of size n that are connected by a set of edges (or links) $\mathcal{E}$ of size m. Such a graph will be denoted $G = (\mathcal{V}, \mathcal{E})$. The nodes usually represent objects, which are linked by edges when a relationship exists between them. For example, the nodes of the World Wide Web network are web pages and they are connected when there is a hyperlink from one to another.

An edge that connects a node to itself is called a self-edge. When there is more than one edge between two pairs of nodes, the edges are called multi-edges. A simple graph does not contain any self-edges or multi-edges. An example of such a graph is shown in Figure 2.1.

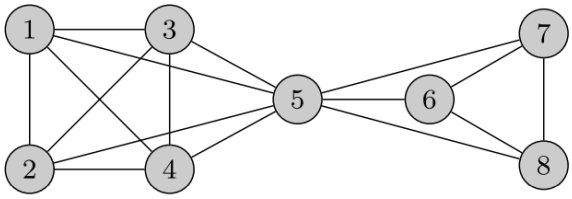

Fig. 2.1.: A simple unweighted undirected graph [32]

A graph is said to be directed when each edge has a direction. Considering a pair of nodes $i, j \in \mathcal{V}$, a directed edge (i, j) runs from node i to node j and is distinct from the edge (j, i) that runs from node j to node i. Whereas in an undirected graph the edges (i, j) and (j, i) will reference the same connection between nodes i and j.

A graph can be weighted or unweighted. In a weighted graph, each edge (i, j) will be associated with a weight w_{ij} which represents a degree of similarity (or closeness) between nodes i and j, this weight can be positive or negative.

The graphs that will be used in this thesis are simple undirected graphs, some weighted and some unweighted.

2.1.1 Adjacency Matrix

One way of mathematically representing a graph is to use an adjacency matrix $\mathbf{A}$. For a weighted graph, this matrix of size $n \times n$ can be defined as follows

$$a_{ij} = [\mathbf{A}]_{ij} = \begin{cases} w_{ij}, & \text{if } (i,j) \in \mathcal{E} \text{ , i.e. if there exists an edge between node } i \text{ and node } j \\ 0, & \text{otherwise} \end{cases} \tag{2.1}$$

If the graph is unweighted, w_{ij} will be equal to 1 for each existing edge.

The adjacency matrix of a simple undirected graph has several inherent properties. First, it will always be symmetric, i.e. $a_{ij} = a_{ji}$. Second, the fact that the graph is simple implies that it contains no self-loops, so all the diagonal elements of $\mathbf{A}$ (i.e. a_{ii}) will be equal to 0.

Two additional pieces of information can be derived from the adjacency matrix. The first one is the volume of the graph, which corresponds to the sum over all elements of $\mathbf{A}$

$$\text{vol}(G) = \sum_{i,j=1}^{n} a_{ij} = a_{\bullet\bullet} \tag{2.2}$$

If the graph is unweighted, the volume can also be computed as $\text{vol}(G) = 2m$, with m the number of edges of the graph.

The second piece of information that can be deduced is the degree of each node d_i, which in an undirected graph corresponds to the number of edges connected to that node,

$$d_i = \sum_{j=1}^{n} a_{ij} = a_{i\bullet} \tag{2.3}$$

The degree vector $\mathbf{d}$ of size $n \times 1$ can thus be defined as follows

$$\mathbf{d} = \mathbf{A}\mathbf{e} \tag{2.4}$$

where $\mathbf{e}$ is the full unit column vector of size $n \times 1$.
From Equation 2.4, the degree matrix $\mathbf{D}$ can be defined as a matrix $n \times n$ where the diagonal is the degree vector and the other elements are 0.

2.1.2 Cost Matrix

Another matrix that can be defined on a graph is the cost matrix $\mathbf{C}$. This matrix is based on the non-negative cost associated with each edge of the graph rather than the affinities of the adjacency matrix. The cost matrix can be defined as follows

$$[\mathbf{C}]_{ij} = \begin{cases} c_{ij}, & \text{if } (i,j) \in \mathcal{E} \\ \infty, & \text{otherwise} \end{cases} \tag{2.5}$$

These costs can sometimes be computed as $c_{ij} = \frac{1}{a_{ij}}$. In this situation, a parallel can be drawn with an electrical network where the affinities a_{ij} would represent the conductances and the costs c_{ij} represent the resistances.

2.1.3 Laplacian Matrix

The Laplacian matrix $\mathbf{L}$ is a matrix of size $n \times n$ closely related to the adjacency matrix which gives additional information on the network [19]. For an undirected graph without self-loops, it can be computed as follows

$$\mathbf{L} = \mathbf{D} - \mathbf{A} \tag{2.6}$$

where $\mathbf{D}$ is the degree matrix and $\mathbf{A}$ the adjacency matrix as defined in Subsection 2.1.1.
Equivalently, each element of $\mathbf{L}$ can be defined as

$$[\mathbf{L}]_{ij} = \begin{cases} d_i, & \text{if } i = j \\ -w_{ij}, & \text{if } i \neq j \text{ and } (i,j) \in \mathcal{E} \\ 0, & \text{otherwise} \end{cases} \tag{2.7}$$

If the graph is unweighted w_{ij} is equal to 1.

An important property of this matrix is that it is positive semi-definite, which means that all its eigenvalues are non-negative. According to Equations 2.4 and 2.6, the Laplacian matrix also has each row and each column summing to 0

$$\mathbf{L}\mathbf{e} = \mathbf{0} \text{ and } \mathbf{e}^\mathsf{T}\mathbf{L} = \mathbf{0}^\mathsf{T} \tag{2.8}$$

with $\mathbf{0}$ a column vector of size n full of 0's. $\mathbf{L}$ is said to be doubly centered [19].

From this property follows that there is a linear dependence between one column/row and the others, which means that $\mathbf{L}$ is rank-deficient. The inverse of $\mathbf{L}$ is therefore not well defined. However, since $\mathbf{L}$ is real, symmetric and positive semi-definite, the Moore-Penrose pseudoinverse matrix $\mathbf{L}^+$ can be computed on its singular value decomposition [1, 16]. Denoting the eigenvalues and eigenvectors of $\mathbf{L}$ as $\{\lambda_k, \mathbf{u}_k\}_{k=1}^n$, the Moore-Penrose pseudoinverse matrix $\mathbf{L}^+$ is

$$\mathbf{L}^+ = \sum_{\substack{k=1 \\ \lambda_k > 0}}^{n} \frac{1}{\lambda_k} \mathbf{u}_k \mathbf{u}_k^\mathsf{T} \tag{2.9}$$

The condition on the sum is necessary because $\mathbf{L}$ being rank-deficient implies that at least one eigenvalue will be equal to 0 and the zero eigenvalues should not be taken into account.

2.1.4 Transition Matrix

The transition probability matrix $\mathbf{P}$, or simply transition matrix, is a stochastic matrix defined over the random walk on a graph G. A random walk is a discrete Markov chain that describes the sequences of nodes visited by a random walker on the graph. The state of the Markov chain at time t will be represented by a random variable $s(t)$ and the probability of being in state i at time t will be $\mathrm{P}(s(t) = i)$. The probability that a random walker in state i at time t moves to state j at time $t+1$ is known and correspond to the transition probability p_{ij}

$$p_{ij} = [\mathbf{P}]_{ij} = \mathrm{P}(s(t+1) = j \mid s(t) = i) = \frac{a_{ij}}{a_{i\bullet}} \tag{2.10}$$

where a_{ij} is defined in Equation 2.1 and $a_{i\bullet}$ in Equation 2.3.
The transition probability from one state to another is proportional to the weight of the edge connecting them. Since the Markov chain considered is of the first order, the transition probability does not depend on the past, but only on the current state at time t. Therefore, the transition matrix $\mathbf{P}$ is the matrix containing the one-step transition probabilities p_{ij} for all i, j.

2.2 Reference Distances and Kernels Between Nodes

One way to define a notion of distance between two nodes is to measure how alike they are, this is the notion of similarity. This notion is not standardised, but some properties are reasonably expected. First, the similarity matrix containing the similarities between each pair of nodes must be symmetric such that $s_{ij} = s_{ji}$ for all nodes i, j. Second, if the similarity between the two nodes i and k increases, s_{ik} should also increase. Third, the similarity between two nodes i and k is always greater than or equal to 0, but less than or equal to the similarity on the node itself (s_{ii} or s_{kk}). The similarity measure can be used to define a distance matrix $\Delta_{ij} = (s_{ii} + s_{jj} - 2s_{ij})^{\frac{1}{2}}$ [16].

The opposite notion of similarity is dissimilarity which, unlike the former, is well standardised [16, 47]. Three properties must be respected by a dissimilarity measure:

- Non-negativity: $\Delta_{ik} \geq 0$ for all i, k
- Symmetry: $\Delta_{ik} = \Delta_{ki}$ for all i, k
- Reflexivity: $\Delta_{ii} = 0$ for all i

For the dissimilarity measure to be a distance metric, an additional condition must be respected:

- Triangle inequality: $\Delta_{ik} \leq \Delta_{ij} + \Delta_{jk}$ for all i, j, k

This distance metric is said to be Euclidean if a configuration in the Euclidean space that preserves exactly the initial distance between the objects exists [16].

A kernel is a function that computes the similarity between objects (e.g. nodes) that cannot be represented naturally by a set of features. This similarity will be implicitly equivalent to computing the inner product of the vector representation of these objects in a higher-dimensional space, i.e. the embedding space [16, 17].

The kernel matrix $\mathbf{K}$ of size $n \times n$ resulting from the computation of the kernel function on each pair of nodes is considered valid if it is a symmetric positive semi-definite matrix (i.e. all its eigenvalues are nonnegative). If the kernel matrix is symmetric but not positive semi-definite, it is called a similarity matrix. Any similarity matrix can be transformed into a kernel matrix by setting all negative eigenvalues of the spectral decomposition of the similarity matrix to 0 [16]. A kernel can be centered using the following equation [16, 18]

$$\mathbf{K} = \mathbf{HKH}, \text{ with } \mathbf{H} = \mathbf{I} - \frac{\mathbf{e}\mathbf{e}^\mathsf{T}}{n} \tag{2.11}$$

In this thesis, the kernels are computed using the multidimensional scaling technique (see Section 4.1) on the distance matrices $\mathbf{\Delta}$

$$\mathbf{K} = -\tfrac{1}{2}\mathbf{H}\mathbf{\Delta}^{(2)}\mathbf{H}, \text{ with } \mathbf{\Delta}^{(2)} = \mathbf{\Delta} \circ \mathbf{\Delta} \tag{2.12}$$

where $\circ$ is the elementwise product and thus $\mathbf{\Delta}^{(2)}$ is the distance matrix squared. If the distance matrix can be embedded in a Euclidean space, the kernel will be a symmetric positive semi-definite matrix (i.e. a valid kernel) [16, 18].

Conversely, a Euclidean distance matrix can be derived from a kernel matrix

$$\mathbf{\Delta}^{(2)} = \mathbf{diag}(\mathbf{K})\mathbf{e}^\mathsf{T} + \mathbf{e}(\mathbf{diag}(\mathbf{K}))^\mathsf{T} - 2\mathbf{K} \tag{2.13}$$

where $\mathbf{diag}(\mathbf{K})$ is the column vector containing the diagonal of the square matrix $\mathbf{K}$ [16, 18].

2.2.1 Shortest-Path Distance

A popular problem when discussing graphs is to find the shortest paths between a pair of nodes. This problem can be transformed into a distance matrix, considering that the shorter the distance between two nodes, the more similar they are [16]. Many algorithms have been designed to solve the shortest path problem such as those of Dijkstra or Bellman-Ford [58]. An important convention is that $\Delta_{ij}^{SP} = \infty$ when there is no path between nodes i and j [16, 49], the shortest-path distance can be computed recursively as follows

$$\mathbf{\Delta}_{\text{SP}}^{k} = \begin{cases} \mathbf{C} & \text{when } k = 0 \\ \min\left(\mathbf{\Delta}_{\text{SP}}^{(k-1)}, \left[\mathbf{\Delta}_{\text{SP}}^{(k-1)}\right]_{\bullet k}\mathbf{e}^\mathsf{T} + \mathbf{e}\left[\mathbf{\Delta}_{\text{SP}}^{(k-1)}\right]_{k\bullet}\right) & \text{when } k \geq 1 \end{cases} \tag{2.14}$$

where $\mathbf{C}$ is the cost matrix defined in Subsection 2.1.2, k belongs to the interval $[0...n]$ and the minimum is taken independently on each element of the matrix. The main drawback of the shortest-path distance is that it fails to integrate the global structure of the graph. Indeed, when computing the shortest path between two nodes, it does not take into account the degree of connectivity of these nodes, whereas strongly connected nodes should have a higher similarity than weakly connected ones [16].

2.2.2 Commute Time Distance

The commute time distance [16, 19, 53, 65] is a distance derived from the transition matrix $\mathbf{P}$ of a graph G (see Subsection 2.1.4) and thus derived from the random walk on this graph. Two quantities computed on this matrix will be necessary to compute the commute time distance. The first one is the average first

passage time $m(i,j)$ which computes the number of steps expected for a random walker to reach node j starting from node i,

$$\begin{cases} m(i,j) = 1 + \sum_{k=1}^{n} p_{ik}\, m(k,j) & \text{for } i \neq j \\ m(i,j) = 0 & \text{for } i = j \end{cases} \tag{2.15}$$

Another way to define $m(i,j)$ is to use the pseudoinverse of the Laplacian matrix defined in Equation 2.9 such that

$$m(i,j) = \sum_{k=1}^{n} (l_{ik}^{+} - l_{ij}^{+} - l_{jk}^{+} + l_{jj}^{+}) d_k \tag{2.16}$$

where d_k is the degree of the node k. The second quantity is the average commute time $n(i,j)$ which computes the average number of steps a random walker needs, starting from node i, to go to node j and get back to i,

$$n(i,j) = m(i,j) + m(j,i) \tag{2.17}$$

Using the pseudoinverse of the Laplacian matrix, Equation 2.17 can be rewritten as

$$n(i,j) = \mathrm{vol}(G)(l_{ii}^{+} + l_{jj}^{+} - 2l_{ij}^{+}) = \mathrm{vol}(G)(\mathbf{e}_i - \mathbf{e}_j)^{\mathsf{T}} \mathbf{L}^{+} (\mathbf{e}_i - \mathbf{e}_j) \tag{2.18}$$

Given that the average commute time $n(i,j)$ satisfies the four properties for being a distance metric expressed in Section 2.2, the commute time distance can be defined as follows

$$[\Delta_{\mathrm{CT}}]_{ij} = n(i,j) = \mathrm{vol}(G)(\mathbf{e}_i - \mathbf{e}_j)^{\mathsf{T}} \mathbf{L}^{+} (\mathbf{e}_i - \mathbf{e}_j) \tag{2.19}$$

or equivalently in matrix form

$$\mathbf{\Delta}_{\mathrm{CT}} = \mathrm{vol}(G)\big(\mathbf{diag}(\mathbf{L}^{+})\mathbf{e}^{\mathsf{T}} + \mathbf{e}(\mathbf{diag}(\mathbf{L}^{+}))^{\mathsf{T}} - 2\mathbf{L}^{+}\big) \tag{2.20}$$

The square root of $\mathbf{\Delta}_{\mathrm{CT}}$ is also a distance. As $\mathbf{L}^{+}$ is positive semi-definite, the Equation 2.13 guarantees that the distance is Euclidean,

$$\mathbf{\Delta}_{\mathrm{ECT}} = \mathbf{\Delta}_{\mathrm{CT}}^{\frac{1}{2}} = \sqrt{\mathrm{vol}(G)}\big(\mathbf{diag}(\mathbf{L}^{+})\mathbf{e}^{\mathsf{T}} + \mathbf{e}(\mathbf{diag}(\mathbf{L}^{+}))^{\mathsf{T}} - 2\mathbf{L}^{+}\big)^{\frac{1}{2}} \tag{2.21}$$

The associated kernel of the Euclidean commute time distance is therefore $\mathrm{vol}(G)\mathbf{L}^{+}$. By getting rid off the scale factor which is the volume of the graph, the commute time kernel can be defined as [17]

$$\mathbf{K}_{\mathrm{CT}} = \mathbf{L}^{+} \tag{2.22}$$

Unlike the shortest-path distance, the commute time distance does take into account the degree of connectivity between two nodes when computing their distance. It decreases as the number of paths connecting the two nodes increases. The more paths there are between the nodes, the smaller their distance. However, when the graph becomes large, the stationary distribution of the natural random walk on the graph strongly influences $\mathbf{\Delta}_{\mathrm{CT}}$ and it becomes too sensitive to the degree of the starting and ending nodes. This is called the "lost in space" problem [64]. This means that in large graph, the commute time distance fails to take into account the global properties of the graph [16, 32, 65].

A variation of the commute time distance that allows to correct this behaviour has been proposed in [64]. It suggests using in Equation 2.19 the normalized Laplacian matrix $\tilde{\mathbf{L}}^{+}$ instead of the Moore-Penrose pseudoinverse Laplacian matrix $\mathbf{L}^{+}$. The corrected commute time distance $\mathbf{\Delta}_{\mathrm{CCT}}$ can thus be computed from the commute time distance as follows [16, 64]

$$\mathbf{\Delta}_{\mathrm{CCT}} = \mathbf{\Delta}_{\mathrm{CT}} - \mathrm{vol}(G)\Big(\mathbf{diag}(\mathbf{S})\mathbf{e}^{\mathsf{T}} + \mathbf{e}(\mathbf{diag}(\mathbf{S}))^{\mathsf{T}} - 2\mathbf{S}\Big) \text{ with } \mathbf{S} = \mathbf{D}^{-1} + \mathbf{D}^{-1}\mathbf{A}\mathbf{D}^{-1} \tag{2.23}$$

By using the Equation 2.12, the corrected commute time kernel $\mathbf{K}_{\mathrm{CCT}}$ associated with this Euclidean distance can be computed as

$$\mathbf{K}_{\mathrm{CCT}} = \mathbf{H}\mathbf{D}^{-\frac{1}{2}}\mathbf{M}(\mathbf{I} - \mathbf{M})^{-1}\mathbf{M}\mathbf{D}^{-\frac{1}{2}}\mathbf{H} \text{ with } \mathbf{M} = \mathbf{D}^{-\frac{1}{2}}\left(\mathbf{A} - \frac{\mathbf{d}\mathbf{d}^{\mathsf{T}}}{\mathrm{vol}(G)}\right)\mathbf{D}^{-\frac{1}{2}} \tag{2.24}$$

2.2.3 Sigmoid Commute Time Kernel

The sigmoid commute time kernel $\mathbf{K}_{\mathrm{SCT}}$ is obtained by applying a sigmoid transformation on the commute time kernel $\mathbf{K}_{\mathrm{CT}}$

$$[\mathbf{K}_{\mathrm{SCT}}]_{ij} = \frac{1}{1+\exp\left(\frac{\alpha[\mathbf{K}_{\mathrm{CT}}]_{ij}}{\sigma}\right)} = \frac{1}{1+\exp\left(\frac{\alpha\, l^+_{ij}}{\sigma}\right)} \tag{2.25}$$

where σ is a normalisation factor corresponding to the standard deviation of the elements of $\mathbf{L}^+$ and α is a strictly positive constant value which must be tuned [69, 70].
The kernel resulting of this transformation is not always valid because it is not always positive semi-definite. However, a valid kernel can be computed using the transformation described in Section 2.2.

The idea behind this kernel is to reduce the scale of the elements of the Laplacian matrix, i.e. the l^+_{ij}, which are usually very large by performing a sigmoid transformation. This will constrain the similarities between 0 and 1 and thus reduce their spread. Experimental comparisons have shown that $\mathbf{K}_{\mathrm{SCT}}$ performs better than $\mathbf{K}_{\mathrm{CT}}$ which gives very poor results [64, 70].

The sigmoid transformation used in Equation 2.25 can also be used on the corrected commute time kernel $\mathbf{K}_{\mathrm{CCT}}$ to obtain the sigmoid corrected commute time kernel $\mathbf{K}_{\mathrm{SCCT}}$.

2.2.4 Logarithmic Forest Distance

This new class of distances introduced by Chebotarev is based on the matrix forest theorem and the transition inequality [5, 6, 7, 16].
Chebotarev starts by defining a new positive semi-definite similarity matrix from the Laplacian matrix (see Subsection 2.1.3), called the regularized Laplacian kernel [18],

$$\mathbf{K}_{\mathrm{RL}} = (\mathbf{I}+\alpha\mathbf{L})^{-1}, \text{ with } \alpha > 0 \tag{2.26}$$

A new matrix $\mathbf{H}_\alpha$ is then computed using this kernel

$$\mathbf{H}_\alpha = \begin{cases} (\alpha-1)\log_\alpha(\mathbf{K}_{\mathrm{RL}}) & \text{if } \alpha \neq 1 \\ \ln(\mathbf{K}_{\mathrm{RL}}) & \text{if } \alpha = 1 \end{cases} \tag{2.27}$$

with $\log_\alpha$ the function computing elementwise the logarithm in base α.
The logarithmic forest distance can thus be computed as follows

$$\mathbf{\Delta}_{\mathrm{LF}} = \mathbf{diag}(\mathbf{H}_\alpha)\mathbf{e}^\mathsf{T} + \mathbf{e}(\mathbf{diag}(\mathbf{H}_\alpha))^\mathsf{T} - 2\mathbf{H}_\alpha \tag{2.28}$$

This equation is the one used to obtain a squared distance on a kernel (Equation 2.13). Chebotarev proves in [5] that $\mathbf{\Delta}_{\mathrm{LF}}$ is a metric.
When α converges to 0^+, the logarithmic forest distance tends to be the shortest-path distance and when α converges to ∞, it tends to the commute time distance up to a scaling factor.

The following chapter will present additional distance measures to those presented in this chapter by introducing the bag-of-paths framework and some well-defined distances derived from it. The new investigated Poisson surprisal distance will also be presented.

3 Bag-of-Paths Framework

This chapter will introduce the bag-of-paths, or BoP, and the bag-of-hitting-paths, or BoHP, frameworks as well as some distance matrices based on these frameworks. Some of these distance matrices will be used as reference distances, such as the distances described in the previous chapter, to evaluate the performance of the new distance matrix based on the BoP that is also defined in this chapter. The new distance is called the *Poisson surprisal distance*. The *Poisson weighted surprisal distance* also described in this chapter has already been evaluated in semi-supervised classification, this thesis aims to evaluate it in clustering.

3.1 Framework Description

The BoP framework [16, 20, 68] is based upon the probability of drawing a path $\wp$ starting at node i and ending a node j from the bag of paths $\mathcal{P}$, i.e. the set of all possible paths in the graph G. A path $\wp$ can be considered as a finite sequence of jumps between adjacent nodes in G starting at a node i, ending at a node j and where loops are allowed. The total cost of a path is denoted $\tilde{c}(\wp)$ and can be computed, using the cost matrix $\mathbf{C}$ of G (see Subsection 2.1.2), as the sum of the cost of each edge along that path. The transition probability matrix of the natural random walk on the graph G will be denoted $\mathbf{P}$ (see Subsection 2.1.4).

The probability of drawing a path $\wp$ by walking according to the probability p_{ij} of the natural random walk on the graph can be written $\mathrm{P}^{\mathrm{ref}}(\wp)$ and is proportional to the product of the p_{ij} along the edges of the path $\wp$. The probability of drawing a path $\wp$ from $\mathcal{P}$ can be seen as a Gibbs-Boltzmann probability distribution

$$\mathrm{P}(\wp) = \frac{\mathrm{P}^{\mathrm{ref}}(\wp)\exp(-\theta\tilde{c}(\wp))}{\sum_{\wp'\in\mathcal{P}} \mathrm{P}^{\mathrm{ref}}(\wp')\exp(-\theta\tilde{c}(\wp'))} \tag{3.1}$$

where θ is the strictly positive inverse temperature parameter. Low-cost paths will have a higher probability of being drawn from $\mathcal{P}$ than high-cost paths. The probability also captures a notion of accessibility, two nodes will be strongly related if they are connected by many low-cost paths.

The BoP probability can be computed in closed form. The first step is to define the matrix $\mathbf{W}$ from $\mathbf{C}$, $\mathbf{P}$ and the inverse temperature parameter such that

$$\mathbf{W} = \mathbf{P} \circ \exp(-\theta\mathbf{C}) \tag{3.2}$$

where $\circ$ is the elementwise (Hadamard) matrix product and the exponential is also taken elementwise. This matrix is generally not symmetric and is substochastic, which means that the sum of each row is smaller than 1.

The fundamental matrix $\mathbf{Z}$ can then be computed as follows

$$\mathbf{Z} \triangleq \sum_{t=0}^{\infty} \mathbf{W}^t = (\mathbf{I} - \mathbf{W})^{-1} \tag{3.3}$$

From this matrix $\mathbf{Z}$ the BoP probability of drawing a path of arbitrary length starting at node i and ending at node j can be computed as

$$\mathrm{P}(s = i, e = j) = \frac{[\mathbf{Z}]_{ij}}{\sum_{i',j'=1}^{n} [\mathbf{Z}]_{i'j'}} = \frac{z_{ij}}{z_{\bullet\bullet}} \tag{3.4}$$

or in matrix form, the BoP probability matrix $\mathbf{\Pi}$,

$$\mathbf{\Pi} = \frac{\mathbf{Z}}{z_{\bullet\bullet}} \tag{3.5}$$

This matrix $\mathbf{\Pi}$ is generally not symmetric.

The BoHP framework [16, 20] is a restriction of the BoP framework where the set of paths considered is restricted to paths that starts from a node i and stops the first time they reach the node j, i.e. node j is made absorbing and appears only once in the path. An absorbing node is a node for which the probability of leaving it is 0. Considering the same reasoning as for the BoP probability matrix, but where the matrix $\mathbf{W}_\mathrm{h}$ is the matrix $\mathbf{W}$ with the j^{th} row set to $\mathbf{0}^\mathsf{T}$, the BoHP probability matrix can be defined as follows

$$\mathbf{\Pi}_\mathrm{h} = \frac{\mathbf{Z}_\mathrm{h}}{\mathbf{e}\mathbf{Z}_\mathrm{h}\mathbf{e}^\mathsf{T}}, \text{ with } \mathbf{Z}_\mathrm{h} = (\mathbf{I} - \mathbf{W}_\mathrm{h})^{-1} = \mathbf{Z}\mathbf{D}_\mathrm{h}^{-1} \text{ and } \mathbf{D}_\mathrm{h} = \mathbf{Diag}(\mathbf{Z}) \tag{3.6}$$

where $\mathbf{Diag}(\mathbf{Z})$ is a matrix which contains the diagonal of $\mathbf{Z}$ on its diagonal and 0 elsewhere. The BoHP probability of drawing a path of arbitrary length from node i to node j is therefore

$$\mathrm{P}_\mathrm{h}(s = i, e = j) = \frac{z^\mathrm{h}_{ij}}{z^\mathrm{h}_{\bullet\bullet}}, \text{ with } z^\mathrm{h}_{ij} = \frac{z_{ij}}{z_{jj}} \tag{3.7}$$

3.2 Bag-of-Paths Based Distances Between Nodes

3.2.1 Free Energy Distance

The free energy distance [16, 20, 32] is based on the BoHP probability. It can be defined as

$$[\Delta^\Phi_\mathrm{FE}]_{ij} = \begin{cases} \Phi(i,j) + \Phi(j,i) & \text{if } i \neq j \\ 0 & \text{if } i = j \end{cases}, \text{ where } \Phi(i,j) = -\tfrac{1}{\theta}\log(z^\mathrm{h}_{ij}) = -\tfrac{1}{\theta}\log(\tfrac{z_{ij}}{z_{jj}}) \tag{3.8}$$

The metric $\Phi(i,j)$ is called the directed free energy potential of node i with respect to node j.

The free energy distance is a valid metric. When θ tends to ∞, the free energy distance converges to the shortest-path distance and when θ tends to 0^+, it converges to half the commute time distance.

3.2.2 Surprisal Distance

The surprisal distance [16, 20] is a distance that is also derived from the BoHP framework and aims to quantify the "surprise" of starting at a node i and ending at a node j, it can be computed as follows

$$[\Delta_\mathrm{SURP}]_{ij} = \begin{cases} -\frac{1}{2}\left(\log\left(\frac{z^\mathrm{h}_{ij}}{z^\mathrm{h}_{\bullet\bullet}}\right) + \log\left(\frac{z^\mathrm{h}_{ji}}{z^\mathrm{h}_{\bullet\bullet}}\right)\right) & \text{if } i \neq j \\ 0 & \text{if } i = j \end{cases} \tag{3.9}$$

The surprisal distance $\mathbf{\Delta}_\mathrm{SURP}$ is symmetrical, non-negative, reflexive and respects the triangle inequality. It is consequently a distance measure.

3.2.3 Randomized Shortest Path Dissimilarity

The randomized shortest-path dissimilarity [16, 32, 59, 68] is a dissimilarity interpolating between the shortest-path distance and half the commute time distance based on a pure random walk for which the "random" strategy favours the lowest-cost paths. The randomized shortest-path probability can be interpreted as the a posteriori probability of drawing a path $\wp$ given that the starting node i and the ending node j are known [12].

First the matrix $\mathbf{S}$ will be defined as

$$\mathbf{S} = (\mathbf{Z}(\mathbf{C} \circ \mathbf{W})\mathbf{Z}) \div \mathbf{Z} \tag{3.10}$$

where $\circ$ is the elementwise multiplication and $\div$ the elementwise division. $\mathbf{C}$ is the cost matrix (see Subsection 2.1.2) and the matrix $\mathbf{W}$ as well as the fundamental matrix $\mathbf{Z}$ are described in the Section 3.1. This matrix $\mathbf{S}$ contains the expected cost over the non-hitting paths [32].

The randomized shortest-path cost $\langle \tilde{c} \rangle_{ij}$ is the expected cost on all hitting paths that connects nodes i and j. It can be defined as

$$\langle \tilde{c} \rangle_{ij} = [\mathbf{S}]_{ij} - [\mathbf{S}]_{jj} = [\mathbf{S} - \mathbf{e}(\mathbf{Diag}(\mathbf{S}))^{\mathsf{T}}]_{ij} \tag{3.11}$$

From the randomized shortest-path cost can be computed the randomized shortest-path dissimilarity

$$[\Delta_{RSP}]_{ij} = \begin{cases} \frac{\langle \tilde{c} \rangle_{ij} + \langle \tilde{c} \rangle_{ji}}{2} & \text{if } i \neq j \\ 0 & \text{if } i = j \end{cases} \tag{3.12}$$

The randomized shortest-path dissimilarity matrix $\mathbf{\Delta}_{RSP}$ does not respect the triangle inequality for $0 < \theta < \infty$ which means that it is not a distance measure, only a semi-metric. However, it still provides good results when used to find clusters with a kernel k-means algorithm (described in Subsection 5.1.2). Like the free energy, when θ tends to ∞, the randomized shortest-path dissimilarity converges to the shortest-path distance and when θ tends to 0^+, it converges to half the commute time distance [16, 68].

3.2.4 Poisson Weighted Surprisal Distance

The Poisson weighted surprisal distance is based on the BoP framework where a probability mass weighting the length of the paths by a Poisson probability distribution has been introduced [12]. This distance assumes that the path length and the path likelihood are independent, which leads to a similarity measure close to the log-communicability one [27]. This measure aims to explicitly take into account the length of a path when expressing its probability and thus to penalize the longer paths more easily.

For computing this distance, the length $l(\wp)$ of a path $\wp$ is weighted according to a Poisson distribution so that its probability modulates the likelihood of this path. The probability of drawing a path $\wp$ with a given length τ (i.e. $l(\wp) = \tau$) will be proportional to

$$\mathrm{P}(\wp) \propto f(\tau, \lambda)\, w(\wp), \text{ with } w(\wp) = \mathrm{P}^{\mathrm{ref}}(\wp) \exp(-\theta \tilde{c}(\wp)) \tag{3.13}$$

where $f(\tau, \lambda) = -\lambda^{\tau} \frac{\exp(-\lambda)}{\tau\,!}$ is the Poisson distribution which provides the probability mass of drawing a path of length τ with the parameter λ. $\mathrm{P}^{\mathrm{ref}}(\wp)$ and $\tilde{c}(\wp)$ are introduced in Section 3.1. The λ parameter is a parameter that must be tuned and that modulates the region of influence in the network based on the path lengths. To ensure the probability is properly distributed (i.e. for its sum to be equal to 1), the expression in Equation 3.13 must be normalized.

The probability of a path starting at node i and ending at node j can therefore be computed in closed form as follows

$$\begin{aligned} \pi_{ij}(\lambda) = \mathrm{P}(s = i, e = j) &\propto \sum_{\wp \in \mathcal{P}} \sum_{\tau=0}^{\infty} \delta(\wp(0) = i)\delta(\wp(l) = j)\delta(l(\wp) = \tau) f(\tau, \lambda)\, w(\wp) \\ &= \sum_{\wp \in \mathcal{P}_{ij}} f(l(\wp), \lambda)\, w(\wp) = \sum_{\tau=0}^{\infty} \sum_{\wp \in \mathcal{P}_{ij}^{\tau}} f(\tau, \lambda)\, w(\wp) \\ &= \sum_{\tau=0}^{\infty} f(\tau, \lambda)\, [\mathbf{W}^{\tau}]_{ij} = \exp(-\lambda) \left[\sum_{\tau=0}^{\infty} \frac{\lambda^{\tau} \mathbf{W}^{\tau}}{\tau!} \right]_{ij} \\ &= [\mathrm{expm}(\lambda \mathbf{W})]_{ij} \end{aligned} \tag{3.14}$$

where $\mathcal{P}_{ij}$ is the restriction on $\mathcal{P}$ of all the paths with a start node i and an end node j, $\mathcal{P}_{ij}^{\tau}$ is a restriction on $\mathcal{P}_{ij}$ where the length of the path is exactly τ and $\mathrm{expm}(.)$ is the matrix exponential.

The probability after having normalized is

$$\pi_{ij}(\lambda) = \frac{[\text{expm}(\lambda \mathbf{W})]_{ij}}{\sum_{i',j'=1}^{n} [\text{expm}(\lambda \mathbf{W})]_{i'j'}} \tag{3.15}$$

or in matrix form

$$\mathbf{\Pi}(\lambda) = \frac{\text{expm}(\lambda \mathbf{W})}{\mathbf{e}^\intercal \text{expm}(\lambda \mathbf{W}) \mathbf{e}} \tag{3.16}$$

This probability matrix takes into account the closeness of nodes i and j and the number of paths connecting them to quantify how easily they are accessible from the other.

To derive a dissimilarity matrix from the probability matrix, opposite the elementwise logarithm will be taken, so the Poisson weighted surprisal dissimilarity will be

$$\mathbf{\Delta} = -\log \mathbf{\Pi}(\lambda) \tag{3.17}$$

The distance of a node to itself will be set to 0 such that

$$\mathbf{\Delta} = \mathbf{\Delta} - \mathbf{Diag}(\mathbf{\Delta}) \tag{3.18}$$

As this matrix is not symmetric, the Poisson weighted surprisal distance can be computed as

$$\mathbf{\Delta}_{\text{PWSURP}} = \frac{\mathbf{\Delta} + \mathbf{\Delta}^\intercal}{2} \tag{3.19}$$

which is closely related to the log-communicability measure $\mathbf{K} = \log(\text{expm}(\alpha \mathbf{A}))$ [27]. The main difference being that the latter is computed on the adjacency matrix $\mathbf{A}$ (see Subsection 2.1.1) and not on the $\mathbf{W}$ one (see Equation 3.2). It can be noticed that Equations 3.17, 3.18 and 3.19 are together equivalent to the Equation 3.9 used to define the Surprisal distance. The algorithm for computing $\mathbf{\Delta}_{\text{PWSURP}}$ is presented in Algorithm 1.

Algorithm 1 Poisson weighted surprisal distance

Input :

- the $n \times n$ adjacency matrix $\mathbf{A}$ associated to the graph G containing the affinities
- the $n \times n$ cost matrix $\mathbf{C}$ associated to the graph G
- the inverse temperature parameter θ
- the parameter λ of the Poisson Distribution

Output :

- the $n \times n$ Poisson weighted surprisal distance matrix $\mathbf{\Delta}_{\text{PWSURP}}$

$\mathbf{D} \leftarrow \mathbf{Diag}(\mathbf{Ae})$ ▷ The row-normalization matrix
$\mathbf{P} \leftarrow \mathbf{D}^{-1}\mathbf{A}$ ▷ The reference transition probability matrix
$\mathbf{W} \leftarrow \mathbf{P} \circ \exp(-\theta \mathbf{C})$
$\mathbf{W}^{\text{exp}} \leftarrow \text{expm}(\lambda \mathbf{W})$
$\mathbf{\Pi} \leftarrow \frac{\mathbf{W}^{\text{exp}}}{\mathbf{e}^\intercal \mathbf{W}^{\text{exp}} \mathbf{e}}$ ▷ The probability matrix
$\mathbf{\Delta} \leftarrow -\log \mathbf{\Pi}$ ▷ The Poisson weighted surprisal directed distance matrix
$\mathbf{\Delta} \leftarrow \mathbf{\Delta} - \mathbf{Diag}(\mathbf{\Delta})$ ▷ Self-distances are set to 0
$\mathbf{\Delta}_{\text{PWSURP}} \leftarrow \frac{\mathbf{\Delta} + \mathbf{\Delta}^\intercal}{2}$ ▷ The Poisson weighted surprisal distance matrix

In general, the assumption that path likelihood and path length are independent is not respected because it is not realistic.

3.2.5 New Investigated Distance: Poisson Surprisal Distance

The Poisson surprisal distance constrains the probability of sampling a path $\wp$ of length $\tau = l(\wp)$ to follow a Poisson probability distribution that will tune the optimal path length at which the relevant information is located [11]. As with the standard BoP framework, the free energy objective function will be minimized. However a new constraint will be added to represent the path length distribution

$$\begin{cases} \underset{\mathrm{P}(\wp)}{\text{minimize}} \sum_{i,j=1}^{n} \sum_{\tau=0}^{\infty} \sum_{\wp \in \mathcal{P}_{ij}^{\tau}} \mathrm{P}(\wp)\tilde{c}(\wp) + T\,\mathrm{P}(\wp)\log(\dfrac{\mathrm{P}(\wp)}{\tilde{\pi}(\wp)}) \\ \text{subject to} \sum_{i,j=1}^{n} \sum_{\wp \in \mathcal{P}_{ij}^{\tau}} \mathrm{P}(\wp) = f(\tau,\lambda) \qquad \text{for each } \tau \end{cases} \tag{3.20}$$

where T is the temperature parameter, $\tilde{\pi}(\wp)$ is the likelihood of the path $\wp$ and $\mathcal{P}_{ij}^{\tau}$ is the set of all paths of length τ starting at node i and ending at node j. The condition on $f(\tau,\lambda)$, the Poisson probability mass with parameter λ, implies that the probability distribution sums to 1 since $\sum_{\tau=0}^{\infty} f(\tau,\lambda) = 1$. The problem presented in 3.20 can be solved by using a Lagrangian formulation that integrates the equality constraints as

$$\begin{aligned} \mathcal{L}(\mathrm{P}(\cdot)) = &\sum_{i,j=1}^{n} \sum_{\tau=0}^{\infty} \sum_{\wp \in \mathcal{P}_{ij}^{\tau}} \mathrm{P}(\wp)\tilde{c}(\wp) + T\,\mathrm{P}(\wp)\log(\frac{\mathrm{P}(\wp)}{\tilde{\pi}(\wp)}) \\ &+ \sum_{\tau=0}^{\infty} \mu_{\tau}\Big(f(\tau,\lambda) - \sum_{i,j=1}^{n} \sum_{\wp \in \mathcal{P}_{ij}^{\tau}} \mathrm{P}(\wp)\Big) \end{aligned} \tag{3.21}$$

with μ_{τ} the Lagrangian multiplier for a particular length τ. By setting the partial derivative with respect to $\mathrm{P}(\wp)$ to 0 for the path length τ, i.e. $\wp \in \mathcal{P}_{ij}^{\tau}$, we get

$$\frac{\partial \mathcal{L}(\mathrm{P}(\cdot))}{\partial \mathrm{P}(\wp)} = \tilde{c}(\wp) + T\log(\frac{\mathrm{P}(\wp)}{\tilde{\pi}(\wp)}) + T - \mu_{\tau} = 0 \;\text{ for } \wp \in \mathcal{P}_{ij}^{\tau} \tag{3.22}$$

The probability for the path $\wp$ to be drawn from $\mathcal{P}$ can therefore be obtained from Equation 3.22 such that

$$\mathrm{P}(\wp) = \tilde{\pi}(\wp)\exp(-\theta\tilde{c}(\wp))\,\exp(\theta\mu_{\tau} - 1) \tag{3.23}$$

where $\theta = \frac{1}{T}$ is the inverse temperature parameter. The path length constraints can be rewritten as follows

$$f(\tau,\lambda) = \sum_{i,j=1}^{n} \sum_{\wp \in \mathcal{P}_{ij}^{\tau}} \tilde{\pi}(\wp)\exp(-\theta\tilde{c}(\wp))\,\exp(\theta\mu_{\tau} - 1) \tag{3.24}$$

leading to

$$\exp(\theta\mu_{\tau} - 1) = \frac{f(\tau,\lambda)}{\sum_{i',j'=1}^{n} \sum_{\wp' \in \mathcal{P}_{i'j'}^{\tau}} \tilde{\pi}(\wp')\exp(-\theta\tilde{c}(\wp'))} \tag{3.25}$$

From this last result, the probability that a particular path $\wp$ starting at node i, ending at node j and having length τ is drawn is therefore

$$\mathrm{P}(\wp) = \frac{f(\tau,\lambda)\tilde{\pi}(\wp)\exp(-\theta\tilde{c}(\wp))}{\sum_{i',j'=1}^{n} \sum_{\wp' \in \mathcal{P}_{i'j'}^{\tau}} \tilde{\pi}(\wp')\exp(-\theta\tilde{c}(\wp'))} \tag{3.26}$$

The probability of drawing a path starting at node i and ending at node j is computed as

$$\pi_{ij} = \mathrm{P}(s=i, e=j) = \sum_{\tau=0}^{\infty} \sum_{\wp \in \mathcal{P}_{ij}^{\tau}} \mathrm{P}(\wp) = \sum_{\tau=0}^{\infty} f(\tau,\lambda) \frac{\sum_{\wp \in \mathcal{P}_{ij}^{\tau}} \tilde{\pi}(\wp)\exp(-\theta\tilde{c}(\wp))}{\sum_{i',j'=1}^{n} \sum_{\wp' \in \mathcal{P}_{i'j'}^{\tau}} \tilde{\pi}(\wp')\exp(-\theta\tilde{c}(\wp'))} \tag{3.27}$$

As for the BoP framework, a fundamental matrix $\mathbf{Z}(\tau)$ can be defined for a given τ

$$z_{ij}(\tau) = [\mathbf{Z}(\tau)]_{ij} = \sum_{\wp \in \mathcal{P}_{ij}^{\tau}} w(\wp) = \sum_{\wp \in \mathcal{P}_{ij}^{\tau}} \tilde{\pi}(\wp) \exp(-\theta \tilde{c}(\wp)) \tag{3.28}$$

This means that π_{ij} can be rewritten as

$$\pi_{ij} = \sum_{\tau=0}^{\infty} f(\tau, \lambda) \frac{z_{ij}(\tau)}{z_{\bullet\bullet}(\tau)} \tag{3.29}$$

To compute each term of the series, a recurrence equation is needed. Using Equation 3.3, it can be defined as

$$\begin{cases} \mathbf{Z}(\tau+1) = \mathbf{W}^{\tau+1} = \mathbf{Z}(\tau)\mathbf{W} \\ \mathbf{\Pi}(\tau+1, \lambda) = \mathbf{\Pi}(\tau, \lambda) + f(\tau+1, \lambda) \dfrac{\mathbf{Z}(\tau+1)}{z_{\bullet\bullet}(\tau+1)} \end{cases} \tag{3.30}$$

The initial condition at $\tau = 0$, when the source and target nodes are the same, is

$$\begin{cases} \mathbf{Z}(0) = \mathbf{I} \\ \mathbf{\Pi}(0, \lambda) = f(0, \lambda) \frac{\mathbf{I}}{n} \end{cases} \tag{3.31}$$

As most spatial interactions in the real world are expected to be local, the algorithm should converge quickly to a stable $\mathbf{Z}$ matrix. This means that the value chosen for λ must be small, but also that $f(\tau, \lambda)$ will quickly fall to 0.

When the algorithm has converged, the Poisson surprisal distance matrix can be computed using $\mathbf{\Pi}(\lambda)$ as for the Poisson weighted surprisal distance.

$$\mathbf{\Delta} = -\log \mathbf{\Pi}(\lambda) \tag{3.32}$$

$$\mathbf{\Delta} = \mathbf{\Delta} - \mathbf{Diag}(\mathbf{\Delta}) \tag{3.33}$$

$$\mathbf{\Delta}_{\mathrm{PSURP}} = \frac{\mathbf{\Delta} + \mathbf{\Delta}^{\mathsf{T}}}{2} \tag{3.34}$$

The algorithm for computing $\mathbf{\Delta}_{\mathrm{PSURP}}$ is shown in Algorithm 2.

Algorithm 2 Poisson surprisal distance

Input :

- the $n \times n$ adjacency matrix $\mathbf{A}$ associated to the graph G containing the affinities
- the $n \times n$ cost matrix $\mathbf{C}$ associated to the graph G
- the inverse temperature parameter θ
- the parameter λ of the Poisson Distribution

Output :

- the $n \times n$ Poisson surprisal distance matrix $\mathbf{\Delta}_{\mathrm{PSURP}}$

$\mathbf{D} \leftarrow \mathbf{Diag}(\mathbf{Ae})$ ▷ The row-normalization matrix
$\mathbf{P} \leftarrow \mathbf{D}^{-1}\mathbf{A}$ ▷ The reference transition probability matrix
$\mathbf{W} \leftarrow \mathbf{P} \circ \exp(-\theta \mathbf{C})$
$\mathbf{Z} \leftarrow \mathbf{I}$ ▷ Initialize the fundamental matrix
$\mathbf{\Pi} \leftarrow f(0, \lambda) \frac{\mathbf{I}}{n}$ ▷ Initialize the probability matrix
$\mathrm{update} \leftarrow \mathbf{I}$ ▷ Needed to evaluate the convergence
$\mathrm{iter} = 1$
while $\mathrm{iter} \leq \lambda$ or $\mathrm{sum}(\mathrm{update}) > 10^{-6}$ **do**
    $\mathbf{Z} \leftarrow \mathbf{ZW}$
    $\mathrm{update} \leftarrow f(\mathrm{iter}, \lambda) \frac{\mathbf{Z}}{\mathbf{e}^{\mathsf{T}}\mathbf{Ze}}$
    $\mathbf{\Pi} \leftarrow \mathbf{\Pi} + \mathrm{update}$

12: $\text{iter} \leftarrow \text{iter} + 1$
13: **end while**
14: $\mathbf{\Delta} \leftarrow -\log \mathbf{\Pi}$ ▷ The Poisson surprisal directed distance matrix
15: $\mathbf{\Delta} \leftarrow \mathbf{\Delta} - \mathbf{Diag}(\mathbf{\Delta})$ ▷ Self-distances are set to 0
16: $\mathbf{\Delta}_{\text{PSURP}} \leftarrow \frac{\mathbf{\Delta}+\mathbf{\Delta}^{\mathsf{T}}}{2}$ ▷ The Poisson surprisal distance matrix

The next chapter will present two techniques that provide an embedding from a distance measure such as the ones presented in this chapter and the previous. It will also present the new embedding technique based on the BoP framework that will be experimented with in this thesis.

4 Graph Embedding Algorithms

This chapter presents two well-defined techniques for computing an embedding on a distance or dissimilarity matrix, such as those presented in the Chapters 2 and 3. It will also present a new technique based on the bag-of-paths framework described in Section 3.1.

4.1 Classical Multidimensional Scaling

The multidimensional scaling, or MDS, [8, 33, 46, 47] is a dimensionality reduction technique whose goal is to construct a configuration in the Euclidean space of a graph's nodes using the distance or dissimilarity matrix of this graph. The low-dimensional configuration constructed should preserve the input distances/dissimilarities or reproduce them as closely as possible. Therefore, the MDS representation aims to minimize the residual sum between the initial and the computed representation so that

$$MDS(\boldsymbol{\phi}_1 ... \boldsymbol{\phi}_n) = \sum_{i,j=1}^{n} \left(\Delta_{ij} - \left\| \boldsymbol{\phi}_i - \boldsymbol{\phi}_j \right\| \right)^2 \tag{4.1}$$

with Δ_{ij} the distance/dissimilarity between the nodes i and j and $\boldsymbol{\phi}_i$ the representation of node i in lower dimensionality.

MDS is a technique that extracts the reduced dimensionality from the eigendecomposition of the distance matrix. Using Equation 2.12, a kernel can be computed on the distance matrix. This kernel has been centered (Equation 2.11) to have a reference point as inner products, unlike distances, are not invariant to translation. On this centered kernel can be computed the eigendecomposition. If the dimensionality of the representation to be computed is m, the m largest eigenvalues λ and their associated eigenvectors $\mathbf{u}$ will be kept. The MDS embedding representation can finally be computed as follows

$$\boldsymbol{\Phi}_m = \mathbf{U}_m \sqrt{\boldsymbol{\lambda}_m} \tag{4.2}$$

where $\mathbf{U}_m$ is the matrix containing the m eigenvectors $\mathbf{u}$ on its columns and $\boldsymbol{\lambda}_m$ is the column vector containing the m eigenvalues λ.

Classical multidimensional scaling is, by design, a greedy algorithm, as each added dimension reduces the error of Equation 4.1 by the maximum possible [33]. It is one of the oldest dimensionality reduction techniques but is still widely used. However, MDS fails to flatten intrinsically low-dimensional curved manifolds, e.g. the "Swiss Roll" a two-dimensional spiralling manifold for which MDS will fail to eliminate the dimensions taken up by the curvature [8].

4.2 t-Distributed Stochastic Neighbor Embedding

The t-Distributed Stochastic Neighbor Embedding[2], or tSNE, [4, 61, 62] is a technique that aims to visualize high-dimensional data in a lower dimensionality, e.g. two or three dimensions. tSNE is able to capture the local structure of the high-dimensional data very well. It can also reveal the global structure of the data, such as the presence of clusters at several scales.

[2]The code used in the experiments can be found at `https://lvdmaaten.github.io/tsne/` (visited on Feb. 10, 2022)

Given that the high-dimensional data can be represented by the vectors $\{\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, ..., \boldsymbol{\xi}_n\}$ and that vector representation in low-dimensional space of each data will be denoted by $\{\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, ..., \boldsymbol{\phi}_n\}$, tSNE can compute two probabilities. The first is the symmetrized conditional probability p_{ij}

$$p_{ij} = \frac{p_{i|j} + p_{j|i}}{2n} \text{ , with } p_{j|i} = \frac{\exp(\frac{-\|\boldsymbol{\xi}_i - \boldsymbol{\xi}_j\|^2}{2\sigma_i^2})}{\sum_{k \neq i} \exp(\frac{-\|\boldsymbol{\xi}_i - \boldsymbol{\xi}_k\|^2}{2\sigma_i^2})} \tag{4.3}$$

where $p_{j|i}$ represents the similarity of the data $\boldsymbol{\xi}_j$ and $\boldsymbol{\xi}_i$ which can be defined as the probability that $\boldsymbol{\xi}_i$ will pick $\boldsymbol{\xi}_j$ as its neighbor if the neighbors were picked in proportion to their probability density under a Gaussian centered on $\boldsymbol{\xi}_i$. The parameter σ_i represents the variance of this Gaussian.

The second probability computed by tSNE is the pairwise similarity in the low-dimensional space q_{ij} which follows a Student t-distribution with one degree of freedom

$$q_{ij} = \frac{(1 + \|\boldsymbol{\phi}_i - \boldsymbol{\phi}_j\|^2)^{-1}}{\sum_{k \neq l} (1 + \|\boldsymbol{\phi}_k - \boldsymbol{\phi}_l\|^2)^{-1}} \tag{4.4}$$

Like the probability p_{ij}, the probability q_{ij} is symmetric so that $q_{ij} = q_{ji} \; \forall \; i, j$. Using a Student t-distribution provides a heavy-tailed distribution that solves the crowding problem of the Stochastic Neighbor Embedding technique, or SNE, on which tSNE is based [62].

The tSNE attempts to find a low-dimensional representation that minimizes the difference between the two defined probabilities. The cost function of tSNE can thus be defined as the Kullback-Leibler divergence between the joint probability distribution in high-dimensional space P and the Student-t based joint probability in low-dimensional space Q

$$\mathbf{KL}(\mathrm{P} \| \mathrm{Q}) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}} \tag{4.5}$$

The gradient associated to this Kullback-Leibler divergence is

$$\frac{\partial \mathbf{KL}(\mathrm{P} \| \mathrm{Q})}{\partial \boldsymbol{\phi}_i} = 4 \sum_j (p_{ij} - q_{ij})(\boldsymbol{\phi}_i - \boldsymbol{\phi}_j)(1 + \|\boldsymbol{\phi}_i - \boldsymbol{\phi}_j\|^2)^{-1} \tag{4.6}$$

The gradient descent that will minimize $\mathbf{KL}(\mathrm{P} \| \mathrm{Q})$ is initiated by randomly sampling low-dimensional datapoints from a Gaussian with a small variance centered around the origin. A momentum term will be added to the gradient to speed up the optimization and avoid local minima. This term should be small in early iterations until the low-dimensional datapoints become moderately well organized. The gradient descent process can also be accelerated by adding a learning rate which will be higher when the gradient is stable.

The variance σ_i of the Gaussian centered at $\boldsymbol{\xi}_i$ is a parameter that must be tuned as a single value is unlikely to be optimal given that the density of data is likely to vary. Any value of σ_i will induce a probability distribution P_i on all the other data, the entropy of this distribution will increase as σ_i increases. To find σ_i, a binary search will be performed to match a pre-specified perplexity value. The perplexity can be interpreted as a smooth measure of the effective number of neighbors, i.e. a higher perplexity induces a higher number of significant neighbors. The perplexity increases monotonically with σ_i. Typical values of perplexity are between 5 and 50 [4, 62].

The tSNE focuses on modelling dissimilar data using large pairwise distances and similar data using small pairwise distances. Unfortunately, this technique has some weaknesses, the most important being due to the heavy tail of the Student t-distribution with one degree of freedom, which leads to a representation in more than three dimensions where the local structure of the data is not well preserved. For the probability Q, a more appropriate choice would be to use a Student t-distribution with $m - 1$ degrees of freedom, with m the number of dimensions of the low-dimensional space [61]. Another drawback of tSNE is that it does not perform well when used on a dataset with high intrinsic dimensionality, as the tSNE reduction is mainly based on local properties.

4.3 New Investigated Embedding: Bag-of-paths Embedding

Bag-of-paths embedding is a technique based on the BoHP framework (described in Section 3.1) which aims at creating an embedding of the nodes of a graph G.
Let's consider that each node i of the graph G is associated to a $m \times 1$ vector $\boldsymbol{\phi}_i$ representing its coordinates in a m-dimensional Euclidean space (i.e. its embedding). The embedding matrix $\boldsymbol{\Phi}$ contains on its rows the n node embeddings $\boldsymbol{\phi}_i^{\mathsf{T}}$.

The idea behind this technique is to solve the following Kullback-Leibler problem

$$\underset{\boldsymbol{\Phi}}{\text{minimize}} \ \mathbf{KL}(\mathrm{P}\|\mathrm{P}_{\boldsymbol{\phi}}) = \sum_{\wp \in \mathcal{P}^{\mathrm{h}}} \mathrm{P}(\wp) \log\left(\frac{\mathrm{P}(\wp)}{\mathrm{P}_{\boldsymbol{\phi}}(\wp)}\right) \tag{4.7}$$

where $\mathcal{P}^{\mathrm{h}}$ is the set of all hitting paths as defined in Section 3.1. $\mathrm{P}(\wp)$, the BoHP probability of sampling a path, and $\mathrm{P}_{\boldsymbol{\phi}}(\wp)$ can be defined as follows

$$\begin{cases} \mathrm{P}(\wp) = \dfrac{\mathrm{P}^{\mathrm{ref}}(\wp)\exp(-\theta\tilde{c}(\wp))}{\sum_{\wp' \in \mathcal{P}^{\mathrm{h}}} \mathrm{P}^{\mathrm{ref}}(\wp')\exp(-\theta\tilde{c}(\wp'))} = \dfrac{\tilde{\pi}(\wp)\exp(-\theta\tilde{c}(\wp))}{\sum_{\wp' \in \mathcal{P}^{\mathrm{h}}} \tilde{\pi}(\wp')\exp(-\theta\tilde{c}(\wp'))} \\ \mathrm{P}_{\boldsymbol{\phi}}(\wp) = \dfrac{\mathrm{P}^{\mathrm{ref}}(\wp)\exp(-\theta\Delta_{\boldsymbol{\phi}}(\wp))}{\sum_{\wp' \in \mathcal{P}^{\mathrm{h}}} \mathrm{P}^{\mathrm{ref}}(\wp')\exp(-\theta\Delta_{\boldsymbol{\phi}}(\wp'))} = \dfrac{\tilde{\pi}(\wp)\exp(-\theta\Delta_{\boldsymbol{\phi}}(\wp))}{\sum_{\wp' \in \mathcal{P}^{\mathrm{h}}} \tilde{\pi}(\wp')\exp(-\theta\Delta_{\boldsymbol{\phi}}(\wp'))} \end{cases} \tag{4.8}$$

with $\mathrm{P}^{\mathrm{ref}}(\wp)$ and $\tilde{c}(\wp)$ defined in Section 3.1. Intuitively, in order to minimize the Kullback-Leibler divergence, the bag-of-paths embedding seeks for the nodes embedding representation that best preserves the BoHP probabilities. This means that the Euclidean distances $\Delta_{\boldsymbol{\phi}}(\wp)$ must define the best configuration of nodes such that the BoHP probabilities are best preserved when the distances replace the cost of the paths connecting two nodes. Consequently, there are two possible choices for the distances $\Delta_{\boldsymbol{\phi}}(\wp)$

$$\begin{cases} \Delta_{\boldsymbol{\phi}}(\wp_{ij}) = \Delta_{ij}^{\boldsymbol{\phi}} = \|\boldsymbol{\phi}_i - \boldsymbol{\phi}_j\| \\ \Delta_{\boldsymbol{\phi}}(\wp_{ij}) = \Delta_{ij}^{\boldsymbol{\phi}} = \frac{1}{2}\|\boldsymbol{\phi}_i - \boldsymbol{\phi}_j\|^2 \end{cases} \quad \text{or,} \tag{4.9}$$

Now, instead of directly computing the embeddings $\{\boldsymbol{\phi}_i\}_{i=1}^n$ of the nodes, let's compute a two steps approximation of the Problem 4.7. The first step consists in deriving the optimal distances between the pairs of nodes and the second step consists in deriving, from these distances, the corresponding embedding through multidimensional scaling (see Section 4.1).
The first problem to solve is therefore

$$\underset{\boldsymbol{\Delta}_{\boldsymbol{\phi}}}{\text{minimize}} \ \mathbf{KL}(\mathrm{P}\|\mathrm{P}_{\boldsymbol{\phi}}) = \sum_{\wp \in \mathcal{P}^{\mathrm{h}}} \mathrm{P}(\wp) \log\left(\frac{\mathrm{P}(\wp)}{\mathrm{P}_{\boldsymbol{\phi}}(\wp)}\right) \tag{4.10}$$

We will use the two following equations $\sum_{\wp \in \mathcal{P}_{ij}^{\mathrm{h}}} \tilde{\pi}(\wp)\exp(-\theta\tilde{c}(\wp)) = z_{ij}^{\mathrm{h}}$ and $\sum_{\wp \in \mathcal{P}_{ij}^{\mathrm{h}}} \tilde{\pi}(\wp) = 1$ where z_{ij}^{h} belongs to the fundamental matrix $\mathbf{Z}_{\mathrm{h}}$ of the BoHP framework (see Equation 3.6). From these equations, inserting Equations 4.8 in Problem 4.10 gives

$$\begin{aligned} \mathbf{KL}(\mathrm{P}\|\mathrm{P}_{\boldsymbol{\phi}}) &= \sum_{\wp \in \mathcal{P}^{\mathrm{h}}} \mathrm{P}(\wp) \log\exp[-\theta(\tilde{c}(\wp) - \Delta_{\boldsymbol{\phi}}(\wp))] - \log \sum_{\wp \in \mathcal{P}^{\mathrm{h}}} \tilde{\pi}(\wp)\exp(-\theta\tilde{c}(\wp)) \\ &\quad + \log \sum_{\wp \in \mathcal{P}^{\mathrm{h}}} \tilde{\pi}(\wp)\exp(-\theta\Delta_{\boldsymbol{\phi}}(\wp)) \\ &= \theta\left(\sum_{i,j=1}^{n} \frac{z_{ij}^{\mathrm{h}}}{z_{\bullet\bullet}^{\mathrm{h}}} \Delta_{ij}^{\boldsymbol{\phi}} - \langle\tilde{c}(\wp)\rangle\right) - \log z_{\bullet\bullet}^{\mathrm{h}} + \log \sum_{i,j=1}^{n} \exp(-\theta\Delta_{ij}^{\boldsymbol{\phi}}) \end{aligned} \tag{4.11}$$

Taking the partial derivative with respect to $\Delta_{ij}^{\boldsymbol{\phi}}$ given that $i \neq j$ and the distance matrix $\mathbf{\Delta}_{\boldsymbol{\phi}}$ is symmetric, we obtain the equation

$$\begin{aligned}\frac{\partial \mathbf{KL}(\mathrm{P}\|\mathrm{P}_{\boldsymbol{\phi}})}{\partial \Delta_{ij}^{\boldsymbol{\phi}}} &= \theta\left(\frac{z_{ij}^{\mathrm{h}} + z_{ji}^{\mathrm{h}}}{z_{\bullet\bullet}^{\mathrm{h}}}\right) - \theta\left(\frac{\exp(-\theta\Delta_{ij}^{\boldsymbol{\phi}}) + \exp(-\theta\Delta_{ji}^{\boldsymbol{\phi}})}{\sum_{k,l=1}^{n}\exp(-\theta\Delta_{kl}^{\boldsymbol{\phi}})}\right) \\ &= 2\theta\left(\left(\frac{z_{ij}^{\mathrm{h}} + z_{ji}^{\mathrm{h}}}{2z_{\bullet\bullet}^{\mathrm{h}}}\right) - \frac{\exp(-\theta\Delta_{ij}^{\boldsymbol{\phi}})}{\sum_{k,l=1}^{n}\exp(-\theta\Delta_{kl}^{\boldsymbol{\phi}})}\right)\end{aligned} \tag{4.12}$$

Setting the partial derivative to 0 gives as optimal distance matrix [3]

$$\begin{cases}\Delta_{ij}^{\boldsymbol{\phi}} = -\frac{1}{\theta}\log\left(\frac{z_{ij}^{\mathrm{h}}+z_{ji}^{\mathrm{h}}}{2}\right) & \text{if } i \neq j \\ \Delta_{ij}^{\boldsymbol{\phi}} = 0 & \text{if } i = j\end{cases} \tag{4.13}$$

which is closely related to the Free Energy distance defined in Equation 3.8. Classical multidimensional scaling can now be used on this distance matrix to find an initial embedding. To solve the Problem 4.7, a gradient descent algorithm in terms of the embedding vectors $\boldsymbol{\phi}_i$ will be applied to this initial embedding

$$\frac{\partial \mathbf{KL}(\mathrm{P}\|\mathrm{P}_{\boldsymbol{\phi}})}{\partial \boldsymbol{\phi}_i} = \sum_{\substack{j=1\\ j\neq i}}^{n}\frac{\partial \mathbf{KL}(\mathrm{P}\|\mathrm{P}_{\boldsymbol{\phi}})}{\partial \Delta_{ij}^{\boldsymbol{\phi}}}\frac{\partial \Delta_{ij}^{\boldsymbol{\phi}}}{\partial \boldsymbol{\phi}_i} = 2\theta\sum_{\substack{j=1\\ j\neq i}}^{n}\left(\left(\frac{z_{ij}^{\mathrm{h}} + z_{ji}^{\mathrm{h}}}{2z_{\bullet\bullet}^{\mathrm{h}}}\right) - \frac{\exp(-\theta\Delta_{ij}^{\boldsymbol{\phi}})}{\sum_{k,l=1}^{n}\exp(-\theta\Delta_{kl}^{\boldsymbol{\phi}})}\right)\frac{\partial \Delta_{ij}^{\boldsymbol{\phi}}}{\partial \boldsymbol{\phi}_i} \tag{4.14}$$

where we use the fact that only the $\Delta_{ij}^{\boldsymbol{\phi}}$ with $i < j$ are independent and $\frac{\partial \Delta_{ij}^{\boldsymbol{\phi}}}{\partial \boldsymbol{\phi}_i}$ can be computed as

$$\begin{cases}\frac{\partial \Delta_{ij}^{\boldsymbol{\phi}}}{\partial \boldsymbol{\phi}_i} = \frac{(\boldsymbol{\phi}_i - \boldsymbol{\phi}_j)}{\|\boldsymbol{\phi}_i - \boldsymbol{\phi}_j\|} & \text{or,} \\ \frac{\partial \Delta_{ij}^{\boldsymbol{\phi}}}{\partial \boldsymbol{\phi}_i} = (\boldsymbol{\phi}_i - \boldsymbol{\phi}_j)\end{cases} \tag{4.15}$$

depending of the choice made at Equation 4.9.

Chapter 5 will present three techniques for measuring how "good" are the embeddings produced by the three embedding methods proposed in this chapter.

---

[3] This can be verified by injecting the Result 4.13 in Equation 4.12.

5 Evaluating Embeddings

The purpose of this chapter is to describe three techniques that can be used to evaluate how well an embedding will represent the graph from which it was extracted. Each of these techniques will evaluate a different criterion, the first evaluates the community preservation, the second is based on neighborhood preservation and the last one evaluates the edge preservation and edge densities.

5.1 Clustering

Clustering, or community detection, [14, 15, 41] is a process that aims to find communities in a graph G. A community can be described as a subgraph where the probability for two nodes to be connected when the nodes belong to this subgraph is higher than when one of the nodes does not. This means that nodes within a community interact more strongly with each other than with the nodes of other communities. Community detection provides insight into the organisation of the graph because nodes in the same cluster share a high similarity.

In this thesis, we will only consider clustering where a node can belong to one community. This means that there can be no overlap of nodes between two communities.

A clustering technique can be evaluated by examining its ability to predict clusters in a graph with a known community structure. Many clustering techniques exist, two will be used as part of this thesis and explained in this chapter. The first is the standard k-means algorithm which will be computed on an embedding of the nodes of the graph and the second is the kernel k-means algorithm performed on the kernel representing the graph.

5.1.1 Standard K-means

The standard k-means algorithm [16, 41, 45] is a two-stage iterative process whose goal is to partition the nodes of a graph into clusters such that this partition minimizes the total sum of squared distances within the cluster. This measure will represent the quality of the partition produced by the standard k-means algorithm.

Each node in the graph will be represented in space by its embedding vector and each cluster $\mathcal{C}_k$ of the partition will be represented by a prototype, or centroid, $\mathbf{q}_k$. This centroid is the most typical node in the cluster, i.e. the node whose sum of distances to all other nodes in the cluster is the smallest. This prototype can be a real node of the graph or an artificial one used only for convergence purposes. The distance between the embedding $\boldsymbol{\phi}_i$ of dimension m of a node i and the prototype $\mathbf{q}_k$ of a cluster $\mathcal{C}_k$ can be computed as the Euclidean distance

$$\Delta^2_{\boldsymbol{\phi}_i \mathbf{q}_k} = \sum_{l=1}^{m} (\boldsymbol{\phi}_i^l - \mathbf{q}_k^l)^2 \tag{5.1}$$

The standard k-means algorithm has two major drawbacks. The first is that the number of clusters K of the partition must be known beforehand. The second is that the algorithm is a local search, which means that it may find locally optimal solutions depending on the initialisation of the prototypes, but it will generally not converge to the optimal partition.

The total sum of distances within a cluster can be expressed as the sum over all nodes of the distance between that node and the prototype of the cluster to which it has been assigned, as follows

$$J = \sum_{k=1}^{K} J_k \text{ where } J_k = \sum_{\boldsymbol{\phi}_i \in \mathcal{C}_k} \Delta^2_{\boldsymbol{\phi}_i \mathbf{q}_k} \tag{5.2}$$

J_k represents the sum of distances within the cluster for the cluster $\mathcal{C}_k$ and quantifies the compactness of the cluster. The smaller J_k is, the better.

The membership of a node in a cluster can be represented by the binary membership value u_{ik}

$$u_{ik} = \begin{cases} 1 & \text{if } \boldsymbol{\phi}_i \in \mathcal{C}_k \\ 0 & \text{otherwise} \end{cases} \tag{5.3}$$

The Equation 5.2 can be redefined according to these new values as

$$J = \sum_{i=1}^{n} \sum_{k=1}^{K} u_{ik} \Delta^2_{\boldsymbol{\phi}_i \mathbf{q}_k} \tag{5.4}$$

The iterative application of the following two steps on a given initialisation will decrease J until it converges to a local minima. A simple initialisation consists in choosing K nodes among all others to be the initial prototypes of the clusters. The two steps are:

- **Allocation step:** For each node, its distance to all the prototypes will be computed and the node will be assigned to the cluster for which the distance to the prototype is minimal

$$k_i^* = \underset{k \in \{1 \dots K\}}{\arg\min} \Delta^2_{\boldsymbol{\phi}_i \mathbf{q}_k} \tag{5.5}$$

- **Computation of the prototypes step:** For each cluster, the prototype will be (re-)computed as follows

$$\mathbf{q}_k = \underset{\boldsymbol{\phi}_i \in C_k}{\arg\min} \{ \sum_{\boldsymbol{\phi}_i \in C_k} \Delta^2_{\boldsymbol{\phi}_i \mathbf{q}_k} \} \tag{5.6}$$

5.1.2 Kernel K-means

The kernel k-means algorithm, based on the standard k-means algorithm, [16, 70] is computed on the valid kernel matrix $\mathbf{K}$ containing the similarities between the nodes to find the partition of the graph. The prototype of each cluster is redefined in the sample vector space, i.e. the Euclidean space having as dimension the number of nodes n of the graph.

The algorithm can be described as a four steps process. As for the standard k-means algorithm, the number of clusters K must be fixed a priori.

- **Compute a Kernel:** The first step is to compute a valid kernel for the algorithm to converge.

- **Define the criterion:** The total within-cluster inertia on the cluster prototype vectors in the embedding space can be used as a criterion to be minimized. This measure requires the number of clusters to be fixed as it always decreases as the number of clusters increases. Defining $\mathbf{q}_k$ as the prototype vector in the embedding space of the cluster $\mathcal{C}_k$ and $\boldsymbol{\phi}_i$ as the representation in the embedding space of the node i, the total within-cluster inertia is

$$J(\mathbf{q}_1, ..., \mathbf{q}_n) = \sum_{k=1}^{K} \sum_{\boldsymbol{\phi}_i \in \mathcal{C}_k} \|\boldsymbol{\phi}_i - \mathbf{q}_k\|^2 \tag{5.7}$$

  where $\|\boldsymbol{\phi}_i - \mathbf{q}_k\|$ is the Euclidean distance between $\boldsymbol{\phi}_i$ and $\mathbf{q}_k$.

- **Translate the prototype vectors:** This step aims to define the prototype vectors in the sample space from the prototype vectors in the embedding space and to express the within-cluster inertia in terms of the kernel matrix and the prototype vectors in sample space. The prototype vector in sample space of the cluster $\mathcal{C}_k$ is denoted $\mathbf{h}_k$, the transformation uses the so-called kernel trick [57]

$$\mathbf{q}_k = \mathbf{\Phi}^\mathsf{T}\mathbf{h}_k \tag{5.8}$$

where $\mathbf{\Phi}$ is the $n \times m$ data matrix containing the transposed node vectors on its rows and m is the number of features but also the dimensionality of the embedding space. Equation 5.7 can be rewritten as follows

$$J(\mathbf{h}_1, ..., \mathbf{h}_n) = \sum_{k=1}^{K} \sum_{i \in \mathcal{C}_k} (\mathbf{e}_i - \mathbf{h}_k)^\mathsf{T}\mathbf{K}(\mathbf{e}_i - \mathbf{h}_k) \tag{5.9}$$

where $\mathbf{e}_i$ is a column vector of size n full of 0's except on row i where it contains a 1. By using the binary membership values defined in Equation 5.3, the total within-cluster inertia can be expressed as

$$J(\mathbf{h}_1, ..., \mathbf{h}_n) = \sum_{i=1}^{n} \sum_{k=1}^{K} u_{ik}(\mathbf{h}_k - \mathbf{e}_i)^\mathsf{T}\mathbf{K}(\mathbf{h}_k - \mathbf{e}_i) \tag{5.10}$$

- **Optimize criterion:** This last step optimises the criterion J with respect to the prototype vectors $\mathbf{h}_k$ and the member values u_{ik} by iterating two steps until convergence. These steps are the same than for the standard k-means algorithm. First, the allocation step where each node is assigned to the cluster whose prototype is closest

$$k_i^* = \underset{k \in \{1...K\}}{\arg\min} \left[(\mathbf{h}_k - \mathbf{e}_i)^\mathsf{T}\mathbf{K}(\mathbf{h}_k - \mathbf{e}_i)\right] \tag{5.11}$$

Second, the computation of the prototypes step where each cluster prototype $\mathbf{h}_k$ is recomputed on the nodes that have been assigned to the cluster k. In order to recompute them, the gradient of J with respect to $\mathbf{h}_k$ is computed and the result is set to 0 in order to have

$$\mathbf{K}\mathbf{h}_k = \mathbf{K}\frac{1}{n_k}\sum_{i \in \mathcal{C}_k} \mathbf{e}_i \tag{5.12}$$

where n_k is the number of nodes in the cluster $\mathcal{C}_k$. One particular solution can be expressed as

$$\mathbf{h}_k = \frac{1}{n_k}\sum_{i \in \mathcal{C}_k} \mathbf{e}_i \tag{5.13}$$

5.2 Dimensionality Reduction Quality: Rank-based Criteria

Dimensionality reduction, or DR, is a term representing techniques that provide a meaningful low-dimensional representation of high-dimensional data. It is an essential tool for the visualisation of such high-dimensional data [8]. In [36, 37, 38], a method for evaluating embeddings based on preserving the κ-ary neighborhood of each node from the high-dimensional space to the low-dimensional one has been introduced by J. Lee and M. Verleysen[4].

Using dimensionality reduction, the n high-dimensionality vectors $[\boldsymbol{\xi}_i]_{1 \leq i \leq n}$ can be represented by the n low-dimensionality vectors $[\boldsymbol{\phi}_i]_{1 \leq i \leq n}$. In high-dimensionality, the distance between the vector $\boldsymbol{\xi}_i$ representing node i and the vector $\boldsymbol{\xi}_j$ representing node j can be denoted δ_{ij}. The distance between the vector $\boldsymbol{\phi}_i$ and the vector $\boldsymbol{\phi}_j$ in low-dimensionality can be denoted Δ_{ij}. It is assumed that these two distances must be symmetrical so that $\delta_{ij} = \delta_{ji}$ and $\Delta_{ij} = \Delta_{ji}$. From these distances the ranks between node i and node j can be computed, they are denoted ρ_{ij} in high-dimensionality and r_{ij} in low-dimensionality

$$\rho_{ij} = |\{k : \delta_{ik} < \delta_{ij} \text{ or } (\delta_{ik} = \delta_{ij} \text{ and } 1 \leq k < j \leq n)\}| \tag{5.14}$$

[4]The code used in the experiments is written by C. de Bodt and comes from the Master thesis of A. M. Safi [54]

$$r_{ij} = |\{k : \Delta_{ik} < \Delta_{ij} \text{ or } (\Delta_{ik} = \Delta_{ij} \text{ and } 1 \leq k < j \leq n)\}| \tag{5.15}$$

with $\rho_{ii} = r_{ii} = 0$ and $|\cdot|$ denoting the set cardinality. These ranks are defined to be unique and, consequently, the non-reflexive ranks belong to the set $\{1...n-1\}$. The non-reflexive κ-ary neighborhood of $\boldsymbol{\xi}_i$ and $\boldsymbol{\phi}_i$ can be respectively noted ν_i^κ and η_i^κ and defined as follows

$$\nu_i^\kappa = \{j : 1 \leq \rho_{ij} \leq \kappa\} \text{ and } \eta_i^\kappa = \{j : 1 \leq r_{ij} \leq \kappa\} \tag{5.16}$$

From these values, it is possible to compute an overall quality criterion $Q_{NX}(\kappa)$, called the Co-ranking score,

$$Q_{NX}(\kappa) = \frac{1}{\kappa n} \sum_{i=1}^{n} |\nu_i^\kappa \cap \eta_i^\kappa| \tag{5.17}$$

It can be considered as the average κ-ary neighborhood agreement, i.e. the weighted sum over all nodes of all neighbors that are in both κ-ary neighborhoods of a node i. Consequently, $Q_{NX}(\kappa)$ aims at measuring how well the neighborhoods have been preserved during the data transformation and lies in the interval $[0, 1]$. The higher the Co-ranking score, the better. A more suitable criterion of overall quality $R_{NX}(\kappa)$ can be computed as the relative improvement of the embedding compared to a random one

$$R_{NX}(\kappa) = \frac{(n-1)Q_{NX}(\kappa) - \kappa}{n - \kappa - 1} \tag{5.18}$$

This score stands in the interval $[-1, 1]$, but a value less than 0 indicates that the embedding evaluated is worse than a random one. This means that the useful range is the same as for the Co-ranking score. An advantage of $R_{NX}(\kappa)$ over $Q_{NX}(\kappa)$ is that the expectation of the former is always 0, while the expectation of the latter is $\frac{\kappa}{n-1}$ which increases with κ.

To give higher priority to the preservation of local properties of a graph, local neighborhoods must be considered more important than larger ones. This means that the graph representing $Q_{NX}(\kappa)$ or $R_{NX}(\kappa)$ as a function of κ should highlight the left-hand side of the curve using a logarithmic scale on the κ axis as shown in Figure 5.1. The higher the left-hand side of the curve, the better the embedding. The figure illustrates that the tSNE embeddings perform better in the local neighborhoods because their curve is higher in the left part of the graph than the curve of the MDS embeddings.

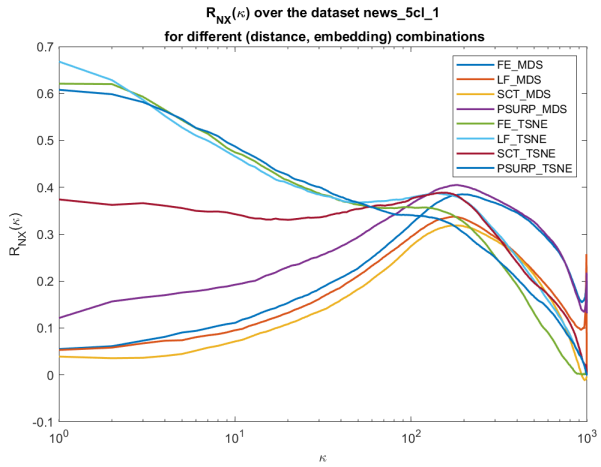

Fig. 5.1.: Illustration of the relative improvement $R_{NX}(\kappa)$ with a semi-logarithmic scale

However, the scores $Q_{NX}(\kappa)$ and $R_{NX}(\kappa)$ have a drawback when searching for the best embedding among several because they depend on the number of neighbors κ considered. To assign a unique score to each embedding, the Area Under the Curve $R_{NX}(\kappa)$ plot with a logarithmic scale on κ, or AUC, can be computed as follows

$$AUC = \frac{\sum_{\kappa=1}^{n-2} \frac{R_{NX}(\kappa)}{\kappa}}{\sum_{\kappa=1}^{n-2} \frac{1}{\kappa}} \tag{5.19}$$

This scalar score ranging from -1 to 1 summarizes the dimensionality reduction quality at all scales with an emphasis on the smallest. As the objective is to preserve the κ-ary neighborhoods, the higher the AUC, the better the embedding.

5.3 Combined Divergence Score

Proposed by B. Kamiński, P. Prałat and F. Théberge in [28, 29, 30][5], this technique aims to evaluate an embedding by assigning two scores to it, one that evaluates the global properties of the graph and one that evaluates the local properties.

The global divergence score evaluates the ability of an embedding to capture edge densities within and between the communities from the coordinates of the nodes in the embedding space. To compute this ability, the actual edge densities will be compared to the probabilities computed on the Geometric Chung-Lu model [29]. It will use the property that, if the distance between two nodes is high, their probability of being connected by an edge should be low. This score allows for identifying embeddings that will perform well in tasks that require global knowledge of the graph, e.g. community detection and node classification.

On the other hand, the local divergence score evaluates the ability of an embedding to capture the presence of edges between pairs of nodes. This means that it evaluates the ability of an embedding to predict a link between two nodes. The probability that two nodes are linked is assumed to be monotonically related to their distance in the embedding and their degrees.

As explained before, the Geometric Chung-Lu model, or GCL, will be used to compute the global score. The GCL is based upon the Chung-Lu model [9], which is itself a generalisation of the Erdös-Rényi random graph. The Chung-Lu model is a random graph where the expected degree distribution is given and must be preserved. The expected number of edges between nodes i and j is computed as the probability

$$\mathrm{P}(i,j) = \begin{cases} \frac{d_i d_j}{2m} & \text{if } i \neq j \\ \frac{d_i^2}{4m} & \text{if } i = j \end{cases} \tag{5.20}$$

with m the number of edges in the graph G and d_i the degree of the node i as defined in Equation 2.3.

The GCL model aims to incorporate the embedding into the Chung-Lu model so that the probability of two nodes being adjacent is also influenced by their distance in the embedding. The closer they are, the more likely they are to be linked. From the embedding matrix $\mathbf{\Phi}$, the distance Δ_{ij} between the embedding representation $\boldsymbol{\phi}_i$ of node i and the embedding representation $\boldsymbol{\phi}_j$ of node j is computed. It is desired that the probability of two nodes being adjacent depends on a decreasing function evaluated on the distance between these two nodes $g(\Delta_{ij})$. This function must be decreasing because long edges are assumed to occur less frequently than short ones. A possible g function that can be used is

$$g(\Delta_{ij}) = \left(1 - \frac{\Delta_{ij} - \Delta_{\min}}{\Delta_{\max} - \Delta_{\min}}\right)^{\alpha} \tag{5.21}$$

where $\Delta_{\min}$ is the minimum distance found between each possible pair of nodes in the graph G and $\Delta_{\max}$ is the maximum distance found. The parameter α determines the extent to which the embedding is taken into account in the GCL model. If $\alpha = 0$, the model is equivalent to the original Chung-Lu model without taking the embedding into account. The larger α, the more long edges will be penalized.

The probability of existence of a link between two nodes in the GCL model is independent of the probability of existence of the other edges and can be computed as follows

$$\mathrm{P}(i,j) = p_{ij} = w_i w_j \Delta_{ij} \tag{5.22}$$

[5]The code used in the experiments can be found at `https://github.com/KrainskiL/CGE.jl` (visited on Mar. 5, 2022)

where w_i is the weight of the node i which can be approximated numerically such that

$$d_i = w_i \sum_{j=1}^{n} w_j g(\Delta_{ij}) \tag{5.23}$$

which means that the weights must be selected so that the expected degree of the node i is still d_i.

The global and the local divergence scores can be computed thanks to a seven-steps procedure:

1. Find the partition $\mathcal{C}$ in k communities of the graph G either by giving it as an argument to the algorithm if it is known, or by using a clustering algorithm.

2. Define the graphs vectors $\bar{c}$ and $\hat{c}$ which characterize the partition $\mathcal{C}$

$$\begin{cases} \hat{c} = (c_1, ..., c_k) \\ \bar{c} = (c_{11}, ..., c_{1k}, c_{23}, ..., c_{k-1k}) \end{cases} \tag{5.24}$$

   where c_i is the proportion of edges whose two extremities are in $\mathcal{C}_i \in \mathcal{C}$ and c_{ij} is the proportion of edges whose one extremity is in $\mathcal{C}_i$ and the other in $\mathcal{C}_j$ with $\mathcal{C}_i, \mathcal{C}_j \subset \mathcal{C}, \mathcal{C}_i \cap \mathcal{C}_j = \emptyset$. The vector $\hat{c}$ has k entries and the vector $\bar{c}$ has $\binom{k}{2}$ entries.

Global divergence score:

3. Define the GCL model $\mathcal{G}(\mathbf{d}, \mathbf{\Phi}, \alpha)$ as a function of the degree vector $\mathbf{d}$, the parameter α and the embedding $\mathbf{\Phi}$. The model vectors $\bar{b}_{\mathbf{\Phi}}$ and $\hat{b}_{\mathbf{\Phi}}$ characterizing the partition $\mathcal{C}$ from the perspective of the embedding $\mathbf{\Phi}$ can be defined in a similar way to the graph vectors with $\hat{b}_i^{\mathbf{\Phi}}$ representing the expected probability of edges having both extremities in the community $\mathcal{C}_i$ and $\bar{b}_{ij}^{\mathbf{\Phi}}$ the expected probability of edges having one extremity in the community $\mathcal{C}_i$ and the other in the community $\mathcal{C}_j$.

4. Use the Jensen-Shannon divergence [42], a smoothed version of the Kullback-Leibler divergence, to measure the dissimilarity between graph vectors and model vectors

$$\Delta_\alpha^g = \tfrac{1}{2}\Big(JSD(\hat{c}, \hat{b}_{\mathbf{\Phi}}) + JSD(\bar{c}, \bar{b}_{\mathbf{\Phi}})\Big) \tag{5.25}$$

   This allows to measure the quality of the fit of the model $\mathcal{G}$ to the graph G. The vectors $\hat{c}$ and $\hat{b}_{\mathbf{\Phi}}$ represent the internal edges of a cluster and the vectors $\bar{c}$ and $\bar{b}_{\mathbf{\Phi}}$ represent the external edges between clusters. If more importance should be given to the inner or outer edges, the factor one half can be updated.

5. Select the parameter α, located in the interval $[0, ..., 10]$, such that

$$\tilde{\alpha} = \arg\min_{\alpha} \Delta_\alpha^g \tag{5.26}$$

   The global divergence score will then be

$$\Delta_{\mathbf{\Phi}}^g(G) = \Delta_{\tilde{\alpha}}^g \tag{5.27}$$

Local divergence score:

6. Define the classes S^+ and S^- as

$$\begin{cases} S^+ = \{(u, v) \mid u \neq v \text{ and } (u, v) \in \mathcal{E}\} \\ S^- = \{(u, v) \mid u \neq v \text{ and } (u, v) \notin \mathcal{E}\} \end{cases} \tag{5.28}$$

   Using the probability p_{uv} that an edge (u, v) is present under the GCL model, the AUC p_α will measure how well this model is able to distinguish between the two classes S^+ and S^-

$$p_\alpha = \frac{\sum_{(s,t)\in S^+} \sum_{(u,v)\in S^-} \mathbb{1}\{p_{st} > p_{uv}\}}{|S^+|\,|S^-|} \tag{5.29}$$

This can be seen as the probability that $p_{st} > p_{uv}$ provided that (s,t) is randomly selected in S^+ and (u,v) is randomly selected in S^-.

7. Select the parameter α such that

$$\hat{\alpha} = \arg\min_{\alpha}(1 - p_\alpha) \tag{5.30}$$

The local divergence score will then be

$$\Delta^l_{\mathbf{\Phi}}(G) = 1 - p_{\hat{\alpha}} \tag{5.31}$$

Despite their diametrically different points of view, the global and local divergence scores are often correlated and a good embedding must score well in both to capture the global and local properties of the graph. The global and local divergence scores can be combined linearly to give a unique score for the i^{th} embedding $\mathbf{\Phi}_i$

$$\Delta_{\mathbf{\Phi}_i}(G) = q\frac{\Delta^g_{\mathbf{\Phi}_i}(G) + \epsilon}{\min_{i' \in 1..l} \Delta^g_{\mathbf{\Phi}'_i}(G) + \epsilon} + (1-q)\frac{\Delta^l_{\mathbf{\Phi}_i}(G) + \epsilon}{\min_{i' \in 1..l} \Delta^l_{\mathbf{\Phi}'_i}(G) + \epsilon} \tag{5.32}$$

where l is the number of embeddings compared and ϵ prevents division by zero. The parameter q is used to weight the combined divergence score according to what is to be highlighted, i.e. local or global properties. A typical value is $q = \frac{1}{2}$. The global and local scores must be normalized separately as they have different orders of magnitude. $\Delta_{\mathbf{\Phi}_i}(G)$ will be equal to 1 if and only if $\mathbf{\Phi}_i$ has no better competitor in either of the two divergent scores. The best embedding among all considered embeddings can be found with

$$\Delta_{\mathbf{\Phi}}(G) = \arg\min_{i \in 1..l} \Delta_{\mathbf{\Phi}_i}(G) \tag{5.33}$$

The lower the score, the better the embedding.

The combined divergence score gives a single score to each embedding, but to make a more informed decision on the best embedding it can be useful to look at the pair of scores $\left(\Delta^g_{\mathbf{\Phi}_i}(G), \Delta^l_{\mathbf{\Phi}_i}(G)\right)$ separately, e.g. in a graph where each score is placed on a different axis as shown in Figure 5.2. In this Figure, each point represents a set of parameters for the corresponding distance measure. It can be seen that for some distance measures such as logarithmic forest (LF) and the randomized shortest-path (RSP), the chosen parameter has a lot of influence as some points are far from the origin while others are well ranked. For the Poisson surprisal distance measure (PSURP), the figure shows that the parameters have a strong influence on the performance of the local divergence score but less on the global score.

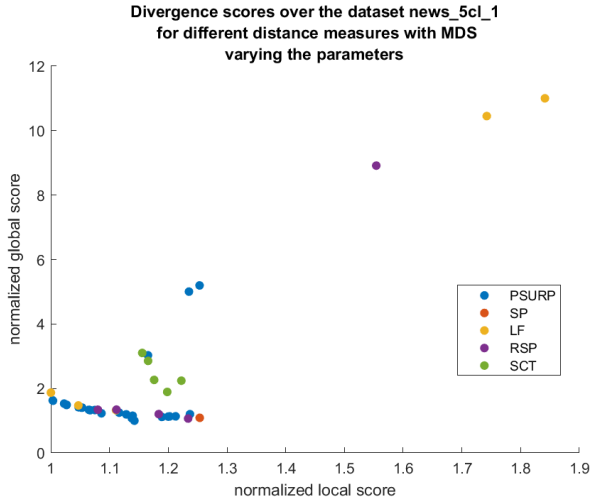

Fig. 5.2.: Illustration of the plot representing the global and local scores

The following chapter will present quality measures that allow the evaluation of the partition computed by the clustering techniques presented in this chapter but also statistical tests that will allow comparing the performances of the embeddings between them.

---

6 Assessing Methods and Experimental Methodology

This chapter first presents four performance metrics to evaluate the similarity between the clustering computed on a graph G and the ground-truth clustering of this graph. Then, it introduces different methods and statistical tests to evaluate the performance of one technique against another. Once these statistical tests have been introduced, this chapter presents the datasets on which the experiments have been performed as well as the methodology used to carry out these experiments which will use all the theoretical notions described in the previous chapters.

6.1 Quality Measures

6.1.1 Normalized Mutual Information

The normalized mutual information, or NMI, [15, 21] is a measure based on the mutual information that evaluates the similarity of two partitions in clusters of a set of nodes.

The mutual information between the set of natural clusters Ω and the computed set of clusters $\mathcal{C}$ can be defined as follows

$$I(\mathcal{C},\Omega) = \sum_k \sum_j \mathrm{P}(\mathcal{C}_k \cap \Omega_j) \log \frac{\mathrm{P}(\mathcal{C}_k \cap \Omega_j)}{\mathrm{P}(\mathcal{C}_k)\mathrm{P}(\Omega_j)} \tag{6.1}$$

where $\mathrm{P}(\Omega_j)$ (respectively $\mathrm{P}(\mathcal{C}_k)$) is the probability that a node belongs to the class $\Omega_j \in \Omega$ ($\mathcal{C}_k \in \mathcal{C}$) and $\mathrm{P}(\mathcal{C}_k \cap \Omega_j)$ is the probability that a node belongs to the classes Ω_j and $\mathcal{C}_k$.

The minimum value that the mutual information can take is 0 when all the clusters are randomly distributed among the true classes. The maximum value of the mutual information will be reached when the computed clusters are the same as the real ones. However, this measure is sensitive to the number of clusters of the detected partition. The higher the number of clusters, the higher the mutual information, which may give a wrong perception of the relative performance of the algorithm. To penalize this behavior, the normalized mutual information will divide the mutual information by the arithmetic average of the entropies of Ω and $\mathcal{C}$, such that

$$NMI(\mathcal{C},\Omega) = \frac{2\, I(\mathcal{C},\Omega)}{H(\mathcal{C}) + H(\Omega)} \tag{6.2}$$

with

$$\begin{aligned} H(\mathcal{C}) &= -\sum_k \mathrm{P}(\mathcal{C}_k) \log \mathrm{P}(\mathcal{C}_k) \\ H(\Omega) &= -\sum_j \mathrm{P}(\Omega_j) \log \mathrm{P}(\Omega_j) \end{aligned} \tag{6.3}$$

This will allow to normalize the NMI between 0 and 1, but also to have a trade-off between the quality of the computed clustering and the number of clusters found.

6.1.2 Adjusted Rand Index

The adjusted Rand index, or ARI, is a measure that is derived from the Rand index. The Rand index [15, 26, 52, 55] aims at comparing two partitions $\mathcal{C}$ and Ω by looking for each pair of nodes if they are classified in the same clusters in both partitions. Let's define four quantities to characterize the relation between the pairs of nodes

TP : Number of pairs of nodes that are in the same clusters for both partitions

FP : Number of pairs of nodes that are in the same clusters for the natural clustering Ω, but in different clusters for the computed clustering $\mathcal{C}$

FN : Number of pairs of nodes that are in different clusters for Ω, but in the same clusters for $\mathcal{C}$

TN : Number of pairs of nodes that are in different clusters for both partitions

From these quantities, the Rand index will compute the number of well-classified pairs in the two partitions out of the total number of pairs

$$RI(\mathcal{C}, \Omega) = \frac{TP + TN}{TP + TN + FP + FN} \tag{6.4}$$

However, this measure approaches its upper limit of 1 when the number of clusters increases even if the partitions are not the same. Furthermore, when comparing two random partitions, the Rand index does not take a constant value. To correct these problems, the adjusted Rand index was introduced [26]

$$ARI(\mathcal{C}, \Omega) = \frac{\binom{n}{2}(TP + TN) - [(TP + FP)(TP + FN) + (TN + FN)(TN + FP)]}{\binom{n}{2}^2 - [(TP + FP)(TP + FN) + (TN + FN)(TN + FP)]} \tag{6.5}$$

6.1.3 Correct Classification Rate

The correct classification rate, or CCR, [55, 60] is a measure that evaluates the number of nodes that have been correctly classified when looking at the ground truth clustering Ω. Unlike the classification rate from which CCR is derived, it attempts to permute the labels returned by the algorithm to find the order of the labels with the maximum matching with respect to Ω. By defining $\mathbf{T}$ as the confusion matrix between the partition $\mathcal{C}$ and Ω, the Correct Classification Rate can be defined as

$$CCR = \frac{\mathrm{tr}(\mathbf{T})}{n} \tag{6.6}$$

where $\mathrm{tr}(\mathbf{T})$ is the trace of the confusion matrix and CCR is maximized by permuting the rows of $\mathbf{T}$. A drawback of this measure is that it leads to erroneous conclusions when the number of nodes in each class is highly unbalanced.

6.1.4 Modularity Criterion

The modularity [16, 49] is a criterion that evaluates the quality of a clustering partition by summing over all clusters the difference between the observed fraction of edges within each cluster and the expected one. The modularity criterion is bounded by 1 when the partition is perfect, it has positive values if the number of edges between the nodes of a cluster is higher than random chance would allow. It can be defined as

$$\mathcal{Q} = \frac{1}{2m} \sum_{ij} \left(a_{ij} - \frac{d_i d_j}{2m} \right) \delta_{\mathcal{C}_i \mathcal{C}_j} \tag{6.7}$$

where m is the number of edges, a_{ij} and d_i are defined in Subsection 2.1.1 and $\delta_{\mathcal{C}_i \mathcal{C}_j}$ is the Kronecker delta indicating whether the cluster containing i ($\mathcal{C}_i$) is the same as the one containing j ($\mathcal{C}_j$), i.e. $\mathcal{C}_i = \mathcal{C}_j$.

6.2 Performance Comparison

6.2.1 Borda Count Method

Borda count method [63] is a voting technique that allows a rank to be assigned to each of the methods that will be compared. For each dataset, the M methods will be sorted according to their performances

(see Section 6.1). The method with the best performance will get a score of M, the second best will get a score of $M-1$, the third a score of $M-2$, etc. The final ranking computed by the Borda count is obtained by adding up for each method the scores received for all the datasets. The method with the highest total score is considered the best.

The advantage of this technique is that it uses the full ranking information for each dataset, but also returns a ranking for all methods. However, an important assumption made when performing the summation operation is that all datasets are of equal importance, which will not always be the case.

6.2.2 Friedman Test

The Friedman test [13, 22] is a non-parametric test equivalent to the repeated-measures ANOVA. It creates a decreasing ranking of algorithms for each dataset based on their performance with r_i^j the rank of the j^{th} algorithm on the i^{th} dataset. The mean rank of an algorithm can therefore be computed over all datasets as follows

$$R_j = \frac{1}{N}\sum_{i=1}^{N} r_i^j \tag{6.8}$$

with N the number of datasets considered. The comparison between the algorithms is carried out under the null hypothesis that all the algorithms are equivalent and therefore their mean rank R_j should be equal at a confidence level α. The Friedman statistical value associated with this comparison can be defined as

$$\chi_F^2 = \frac{12N}{M(M+1)}\left[\sum_{j=1}^{M} R_j^2 - \frac{M(M+1)^2}{4}\right] \tag{6.9}$$

with M the number of algorithms considered. The Friedman statistic is distributed according to a χ_F^2 distribution with $M-1$ degrees of freedom when M and N are large enough (i.e. $M > 5$ and $N > 10$).

6.2.3 Nemenyi Test

The Nemenyi test [13, 48] is a post-hoc test, equivalent to the Tukey test for ANOVA, which will be used when the null hypothesis of the Friedman test is rejected, i.e. at least one of the algorithms differs from another. This test will compare all the algorithms with each other on all datasets and measure if they are significantly different. Two algorithms are significantly different if their corresponding mean ranks differ in an absolute way by at least the critical difference CD

$$CD = q_\alpha \sqrt{\frac{M(M+1)}{6N}} \tag{6.10}$$

where q_α is a critical value based on the Studentized range statistics divided by $\sqrt{2}$.

6.2.4 Wilcoxon Signed-Rank Test

The Wilcoxon signed-rank test [13, 66] is a non-parametric alternative to the paired t-test. It aims to rank for each dataset the difference in the performance of two algorithms, comparing ranks for the positive and the negative differences.

Considering d_i as the difference in performance of the two algorithms on the i^{th} dataset, two quantities can be defined. R^+ computes the sum of the ranks of the datasets on which the second algorithm outperformed the first and R^- computes the sum of the ranks on which the first algorithm outperformed the second

$$\begin{aligned} R^+ &= \sum_{\mathrm{d}_i>0} \mathrm{rank}(\mathrm{d}_i) + \tfrac{1}{2}\sum_{\mathrm{d}_i=0} \mathrm{rank}(\mathrm{d}_i) \\ R^- &= \sum_{\mathrm{d}_i<0} \mathrm{rank}(\mathrm{d}_i) + \tfrac{1}{2}\sum_{\mathrm{d}_i=0} \mathrm{rank}(\mathrm{d}_i) \end{aligned} \tag{6.11}$$

If there is a tie (i.e. $\mathrm{d}_i = 0$), the rank is shared between the two quantities. From these quantities can be defined the statistical test

$$z = \frac{\min(R^+, R^-) - \frac{1}{4}N(N+1)}{\sqrt{\frac{1}{24}N(N+1)(2N+1)}} \tag{6.12}$$

with N the number of datasets considered. Compared to the t-test, the Wilcoxon test is more sensitive, but outliers have fewer impacts [13].

6.3 Datasets

This section describes the datasets that will be used to compare the different techniques described in Section 2.2 and Chapters 3 and 4. All these datasets have ground-truth communities.

Dolphin Datasets

The dolphin datasets [43, 44] are based on the research made by Lusseau into communication between bottlenose dolphins in New Zealand. Dolphins are linked to each other if they interact more often than chance allows. This dataset consists of 62 dolphins that can be divided into 2 or 4 communities.

Football Dataset

This dataset [23] is based on the Division 1 game schedule for the 2000 United States College Football season. Each node, representing a participating team, is linked to another one if the teams have regular-season games. The communities are easy to see because the teams are divided into 12 conferences of 8 to 12 teams and games are more frequent between teams in the same conference.

LFR Datasets

The LFR datasets [34] are artificially generated using the Lancichinetti-Fortunato-Radicchi model which attempts to generate graphs that mimic the properties of real-world networks. Three datasets of 600 nodes are proposed, one with 3 clusters and the other two with 6.

Newsgroup Datasets

In 1995, Lang [35] collected 20 000 unstructured documents from 20 different newsgroups or topics. From this database, 9 datasets composed of 2, 3 or 5 topics were extracted, each topic containing 200 documents [70]. The adjacency matrix of these datasets is weighted, with the weight of an edge representing the number of keywords shared by the two documents.

Political Books Dataset

This dataset is composed of US political books [67]. It was proposed by Krebs and labelled by Newman. Two books are connected if they are frequently purchased together by a customer on the Amazon online shop. The dataset consists of 105 books that can be classified into three communities: conservative, neutral and liberal.

Zachary Dataset

The Zachary dataset [71] is based on a 34-person karate club that was split in two after an argument. The graph represents the relationship between the members and two clusters can be highlighted.

The Table 6.1 presents an overview of all datasets by summarizing their number of nodes and their number of clusters.

Database name	Dataset name	Number of nodes	Number of ground-truth clusters
Dolphin	dolphins_2	62	2
	dolphins_4	62	4
Football	football	115	12
LFR	LFR1	600	3
	LFR2	600	6
	LFR	600	6
Newsgroup	news_2cl_1	400	2
	news_2cl_2	400	2
	news_2cl_3	400	2
	news_3cl_1	600	3
	news_3cl_2	600	3
	news_3cl_3	600	3
	news_5cl_1	1000	5
	news_5cl_2	1000	5
	news_5cl_3	1000	5
Political Books	polbooks	105	3
Zachary	zachary	34	2

Tab. 6.1.: Overview of all datasets used

6.4 Experimental Procedure

The first step in the experiments for each dataset is to compute its kernel and embedding matrices. These matrices are built on a particular distance measure for which the best parameter must be tuned (see Table 6.2). For each tuple (dataset, distance method, parameter(s)) the associated kernel matrix will be computed, as well as the multidimensional scaling embedding and the t-Distributed Stochastic Neighbor embedding, which requires an additional parameter to tune the perplexity. The bag-of-paths embedding, or BoPE (see Section 4.3), will also be computed on each dataset with the appropriate parameter to tune.

Parameter	Tested Values	Methods
θ	0.001, 0.01, 0.1, 1, 10	FE, SURP, RSP, PSURP, PWSURP, FEM, BoPE
α	0.001, 0.01, 0.1, 1, 10	LF
α	10, 20, 30, 40, 50	SCT, SCCT
γ	1, 2, 3, 5, 10	PSURP, PWSURP
perplexity	5, 10, 30, 50	tSNE

Tab. 6.2.: Set of values tested for each parameter associated with the distance methods that use this parameter

The Tables 6.3, 6.4 and 6.5 shows the abbreviations used in the following chapter for, respectively, the distance measures, embedding methods and clustering methods.

Abbreviation	Distance measure
SP	Shortest-Path distance
CT	Commute Time distance
SCT	Sigmoid Commute Time distance
SCCT	Sigmoid Corrected Commute Time distance
LF	Logarithmic Forest distance
FE	Free Energy distance
SURP	Surprisal distance
RSP	Randomized Shortest Path distance
PWSURP	Poisson Weighted Surprisal distance
PSURP	Poisson Surprisal distance
FEM	Free Energy Modified distance

Tab. 6.3.: Abbreviations used for each distance measure

Abbreviation	Embedding method
MDS	Multidimensional Scaling performed on given distance measure
TSNE	t-Distributed Stochastic Neighbors Embedding performed on given distance measure
BoPE	Bag-of-paths Embedding (performed on FEM)

Tab. 6.4.: Abbreviations used for each embedding method

Abbreviation	Clustering method
KKM	Kernel k-means performed on kernel matrix
EKM	Standard k-means performed on embedding matrix

Tab. 6.5.: Abbreviations used for each clustering method

6.4.1 Clustering Procedure

One of the objectives of the clustering is to find the optimal parameter values for each distance measure. The clustering methods used will be kernel k-means on kernel matrices and standard k-means on embedding matrices. For each method and parameter, the clustering algorithm will be run 30 times with different initialisations as both algorithms are initialisation sensitive, and the partition with the highest modularity will be kept. This procedure will be performed 30 times and the four quality scores (NMI, ARI, CCR and modularity) will be averaged over these 30 trials. For each distance measure, the set of parameters chosen will be the one with the highest averaged modularity [10, 39].

6.4.2 Rank-based Criteria Procedure

For the rank-based criteria, two distance matrices will be defined for each tuple (dataset, distance measure, set of parameter(s)) to compute the Q_{NX} vector depending on κ, the R_{NX} vector and the AUC, . These are the distance matrix in high-dimensional space computed with Dijkstra on the cost matrix of the graph and the distance matrix in the low-dimensional space computed as the Euclidean distance between each pair of nodes of the given embedding. For each dataset and distance measure, the best parameter(s) will be the set with the highest AUC.

6.4.3 Combined Divergence Score Procedure

The combined divergence score is a technique that receives the edges of the graph G, their real clustering and the embedding to be evaluated as arguments. It returns two scores, the global divergence and the local divergence. For each dataset and each distance measure, the set of parameters with the lowest local divergence score will be considered the best embedding. The criterion is the local score because this score does not depend on the ground-truth partition of the graph, unlike the global divergence score.

The next chapter will present the results of the three research questions investigated in this thesis based on the datasets and procedures presented in this chapter.

7 Results and Discussion

In this chapter the results of our experiments will be analyzed based on the Statistical tests described in Section 6.2. The aim is to investigate the research questions that were introduced in Chapter 1. All statistical tests in this chapter are performed at the 95% confidence level, which corresponds to an α value of 0.05. The methodology and datasets used to conduct the experiments are described in the previous chapter.

7.1 First Research Question

Firstly, this section will analyze the results obtained for the three embedding assessment techniques by varying the combination (distance measure, embedding method) used to answer the question

- *Which combinations provide the best results in a three-dimensional node embedding task and, in this context, how does the introduced Poisson surprisal distance perform?*

Secondly, this section will analyze the results obtained for the three embedding assessment techniques when using MDS embedding to answer the following question

- *Which distances provide the best results in a low-dimensional (5%) node embedding task and, in this context, how does the introduced Poisson surprisal distance perform?*

The three embedding assessment techniques that will be used, presented in Chapter 5, are the standard k-means applied to the nodes embedding with the ARI score (see Subsection 6.1.2), the rank-based criteria and the combined divergence score. The embedding methods compared for the three-dimensional case are the MDS and the tSNE (see Chapter 4).

7.1.1 Three-dimensional embedding

First of all, a Borda count was performed on each assessment technique to obtain a ranking of each possible combination. The weighted sum of the three scores was computed in the last column of Table 7.1 and used to rank the methods. The Borda count is mainly informative, providing a partial view of the information and a pre-analysis of the results. From this pre-analysis, the best distance among the two embedding methods for each family of distances will be retained for use in the Friedman-Nemenyi and Wilcoxon tests. This will reduce the number of combinations compared to a single statistical test. One family of distances is the BoP family with FE, RSP and FEM, another family is the Commute Time (CT) family with CT, SCT and SCCT. The shortest-path distance and the logarithmic forest distance do not belong to any family and will always be kept as such. The remaining distances are the SURP, PSURP and PWSURP. Although they can be considered as belonging to the same family, i.e. the Surprisal family, the Poisson surprisal distances will always be kept as they are the investigated ones. The SURP will only be kept if it has a higher rank than the two Poisson surprisal distances.

Table 7.1 shows that, for the datasets studied, the clustering technique gives a higher score to combinations where the embedding was provided by the MDS method while the rank-based criteria give a higher score to combinations where the embedding was provided by the tSNE method. This phenomenon can be explained by the properties of each method and technique. The MDS embedding will try to compute a low-dimensional representation where the distances between each pair of nodes remain as close as possible to the actual distances by minimizing the residual sum, which can be considered as preserving the global properties of the graph [8]. The tSNE embedding, on the other hand, will mainly preserve the local properties of the graph by capturing its local structure [62]. As clustering will evaluate the ability of the embedding to reproduce the ground-truth clusters, it will focus on preserving the global

Criteria	Clustering		Rank-based Criteria		Combined Divergence Score		All	
Embedding	*rank*	*score*	*rank*	*score*	*rank*	*score*	*rank*	*score [%]*
PWSURP_TSNE	7	237	1	336	14	210	1	5.972
LF_TSNE	9	216	2	309	11	222	2	5.701
PWSURP_MDS	1	287	9	215	4	242	3	5.653
SURP_TSNE	15	206	4	289	3	244	4	5.644
PSURP_MDS	2	283	8	229	8	229	5	5.631
RSP_TSNE	13	208	3	291	12	218	6	5.472
FE_TSNE	15	206	6	279	8	229	7	5.450
FEM_TSNE	10	214	7	264	6	236	8	5.447
FE_MDS	4	271	14	174	1	264	9	5.389
FEM_MDS	5	270	15	165	2	249	10	5.196
SURP_MDS	3	278	15	165	5	240	11	5.185
RSP_MDS	6	259	12	187	7	235	12	5.176
PSURP_TSNE	13	208	4	289	15	178	13	5.147
LF_MDS	12	209	18	151	8	229	14	4.483
SCT_TSNE	8	218	11	207	18	148	15	4.354
SCCT_TSNE	18	166	13	185	17	171	16	3.979
SP_MDS	19	145	17	156	13	211	17	3.910
SP_TSNE	20	124	10	209	16	172	18	3.863
SCT_MDS	11	212	20	62	20	125	19	3.011
SCCT_MDS	17	187	21	54	19	134	20	2.834
CT_MDS	21	89	19	76	22	40	21	1.554
CT_TSNE	22	29	22	20	21	75	22	0.950

Tab. 7.1.: Borda count in 3D-space of the three assessment techniques when using the MDS and tSNE embedding methods for each distance measure

properties and communities of the graph (see Section 5.1). The rank-based criteria, for its part, use a score depending on the number of neighbors and focus on small neighborhoods trying to preserve mainly the local structure (see Section 5.2). Therefore, clustering will favour MDS embeddings and rank-based criteria will favour tSNE embeddings. The combined divergence score, which uses both global and local properties, and the weighted sum of the three scores will favour both embeddings depending on the distance measure used rather than the embedding method used.
Figures 7.1 and 7.2 show the representation of embeddings computed with the same distance measure but with different embedding methods. The colors represent the ground-truth partition, one can notice that the two embedding methods separate the clusters but in a very different way[6].

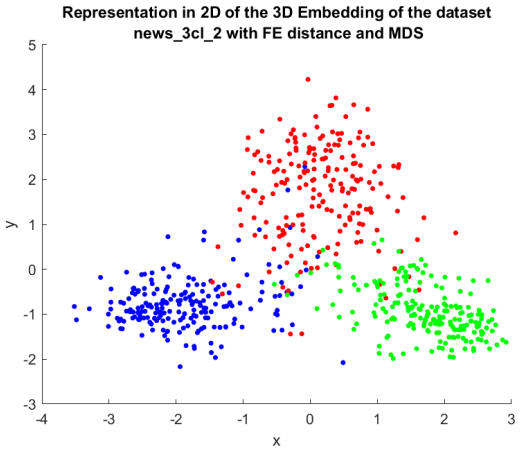

Fig. 7.1.: Embedding computed with MDS

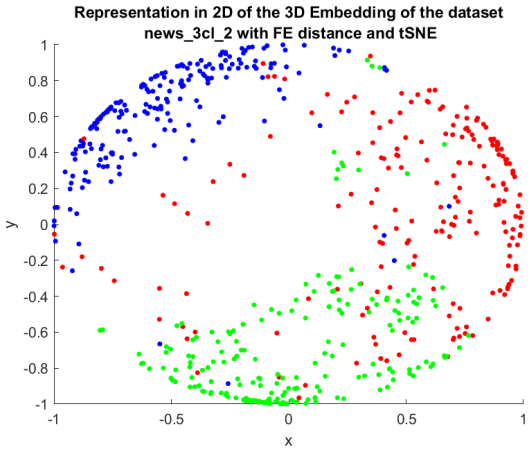

Fig. 7.2.: Embedding computed with tSNE

Focusing on the best distances, the Borda count of Table 7.1 shows that the Surprisal and the BoP families seem to give better overall scores for both embedding methods. It also indicates that, as expected and already shown in [64, 70], the SCT and SCCT distances perform better than the CT distance. Furthermore, the FEM, which is a modified version of FE, is always ranked close to the FE, which tends to imply that they give similar scores.

The second step of the analysis is to compute for each evaluation technique a Friedman test on the combinations chosen. The p-values obtained are 2.5×10^{-4} for clustering, 9×10^{-17} for the rank-based criteria and 1.3×10^{-2} for the combined divergence score. As all these p-values are below the threshold α

[6]For the interested readers, an example with another distance measure and another dataset is presented in the Appendix A

of 0.05, the null hypothesis of the Friedman test can be rejected, which means that at least one combination differs from the others. A multiple comparison for each technique can therefore be performed with the Nemenyi test as shown in Figures 7.3, 7.4 and 7.5. These figures show the Nemenyi test for the clustering, the rank-based criteria and the divergence score respectively. As can be seen from these three Figures and as might be expected given the Borda count, the Nemenyi tests are very different when using each of the evaluation techniques.
In a Nemenyi test, two combinations are considered significantly different if their confidence intervals do not overlap. The horizontal axis represents the average rank of the combinations. The higher the rank, the better the combination. The best combination is highlighted in blue in each test and this combination is significantly better than the combinations highlighted in red.

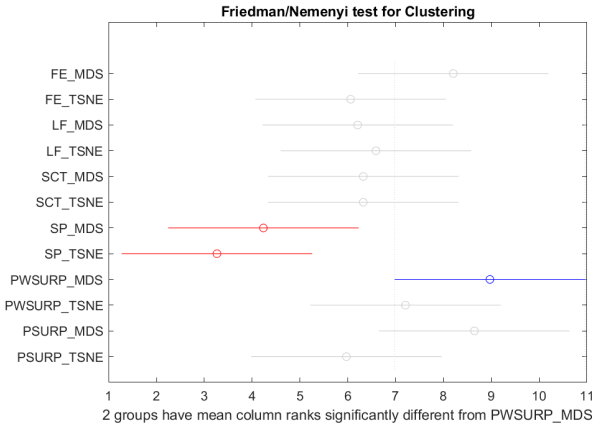

Fig. 7.3.: Nemenyi test for the clustering technique in 3D-space

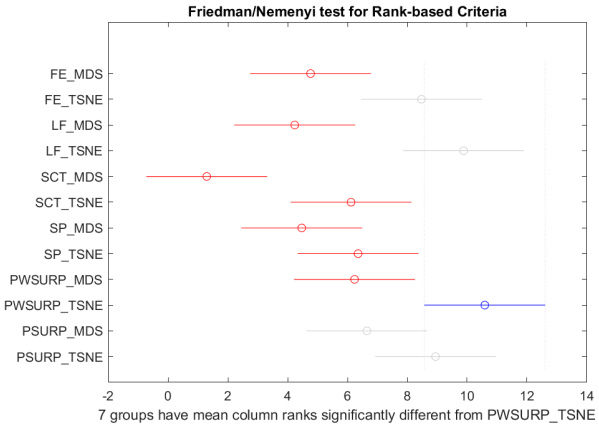

Fig. 7.4.: Nemenyi test for the rank-based criteria technique in 3D-space

According to Figure 7.3, the best combination is PWSURP with MDS and the second best is PSURP with MDS. They both perform significantly better than the two combinations with the lowest mean rank, namely the distance measure SP with MDS and with tSNE. When looking at all combinations including a Poisson surprisal distance, the clustering technique finds none that perform significantly better or worse than the others. Figure 7.4 shows that the rank-based criteria found that the PWSURP with tSNE is the best combination. It also shows that this combination is significantly better than all the others except for the other four highest-ranked combinations, namely LF with tSNE, PSURP with tSNE, FE with tSNE and PSURP with MDS. It can thus be noticed that the PWSURP with tSNE is significantly better than the PWSURP with MDS. The best combination of the divergence score (Figure 7.5) is FE with MDS, it performs significantly better than the method with the worst mean rank SCT with MDS. Focusing on the Poisson surprisal distance combinations, the Figure shows that they are not significantly worse than the best combination and none is significantly different from any other.

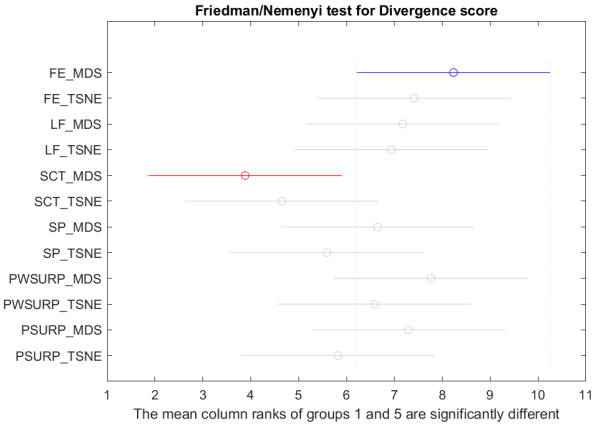

Fig. 7.5.: Nemenyi test for the divergence score technique in 3D-space

Comparing each distance measure used with MDS to the same distance measure used with tSNE, the mean ranks of the combinations using MDS are, for the clustering technique (Figure 7.3) and the divergence

score (Figure 7.5), generally higher than those of the combinations using tSNE. The exceptions are the LF for the clustering and the SP for the divergence score. However, the opposite is true for the rank-based criteria (Figure 7.4) as the mean ranks of all distance measures used with tSNE are significantly better than those of the same distance measure used with MDS. Considering the mean ranks of the Poisson surprisal distance combinations, the mean rank of the PWSURP is overall higher than those of the PSURP. Moreover, PWSURP combinations are significantly better for a larger number of combinations than the PSURP combinations.

To compare the Poisson surprisal distances to the other distances with more accurate results, the third step is to perform pairwise Wilcoxon signed-rank tests on all combinations. Table 7.2 presents the p-values of all these Wilcoxon tests according to the evaluation technique considered, i.e. clustering (denoted *Cl*), rank-based criteria (denoted *R-b*) and divergence score (denoted *D.s.*). Two methods are considered significantly different if the p-value of their test is less than 0.05. As expected from the Nemenyi tests, the p-values of the rank-based criteria more often indicate a significant difference between two combinations than the clustering and the divergence score.

		FE MDS	FE TSNE	LF MDS	LF TSNE	SCT MDS	SCT TSNE	SP MDS	SP TSNE	PWSURP MDS	PWSURP TSNE	PSURP MDS	PSURP TSNE
PWSURP MDS	*Cl*	0.670	**0.025**	**0.008**	0.055	**0.018**	0.056	**0.002**	**0.001**	/	0.177	0.641	0.062
	R-b	**$<10^{-4}$**	**0.007**	**$<10^{-4}$**	**0.007**	**$<10^{-4}$**	**0.044**	0.981	**0.049**	/	**0.006**	**0.049**	**0.006**
	D.s.	0.492	0.407	0.795	0.723	**0.001**	0.124	0.332	0.055	/	0.332	0.193	0.076
PWSURP TSNE	*Cl*	0.301	0.173	0.605	0.952	0.619	0.717	**0.034**	**0.013**	0.177	/	0.196	0.424
	R-b	**0.002**	**$<10^{-4}$**	**0.003**	0.554	**$<10^{-4}$**	**$<10^{-4}$**	**$<10^{-4}$**	**0.001**	**0.006**	/	**0.006**	**0.002**
	D.s.	0.463	0.356	0.435	0.619	0.723	0.619	0.831	0.407	0.332	/	0.332	0.492
PSURP MDS	*Cl*	0.635	**0.028**	**0.011**	0.055	**0.022**	0.070	**0.001**	**0.001**	0.641	0.196	/	0.068
	R-b	**$<10^{-4}$**	**0.007**	**0.002**	**0.007**	**$<10^{-4}$**	0.163	**0.001**	0.055	**0.049**	**0.006**	/	**0.006**
	D.s.	0.435	0.407	0.943	0.723	**0.001**	0.136	0.356	0.177	0.193	0.332	/	0.193
PSURP TSNE	*Cl*	0.149	0.639	0.795	0.715	0.981	0.796	0.109	**0.017**	0.062	0.424	0.068	/
	R-b	**0.004**	0.619	**0.004**	0.113	**$<10^{-4}$**	**0.001**	**$<10^{-4}$**	0.055	**0.006**	**0.002**	**0.006**	/
	D.s.	**0.028**	0.227	0.210	0.332	0.463	0.523	0.102	0.653	0.076	0.492	0.193	/

Tab. 7.2.: Comparison of the p-values of the Wilcoxon signed-rank tests performed between the Poisson surprisal distances and the other considered distances in 3D space

In the Table 7.2, the p-values of the clustering technique are more discriminating than what was observed in the corresponding Nemenyi test because the two best combinations (PWSURP with MDS and PSURP with MDS) are not only significantly better than the SP combinations but also FE with tSNE, LF with MDS and SCT with MDS. Table 7.2 shows also that the best combination in the Nemenyi test of the rank-based criteria, i.e. PWSURP with tSNE, is significantly better than all the combinations except the second-best LF with tSNE. Focusing on the divergence score, the table shows that only the PSURP with tSNE is significantly worse than the best method of the Nemenyi test, i.e. FE with MDS. It also shows that PWSURP with MDS and PSURP with MDS are significantly better than the worse distance of the Nemenyi test, i.e. SCT with MDS. When looking at the comparison of the Poisson surprisal distances between them in Table 7.2, the rank-based criteria indicate that they are all significantly different while the clustering and the divergence scores show that there is no significant difference between them for these assessment tasks. This confirms what was shown in the Nemenyi tests, except for the rank-based criteria for which the Wilcoxon tests are more discriminating than the Nemenyi test.

Pairwise comparisons can also be made for each distance measure to determine whether there is a significant difference between the embedding performed with MDS and the one performed with tSNE. As expected from the Nemenyi tests, Table 7.3 shows that the rank-based criteria consider the performances of each distance with MDS to be significantly inferior to the performances of the same distance with tSNE while no significant difference can be highlighted for the clustering and the divergence score.

	FE	LF	SCT	SP	PWSURP	PSURP
Cl	0.063	0.906	0.535	0.088	0.177	0.068
R-b	**0.006**	**0.004**	**$<10^{-4}$**	**0.028**	**0.006**	**0.006**
D.s.	0.723	0.943	0.687	0.177	0.332	0.193

Tab. 7.3.: Comparison of the p-values of the Wilcoxon signed-rank tests performed over each distance measure between MDS and tSNE embeddings in 3D space

In conclusion, when considering the embedding methods used on the studied datasets, the rank-based criteria are by far the most discriminating and seem to have a high preference for the embeddings provided by tSNE. Focusing on the Poisson surprisal distances, they perform well on all three assessment techniques used. In the three-dimensional space, PWSURP combinations seem to perform slightly better than PSURP combinations as their mean ranks are better and their performances are significantly better from more combinations.

7.1.2 Low-dimensional embedding using 5% of the total dimensionality

To answer the second sub-question, a Borda count (see Table 7.4) will also be computed to determine which distance will be chosen for each distance family and to give an overview of the best distance measures in a low-dimensional space. As for the three-dimensional space, the Surprisal and BoP families seem to give higher scores than the other families. However, within these families, the performances of the PWSURP seem to be slightly inferior to the performances of the PSURP. As expected, the SCT is still better than the CT. The best distance for each family and the Poisson surprisal distances will be kept to run the Friedman test.

Criteria	**Clustering**		**Rank-based Criteria**		**Combined Divergence Score**		**All**	
Distance	*rank*	*score*	*rank*	*score*	*rank*	*score*	*rank*	*score [%]*
PSURP	3	140	1	159	2	131	1	12.262
RSP	6	126	3	137	1	153	2	11.898
PWSURP	2	144	2	155	4	118	3	11.863
SURP	1	157	7	95	3	122	4	10.549
FE	4	138	6	123	5	108	5	10.463
FEM	5	135	5	127	6	106	6	10.443
SP	10	81	4	130	8	91	7	8.669
LF	7	119	7	95	9	84	8	8.424
SCT	8	105	9	46	7	102	9	7.143
SCCT	9	87	10	35	10	83	10	5.782
CT	11	39	11	26	11	24	11	2.504

Tab. 7.4.: Borda count in low-dimensional space of the three assessment techniques when using the MDS embedding method for each distance measure

The p-values obtained for the Friedman tests are 2×10^{-2} for clustering, 2.1×10^{-7} for the rank-based criteria and 2.5×10^{-3} for the divergence score. As they are all below the threshold α the null hypothesis can be rejected and a multiple comparison with the Nemenyi test can be performed for each evaluation technique.

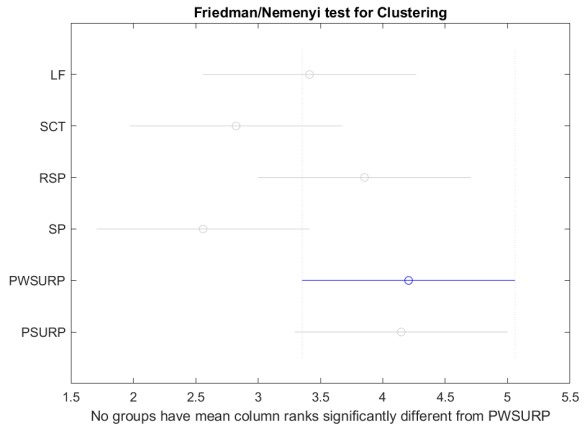

Fig. 7.6.: Nemenyi test for the clustering technique in low-dimensional space

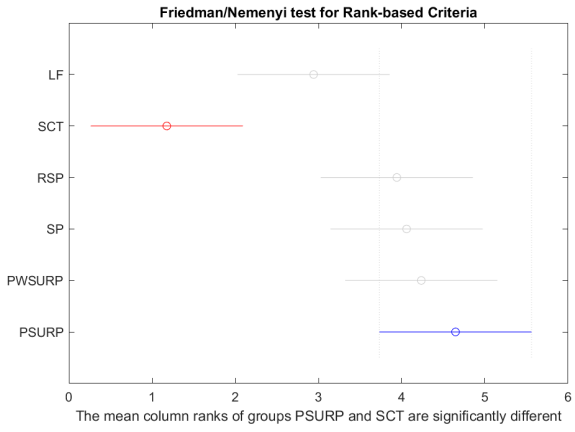

Fig. 7.7.: Nemenyi test for the rank-based criteria technique in low-dimensional space

The Nemenyi test for the clustering technique (Figure 7.6) shows that there is no significant difference between the distance measure with the best mean rank, PWSURP, and the other distances. This means that even if the null hypothesis of the Friedman test is rejected, the Nemenyi test does not find any pair whose difference is greater than the critical difference (see Subsection 6.2.3). This figure also shows that the mean ranks of PWSURP and PSURP are almost equal. Figure 7.7 depicting the Nemenyi test for the rank-based criteria shows that the distance measure with the best mean rank is PSURP and that its performance is significantly superior to those of the SCT. It can be noted, that SP seems to perform surprisingly well as its mean rank is only slightly lower than that of PSURP. Figure 7.8 representing the Nemenyi test for the divergence score shows that the best distance measure is the RSP which performs significantly better than the LF and the SP. The Poisson surprisal distances do not perform significantly worse than the RSP and are ranked second for the PSURP and third for the PWSURP. It can be observed in Figures 7.7 and 7.8 that the mean rank of PSURP is slightly higher than that of PWSURP.

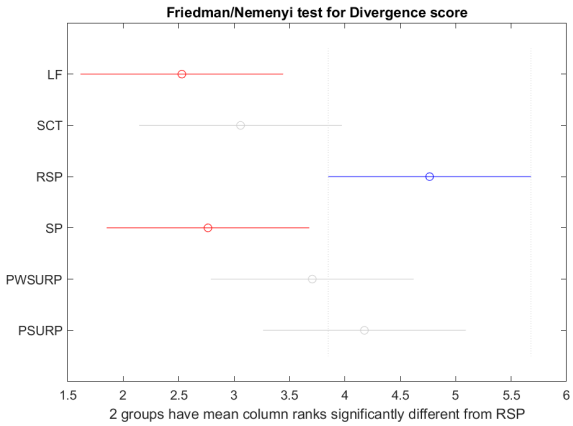

Fig. 7.8.: Nemenyi test for the divergence score technique in low-dimensional space

Wilcoxon tests can be performed to compare the Poisson distances with the other distances, as shown in Table 7.5. The p-values of the clustering technique show, in contrast to the corresponding Nemenyi test, that there is one distance measure that performs significantly worse than the Poisson distances, i.e. the SCT. The p-values of the best distance for the rank-based criteria, namely PSURP, show that it performs significantly better than not only the SCT, but also all other distance measures except for the SP which is ranked third in the Nemenyi test. Looking at the divergence score, the table indicates, as observed on the corresponding Nemenyi test, that the Poisson distances are not significantly worse than the top-ranked distance, i.e. the RSP. Comparing the PWSURP to the PSURP, the test applied to the rank-based criteria shows, as for the three-dimensional case, that they are significantly different, but the clustering and divergence score tests do not show a significant difference.

		LF	SCT	RSP	SP	PWSURP	PSURP
PWSURP	*Cl*	0.191	**0.040**	0.305	0.076	/	1
	R-b	**0.001**	**$<10^{-4}$**	0.435	0.723	/	**0.031**
	D.s.	0.055	0.163	0.309	0.619	/	0.084
PSURP	*Cl*	0.273	**0.048**	0.305	0.093	1	/
	R-b	**0.001**	**$<10^{-4}$**	**0.010**	0.554	**0.031**	/
	D.s.	**0.010**	0.163	0.356	0.381	0.084	/

Tab. 7.5.: Comparison of the p-values of the Wilcoxon signed-rank tests performed between the Poisson surprisal distances and the other considered distances in low-simensional space

In conclusion, in low-dimensional space, the Poisson distances perform quite well on the analyzed datasets. However, they do not perform better than the RSP when looking at the divergence score. In contrast to the three-dimensional space, the PSURP seems to have a slightly better mean rank than the PWSURP when used in the low-dimensional space.

7.2 Second Research Question

As clustering is an important supervised technique, this section will analyze the results obtained for the quality measures by varying the combination (distance measure, clustering method) used to answer the following question

- *Which combinations provide the best results in a node clustering task and, in this context, how does the introduced Poisson surprisal distance perform?*

The two clustering methods compared are the kernel k-means, or KKM, and the standard k-means performed on the embedding, or EKM, in a low-dimensional space provided by MDS.

The first step will be to compute a Borda count over the three quality measures considered: the ARI, the NMI and the CCR (see Section 6.1). Table 7.6 shows these Borda counts and the overall ranking allows us to see that the tested combinations have a similar ranking for the three measures on the considered datasets. The Surprisal and the BoP families are the best-ranked combinations, similar to the Borda counts in Section 7.1. It can already be seen that the computation of the clustering on 5% of the information

when using the low-dimensional embedding or 100% of the information when using the kernel gives, for several distance measures, a similar quality score. The notable exceptions are the SURP and the FEM which seem to work better when used on the embedding. As before, the best combination per family and the Poisson surprisal distances will be kept to perform the Friedman test. As the ranking and the statistical tests for the three scores are very similar, only the ARI score, the most widely used measure in the literature, will be explored in this thesis.

Criteria	**ARI**		**NMI**		**CCR**		**All**	
Clustering	*rank*	*score*	*rank*	*score*	*rank*	*score*	*rank*	*score [%]*
SURP_EKM	1	305	1	299	1	304	1	6.284
PSURP_KKM	5	275	2	285	3	281	2	5.821
PSURP_EKM	2	277	5	277	2	285	3	5.806
PWSURP_EKM	3	276	3	280	5	280	4	5.786
PWSURP_KKM	6	271	4	279	6	274	5	5.703
FEM_EKM	3	276	7	262	3	281	6	5.668
FE_EKM	6	271	6	263	7	267	7	5.544
FE_KKM	8	267	7	262	8	260	8	5.461
RSP_EKM	10	245	9	250	10	247	9	5.135
SURP_KKM	11	235	10	249	9	252	10	5.093
FEM_KKM	9	249	11	235	11	245	11	5.045
RSP_KKM	12	233	12	231	12	234	12	4.831
LF_EKM	13	225	13	217	13	222	13	4.596
SCT_KKM	15	202	14	214	15	211	14	4.339
SCT_EKM	14	205	15	209	14	213	14	4.339
LF_KKM	16	197	17	189	16	208	16	4.110
SCCT_KKM	17	190	16	198	17	195	17	4.035
SCCT_EKM	18	169	18	175	18	174	18	3.585
SP_EKM	19	160	19	155	19	147	19	3.198
SP_KKM	20	137	20	132	20	133	20	2.782
CT_EKM	21	79	21	80	21	87	21	1.702
CT_KKM	22	54	22	57	22	53	22	1.135

Tab. 7.6.: Borda count of the three quality measures considered when using the kernel k-means and the standard k-means clustering technique for each distance measure

The p-value of the Friedman test for the ARI quality measure is 4.6×10^{-5} which means that the null hypothesis can be rejected and a multiple comparison can be performed. The Nemenyi test for the ARI score is presented in Figure 7.9. It can be seen from this Figure that the best combination is the EKM

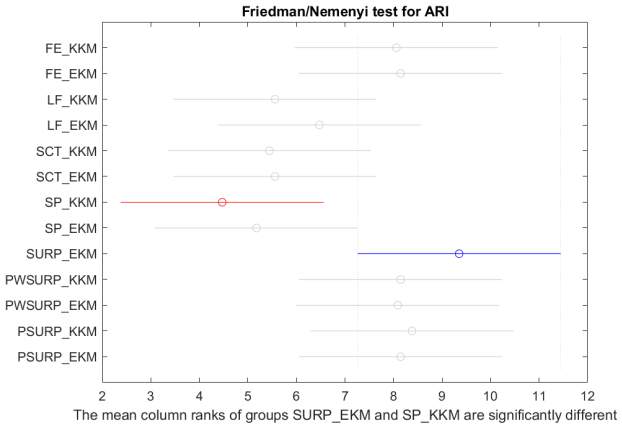

Fig. 7.9.: Nemenyi test for the ARI score when performing clustering with KKM or EKM

computed on the SURP and that it performs significantly better than the lowest-ranked combination, namely the KKM computed on the SP. It can also be observed that the mean ranks of EKM and KKM performed on the Poisson surprisal distances or the FE are close and not significantly worse than the EKM on the SURP. Wilcoxon tests can be computed to compare combinations with Poisson surprisal distances to other combinations, as shown in Table 7.7. This Table shows that all Poisson surprisal distances with EKM or KKM are significantly better than the combination with the lowest rank, i.e. SP with KKM. They are also all significantly better than KKM or EKM performed on the SCT and PSURP with EKM performs significantly better than LF with KKM. However, none of the Poisson surprisal distances performs significantly worse than the top-ranked combination SURP with EKM. Looking at the performance of the Poisson surprisal distances against each other, the performances are not significantly different from each other, as already shown in the Nemenyi test.

	FE KKM	FE EKM	LF KKM	LF EKM	SCT KKM	SCT EKM	SP KKM	SP EKM	SURP EKM	PWSURP KKM	PWSURP EKM	PSURP KKM	PSURP EKM
PWSURP KKM	0.839	0.542	0.119	0.241	**0.008**	**0.025**	**0.017**	0.102	0.296	/	0.839	0.898	0.855
PWSURP EKM	0.808	0.839	0.068	0.191	**0.049**	**0.04**	**0.031**	0.076	0.273	0.839	/	0.414	0.831
PSURP KKM	0.685	0.903	0.078	0.135	**0.006**	**0.025**	**0.017**	0.113	0.626	0.898	0.414	/	0.426
PSURP EKM	0.626	1	**0.049**	0.244	**0.049**	**0.048**	**0.028**	0.093	0.305	0.855	0.831	0.426	/

Tab. 7.7.: Comparison of the p-values of the Wilcoxon signed-rank tests performed between the Poisson surprisal distances with both clustering techniques and the other combinations of (distance, clustering)

Table 7.8 shows the p-values of the Wilcoxon signed-rank tests performed on the same distance measure when computed using EKM or KKM. It can be seen that none of the p-value is below the 0.05 threshold, which means that there is no significant difference between using 5% and 100% of the information from the chosen distance measure.

FE	LF	SCT	SP	PWSURP	PSURP
0.489	0.268	0.588	0.21	0.839	0.426

Tab. 7.8.: Comparison of the p-values of the Wilcoxon signed-rank tests performed for each distance measure between the clustering made over the kernel and the clustering made over the embedding

From this analysis, we cannot conclude that using 100% of the information gives significantly superior performance than using only 5% of the information. Nevertheless, it can be noticed that the Poisson surprisal distances perform reasonably well since their mean ranks are close to those of the combinations with the FE and as they do not perform significantly worse than the top-ranked combination SURP with EKM.

7.3 Third Research Question

This section will analyze the results obtained by comparing the bag-of-paths embedding technique with those analysed in the Section 7.1 to answer the following questions

- *Which embedding provides the best results on a three-dimensional node embedding task and, in this context, how does the introduced bag-of-paths embedding perform?*
- *Which embedding provides the best results on a low-dimensional (5%) node embedding task and, in this context, how does the introduced bag-of-paths embedding perform?*

To compare the bag-of-paths embedding, or BoPE, to other embedding techniques, only the best methods per family selected in Section 7.1 will be used. The Surprisal family has also been reduced to one distance per embedding method.

7.3.1 Three-dimensional Embedding

Figure 7.10 shows the representation of the embedding computed with the BoPE where the colors represent the ground-truth partition of the graph. It can easily be seen that this embedding is close to the one represented in Figure 7.1. This can be explained by the way BoPE is computed. Indeed, it uses a variation of the FE as a distance and computes an initial embedding thanks to MDS before proceeding to the gradient descent whereas Figure 7.1 uses the FE with MDS to compute the embedding.
The first step of the analysis is, as before, to compute the Friedman tests. Their p-value are 8.8×10^{-5} for clustering, 4.4×10^{-15} for the rank-based criteria and 6.9×10^{-3} for the divergence score. This means that the Nemenyi tests can be performed because the null hypothesis has been rejected for all three criteria. The BoPE will be highlighted in blue in the following Nemenyi tests, while combinations significantly different from it will be highlighted in red.

The Nemenyi test for the clustering technique, shown in Figure 7.11, indicates that the performances of the BoPE are significantly superior to the performances of the SP used with tSNE, the combination with the lowest mean rank. The Nemenyi test for the rank-based criteria (Figure 7.12) shows that the BoPE

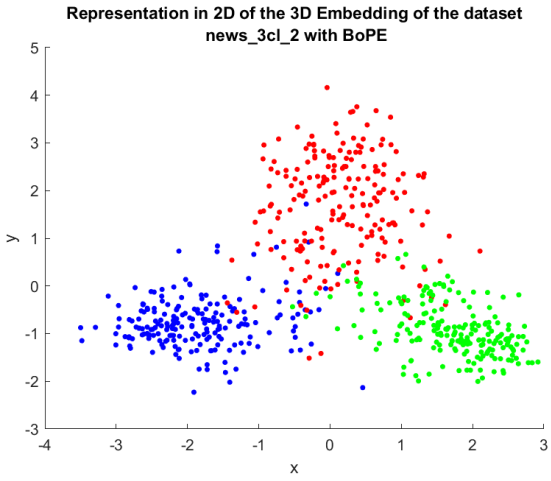

Fig. 7.10.: Embedding computed with BoPE

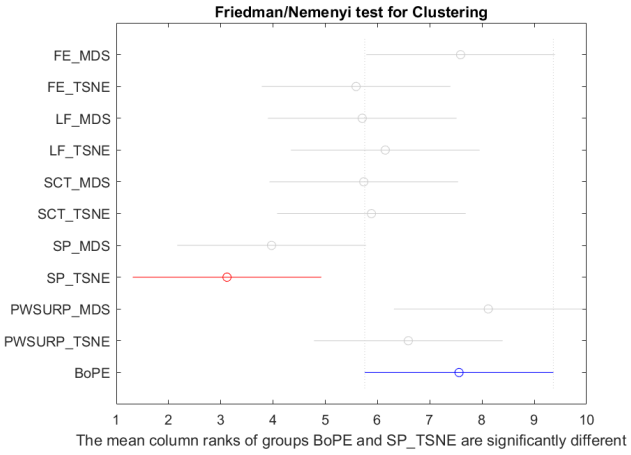

Fig. 7.11.: Nemenyi test for the clustering technique in 3D-space with BoPE

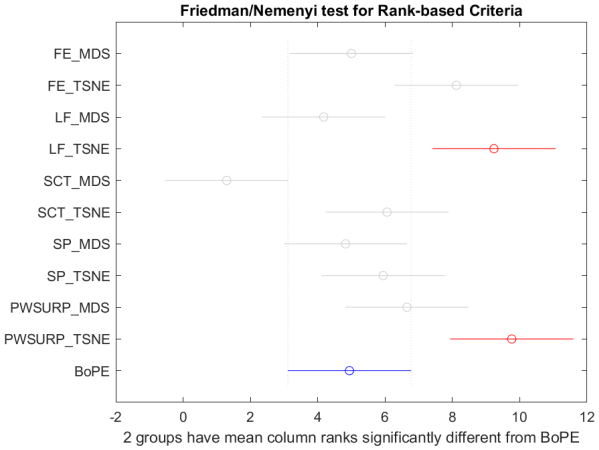

Fig. 7.12.: Nemenyi test for the rank-based criteria technique in 3D-space with BoPE

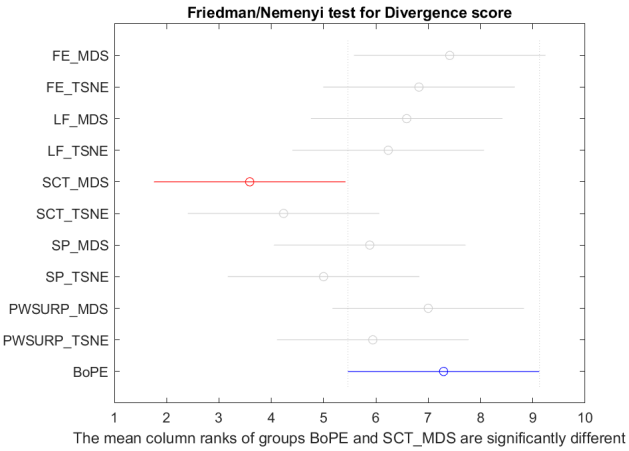

Fig. 7.13.: Nemenyi test for the divergence score technique in 3D-space with BoPE

does not perform well, which is expected since it is computed using MDS. Moreover, its performances are significantly lower than that of the two combinations with the highest mean ranks, namely PWSURP with tSNE and LF with tSNE. Figure 7.13 depicting the Nemenyi test for the divergence score shows that the BoPE performs significantly better than the combination with the worst mean rank, i.e. the SCT with MDS. Looking at the mean ranks of each combination, it can be seen from these three Nemenyi tests that the BoPE and the FE with MDS are very close which can be explained by the way the BoPE is computed. This means that the gradient descent used on the initial embedding does not improve the final embedding representation as was intended when designing the embedding method.

To determine whether the BoPE is significantly different from the other embeddings in a pairwise comparison or not, pairwise Wilcoxon signed-rank tests can be performed (as shown in Table 7.9). As expected, BoPE is not significantly different from FE with MDS for the three evaluation techniques considered. The p-values of the Wilcoxon tests performed on the clustering technique indicate that the BoPE performs significantly better than the SP with tSNE as shown in Figure 7.11, but also from the LF, the SCT and the SP with MDS. Considering the rank-based criteria, the p-values consider all combinations, except for the FE with MDS and the SP with MDS, to perform significantly better or worse than the BoPE, while the divergence score confirms the results found in Figure 7.13.

		FE MDS	FE TSNE	LF MDS	LF TSNE	SCT MDS	SCT TSNE	SP MDS	SP TSNE	PWSURP MDS	PWSURP TSNE
BoPE	*Cl*	0.519	0.093	**0.017**	0.093	**0.030**	0.179	**0.003**	**0.002**	0.787	0.326
	R-b	0.492	**0.004**	**0.013**	**0.004**	$<10^{-4}$	**0.015**	0.055	**0.049**	$<10^{-4}$	**0.002**
	D.s.	0.149	0.687	0.653	0.266	**0.013**	0.113	0.266	0.266	0.586	0.586

Tab. 7.9.: Comparison of the p-values of the Wilcoxon signed-rank tests performed between the bag-of-paths embedding and the other considered embedding in 3D space

In conclusion, in the three-dimensional space, the gradient descent used in the BoPE does not seem to improve the results given by the FE with MDS that do not use gradient descent. However, the use of a gradient descent does not degrade the scores either.

7.3.2 Low-dimensional embedding using 5% of the total dimensionality

As for the three-dimensional space, only the best method by family chosen in Section 7.1 will be kept for the 5% low-dimensional space, using only the best Poisson surprisal distance. The p-values of the Friedman test computed over each evaluation technique are 5.1×10^{-2} for clustering, 7.1×10^{-8} for the rank-based criteria and 1.8×10^{-3} for the divergence score. This means that the null hypothesis cannot be rejected for the clustering technique. However, a Nemenyi test can be produced for the other two assessment techniques. The rank-based criteria test (Figure 7.14) shows that the BoPE performs significantly better than the SCT with MDS and is not significantly worse than the top-ranked distance measure, namely the PSURP. On the other hand, the Nemenyi test for the divergence score does not indicate that the BoPE performs significantly better or worse than any other distance measure, as shown in Figure 7.15. In both Figures, looking at the mean ranks, the best method of the BoP family, i.e. the RSP, gives a higher rank than the BoPE.

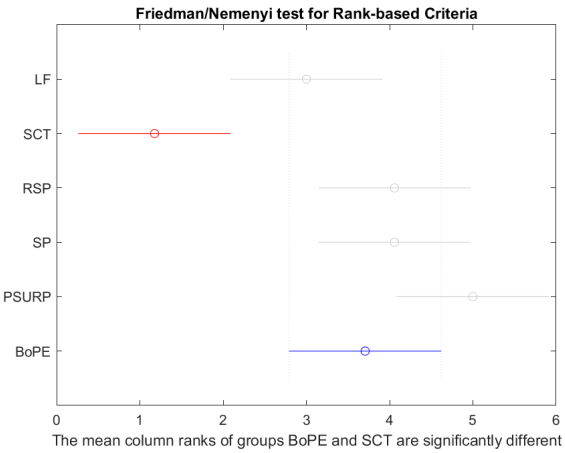

Fig. 7.14.: Nemenyi test for the rank-based criteria technique in low-dimensional space with BoPE

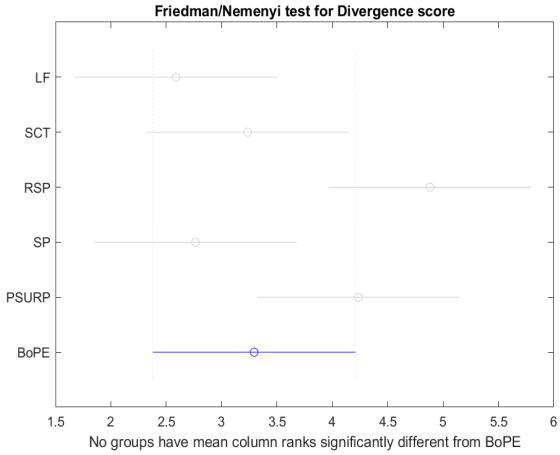

Fig. 7.15.: Nemenyi test for the divergence score technique in low-dimensional space with BoPE

The p-values of the Wilcoxon signed-rank tests between the BoPE and the other embeddings for each technique are presented in Table 7.10. As usual, the rank-based criteria discriminate the scores the most, so that only the SP is not significantly different from the BoPE, which means that the Wilcoxon tests consider most distance measures to perform significantly better or worse than in the corresponding Nemenyi test. The clustering technique finds that the SCT is significantly different from the BoPE and the divergence score finds no significant difference, which coincides with the results in Figure 7.15.

		LF	SCT	RSP	SP	PSURP
BoPE	*Cl*	0.119	**0.025**	0.135	0.084	0.855
	R-b	**0.003**	**$<10^{-4}$**	**0.028**	0.124	**$<10^{-4}$**
	D.s.	0.332	0.906	0.055	0.795	0.102

Tab. 7.10.: Comparison of the p-values of the Wilcoxon signed-rank tests performed between the bag-of-paths embedding and the other considered embedding in low-dimensional space

In conclusion, on the datasets investigated, the gradient descent of the BoPE did not improve the results of the classical BoP family distance with MDS in low-dimensional space. However, it still performs reasonably well compared to the distances from the CT family or the SP.

The following chapter will present a general conclusion of the three research questions investigated in this thesis and some limitations that appeared during the realisation of the experiments.

8 Conclusion

To conclude this thesis, the main results found will be summarised. Then, the limitations of this work will be explained as well as the future work that can be explored on this subject. Finally, the acquired skills will be presented.

8.1 Main Results

The first research question aims at evaluating the performances of the new Poisson surprisal distance and the Poisson weighted surprisal distance by three different embedding evaluation techniques. Two cases are considered, the three-dimensional space where multidimensional scaling and t-Distributed Stochastic Neighbor Embedding are compared and the low-dimensional space using only multidimensional scaling. The main results found for the first case are that the Poisson surprisal distances performed well on the studied datasets compared to other distances used with the same embedding method, MDS or tSNE. However, it can be noticed that the three evaluation techniques gave very different rankings and Wilcoxon p-values which can be explained by the fact that they do not evaluate the preservation of the same properties on the graph. The clustering technique promotes the FE, PWSURP and PSURP with MDS, while the rank-based criteria prefer the LF, PWSURP and PSURP with tSNE and the divergence score favors the FE with MDS or tSNE and PWSURP with MDS. All three techniques allow us to generalize that the BoP and Surprisal families perform better than the CT family and the SP. Focusing on the comparison between the embedding computed with MDS or tSNE, the rank-based criteria showed that the MDS embedding method performs very poorly when considering the properties that the rank-based criteria want to preserve, i.e. the local neighborhoods. For the second case study in the first question, the three assessment techniques find that the PWSURP, PSURP and RSP perform the best.

The second research question aims at evaluating whether computing the partition of the graph into clusters on the embedding of this graph in a low dimensional space, i.e. 5%, produces better or worse results than performing this partition on the kernel of the graph. Experiments have shown that on the datasets used, there is no significant difference between using kernel k-means on the kernel or standard k-means on the embedding when the same distance measure is used to compute the embedding and the kernel. Focusing on the highest-ranked combinations, it has been shown that the EKM computed on the SURP gives the best results for these datasets. However, the KKM and EKM computed on the FE or the Poisson surprisal distances give very similar mean ranks. These are close to the mean rank of the SURP with EKM.

The third research question aims at evaluating the performance of the newly introduced embedding: the bag-of-paths embedding. Contrarily to what was expected, it was observed that on the datasets used the addition of gradient descent on the embedding computed by MDS on the FEM distance did not improve the performances. In three-dimensional space, the mean rank of this new embedding was very close to that of the FE with MDS. Indeed, the distance used by BoPE, namely the FEM, is derived from the FE. In low-dimensional space, the mean rank of the BoPE was lower than the mean rank of the RSP, the best distance measure of the BoP family in this case. In conclusion, as the BoPE did not improve the performance compared to the other distance measures of the BoP family, it is more interesting to use a distance measure like the FE or the RSP.

8.2 Research Limitations and Further Works

This thesis presents some limitations, here we present some of them in a non-exhaustive list

- The number of datasets used is quite limited. The statistical tests would have been more conclusive if the number of datasets used to perform the experiments had been much larger than the number of combinations compared. In addition, the overall performances on all datasets of a distance measure would have been more accurate as they would have been less influenced by a particular dataset. Another drawback of these datasets is that they are all based on social networks and half of them are derived from the Newsgroup database, which means that they do not represent well the real-world graph and the results found may not apply to other sectors, such as the biological or technological sector. Furthermore, these are rather small datasets, the largest dataset having only 1000 nodes and the smallest 34, so the results cannot be generalized to graphs with hundreds of thousands of nodes. However, these problems are not easy to solve because it is difficult to find datasets representing graphs with ground-truth communities.

- Another problem with these datasets is that they all have ground-truth communities, which means that the results found may not apply to datasets that do not have inherent communities.

- The complexity of the introduced Poisson surprisal distance is of the order of $\mathcal{O}(3)$, which means that it does not scale well to large graphs and that a computational improvement must be found to use this distance on large graphs.

- When choosing the dimension of the embedding on which the experiments were computed, it was decided to keep only 5% of the total dimensionality for the embedding in one case and 3 dimensions in the other case. This is a rather important limitation as the results presented depend on this chosen dimensions. The results cannot, therefore, be generalized with certainty without carrying out more experiments by varying the size of the dimensionality kept.

- There are a large number of embedding techniques in the literature as presented in Chapter 1, but only a few have been used in this thesis to compare them to the new distance and embedding studied. The techniques used have shown to perform well in preceding papers such as [59].

- The number of parameters tested for each distance is limited to reduce the running time. This influences the results as other parameters could have given different results.

- Other embedding evaluation techniques and statistical tests exist and could have been used to confirm the results given by the used techniques or to give other results.

As far as future work is concerned, some limitation points seem to be interesting and should be explored further. The first study that could be made is the analysis of the results when applying the new distance measure and the new embedding studied to larger graphs, but also to graphs without inherent communities.

Another point that can be explored is to compare the results obtained in this thesis with the results obtained with other embedding evaluation techniques. For example, the ratio between the intra-class variance and the total variance could be used as an assessment technique. It would also be interesting to compare the best embedding methods found in this thesis with embeddings produced by Matrix factorization based-methods, such as Graph Laplacian eigemaps [2] or HOPE [50], or Deep Learning based-methods, such as DeepWalk [51] or node2vec [24]. This would allow the new investigated Poisson surprisal distance to be compared to other embeddings not based on the BoP framework.

Currently, the best parameters of the divergence score are found by searching the parameters that minimize the local score. An improvement would be to tune the parameters of this assessment embedding technique in an unsupervised way, as for clustering, but taking into account both the global and the local score.

8.3 Acquired Skills

The work on this thesis has allowed us to acquire new skills and deepen some existing ones. The most important knowledge that has been deepened is the graph mining field and the understanding of the scientific process by which experimental results can be assessed. This thesis required working with

different computer languages, namely Matlab, Julia, and Python, and improving the skills we had in the first two as we did not use them frequently before working on this thesis. Writing this work also allows us to improve our English language skills and expand our vocabulary.

Bibliography

[1]A.-L. Barabási. *Network Science*. 1st. Cambridge University Press, 2016.

[2]M. Belkin and P. Niyogi. “Laplacian Eigenmaps for Dimensionality Reduction and Data Representation”. In: *Neural Computation* 15.6 (2003), pp. 1373–1396.

[3]H. Cai, V. W. Zheng, and K. C. Chang. “A Comprehensive Survey of Graph Embedding: Problems, Techniques, and Applications”. In: *IEEE Transactions on Knowledge and Data Engineering* 30.9 (2018), pp. 1616–1637.

[4]Y. Cao and L. Wang. *Automatic Selection of t-SNE Perplexity*. 2017. URL: `https://arxiv.org/abs/1708.03229`.

[5]P. Chebotarev. “A Class of Graph-Geodetic Distances Generalizing the Shortest-Path and the Resistance Distances”. In: *Discrete Applied Mathematics* 159(5) (2011), pp. 295–302.

[6]P. Chebotarev. “The graph bottleneck identity”. In: *Advances in Applied Mathematics* 47.3 (2011), pp. 403–413.

[7]P. Chebotarev and E. Shamis. “The matrix-forest theorem and measuring relations in small social groups”. In: *Automation and Remote Control* 58.9 (1997), pp. 1505–1514.

[8]L. Chen and A. Buja. “Local Multidimensional Scaling for Nonlinear Dimension Reduction, Graph Drawing, and Proximity Analysis”. In: *Journal of the American Statistical Association* 104.485 (2009), pp. 209–219.

[9]F. R. K. Chung and L. Lu. *Complex Graphs and Networks*. 1st. American Mathematical Society, 2006.

[10]S. Courtain. “Community detection in networks by soft modularity maximization : A new approach and empirical comparisons”. MA thesis. Louvain School of Management, 2017.

[11]S. Courtain. “Essays on network data analysis through the bag-of-paths framework”. Publication will be available in September 2022. PhD thesis. Université catholique de Louvain, Belgium, 2022.

[12]S. Courtain and M. Saerens. “A Simple Extension of the Bag-of-Paths Model Weighting Path Lengths by a Poisson Distribution”. In: *Proceedings of the 10th International Conference on Complex Networks and their Applications (CNA’21)*. Springer, 2022, pp. 220–233.

[13]J. Demsar. “Statistical Comparisons of Classifiers over Multiple Data Sets”. In: *Journal of Machine Learning Research* 7 (2006), pp. 1–30.

[14]W. D. Fisher. “On Grouping for Maximum Homogeneity”. In: *Journal of the American Statistical Association* 53.284 (1958), pp. 789–798.

[15]S. Fortunato and D. Hric. “Community detection in networks : A user guide”. In: *Physics Reports* 659 (2016), pp. 1–44.

[16]F. Fouss, M. Saerens, and M. Shimbo. *Algorithms and models for network data and link analysis*. 1st. Cambridge University Press, 2016.

[17]F. Fouss et al. “An experimental investigation of graph kernels on a collaborative recommendation task”. In: *Proceedings of the 6th IEEE International Conference on Data Mining (ICDM’06)*. 2006, pp. 863–868.

[18]F. Fouss et al. “An experimental investigation of kernels on graphs for collaborative recommendation and semisupervised classification”. In: *Neural Networks* 31 (2012), pp. 53–72.

[19]F. Fouss et al. “Random-walk computation of similarities between nodes of a graph, with application to collaborative recommendation”. In: *IEEE Transactions on Knowledge and Data Engineering* 19.3 (2007), pp. 355–369.

[20]K. Françoisse et al. “A bag-of-paths framework for network data analysis”. In: *Neural Networks* 90 (2017), pp. 90–111.

[21]A. L. N. Fred and A. K. Jain. “Robust Data Clustering”. In: vol. 2. 2003.

[22]M. Friedman. “A Comparison of Alternative Tests of Significance for the Problem of m Rankings”. In: *The Annals of Mathematical Statistics* 11(1) (1990), pp. 86–92.

[23]M. Girvan and M. E. J. Newman. “Community structure in social and biological networks”. In: *Proceedings of the National Academy of Sciences of the USA (PNAS’02)*. Vol. 99. 12. 2002, pp. 7821–7826.

[24]A. Grover and J. Leskovec. “Node2vec: Scalable Feature Learning for Networks”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD’16)*. 2016, pp. 855–864.

[25]W. L. Hamilton. *Graph Representation Learning*. 1st. Morgan and Clayppol Publishers, 2020.

[26]L. Hubert and P. Arabie. “Comparing Partitions”. In: *Journal of classification* 2 (1985), pp. 193–218.

[27]V. Ivashkin and P. Chebotarev. “Do Logarithmic Proximity Measures Outperform Plain Ones in Graph Clustering?” In: *Proceedings of the 6th International Conference on Network Analysis (NET’16)*. 2016, pp. 87–105.

[28]B. Kamiński, P. Prałat, and F. Théberge. “A Scalable Unsupervised Framework for Comparing Graph Embeddings”. In: *Proceedings of the 17th International Workshop of Algorithms and Models for the Web Graph (WAW’20)*. Springer International Publishing, 2020, pp. 52–67.

[29]B. Kamiński, P. Prałat, and F. Théberge. “An unsupervised framework for comparing graph embeddings”. In: *Journal of Complex Networks* 8.5 (2019).

[30]B. Kamiński et al. *A Multi-purposed Unsupervised Framework for Comparing Embeddings of Undirected and Directed Graphs*. 2021. URL: `https://arxiv.org/abs/2112.00075`.

[31]T. N. Kipf and M. Welling. “Semi-supervised classification with graph convolutional networks”. In: *International conference on learning representations* (2017).

[32]I. Kivimäki, M. Shimbo, and M. Saerens. “Developments in the theory of randomized shortest paths with a comparison of graph node distances”. In: *Physica A: Statistical Mechanics and its Applications* 393 (2014), pp. 600–616.

[33]M. Klimenta and U. Brandes. “Graph drawing by classical multidimensional scaling: New perspectives”. In: *Graph drawing*. Vol. 7704. Lecture Notes in Computer Science. Springer, 2013, pp. 55–66.

[34]A. Lancichinetti, S. Fortunato, and F. Radicchi. “Benchmark graphs for testing community detection algorithms”. In: *Physical Review E* 78(4):046110 (2008).

[35]K. Lang. “NewsWeeder: Learning to Filter Netnews”. In: *Proceedings of the 12th International Conference on Machine Learning (ICML’95)*. 1995, pp. 331–339.

[36]J. A. Lee and M. Verleysen. “Quality assessment of dimensionality reduction: Rank-based criteria”. In: *Neurocomputing* 72.7 (2009), pp. 1431–1443.

[37]J. A. Lee and M. Verleysen. “Quality assessment of nonlinear dimensionality reduction based on K-ary neighborhoods”. In: *Proceedings of the Workshop on New Challenges for Feature Selection in Data Mining and Knowledge Discovery (FSDM’08)*. Vol. 4. 2008, pp. 21–35.

[38]J. A. Lee and M. Verleysen. “Scale-independent quality criteria for dimensionality reduction”. In: *Pattern Recognition Letters* 31.14 (2010), pp. 2248–2257.

[39]P. Leleux et al. “Sparse randomized shortest paths routing with Tsallis divergence regularization”. In: *Data Mining and Knowledge Discovery* 35.3 (2021), pp. 986–1031.

[40]B. Li and D. Pi. “Network representation learning: a systematic literature review”. In: *Neural Computing and Applications* 32 (2020), pp. 16647–16679.

[41]A. Likas, N. Vlassis, and J. J. Verbeek. “The global k-means clustering algorithm”. In: *Pattern Recognition* 36.2 (2003), pp. 451–461.

[42]J. Lin. “Divergence measures based on the Shannon entropy”. In: *IEEE Transactions on Information Theory* 37.1 (1991), pp. 145–151.

[43]D. Lusseau. “The emergent properties of a dolphin social network”. In: *Proceeding of the Royal Society of London B. (Proc. Royal Soc. B’03)*. Vol. 270. 2003, pp. 186–188.

[44]D. Lusseau et al. “The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations”. In: *Behavioral Ecology and Sociobiology* 54 (2003), pp. 396–405.

[45]J. MacQueen. “Some methods for classification and analysis of multivariate observations”. In: *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics (BSMSP’67)*. Vol. 5. 1. 1967, pp. 281–297.

[46]K. V. Mardia, J. T. Kent, and J. M. Bibby. *Multivariate analysis*. Academic Press, 1979.

[47]A. Mead. “Review of the Development of Multidimensional Scaling Methods”. In: *Journal of the Royal Statistical Society. Series D (The Statistician)* 41.1 (1992), pp. 27–39.

[48]P. B. Nemenyi. “Distribution-free Multiple Comparisons”. PhD thesis. Princeton University, United States, 1963.

[49]M. E. J. Newman. *Networks : An Introduction*. 2nd. Oxford University Press, 2018.

[50]M. Ou et al. “Asymmetric Transitivity Preserving Graph Embedding”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD’16)*. 2016, pp. 1105–1114.

[51]B. Perozzi, R. Al-Rfou, and S. Skiena. “DeepWalk: Online Learning of Social Representations”. In: *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD’14)*. 2014, pp. 701–710.

[52]W. M. Rand. “Objective Criteria for the Evaluation of Clustering Methods”. In: *Journal of the American Statistical Association* 66(336) (1971), pp. 846–850.

[53]M. Saerens et al. “The Principal Components Analysis of a Graph, and its Relationships to Spectral Clustering”. In: *Proceedings of the 15th European Conference on Machine Learning (ECML’04)*. Vol. 3201. Lecture Notes in Artificial Intelligence. Springer, 2004, pp. 371–383.

[54]A. M. Safi. “Graph embedding with application to semi-supervised classification, visualization, reconstruction and neighborhood preservation tasks: an experimental comparison”. MA thesis. Ecole polytechnique de Louvain, 2019.

[55]J. M. Santos and M. Embrechts. “On the Use of the Adjusted Rand Index as a Metric for Evaluating Supervised Classification”. In: *Proceedings of the 19th International Conference on Artificial Neural Networks (ICANN’09)*. Springer. 2009, pp. 175–184.

[56]F. Scarselli et al. “The Graph Neural Network Model”. In: *IEEE Transactions on Neural Networks* 20.1 (2009), pp. 61–80.

[57]B. Schölkopf and A. Smola. *Learning with Kernels*. 1st. MIT Press, 2002.

[58]R. Sedgewick and K. Wayne. *Algorithms*. 4th. Addison-Wesley, 2011.

[59]F. Sommer, F. Fouss, and M. Saerens. “Comparison of graph node distances on clustering tasks”. In: *Proceedings of the 25th International Conference on Artificial Neural Networks (ICANN’16)*. Vol. 9886. Springer. 2016, pp. 192–201.

[60]D. Steinley. “Properties of the Hubert–Arabie Adjusted Rand Index”. In: *Psychological Methods* 9(3) (2004), pp. 386–396.

[61]L. van der Maaten. “Learning a Parametric Embedding by Preserving Local Structure”. In: *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS’09)*. Vol. 5. Proceedings of Machine Learning Research. 2009, pp. 384–391.

[62]L. van der Maaten and G. Hinton. “Visualizing Data using t-SNE”. In: *Journal of Machine Learning Research* 9 (2008), pp. 2579–2605.

[63]M. Van Erp and S. Lambert. “Variants of the Borda count method for combining ranked classifier hypotheses”. In: *Proceedings of the 7th International Workshop on frontiers in handwriting recognition (IWFHR’07)*. International Unipen Foundation, 2000, pp. 443–452.

[64]U. von Luxburg, A. Radl, and M. Hein. “Getting lost in space: large sample analysis of the commute distance”. In: *Proceedings of the 23rd Annual Conference on Neural Information Processing Systems (NIPS’10)*. MIT Press, 2010, pp. 2622–2630.

[65]U. von Luxburg, A. Radl, and M. Hein. “Hitting and commute times in large random neighborhood graphs”. In: *Journal of Machine Learning Research* 15 (2014), pp. 1751–1798.

[66]F. Wilcoxon. “Individual Comparisons by Ranking Methods”. In: *Biometrics Bulletin* 1(6) (1945), pp. 80–83.

[67]X. Xu et al. “SCAN: a structural clustering algorithm for networks”. In: *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD’07)*. 2007, pp. 824–833.

[68]L. Yen et al. “A family of dissimilarity measures between nodes generalizing both the shortest-path and the commute-time distances”. In: *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD’08)*. 2008, pp. 785–793.

[69]L. Yen et al. “Graph nodes clustering based on the commute-time kernel”. In: *Proceedings of the 11th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD’07)*. Vol. 4426. Lecture Notes in Artificial Intelligence. Springer, 2007, pp. 1037–1045.

[70]L. Yen et al. “Graph nodes clustering with the sigmoid commute-time kernel: A comparative study”. In: *Data & Knowledge Engineering* 68(3) (2009), pp. 338–361.

[71]W. W. Zachary. “An information flow model for conflict and fission in small groups”. In: *Journal of Anthropological Research* 33.4 (1977), pp. 452–473.

Appendix: First Research Question

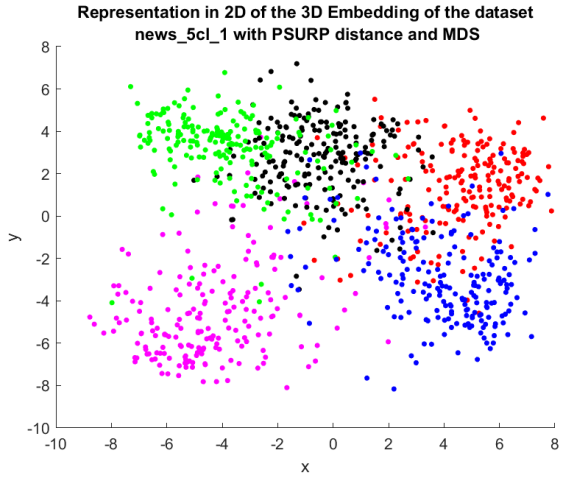

Fig. A.1.: Embedding computed with MDS: partial view 1

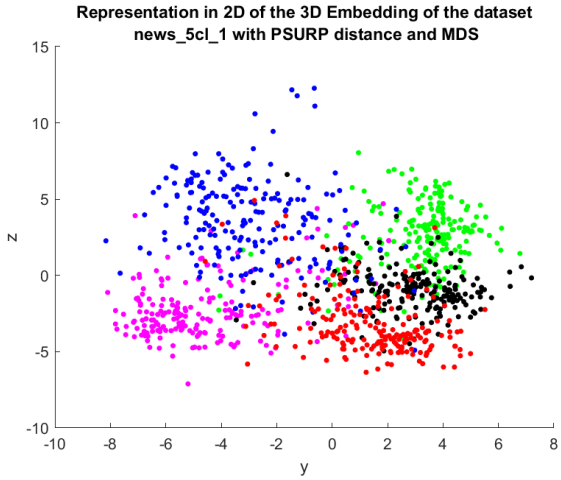

Fig. A.2.: Embedding computed with MDS: partial view 2

As for Figure 7.1, the embeddings computed with MDS method represent nodes close from each other. The clusters are distinguishable, but their proximity produces some mistakes as the boundaries of each cluster are not clearly defined. The embedding computed with tSNE method, as for Figure 7.2, represents the nodes as if they were all on the border of a sphere. As the number of clusters is higher than in Figure 7.2, the arrangement of clusters around the sphere is tighter and more nodes are too close to another cluster. The main point to emphasise is that the arrangement of the clusters is very different depending on the embedding method used, but both allow most of the nodes in the clusters to be distinguished.

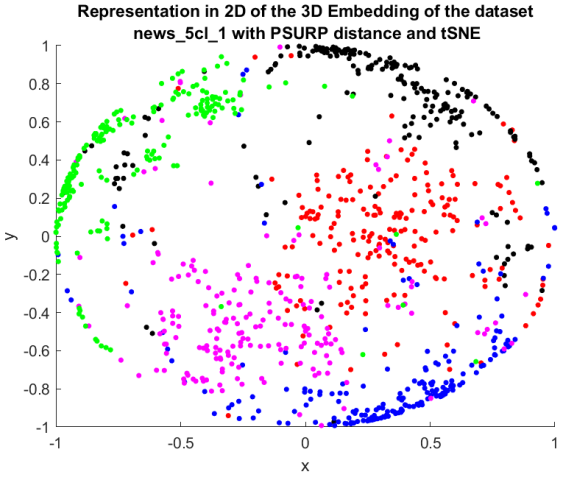

Fig. A.3.: Embedding computed with tSNE: partial view 1

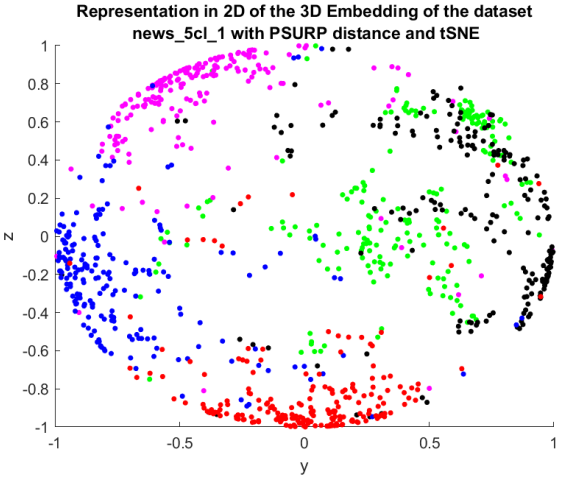

Fig. A.4.: Embedding computed with tSNE: partial view 2

	FE	SURP	LF	SCCT	SCT	RSP	SP	CT	PWSURP	PSURP	FEM
dolphins_2	0.872	0.872	0.935	0.872	0.872	0.812	0.935	0.935	0.872	0.872	0.872
dolphins_4	0.867	0.826	0.847	0.557	0.571	0.890	0.704	0.711	0.839	0.839	0.855
football	0.776	0.790	0.785	0.481	0.520	0.779	0.470	0.791	0.793	0.792	0.779
LFR1	0.990	0.990	0.990	0.990	0.990	0.990	0.851	0.005	0.990	0.990	0.990
LFR2	0.891	0.897	0.880	0.988	0.996	0.893	0.760	0.669	0.927	0.927	0.885
LFR3	0.902	0.933	0.858	0.855	0.853	0.844	0.753	0.455	0.898	0.898	0.904
news_2cl_1	0.902	0.912	0.902	0.883	0.902	0.921	0.739	$<10^{-4}$	0.912	0.912	0.912
news_2cl_2	0.662	0.678	0.662	0.662	0.687	0.678	0.614	$<10^{-4}$	0.704	0.695	0.662
news_2cl_3	0.864	0.874	0.836	0.809	0.818	0.827	0.912	$<10^{-4}$	0.902	0.902	0.864
news_3cl_1	0.846	0.846	0.818	0.832	0.846	0.841	0.792	$<10^{-4}$	0.837	0.837	0.846
news_3cl_2	0.830	0.828	0.787	0.748	0.803	0.834	0.681	$<10^{-4}$	0.825	0.815	0.829
news_3cl_3	0.827	0.827	0.731	0.777	0.781	0.818	0.722	$<10^{-4}$	0.813	0.814	0.818
news_5cl_1	0.654	0.644	0.491	0.503	0.497	0.678	0.664	0.001	0.691	0.689	0.646
news_5cl_2	0.558	0.521	0.472	0.379	0.373	0.500	0.504	$<10^{-4}$	0.512	0.521	0.556
news_5cl_3	0.493	0.486	0.466	0.381	0.449	0.477	0.453	$<10^{-4}$	0.479	0.481	0.495
polbooks	0.650	0.665	0.650	0.671	0.682	0.688	0.642	0.708	0.650	0.650	0.650
zachary	1.000	1.000	1.000	1.000	1.000	1.000	0.882	0.882	1.000	1.000	1.000

Tab. A.1.: Results of ARI with MDS in 3 dimensional space for each distance (best parameters) and dataset

	FE	SURP	LF	SCCT	SCT	RSP	SP	CT	PWSURP	PSURP	FEM
dolphins_2	0.644	0.644	0.593	0.812	0.812	0.543	0.754	0.013	0.543	0.543	0.644
dolphins_4	0.781	0.754	0.800	0.720	0.751	0.784	0.749	0.059	0.800	0.800	0.821
football	0.844	0.826	0.834	0.768	0.761	0.855	0.840	-0.007	0.847	0.857	0.853
LFR1	0.956	0.931	0.850	0.504	0.710	0.946	0.002	0.400	0.990	0.980	0.897
LFR2	1.000	1.000	1.000	1.000	1.000	1.000	$<10^{-4}$	0.196	1.000	1.000	1.000
LFR3	0.745	0.725	0.739	0.978	0.986	0.788	0.003	0.517	0.908	0.734	0.720
news_2cl_1	0.828	0.792	0.765	0.846	0.874	0.852	0.672	0.001	0.865	0.855	0.801
news_2cl_2	0.746	0.755	0.755	0.670	0.704	0.809	0.606	$<10^{-4}$	0.746	0.791	0.818
news_2cl_3	0.864	0.874	0.883	0.846	0.883	0.855	0.855	0.004	0.883	0.874	0.846
news_3cl_1	0.774	0.783	0.819	0.752	0.778	0.787	0.788	0.003	0.806	0.829	0.797
news_3cl_2	0.745	0.732	0.785	0.692	0.746	0.700	0.672	0.014	0.733	0.739	0.716
news_3cl_3	0.736	0.727	0.771	0.688	0.717	0.715	0.688	$<10^{-4}$	0.740	0.727	0.773
news_5cl_1	0.695	0.707	0.685	0.636	0.685	0.651	0.682	0.016	0.695	0.691	0.655
news_5cl_2	0.486	0.595	0.507	0.417	0.501	0.532	0.469	0.011	0.523	0.483	0.551
news_5cl_3	0.463	0.438	0.472	0.435	0.446	0.451	0.439	0.037	0.433	0.425	0.487
polbooks	0.665	0.680	0.665	0.685	0.674	0.653	0.688	0.046	0.652	0.652	0.653
zachary	0.882	0.882	0.882	1.000	1.000	1.000	0.882	0.044	0.882	0.882	0.882

Tab. A.2.: Results of ARI with tSNE in 3 dimensional space for each distance (best parameters) and dataset

	FE	SURP	LF	SCCT	SCT	RSP	SP	CT	PWSURP	PSURP	FEM
dolphins_2	0.472	0.474	0.479	0.257	0.267	0.469	0.421	0.420	0.480	0.474	0.462
dolphins_4	0.472	0.474	0.479	0.257	0.267	0.469	0.421	0.420	0.480	0.474	0.462
football	0.318	0.320	0.317	0.248	0.250	0.319	0.270	0.311	0.323	0.322	0.318
LFR1	0.076	0.074	0.076	0.038	0.040	0.073	0.071	0.047	0.077	0.076	0.077
LFR2	0.133	0.131	0.132	0.126	0.123	0.131	0.118	0.123	0.136	0.136	0.132
LFR3	0.124	0.123	0.122	0.108	0.104	0.116	0.104	0.106	0.122	0.122	0.120
news_2cl_1	0.230	0.227	0.195	0.156	0.162	0.249	0.287	0.129	0.256	0.297	0.230
news_2cl_2	0.241	0.238	0.170	0.126	0.141	0.273	0.291	0.127	0.267	0.306	0.241
news_2cl_3	0.267	0.264	0.213	0.182	0.187	0.298	0.340	0.174	0.292	0.342	0.267
news_3cl_1	0.218	0.216	0.194	0.164	0.169	0.241	0.283	0.124	0.242	0.290	0.218
news_3cl_2	0.189	0.189	0.157	0.113	0.117	0.210	0.251	0.094	0.219	0.267	0.190
news_3cl_3	0.214	0.212	0.175	0.147	0.151	0.245	0.282	0.118	0.240	0.274	0.214
news_5cl_1	0.189	0.184	0.159	0.133	0.136	0.210	0.241	0.088	0.205	0.245	0.190
news_5cl_2	0.173	0.170	0.152	0.119	0.126	0.193	0.235	0.082	0.190	0.237	0.174
news_5cl_3	0.145	0.142	0.120	0.098	0.099	0.163	0.197	0.070	0.163	0.196	0.144
polbooks	0.387	0.401	0.392	0.299	0.296	0.390	0.365	0.371	0.404	0.402	0.386
zachary	0.385	0.388	0.359	0.303	0.283	0.417	0.349	0.247	0.406	0.405	0.383

Tab. A.3.: Results of rank-based criteria with MDS in 3 dimensional space for each distance (best parameters) and dataset

	FE	SURP	LF	SCCT	SCT	RSP	SP	CT	PWSURP	PSURP	FEM
dolphins_2	0.417	0.425	0.416	0.357	0.362	0.421	0.375	-0.002	0.425	0.421	0.419
dolphins_4	0.424	0.423	0.416	0.358	0.376	0.421	0.383	-0.002	0.430	0.426	0.411
football	0.345	0.335	0.344	0.309	0.309	0.345	0.343	-0.010	0.346	0.350	0.339
LFR1	0.095	0.091	0.095	0.082	0.086	0.095	0.001	0.022	0.096	0.086	0.095
LFR2	0.180	0.181	0.180	0.177	0.181	0.181	0.001	0.038	0.181	0.182	0.179
LFR3	0.142	0.141	0.140	0.136	0.136	0.139	0.003	0.054	0.146	0.133	0.141
news_2cl_1	0.455	0.450	0.460	0.333	0.375	0.457	0.459	0.007	0.458	0.458	0.444
news_2cl_2	0.441	0.451	0.449	0.304	0.342	0.448	0.448	0.004	0.454	0.445	0.433
news_2cl_3	0.474	0.472	0.479	0.344	0.372	0.482	0.470	0.021	0.477	0.464	0.481
news_3cl_1	0.456	0.451	0.463	0.331	0.371	0.454	0.455	0.011	0.459	0.451	0.454
news_3cl_2	0.434	0.435	0.441	0.287	0.336	0.435	0.431	0.015	0.438	0.433	0.432
news_3cl_3	0.455	0.451	0.471	0.331	0.387	0.448	0.453	0.013	0.459	0.458	0.446
news_5cl_1	0.424	0.420	0.433	0.304	0.330	0.415	0.428	0.010	0.425	0.421	0.418
news_5cl_2	0.413	0.417	0.427	0.277	0.316	0.419	0.416	0.010	0.418	0.415	0.410
news_5cl_3	0.382	0.384	0.392	0.251	0.288	0.386	0.388	0.017	0.392	0.385	0.385
polbooks	0.361	0.374	0.366	0.350	0.348	0.359	0.356	$<10^{-4}$	0.368	0.368	0.365
zachary	0.426	0.436	0.428	0.365	0.370	0.421	0.311	0.004	0.443	0.442	0.429

Tab. A.4.: Results of rank-based criteria with tSNE in 3 dimensional space for each distance (best parameters) and dataset

	FE	SURP	LF	SCCT	SCT	RSP	SP	CT	PWSURP	PSURP	FEM
dolphins_2	1301.783	1.024	1400.488	49191.47	9354.770	2233.768	49890.53	145168.160	4264.176	3436.151	1352.858
dolphins_4	1.338	1.355	2.159	59.222	59.861	1.014	4.543	51.218	1.315	1.407	1.468
football	1.226	1.205	1.210	2.475	2.529	1.274	1.403	1.147	1.205	1.206	1.224
LFR1	2.727	8.698	9.515	11.743	48.699	12.108	7.471	12519.344	12.272	12.465	2.210
LFR2	4.298	4.037	3.837	7.766	5.012	3.952	3.837	6.263	3.856	3.849	4.014
LFR3	4.539	4.687	4.703	6.381	5.836	4.643	5.822	11.461	4.668	4.900	4.399
news_2cl_1	3224.587	9311.759	1191.107	5451.524	14622.80	5820.078	5586.205	12140396.00	1703.655	1800.319	6516.349
news_2cl_2	70.158	301.910	592.193	28.426	775.016	538.577	265.420	1690782.700	295.756	338.925	84.119
news_2cl_3	609.911	1853.065	565.125	3497.291	54.137	779.774	96.344	2331187.100	4.974	5.078	2344.640
news_3cl_1	9.011	13.948	16.151	16.690	15.047	1.028	16.905	10568.365	5.522	5.502	4.049
news_3cl_2	1.040	1.315	1.030	4.457	4.656	1.482	1.505	1841.982	2.974	2.973	2.604
news_3cl_3	2.298	1.406	2.236	4.487	4.376	1.026	6.435	565.949	1.600	1.599	3.682
news_5cl_1	1.623	1.625	2.159	3.300	3.444	1.680	1.521	11.622	1.924	1.924	1.596
news_5cl_2	1.292	1.298	4.295	4.439	4.379	1.359	1.689	16.605	3.323	3.297	1.251
news_5cl_3	1.262	1.325	1.080	2.291	2.326	1.368	1.508	10.413	1.157	1.162	1.256
polbooks	9.024	9.028	12.129	1.726	1.686	10.014	11.293	17.014	9.702	9.706	9.114
zachary	31.917	47.581	17.019	224.048	527.020	39.432	4.344	958.573	38.569	38.637	50.952

Tab. A.5.: Results of divergence score with MDS in 3 dimensional space for each distance (best parameters) and dataset

	FE	SURP	LF	SCCT	SCT	RSP	SP	CT	PWSURP	PSURP	FEM
dolphins_2	11069.003	204.753	13956.453	4848.512	12723.188	90745.405	94380.431	1118307.700	14421.424	13299.236	22740.336
dolphins_4	6.566	5.125	7.255	10.747	9.512	4.053	4.442	32.346	5.150	5.157	5.529
football	1.133	1.127	1.143	1.591	1.502	1.162	1.167	6.657	1.130	1.281	1.399
LFR1	48.163	24.783	7.240	15.353	57.181	42.745	1302.093	15.612	66.291	30.696	16.385
LFR2	2.622	1.385	2.117	2.380	2.623	2.119	7.020	5.471	2.178	2.175	2.079
LFR3	2.915	1.614	1.228	4.199	4.899	3.156	7.477	3.835	3.107	2.902	1.484
news_2cl_1	900.396	1798.695	8317.846	3688.938	52.466	2464.324	3170.779	2648284.700	92.436	10528.731	1.794
news_2cl_2	4.598	28.042	116.344	419.876	600.461	47.947	332.375	344161.100	2.078	325.181	8.906
news_2cl_3	45.787	282.909	5.670	899.887	2360.794	22.460	2.595	537024.830	570.327	5795.504	762.960
news_3cl_1	22.428	47.198	6.949	9.076	75.070	19.447	16.972	3456.715	25.034	56.174	30.743
news_3cl_2	5.430	18.459	6.994	81.228	5.025	2.423	5.266	229.949	47.751	10.682	2.944
news_3cl_3	22.246	15.024	17.964	27.221	52.503	14.552	17.166	142.563	12.243	16.165	28.067
news_5cl_1	1.427	1.837	1.995	2.169	2.175	1.771	1.829	4.599	1.806	1.696	1.618
news_5cl_2	1.605	1.658	2.876	1.415	1.377	1.708	1.807	6.683	2.045	2.371	1.737
news_5cl_3	1.453	1.277	1.622	1.406	1.851	1.826	1.524	3.445	1.612	1.632	1.450
polbooks	2.123	1.878	2.023	2.545	2.810	2.010	2.575	155.593	1.723	1.982	1.453
zachary	65.229	52.529	28.040	82.100	26.079	111.424	20.765	13.279	122.067	6.008	18.616

Tab. A.6.: Results of divergence score with tSNE in 3 dimensional space for each distance (best parameters) and dataset

	FE MDS	FE TSNE	LF MDS	LF TSNE	SCT MDS	SCT TSNE	SP MDS	SP TSNE	PWSURP MDS	PWSURP TSNE	PSURP MDS	PSURP TSNE
FE MDS	/	0.063	0.021	0.084	0.035	0.163	0.003	0.002	0.670	0.301	0.635	0.149
FE TSNE	0.063	/	0.619	0.855	0.943	0.877	0.070	0.015	0.025	0.173	0.028	0.639
LF MDS	0.021	0.619	/	0.906	0.796	0.918	0.034	0.019	0.008	0.605	0.011	0.795
LF TSNE	0.084	0.855	0.906	/	0.943	0.796	0.039	0.010	0.055	0.952	0.055	0.715
SCT MDS	0.035	0.943	0.796	0.943	/	0.535	0.332	0.177	0.018	0.619	0.022	0.981
SCT TSNE	0.163	0.877	0.918	0.796	0.535	/	0.102	0.007	0.056	0.717	0.070	0.796
SP MDS	0.003	0.070	0.034	0.039	0.332	0.102	/	0.088	0.002	0.034	0.001	0.109
SP TSNE	0.002	0.015	0.019	0.010	0.177	0.007	0.088	/	0.001	0.013	0.001	0.017
PWSURP MDS	0.670	0.025	0.008	0.055	0.018	0.056	0.002	0.001	/	0.177	0.641	0.062
PWSURP TSNE	0.301	0.173	0.605	0.952	0.619	0.717	0.034	0.013	0.177	/	0.196	0.424
PSURP MDS	0.635	0.028	0.011	0.055	0.022	0.070	0.001	0.001	0.641	0.196	/	0.068
PSURP TSNE	0.149	0.639	0.795	0.715	0.981	0.796	0.109	0.017	0.062	0.424	0.068	/

Tab. A.7.: Comparison of the p-values of the Wilcoxon signed-rank tests performed between the (distance measure, embedding method) combinations in 3D space for the clustering

	FE MDS	FE TSNE	LF MDS	LF TSNE	SCT MDS	SCT TSNE	SP MDS	SP TSNE	PWSURP MDS	PWSURP TSNE	PSURP MDS	PSURP TSNE
FE MDS	/	0.006	0.006	0.006	$<10^{-4}$	0.019	0.068	0.049	$<10^{-4}$	0.002	$<10^{-4}$	0.004
FE TSNE	0.006	/	0.005	0.017	$<10^{-4}$	$<10^{-4}$	0.001	0.084	0.007	$<10^{-4}$	0.007	0.619
LF MDS	0.006	0.005	/	0.004	$<10^{-4}$	0.009	0.055	0.049	$<10^{-4}$	0.003	0.002	0.004
LF TSNE	0.006	0.017	0.004	/	$<10^{-4}$	$<10^{-4}$	0.001	$<10^{-4}$	0.007	0.554	0.007	0.113
SCT MDS	$<10^{-4}$	$<10^{-4}$	$<10^{-4}$	$<10^{-4}$	/	$<10^{-4}$	0.001	0.004	$<10^{-4}$	$<10^{-4}$	$<10^{-4}$	$<10^{-4}$
SCT TSNE	0.019	$<10^{-4}$	0.009	$<10^{-4}$	$<10^{-4}$	/	0.006	0.163	0.044	$<10^{-4}$	0.163	0.001
SP MDS	0.068	0.001	0.055	0.001	0.001	0.006	/	0.028	0.981	$<10^{-4}$	0.001	$<10^{-4}$
SP TSNE	0.049	0.084	0.049	$<10^{-4}$	0.004	0.163	0.028	/	0.049	0.001	0.055	0.055
PWSURP MDS	$<10^{-4}$	0.007	$<10^{-4}$	0.007	$<10^{-4}$	0.044	0.981	0.049	/	0.006	0.049	0.006
PWSURP TSNE	0.002	$<10^{-4}$	0.003	0.554	$<10^{-4}$	$<10^{-4}$	$<10^{-4}$	0.001	0.006	/	0.006	0.002
PSURP MDS	$<10^{-4}$	0.007	0.002	0.007	$<10^{-4}$	0.163	0.001	0.055	0.049	0.006	/	0.006
PSURP TSNE	0.004	0.619	0.004	0.113	$<10^{-4}$	0.001	$<10^{-4}$	0.055	0.006	0.002	0.006	/

Tab. A.8.: Comparison of the p-values of the Wilcoxon signed-rank tests performed between the (distance measure, embedding method) combinations in 3D space for the rank-based criteria

	FE MDS	FE TSNE	LF MDS	LF TSNE	SCT MDS	SCT TSNE	SP MDS	SP TSNE	PWSURP MDS	PWSURP TSNE	PSURP MDS	PSURP TSNE
FE MDS	/	0.723	0.463	0.287	0.013	0.076	0.044	0.193	0.492	0.463	0.435	0.028
FE TSNE	0.723	/	0.906	0.943	0.102	0.044	0.435	0.332	0.407	0.356	0.407	0.227
LF MDS	0.463	0.906	/	0.943	0.019	0.062	0.906	0.163	0.795	0.435	0.943	0.210
LF TSNE	0.287	0.943	0.943	/	0.039	0.266	0.653	0.687	0.723	0.619	0.723	0.332
SCT MDS	0.013	0.102	0.019	0.039	/	0.687	0.227	0.653	0.001	0.723	0.001	0.463
SCT TSNE	0.076	0.044	0.062	0.266	0.687	/	0.309	0.758	0.124	0.619	0.136	0.523
SP MDS	0.044	0.435	0.906	0.653	0.227	0.309	/	0.177	0.332	0.831	0.356	0.102
SP TSNE	0.193	0.332	0.163	0.687	0.653	0.758	0.177	/	0.055	0.407	0.177	0.653
PWSURP MDS	0.492	0.407	0.795	0.723	0.001	0.124	0.332	0.055	/	0.332	0.193	0.076
PWSURP TSNE	0.463	0.356	0.435	0.619	0.723	0.619	0.831	0.407	0.332	/	0.332	0.492
PSURP MDS	0.435	0.407	0.943	0.723	0.001	0.136	0.356	0.177	0.193	0.332	/	0.193
PSURP TSNE	0.028	0.227	0.210	0.332	0.463	0.523	0.102	0.653	0.076	0.492	0.193	/

Tab. A.9.: Comparison of the p-values of the Wilcoxon signed-rank tests performed between the (distance measure, embedding method) combinations in 3D space for the divergence score

	FE	SURP	LF	SCCT	SCT	RSP	SP	CT	PWSURP	PSURP	FEM
dolphins_2	0.872	0.935	0.935	0.872	0.872	0.812	0.935	0.935	0.872	0.872	0.872
dolphins_4	0.791	0.752	0.765	0.750	0.753	0.786	0.843	0.614	0.755	0.755	0.781
football	0.862	0.889	0.889	0.798	0.798	0.877	0.871	0.853	0.889	0.881	0.864
LFR1	0.990	0.990	0.990	0.990	0.990	0.990	0.937	$<10^{-4}$	0.990	0.990	0.990
LFR2	1.000	1.000	1.000	1.000	1.000	1.000	0.988	0.921	1.000	1.000	1.000
LFR3	1.000	1.000	1.000	1.000	1.000	1.000	0.995	0.938	1.000	1.000	1.000
news_2cl_1	0.902	0.912	0.893	0.886	0.912	0.902	0.764	0.819	0.904	0.902	0.902
news_2cl_2	0.737	0.759	0.768	0.695	0.695	0.742	0.625	0.687	0.772	0.772	0.788
news_2cl_3	0.883	0.883	0.836	0.844	0.843	0.855	0.912	0.786	0.902	0.892	0.883
news_3cl_1	0.833	0.828	0.814	0.809	0.819	0.846	0.804	0.618	0.824	0.827	0.833
news_3cl_2	0.824	0.825	0.818	0.781	0.809	0.821	0.698	0.458	0.817	0.817	0.811
news_3cl_3	0.836	0.841	0.760	0.812	0.827	0.828	0.778	0.423	0.836	0.835	0.829
news_5cl_1	0.709	0.709	0.636	0.648	0.669	0.693	0.715	0.057	0.711	0.714	0.707
news_5cl_2	0.597	0.603	0.553	0.485	0.533	0.579	0.551	0.290	0.604	0.601	0.597
news_5cl_3	0.640	0.635	0.592	0.493	0.564	0.617	0.449	0.390	0.622	0.632	0.638
polbooks	0.650	0.655	0.650	0.669	0.669	0.630	0.655	0.665	0.596	0.599	0.650
zachary	0.988	1.000	1.000	1.000	1.000	1.000	0.965	0.882	1.000	1.000	1.000

Tab. A.10.: Results of ARI with MDS in low-dimensional space for each distance (best parameters) and dataset

	FE	SURP	LF	SCCT	SCT	RSP	SP	CT	PWSURP	PSURP	FEM
dolphins_2	0.626	0.612	0.634	0.437	0.417	0.598	0.544	0.495	0.626	0.621	0.617
dolphins_4	0.626	0.612	0.634	0.437	0.417	0.598	0.544	0.495	0.626	0.621	0.617
football	0.423	0.424	0.424	0.367	0.359	0.425	0.417	0.367	0.425	0.419	0.423
LFR1	0.215	0.204	0.206	0.125	0.144	0.212	0.212	0.106	0.243	0.241	0.242
LFR2	0.306	0.296	0.299	0.260	0.272	0.289	0.209	0.212	0.315	0.315	0.312
LFR3	0.256	0.234	0.247	0.182	0.199	0.239	0.196	0.163	0.284	0.284	0.270
news_2cl_1	0.620	0.610	0.358	0.322	0.354	0.665	0.708	0.268	0.659	0.669	0.621
news_2cl_2	0.620	0.613	0.357	0.293	0.342	0.678	0.698	0.253	0.665	0.681	0.620
news_2cl_3	0.619	0.611	0.367	0.344	0.369	0.660	0.709	0.308	0.660	0.674	0.620
news_3cl_1	0.627	0.622	0.367	0.335	0.364	0.678	0.725	0.284	0.668	0.677	0.628
news_3cl_2	0.616	0.609	0.339	0.295	0.333	0.677	0.725	0.238	0.659	0.675	0.616
news_3cl_3	0.652	0.645	0.377	0.344	0.373	0.702	0.732	0.263	0.689	0.694	0.652
news_5cl_1	0.615	0.610	0.345	0.315	0.341	0.659	0.710	0.239	0.654	0.663	0.615
news_5cl_2	0.624	0.620	0.339	0.285	0.322	0.670	0.713	0.246	0.663	0.671	0.624
news_5cl_3	0.604	0.598	0.326	0.269	0.302	0.659	0.704	0.218	0.649	0.660	0.604
polbooks	0.479	0.480	0.474	0.345	0.340	0.473	0.442	0.397	0.491	0.489	0.484
zachary	0.502	0.499	0.479	0.323	0.350	0.517	0.442	0.332	0.525	0.533	0.509

Tab. A.11.: Results of rank-based criteria with MDS in low-dimensional space for each distance (best parameters) and dataset

	FE	SURP	LF	SCCT	SCT	RSP	SP	CT	PWSURP	PSURP	FEM
dolphins_2	50.581	220.761	17.879	278.168	216.042	210.816	5.511	4107.972	45.549	23.332	49.553
dolphins_4	1.489	1.765	2.283	24.189	27.444	1.405	5.011	39.291	2.481	2.056	1.456
football	1.601	1.557	1.540	5.338	5.633	1.574	1.830	3.454	1.611	1.403	1.600
LFR1	15.207	14.964	16.615	30.013	6.708	15.782	19.643	6837.576	4.783	4.092	37.912
LFR2	1.803	1.318	4.453	2.902	4.139	2.416	8.545	863.039	1.563	1.558	2.102
LFR3	1.111	1.104	2.524	3.829	2.485	1.599	5.287	169.156	1.063	1.064	1.334
news_2cl_1	11.045	12.106	1.921	22.679	5.322	9.779	32.905	14407.58	4.295	4.220	6.078
news_2cl_2	63.570	94.365	143.912	90.574	50.004	6.713	53.059	75108.65	105.539	105.440	100.792
news_2cl_3	780.474	938.279	819.667	281.393	211.391	162.630	129.349	185109.4	378.662	378.817	760.913
news_3cl_1	5.966	6.793	13.190	3.713	4.134	2.383	3.335	907.409	4.824	4.823	6.860
news_3cl_2	3.329	3.738	3.347	2.268	1.493	1.598	12.510	493.186	1.342	1.342	3.325
news_3cl_3	22.418	2.845	12.321	5.196	6.511	2.982	6.251	884.166	7.301	7.301	20.447
news_5cl_1	1.705	1.897	3.838	2.304	2.719	1.499	3.436	101.526	2.145	2.145	1.800
news_5cl_2	6.964	2.027	7.271	5.554	4.200	1.987	4.740	130.694	4.798	4.799	7.295
news_5cl_3	2.813	2.882	4.224	2.818	2.494	1.856	8.649	92.884	3.051	3.050	2.722
polbooks	1.397	1.332	1.632	4.127	4.164	1.207	2.452	4.184	1.441	1.480	1.409
zachary	9.517	2.660	7.634	52.308	76.937	2.130	8.609	8.082	9.936	5.640	3.179

Tab. A.12.: Results of divergence score with MDS in low-dimensional space for each distance (best parameters) and dataset

	LF	SCT	RSP	SP	PWSURP	PSURP
LF	/	0.463	0.305	0.163	0.191	0.273
SCT	0.463	/	0.091	0.435	0.040	0.048
RSP	0.305	0.091	/	0.163	0.305	0.305
SP	0.163	0.435	0.163	/	0.076	0.093
PWSURP	0.191	0.040	0.305	0.076	/	1.000
PSURP	0.273	0.048	0.305	0.093	1.000	/

Tab. A.13.: Comparison of the p-values of the Wilcoxon signed-rank tests performed between distances in low-dimensional space for the clustering

	LF	SCT	RSP	SP	PWSURP	PSURP
LF	/	$<10^{-4}$	0.011	0.044	0.001	0.001
SCT	$<10^{-4}$	/	$<10^{-4}$	0.001	$<10^{-4}$	$<10^{-4}$
RSP	0.011	$<10^{-4}$	/	0.943	0.435	0.010
SP	0.044	0.001	0.943	/	0.723	0.554
PWSURP	0.001	$<10^{-4}$	0.435	0.723	/	0.031
PSURP	0.001	$<10^{-4}$	0.010	0.554	0.031	/

Tab. A.14.: Comparison of the p-values of the Wilcoxon signed-rank tests performed between distances in low-dimensional space for the rank-based criteria

	LF	SCT	RSP	SP	PWSURP	PSURP
LF	/	0.619	0.025	0.943	0.055	0.010
SCT	0.619	/	0.015	0.906	0.163	0.163
RSP	0.025	0.015	/	0.035	0.309	0.356
SP	0.943	0.906	0.035	/	0.619	0.381
PWSURP	0.055	0.163	0.309	0.619	/	0.084
PSURP	0.010	0.163	0.356	0.381	0.084	/

Tab. A.15.: Comparison of the p-values of the Wilcoxon signed-rank tests performed between distances in low-dimensional space for the divergence score

Appendix: Second Research Question

B

	FE	SURP	LF	SCCT	SCT	RSP	SP	CT	PWSURP	PSURP	FEM
dolphins_2	0.814	0.889	0.889	0.814	0.814	0.753	0.889	0.889	0.814	0.814	0.814
dolphins_4	0.820	0.792	0.802	0.798	0.788	0.819	0.853	0.701	0.790	0.793	0.813
football	0.913	0.922	0.925	0.875	0.876	0.919	0.917	0.908	0.924	0.921	0.913
LFR1	0.983	0.983	0.983	0.981	0.981	0.983	0.900	0.008	0.983	0.983	0.981
LFR2	1.000	1.000	1.000	1.000	1.000	1.000	0.985	0.960	1.000	1.000	1.000
LFR3	1.000	1.000	1.000	1.000	1.000	1.000	0.994	0.940	1.000	1.000	1.000
news_2cl_1	0.832	0.845	0.820	0.809	0.846	0.832	0.682	0.741	0.833	0.831	0.832
news_2cl_2	0.639	0.667	0.672	0.593	0.594	0.649	0.518	0.581	0.677	0.678	0.696
news_2cl_3	0.808	0.808	0.748	0.794	0.793	0.769	0.845	0.687	0.838	0.822	0.808
news_3cl_1	0.765	0.760	0.751	0.760	0.760	0.780	0.737	0.584	0.762	0.762	0.764
news_3cl_2	0.764	0.765	0.760	0.726	0.751	0.763	0.636	0.519	0.758	0.758	0.755
news_3cl_3	0.771	0.775	0.708	0.759	0.768	0.766	0.721	0.453	0.770	0.770	0.765
news_5cl_1	0.687	0.688	0.651	0.665	0.678	0.677	0.682	0.234	0.688	0.690	0.683
news_5cl_2	0.653	0.656	0.612	0.592	0.614	0.642	0.599	0.407	0.659	0.656	0.651
news_5cl_3	0.628	0.626	0.589	0.525	0.573	0.612	0.484	0.474	0.617	0.622	0.625
polbooks	0.547	0.556	0.547	0.563	0.563	0.566	0.572	0.563	0.544	0.546	0.547
zachary	0.984	1.000	1.000	1.000	1.000	1.000	0.967	0.837	1.000	1.000	1.000

Tab. B.1.: Results of NMI with standard k-means on embedding for each distance (best parameters) and dataset

	FE	SURP	LF	SCCT	SCT	RSP	SP	CT	PWSURP	PSURP	FEM
dolphins_2	0.872	0.872	0.935	0.872	0.872	0.812	0.935	0.935	0.872	0.872	0.872
dolphins_4	0.867	0.826	0.847	0.557	0.571	0.890	0.704	0.711	0.839	0.839	0.855
football	0.776	0.790	0.785	0.481	0.520	0.779	0.470	0.791	0.793	0.792	0.779
LFR1	0.990	0.990	0.990	0.990	0.990	0.990	0.851	0.005	0.990	0.990	0.990
LFR2	0.891	0.897	0.880	0.988	0.996	0.893	0.760	0.669	0.927	0.927	0.885
LFR3	0.902	0.933	0.858	0.855	0.853	0.844	0.753	0.455	0.898	0.898	0.904
news_2cl_1	0.902	0.912	0.902	0.883	0.902	0.921	0.739	$<10^{-4}$	0.912	0.912	0.912
news_2cl_2	0.662	0.678	0.662	0.662	0.687	0.678	0.614	$<10^{-4}$	0.704	0.695	0.662
news_2cl_3	0.864	0.874	0.836	0.809	0.818	0.827	0.912	$<10^{-4}$	0.902	0.902	0.864
news_3cl_1	0.846	0.846	0.818	0.832	0.846	0.841	0.792	$<10^{-4}$	0.837	0.837	0.846
news_3cl_2	0.830	0.828	0.787	0.748	0.803	0.834	0.681	$<10^{-4}$	0.825	0.815	0.829
news_3cl_3	0.827	0.827	0.731	0.777	0.781	0.818	0.722	$<10^{-4}$	0.813	0.814	0.818
news_5cl_1	0.654	0.644	0.491	0.503	0.497	0.678	0.664	0.001	0.691	0.689	0.646
news_5cl_2	0.558	0.521	0.472	0.379	0.373	0.500	0.504	$<10^{-4}$	0.512	0.521	0.556
news_5cl_3	0.493	0.486	0.466	0.381	0.449	0.477	0.453	$<10^{-4}$	0.479	0.481	0.495
polbooks	0.650	0.665	0.650	0.671	0.682	0.688	0.642	0.708	0.650	0.650	0.650
zachary	1.000	1.000	1.000	1.000	1.000	1.000	0.882	0.882	1.000	1.000	1.000

Tab. B.2.: Results of CCR with standard k-means on embedding for each distance (best parameters) and dataset

	FE	SURP	LF	SCCT	SCT	RSP	SP	CT	PWSURP	PSURP	FEM
dolphins_2	0.860	0.731	0.886	0.868	0.860	0.812	0.888	0.911	0.860	0.860	0.862
dolphins_4	0.784	0.755	0.801	0.786	0.776	0.795	0.838	0.684	0.781	0.781	0.817
football	0.861	0.822	0.858	0.761	0.744	0.849	0.760	0.721	0.826	0.826	0.861
LFR1	0.999	0.990	0.990	0.994	0.993	0.995	0.922	0.035	0.998	0.995	0.998
LFR2	1.000	1.000	1.000	1.000	1.000	1.000	0.991	0.104	1.000	1.000	1.000
LFR3	1.000	1.000	1.000	1.000	1.000	1.000	0.993	0.140	1.000	1.000	1.000
news_2cl_1	0.895	0.898	0.889	0.911	0.893	0.892	0.769	0.045	0.901	0.902	0.893
news_2cl_2	0.750	0.736	0.740	0.742	0.737	0.739	0.629	0.066	0.759	0.793	0.732
news_2cl_3	0.883	0.911	0.837	0.843	0.879	0.856	0.911	0.124	0.902	0.902	0.892
news_3cl_1	0.829	0.830	0.815	0.806	0.810	0.838	0.806	0.083	0.824	0.823	0.826
news_3cl_2	0.829	0.827	0.803	0.789	0.806	0.831	0.697	0.121	0.823	0.823	0.811
news_3cl_3	0.834	0.839	0.769	0.823	0.829	0.827	0.773	0.125	0.838	0.838	0.827
news_5cl_1	0.710	0.713	0.626	0.648	0.676	0.692	0.716	0.058	0.719	0.720	0.711
news_5cl_2	0.576	0.585	0.555	0.512	0.546	0.585	0.555	0.049	0.600	0.595	0.587
news_5cl_3	0.630	0.615	0.576	0.516	0.570	0.617	0.463	0.055	0.624	0.630	0.623
polbooks	0.646	0.646	0.651	0.669	0.669	0.614	0.649	0.659	0.634	0.633	0.643
zachary	1.000	1.000	1.000	0.980	1.000	1.000	0.949	0.890	1.000	1.000	1.000

Tab. B.3.: Results of ARI with kernel k-means for each distance (best parameters) and dataset

	FE	SURP	LF	SCCT	SCT	RSP	SP	CT	PWSURP	PSURP	FEM
dolphins_2	0.802	0.692	0.834	0.810	0.802	0.753	0.836	0.864	0.802	0.802	0.805
dolphins_4	0.816	0.791	0.831	0.826	0.808	0.825	0.851	0.752	0.811	0.811	0.840
football	0.912	0.891	0.910	0.861	0.856	0.907	0.859	0.848	0.894	0.894	0.913
LFR1	0.998	0.981	0.981	0.988	0.987	0.990	0.879	0.060	0.996	0.990	0.997
LFR2	1.000	1.000	1.000	1.000	1.000	1.000	0.989	0.276	1.000	1.000	1.000
LFR3	1.000	1.000	1.000	1.000	1.000	1.000	0.991	0.334	1.000	1.000	1.000
news_2cl_1	0.821	0.825	0.813	0.846	0.818	0.817	0.684	0.092	0.830	0.831	0.818
news_2cl_2	0.658	0.649	0.649	0.657	0.647	0.646	0.523	0.092	0.668	0.703	0.634
news_2cl_3	0.811	0.855	0.749	0.793	0.827	0.770	0.844	0.168	0.838	0.838	0.822
news_3cl_1	0.764	0.765	0.755	0.752	0.757	0.772	0.740	0.168	0.759	0.757	0.758
news_3cl_2	0.770	0.765	0.743	0.736	0.749	0.772	0.635	0.181	0.765	0.765	0.755
news_3cl_3	0.768	0.773	0.698	0.766	0.769	0.760	0.714	0.202	0.772	0.772	0.761
news_5cl_1	0.687	0.690	0.646	0.667	0.684	0.676	0.685	0.161	0.694	0.694	0.687
news_5cl_2	0.646	0.650	0.607	0.598	0.617	0.644	0.597	0.142	0.653	0.654	0.649
news_5cl_3	0.623	0.614	0.582	0.541	0.579	0.611	0.497	0.138	0.618	0.622	0.614
polbooks	0.546	0.555	0.548	0.563	0.563	0.561	0.565	0.569	0.557	0.556	0.544
zachary	1.000	1.000	1.000	0.973	1.000	1.000	0.929	0.847	1.000	1.000	1.000

Tab. B.4.: Results of NMI with kernel k-means for each distance (best parameters) and dataset

	FE	SURP	LF	SCCT	SCT	RSP	SP	CT	PWSURP	PSURP	FEM
dolphins_2	0.965	0.927	0.971	0.967	0.965	0.952	0.972	0.977	0.965	0.965	0.965
dolphins_4	0.907	0.898	0.920	0.917	0.912	0.919	0.938	0.823	0.908	0.908	0.925
football	0.900	0.866	0.896	0.827	0.811	0.888	0.820	0.786	0.871	0.870	0.894
LFR1	1.000	0.997	0.997	0.998	0.998	0.998	0.973	0.388	0.999	0.998	0.999
LFR2	1.000	1.000	1.000	1.000	1.000	1.000	0.996	0.301	1.000	1.000	1.000
LFR3	1.000	1.000	1.000	1.000	1.000	1.000	0.997	0.403	1.000	1.000	1.000
news_2cl_1	0.973	0.974	0.971	0.977	0.973	0.972	0.939	0.573	0.975	0.975	0.973
news_2cl_2	0.933	0.929	0.930	0.931	0.929	0.930	0.897	0.602	0.936	0.945	0.928
news_2cl_3	0.970	0.977	0.958	0.959	0.969	0.963	0.977	0.631	0.975	0.975	0.972
news_3cl_1	0.941	0.941	0.935	0.933	0.934	0.944	0.932	0.438	0.939	0.938	0.940
news_3cl_2	0.940	0.939	0.930	0.926	0.932	0.940	0.888	0.472	0.938	0.938	0.933
news_3cl_3	0.942	0.944	0.918	0.938	0.940	0.939	0.918	0.462	0.944	0.944	0.939
news_5cl_1	0.874	0.876	0.830	0.848	0.860	0.866	0.876	0.292	0.879	0.879	0.875
news_5cl_2	0.721	0.748	0.742	0.732	0.745	0.761	0.743	0.287	0.781	0.763	0.749
news_5cl_3	0.823	0.812	0.774	0.713	0.764	0.810	0.680	0.295	0.821	0.828	0.821
polbooks	0.827	0.833	0.829	0.838	0.838	0.811	0.823	0.836	0.827	0.827	0.827
zachary	1.000	1.000	1.000	0.995	1.000	1.000	0.987	0.973	1.000	1.000	1.000

Tab. B.5.: Results of CCR with kernel k-means for each distance (best parameters) and dataset

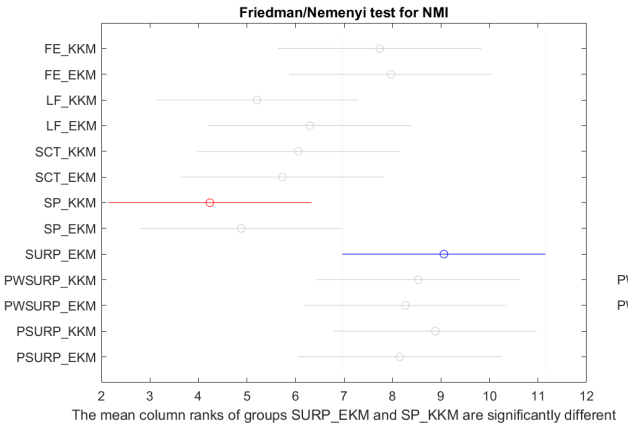

Fig. B.1.: Nemenyi test of the NMI score when performing clustering with KKM or EKM

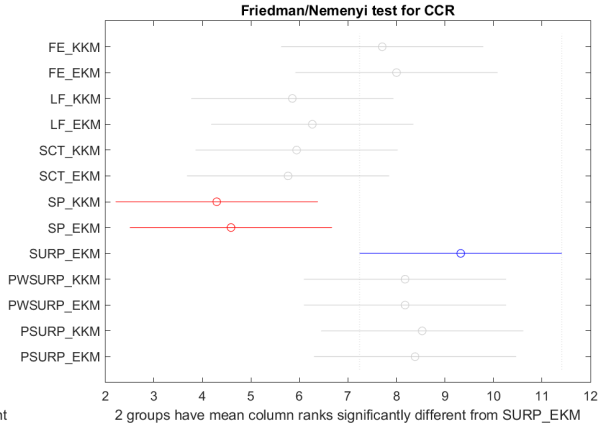

Fig. B.2.: Nemenyi test of the CCR score when performing clustering with KKM or EKM

	FE KKM	FE EKM	LF KKM	LF EKM	SCT KKM	SCT EKM	SP KKM	SP EKM	SURP EKM	PWSURP KKM	PWSURP EKM	PSURP KKM	PSURP EKM
FE KKM	/	0.296	0.030	0.194	0.006	0.020	0.022	0.093	0.194	0.839	0.808	0.685	0.626
FE EKM	0.296	/	0.017	0.094	0.005	0.013	0.019	0.084	0.340	0.542	0.839	0.903	1.000
LF KKM	0.030	0.017	/	0.268	1.000	0.855	0.068	0.332	0.005	0.119	0.068	0.078	0.049
LF EKM	0.194	0.094	0.268	/	0.761	0.463	0.093	0.163	0.021	0.241	0.191	0.135	0.244
SCT KKM	0.006	0.005	1	0.761	/	0.588	0.136	0.332	0.013	0.008	0.049	0.006	0.049
SCT EKM	0.020	0.013	0.855	0.463	0.588	/	0.136	0.435	0.004	0.025	0.040	0.025	0.048
SP KKM	0.022	0.019	0.068	0.093	0.136	0.136	/	0.210	0.007	0.017	0.031	0.017	0.028
SP EKM	0.093	0.084	0.332	0.163	0.332	0.435	0.210	/	0.023	0.102	0.076	0.113	0.093
SURP EKM	0.194	0.340	0.005	0.021	0.013	0.004	0.007	0.023	/	0.296	0.273	0.626	0.305
PWSURP KKM	0.839	0.542	0.119	0.241	0.008	0.025	0.017	0.102	0.296	/	0.839	0.898	0.855
PWSURP EKM	0.808	0.839	0.068	0.191	0.049	0.040	0.031	0.076	0.273	0.839	/	0.414	0.831
PSURP KKM	0.685	0.903	0.078	0.135	0.006	0.025	0.017	0.113	0.626	0.898	0.414	/	0.426
PSURP EKM	0.626	1	0.049	0.244	0.049	0.048	0.028	0.093	0.305	0.855	0.831	0.426	/

Tab. B.6.: Comparison of the p-values of the Wilcoxon signed-rank tests performed between each pair of (distance, clustering) combinations for ARI score

Appendix: Third Research Question

C

	BoPE_3D_ARI	BoPE_3D_R-b	BoPE_3D_D.s.	BoPE_LD_ARI	BoPE_LD_R-b	BoPE_LD_D.s.
dolphins_2	0.872	0.463	1372.562	0.872	0.610	46.692
dolphins_4	0.855	0.463	1.452	0.781	0.610	1.442
football	0.779	0.318	1.224	0.864	0.423	1.600
LFR1	0.990	0.077	2.191	0.990	0.231	38.078
LFR2	0.885	0.132	4.013	1.000	0.310	2.084
LFR3	0.904	0.120	4.398	1.000	0.269	1.336
news_2cl_1	0.912	0.231	6469.275	0.902	0.621	6.074
news_2cl_2	0.662	0.242	83.914	0.788	0.621	100.022
news_2cl_3	0.864	0.267	2360.405	0.883	0.619	762.250
news_3cl_1	0.846	0.218	4.052	0.834	0.628	6.858
news_3cl_2	0.829	0.190	2.607	0.811	0.616	3.311
news_3cl_3	0.818	0.214	3.681	0.828	0.652	20.450
news_5cl_1	0.646	0.190	1.596	0.707	0.615	1.803
news_5cl_2	0.556	0.174	1.251	0.597	0.624	7.297
news_5cl_3	0.495	0.144	1.256	0.639	0.604	2.723
polbooks	0.650	0.387	9.118	0.650	0.478	1.421
zachary	1.000	0.384	42.650	0.988	0.502	3.141

Tab. C.1.: Results of Bag-of-Paths Embedding in 3 dimensional and low-dimensional space for each dataset and assessment technique

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain
Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | **www.uclouvain.be/epl**