UCLouvain

École polytechnique de Louvain

# Graph embedding with application to semi-supervised classification, visualization, reconstruction and neighborhood preservation tasks: an experimental comparison

Author: **Alain Mbungu Safi**
Supervisor: **Marco Saerens**
Readers: **John Lee, Sylvain Courtain**
Academic year 2018–2019
Master [120] in Computer Science

# Acknowledgement

I would like first to express my gratitude to my supervisor, Marco Saerens, for his guidance through each stage of the process.

Furthermore I would like to thank the Professor John Lee, the teaching assistants Sylvain Courtain and Cyril De Bodt for having accompanied me during the experimental part by answering my questions.

I must express my very profound gratitude to Godelieve Lagae for providing me with unfailing support. My master degree would not have been achieved without you.

Last but not the least, I would like to thank all my family: my mother Jeanne Mawota for supporting me morally and spiritually. Also I thank my best friend Judith Matiela and any person of good will who has supported me in one way or another.

To God be the glory.

# Contents

# List of Symbols and Notation

**General**

| | |
|---|---|
| $a$, $b$, $c$, ..., $x$, $y$, $z$ | scalar variables or random variables, depending on the context |
| ***a, b, c, . . . , x, y, z*** | random vectors (bold italic) |
| $\alpha, \theta$ | scalar quantity, parameter, or constant |
| $\lvert\mathcal{S}\rvert$ | number of elements, or cardinality, of a set $\mathcal{S}$ |

**Matrices and Vectors**

| | |
|---|---|
| $\mathbf{M}$ | a matrix (uppercase bold) |
| $\mathbf{M}^T$ | transpose of matrix $\mathbf{M}$ |
| $\mathbf{M}^q$ | matrix $q$-power of $\mathbf{M}$ |
| $\mathbf{M}^{(q)}$ | Hadamard (elementwise) $q$-power of $\mathbf{M}$ containing elements $m_{ij}^q$ |
| $\mathbf{M} \circ \mathbf{N}$ | Hadamard (elementwise) matrix product providing elements $m_{ij}n_{ij}$ |
| **Diag(M)** | diagonal matrix containing the diagonal of the square matrix $\mathbf{M}$ |
| exp($\mathbf{M}$) | elementwise exponential of matrix $\mathbf{M}$ |
| $\mathbf{e}$ | unit column vector full of 1s of the appropriate size |
| $\mathbf{e}_i$ | $i$-th column of identity matrix $\mathbf{I}$, containing zero everywhere, except on row $i$, containing a 1 |
| $\lvert\lvert\boldsymbol{v}\rvert\rvert = \lvert\lvert\boldsymbol{v}\rvert\rvert_2$ | $L_2$-norm of vector $\boldsymbol{v}$ |

**Graphs**

| | |
|---|---|
| $G$ | a graph |
| $\mathcal{V}$ | set of nodes of a graph $G$ |
| $\mathcal{E}$ | set of edges (or arcs, links, connections) of a graph $G$ |

| | |
|---|---|
| **A** | adjacency matrix of *G*: $a_{ij} = w_{ij}$ when there is an edge between nodes $i$ and $j$; $a_{ij}$=0 otherwise |
| **C** | cost matrix associated with a graph *G* containing transition costs $c_{ij}$ |
| $w_{ij}$>0 | weights associated with edges $(i, j)$ of the graph *G*; they represent affinities between pairs of nodes |
| **D=Diag(Ae)** | diagonal degree matrix of undirected graph *G* containing degrees $a_{i\bullet} = \sum_j a_{ij}$ on its diagonal |
| $\mathbf{\Delta}^{(q)}$ | $n \times n$ dissimilarity matrix, where $[\mathbf{\Delta}^{(q)}]_{ij} = \Delta_{ij}^{q}$ |

**Embedding Methods**

| | |
|---|---|
| **Φ** | Latent social representation (or embedding) associated to each vertex $v$ in the graph |
| $\Phi(v_i)$ | Latent social representation column vector of a particular node $i$ |
| $\mathbf{K}_{BoPP}^{m}(\nu)$ | Similarity matrix constructed from the embedding provided by BoPP-m using appropriate hyper-parameter value *ν* |
| $\mathbf{K}_{BoPP}^{g}(\nu)$ | Similarity matrix constructed from the embedding provided by BoPP-g using appropriate hyper-parameter value *ν* |
| $\mathbf{K}_{\mathcal{M}}(\nu)$ | Similarity matrix constructed from the embedding provided by method $\mathcal{M}$ using appropriate hyper-parameter value *ν* |

# 1 Introduction

Nowadays, graphs have received significant attention, they became omnipresent across a large spectrum of real-world applications such as in social networks, telecommunication networks, citation networks, biological networks, electronic commerce, knowledge graph, etc. Effective graph analytics provides users a deeper understanding of what is behind the data and insights into how to make good use of the information hidden in graphs, and thus can benefit a lot of useful applications such as *link prediction* to predict a friendship in a social network like facebook or twitter, *community detection* and *node clustering* to group similar nodes together, *node classification* to assign a class label to each node in a graph based on the rules learned from the labeled nodes, *node recommendation* to recommend top $k$ nodes of interest to a given node based on certain criteria such as similarity, *seed identification* to detect good candidate users in order to spread optimally the information in a network, and *network visualization* where all nodes are embedded as 2D vectors and then plotted in a 2D space with different colours indicating nodes categories, . . . to name a few popular ones [7, 17]. All these applications benefit from both graph mining and machine learning tools.

In many real-word problems data is incomplete, which appears sometimes problematic, for instance, for making a decision (e.g., in the field of business), predicting the behavior of consumers (e.g., in marketing) or people (e.g., in social networks) . . . To remedy this situation, an estimate of missing data from the information in hand could be more convenient. This is where a *semi-supervised learning* comes in. In a context of node classification tasks, e.g., one could use semi-supervised classification algorithms to predict the labels of unlabeled nodes from a set of labeled ones. For instance, A. Mantrach et al. [38] developed three algorithms to avoid explicit computation of pairwise proximity between the nodes of the graph (which would be impractical for graphs containing millions of nodes) for semi-supervised node classification. Two of these algorithms were based on the *sum of the similarities* between the nodes to classify and the labeled nodes of the class whereas the other algorithm was based on the *discriminative random walks*. They tested the performance of their algorithms for the multi-class classification problem on the U.S. patents citation network containing 3 million nodes (of six different classes) and 38 million edges, and achieved competitive results (around 85% classification rate for a labeling rate of 10%, i.e. only one node over 10 is labelled) on this large

network; they classified the unlabeled nodes within a few minutes on a standard workstation.

Graph embedding is an effective yet efficient way to solve the graph analytics problem. It learns a low-dimensional representation of the graph such that the graph structural information and properties are maximally preserved. Thereby, choosing a good embedding method is a determining factor of performance. There exist a variety of embedding techniques in the literature such as Matrix factorization based methods (Graph Laplacian Eigenmaps [4], Node Proximity Matrix Factorization [50], ...) and Deep Learning based methods (DeepWalk [51], Node2vec [24], ...). Besides, there exist also some frameworks such as Bag-of-paths [17], defining relatedness as well as distance measures between nodes through kernels, from which the embedding can be extracted.

The differences between different graph embedding algorithms lie in how they define the graph property to be preserved. Different algorithms have different insights of the node similarities and how to preserve them in the embedded space. This thesis is built around the following research questions:

1. Which kernel embedding method performs well on classification task? More precisely, which one of the following kernels outperforms in term of semi-supervised classification accuracy: free energy, covariance of node co-presences on hitting paths, correlation of node co-occurrences on hitting paths, regularized commute-time and modularity matrix?
2. Are the deep learning based embedding methods competitive or improve the shortcomings of kernel based methods in terms of node embedding ?
3. Similarly, are the deep learning based embedding methods competitive or improves the shortcomings of kernel based methods on node classification tasks ?
4. Are deep learning based methods more efficient in terms of visualization, graph reconstruction, nonlinear dimensionality reduction compared to kernel based approaches?

To answer all the aforementioned questions, experiments will be performed on fourteen different datasets.

We structured this thesis as follows: the next chapter is devoted to give the reader a short overview on graph theory, machine learning, succinct description of the baseline node embedding techniques as well as some statistical concepts used for assessing the methods. In the third chapter, we will present the investigated techniques, which will be later on compared with the baselines techniques addressed in the second chapter. The fourth and fifth chapters, as far as they are concerned, will be dedicated

to the embedding evaluation through semi-supervised node classification, graph visualization, graph reconstruction, quality assessment of nonlinear dimensionality reduction rank-based criteria as well as the parameter sensitivity. Finally, a global conclusion going over different results obtained along this work will be given.

# 2 Basic Concepts

Graph mining and machine learning are the core of this work, that is why in this chapter we present some basic concepts on graph theory, machine learning with more focus on classification tasks, node embedding techniques by briefly presenting our baseline ones, and finally statistical procedures and tests aiming to assess the quality of the results.

## 2.1 Networks or graphs

A *graph* or *network* $G$ [3, 16, 48] is a mathematical structure that can be formally defined by providing

- a finite nonempty set $\mathcal{V}$, the elements of which are called **nodes** (or **vertices**)
- a set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, the elements of which are (ordered or not) pairs of nodes called **edges** (or **arcs**, **links**)

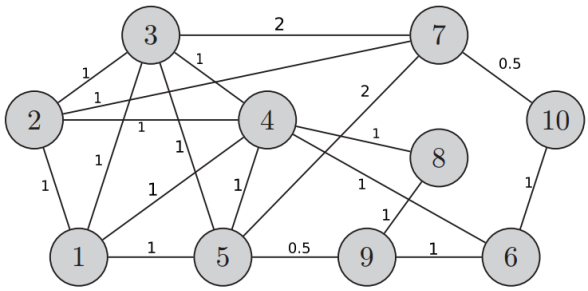
Thus, a graph is a collection of nodes linked by edges, $(\mathcal{V}, \mathcal{E})$. In social network, for instance, nodes are **people** while edges represent the existence of a **relationship** between them. The structure of a graph $G$ can be captured in a $\mathcal{V} \times \mathcal{V}$ matrix $\mathbf{A}$ called the **adjacency matrix** defined in a standard manner as

$$a_{ij} = [\mathbf{A}]_{ij} = \begin{cases} w_{ij} & \text{if there is an edge between nodes } i \text{ and } j, \\ 0 & \text{otherwise.} \end{cases} \tag{2.1}$$

For an *unweigthed graph* $w_{ij} \in \{0, 1\}$ while for a *weigthed graph* $w_{ij} \in [0, +\infty]$ since in this work we only consider graphs that are weighted positively. Weights represent affinities between nodes. In some situations, instead of affinities forming the adjacency matrix, nonnegative costs $c_{ij} = 1/a_{ij} > 0$ are assigned[1] to the edges of $G$. So, the **cost matrix** is defined as

$$c_{ij} = [\mathbf{C}]_{ij} = \begin{cases} c_{ij} & \text{if there is an edge between nodes } i \text{ and } j, \\ \infty & \text{otherwise.} \end{cases} \tag{2.2}$$

[1] Note that costs can also be assigned independently of the adjacency matrix.




(a) Visual representation of the graph

$$\begin{pmatrix}
0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 1 & 0 & 1 & 1 & 0 & 2 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
1 & 0 & 1 & 1 & 0 & 0 & 2 & 0 & 0.5 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 1 & 2 & 0 & 2 & 0 & 0 & 0 & 0 & 0.5 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0.5 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0.5 & 0 & 0 & 0
\end{pmatrix}$$

(b) Adjacency matrix of the graph

**Fig. 2.1.:** An undirected weighted graph [16].

In fact, the cost matrix **C** has opposite concept to the adjacency matrix. In the electrical network, for instance, costs play the role of resistances and affinities are considered as conductances. An illustration of a graph is given in the Fig. 2.1. Note that $w_{ij} = w_{ji}\ \forall i, j$, since we are only working on **undirected** or **symmetric** graphs. The degrees of nodes can be computed from the adjacency matrix. For an undirected graph $G$, $\sum_{j=1}^{|\mathcal{V}|} a_{ij} = \sum_{j=1}^{|\mathcal{V}|} a_{ji}$ , and it corresponds to the **degree** $d_i(G) = d_i$ of node $i$. The **volume** of the graph is simply the sum over all the elements of matrix **A**, mathematically $\text{vol}(G) = \sum_{i=1}^{|\mathcal{V}|} \sum_{j=1}^{|\mathcal{V}|} a_{ij}$. Note that any adjacency matrix **A** can be symmetrized by taking $(\mathbf{A} + \mathbf{A}^T)/2$.

## 2.2 Machine Learning Methods

*Machine Learning* (ML) is a part of Artificial Intelligence getting machines capable of learning without being explicitly programmed [2, 5, 11, 43, 46]. The main intuition behind ML is that from a given set of input-output pairs, the system learns a model (target function) by applying some maths, so that it can predict results for future inputs (unseen data).

Nowadays, big companies like Microsoft, Google, Facebook, Amazon, ... are intensively exploiting ML models and the intelligent systems built on these models sometimes reveal high performance. ML is also used in many other real-world problems such as autonomous driving, image recognition, speech recognition, medical diagnosis, find terrorist suspects, classification and prediction tasks, ...

### 2.2.1 Types of machine learning

ML is usually divided into two main types. The first approach is called *predictive* or *supervised learning*. The goal is to learn a mapping from inputs $\boldsymbol{x}$ to outputs $y$, given a labeled set of input-output pairs $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$. Here $\mathcal{D}$ is called the

*training set*, and $n$ is the number of training examples. $y_i \in \{1, \ldots, m\}$ (here $m$ is the number of classes or categories, e.g., male and female) is a real-valued scalar (such as income level). When $y_i$ is *categorical*, the problem is known as *classification* or *pattern recognition*, and when $y_i$ is real-valued, the problem is known as *regression*. There exist many learning algorithms dedicated to this type of machine ML such as Random Forests, Perceptron, Support Vector Machines, Naive Bayes Classifier, ...

The second approach is called *descriptive* or *unsupervised learning*. Here we are only given inputs, $\mathcal{D}=\{\boldsymbol{x}_i\}_{i=1}^n$, and the goal is to find "interesting patterns" in the data. This is sometimes called *knowledge discovery*. $K$-means and hierarchical clustering are the most popular algorithms for this type of ML. See more details in the reference books [2, 5, 11, 43, 46]. However, between these two classical approaches there is the so-called *semi-supervised learning*, where the training set is partially labeled. In this work, we are interested in this type of ML because we will evaluate the performance of node embedding techniques on semi-supervised node classification tasks. So, the classification model will predict the labels of unlabeled nodes and its prediction will be compared to the true labels which were hidden.

### 2.2.2 Support Vector Machines

*Support Vector Machines (SVM)* [27] are one of the most popular machine learning models, aiming to find out the separating hyperplane by maximizing the margin, that is it places the decision boundary such that it maximizes the distance between the two classes (e.g, positive and negative). The advantage of using such a model is that small variations will be less likely to affect the classification (robustness). We introduce SVM classifier as well as other classification approaches because they will be used for semi-supervised node classification in the chapter 4.

Given a training set of $n$ instance-label pairs $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)$ with $\boldsymbol{x}_i \in \mathbb{R}^d$ and $y_i \in \{1, -1\}$. We define a *hyperplane* (or *linear discriminant*) by

$$g(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + w_0 = \langle \boldsymbol{w}, \boldsymbol{x} \rangle + w_0 = 0 \tag{2.3}$$

where $\boldsymbol{w}$ is a unit vector ($||\boldsymbol{w}|| = 1$) and $w_0$ the intercept. Computing the support vector classifier amounts to solve the following optimization problem

$$\begin{aligned} \min_{w,\xi} \quad & \frac{1}{2}||\boldsymbol{w}||^2 + c\sum_{i=1}^{n}\xi_i \\ \text{subject to} \quad & y_i(\boldsymbol{w}^T\boldsymbol{x}_i + w_0) \geq 1 - \xi_i \\ & \xi_i \geq 0. \end{aligned} \tag{2.4}$$

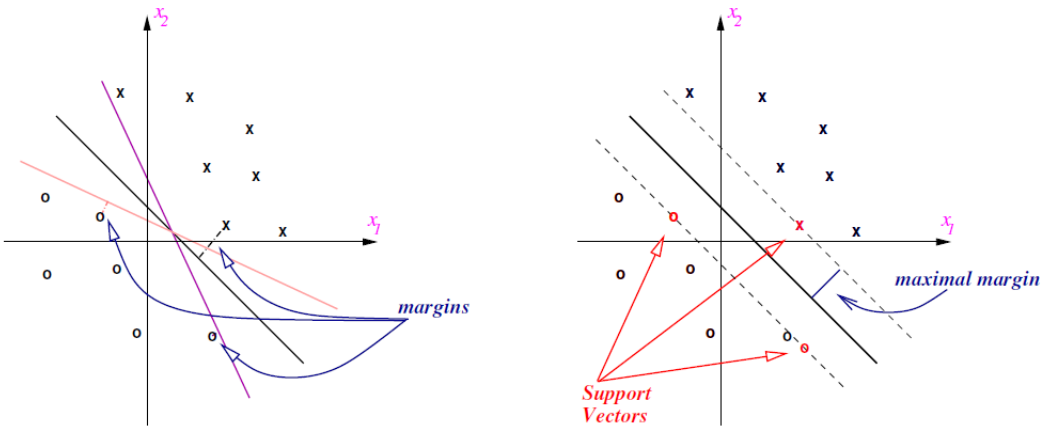



**Fig. 2.2.:** Support vector machine. Left: the separating hyper-planes with small margin. Right: a separating hyper-plane with large margin. Illustration from [12]

where the "cost" parameter $c > 0$ is the *penalty parameter* of the error term (the separable case corresponds to $c = \infty$) and $\xi_i$ are called *slack variables*, which measure by how much the constraint in equation 2.4 is violated for each training points. Thus, if $0 < \xi_i < 1$ the margin is not satisfied but $\boldsymbol{x}_i$ is still correctly classified while if $\xi_i > 1$ then $\boldsymbol{x}_i$ is misclassified. After solving the problem 2.4, the classification rule induced by $g(\boldsymbol{x})$ is

$$f(\boldsymbol{x}) = \text{sign}[\langle \boldsymbol{w}, \boldsymbol{x} \rangle + w_0] = \text{sign}[\sum_{\boldsymbol{x}_i \in \mathcal{SV}} \alpha_i y_i \langle \boldsymbol{x}_i, \boldsymbol{x} \rangle + w_0] \tag{2.5}$$

$\alpha_i$ is the positive weight of the support vector $\boldsymbol{x}_i$ and $\mathcal{SV}$ is a set of support vectors. Note that a support vector $\boldsymbol{x}_i$ is a training example for which $y_i(\boldsymbol{w}^T \boldsymbol{x}_i + w_0) = 1$. An overview of SVM classifier is given in Fig. 2.2.

However, if the training samples are not linearly separable with margin 1 in the input space, we could then consider a *non-linear mapping function* $\phi$ to a new feature space, defined through a *kernel* $\mathbf{K}(\boldsymbol{x}, \boldsymbol{x}') = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x}') \rangle$. Hence, the decision function with kernel becomes

$$f(\boldsymbol{x}) = \text{sign}[\langle \boldsymbol{w}, \boldsymbol{x} \rangle + w_0] = \text{sign}[\sum_{\boldsymbol{x}_i \in \mathcal{SV}} \alpha_i y_i \mathbf{K}(\boldsymbol{x}_i, \boldsymbol{x}) + w_0] \tag{2.6}$$

### 2.2.3 Sum-of-similarities based on a Kernel

Sum-of-Similarities is one of the kernel-based approach for classification tasks. Since, a *kernel* can be seen as a similarity function over pairs of data points in matrix representation. Thus, from an input matrix $\mathbf{X}$, the inner product kernel $\mathbf{K}$ is given by $\mathbf{K}=\mathbf{X}\mathbf{X}^{\mathrm{T}}$, with $\mathbf{X}$ in $\mathbb{R}^{|\mathcal{V}| \times d}$ and $\mathbf{K}$ in $\mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$. In a context of semi-supervised classification, the goal is to classify unlabeled nodes based on the knowledge of

labeled nodes and the graph structure. *Label consistency* is assumed in this model, stating that neighbor nodes are likely to share the same class label [30, 59].

Given the partially labeled input data, we associate a class membership $y_i^c \in \{1, -1, 0\}$ to each training node $i$, with a value equal to 1 when the label is known and it belongs to class $c$, -1 when it belongs to another class and 0 otherwise (it is not labeled). We can then construct a ternary matrix $\mathbf{Y}$ in $\mathbb{R}^{|\mathcal{V}| \times m}$, containing the class memberships for all the nodes with $m$ class labels.

Therefore, the sum of similarities $\mathbf{S} = \mathbf{KY}$ is computed over every node of the graph belonging to class $c$. Finally, the class prediction $\hat{y}_i^c$ for node $i$ is then obtained as follows

$$\hat{y}_i^c = \underset{c \in \{1,\ldots,m\}}{\operatorname{argmax}} \; \boldsymbol{s}^c \tag{2.7}$$

that is, we assign node $i$ to class $c$ for which the sum of similarities is maximal. Here, $\boldsymbol{s}^c$ is a row corresponding to node $i$ in the matrix $\mathbf{S}$. The sum-of-similarities is an equivalent of *Nearest neighbor classification*, but this time based on a similarity matrix. This method will be denoted by a suffix '-s' in the chapter 4, dedicated to the classification tasks.

### 2.2.4 Bounded normalized random walk with restart

An alternative approach to efficiently estimate the sum-of-similarities is the so-called *Bounded normalized random walk with restart* [15, 38]. Indeed, the normalized random walk with restart matrix $\mathbf{K}$ is given by

$$\mathbf{K} = (\mathbf{D} - \alpha \mathbf{A}^T)^{-1} \tag{2.8}$$

where $\mathbf{A}$ is the adjacency matrix, $\mathbf{D} = \mathbf{Diag(Ae)}$, and $\mathbf{e}$ is a column vector full of 1's. If matrix $\mathbf{A}$ is symmetric, equation 2.8 defines a valid kernel on a graph. The parameter $\alpha \in ]0, 1[$ denotes, at each time step of a random walk, the probability that the random walker continues his walk. We are looking for a way to bound the sum-of-similarities $\mathbf{S}$=$\mathbf{KY}$ up to a-priori-specified walk length $\tau$. Since the transition matrix of the natural random walk on the graph is $\mathbf{P} = \mathbf{D}^{-1}\mathbf{A}$, the similarity up to $\tau$, denoted by $\hat{\mathbf{S}}(\tau)$, can be computed using the following recurrence equation:

$$\begin{cases} \tilde{\mathbf{S}}(0) \leftarrow \mathbf{Y} \\ \tilde{\mathbf{S}}(t) \leftarrow \alpha \mathbf{P}^T \tilde{\mathbf{S}}(t-1) + \tilde{\mathbf{S}}(0), \text{ for } t = 1, \ldots, \tau \\ \hat{\mathbf{S}}(\tau) \leftarrow \mathbf{D}^{-1} \tilde{\mathbf{S}}(\tau) \end{cases} \tag{2.9}$$

The decision rule remains the same as in equation 2.7. Thereafter, bounded normalized random walk with restart will be denoted by SoS.

## 2.3 Distances and Kernels on Graphs

### 2.3.1 The Bag-of-paths framework and the Bag-of-hitting-paths probabilities

The Bag-of-paths framework [13, 16, 17, 26] is a model that is based on the probability of drawing a *path* (or *walk*) $\wp$ starting in node $i$ and ending in node $j$ from a bag of paths $\mathcal{P}$, which is simply the set of all the possible paths in $G$ . A path $\wp$ is a sequence of jumps to adjacent nodes on $G$ (including loops), initiated from a starting node $s(\wp) = i$ and stopping in an ending node $e(\wp) = j$. The total cost of a path $\wp$ is simply the sum of the local costs along $\wp$ and is denoted as $\tilde{c}(\wp)$. Thus, we can define *the generalized bag-of-paths probabilities* of drawing the path $\wp$ as follows:

$$P(\wp) = \frac{w(\wp)}{\sum_{\wp' \in \mathcal{P}} w(\wp')} = \frac{w(\wp)}{w(\mathcal{P})} \tag{2.10}$$

where $w(\wp)$ is the *weight* (see the next paragraph) of a path $\wp = (i_0, \ldots, i_l)$, defined as the product of the weights on its edges, i.e.,

$$w(\wp) = \prod_{\tau=0}^{l-1} w_{i_\tau, i_{\tau+1}} \tag{2.11}$$

Given a graph $G = (\mathcal{V}, \mathcal{E})$ where each edge linking two nodes $i$ and $j$ is associated with a positive scalar $c_{ij} \geq 0$ representing the immediate cost of following this edge, the adjacency matrix $\mathbf{A}$, the cost matrix $\mathbf{C}$, the transition probability matrix of the *natural random walk* $\mathbf{P}^{\text{ref}}$, and the inverse temperature $\theta = 1/T$ parameter; we can define the matrix

$$\mathbf{W} = \exp(-\theta \mathbf{C}) \circ \mathbf{P}^{\text{ref}} \tag{2.12}$$

where $\circ$ marks elementwise (Hadamard) matrix product and $\mathbf{W}$ is called the *weighted adjacency matrix*. Note that this matrix is not a usual adjacency matrix, as it will be used to construct path weights and path probabilities. So, using this matrix we define the *fundamental matrix*, but non-absorbing Markov Chain

$$\mathbf{Z} = (\mathbf{I} - \mathbf{W})^{-1} \tag{2.13}$$

Two kinds of path can be derived from, *regular path*, for which the last node is an non-absorbing node, while for the *hitting path*, the final node is considered as an absorbing node: it can thus only appear once, at the end of the path. Since we are

only interested in by hitting paths, the bag-of-paths probability matrix is then given by

$$\mathbf{\Pi}_h = \frac{\mathbf{Z}\mathbf{D}_h^{-1}}{\boldsymbol{e}^T\mathbf{Z}_h\boldsymbol{e}}, \text{ with } \mathbf{Z} = (\mathbf{I} - \mathbf{W})^{-1}, \mathbf{D}_h = \mathbf{Diag}(\mathbf{Z}) \text{ and } \mathbf{Z}_h = \mathbf{Z}\mathbf{D}_h^{-1} \tag{2.14}$$

### 2.3.2 Free energy distance

The *Potential* or *Free energy distance* [13, 17, 29], is one of the distances between nodes based on the bag of hitting paths. It is defined as follows

$$\Delta_{ij}^{\phi} = \begin{cases} \frac{\phi(i,j)+\phi(j,i)}{2} & \text{if } i \neq j \\ 0 & \text{if } i = j \end{cases}, \text{ where } \phi(i,j) = -\tfrac{1}{\theta}\log z_{ij}^h = -\tfrac{1}{\theta}\log\frac{z_{ij}}{z_{jj}} \tag{2.15}$$

and $z_{ij}$ is element $i, j$ of the fundamental matrix $\mathbf{Z}$ (see equation 2.13). We adopt two ways of mapping a distance matrix to a kernel matrix like in [17]:

- From classical multidimensional scaling (MDS), a centered kernel $\mathbf{K}$ can be derived from a matrix of squared distances $\mathbf{\Delta}^{(2)}$ as follows

  $$\mathbf{K} = -\frac{1}{2}\mathbf{H}\mathbf{\Delta}^{(2)}\mathbf{H} \tag{2.16}$$

  where $\mathbf{H} = (\mathbf{I} - \mathbf{e}\mathbf{e}^T/n)$ is the centering matrix and matrix $\mathbf{\Delta}^{(2)}$ contains the elementwise squared distances. This method is denoted by BoPP-m and has one hyper-parameter $\theta$.
- Using Gaussian kernel

  $$\mathbf{K} = \exp(-\mathbf{\Delta}^{(2)}/2\sigma^2) \tag{2.17}$$

  where the exponential is taken elementwise. This method is denoted by BoPP-g and has one hyper-parameter $\theta$.

Note that both approaches will be investigated (see chapters 4 and 5) and the five dominant eigenvectors of these kernels as well as those we address in the next sections, will be extracted and then injected as features (social dimensions) either into a SVM classifier or used in sum-of-similarities kernel fashions (see sections 2.2.3 and 2.2.4). In addition, embeddings using only two social dimensions will also be investigated.

### 2.3.3 Covariance matrix (presence on hitting paths)

Before diving into the core of this section, let us first address the nodes *(co-)presence* and *(co-)occurrence* on paths [26]. Indeed, the *presence* variable (indicator variable) of node $i$ on a given observed path $\wp$ is defined by

$$\delta(i \in \wp) \triangleq \begin{cases} 1 & \text{if } i \in \wp \\ 0 & \text{otherwise.} \end{cases} \tag{2.18}$$

Let $\eta(i \in \wp)$ be the number of *occurrences* (or simply *occurrence*) of node $i$, that is number of times we visited node $i$ along a path $\wp$. The co-presence and co-occurrence of nodes $i$ and $j$ on path $\wp$ are respectively $\delta(i \in \wp)\delta(j \in \wp)$ and $\eta(i \in \wp)\eta(j \in \wp)$, and the covariance based on node presences on hitting paths, denoted by CovH, is computed as follows

$$\mathrm{Cov}(\delta(i \in \wp^{\mathrm{h}}), \delta(j \in \wp^{\mathrm{h}})) = \mathbb{E}^{\mathrm{h}}[\delta(i \in \wp^{\mathrm{h}})\delta(j \in \wp^{\mathrm{h}})] - \mathbb{E}^{\mathrm{h}}[\delta(i \in \wp^{\mathrm{h}})]\mathbb{E}^{\mathrm{h}}[\delta(j \in \wp^{\mathrm{h}})] \tag{2.19}$$

where

$$\mathbb{E}^{\mathrm{h}}[\delta(i \in \wp^{\mathrm{h}})\delta(j \in \wp^{\mathrm{h}})] = \sum_{\wp^{\mathrm{h}} \in \mathcal{P}^{\mathrm{h}}} \delta(i \in \wp^{\mathrm{h}})\delta(j \in \wp^{\mathrm{h}})\mathrm{P}(\wp^{\mathrm{h}}) = \frac{w(\mathcal{P}^{\mathrm{h}(+\{i,j\})})}{w(\mathcal{P}^{\mathrm{h}})} \tag{2.20}$$

$$\mathbb{E}^{\mathrm{h}}[\delta(i \in \wp^{\mathrm{h}})] = \sum_{\wp^{\mathrm{h}} \in \mathcal{P}^{\mathrm{h}}} \delta(i \in \wp^{\mathrm{h}})\mathrm{P}(\wp^{\mathrm{h}}) = \frac{w(\mathcal{P}^{\mathrm{h}(+i)})}{w(\mathcal{P}^{\mathrm{h}})} \tag{2.21}$$

$$w(\mathcal{P}^{\mathrm{h}(+\{i,j\})}) = \sum_{s,t \in \mathcal{V}} w\left(\mathcal{P}_{st}^{\mathrm{h}(+\{i,j\})}\right) \tag{2.22}$$

where $\mathcal{P}_{st}^{\mathrm{h}(+\{i,j\})}$ refers to the subset of hitting paths connecting $s$ to $t$ and involving nodes $i$ and $j$. Note that it has been shown that $\mathcal{P}_{st}^{\mathrm{h}(+\{i,i\})}$ reduces to $\mathcal{P}_{st}^{\mathrm{h}(+i)}$. However,

$$w(\mathcal{P}^{\mathrm{h}}) = \sum_{i,j=1}^{n} z_{ij}^{h} = z_{\bullet\bullet}^{\mathrm{h}} \tag{2.23}$$

which is simply the weight of the set of all hitting paths. For more details, see the reference paper [26] about the computation of the quantities $w\left(\mathcal{P}_{st}^{\mathrm{h}(+\{i,j\})}\right)$ and $w\left(\mathcal{P}_{st}^{\mathrm{h}(+i)}\right)$.

### 2.3.4 Correlation matrix (number of occurrences on hitting paths)

Already familiarized with the definition of nodes *(co-)presence* and *(co-)occurrence* on paths, let's now compute the correlation for co-occurrences of node on hitting paths, denoted by NCorH, and defined as follows:

$$\mathrm{Cor}(\eta(i \in \wp^{\mathrm{h}}), \eta(j \in \wp^{\mathrm{h}})) = \frac{Cov(\eta(i \in \wp^{\mathrm{h}}), \eta(j \in \wp^{\mathrm{h}}))}{\sqrt{Cov(\eta(i \in \wp^{\mathrm{h}}), \eta(i \in \wp^{\mathrm{h}}))Cov(\eta(j \in \wp^{\mathrm{h}}), \eta(j \in \wp^{\mathrm{h}}))}} \tag{2.24}$$

where

$$\mathrm{Cov}(\eta(i \in \wp^{\mathrm{h}}), \eta(j \in \wp^{\mathrm{h}})) = \mathbb{E}^{\mathrm{h}}[\eta(i \in \wp^{\mathrm{h}})\eta(j \in \wp^{\mathrm{h}})] - \mathbb{E}^{\mathrm{h}}[\eta(i \in \wp^{\mathrm{h}})]\mathbb{E}^{\mathrm{h}}[\eta(j \in \wp^{\mathrm{h}})] \tag{2.25}$$

$$\mathbb{E}^{\mathrm{h}}[\eta(i \in \wp^{\mathrm{h}})\eta(j \in \wp^{\mathrm{h}})] = \sum_{s,t \in \mathcal{V}} \sum_{\wp^{\mathrm{h}}_{st} \in \mathcal{P}^{\mathrm{h}}_{st}} \eta(i \in \wp^{\mathrm{h}}_{st})\eta(j \in \wp^{\mathrm{h}}_{st}) \frac{w(\wp^{\mathrm{h}}_{st})}{w(\mathcal{P}^{\mathrm{h}})} \tag{2.26}$$

$$\mathbb{E}^{\mathrm{h}}[\eta(i \in \wp^{\mathrm{h}})] = \sum_{s,t \in \mathcal{V}} \sum_{\wp^{\mathrm{h}}_{st} \in \mathcal{P}^{\mathrm{h}}_{st}} \eta(i \in \wp^{\mathrm{h}}_{st}) \frac{w(\wp^{\mathrm{h}}_{st})}{w(\mathcal{P}^{\mathrm{h}})} \tag{2.27}$$

Note that such covariance and correlation matrices are positive semidefinite (Gram matrices [49]) and are therefore valid kernels on a graph (see [15, 16, 25, 52, 53]). See more details in the reference paper [26].

### 2.3.5 Modularity matrix

Modularity has recently become quite popular as a way to measure the goodness of a clustering of a graph. The intuition behind the definition of modularity is that the farther the subgraph corresponding to each community is from a random subgraph (the null model), the better or more signifiant the discovered community structure is [1]. Mathematically, the modularity [48] is defined as follows

$$\mathcal{Q} = \frac{1}{\mathrm{vol}(G)} \sum_{k=1}^{m} \boldsymbol{u}_k^T \mathbf{Q} \boldsymbol{u}_k = \frac{1}{\mathrm{vol}(G)} \sum_{k=1}^{m} \boldsymbol{u}_k^T \Big( \mathbf{A} - \frac{\mathbf{d}_o \mathbf{d}_i^T}{\mathrm{vol}(G)} \Big) \boldsymbol{u}_k \tag{2.28}$$

where $\mathbf{d}_i = \mathbf{A}^T\mathbf{e}$ and $\mathbf{d}_o = \mathbf{A}\mathbf{e}$ are respectively the indegree and the outdegree vectors. The goal is to maximize the modularity $\mathcal{Q}$ with respect to the number of clusters $m$ as well as the binary membership vectors $\mathbf{u}_k$. $\mathbf{Q}$ is called *modularity matrix*. Later on, we will denote this method by Q.

## 2.4 Entropy and Perplexity

*Entropy* and *perplexity* are the most common metrics used in information theory to evaluate a model [28]. We introduce these two concepts because they are very important for one of the embedding techniques (namely tSNE) that we will present in the next chapter.

- **Entropy**: is a measure of the average information contained in the data. For a random variable $x$ that ranges over a set $\mathcal{X}$ of whatever we are predicting (words, letters, parts of speech, class label, ...), and that has a particular probability function $p(x)$. The entropy is given by

$$H(x) = -\sum_{x \in \mathcal{X}} p(x) \log_2 p(x) \tag{2.29}$$

- **Perplexity**: measures how good a probability distribution predicts a sample. A low perplexity indicates that distribution function is good at predicting sample. It is given by

$$Perp(x) = 2^{H(x)} \tag{2.30}$$

  where $H(x)$ is the entropy of the distribution (see equation 2.29).

## 2.5 Assessing the methods: Ranking and Statistical Tests

The goal of this work is to assess empirically the performance of each embedding technique. To this end, we need statistical tests to rate and compare the results across all the datasets. Thus, all the statistical procedures and tests that we introduce in this section will be used in experimental part (see chapters 4 and 5). Our statistical level of significance $\alpha$ is set to 5% during all the experiments.

### 2.5.1 Borda Count Method

*Borda count method* or *Borda's Ranking* [55], is a voting method widely used in order to rate candidates in order of preferences. Here, the simple Borda's ranking[2] will be used for rating globally the results of each embedding techniques (*candidates* or *alternatives*), where preferences are classification accuracies across each dataset. The highest ranked technique (if for example an $n$-way vote) gets $n$ votes and each subsequent alternative gets one vote less (so the number two gets $n-1$ votes and

[2]Simple Borda's Ranking because all the methods are assumed equally weighted.

the number three $n-2$ and so on). Then, for each alternative, all the votes are added up and the alternative with the highest number of votes wins the election.

### 2.5.2 Paired Wilcoxon signed-rank test

The *paired samples Wilcoxon test* also known as *Wilcoxon signed-rank test* [21], is a non-parametric alternative test of location used to compare paired data. The right-sided test will be used for getting more precise information concerning the relative performance of each method by mean of pairwise comparisons across all the datasets. Given a random sample (i.i.d) of $n$ pairs $(x_1, y_1), \ldots, (x_n, y_n)$, we are interested in the differences $d_i = x_i - y_i$, that is the test focuses on the median of the differences $d_i$. The hypothesis to test is then

$$H_0 : M_d = 0$$

against the alternative

$$H_1 : M_d > 0$$

we form the $n$ differences $d_i = x_i - y_i$ and rank their absolute magnitudes from smallest to largest using integers $\{1, 2, \ldots, n\}$, keeping track of the original sign of each difference. The test statistic (sum of the positive ranks) is given by

$$S_+ = \sum_{i=1}^{n} I\{d_i > 0\} \tag{2.31}$$

Under the null hypothesis, $S_+ \sim Bin(n; 0.5)$. Thus, at level of significance of .05, reject $H_0$ when the exact p-value $P(S_+ \geq t | H_0) \leq .05$, where $t$ is the observed value of $S_+$.

### 2.5.3 Friedman Rank Test

Friedman Rank Test [10, 18, 19] is a non-parametric equivalent of the repeated-measures ANOVA (equivalent of the one-way ANOVA). Friedman test as well as Nemenyi test (see the next section) will be used to detect differences in performances between the embedding techniques. Given $k$ algorithms and $n$ datasets, Friedman test determine whether the $k$ algorithms have equal medians across all the $n$ datasets. The hypothesis to test is then

$$H_0 : M_{.1} = M_{.2} = \ldots = M_{.k}$$

against the alternative

$$H_1 : \text{Not all } M_{.j} \text{ are equal}$$

To conduct the test, for each dataset separately, the best performing algorithm getting the rank of 1, the second best rank 2, ... In case of ties average ranks are assigned. Let $r_i^j$ be the rank of the $j$-th of $k$ algorithms on the $i$-th of $n$ data sets. The Friedman test compares the average ranks of algorithms,

$$r_j = \frac{1}{n}\sum_{i=1}^{n} r_i^j \tag{2.32}$$

Under the null-hypothesis, which states that all the algorithms are equivalent and so their ranks $r_j$ should be equal, the Friedman statistic

$$\chi_F^2 = \frac{12n}{k(k+1)}\left[\sum_{j=1}^{k} r_j^2 - \frac{k(k+1)^2}{4}\right] \tag{2.33}$$

is distributed according to $\chi_F^2$ with $k-1$ degrees of freedom, when $n$ and $k$ are big enough ($n > 10$ and $k > 5$). For small number of algorithms and datasets, exact critical values can be picked up from the table. Thus, we reject $H_0$ if $\chi_F^2 > \chi_{k-1;.05}^2$.

### 2.5.4 Nemenyi test

*Nemenyi test* [10, 35, 47] is a post-hoc pairwise multiple comparisons procedure aiming to determine which algorithms are different, since Friedman rank test seen previously only tells us whether at least one of the algorithms differs from at least one other algorithms. Note that this test is relevant only after significant results of the Friedman rank test. We apply the Nemenyi test to the set of mean ranks resulting from the Friedman's test using the following procedure:

- Order the algorithms according to ascending mean rank
- Given the level of significance $\alpha$ of 5%, $k$ algorithms and $n$ datasets, decide that algorithm, $\tau_u \neq \tau_v$ if

$$|r_u - r_v| \geq q_{.05}\left[\frac{k(k+1)}{6n}\right]^{\frac{1}{2}} \tag{2.34}$$

  otherwise conclude $\tau_u = \tau_v$ , where $u = 1, \ldots, k$ , $v = 1, \ldots, k$, and $u \neq v$. The critical value $q_{.05}$ can be read in the table.

# 3 Investigated Graph embedding algorithms

In the previous chapter we presented all the necessary concepts needed throughout this work. In this chapter we present in details the investigated techniques. The embeddings that they will provide will be then used for the semi-supervised node classification task (see next chapter) and other evaluation tasks (see chapter 5).

## 3.1 DeepWalk: Online Learning of Social Representations

DeepWalk (DW) [51] is an algorithm that learns "*social representations*" of vertices in a network by modeling a stream of *short random walks*. Social representations are latent features that capture *neighborhood similarity* (or *neighborhood structure*) and *community membership* (high-order node proximity).
These latent representations encode social relations in a continuous vector space with a relatively small number of dimensions, which is easily exploited by statistical models. Thus, DW takes a graph as input and produces a latent representation as an output. DW is one of the first techniques that introduced deep learning (unsupervised learning feature) techniques, which have been proven successful in Natural Language Processing (NLP), for graph embedding. By analogy to NLP, a random walk is equivalent to a *sentence* and a node to a *word*.

### 3.1.1 Problem definition

Given a (partially) labeled social network $G = (\mathcal{V}, \mathcal{E}, \mathbf{X}, \mathbf{Y})$, with attributes $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}| \times s}$ where $s$ is the size of the feature space for each attribute vector, and $\mathbf{Y} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{Y}|}$, $\mathcal{Y}$ is the set of labels. The goal is to learn $\mathbf{X}_E \in \mathbb{R}^{|\mathcal{V}| \times d}$, where $\mathbf{X}_E$ is a latent social representation and $d$ is small number of latent dimensions. We denote a random walk rooted at vertex $v_i$ as $\wp_i$. It is a stochastic process with random variables $\wp_i(t)$ (node on $\wp_i$ at position $t = 0, 1, \ldots$) such that $\wp_i(t+1)$ is a vertex chosen at random from the neighbors of the last visited vertex. From the Fig. 3.1(a), a possible random walk of length 8 rooted at $v_4$ could be $\wp_4 = v_4 \rightarrow v_3 \rightarrow v_1 \rightarrow v_5 \rightarrow v_1 \rightarrow v_6 \rightarrow v_1 \rightarrow v_9$.

### 3.1.2 Language Modeling

*Language modeling* aims to estimate the likelihood of a specific sequence of words appearing in a corpus. More formally, given a **sequence of words** $W_1^n = (w_0, w_1, \ldots, w_n)$ where $w_i \in \mathcal{V}$ ($\mathcal{V}$ is the vocabulary), we would like to maximize the

$$P(w_n | w_0, w_1, \ldots, w_{n-1}) \tag{3.1}$$

over all the training corpus. The direct analog is to estimate the likelihood of observing vertex $v_i$ given all the previous vertices visited so far in the random walk:

$$P(v_i | v_1, v_2, \ldots, v_{i-1}) \tag{3.2}$$

Given a sequence of vertices $S = \{v_1, v_2, \ldots, v_k\}$ generated by a random walk of length $k$, we regard the vertices $v \in \{v_{i-w}, \ldots, v_{i-1}, v_{i+1}, \ldots, v_{i+w}\} \backslash \{v_i\}$ as **the context** of the center vertex $v_i$, where $w$ is the window size [57]. Thus, the goal of DW is to learn a latent representation, not only a probability distribution of node co-occurrences, and so we introduce a mapping function $\mathbf{\Phi} : \mathcal{V} \mapsto \mathbb{R}^{|\mathcal{V}| \times d}$. The problem then, is to estimate (maximize) the likelihood

$$P(v_i | (\Phi(v_1), \Phi(v_2), \ldots, \Phi(v_{i-1}))) \tag{3.3}$$

Note that when the walk length grows, computing this quantity becomes unfeasible. Based on [41, 42], two relaxations have been done on the model:

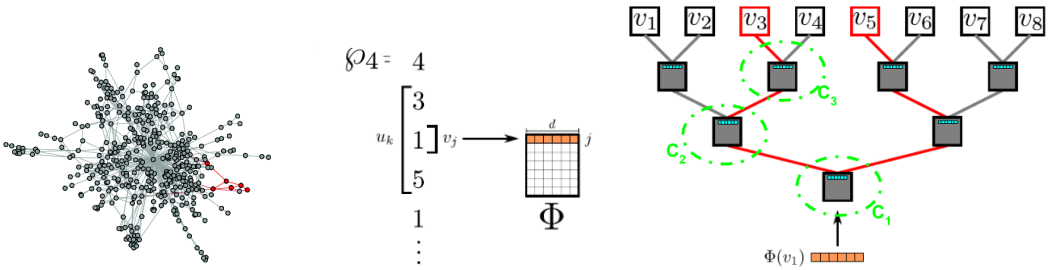
- First, instead of using the context to predict a missing word, it uses one word to predict the context.
- Secondly, the context is composed of the words appearing to the right side of the given word as well as the left side. So, it removes the ordering constraint on the problem.

From these relaxations, the model is required to maximize the probability of any word appearing in the context without the knowledge of its offset from the given word. Therefore, in terms of vertex representation modeling, this yields the optimization problem:

$$\min_{\Phi} -\log \mathrm{P}(\{v_{i-w}, \ldots, v_{i-1}, v_{i+1}, \ldots, v_{i+w}\} | \Phi(v_i)) \tag{3.4}$$

Thanks to the independence assumption on vertices, the objective becomes

$$\min_{\Phi} \sum_{-w \leq j \leq w} -\log \mathrm{P}(v_{i+j} | \Phi(v_i)) \tag{3.5}$$




(a) Random walk generation

(b) Representation mapping

(c) Hierarchical Softmax

**Fig. 3.1.:** Overview of DeepWalk. We slide a window of length $2w + 1$ over the random walk $\wp_4$ , mapping the central vertex $v_1$ to its representation $\Phi(v_1)$. Hierarchical Softmax factors out $P(v_3|\Phi(v_1))$ and $P(v_5|\Phi(v_1))$ over sequences of probability distributions corresponding to the paths starting at the root and ending at $v_3$ and $v_5$ . The representation $\Phi$ is updated to maximize the probability of $v_1$ co-occurring with its context $\{v_3, v_5\}$ [51].

where $\Phi(v_i)$ is the current representation of $v_i \in \mathbb{R}^d$ and $\mathrm{P}(v_{i+j}|\Phi(v_i))$ is defined using the softmax function

$$\mathrm{P}(v_{i+j}|\Phi(v_i)) = \frac{\exp(\Phi^T(v_{i+j})\Phi(v_i))}{\sum_{k=1}^{|\mathcal{V}|}\exp(\Phi^T(v_k)\Phi(v_i)))} \tag{3.6}$$

DW uses the *SkipGram* [51] model (see Algorithm 2) to update these representations in accordance with the objective function in equation 3.4. Thus, solving the problem 3.5 builds representations that capture the shared similarities in local graph structure between vertices. Vertices which have *similar neighborhoods* will acquire *similar representations*. However, computing the full softmax in equation 3.6 is not feasible as the normalization factor (summation over all inner product with every node in a graph) is expensive. In order to approximate it and speed up the training time, *Hierarchical Softmax*[1] [44, 45] is used. For more details see the reference paper [51].

As illustrate in Algorithm 1, DW algorithm consists of two main components; first a random walk generator and second an update procedure. A walk samples uniformly from the neighbors of the last vertex visited until the maximum length $t$ is reached. Finally, the node representation is given by the matrix $\mathbf{\Phi}$.

For the sake of experimentation, we will use DW in two ways, DWg for which we fixed the hyper-parameter $t$ while varying $\gamma$, and DWt which is exactly the opposite of the fist one.

[1] A short overview is given in Appendix.

**Algorithm 1** DEEPWALK($G$,$w$,$d$,$\gamma$,$t$)

**Input:** graph($G$,$\mathcal{V}$,$\mathcal{E}$)
window size $w$
embedding size $d$
walks per vertex $\gamma$
walk length $t$
**Output:** matrix of vertex representations $\mathbf{\Phi} \in \mathbb{R}^{|\mathcal{V}| \times d}$

Initialization: Sample $\mathbf{\Phi}$ from $\mathcal{U}^{|\mathcal{V}| \times d}$
Build a binary Tree $T$ from $\mathcal{V}$
**for** $i = 0$ to $\gamma$ **do**
    $\mathcal{O}$=Shuffle($\mathcal{V}$)
    **for each** $v_i \in \mathcal{O}$ **do**
        $\wp_i = RandomWalk(G, v_i, t)$
        SkipGram($\mathbf{\Phi}$, $\wp_i$, $w$)

**Algorithm 2** SkipGram($\mathbf{\Phi}$, $\wp_i$, $w$)

**for each** $v_j \in \wp_i$ **do**
    **for each** $u_k \in \wp_i[j\text{-}w{:}j{+}w]$ **do**
        $J(\mathbf{\Phi}) = -\log P(u_k|\Phi(v_j))$
        $\mathbf{\Phi} = \mathbf{\Phi} - \alpha * \frac{\partial J}{\partial \mathbf{\Phi}}$

## 3.2 Matrix Factorization of DeepWalk

### 3.2.1 Problem definition

DW seen in the section 3.1, is an algorithm that learns social representations of a graph's vertices by modeling a stream of short random walks. Indeed, [57] proved that DW actually factorizes a matrix $\mathbf{M}$, the so-called *vertex-context* or *co-occurrence matrix*. Let $\mathcal{D}$ be a vertex-context set generated from random walk sequences where each member of $\mathcal{D}$ is a vertex-context pair $(v, c)$. $N(v, c)$ denotes the number of times that $(v, c)$ appears in $\mathcal{D}$. $N(v) = \sum_{c' \in \mathcal{V}_C} (v, c')$ and $N(c) = \sum_{v' \in \mathcal{V}} N(v', c)$ denote the numbers of times $v$ and $c$ appear in $\mathcal{D}$, respectively. $\mathcal{V}$ is the set of vertices and $\mathcal{V}_C$ is the set of context vertices. In most cases, $\mathcal{V}$=$\mathcal{V}_C$. Each entry in $\mathbf{M}$ is formalized as

$$m_{ij} = \log \frac{[\mathbf{e}_i^T(\mathbf{P} + \mathbf{P}^2 + \cdots + \mathbf{P}^t)]_j}{t} = \log \frac{N(v_i, v_j)}{N(v_i)} \tag{3.7}$$

where $[\mathbf{e}_i^T(\mathbf{P} + \mathbf{P}^2 + \cdots + \mathbf{P}^t)]_j$ is the expectation times that $v_j$ appears in right $t$ neighbors of $v_i$, $\mathbf{P} = \mathbf{D}^{-1}\mathbf{A}$ is the transition matrix of the natural random walk on the graph and $\boldsymbol{e}_i^T$ a $|\mathcal{V}|$-dimensional row vector where all entries are 0 except the $i$-th entry is 1. The entry $m_{ij}$ is logarithm of the average probability that vertex $i$ walks to vertex $j$ in $t$ steps.

### 3.2.2 Proximity Matrix Construction

Computing an accurate $\mathbf{M}$ from equation 3.7 has a complexity of $O(|\mathcal{V}|^3)$ when $t$ gets large. In fact, DW uses a sampling method based on random walk to avoid explicitly computing matrix $\mathbf{M}$. When DW samples more walks, the performance will be better while it will be less efficient.

Interestingly, [57] found out the trade-off between speed and accuracy by simply factorizing the matrix $\mathbf{M}$ as

$$\mathbf{M} = \frac{\mathbf{P} + \mathbf{P}^2}{2} \tag{3.8}$$

Here, we factorize $\mathbf{M}$ instead of $\log \mathbf{M}$ for computational efficiency. The reason is that $\log \mathbf{M}$ has much more non-zero entries than $\mathbf{M}$, and the complexity of matrix factorization with square loss is proportional to the number of non-zero elements in matrix $\mathbf{M}$ [58].

### 3.2.3 Optimization

In this work, we factorized the matrix $\mathbf{M}$ in equation 3.8 using the *Singular Value Decomposition* (SVD) [22] approach. Indeed, SVD factorizes matrix $\mathbf{M}$ into the product of three matrices $\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, where $\mathbf{M} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$, $\mathbf{U} \in \mathbb{R}^{|\mathcal{V}|\times k}$, $\mathbf{\Sigma} \in \mathbb{R}^{k\times k}$ and $\mathbf{V} \in \mathbb{R}^{k\times|\mathcal{V}|}$. $\mathbf{U}$ and $\mathbf{V}$ are orthonormal matrices and $\mathbf{\Sigma}$ is diagonal matrix with positive real entries. The node representation is given by $\mathbf{U}\mathbf{\Sigma}$.

## 3.3 t-Distributed Stochastic Neighbor Embedding

*t-Distributed Stochastic Neighbor Embedding* (tSNE) [37] is a new technique for visualizing high-dimensional data in two or three dimensions. tSNE is capable of capturing much of the local structure of high-dimensional data very well, as well as revealing global structure such as presence of clusters at several scales. In the context of this work, tSNE will take the free energy distance matrix as input and will produce a latent representation of either 2 or 3 dimensions as an output.

### 3.3.1 Problem definition

Given a set of $n$ high-dimensional data $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$, tSNE first computes the symmetrized conditional probabilities $p_{ij}$ that are proportional to the pairwise similarities as follows:

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}, \qquad (p_{ii} = 0, \forall i) \tag{3.9}$$

where $p_{j|i}$ is the *similarity* of datapoint $\boldsymbol{x}_j$ to datapoint $\boldsymbol{x}_i$, that is the conditional probability that $\boldsymbol{x}_i$ would pick $\boldsymbol{x}_j$ as its neighbor if neighbors were picked in proportion to their probability density under a Gaussian centered at $\boldsymbol{x}_i$. Mathematically, the conditional probability $p_{j|i}$ is given by:

$$p_{j|i} = \frac{\exp(-||\boldsymbol{x}_i - \boldsymbol{x}_j||^2/2\sigma_i^2)}{\sum_{k\neq i}\exp(-||\boldsymbol{x}_i - \boldsymbol{x}_k||^2)/2\sigma_i^2)}, \qquad (p_{i|i} = 0, \ \ \forall i) \tag{3.10}$$

where $k$ is the effective number of the local neighbors, $\sigma_i^2$ is the variance of the Gaussian that is centered on datapoint $\boldsymbol{x}_i$. In the low-dimensional map, Student t-distribution with one degree of freedom $(\nu = 1)$ is used to convert distances into probabilities. Similarly, the similarity $q_{ij}$ between two points $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ is computed as follows

$$q_{ij} = \frac{(1 + ||\boldsymbol{y}_i - \boldsymbol{y}_j||^2)^{-1}}{\sum_{k\neq l}(1 + ||\boldsymbol{y}_k - \boldsymbol{y}_l||^2)^{-1}} \tag{3.11}$$

$\boldsymbol{y}_i$'s are the new coordinates of datapoints in the 2 or 3-dimensional embedding space. We refer to this type of SNE as *symmetric SNE*, because $p_{ij} = p_{ji}$ and $q_{ij}$=$q_{ji}$ for $\forall i, j$.

### 3.3.2 Perplexity and Variance of the Gaussian

For tSNE, the *perplexity* (see section 2.4) can be interpreted as a smooth measure of the effective number of neighbors and balances the local and global aspects of the dataset. Any particular value of $\sigma_i^2$ induces a probability distribution, $P_i$, over all of the other datapoints. This distribution has an entropy which increases as $\sigma_i^2$ increases. tSNE performs a binary search for the value of $\sigma_i^2$ that produces a $P_i$ with a fixed perplexity that is specified by the user, knowing that the perplexity increases monotonically with the variance $\sigma_i^2$. Note that tSNE is more sensitive to the change in perplexity, thus, the authors suggested to use a typical value between 5 and 50.

### 3.3.3 Optimization

The cost function of tSNE is a *single Kullback-Leibler divergence* (KL) between two joint probability distributions, $P$ in the high-dimensional space and the Student-t based joint probability distribution $Q$ in the low-dimensional space:

$$C = KL(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}} \tag{3.12}$$

**Algorithm 3** Simple version of t-Distributed Stochastic Neighbor Embedding

**Input:** dataset $\mathcal{X}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_n\}$
cost function parameters: perplexity $Perp$,
optimization parameters: number of iterations $T$, learning rate $\eta$,
momentum $\alpha(t)$.
**Output:** low-dimensional data representation $\mathcal{Y}^{(T)} = \{\boldsymbol{y}_1,\boldsymbol{y}_2,\ldots,\boldsymbol{y}_n\}$.

1: compute pairwise affinities $p_{j|i}$ with perplexity $Perp$ (using equation 3.10)
2: set $p_{ij} = \frac{p_{j|i}+p_{i|j}}{2n}$
3: sample initial solution $\mathcal{Y}^{(0)} = \{\boldsymbol{y}_1,\boldsymbol{y}_2,\ldots,\boldsymbol{y}_n\}$ from $\mathcal{N}(0;10^{-4}I)$
4: **for** $t = 1$ to $T$ **do**
5: compute low-dimensional affinities $q_{ij}$ (using equation 3.11)
6: compute gradient $\frac{\delta C}{\delta \mathcal{Y}}$ (using equation 3.13)
7: $\mathcal{Y}^{(t)} = \mathcal{Y}^{(t-1)} + \eta\frac{\delta C}{\delta \mathcal{Y}} + \alpha(t)(\mathcal{Y}^{(t-1)} - \mathcal{Y}^{(t-2)})$

thus, the gradient of the KL is given by

$$\frac{\delta C}{\delta y_i} = 4\sum_j (p_{ij} - q_{ij})(y_i - y_j)(1 + ||y_i - y_j||^2)^{-1} \tag{3.13}$$

The gradient descent is initialized by sampling map points randomly from an isotropic Gaussian with small variance that is centered around the origin. Pseudo code for tSNE is presented in Algorithm 3. This simple procedure uses a momentum term to reduce the number of iterations required and it works best if the momentum term is small until the map points have become moderately well organized.

Interestingly, tSNE can also be applied to datasets that consists of pairwise similarities between objects rather than high-dimensional vector representations of each object. In this work, we applied tSNE on the free energy distance matrix (addressed in section 2.3.2), which in itself represents similarities between nodes of the graph (see the procedure in chapter 4).

However, despite the good performance of tSNE compared to other techniques for data visualization like SNE, Isomap, etc., the behavior of tSNE when reducing data to two or three dimensions cannot readily be extrapolated to $d > 3$ dimensions because of the heavy tails of the Student t-distribution. The authors suggested to set $\nu = d-1$ (see [36]). So, for dimensionality higher than three[2], Student t-distributions with more than one degree of freedom are likely to be more appropriate. For more details see the reference paper [37].

[2]In practice, one degree of freedom works well up to three dimensions.

# 4 Experiments: semi-supervised node classification tasks

In the previous chapter, we presented DW, MFDW and tSNE, which are in comparison with the baselines methods addressed in chapter 2. Now, we are going to assess the performances of all these methods on semi-supervised node classification task using SVM classifier and sum-of-similarities kernel fashions (see sections 2.2.3 and 2.2.4). At each step, the appropriate statistical tests and procedures (see section 2.5) are performed to validate the results. In addition to the classification tasks, we will address other ways of assessing the quality of the embedding in the next chapter.

## 4.1 Datasets

In order to comprehensively evaluate the effectiveness of each embedding technique, we use 14 well-known graph datasets, used also in [17, 26]. **WebKB** (4 datasets) come from networks of cocitation between webpages of computer science departments of 4 different american universities, **Newsgroup** (9 subsets) consists of 20.000 documents taken from 20 discussion groups of the Usenet diffusion list, and **Imdb** which comes from the well-known Internet Movie Database. Basic statistics are given in Tab. 4.1.

| Dataset | #Nodes | #Edges | #labels |
|---|---|---|---|
| WebKB-texas | 334 | 32988 | 6 |
| WebKB-washington | 434 | 30462 | 6 |
| WebKB-wisconsin | 348 | 33250 | 6 |
| WebKB-cornell | 346 | 26832 | 6 |
| Imdb | 1169 | 40564 | 2 |
| News-2cl-1 | 400 | 67708 | 2 |
| News-2cl-2 | 398 | 42960 | 2 |
| News-2cl-3 | 399 | 73054 | 2 |
| News-3cl-1 | 600 | 141182 | 3 |
| News-3cl-2 | 598 | 136402 | 3 |
| News-3cl-3 | 595 | 128338 | 3 |
| News-5cl-1 | 998 | 353924 | 5 |
| News-5cl-2 | 999 | 328904 | 5 |
| News-5cl-3 | 997 | 311236 | 5 |

**Tab. 4.1.:** Basic statistics on fourteen datasets. WebKB (4 datasets), Newsgroup (9 subsets) and Imdb.

## 4.2 Experimental settings and Methodology

In this experiment, we address the task of classifying unlabeled nodes in partially labeled graphs. In other words, the classification model predicts the labels of unlabeled nodes and its prediction is compared to the true labels which were hidden. To avoid any possible confusion, all the techniques as well as their abbreviations are recapped in Tab. 4.2.

| Method | Description |
|---|---|
| BoPP-m | multidimensional scaling of the free energy distance matrix* |
| BoPP-g | Gaussian kernel of the free energy distance matrix* |
| SoS | sum-of-similarities provided by the bounded normalized random walk with restart. Only the classification accuracies are returned. |
| Q | modularity matrix* |
| CovH | covariance of node co-presences on hitting paths* |
| NCorH | correlation of node co-occurrences on hitting paths* |
| DW | DeepWalk embedding obtained using the original implementation [9] and then injected as input feature into a SVM classifier |
| DW-s | sum-of-similarities classification (see section 2.2.3) based on the kernel constructed from the DW embedding data |
| MFDW | matrix factorization of DW. The embedding is obtained by SVD approach and then injected as input feature into a SVM classifier |
| MFDW-s | sum-of-similarities classification based on the kernel constructed from the MFDW embedding data |
| tSNE | embedding obtained by applying tSNE algorithm on the free energy distance matrix and then injected as input feature into a SVM classifier. The embedding is obtained using the original implementation [54]. |
| tSNE-s | sum-of-similarities classification based on the kernel constructed from the tSNE embedding data |

**Tab. 4.2.:** Description of the various classification methods. *The embedding is obtained by extracting the five (or two) dominant eigenvectors of this kernel and then injected as features into a SVM classifier.

We use the same experimental methodology as in [17, 26]. The number of social dimensions has been set to 5 for all the methods except 3 dimensions for tSNE. For our baseline techniques, we extracted the five dominant eigenvectors from the resulting kernels (see section 2) while for DW and tSNE the embeddings have been directly obtained from the original implementations, by proving a set of suitable hyper-parameters, described below.

In order to reduce variance in accuracy, methods are tested on five repetitions (runs) of a standard nested cross-validation methodology. Each cross-validation contains 5 folds, and methods are tested with a labeling rate of 20%. To tune hyper parameters, an internal 5-fold cross-validation on the training fold is performed, by taking 4/5 of the training fold as labeled and 1/5 as unlabeled. For the SVM, the penalization constant $c$ is tuned inside $\{10^{-2}, 10^{-1}, 1, 10, 100\}$ and we used LIBLINEAR, which is a *linear classifier* for data with *millions* of instances and features [14].

About the hyper-parameters, the bag-of-paths methods (BoPP-m, BoPP-g, CovH, NCorH, tSNE) investigate tuning values of $\theta = \{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 0.1, 1, 10\}$, for the SoS method $\alpha = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$. Specially for DW, we sampled nodes according to probabilities computed from the edge weights instead of uniformly sampling from the neighbors of the last vertex visited, and the learning rate $\alpha$, the window size $w$, the walk length $t$ and the number of walks per node $\gamma$ were set to 0.25, 10, 40 and 50, respectively. However, there is no hyper-parameter for the modularity matrix Q while for MFDW the order $k$ was tuned in $\{2, 3, 5, 10, 20, 40, 50\}$ and we used the adjacency matrix[1] in the factorization since our graphs are dense (see the reference paper [57]).

We warn you again that we applied tSNE algorithm on the free energy distance matrix. So, we proceeded in two stages: Firstly, we identified the required precision $(\frac{1}{\sigma^2})$ to obtain a Gaussian kernel (pairwise affinities) in the high-dimensional space with a certain uncertainty for every datapoint. This desired uncertainty was specified through the perplexity and obtained up to a tolerance of $10^{-4}$. Once the Gaussian kernel was found, we computed the symmetrized conditional probability matrix **P**. Secondly, we performed symmetric t-SNE on the matrix **P** to create a low-dimensional representation (embedding) in 2 or 3 dimensions. For the classification tasks we tuned the perplexity value in $\{5, 20, 30, 40, 50\}$ whereas for the other evaluations tasks (see chapter 5) we set it to 50. Also, the number of gradient descent iterations $T$ was set to 1000, the momentum term $\alpha^{(t)} = 0.5$ for $t < 250$ and $\alpha^{(t)} = 0.8$ for $t \geq 250$, and the learning rate $\eta$ set to 100.

## 4.3 Results and discussion

Tab. 4.3 reports average classifications accuracies of each method across all the datasets. Afterwards, a simple Borda ranking of the methods is performed (see Tab. 4.4). From these two tables, it can be observed that DW is ranked first, in comparison to the other methods. Indeed, NCorH, BoPP-g and CovH consistently provided good results even if they do not have an explicit objective function trying to optimize in order to capture the network structure. Moreover, the differences in performance among the best performing methods is small. For instance, for the five best techniques (DW, NCorH, DW-s, BoPP-g, CovH; see Tab. 4.4), the average difference between the accuracy (see Tab. 4.3) of DW and the other methods across all datasets is 0.46 and the maximum difference is only 6.96 (versus CovH for News-5cl-2 dataset). Besides, we noticed that the best method is **dataset-dependent**; that is why is it useful to investigate different methods when facing a network-based

[1] This yielded even better results than the factorization with the transition probability matrix **P**. We did the same thing for all the other experiments.

| Classification Method : Dataset : | BoPP-m | BoPP-g | SoS | Q | CovH | NCorH | DW | DW-s | MFDW | MFDW-s | tSNE | tSNE-s |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| WebKB-texas | 73.88 | 75.89 | 74.48 | 74.33 | 74.70 | **77.55** | 76.20 | 75.30 | 64.82 | 58.76 | 67.96 | 67.89 |
| webKB-washington | 70.97 | **72.86** | 66.19 | 59.51 | 66.36 | 67.40 | 70.68 | 70.68 | 58.18 | 56.34 | 64.52 | 63.31 |
| WebKB-wisconsin | 65.60 | 72.99 | 73.78 | 72.70 | 74.78 | 73.06 | **75.00** | 72.63 | 73.28 | 65.73 | 65.16 | 65.80 |
| WebKB-cornell | 52.82 | 57.23 | 59.03 | 51.01 | 57.66 | 59.03 | 57.37 | **60.04** | 49.57 | 45.88 | 53.76 | 55.78 |
| Imdb | 75.40 | 75.18 | **78.13** | 68.18 | 77.00 | 75.88 | 74.11 | 73.89 | 72.74 | 72.25 | 76.51 | 77.53 |
| News-2cl-1 | 83.13 | 96.56 | 93.00 | 94.69 | 96.75 | **97.25** | 96.88 | 96.81 | 96.19 | 95.25 | 92.50 | 95.00 |
| News-2cl-2 | 79.71 | 90.27 | 89.83 | 91.08 | 91.58 | 90.90 | 91.46 | 91.77 | 92.53 | 92.40 | 92.97 | **94.47** |
| News-2cl-3 | 95.36 | 95.74 | 94.30 | 94.11 | 95.55 | 96.68 | 96.68 | 96.56 | 94.99 | 95.99 | 96.37 | **97.49** |
| News-3cl-1 | 93.58 | 93.92 | 90.96 | 93.21 | **94.67** | 93.33 | 93.54 | 92.79 | 93.92 | 93.63 | 92.83 | 93.21 |
| News-3cl-2 | 92.98 | 92.85 | 90.26 | 93.40 | 92.77 | **93.60** | 93.02 | 92.98 | 93.10 | 93.06 | 87.50 | 90.18 |
| News-3cl-3 | **93.61** | 93.40 | 91.55 | 90.97 | 91.64 | 91.47 | 93.36 | 92.98 | 90.25 | 84.96 | 90.29 | 89.96 |
| News-5cl-1 | 88.00 | 88.03 | 86.77 | 78.26 | 85.50 | 86.77 | 88.08 | **88.38** | 79.99 | 75.43 | 86.22 | 86.90 |
| News-5cl-2 | 81.46 | 78.65 | **82.73** | 76.33 | 73.70 | 76.33 | 80.66 | 80.83 | 69.19 | 67.52 | 77.53 | 79.43 |
| News-5cl-3 | 80.72 | 80.34 | **81.62** | 75.85 | 78.66 | 76.76 | 78.16 | 78.16 | 69.71 | 67.75 | 70.36 | 69.99 |

**Tab. 4.3.:** Classification accuracies in percent for the various classification methods obtained on the different datasets using 5 social dimensions (3 for tSNE). The best performing method is highlighted in boldface for each dataset.

semi-supervised classification problem. Furthermore, in order to rate globally the

| Method | Score | Rank |
|---|---|---|
| DW | 127 | 1 |
| NCorH | 116 | 2 |
| DW-s | 115 | 3 |
| BoPP-g | 113 | 4 |
| CovH | 106 | 5 |
| SoS | 95 | 6 |
| BoPP-m | 93 | 7 |
| tSNE-s | 85 | 8 |
| MFDW | 69 | 9 |
| tSNE | 68 | 10 |
| Q | 62 | 11 |
| MFDW-s | 52 | 12 |

**Tab. 4.4.:** Ranking of the different classification methods (see Tab. 4.3) according to Borda's method (the higher the score, the better).

results of each method across all the datasets, we run first the Friedman test. The null hypothesis was rejected due to a p-value of 0.0002, which means that at least one of the methods differs from at least one other methods. This conclusion allowed us to conduct a multiple comparison with the Nemenyi test depicted in the Fig. 4.1. Indeed, we can see that the Nemenyi test confirms that the DW provided good results, which are significantly superior to the results obtained by the modularity matrix Q and MFDW-s. These results are similar to those provided by the Borda ranking.

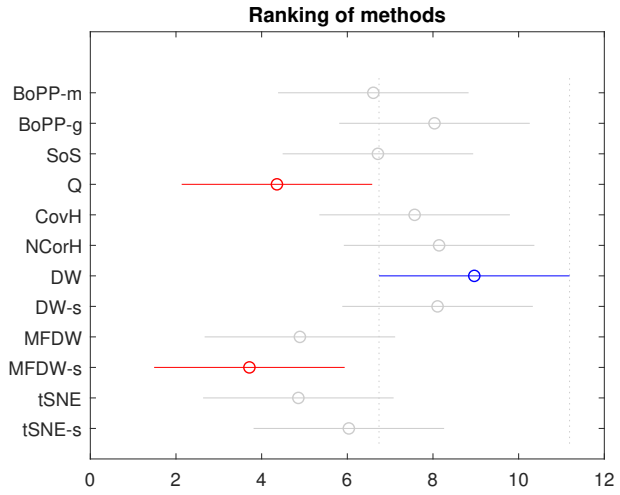



**Fig. 4.1.:** Mean ranks and 95% Nemenyi confidence intervals for the 12 methods (see Tab. 4.3). Two methods are considered significantly different if their confidence intervals do not overlap. The worse methods (Q, MFDW-s) and the best method (DW) overall are highlighted. (p-value Friedman test=0.0002)

We also performed the right-sided Wilcoxon test to obtain more precise information concerning the relative performance of the methods. As reported in Tab. 4.5: First,

DW and DW-s are not significantly different from BoPP-m, BoPP-g, SoS, CovH and NCorH; whereas they are significantly different from Q, MFDW, MFDW-s, tSNE and tSNE-s. Second, BoPP-g is better than BoPP-m, Q, MFDW, MFDW-s, tSNE and tSNE-s; while BoPP-m did not beat any other method. Third, CovH and NCorH are significantly different from Q, MFDW, MFDW-s and tSNE. Also, tSNE is only better than MFDW-s while tSNE-s is better than MFDW-s and tSNE.

An important observation to emphasize, having a look at Tab. 4.4 and 4.5, we see that the Borda ranking is only informative (gives a partial view of the information). Therefore, if one method is less performing than another, this statement is valid as soon as the Wilcoxon test gives a significant p-value, otherwise this ranking could be misleading. This is the case, e.g. for DW, ranked first whereas not significantly different from the four other best methods. Besides, Borda ranking is some times less informative, usually when the performances of the methods are tight.

However, it is crucial to check whether the local structures (groups or classes) in the graph are preserved (still distinguishable) because a good graph embedding method should ensure that the learned embeddings can preserve the original network structure. As depicted in Fig. 4.2, Heatmap plots are made up from various kernels constructed from the embedding matrices[2]. We only reported the HeatMaps for the News-3cl-1 dataset, where nodes have been sorted according to class labels. Indeed, there is a good class discrimination for BoPP-m, BoPP-g, DW, tSNE and Q, while some overlaps more precisely on class 1 and clear discrimination between classes 2 et 3 for MFDW, CovH and NCorH. Note that we will develop the quality of the embedding widely in the next chapter.

Finally, taking a look on different results found so far, you can notice that there is a bias we voluntarily introduced since were are evaluating the embeddings into three dimensions (3D) for tSNE equally with the ones into five dimensions (5D). To equilibrate, we repeated the same experimental protocol considering now only two dimensions (2D) for all the methods.

[2]Remember that a kernel $\mathbf{K}$ is constructed from the embedding matrix $\mathbf{X}$ as $\mathbf{K}=\mathbf{X}\mathbf{X}^{\mathbf{T}}$

| Method | Score | Rank |
|---|---|---|
| tSNE-s | 132 | 1 |
| BoPP-m | 108 | 2 |
| NCorH | 99 | 3 |
| tSNE | 98 | 4 |
| CovH | 94 | 5 |
| BoPP-g | 93 | 6 |
| Q | 81 | 7 |
| DW | 78 | 8 |
| DW-s | 63 | 9 |
| MFDW | 50 | 10 |
| MFDW-s | 33 | 11 |

**Tab. 4.7.:** Ranking of the different classification methods for 2D embedding (see Tab. 4.6) according to Borda's method (the higher the score, the better)

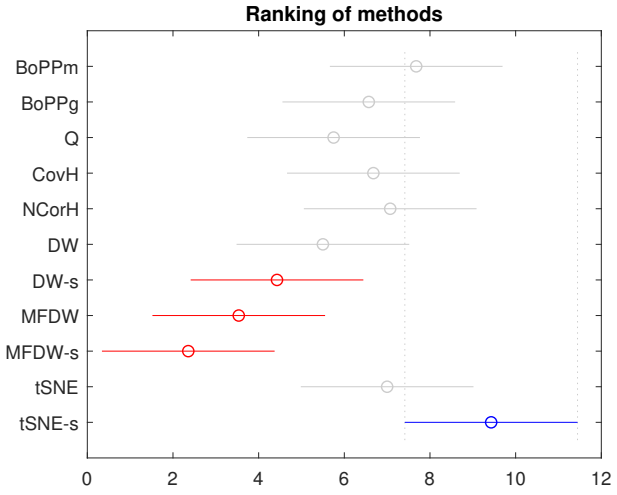



**Fig. 4.3.:** Mean ranks and 95% Nemenyi confidence intervals for the 11 methods (see Tab. 4.6). Two methods are considered significantly different if their confidence intervals do not overlap. The worse methods (DW-s, MFDW, MFDW-s) and the best method (tSNE-s) overall are highlighted. (p-value Friedman test=2.1379e-07)

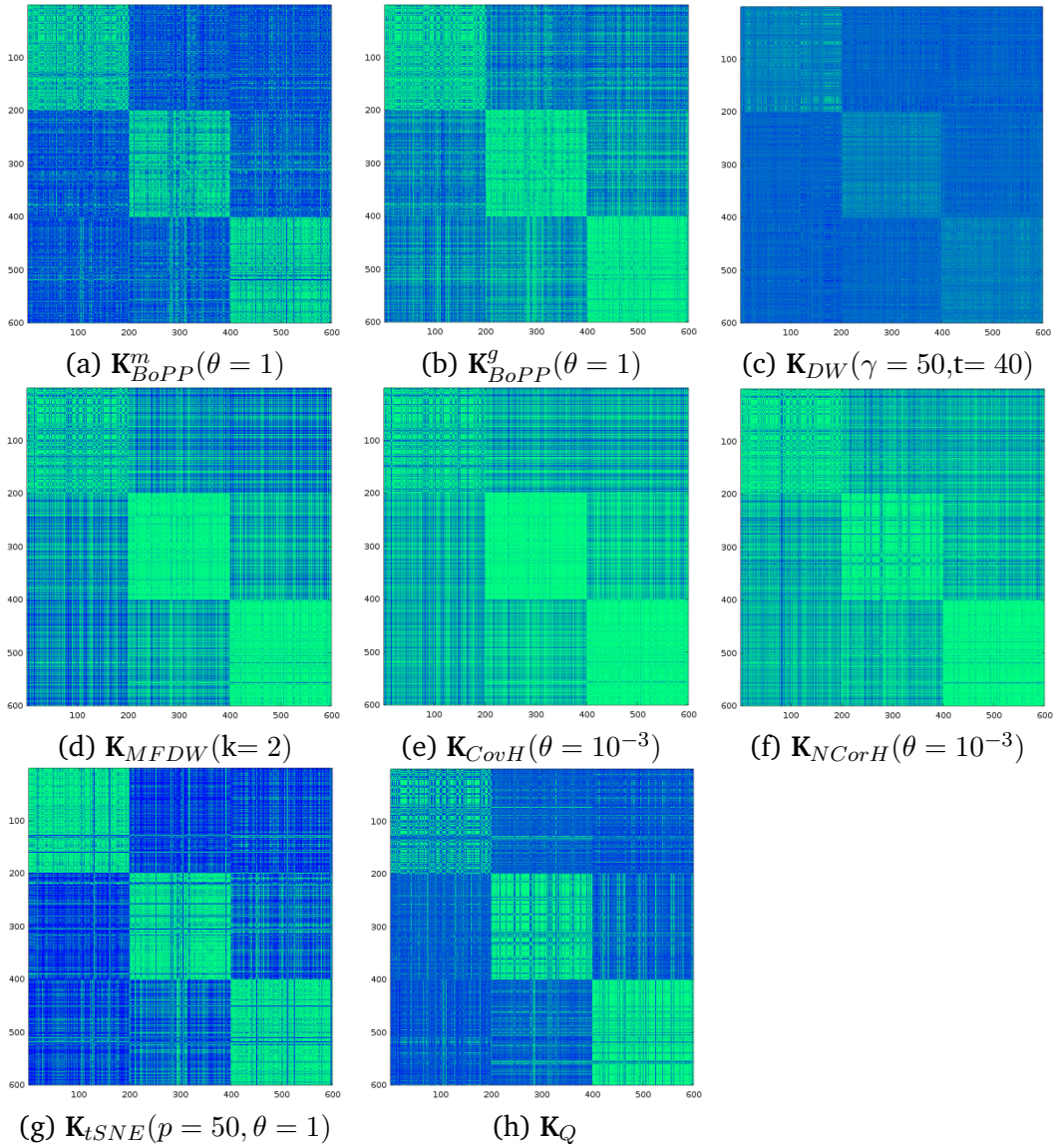
From Tab. 4.6 and Tab. 4.7, we can see that tSNE-s got the highest score in comparison with the other methods, while tSNE and BoPP-m are more competitive than CovH and NCorH. Moreover, the differences in performance among the best performing methods are considerable. For instance, for the five best techniques (tSNE-s, BoPP-m, NCorH, tSNE, CovH; see Tab. 4.7), the average difference between the accuracy of tSNE-s and the other methods across all datasets is 4.64 and the maximum difference is 32.29 (versus NCorH for News-5cl-1 dataset), the gap is substantial. The Nemenyi test (see Fig. 4.3) confirms that tSNE-s provided good results, which are significantly superior to the results obtained by the DW-s, MFDW and MFDW-s. Besides, the Wilcoxon test in Tab. 4.8 reports that, on one hand, tSNE-s is better than all the other methods, except BoPP-m. On the other hand, BoPP-g,

NCorH and tSNE are significantly different from DW, DW-s, MFDW and MFDW-s. Finally, CovH is only significantly different from DW-s, MFDW and MFDW-s.

Remember that tSNE algorithm is applied on the free energy distance matrix in order to extract 3D or 2D embedding because we do not have initially the feature matrix. So, it would have been better to analyse the behavior of tSNE by using other node similarity measures (e.g., Jaccard similarity). For the moment the question remains open since it goes beyond the scope of this thesis but does not prevent to be investigated later. Finally, concerning MFDW, we noticed that during the tuning process, the optimal value of $k$ was 2 in the most of the cases.

| | BoPP-m | BoPP-g | SoS | Q | CovH | NCorH | DW | DW-s | MFDW | MFDW-s | tSNE | tSNE-s |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BoPP-m | 1.0000 | 0.9562 | 0.8212 | 0.2316 | 0.7292 | 0.8371 | 0.8826 | 0.8594 | 0.1788 | 0.0676 | 0.1955 | 0.3747 |
| BoPP-g | **0.0468** | 1.0000 | 0.1629 | **0.0009** | 0.2177 | 0.3349 | 0.7036 | 0.4276 | **0.0031** | **0.0015** | **0.0034** | **0.0453** |
| SoS | 0.1955 | 0.8521 | 1.0000 | 0.0520 | 0.6871 | 0.7407 | 0.8662 | 0.8794 | **0.0392** | **0.0083** | **0.0290** | 0.1083 |
| Q | 0.7869 | 0.9994 | 0.9547 | 1.0000 | 0.9966 | 0.9998 | 0.9999 | 0.9994 | 0.4039 | **0.0290** | 0.5724 | 0.7513 |
| CovH | 0.2915 | 0.7914 | 0.3349 | **0.0043** | 1.0000 | 0.7492 | 0.9058 | 0.8917 | **0.0008** | **0.0009** | **0.0142** | 0.0863 |
| NCorH | 0.1788 | 0.6871 | 0.2847 | **0.0004** | 0.2708 | 1.0000 | 0.8781 | 0.8212 | **0.0026** | **0.0006** | **0.0123** | 0.0594 |
| DW | 0.1238 | 0.3077 | 0.1479 | **0.0002** | 0.0995 | 0.1367 | 1.0000 | 0.2349 | **0.0012** | **0.0012** | **0.0015** | **0.0209** |
| DW-s | 0.1486 | 0.5962 | 0.1338 | **0.0008** | 0.1206 | 0.1955 | 0.7881 | 1.0000 | **0.0054** | **0.0015** | **0.0020** | **0.0148** |
| MFDW | 0.8371 | 0.9977 | 0.9662 | 0.6196 | 0.9994 | 0.9980 | 0.9991 | 0.9957 | 1.0000 | **0.0009** | 0.8794 | 0.9324 |
| MFDW-s | 0.9406 | 0.9988 | 0.9933 | 0.9753 | 0.9994 | 0.9996 | 0.9991 | 0.9988 | 0.9994 | 1.0000 | 0.9763 | 0.9966 |
| tSNE | 0.8212 | 0.9974 | 0.9753 | 0.4516 | 0.9873 | 0.9899 | 0.9988 | 0.9985 | 0.1338 | **0.0258** | 1.0000 | 0.9933 |
| tSNE-s | 0.6370 | 0.9608 | 0.9031 | 0.2709 | 0.9235 | 0.9480 | 0.9824 | 0.9877 | 0.0765 | **0.0043** | **0.0083** | 1.0000 |

**Tab. 4.5.:** p-values of pairwise paired Wilcoxon rank test (right-sided) on classification accuracies (see Tab. 4.3). Level of significance $\alpha$=.05 and significant p-values are in boldface.



(a) $\mathbf{K}^{m}_{BoPP}(\theta = 1)$ (b) $\mathbf{K}^{g}_{BoPP}(\theta = 1)$ (c) $\mathbf{K}_{DW}(\gamma = 50$,t$= 40)$

(d) $\mathbf{K}_{MFDW}$(k$= 2)$ (e) $\mathbf{K}_{CovH}(\theta = 10^{-3})$ (f) $\mathbf{K}_{NCorH}(\theta = 10^{-3})$

(g) $\mathbf{K}_{tSNE}(p = 50, \theta = 1)$ (h) $\mathbf{K}_{Q}$

**Fig. 4.2.:** Images of the different similarity matrices computed on the News-3cl-1 dataset using 5 social dimensions (3 for tSNE). Nodes have been sorted according to classes.

| Classification Method : Dataset : | BoPP-m | BoPP-g | Q | CovH | NCorH | DW | DW-s | MFDW | MFDW-s | tSNE | TSNE-s |
|---|---|---|---|---|---|---|---|---|---|---|---|
| WebKB-texas | 59.58 | 61.91 | 62.05 | 58.91 | 62.80 | 59.28 | 58.31 | 59.88 | 50.60 | 65.64 | **66.24** |
| WebKB-washington | 61.24 | 64.57 | 59.33 | 65.67 | 65.61 | 66.01 | **66.88** | 59.22 | 39.69 | 60.43 | 59.91 |
| WebKB-wisconsin | 64.51 | 68.03 | **70.91** | 70.26 | 63.58 | 48.42 | 46.63 | 54.88 | 50.79 | 63.00 | 63.08 |
| WebKB-cornell | **53.83** | 47.04 | 47.62 | 49.57 | 51.88 | 48.92 | 47.91 | 44.51 | 43.50 | 51.45 | 53.47 |
| Imdb | 49.69 | 75.53 | 66.19 | 75.18 | 74.76 | 74.51 | 74.56 | 72.87 | 68.58 | 75.56 | **77.80** |
| News-2cl-1 | 97.13 | 97.25 | 96.31 | 96.31 | 95.75 | **98.00** | **98.00** | 94.38 | 85.19 | 91.94 | 96.63 |
| News-2cl-2 | 89.32 | 91.77 | 91.33 | 92.53 | 91.83 | 91.46 | 91.58 | 91.77 | 92.09 | 91.08 | **93.28** |
| News-2cl-3 | 95.49 | 95.49 | 95.74 | 95.30 | 95.99 | **97.24** | **97.24** | 93.80 | 93.67 | 94.86 | 96.43 |
| News-3cl-1 | **94.00** | 84.33 | 84.71 | 80.38 | 84.88 | 85.00 | 71.96 | 67.75 | 64.58 | 92.08 | 92.63 |
| News-3cl-2 | **93.23** | 88.13 | 84.49 | 86.87 | 88.42 | 84.62 | 71.87 | 83.53 | 79.64 | 87.42 | 89.26 |
| News-3cl-3 | **92.02** | 79.54 | 75.04 | 79.87 | 78.11 | 73.32 | 65.59 | 51.64 | 45.08 | 89.37 | 90.13 |
| News-5cl-1 | 78.28 | 54.96 | 58.92 | 58.19 | 53.56 | 49.22 | 42.86 | 55.91 | 55.61 | 84.70 | **85.85** |
| News-5cl-2 | 74.55 | 54.43 | 57.11 | 57.91 | 58.43 | 49.05 | 40.79 | 51.75 | 49.53 | 72.37 | **77.85** |
| News-5cl-3 | 60.41 | 52.28 | 57.07 | 46.82 | 52.13 | 42.95 | 38.84 | 39.37 | 37.76 | 58.68 | **65.85** |

**Tab. 4.6.:** Classification accuracies in percent for the various classification methods obtained on the different datasets using 2 social dimensions. The best performing method is highlighted in boldface for each dataset.

| | BoPPm | BoPPg | Q | CovH | NCorH | DW | DW-s | MFDW | MFDW-s | tSNE | tSNE-s |
|---|---|---|---|---|---|---|---|---|---|---|---|
| BoPPm | 1.0000 | 0.1219 | 0.0765 | 0.0969 | 0.1206 | 0.0520 | **0.0392** | **0.0123** | **0.0026** | 0.3129 | 0.8662 |
| BoPPg | 0.8918 | 1.0000 | 0.4039 | 0.5000 | 0.5484 | **0.0338** | **0.0101** | **0.0002** | **0.0003** | 0.9031 | 0.9917 |
| Q | 0.9324 | 0.6196 | 1.0000 | 0.7513 | 0.8212 | 0.2459 | **0.0338** | **0.0043** | **0.0004** | 0.9791 | 0.9988 |
| CovH | 0.9137 | 0.5242 | 0.2709 | 1.0000 | 0.6426 | 0.0520 | **0.0123** | **0.0002** | **0.0001** | 0.9406 | 0.9933 |
| NCorH | 0.8917 | 0.4758 | 0.1955 | 0.3804 | 1.0000 | **0.0101** | **0.0043** | **0.0006** | **0.0003** | 0.8794 | 0.9974 |
| DW | 0.9547 | 0.9710 | 0.7637 | 0.9547 | 0.9917 | 1.0000 | **0.0034** | 0.0765 | **0.0083** | 0.9899 | 0.9980 |
| DW-s | 0.9662 | 0.9917 | 0.9710 | 0.9899 | 0.9966 | 0.9976 | 1.0000 | 0.5000 | 0.1479 | 0.9933 | 0.9985 |
| MFDW | 0.9899 | 0.9999 | 0.9966 | 0.9999 | 0.9996 | 0.9324 | 0.5242 | 1.0000 | **0.0003** | 0.9996 | 1.0000 |
| MFDW-s | 0.9980 | 0.9998 | 0.9997 | 1.0000 | 0.9998 | 0.9933 | 0.8662 | 0.9998 | 1.0000 | 0.9999 | 1.0000 |
| tSNE | 0.7085 | 0.1083 | **0.0247** | 0.0676 | 0.1338 | **0.0123** | **0.0083** | **0.0006** | **0.0001** | 1.0000 | 0.9999 |
| tSNE-s | 0.1479 | **0.0101** | **0.0015** | **0.0083** | **0.0034** | **0.0026** | **0.0020** | **0.0001** | **0.0001** | **0.0002** | 1.0000 |

**Tab. 4.8.:** p-values of pairwise paired Wilcoxon rank test (right-sided) on classification accuracies (see Tab. 4.6). Level of significance $\alpha$=.05 and significant p-values are in boldface.

# 5 Experiments: graph embedding evaluation

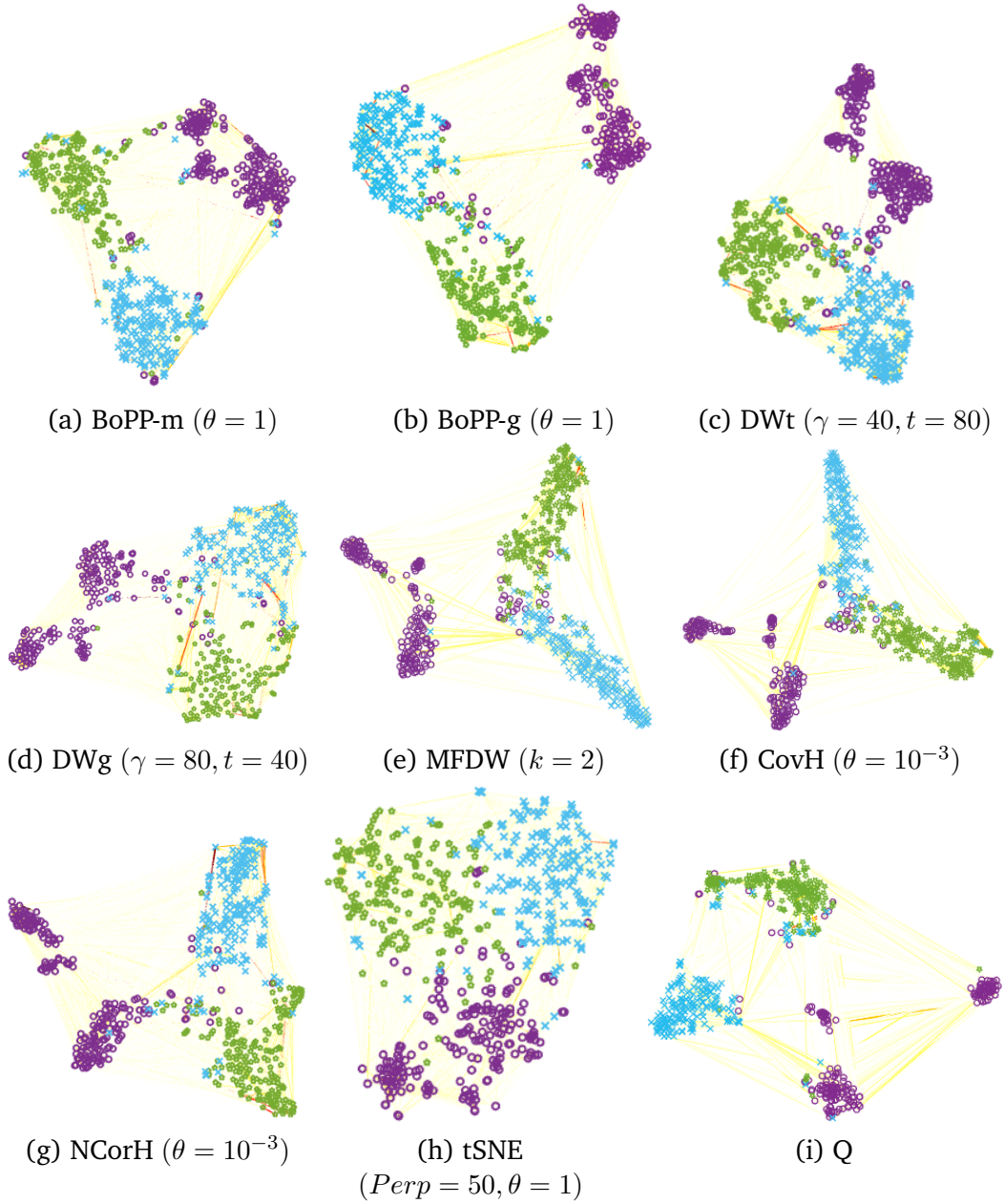
Assessing the quality of the embedding through semi-supervised node classification was the goal of the previous chapter. Indeed, a good graph embedding method should ensure that the learned embeddings can preserve the *original graph structure*. That is why in these experiments, we more focus on graph visualization in two dimensions using tSNE algorithm, graph reconstruction, the quality assessment of nonlinear dimensionality reduction, which focus more on the high-dimensional neighborhood preservations in the low-dimensional space.

As announced in the section 3.1, in this experiment we consider two versions of DW: First, DWg for which the walk length $t$ is set to 40 while the number of walks per node $\gamma$ ranges in $\{5, 15, 30, 40, 80, 120, 160, 300, 500, 1000\}$. Thus, for each particular value of $\gamma$ we generated an embedding file. Second, DWt, which is exactly the opposite of the fist one. In both settings, the window size $w$ remains 10.

## 5.1 Visualization

Visualization techniques play critical roles in the graph embedding. Thus, a good embedding must yield a good visualization in a sense that nodes of same class (color) must remain near from each other. For illustrative purpose, we only consider the News-3cl-1, which contains three classes (class 1=green, class 2=blue, class 3=violet).

From Fig. 5.1, we can see that the structure of the network (presence of clusters) is preserved by all the methods, that is, the similar nodes (nodes of the same class label) are closer to each other than dissimilar ones, even if there are some overlaps. For bag-of-paths methods (BoPP-m, BoPP-g, CovH, NCorH), classes 1 and 2 are projected close to each other without a clear boundary while the class 3 is apart. For tSNE, even if the classes are visible and compact, the segmentation between them failed, as consequence, no boundary can be drawn. Surprisingly, Q well discriminated between datapoints, since each group is far apart from the others and the boundary is drawable even if the class 3 is segmented into three sub-clusters.



(a) BoPP-m ($\theta = 1$) (b) BoPP-g ($\theta = 1$) (c) DWt ($\gamma = 40, t = 80$)

(d) DWg ($\gamma = 80, t = 40$) (e) MFDW ($k = 2$) (f) CovH ($\theta = 10^{-3}$)

(g) NCorH ($\theta = 10^{-3}$) (h) tSNE ($Perp = 50, \theta = 1$) (i) Q

**Fig. 5.1.:** Best visualization of News-3cl-1 using t-SNE algorithm (original dimension of embedding is 5 except for t-SNE for which we directly embedded the free energy distance matrix into 2 dimensions) for each embedding technique. Each point corresponds to a node in the graph and color of a node denotes its class label.

## 5.2 Graph reconstruction

The intuition behind the *graph reconstruction* is that a good graph embedding technique should ensure that the learned embeddings can preserve the *local neighborhoods*. Thus, we use *Precision@k* and *Mean Average Precision (MAP)* [8, 23, 56] to evaluate the performance of methods on this task. *Precision@k* evaluates to what extent neighbors are preserved for each node in the embedding space up to level $k$, referring to the adjacency matrix. The MAP value is the arithmetic mean of individual node average precision values (nodes are weighted equally). Note that MAP is commonly used for information retrieval and object detection tasks. Also, it has been shown that MAP is a metric with good discrimination and stability [39]. The two aforementioned quantities are defined as follows:

$Precision@k$ is the fraction of correct predicted neighbors in top $k$ predicted neighbors. It is defined as follows:

$$Precision@k(i) = \frac{|\{j|i, j \in \mathcal{V}, index(j) \leq k, \Delta_i(j) = 1\}|}{k} \tag{5.1}$$

where $index(j)$ is the ranked index (with respect to the similarity measure) of the $j$-th vertex and $\Delta_i(j) = 1$ indicates that nodes $i$ and $j$ have a link.

MAP estimates the *Average precision (AP)* for every node and computes the average over all nodes. It is defined as follows:

$$AP(i) = \frac{\sum_j Precision@j(i) * \Delta_i(j)}{|\{\Delta_i(j) = 1\}|} \tag{5.2}$$

$$MAP = \frac{\sum_{i \in \mathcal{V}} AP(i)}{|\mathcal{V}|} \tag{5.3}$$

The higher the MAP, the better is the method. Moreover, the node proximity can be computed using different metrics such as Euclidean distance, cosine similarity, ... here we adopted the cosine similarity, defined as follows:

$$\text{similarity} \equiv \cos(\theta) = \frac{\Phi^T(v_i).\Phi(v_j)}{||\Phi(v_i)||\ ||\Phi(v_j)||} \tag{5.4}$$

where $\Phi(v_i)$ and $\Phi(v_j)$ are the column embedding vectors of nodes $i$ and $j$, respectively. As you can see, the cosine similarity tends toward 1 for similar embedding vectors. So, nodes are sorted in descending order of similarity.

The best MAPs per method across all the datasets are shown in Tab. 5.1 as well as their Borda ranking in Tab. 5.2. From these two tables, we can see that the best performing method is BoPP-g, which provided results (see Fig. 5.2) that are significantly superior to the results obtained by Q, CovH, DWg, DWt, MFDW and tSNE. Also, there is no benefit to consider two different versions of DW (DWg and

| Method: Dataset: | BoPP-m | BoPP-g | Q | CovH | NcorH | DWg | DWt | MFDW | tSNE |
|---|---|---|---|---|---|---|---|---|---|
| WebKB-texas | 51.91 | 53.37 | 50.59 | **53.66** | 52.95 | 50.09 | 50.29 | 51.31 | 47.85 |
| WebKB-washington | 31.28 | **32.98** | 25.69 | 31.34 | 31.94 | 32.84 | 32.70 | 29.67 | 27.69 |
| WebKB-wisconsin | 49.40 | 50.51 | 47.57 | 49.72 | **50.56** | 47.86 | 47.87 | 47.43 | 43.66 |
| WebKB-cornell | 41.43 | **42.65** | 39.96 | 41.57 | 42.59 | 42.15 | 42.21 | 40.56 | 36.99 |
| Imdb | 10.78 | 11.22 | 10.49 | 11.01 | 11.02 | **14.55** | 14.55 | 10.27 | 10.85 |
| News-2cl-1 | 53.13 | **55.05** | 52.47 | 52.82 | 54.31 | 52.82 | 52.81 | 52.90 | 50.51 |
| News-2cl-2 | 37.85 | **39.49** | 36.65 | 37.03 | 38.77 | 36.74 | 36.89 | 37.21 | 34.71 |
| News-2cl-3 | 59.51 | **61.44** | 57.51 | 58.05 | 60.64 | 59.03 | 59.00 | 58.35 | 56.04 |
| News-3cl-1 | 50.83 | **52.76** | 49.55 | 49.68 | 51.60 | 51.68 | 51.79 | 49.42 | 48.64 |
| News-3cl-2 | 51.09 | **52.33** | 50.16 | 49.53 | 50.97 | 51.44 | 51.48 | 50.14 | 47.76 |
| News-3cl-3 | 46.69 | **48.77** | 45.66 | 46.80 | 47.96 | 45.50 | 45.44 | 46.28 | 44.39 |
| News-5cl-1 | 47.15 | **47.84** | 45.74 | 45.70 | 47.28 | 46.16 | 46.08 | 45.04 | 43.83 |
| News-5cl-2 | 43.18 | **43.97** | 41.79 | 40.75 | 43.23 | 41.84 | 41.66 | 40.83 | 39.86 |
| News-5cl-3 | 41.50 | **42.84** | 40.98 | 40.35 | 41.79 | 40.72 | 40.69 | 40.29 | 37.97 |

**Tab. 5.1.:** Best MAPs in percent for the various embedding techniques obtained on the different datasets using 5 social dimensions (3 for tSNE). The higher the MAP. the better. The best performing method is highlighted in boldface for each dataset.

| Method | Score | Rank |
|---|---|---|
| BoPP-g | 124 | 1 |
| NCorH | 123 | 2 |
| BoPP-m | 114 | 3 |
| CovH | 109 | 4 |
| DWg | 109 | 4 |
| DWt | 109 | 4 |
| MFDW | 96 | 5 |
| Q | 94 | 6 |
| tSNE | 29 | 7 |

**Tab. 5.2.:** Ranking of the best MAPs (see Tab. 5.1) according to Borda's method. (the higher the score, the better)

DWt) since they both led to the same reconstruction ability. Moreover, the differences in performance among the best performing methods are small. For instance, for the five best techniques (BoPP-g, NCorH, BoPP-m, CovH, DWg; see Tab. 5.2), the average difference between the MAP (see Tab. 5.1) of BoPP-g and the other methods across all datasets is 1.39 and the maximum difference is only 3.39 (versus CovH for News-2cl-3 dataset). However, we noticed that very often the differences between the performances of BoPP-g and BoPP-m are not significantly different. The reasons may be twofold. Firstly, the Gaussian kernel and the MDS kernel are both built from the same free energy distance matrix (derived from the same framework; see section 2.3.2). Secondly, the Nemenyi test is rather conservative, especially when comparing many different techniques. Finally, Wilcoxon test in Tab. 5.3 confirms that BoPP-g is significantly different from all the other methods, followed by NCorH. Also, BoPP-m, CovH, Dwg and DWt are significantly different from Q, MFDW and tSNE, whereas MFDW only beat tSNE.

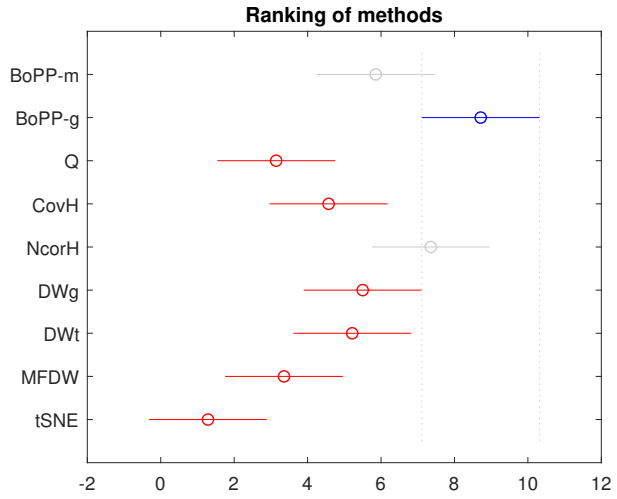



**Fig. 5.2.:** Mean ranks and 95% Nemenyi confidence intervals for the 9 methods (see Tab. 5.1). Two methods are considered significantly different if their confidence intervals do not overlap. The worse methods (Q, CovH, DWg, DWt, MFDW, tSNE) and the best method (BoPP-g) overall are highlighted. (p-value Friedman test=3.7074e-13)

| | BoPP-m | BoPP-g | Q | CovH | NCorH | DWg | DWt | MFDW | tSNE |
|---|---|---|---|---|---|---|---|---|---|
| BoPP-m | 1.0000 | 1.0000 | **0.0001** | 0.0765 | 0.9999 | 0.1955 | 0.1788 | **0.0001** | **0.0001** |
| BoPP-g | **0.0001** | 1.0000 | **0.0001** | **0.0002** | **0.0001** | **0.0067** | **0.0054** | **0.0001** | **0.0001** |
| Q | 1.0000 | 1.0000 | 1.0000 | 0.9608 | 1.0000 | 0.9933 | 0.9957 | 0.7492 | **0.0026** |
| CovH | 0.9324 | 0.9999 | **0.0453** | 1.0000 | 0.9998 | 0.8794 | 0.8662 | **0.0338** | **0.0001** |
| NCorH | **0.0002** | 0.9999 | **0.0001** | **0.0003** | 1.0000 | **0.0290** | **0.0290** | **0.0001** | **0.0001** |
| DWg | 0.8212 | 0.9946 | **0.0083** | 0.1338 | 0.9753 | 1.0000 | 0.4097 | **0.0209** | **0.0001** |
| DWt | 0.8371 | 0.9947 | **0.0054** | 0.1479 | 0.9753 | 0.6020 | 1.0000 | **0.0148** | **0.0001** |
| MFDW | 1.0000 | 1.0000 | 0.2708 | 0.9710 | 1.0000 | 0.9824 | 0.9877 | 1.0000 | **0.0001** |
| tSNE | 0.9999 | 1.0000 | 0.9980 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0.9999 | 1.0000 |

**Tab. 5.3.:** p-values of pairwise paired Wilcoxon rank test (right-sided) for best MAPs (see Tab. 5.1). Level of significance $\alpha$=.05 and significant p-values are in boldface.

As we said in the previous paragraph, the variability of the MAP value across each dataset in Tab. 5.1 is really too weak and this may seem counterintuitive. This could be justified by the fact that the MAP value is high when the individual average precisions are high for a large number of nodes in the graph. In many real-word datasets, it could happen that neighbors are very well preserved only for some nodes (increasing in AP's) but are not for the majority of nodes (drop of the MAP value because nodes are equally weighted), since we are averaging over AP's, we could end up with the same overall result. Moreover, to make sure that there was not any bug in the source code, we printed all the AP's during the process and we saw that they were really different and correctly computed.

Finally, by carefully observing Tab. 5.2, tSNE is ranked last with the lowest score of 29. This accommodate with the results of the Wilcoxon test (see Tab. 5.3) because it is really outperformed by all the other methods. That is why we pointed out the

fact that the Borda ranking reflects the reality as soon as the Wilcoxon test gives a significant p-value when comparing the relative performances of the methods.

As done for the classification tasks, we repeated the experiment using only two social dimensions for all the methods. The first ranked method is BoPP-m but the overall differences in performance are also small. If we consider only the five-best techniques (BoPP-m, BoPP-g, CovH, NCorH, Q; see Tab. 5.5), the average difference between the MAP (see Tab. 5.4) of BoPP-m and the other methods across all datasets is 0.63 and the maximum difference is only 3.92 (versus NCorH for WebKB-wisconsin dataset). However, we can see that the Nemenyi test (see Fig. 5.3) confirms that the BoPP-m provided good results, which are significantly different from CovH, DW, MFDW and tSNE; whereas BoPP-g, Q and NCorH provided competitive results. The Wilcoxon test reported in Tab 5.6 confirms that BoPP-m is better than all the other methods, except BoPP-g. Besides, BoPP-g is significantly different from CovH, NCorH, DW, MFDW and tSNE; while Q and NCorH are significantly different from CovH, DW, MFDW and tSNE.

| Method:<br>Dataset: | BoPPm | BoPPg | Q | CovH | NCorH | DW | MFDW | tSNE |
|---|---|---|---|---|---|---|---|---|
| WebKB-texas | **48.97** | 47.23 | 47.86 | 48.04 | 47.33 | 42.54 | 44.49 | 46.09 |
| WebKB-washington | **29.59** | 28.77 | 25.23 | 28.46 | 29.11 | 28.08 | 27.16 | 24.93 |
| WebKB-wisconsin | **45.86** | 43.04 | 44.89 | 42.21 | 41.94 | 38.97 | 38.21 | 42.00 |
| WebKB-cornell | 37.95 | 37.49 | 38.94 | 36.85 | **38.95** | 37.22 | 35.49 | 35.75 |
| Imdb | 9.03 | 9.03 | 9.08 | 9.15 | 9.35 | 8.91 | 8.44 | **9.75** |
| News-2cl-1 | 50.47 | **50.93** | 50.13 | 49.92 | 50.44 | 49.53 | 50.13 | 49.11 |
| News-2cl-2 | 33.96 | **35.32** | 33.56 | 33.17 | 34.61 | 33.47 | 34.08 | 33.45 |
| News-2cl-3 | 56.13 | **56.55** | 55.06 | 54.97 | 55.45 | 54.99 | 55.93 | 54.62 |
| News-3cl-1 | **48.03** | 46.44 | 46.00 | 45.02 | 46.31 | 46.64 | 44.87 | 47.34 |
| News-3cl-2 | **48.07** | 47.88 | 47.81 | 46.67 | 47.54 | 47.53 | 46.82 | 46.91 |
| News-3cl-3 | 43.74 | **44.53** | 43.69 | 42.59 | 44.00 | 42.08 | 40.83 | 43.46 |
| News-5cl-1 | **44.13** | 43.73 | 43.44 | 42.49 | 42.73 | 42.02 | 42.55 | 43.02 |
| News-5cl-2 | 40.03 | **40.10** | 38.99 | 38.43 | 39.70 | 37.82 | 38.04 | 38.79 |
| News-5cl-3 | 38.12 | **38.79** | 37.64 | 36.60 | 37.84 | 36.77 | 35.45 | 37.48 |

**Tab. 5.4.:** Best MAPs in percent for the various embedding techniques obtained on the different datasets using 2 social dimensions. The higher the MAP. the better. The best performing method is highlighted in boldface for each dataset.

| Method | Score | Rank |
|---|---|---|
| BoPP-m | 112 | 1 |
| BoPP-g | 110 | 2 |
| CovH | 110 | 2 |
| NCorH | 110 | 2 |
| Q | 106 | 3 |
| DW | 99 | 4 |
| tSNE | 93 | 5 |
| MFDW | 80 | 6 |

**Tab. 5.5.:** Ranking of the best MAPs for 2D embedding (see Tab. 5.4) according to Borda's method. (the higher the score, the better.)

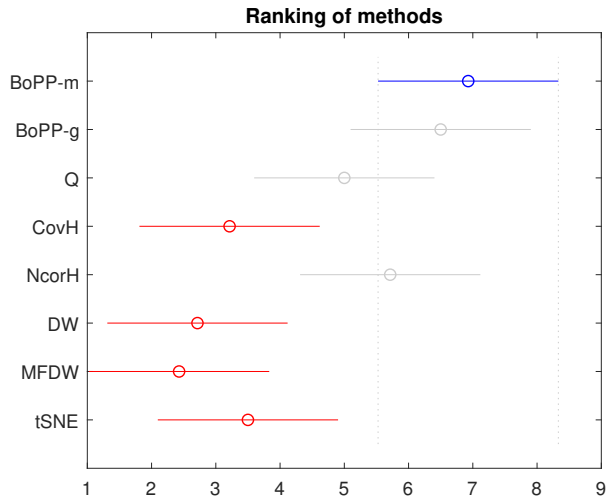



**Fig. 5.3.:** Mean ranks and 95% Nemenyi confidence intervals for the 8 methods (see Tab. 5.4). Two methods are considered significantly different if their confidence intervals do not overlap. The worse methods (CovH, DW, MFDW, tSNE) and the best method (BoPP-m) overall are highlighted. (p-value Friedman test=1.0234e-08)

| | BoPP-m | BoPP-g | Q | CovH | NCorH | DW | MFDW | tSNE |
|---|---|---|---|---|---|---|---|---|
| BoPP-m | 1.0000 | 0.2508 | **0.0034** | **0.0001** | **0.0392** | **0.0001** | **0.0001** | **0.0006** |
| BoPP-g | 0.7684 | 1.0000 | 0.0765 | **0.0006** | **0.0290** | **0.0002** | **0.0001** | **0.0006** |
| Q | 0.9974 | 0.9324 | 1.0000 | **0.0148** | 0.8521 | **0.0148** | **0.0083** | **0.0176** |
| CovH | 0.9999 | 0.9996 | 0.9877 | 1.0000 | 0.9980 | 0.1479 | **0.0338** | 0.5724 |
| NCorH | 0.9662 | 0.9753 | 0.1629 | **0.0026** | 1.0000 | **0.0002** | **0.0003** | **0.0101** |
| DW | 1.0000 | 0.9999 | 0.9877 | 0.8662 | 0.9999 | 1.0000 | 0.1629 | 0.8794 |
| MFDW | 0.9999 | 1.0000 | 0.9933 | 0.9710 | 0.9998 | 0.8521 | 1.0000 | 0.9324 |
| tSNE | 0.9996 | 0.9996 | 0.9852 | 0.4516 | 0.9917 | 0.1338 | 0.0765 | 1.0000 |

**Tab. 5.6.:** p-values of pairwise paired Wilcoxon rank test (right-sided) for best MAPs (see Tab. 5.4). Level of significance $\alpha$=.05 and significant p-values are in boldface.

An important observation before closing this section, regardless of the dimensionality, we noticed that MAP get improved when $\theta$ for bag-of-paths methods (BoPP-m, BoPP-g, CovH, NCorH, tSNE) increases up to a certain value (depending on the dataset), before decaying. For MFDW, this behavior has been also observed when the order $k$ increases. Specially for DW, fixing either $\gamma$ or $t$ to a reasonable value ($\geq 40$) and varying the other parameter value was not beneficial since the performances remain almost unchanged. This means that DW is capable of learning meaningful latent representations of vertices after only a small number of random walks or using a short walk length. However, it would have been better to evaluate the behavior of MAP using different distance metrics such as Euclidean, Manahattan or Minkowski. Due to the time constraint, we dedicate this for future research (see chapter 6). Note that all the detailed results about MAP are available in Appendix.

## 5.3 Quality assessment of nonlinear dimensionality reduction

Dimensionality reduction (DR) aims at providing low-dimensional representations of high-dimensional datasets. As an alternative way of assessing the quality of the different embeddings, we are interested in the works of John A. Lee and Michel Verleysen [31, 32, 33, 34] who have proposed a suitable approach which are based on distance ranking and $k$-ary neighborhoods. This method aims to quantify the preservation of the high-dimensional neighborhoods in the low-dimensional space.

Let's denote by $\delta_{ij}$, the distance from datapoints $\boldsymbol{\xi}_i$ to $\boldsymbol{\xi}_j$ in the high-dimensional space (HDS). Similarly, the distance from $\boldsymbol{x}_i$ to $\boldsymbol{x}_j$ in the low-dimensional space (LDS) by $d_{ij}$. Note that we assume $\delta_{ij} = \delta_{ji}$ and $d_{ij} = d_{ji}$ and no assumption is made as to the metrics that are associated with the high- and low-dimensional spaces, which can be different. Starting from distances, we can compute ranks. The rank of $\boldsymbol{\xi}_j$ with respect to $\boldsymbol{\xi}_i$ in HDS is written as $\rho_{ij} = |\{k : \delta_{ik} < \delta_{ij}$ or $(\delta_{ik} = \delta_{ij}$ and $1 \leq k < j \leq n)\}|$, where $|\ .\ |$ denotes the set cardinality. Similarly, the rank $\boldsymbol{x}_j$ with respect to $\boldsymbol{x}_i$ in the LDS is $r_{ij} = |\{k : d_{ik} < d_{ij}$ or $(d_{ik} = d_{ij}$ and $1 \leq k < j \leq n)\}|$. Hence, reflexive ranks are set to zero ($\rho_{ii} = r_{ii} = 0$) and ranks are unique, i.e. there are no ex aequo ranks: $\rho_{ij} \neq \rho_{ik}$ for $k \neq j$, even if $\delta_{ij} = \delta_{ik}$ . This means that nonreflexive ranks belong to $\{1, \ldots, n-1\}$. The nonreflexive $K$-ary neighborhoods of $\boldsymbol{\xi}_i$ and $\boldsymbol{x}_i$ are denoted by $v_i^K = \{j : 1 \leq \rho_{ij} \leq K\}$ and $n_i^K = \{j : 1 \leq r_{ij} \leq K\}$, respectively. Thus, we can quantify their average agreement by

$$Q_{NX}(K) = \frac{1}{n} * \sum_{i=1}^{n} \frac{|v_i^K \cap n_i^K|}{K} \tag{5.5}$$

$Q_{NX}(K)$ is called the *Coranking score*, aiming to evaluate the overall quality of the embedding. It varies between 0 and 1, and measures the preservation of $K$-ary neighborhoods in a straightforward way. However, it is better to compare the overall quality through the $R_{NX}(K)$, which is the *relative improvement* with respect to a random embedding. Mathematically

$$R_{NX}(K) = \frac{(n-1) * Q_{NX}(K) - K}{n - 1 - K} \tag{5.6}$$

Local (or small) neighborhoods are more important than global ones, this is why a logarithmic scale for $K$ is often used in order to draw the $R_{NX}(K)$ curve. The left part of the $R_{NX}(K)$ curve is likely to be more important than the right part. Thus, one method outperforms the other if it has the highest value of the $R_{NX}(K)$ for a wide range among the smallest values of $K$. Of course, the same method will perform even better if it keeps the curve as high as possible for all values of $K$.

Note that the value of $R_{NX}(K)$ ranges between -1 and 1, but $R_{NX}(K) < 0$ refers to an embedding worse than random, so the useful range lies between 0 and 1. Moreover, for low-dimensional coordinates sampled at random, the expectation of $R_{NX}(K) = 0$ while the expectation of $Q_{NX}(K) = K/(n-1)$, which increases with $K$. A remaining important indicator is the *Area under the curve* $R_{NX}(K)$, denoted by AUC, quantifying the overall quality with more emphasis on small neighborhoods because a logarithmic scale for $K$ is used to plot the $R_{NX}(K)$ curve. The AUC is defined as follows:

$$AUC = \frac{\sum_{K=1}^{n-2} \frac{R_{NX}(K)}{K}}{\sum_{K=1}^{n-2} \frac{1}{K}} \tag{5.7}$$

In practice, before computing these quantities: Firstly, we computed the distances for each pair of nodes from the cost matrix **C** using the Dijkstra algorithm[1] in the high-dimensional space. Secondly, we computed the Euclidean distances for each pair of nodes from the embeddings in the low-dimensional space, and then we compared the ranks. The results are reported in Tab. 5.7 as well as their Borda

| Method: Dataset: | BoPPm | BoPPg | Q | CovH | NCorH | DWg | DWt | MFDW | tSNE |
|---|---|---|---|---|---|---|---|---|---|
| WebKB-texas | 15.59 | **21.68** | 12.20 | 15.50 | 15.41 | 15.84 | 13.96 | 13.21 | 19.10 |
| WebKB-washington | 11.46 | **19.16** | 4.35 | 9.67 | 14.68 | 15.02 | 13.26 | 7.13 | 17.59 |
| WebKB-wisconsin | 24.40 | 24.30 | 12.75 | 14.72 | 19.36 | 14.56 | 13.21 | 10.84 | **29.84** |
| WebKB-cornell | 10.93 | 16.24 | 8.36 | 12.21 | 14.71 | 12.16 | 10.78 | 10.76 | **16.30** |
| Imdb | 30.16 | 32.52 | 21.22 | 29.61 | 30.89 | 22.91 | 21.64 | 24.74 | **33.05** |
| News-2cl-1 | **40.60** | 35.12 | 20.65 | 20.63 | 26.92 | 23.71 | 23.08 | 17.05 | 40.10 |
| News-2cl-2 | **38.50** | 34.95 | 21.68 | 20.93 | 26.67 | 22.63 | 22.27 | 17.47 | 38.19 |
| News-2cl-3 | **42.56** | 39.51 | 26.50 | 25.34 | 29.21 | 25.86 | 25.29 | 22.16 | 41.29 |
| News-3cl-1 | 38.37 | 34.34 | 22.94 | 22.56 | 25.29 | 22.99 | 23.18 | 19.55 | **39.61** |
| News-3cl-2 | 34.26 | 30.52 | 20.87 | 20.07 | 25.97 | 20.52 | 20.19 | 16.37 | **37.95** |
| News-3cl-3 | 37.41 | 33.72 | 21.23 | 20.90 | 23.14 | 21.31 | 21.09 | 16.74 | **39.60** |
| News-5cl-1 | 30.69 | 27.51 | 19.78 | 18.71 | 21.26 | 19.11 | 19.09 | 14.47 | **37.74** |
| News-5cl-2 | 29.59 | 26.76 | 19.83 | 17.39 | 22.77 | 18.35 | 18.29 | 14.10 | **36.82** |
| News-5cl-3 | 28.17 | 25.01 | 15.91 | 16.17 | 19.62 | 15.73 | 15.33 | 12.07 | **33.67** |

**Tab. 5.7.:** Best AUCs in percent for the various embedding techniques obtained on the different datasets using 5 social dimensions (3 for tSNE). The higher the AUC, the better. The best performing method is highlighted in boldface for each dataset.

ranking in Tab. 5.8. We can see that tSNE is the best performing method but BoPP-g and BoPP-m provided also competitive results. Theoretically, this result about tSNE could be justified by its strength ability for the neighborhood preservation because, as we saw in the section 3.3, its uses Student t-distribution, which has much heavier tails than a Gaussian to convert distances (in the low-dimensional space) into probabilities, making sure that the unwanted attractive forces between low-dimensional datapoints that represent moderately dissimilar high-dimensional datapoints are eliminated. Practically for us, this behavior was unexpected referring to the ranking position of tSNE for the classification tasks (see Tab. 4.3 and 4.4).

[1] Because we assumed that neighbors are given by shortest path distances.

Besides, the differences in performance among the best performing methods are large. For instance, for the five best techniques (tSNE, BoPP-g, BoPP-m, NCorH, DWg; see Tab. 5.8), the average difference between the AUC (see Tab. 5.7) of tSNE and the other methods across all datasets is 7.9 and the maximum difference is 18.63 (versus DWg for News-5cl-1 dataset).

| Method | Score | Rank |
|---|---|---|
| tSNE | 123 | 1 |
| BoPP-g | 107 | 2 |
| BoPP-m | 105 | 3 |
| NCorH | 88 | 4 |
| DWg | 72 | 5 |
| CovH | 70 | 6 |
| DWt | 61 | 7 |
| Q | 51 | 8 |
| MFDW | 23 | 9 |

**Tab. 5.8.:** Ranking of the best AUCs, (see Tab. 5.7) according to Borda's method. (the higher the score, the better)

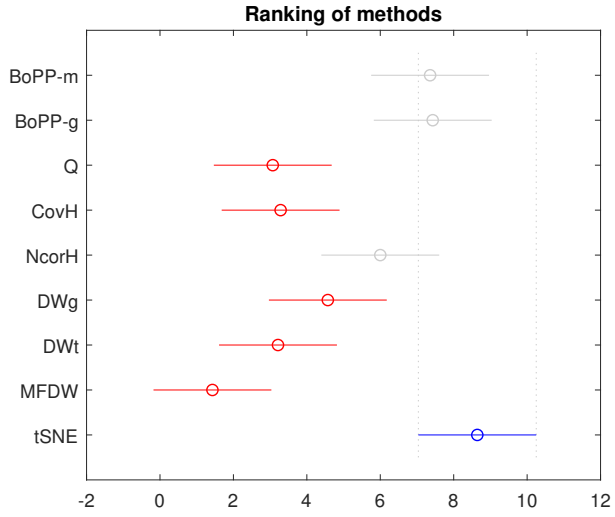



**Fig. 5.4.:** Mean ranks and 95% Nemenyi confidence intervals for the 9 methods (see Tab. 5.7). Two methods are considered significantly different if their confidence intervals do not overlap. The worse methods (Q, CovH, DWg, DWt, MFDW) and the best method (tSNE) overall are highlighted. (p-value Friedman test=3.5920e-16)

Moreover, having a significant p-value of 3.5920e-16 for the Friedman test, we conducted the Nemenyi test (see Fig. 5.4), confirming that tSNE provided good results, which are significantly superior to the results obtained by Q, CovH, DWg, DWt and MFDW; whereas BoPP-m, BoPP-g and NCorH still remain competitive. The Wilcoxon test on its side (see Tab. 5.9) confirms that tSNE is better than all the other methods, whereas BoPP-m and BoPP-g are significantly different from Q, CovH, NCorH, DWt and MFDW. Still on this table, NCorH is significantly different from Q, CovH, DWg, DWt and MFDW; while DWg is significantly different from CovH, DWt and MFDW. The modularity matrix Q, CovH and DWt are only better than MFDW.

| | BoPP-m | BoPP-g | Q | CovH | NCorH | DWg | DWt | MFDW | tSNE |
|---|---|---|---|---|---|---|---|---|---|
| BoPP-m | 1.0000 | 0.2316 | **0.0001** | **0.0003** | **0.0020** | 0.0009 | **0.0003** | **0.0001** | 0.9991 |
| BoPP-g | 0.7869 | 1.0000 | **0.0001** | **0.0001** | **0.0001** | 0.0001 | **0.0001** | **0.0001** | 0.9988 |
| Q | 1.0000 | 1.0000 | 1.0000 | 0.7085 | 1.0000 | 0.9547 | 0.7869 | **0.0067** | 1.0000 |
| CovH | 0.9998 | 1.0000 | 0.3129 | 1.0000 | 0.9999 | 0.9662 | 0.5484 | **0.0001** | 1.0000 |
| NCorH | 0.9985 | 1.0000 | **0.0001** | **0.0001** | 1.0000 | **0.0003** | **0.0001** | **0.0001** | 1.0000 |
| DWg | 0.9994 | 1.0000 | 0.0520 | **0.0392** | 0.9998 | 1.0000 | **0.0003** | **0.0002** | 1.0000 |
| DWt | 0.9998 | 1.0000 | 0.2316 | 0.4758 | 1.0000 | 0.9998 | 1.0000 | **0.0004** | 1.0000 |
| MFDW | 1.0000 | 1.0000 | 0.9946 | 1.0000 | 1.0000 | 0.9999 | 0.9997 | 1.0000 | 1.0000 |
| tSNE | **0.0012** | **0.0015** | **0.0001** | **0.0001** | **0.0001** | **0.0001** | **0.0001** | **0.0001** | 1.0000 |

**Tab. 5.9.:** p-values of pairwise paired Wilcoxon rank test (right-sided) for best AUCs (see Tab. 5.7). Level of significance $\alpha$=.05 and significant p-values are in boldface.

Additionally, the Fig. 5.6 depicts $R_{NX}(K)$ curves per method across the News-3cl-1, News-5cl-1 and WebKB-texas datasets. Remember that a method is best performing if its curve is high for a large range of smaller values of $K$. From this figure, we clearly see that tSNE outperformed the other methods since it more preserved small neighborhoods rather than large ones as done for the other methods. Obviously, this performance decays rapidly for all the methods for very large values of $K$.

Finally, we extrapolated the experiment by considering two social dimensions for all the methods, and once more, tSNE performed best. The results are reported in Tab. 5.10, 5.11 and 5.12 as well as in Fig. 5.5. Specially for the relative performances of the methods, Wilcoxon confirms that tSNE outperformed all the other methods, followed by BoPP-m. Moreover, BoPP-g and NCorH are significantly different from CovH, DW and MFDW; whereas Q and CovH are only significantly different from DW and MFDW. See more detailed results in Appendix.

| Method:<br>Dataset: | BoPPm | BoPPg | Q | CovH | NCorH | DW | MFDW | tSNE |
|---|---|---|---|---|---|---|---|---|
| WebKB-texas | 7.92 | 8.85 | 7.31 | 8.84 | 10.46 | 8.11 | 8.36 | **13.98** |
| WebKB-washington | 8.60 | 10.28 | 3.73 | 5.88 | 8.83 | 3.92 | 5.46 | **11.43** |
| WebKB-wisconsin | 11.66 | 10.06 | 7.66 | 6.45 | 8.84 | 4.36 | 10.71 | **21.36** |
| WebKB-cornell | 7.38 | 8.04 | 4.99 | 6.35 | 9.54 | 5.37 | 6.54 | **10.94** |
| Imdb | 18.65 | 18.07 | 13.87 | 18.36 | 18.68 | 9.54 | 17.15 | **24.22** |
| News-2cl-1 | 17.53 | 12.38 | 12.98 | 10.84 | 12.18 | 9.62 | 9.69 | **26.83** |
| News-2cl-2 | 15.49 | 11.72 | 11.38 | 10.19 | 9.86 | 8.44 | 9.12 | **24.37** |
| News-2cl-3 | 20.75 | 14.80 | 18.13 | 16.06 | 18.04 | 11.71 | 14.62 | **30.75** |
| News-3cl-1 | 16.98 | 13.45 | 14.95 | 12.64 | 14.07 | 12.41 | 9.91 | **26.67** |
| News-3cl-2 | 13.76 | 10.32 | 10.29 | 8.78 | 11.96 | 7.79 | 7.76 | **24.46** |
| News-3cl-3 | 15.59 | 11.69 | 12.58 | 9.63 | 10.97 | 8.79 | 5.30 | **25.60** |
| News-5cl-1 | 14.73 | 9.79 | 10.89 | 8.85 | 9.60 | 7.21 | 7.59 | **23.38** |
| News-5cl-2 | 14.08 | 10.49 | 12.05 | 10.59 | 11.80 | 8.38 | 8.88 | **22.85** |
| News-5cl-3 | 11.09 | 8.93 | 8.62 | 5.42 | 6.56 | 5.14 | 4.73 | **18.85** |

**Tab. 5.10.:** Best AUCs in percent for the various embedding techniques obtained on the different datasets using 2 social dimensions. The higher the AUC, the better. The best performing method is highlighted in boldface for each dataset.

| Method | Score | Rank |
|---|---|---|
| tSNE | 112 | 1 |
| BoPP-m | 95 | 2 |
| BoPP-g | 80 | 3 |
| NCorH | 80 | 3 |
| Q | 72 | 4 |
| CovH | 60 | 5 |
| MFDW | 51 | 6 |
| DW | 42 | 7 |

**Tab. 5.11.:** Ranking of the best AUCs (see Tab. 5.10) for 2D embedding according to Borda's method. (the higher the score, the better)

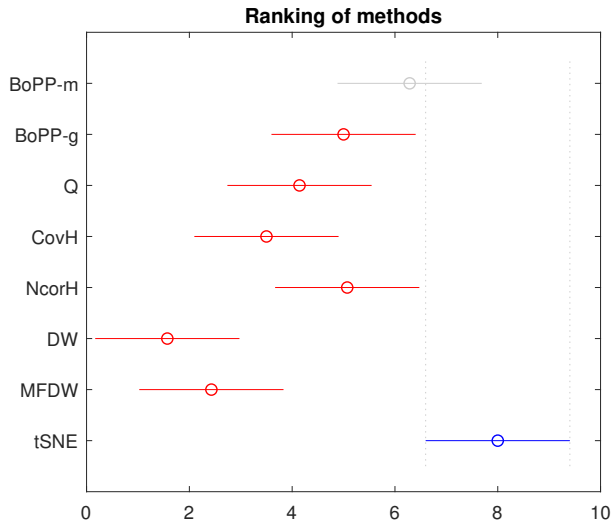



**Fig. 5.5.:** Mean ranks and 95% Nemenyi confidence intervals for the 8 methods (see Tab. 5.10). Two methods are considered significantly different if their confidence intervals do not overlap. The worse methods (BoPP-g, Q, CovH, NCorH, DW, MFDW) and the best method (tSNE) overall are highlighted. (p-value Friedman test=1.4612e-12)

Four global observations to emphasize before closing this chapter, regardless of the dimensionality :

- A best performing method at graph reconstruction is not necessary best at neighborhood preservation. For instance, BoPP-g and BoPP-m performed best for the reconstruction task using 5D and 2D respectively, but they were overthrown by tSNE for the neighborhood preservation task in both dimensionality settings.
- During all the experiments, we noticed that Nemenyi simply gives overall trends on the results (best vs worse methods) and the equivalent methods in terms of performance usually appear among the top-five best methods in the Borda ranking (e.g. BoPP-m, BoPP-g, CovH, NCorH).

| | BoPP-m | BoPP-g | Q | CovH | NCorH | DW | MFDW | tSNE |
|---|---|---|---|---|---|---|---|---|
| BoPP-m | 1.0000 | **0.0026** | **0.0001** | **0.0002** | **0.0054** | **0.0001** | **0.0001** | 1.0000 |
| BoPP-g | 0.9980 | 1.0000 | 0.2915 | **0.0034** | 0.6426 | **0.0001** | **0.0003** | 1.0000 |
| Q | 1.0000 | 0.7292 | 1.0000 | 0.1206 | 0.7684 | **0.0009** | **0.0209** | 1.0000 |
| CovH | 0.9999 | 0.9974 | 0.8917 | 1.0000 | 0.9999 | **0.0001** | **0.0067** | 1.0000 |
| NCorH | 0.9957 | 0.3804 | 0.2508 | **0.0002** | 1.0000 | **0.0001** | **0.0004** | 1.0000 |
| DW | 0.9999 | 1.0000 | 0.9994 | 1.0000 | 1.0000 | 1.0000 | 0.9406 | 1.0000 |
| MFDW | 0.9999 | 0.9998 | 0.9824 | 0.9946 | 0.9997 | 0.0676 | 1.0000 | 1.0000 |
| tSNE | **0.0001** | **0.0001** | **0.0001** | **0.0001** | **0.0001** | **0.0001** | **0.0001** | 1.0000 |

**Tab. 5.12.:** p-values of pairwise paired Wilcoxon rank test (right-sided) for best AUCs (see Tab. 5.10). Level of significance $\alpha$=.05 and significant p-values are in boldface.

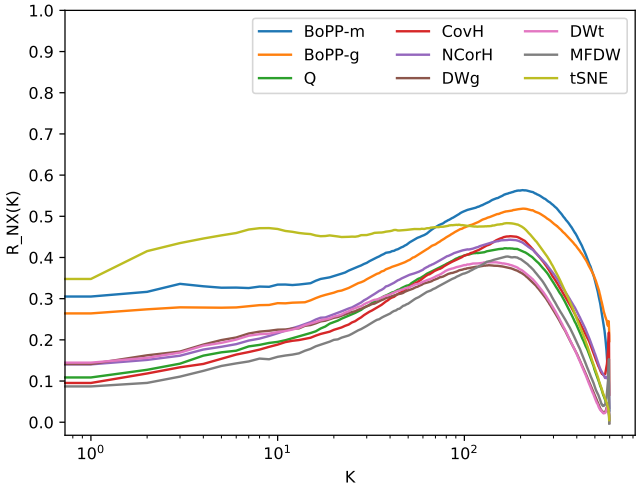
- MAP and AUC are rigorous metrics since the best performing method is not really dataset-dependent, we saw it by taking a look at the results (see Tab. 5.1, 5.4, 5.7 and 5.10), contrariwise to the classification tasks where the best method was dataset-dependent.
- In most of the cases, BoPP-m, BoPP-g and tSNE yield higher performance for MAP and AUC typically when $\theta \geq 1$, whereas CovH and NCorH get improved when $\theta$ increases up to 0.1 and beyond this bound their performances drop.
- For MFDW, the optimal number of steps (order) $k$ is 2 (but sometimes 3 depending on the dataset). More precisely, we noticed that when $k > 5$, the performances in term of MAP and AUC deteriorates rapidly.
- DW is more stable than the other methods because it is less sensitive to the parameter variations.

## 5.4 Parameter Sensitivity

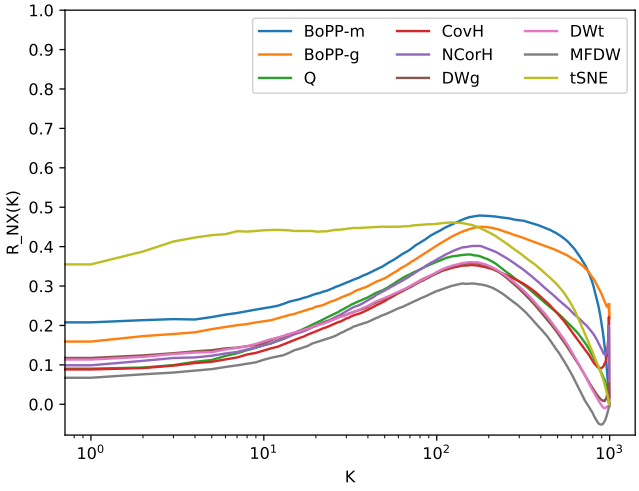
In this section, we are evaluating how changes to the parameterisation of each embedding technique affect its performance in term of semi-supervised node classification task. The Fig. 5.7 depicts the results about the News-3cl-1 and WebKB-texas datasets. We have the following observations and analysis:

- The behavior of bag-of-paths methods (BoPP-m, BoPP-g, CovH and NCorH) is really dataset-dependant. See Fig. 5.7(a, b, f and g).
- The relative performance of DW (see Fig. 5.7c and 5.7d) is relatively stable across different values of whether $\gamma$ or $t$ for both datasets. This is partly because we used reasonable $\gamma$ value of 40 while varying the walk length $t$, and vice versa. So, DW learned meaningful latent representations of vertices after only a small number of random walks or short walk length.

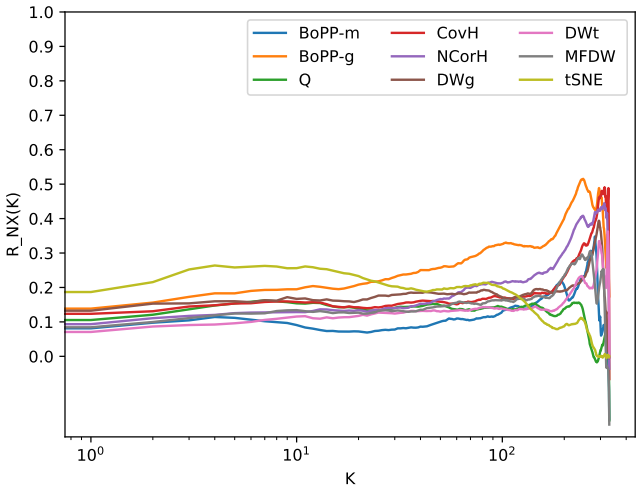
- When $k > 5$, the accuracy of MFDW deteriorates periodically then becomes invariant (from $k = 20$) for both datasets. In fact, $k = 2$ is ideally good to obtain better performances.
- When $\theta > 0.1$, the performance of CovH and NCorH decayed for both datasets. This is due to the fact that these two methods rely on the bag-of-paths framework, and when $\theta$ increases, the most likely paths are the shortest ones. As consequence, similar nodes may become dissimilar.




(a) News-3cl-1




(b) News-5cl-1




(c) WebKB-texas

**Fig. 5.6.:** Best relative improvement per method using 5 social dimensions (3 for tSNE). $K$ onto x-axis (using log scale) is the neighborhood size. The highest value of $R_{NX}(K)$ for a wide range among the smallest values of $K$, the better.

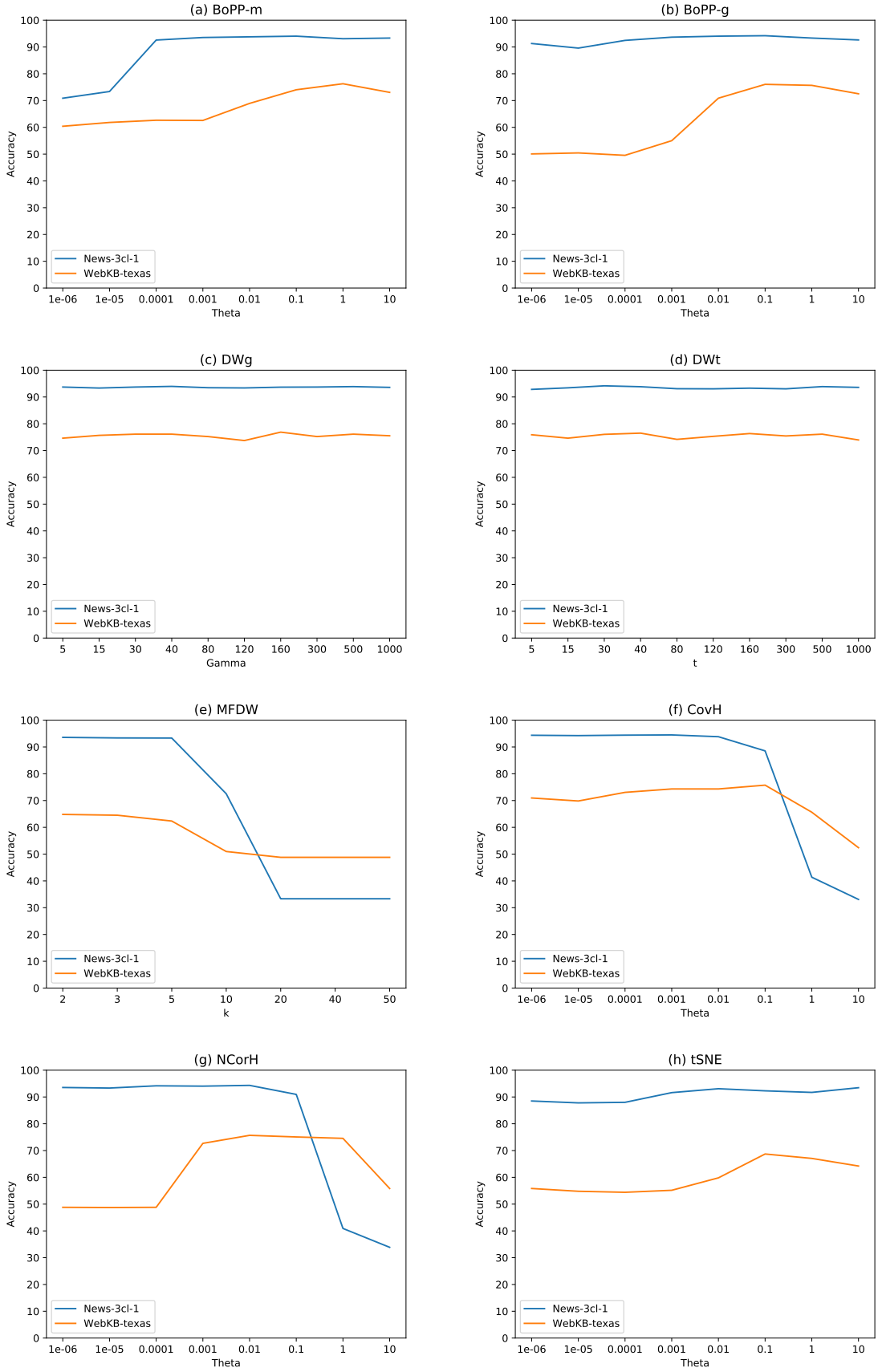



**Fig. 5.7.:** Parameter Sensitivity on News-3cl-1 and WebKB-texas datasets. The hyper-parameter of the model onto x-axis and the accuracy for the semi-supervised node classification onto y-axis.

# 6 Conclusion

Graph data analysis is nowadays a popular topic in data science since graphs become an unavoidable mean to represent large amount of data, used in many real-world applications such as in social networks analysis and many other ones. Graph mining and machine learning make it possible to deal effectively with the various problems that arise there, such as node classification, node recommendation, link prediction, and so on. The graph embedding is an effective yet efficient way to solve the aforementioned problems and the choice of a good embedding method is a determining factor of performance.

The goal of this work was to evaluate the performance of various embedding methods through semi-supervised node classification, graph visualization, graph reconstruction and neighborhood preservation. Firstly, we presented different concepts used throughout this work, with a particular focus on network data analysis, machine learning, baseline embedding techniques, as well as on statistical concepts. We conducted our experiments using five social dimensions[1] (5D) and then two (2D), in order to detect the behavior of the methods when facing up to different dimensionalities.

Secondly, using 5D, the experiments on semi-supervised node classification revealed that DeepWalk (DW), the state-of-the-art, obtained the best results but the difference was not statistically significant in comparison with the bag-of-paths methods (Multi-dimensional scaling of the free energy distance matrix (BoPP-m), Gaussian kernel of the free energy distance matrix (BoPP-g), covariance of nodes co-presence on hitting paths (CovH) and correlation of nodes co-occurrence on hitting path (NCorH)). Afterwards in 2D, t-Distributed Stochastic Neighbor Embedding (tSNE-s) better did the job by outperforming all the other methods with significant differences. However, we noticed that the 'best' method is **dataset-dependent** and **task-dependent**, this is why it is difficult to know in advance which one will perform best.

Thirdly, the graph visualization from these embedding revealed that all these methods have a good capacity for discriminating nodes between the classes, even if the boundaries sometimes are not so clear. For the graph reconstruction task, we saw

[1] Only three social dimensions for tSNE

that all these techniques have, roughly speaking, the same reconstruction ability but low (Mean average precision of 43% on average).

Finally, we assessed the quality of nonlinear dimensionality reduction based on distance ranking and $k$-ary neighborhoods. Here, tSNE outperformed all the other methods with significant differences in both dimensionality settings. Nevertheless, BoPP-m, BoPP-g and NCorH still remain competitive. Also, we observed that all these techniques, including tSNE, have shown low ability for the neighborhood preservation task (Area under the curve of 23% on average).

Concerning the sensitivity to the parameters of the different models, in most of the cases, regardless of the dataset and the dimensionality whether for the classification, graph reconstruction or neighborhood preservation, we observed that the performance is higher for BoPP-m, BoPP-g and tSNE when $\theta \geq 1$, for CovH and NCorH when $\theta$ varies up to 0.1, and for the matrix factorization of DeepWalk (MFDW) when $k \leq 5$ (for computational efficiency $k = 2$ is sufficient). By setting up whether $\gamma$ or $t$ to a reasonable value ($\geq 40$) and varying the other hyper-parameter (even for small values), the performances of DW remain unchanged because it is capable of learning meaningful latent representation of vertices after only a small number of random walks or using a small walk length.

During all the experiments, we noticed that the Nemenyi test simply gives overall trends on the results (best vs worse methods) whereas the Borda ranking gives another facet of the information. Thus, if one method is less performing than another in this ranking, only a significant p-value of the Wilcoxon test can reassure us. Otherwise this ranking could be misleading.

Ultimately, for the semi-supervised node classification task, it is difficult to say what method to choose in first position among DW, NCorH and BoPP-g because they all yielded best performances, but DW is to be avoided when the number of dimensions is too small (e.g., $d = 2$). For the graph reconstruction, the free energy based methods (BoPP-m and BoPP-g) are more preferable, followed by NCorH. The most interesting thing to retain is that the bag-of-paths framework [17, 26] compute the co-occurrence probabilities, mainly for NCorH, in a closed form whereas DW approximates these probabilities by sampling randoms walks according to an uniform distribution. In practice, we observed that DW generally works well for the classification tasks but collapses for the graph reconstruction and the neighborhood preservation. Finally, tSNE is the best choice for the neighborhood preservation task.

# Further work

This master thesis being in the continuity of [17, 26], we plan to explore more recent deep learning based methods such as SDNE (Structural Deep Network Embedding) [56] which exploits the first-order and second-order proximity to preserve the network structure, DANE (Deep Attributed Network Embedding) [20] which can capture the high non-linearity and preserve various proximities in both topological structure and node attributes, Graph2Gauss (Deep Gaussian Embedding of graphs) [6] which adopted a personalized ranking formulation with respect to the node distances that exploits the natural ordering of the nodes imposed by the network structure in order to learn the embedding . . . to name a few ones.

Finally, we have found that there are not many powerful tools for evaluating embeddings efficiently and differently. This year, EvalNE (A Framework for Evaluating Network Embeddings on Link Prediction) [40] was published but there is no available tool for the graph reconstruction task, for instance. With this in mind, we plan to implement a novel framework on top of some existing python libraries, let's call it for the moment **EmbEval**, allowing the classification (supervised and semi-supervised), the network reconstruction (NR) and the neighborhood preservation (NP) tasks. The key strengths of this tool will be: Firstly, the *formatting of outputs* since an embedding or a set of embeddings can be provided as input. Secondly, the *user-friendliness* since it is not always easy to interact (internally) with different (original) implementations of the embedding methods, as is the case for [40]. Different distance metrics and similarity measures will be available for the NR and NP tasks such as Manhattan, Minkowski , Mahalanobis, . . . in addition to the Euclidean distance and cosine similarity used in this work. Thirdly, since we will deal with a large amount of embedding data, the *parallelizability* aspect should also be taken into account. So, one will speed up the process by using multiple threads on the same machine if needed.

The architecture of the EmbEval framework (inspired by [40]) is depicted in Fig. 6.1. Indeed, EmbEval will consist of three main components: First, the *Input module*, it will be in charge of reading the embedding or a set of embedding files (under feature matrix format) as well as the original graph structure under various formats (edge list, adjacency list, adjacency matrix, . . .). Second, the *Evaluation module*, which will implement the three target tasks interacting, on one hand, with the python machine learning library *Scikit-Learn* providing the implementations of many algorithms (SVM, nearest neighbors, random forest, . . .) and distance metrics. On the other hand, with the python module *DR Quality* for assessing the quality of nonlinear dimensionality reduction rank-based criteria [31, 32, 33, 34], giving back the $R_{NX}(K)$ and the $AUC$. Third, the *Formatting module*, which will interact

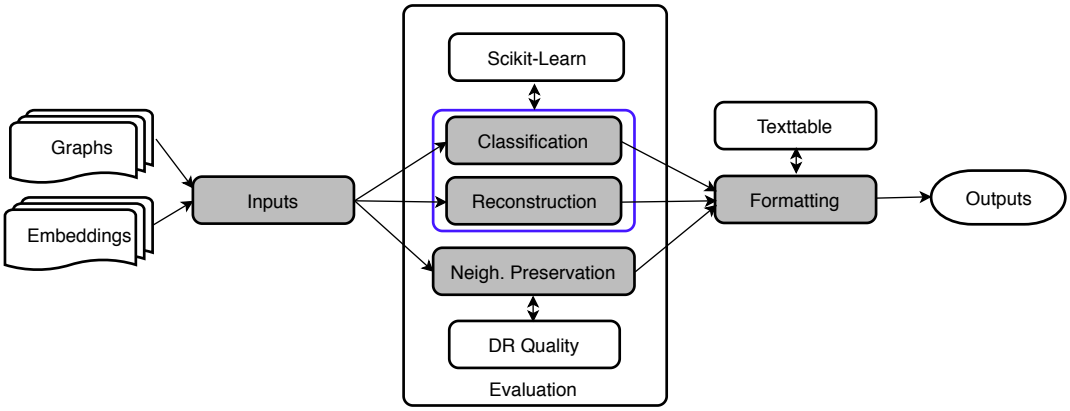



**Fig. 6.1.:** Architecture of EmbEval framework for evaluating network embeddings on classification, reconstruction and neighborhood preservation tasks. The three undarked rectangles are the python external modules that it will interact with.

with the python module *TextTable*, in order to create proper ASCII tables to present the results and save them into a text file for further utilizations. The output will depend on the evaluation task. For instance, for the classification task, the possible columns in a table could be *Dataset name, Accuracy,* $F_1$*-micro,* $F_1$*-macro,* $F_1$*-weighted* and the *best hyper-parameters* for the classifier used (e.g., the kernel type and the penalization constant $c$ for SVM). For NR and NP, two columns could be enough, the *Dataset name* and the *MAP/AUC value*.

# Bibliography


[1]Charu C. Aggarwal. *Social Network Data Analytics*. 1st. Springer Publishing Company, Incorporated, 2011 (cit. on p. 12).

[2]Ethem Alpaydin. *Introduction to Machine Learning*. 2nd. The MIT Press, 2010 (cit. on pp. 5, 6).

[3]Albert-László Barabási and Márton Pósfai. *Network science*. Cambridge University Press, 2016 (cit. on p. 4).

[4]Mikhail Belkin and Partha Niyogi. "Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering". In: *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*. 2001, pp. 585–591 (cit. on p. 2).

[5]Christopher Bishop. *Pattern recognition and machine learning*. Springer, 2006 (cit. on pp. 5, 6).

[6]Aleksandar Bojchevski and Stephan Günnemann. "Deep Gaussian Embedding of Graphs: Unsupervised Inductive Learning via Ranking". In: *International Conference on Learning Representations*. 2018 (cit. on p. 53).

[7]Hongyun Cai, Vincent W Zheng, and Kevin Chang. "A comprehensive survey of graph embedding: problems, techniques and applications". In: *IEEE Transactions on Knowledge and Data Engineering* (2018) (cit. on p. 1).

[8]Peng Cui, Xiao Wang, Jian Pei, and Wenwu Zhu. "A Survey on Network Embedding". In: *Computing Research Repository* abs/1711.08752 (2017) (cit. on p. 37).

[10]Janez Demšar. "Statistical Comparisons of Classifiers over Multiple Data Sets". In: *Journal of Machine Learning Research* 7 (2006), pp. 1–30 (cit. on pp. 14, 15).

[11]R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification*. 2nd. Wiley, 2001 (cit. on pp. 5, 6).

[12]Pierre Dupont. "LINGI2262: Support Vector Machines". In: (2018) (cit. on p. 7).

[13]Sommer F., Fouss F., and M. Saerens. "Comparison of Graph Node Distances on Clustering Tasks". In: *Artificial Neural Networks and Machine Learning – ICANN 2016. Lecture Notes in Computer Science* 9886 (2016). Springer, pp. 192–201 (cit. on pp. 9, 10).

[14]Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. "LIBLINEAR: A Library for Large Linear Classification". In: *Journal of Machine Learning Research* 9 (2008), pp. 1871–1874 (cit. on p. 24).

[15]F. Fouss, K. Francoisse, L. Yen, A. Pirotte, and M. Saerens. “An experimental investigation of kernels on graphs for collaborative recommendation and semisupervised classification”. In: *Neural Networks* 31 (2012), pp. 53–72 (cit. on pp. 8, 12).

[16]F. Fouss, M. Saerens, and M. Shimbo. *Algorithms and models for network data and link analysis*. In preparation. Cambridge University Press, 2016 (cit. on pp. 4, 5, 9, 12).

[17]K. Francoisse, I. Kivimäki, A. Mantrach, F. Rossi, and M. Saerens. “A bag-of-paths framework for network data analysis”. In: *Neural Networks* 90 (2017), pp. 90–111 (cit. on pp. 1, 2, 9, 10, 23, 24, 52, 53).

[18]Milton Friedman. “A Comparison of Alternative Tests of Significance for the Problem of $m$ Rankings”. In: *The Annals of Mathematical Statistics* 11.1 (1940), pp. 86–92 (cit. on p. 14).

[19]Milton Friedman. “The use of ranks to avoid the assumption of normality implicit in the analysis of variance”. In: *Journal of the american statistical association* 32 (1937), pp. 675–701 (cit. on p. 14).

[20]Hongchang Gao and Heng Huang. “Deep Attributed Network Embedding”. In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, 2018, pp. 3364–3370 (cit. on p. 53).

[21]Jean Dickinson Gibbons and Subhabrata Chakraborti. *Nonparametric Statistical Inference*. Springer Berlin Heidelberg, 2011, pp. 977–979 (cit. on p. 14).

[22]Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. 3rd. The Johns Hopkins University Press, 1996 (cit. on p. 20).

[23]Palash Goyal and Emilio Ferrara. “Graph embedding techniques, applications, and performance: A survey”. In: *Knowledge-Based Systems* 151 (2018), pp. 78–94 (cit. on p. 37).

[24]Aditya Grover and Jure Leskovec. “node2vec: Scalable feature learning for networks”. In: *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 2016, pp. 855–864 (cit. on p. 2).

[25]Thomas Grtner. *Kernels For Structured Data*. World Scientific Publishing Co., Inc., 2009 (cit. on p. 12).

[26]Guillaume Guex, Sylvain Courtain, and Marco Saerens. “Covariance and Correlation Kernels on a Graph in the Generalized Bag-of-Paths Formalism”. In: *Computing Research Repository* abs/1902.03002 (2019) (cit. on pp. 9, 11, 12, 23, 24, 52, 53).

[27]Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer New York Inc., 2001 (cit. on p. 6).

[28]Daniel Jurafsky and James H. Martin. *Speech and Language Processing*. 2nd. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2009 (cit. on p. 13).

[29]Ilkka Kivimäki, Masashi Shimbo, and Marco 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 (cit. on p. 10).

[30]E.D. Kolaczyk. “Statistical Analysis of Network Data: Methods and Models”. In: *Springer Series In Statistics* (2009), p. 386 (cit. on p. 8).

[31]John A. Lee and Michel Verleysen. *Nonlinear Dimensionality Reduction*. 1st. Springer Publishing Company, Incorporated, 2007 (cit. on pp. 42, 53).

[32]John A. Lee and Michel Verleysen. "Quality Assessment of Dimensionality Reduction: Rank-based Criteria". In: *Neurocomput.* 72.7-9 (2009), pp. 1431–1443 (cit. on pp. 42, 53).

[33]John A. Lee and Michel Verleysen. "Scale-independent Quality Criteria for Dimensionality Reduction". In: *Pattern Recogn. Lett.* 31.14 (2010), pp. 2248–2257 (cit. on pp. 42, 53).

[34]John Lee and Michel 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 at ECML/PKDD 2008*. Vol. 4. 2008, pp. 21–35 (cit. on pp. 42, 53).

[35]John M. Libert, Shahram Orandi, Michael D. Garris, and John D. Grantham. "Effects of Decomposition Levels and Quality Layers with JPEG 2000 Compression of 1000 ppi Fingerprint Images". In: (2014) (cit. on p. 15).

[36]Laurens van der Maaten. "Learning a Parametric Embedding by Preserving Local Structure". In: *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*. Vol. 5. 2009, pp. 384–391 (cit. on p. 22).

[37]Laurens van der Maaten and Geoffrey Hinton. "Visualizing Data using t-SNE". In: *Journal of Machine Learning Research* 9 (2008), pp. 2579–2605 (cit. on pp. 20, 22).

[38]Amin Mantrach, Nicolas van Zeebroeck, Pascal Francq, et al. "Semi-supervised classification and betweenness computation on large, sparse, directed graphs". In: *Pattern Recognition* 44.6 (2011), pp. 1212–1224 (cit. on pp. 1, 8).

[40]Alexandru Mara, Jefrey Lijffijt, and Tijl De Bie. "EvalNE: A Framework for Evaluating Network Embeddings on Link Prediction". In: *Computing Research Repository* abs/1901.09691 (2019) (cit. on p. 53).

[41]Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. "Efficient Estimation of Word Representations in Vector Space". In: *Computing Research Repository* abs/1301.3781 (2013) (cit. on p. 17).

[42]Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. "Distributed Representations of Words and Phrases and Their Compositionality". In: *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*. 2013, pp. 3111–3119 (cit. on p. 17).

[43]Tom M. Mitchell. *Machine learning, International Edition*. McGraw-Hill, 1997 (cit. on pp. 5, 6).

[44]Andriy Mnih and Geoffrey Hinton. "A Scalable Hierarchical Distributed Language Model". In: *Advances in Neural Information Processing Systems 21*. 2009, pp. 1081–1088 (cit. on p. 18).

[45]Frederic Morin and Yoshua Bengio. "Hierarchical Probabilistic Neural Network Language Model". In: *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*. 2005 (cit. on p. 18).

[46]Kevin P. Murphy. *Machine learning : a probabilistic perspective*. MIT Press, 2013 (cit. on pp. 5, 6).

[47]P. Nemenyi. *Distribution-free Multiple Comparisons*. PhD thesis. Princeton University, 1963 (cit. on p. 15).

[48]M. E. J. Newman. *Networks: an introduction*. Oxford University Press, 2010 (cit. on pp. 4, 12).

[49]P.J. Olver and C. Shakiban. *Applied Linear Algebra*. Prentice Hall, 2006 (cit. on p. 12).

[50]Mingdong Ou, Peng Cui, Jian Pei, Ziwei Zhang, and Wenwu Zhu. "Asymmetric Transitivity Preserving Graph Embedding". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2016, pp. 1105–1114 (cit. on p. 2).

[51]Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. "DeepWalk: Online Learning of Social Representations". In: *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2014, pp. 701–710 (cit. on pp. 2, 16, 18).

[52]B. Schölkopf and AJ. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2002 (cit. on p. 12).

[53]John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004 (cit. on p. 12).

[55]Merijn van Erp and Lambert Schomaker. "Variants of the Borda count method for combining ranked classifier hypotheses". In: *Proceedings 7th International Workshop on frontiers in handwriting recognition (7th IWFHR)*. 2000, pp. 443–452 (cit. on p. 13).

[56]Daixin Wang, Peng Cui, and Wenwu Zhu. "Structural Deep Network Embedding". In: *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2016, pp. 1225–1234 (cit. on pp. 37, 53).

[57]Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Y. Chang. "Network Representation Learning with Rich Text Information". In: *Proceedings of the 24th International Conference on Artificial Intelligence*. 2015, pp. 2111–2117 (cit. on pp. 17, 19, 20, 25).

[58]Hsiang-Fu Yu, Prateek Jain, Purushottam Kar, and Inderjit Dhillon. "Large-scale Multi-label Learning with Missing Labels". In: *Proceedings of the 31st International Conference on Machine Learning*. Vol. 32. 2014, pp. 593–601 (cit. on p. 20).

[59]Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. "Learning with local and global consistency". In: *Advances in Neural Information Processing Systems 16*. 2004, pp. 321–328 (cit. on p. 8).

## Web pages

[9]*DeepWalk*. URL: `https://github.com/phanein/deepwalk` (visited on Nov. 12, 2018) (cit. on p. 24).

[39]*MAP*. URL: `https://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-ranked-retrieval-results-1.html` (visited on Feb. 10, 2019) (cit. on p. 37).

[54]*tSNE*. URL: `https://lvdmaaten.github.io/tsne/` (visited on Nov. 12, 2018) (cit. on p. 24).

# Appendix

# A

## A.1 Hierarchical Softmax

To efficiently solve equation 3.6, we assign the vertices to the leaves of a binary tree. The benefit is that, instead of enumerating all nodes, only the path from the root to the corresponding leaf needs to be evaluated. So, the problem turns into maximizing the probability of a specific path in the tree. If the path to vertex $u_k$ is identified by a sequence of tree nodes $(b_0, b_1, \ldots, b_{\lceil \log |\mathcal{V}| \rceil})$, ($b_0$ = root, $b_{\lceil \log |\mathcal{V}| \rceil} = u_k$) then

$$\mathrm{P}(u_k|\Phi(v_j)) = \prod_{l=1}^{\lceil \log |\mathcal{V}| \rceil} \mathrm{P}(b_l|\Phi(v_j)) \tag{A.1}$$

where $\mathrm{P}(b_l)$ is a binary classifier and $\mathrm{P}(b_l|\Phi(v_j)) = \sigma(\Phi^T(v_{b_l})\Phi(v_j))$. $\sigma(.)$ denotes the sigmoid function, and $\Phi(v_{b_l})$ is the embedding of tree node $b_l$'s parent. The hierarchical softmax reduces time complexity of SkipGram from $O(|\mathcal{V}|^2)$ to $O(|\mathcal{V}| \log(|\mathcal{V}|))$.

## A.2 MAP using 5 social dimensions

| $\theta$ value: Dataset: | $10^{-6}$ | $10^{-5}$ | $10^{-4}$ | $10^{-3}$ | $10^{-2}$ | $10^{-1}$ | 1 | 10 |
|---|---|---|---|---|---|---|---|---|
| WebKB-texas | 0.45 | 0.48 | 0.49 | 0.45 | 0.50 | 0.52 | 0.52 | 0.48 |
| WebKB-washington | 0.31 | 0.31 | 0.31 | 0.31 | 0.28 | 0.31 | 0.31 | 0.28 |
| WebKB-wisconsin | 0.45 | 0.45 | 0.45 | 0.44 | 0.47 | 0.49 | 0.48 | 0.47 |
| WebKB-cornell | 0.40 | 0.39 | 0.38 | 0.34 | 0.36 | 0.41 | 0.41 | 0.38 |
| Imdb | 0.10 | 0.10 | 0.10 | 0.10 | 0.11 | 0.10 | 0.11 | 0.09 |
| News-2cl-1 | 0.47 | 0.47 | 0.51 | 0.53 | 0.53 | 0.52 | 0.51 | 0.50 |
| News-2cl-2 | 0.34 | 0.34 | 0.36 | 0.38 | 0.38 | 0.36 | 0.35 | 0.35 |
| News-2cl-3 | 0.53 | 0.53 | 0.57 | 0.60 | 0.59 | 0.58 | 0.56 | 0.55 |
| News-3cl-1 | 0.46 | 0.45 | 0.49 | 0.51 | 0.51 | 0.50 | 0.49 | 0.48 |
| News-3cl-2 | 0.43 | 0.44 | 0.49 | 0.51 | 0.51 | 0.50 | 0.48 | 0.47 |
| News-3cl-3 | 0.42 | 0.42 | 0.43 | 0.46 | 0.47 | 0.46 | 0.45 | 0.44 |
| News-5cl-1 | 0.40 | 0.41 | 0.44 | 0.47 | 0.47 | 0.46 | 0.44 | 0.44 |
| News-5cl-2 | 0.39 | 0.39 | 0.41 | 0.43 | 0.43 | 0.41 | 0.40 | 0.40 |
| News-5cl-3 | 0.36 | 0.36 | 0.39 | 0.41 | 0.41 | 0.40 | 0.39 | 0.38 |

**Tab. A.1.:** MAP provided by BoPP-m on reconstruction task using five social dimensions.

| $\theta$ value: Dataset: | $10^{-6}$ | $10^{-5}$ | $10^{-4}$ | $10^{-3}$ | $10^{-2}$ | $10^{-1}$ | 1 | 10 |
|---|---|---|---|---|---|---|---|---|
| WebKB-texas | 0.45 | 0.45 | 0.45 | 0.48 | 0.51 | 0.53 | 0.53 | 0.50 |
| WebKB-washington | 0.30 | 0.30 | 0.29 | 0.30 | 0.31 | 0.32 | 0.33 | 0.30 |
| WebKB-wisconsin | 0.43 | 0.43 | 0.44 | 0.45 | 0.49 | 0.51 | 0.50 | 0.48 |
| WebKB-cornell | 0.35 | 0.35 | 0.36 | 0.37 | 0.40 | 0.43 | 0.42 | 0.40 |
| Imdb | 0.08 | 0.08 | 0.09 | 0.11 | 0.11 | 0.11 | 0.11 | 0.11 |
| News-2cl-1 | 0.55 | 0.55 | 0.54 | 0.55 | 0.54 | 0.52 | 0.50 | 0.50 |
| News-2cl-2 | 0.39 | 0.39 | 0.39 | 0.39 | 0.38 | 0.36 | 0.35 | 0.34 |
| News-2cl-3 | 0.60 | 0.61 | 0.61 | 0.60 | 0.59 | 0.57 | 0.55 | 0.55 |
| News-3cl-1 | 0.52 | 0.53 | 0.53 | 0.52 | 0.50 | 0.49 | 0.48 | 0.48 |
| News-3cl-2 | 0.52 | 0.52 | 0.52 | 0.51 | 0.50 | 0.48 | 0.47 | 0.46 |
| News-3cl-3 | 0.46 | 0.48 | 0.49 | 0.49 | 0.47 | 0.45 | 0.44 | 0.44 |
| News-5cl-1 | 0.45 | 0.46 | 0.48 | 0.47 | 0.46 | 0.45 | 0.43 | 0.43 |
| News-5cl-2 | 0.42 | 0.43 | 0.44 | 0.43 | 0.42 | 0.40 | 0.39 | 0.39 |
| News-5cl-3 | 0.42 | 0.43 | 0.42 | 0.42 | 0.41 | 0.39 | 0.38 | 0.37 |

**Tab. A.2.:** MAP provided by BoPP-g on reconstruction task using five social dimensions.

| $\gamma$ value: Dataset: | 5 | 15 | 30 | 40 | 80 | 120 | 160 | 300 | 500 | 1000 |
|---|---|---|---|---|---|---|---|---|---|---|
| WebKB-texas | 0.49 | 0.50 | 0.50 | 0.50 | 0.50 | 0.50 | 0.50 | 0.50 | 0.50 | 0.50 |
| WebKB-washington | 0.30 | 0.32 | 0.33 | 0.33 | 0.32 | 0.32 | 0.32 | 0.32 | 0.32 | 0.32 |
| WebKB-wisconsin | 0.46 | 0.48 | 0.47 | 0.48 | 0.47 | 0.48 | 0.48 | 0.48 | 0.48 | 0.48 |
| WebKB-cornell | 0.39 | 0.42 | 0.42 | 0.42 | 0.42 | 0.42 | 0.42 | 0.42 | 0.42 | 0.41 |
| Imdb | 0.14 | 0.14 | 0.14 | 0.14 | 0.14 | 0.14 | 0.14 | 0.14 | 0.15 | 0.15 |
| News-2cl-1 | 0.51 | 0.53 | 0.52 | 0.53 | 0.53 | 0.53 | 0.53 | 0.53 | 0.53 | 0.53 |
| News-2cl-2 | 0.35 | 0.37 | 0.37 | 0.37 | 0.36 | 0.37 | 0.37 | 0.37 | 0.37 | 0.37 |
| News-2cl-3 | 0.57 | 0.58 | 0.59 | 0.59 | 0.59 | 0.59 | 0.59 | 0.59 | 0.59 | 0.59 |
| News-3cl-1 | 0.50 | 0.51 | 0.52 | 0.51 | 0.52 | 0.52 | 0.52 | 0.52 | 0.52 | 0.52 |
| News-3cl-2 | 0.50 | 0.51 | 0.51 | 0.51 | 0.51 | 0.51 | 0.51 | 0.51 | 0.51 | 0.51 |
| News-3cl-3 | 0.44 | 0.45 | 0.45 | 0.45 | 0.45 | 0.45 | 0.45 | 0.45 | 0.45 | 0.45 |
| News-5cl-1 | 0.44 | 0.46 | 0.46 | 0.46 | 0.46 | 0.46 | 0.46 | 0.46 | 0.46 | 0.46 |
| News-5cl-2 | 0.39 | 0.41 | 0.42 | 0.41 | 0.42 | 0.42 | 0.42 | 0.42 | 0.42 | 0.41 |
| News-5cl-3 | 0.38 | 0.40 | 0.41 | 0.40 | 0.41 | 0.41 | 0.41 | 0.41 | 0.40 | 0.40 |

**Tab. A.3.:** MAP provided by DWg on reconstruction task using five social dimensions.

| *t* value: Dataset: | 5 | 15 | 30 | 40 | 80 | 120 | 160 | 300 | 500 | 1000 |
|---|---|---|---|---|---|---|---|---|---|---|
| WebKB-texas | 0.50 | 0.50 | 0.50 | 0.50 | 0.50 | 0.50 | 0.50 | 0.50 | 0.49 | 0.49 |
| WebKB-washington | 0.31 | 0.32 | 0.33 | 0.33 | 0.32 | 0.32 | 0.32 | 0.32 | 0.31 | 0.30 |
| WebKB-wisconsin | 0.48 | 0.48 | 0.48 | 0.48 | 0.47 | 0.48 | 0.48 | 0.47 | 0.47 | 0.47 |
| WebKB-cornell | 0.41 | 0.42 | 0.42 | 0.42 | 0.42 | 0.42 | 0.41 | 0.41 | 0.41 | 0.41 |
| Imdb | 0.14 | 0.14 | 0.14 | 0.14 | 0.14 | 0.14 | 0.15 | 0.14 | 0.15 | 0.14 |
| News-2cl-1 | 0.52 | 0.53 | 0.53 | 0.53 | 0.53 | 0.52 | 0.53 | 0.53 | 0.53 | 0.53 |
| News-2cl-2 | 0.36 | 0.37 | 0.37 | 0.37 | 0.36 | 0.37 | 0.37 | 0.37 | 0.37 | 0.37 |
| News-2cl-3 | 0.58 | 0.59 | 0.59 | 0.59 | 0.59 | 0.59 | 0.59 | 0.59 | 0.59 | 0.59 |
| News-3cl-1 | 0.51 | 0.52 | 0.51 | 0.52 | 0.52 | 0.51 | 0.52 | 0.52 | 0.52 | 0.52 |
| News-3cl-2 | 0.50 | 0.51 | 0.51 | 0.51 | 0.51 | 0.51 | 0.51 | 0.51 | 0.51 | 0.51 |
| News-3cl-3 | 0.45 | 0.45 | 0.45 | 0.45 | 0.45 | 0.45 | 0.45 | 0.45 | 0.45 | 0.45 |
| News-5cl-1 | 0.45 | 0.46 | 0.46 | 0.46 | 0.46 | 0.46 | 0.46 | 0.46 | 0.46 | 0.46 |
| News-5cl-2 | 0.40 | 0.41 | 0.42 | 0.42 | 0.41 | 0.41 | 0.41 | 0.41 | 0.42 | 0.42 |
| News-5cl-3 | 0.40 | 0.40 | 0.41 | 0.41 | 0.41 | 0.41 | 0.41 | 0.41 | 0.41 | 0.41 |

**Tab. A.4.:** MAP provided by DWt on reconstruction task using five social dimensions.

| *k* order: Dataset: | 2 | 3 | 5 | 10 | 20 | 40 | 50 |
|---|---|---|---|---|---|---|---|
| WebKB-texas | 0.51 | 0.51 | 0.51 | 0.51 | 0.48 | 0.30 | 0.30 |
| WebKB-washington | 0.30 | 0.30 | 0.29 | 0.29 | 0.28 | 0.16 | 0.16 |
| WebKB-wisconsin | 0.47 | 0.45 | 0.41 | 0.38 | 0.38 | 0.30 | 0.30 |
| WebKB-cornell | 0.41 | 0.40 | 0.38 | 0.36 | 0.36 | 0.24 | 0.24 |
| Imdb | 0.10 | 0.10 | 0.10 | 0.10 | 0.09 | 0.09 | 0.09 |
| News-2cl-1 | 0.53 | 0.53 | 0.52 | 0.51 | 0.50 | 0.50 | 0.50 |
| News-2cl-2 | 0.37 | 0.37 | 0.36 | 0.34 | 0.34 | 0.34 | 0.34 |
| News-2cl-3 | 0.58 | 0.58 | 0.58 | 0.57 | 0.56 | 0.56 | 0.49 |
| News-3cl-1 | 0.49 | 0.49 | 0.48 | 0.45 | 0.44 | 0.45 | 0.45 |
| News-3cl-2 | 0.50 | 0.50 | 0.49 | 0.47 | 0.47 | 0.47 | 0.47 |
| News-3cl-3 | 0.46 | 0.46 | 0.46 | 0.45 | 0.44 | 0.42 | 0.41 |
| News-5cl-1 | 0.45 | 0.45 | 0.45 | 0.44 | 0.43 | 0.42 | 0.36 |
| News-5cl-2 | 0.41 | 0.41 | 0.40 | 0.39 | 0.38 | 0.38 | 0.36 |
| News-5cl-3 | 0.40 | 0.40 | 0.40 | 0.39 | 0.38 | 0.36 | 0.34 |

**Tab. A.5.:** MAP provided by MFDW on reconstruction task using five social dimensions.

| $\theta$ value: Dataset: | $10^{-6}$ | $10^{-5}$ | $10^{-4}$ | $10^{-3}$ | $10^{-2}$ | $10^{-1}$ | 1 | 10 |
|---|---|---|---|---|---|---|---|---|
| WebKB-texas | 0.47 | 0.47 | 0.49 | 0.49 | 0.52 | 0.54 | 0.48 | 0.37 |
| WebKB-washington | 0.29 | 0.29 | 0.29 | 0.30 | 0.30 | 0.31 | 0.30 | 0.18 |
| WebKB-wisconsin | 0.46 | 0.46 | 0.46 | 0.47 | 0.49 | 0.50 | 0.42 | 0.36 |
| WebKB-cornell | 0.38 | 0.38 | 0.38 | 0.39 | 0.41 | 0.42 | 0.38 | 0.33 |
| Imdb | 0.09 | 0.09 | 0.10 | 0.11 | 0.11 | 0.11 | 0.10 | 0.05 |
| News-2cl-1 | 0.50 | 0.51 | 0.52 | 0.52 | 0.53 | 0.50 | 0.44 | 0.43 |
| News-2cl-2 | 0.35 | 0.35 | 0.36 | 0.37 | 0.37 | 0.33 | 0.28 | 0.28 |
| News-2cl-3 | 0.56 | 0.56 | 0.57 | 0.57 | 0.58 | 0.54 | 0.50 | 0.46 |
| News-3cl-1 | 0.48 | 0.48 | 0.49 | 0.48 | 0.50 | 0.46 | 0.41 | 0.39 |
| News-3cl-2 | 0.48 | 0.48 | 0.49 | 0.49 | 0.50 | 0.45 | 0.39 | 0.38 |
| News-3cl-3 | 0.45 | 0.45 | 0.46 | 0.45 | 0.47 | 0.43 | 0.39 | 0.37 |
| News-5cl-1 | 0.43 | 0.43 | 0.45 | 0.44 | 0.46 | 0.40 | 0.37 | 0.36 |
| News-5cl-2 | 0.40 | 0.40 | 0.40 | 0.39 | 0.41 | 0.39 | 0.34 | 0.33 |
| News-5cl-3 | 0.38 | 0.38 | 0.40 | 0.39 | 0.40 | 0.37 | 0.33 | 0.32 |

**Tab. A.6.:** MAP provided by CovH on reconstruction task using five social dimensions.

| $\theta$ value: Dataset: | $10^{-6}$ | $10^{-5}$ | $10^{-4}$ | $10^{-3}$ | $10^{-2}$ | $10^{-1}$ | 1 | 10 |
|---|---|---|---|---|---|---|---|---|
| WebKB-texas | 0.46 | 0.46 | 0.48 | 0.52 | 0.52 | 0.53 | 0.52 | 0.38 |
| WebKB-washington | 0.31 | 0.31 | 0.31 | 0.31 | 0.31 | 0.32 | 0.32 | 0.18 |
| WebKB-wisconsin | 0.44 | 0.44 | 0.45 | 0.49 | 0.50 | 0.51 | 0.47 | 0.37 |
| WebKB-cornell | 0.39 | 0.39 | 0.40 | 0.41 | 0.42 | 0.43 | 0.41 | 0.31 |
| Imdb | 0.10 | 0.10 | 0.10 | 0.11 | 0.11 | 0.11 | 0.11 | 0.05 |
| News-2cl-1 | 0.53 | 0.53 | 0.54 | 0.54 | 0.53 | 0.51 | 0.44 | 0.43 |
| News-2cl-2 | 0.39 | 0.39 | 0.39 | 0.38 | 0.37 | 0.35 | 0.28 | 0.28 |
| News-2cl-3 | 0.61 | 0.60 | 0.60 | 0.59 | 0.58 | 0.55 | 0.50 | 0.46 |
| News-3cl-1 | 0.51 | 0.52 | 0.51 | 0.51 | 0.51 | 0.47 | 0.41 | 0.40 |
| News-3cl-2 | 0.51 | 0.51 | 0.51 | 0.50 | 0.50 | 0.47 | 0.40 | 0.38 |
| News-3cl-3 | 0.47 | 0.48 | 0.48 | 0.48 | 0.47 | 0.44 | 0.40 | 0.37 |
| News-5cl-1 | 0.46 | 0.47 | 0.47 | 0.47 | 0.47 | 0.42 | 0.37 | 0.36 |
| News-5cl-2 | 0.42 | 0.43 | 0.43 | 0.42 | 0.41 | 0.40 | 0.34 | 0.33 |
| News-5cl-3 | 0.41 | 0.42 | 0.41 | 0.41 | 0.40 | 0.38 | 0.33 | 0.32 |

**Tab. A.7.:** MAP provided by NCorH on reconstruction task using five social dimensions.

| $\theta$ value: Dataset: | $10^{-6}$ | $10^{-5}$ | $10^{-4}$ | $10^{-3}$ | $10^{-2}$ | $10^{-1}$ | 1 | 10 |
|---|---|---|---|---|---|---|---|---|
| WebKB-texas | 0.40 | 0.39 | 0.40 | 0.39 | 0.42 | 0.48 | 0.47 | 0.46 |
| WebKB-washington | 0.20 | 0.19 | 0.20 | 0.19 | 0.20 | 0.25 | 0.28 | 0.25 |
| WebKB-wisconsin | 0.37 | 0.38 | 0.38 | 0.37 | 0.38 | 0.44 | 0.42 | 0.41 |
| WebKB-cornell | 0.29 | 0.28 | 0.28 | 0.28 | 0.30 | 0.37 | 0.36 | 0.35 |
| Imdb | 0.08 | 0.08 | 0.09 | 0.09 | 0.10 | 0.11 | 0.11 | 0.11 |
| News-2cl-1 | 0.49 | 0.49 | 0.49 | 0.49 | 0.50 | 0.51 | 0.49 | 0.49 |
| News-2cl-2 | 0.35 | 0.34 | 0.34 | 0.34 | 0.35 | 0.34 | 0.34 | 0.34 |
| News-2cl-3 | 0.54 | 0.54 | 0.54 | 0.55 | 0.56 | 0.56 | 0.55 | 0.55 |
| News-3cl-1 | 0.47 | 0.47 | 0.47 | 0.47 | 0.49 | 0.49 | 0.47 | 0.47 |
| News-3cl-2 | 0.47 | 0.47 | 0.47 | 0.47 | 0.48 | 0.48 | 0.47 | 0.46 |
| News-3cl-3 | 0.42 | 0.41 | 0.43 | 0.43 | 0.44 | 0.44 | 0.44 | 0.43 |
| News-5cl-1 | 0.42 | 0.43 | 0.43 | 0.43 | 0.44 | 0.44 | 0.44 | 0.43 |
| News-5cl-2 | 0.37 | 0.38 | 0.38 | 0.39 | 0.40 | 0.40 | 0.39 | 0.39 |
| News-5cl-3 | 0.36 | 0.37 | 0.36 | 0.38 | 0.38 | 0.38 | 0.37 | 0.37 |

**Tab. A.8.:** MAP provided by tSNE on reconstruction task using five social dimensions.

## A.3 MAP using 2 social dimensions

| $\theta$ value: Dataset: | $10^{-6}$ | $10^{-5}$ | $10^{-4}$ | $10^{-3}$ | $10^{-2}$ | $10^{-1}$ | 1 | 10 |
|---|---|---|---|---|---|---|---|---|
| WebKB-texas | 0.43 | 0.43 | 0.43 | 0.43 | 0.44 | 0.49 | 0.49 | 0.47 |
| WebKB-washington | 0.30 | 0.30 | 0.28 | 0.28 | 0.25 | 0.28 | 0.28 | 0.26 |
| WebKB-wisconsin | 0.40 | 0.40 | 0.41 | 0.43 | 0.42 | 0.44 | 0.46 | 0.43 |
| WebKB-cornell | 0.38 | 0.37 | 0.36 | 0.32 | 0.31 | 0.38 | 0.38 | 0.37 |
| Imdb | 0.08 | 0.08 | 0.09 | 0.09 | 0.09 | 0.09 | 0.09 | 0.08 |
| news-2cl-1 | 0.46 | 0.46 | 0.46 | 0.50 | 0.50 | 0.49 | 0.49 | 0.48 |
| news-2cl-2 | 0.31 | 0.31 | 0.33 | 0.34 | 0.34 | 0.33 | 0.32 | 0.32 |
| news-2cl-3 | 0.51 | 0.51 | 0.54 | 0.56 | 0.55 | 0.54 | 0.54 | 0.54 |
| news-3cl-1 | 0.42 | 0.43 | 0.43 | 0.48 | 0.48 | 0.47 | 0.47 | 0.46 |
| news-3cl-2 | 0.42 | 0.42 | 0.42 | 0.48 | 0.48 | 0.47 | 0.47 | 0.46 |
| news-3cl-3 | 0.40 | 0.40 | 0.40 | 0.44 | 0.44 | 0.43 | 0.43 | 0.42 |
| news-5cl-1 | 0.40 | 0.41 | 0.38 | 0.44 | 0.44 | 0.43 | 0.42 | 0.42 |
| news-5cl-2 | 0.38 | 0.38 | 0.39 | 0.40 | 0.40 | 0.39 | 0.38 | 0.38 |
| news-5cl-3 | 0.35 | 0.35 | 0.36 | 0.38 | 0.38 | 0.37 | 0.36 | 0.36 |

**Tab. A.9.:** MAP provided by BoPP-m using 2 social dimensions.

| $\theta$ value: Dataset: | $10^{-6}$ | $10^{-5}$ | $10^{-4}$ | $10^{-3}$ | $10^{-2}$ | $10^{-1}$ | 1 | 10 |
|---|---|---|---|---|---|---|---|---|
| WebKB-texas | 0.43 | 0.43 | 0.43 | 0.41 | 0.46 | 0.47 | 0.47 | 0.45 |
| WebKB-washington | 0.29 | 0.29 | 0.28 | 0.27 | 0.28 | 0.26 | 0.28 | 0.25 |
| WebKB-wisconsin | 0.41 | 0.41 | 0.42 | 0.41 | 0.43 | 0.43 | 0.43 | 0.41 |
| WebKB-cornell | 0.34 | 0.34 | 0.34 | 0.33 | 0.37 | 0.36 | 0.36 | 0.36 |
| Imdb | 0.07 | 0.07 | 0.07 | 0.09 | 0.09 | 0.09 | 0.08 | 0.08 |
| news-2cl-1 | 0.51 | 0.51 | 0.51 | 0.50 | 0.49 | 0.48 | 0.48 | 0.48 |
| news-2cl-2 | 0.35 | 0.35 | 0.35 | 0.34 | 0.33 | 0.32 | 0.32 | 0.32 |
| news-2cl-3 | 0.57 | 0.56 | 0.56 | 0.55 | 0.54 | 0.53 | 0.52 | 0.52 |
| news-3cl-1 | 0.45 | 0.46 | 0.46 | 0.46 | 0.46 | 0.45 | 0.44 | 0.44 |
| news-3cl-2 | 0.48 | 0.48 | 0.48 | 0.47 | 0.47 | 0.45 | 0.44 | 0.44 |
| news-3cl-3 | 0.42 | 0.45 | 0.44 | 0.43 | 0.42 | 0.42 | 0.41 | 0.41 |
| news-5cl-1 | 0.38 | 0.43 | 0.44 | 0.43 | 0.42 | 0.41 | 0.40 | 0.40 |
| news-5cl-2 | 0.36 | 0.40 | 0.39 | 0.39 | 0.38 | 0.37 | 0.37 | 0.37 |
| news-5cl-3 | 0.35 | 0.39 | 0.38 | 0.38 | 0.37 | 0.35 | 0.35 | 0.35 |

**Tab. A.10.:** MAP provided by BoPP-g using 2 social dimensions.

| $k$ order: Dataset: | 2 | 3 | 5 | 10 | 20 | 40 | 50 |
|---|---|---|---|---|---|---|---|
| WebKB-texas | 0.44 | 0.44 | 0.44 | 0.44 | 0.44 | 0.30 | 0.30 |
| WebKB-washington | 0.27 | 0.27 | 0.27 | 0.27 | 0.27 | 0.16 | 0.16 |
| WebKB-wisconsin | 0.38 | 0.38 | 0.38 | 0.38 | 0.38 | 0.30 | 0.30 |
| WebKB-cornell | 0.35 | 0.35 | 0.35 | 0.35 | 0.35 | 0.24 | 0.24 |
| Imdb | 0.08 | 0.08 | 0.08 | 0.08 | 0.08 | 0.08 | 0.08 |
| news-2cl-1 | 0.50 | 0.50 | 0.50 | 0.50 | 0.50 | 0.50 | 0.50 |
| news-2cl-2 | 0.34 | 0.34 | 0.34 | 0.34 | 0.34 | 0.34 | 0.34 |
| news-2cl-3 | 0.55 | 0.55 | 0.55 | 0.55 | 0.55 | 0.56 | 0.49 |
| news-3cl-1 | 0.44 | 0.44 | 0.44 | 0.44 | 0.44 | 0.45 | 0.45 |
| news-3cl-2 | 0.47 | 0.47 | 0.47 | 0.47 | 0.47 | 0.47 | 0.47 |
| news-3cl-3 | 0.41 | 0.41 | 0.41 | 0.41 | 0.41 | 0.41 | 0.41 |
| news-5cl-1 | 0.43 | 0.43 | 0.43 | 0.42 | 0.42 | 0.42 | 0.36 |
| news-5cl-2 | 0.38 | 0.38 | 0.38 | 0.38 | 0.38 | 0.38 | 0.36 |
| news-5cl-3 | 0.35 | 0.35 | 0.35 | 0.35 | 0.35 | 0.35 | 0.34 |

**Tab. A.11.:** MAP provided by MFDW using 2 social dimensions.

| $\theta$ value: Dataset: | $10^{-6}$ | $10^{-5}$ | $10^{-4}$ | $10^{-3}$ | $10^{-2}$ | $10^{-1}$ | 1 | 10 |
|---|---|---|---|---|---|---|---|---|
| WebKB-texas | 0.43 | 0.43 | 0.43 | 0.43 | 0.48 | 0.45 | 0.40 | 0.35 |
| WebKB-washington | 0.28 | 0.28 | 0.28 | 0.28 | 0.27 | 0.27 | 0.27 | 0.19 |
| WebKB-wisconsin | 0.42 | 0.42 | 0.42 | 0.42 | 0.37 | 0.40 | 0.41 | 0.34 |
| WebKB-cornell | 0.34 | 0.34 | 0.34 | 0.34 | 0.35 | 0.37 | 0.36 | 0.31 |
| Imdb | 0.07 | 0.07 | 0.07 | 0.09 | 0.09 | 0.09 | 0.08 | 0.05 |
| news-2cl-1 | 0.44 | 0.44 | 0.50 | 0.50 | 0.50 | 0.45 | 0.44 | 0.43 |
| news-2cl-2 | 0.29 | 0.29 | 0.30 | 0.33 | 0.33 | 0.30 | 0.28 | 0.28 |
| news-2cl-3 | 0.48 | 0.48 | 0.53 | 0.55 | 0.55 | 0.53 | 0.49 | 0.47 |
| news-3cl-1 | 0.42 | 0.42 | 0.44 | 0.45 | 0.45 | 0.44 | 0.40 | 0.39 |
| news-3cl-2 | 0.41 | 0.41 | 0.47 | 0.47 | 0.46 | 0.44 | 0.39 | 0.38 |
| news-3cl-3 | 0.39 | 0.39 | 0.43 | 0.42 | 0.41 | 0.40 | 0.39 | 0.38 |
| news-5cl-1 | 0.38 | 0.38 | 0.42 | 0.42 | 0.42 | 0.38 | 0.37 | 0.36 |
| news-5cl-2 | 0.36 | 0.35 | 0.38 | 0.38 | 0.38 | 0.37 | 0.34 | 0.33 |
| news-5cl-3 | 0.33 | 0.33 | 0.37 | 0.36 | 0.35 | 0.34 | 0.32 | 0.32 |

**Tab. A.12.:** MAP provided by CovH using 2 social dimensions.

| $\theta$ value: Dataset: | $10^{-6}$ | $10^{-5}$ | $10^{-4}$ | $10^{-3}$ | $10^{-2}$ | $10^{-1}$ | 1 | 10 |
|---|---|---|---|---|---|---|---|---|
| WebKB-texas | 0.46 | 0.46 | 0.46 | 0.47 | 0.46 | 0.47 | 0.43 | 0.36 |
| WebKB-washington | 0.27 | 0.27 | 0.28 | 0.28 | 0.29 | 0.29 | 0.29 | 0.19 |
| WebKB-wisconsin | 0.41 | 0.41 | 0.41 | 0.41 | 0.40 | 0.40 | 0.42 | 0.36 |
| WebKB-cornell | 0.37 | 0.37 | 0.38 | 0.39 | 0.38 | 0.37 | 0.38 | 0.30 |
| Imdb | 0.08 | 0.09 | 0.09 | 0.09 | 0.09 | 0.09 | 0.09 | 0.05 |
| news-2cl-1 | 0.47 | 0.50 | 0.50 | 0.50 | 0.50 | 0.45 | 0.44 | 0.43 |
| news-2cl-2 | 0.35 | 0.35 | 0.34 | 0.34 | 0.33 | 0.31 | 0.28 | 0.28 |
| news-2cl-3 | 0.55 | 0.55 | 0.55 | 0.55 | 0.54 | 0.53 | 0.49 | 0.46 |
| news-3cl-1 | 0.46 | 0.46 | 0.46 | 0.46 | 0.45 | 0.45 | 0.41 | 0.40 |
| news-3cl-2 | 0.47 | 0.48 | 0.47 | 0.47 | 0.46 | 0.45 | 0.40 | 0.38 |
| news-3cl-3 | 0.43 | 0.44 | 0.43 | 0.43 | 0.43 | 0.41 | 0.39 | 0.38 |
| news-5cl-1 | 0.42 | 0.41 | 0.43 | 0.43 | 0.43 | 0.38 | 0.37 | 0.36 |
| news-5cl-2 | 0.40 | 0.39 | 0.39 | 0.39 | 0.38 | 0.38 | 0.34 | 0.34 |
| news-5cl-3 | 0.38 | 0.38 | 0.37 | 0.37 | 0.36 | 0.35 | 0.32 | 0.32 |

**Tab. A.13.:** MAP provided by NCorH using 2 social dimensions.

| $\theta$ value: Dataset: | $10^{-6}$ | $10^{-5}$ | $10^{-4}$ | $10^{-3}$ | $10^{-2}$ | $10^{-1}$ | 1 | 10 |
|---|---|---|---|---|---|---|---|---|
| WebKB-texas | 0.41 | 0.40 | 0.39 | 0.40 | 0.41 | 0.46 | 0.45 | 0.45 |
| WebKB-washington | 0.20 | 0.19 | 0.18 | 0.19 | 0.20 | 0.23 | 0.25 | 0.25 |
| WebKB-wisconsin | 0.38 | 0.37 | 0.37 | 0.38 | 0.38 | 0.42 | 0.40 | 0.39 |
| WebKB-cornell | 0.30 | 0.29 | 0.29 | 0.30 | 0.31 | 0.36 | 0.34 | 0.34 |
| Imdb | 0.08 | 0.09 | 0.08 | 0.08 | 0.10 | 0.10 | 0.10 | 0.09 |
| news-2cl-1 | 0.48 | 0.49 | 0.49 | 0.48 | 0.49 | 0.49 | 0.48 | 0.48 |
| news-2cl-2 | 0.32 | 0.29 | 0.33 | 0.33 | 0.33 | 0.33 | 0.33 | 0.33 |
| news-2cl-3 | 0.54 | 0.54 | 0.54 | 0.54 | 0.55 | 0.55 | 0.54 | 0.54 |
| news-3cl-1 | 0.45 | 0.44 | 0.47 | 0.47 | 0.47 | 0.47 | 0.46 | 0.47 |
| news-3cl-2 | 0.45 | 0.46 | 0.47 | 0.46 | 0.47 | 0.46 | 0.46 | 0.45 |
| news-3cl-3 | 0.41 | 0.41 | 0.42 | 0.43 | 0.43 | 0.43 | 0.43 | 0.43 |
| news-5cl-1 | 0.39 | 0.38 | 0.42 | 0.42 | 0.43 | 0.43 | 0.43 | 0.42 |
| news-5cl-2 | 0.34 | 0.35 | 0.37 | 0.39 | 0.38 | 0.39 | 0.39 | 0.39 |
| news-5cl-3 | 0.35 | 0.36 | 0.37 | 0.37 | 0.37 | 0.37 | 0.36 | 0.36 |

**Tab. A.14.:** MAP provided by tSNE using 2 social dimensions.

## A.4 AUC using 5 social dimensions

| $\theta$ value: Dataset: | $10^{-6}$ | $10^{-5}$ | $10^{-4}$ | $10^{-3}$ | $10^{-2}$ | $10^{-1}$ | 1 | 10 |
|---|---|---|---|---|---|---|---|---|
| WebKB-texas | 0.08 | 0.09 | 0.10 | 0.06 | 0.08 | 0.11 | 0.13 | 0.16 |
| WebKB-washington | 0.09 | 0.10 | 0.09 | 0.08 | 0.10 | 0.11 | 0.10 | 0.11 |
| WebKB-wisconsin | 0.13 | 0.13 | 0.11 | 0.10 | 0.15 | 0.18 | 0.17 | 0.24 |
| WebKB-cornell | 0.09 | 0.09 | 0.09 | 0.06 | 0.09 | 0.11 | 0.11 | 0.11 |
| Imdb | 0.21 | 0.20 | 0.21 | 0.24 | 0.27 | 0.28 | 0.30 | 0.21 |
| news-2cl-1 | 0.10 | 0.11 | 0.18 | 0.25 | 0.27 | 0.30 | 0.37 | 0.41 |
| news-2cl-2 | 0.11 | 0.12 | 0.19 | 0.24 | 0.26 | 0.30 | 0.36 | 0.39 |
| news-2cl-3 | 0.13 | 0.14 | 0.23 | 0.31 | 0.30 | 0.33 | 0.39 | 0.43 |
| news-3cl-1 | 0.11 | 0.11 | 0.22 | 0.27 | 0.26 | 0.29 | 0.36 | 0.38 |
| news-3cl-2 | 0.08 | 0.08 | 0.17 | 0.24 | 0.23 | 0.25 | 0.30 | 0.34 |
| news-3cl-3 | 0.08 | 0.08 | 0.14 | 0.24 | 0.25 | 0.28 | 0.34 | 0.37 |
| news-5cl-1 | 0.07 | 0.08 | 0.16 | 0.23 | 0.22 | 0.24 | 0.28 | 0.31 |
| news-5cl-2 | 0.07 | 0.08 | 0.16 | 0.22 | 0.22 | 0.23 | 0.26 | 0.30 |
| news-5cl-3 | 0.06 | 0.06 | 0.13 | 0.18 | 0.18 | 0.20 | 0.25 | 0.28 |

**Tab. A.15.:** AUC provided by BoPP-m using five social dimensions.

| $\theta$ value: Dataset: | $10^{-6}$ | $10^{-5}$ | $10^{-4}$ | $10^{-3}$ | $10^{-2}$ | $10^{-1}$ | 1 | 10 |
|---|---|---|---|---|---|---|---|---|
| WebKB-texas | 0.10 | 0.09 | 0.09 | 0.11 | 0.13 | 0.17 | 0.18 | 0.22 |
| WebKB-washington | 0.14 | 0.14 | 0.14 | 0.13 | 0.14 | 0.16 | 0.16 | 0.19 |
| WebKB-wisconsin | 0.10 | 0.10 | 0.12 | 0.15 | 0.16 | 0.18 | 0.20 | 0.24 |
| WebKB-cornell | 0.09 | 0.09 | 0.09 | 0.11 | 0.12 | 0.16 | 0.15 | 0.16 |
| Imdb | 0.14 | 0.15 | 0.18 | 0.29 | 0.32 | 0.32 | 0.32 | 0.33 |
| news-2cl-1 | 0.18 | 0.19 | 0.19 | 0.21 | 0.23 | 0.25 | 0.32 | 0.35 |
| news-2cl-2 | 0.16 | 0.17 | 0.18 | 0.20 | 0.22 | 0.25 | 0.32 | 0.35 |
| news-2cl-3 | 0.18 | 0.23 | 0.23 | 0.24 | 0.26 | 0.29 | 0.35 | 0.40 |
| news-3cl-1 | 0.18 | 0.19 | 0.21 | 0.22 | 0.23 | 0.25 | 0.31 | 0.34 |
| news-3cl-2 | 0.15 | 0.17 | 0.17 | 0.19 | 0.19 | 0.22 | 0.26 | 0.31 |
| news-3cl-3 | 0.13 | 0.16 | 0.19 | 0.20 | 0.21 | 0.23 | 0.30 | 0.34 |
| news-5cl-1 | 0.10 | 0.13 | 0.17 | 0.17 | 0.18 | 0.20 | 0.24 | 0.28 |
| news-5cl-2 | 0.11 | 0.13 | 0.16 | 0.17 | 0.18 | 0.20 | 0.24 | 0.27 |
| news-5cl-3 | 0.12 | 0.14 | 0.14 | 0.14 | 0.15 | 0.18 | 0.22 | 0.25 |

**Tab. A.16.:** AUC provided by BoPP-g using five social dimensions.

| $\gamma$ value: Dataset: | 5 | 15 | 30 | 40 | 80 | 120 | 160 | 300 | 500 | 1000 |
|---|---|---|---|---|---|---|---|---|---|---|
| WebKB-texas | 0.09 | 0.10 | 0.11 | 0.12 | 0.13 | 0.14 | 0.14 | 0.15 | 0.16 | 0.16 |
| WebKB-washington | 0.02 | 0.07 | 0.09 | 0.09 | 0.11 | 0.13 | 0.13 | 0.15 | 0.15 | 0.15 |
| WebKB-wisconsin | 0.06 | 0.10 | 0.10 | 0.11 | 0.13 | 0.13 | 0.13 | 0.14 | 0.13 | 0.15 |
| WebKB-cornell | 0.03 | 0.05 | 0.07 | 0.08 | 0.09 | 0.10 | 0.11 | 0.12 | 0.12 | 0.12 |
| Imdb | 0.15 | 0.17 | 0.18 | 0.19 | 0.20 | 0.21 | 0.22 | 0.22 | 0.23 | 0.22 |
| news-2cl-1 | 0.15 | 0.20 | 0.21 | 0.22 | 0.22 | 0.23 | 0.23 | 0.23 | 0.23 | 0.24 |
| news-2cl-2 | 0.15 | 0.19 | 0.21 | 0.21 | 0.22 | 0.23 | 0.22 | 0.22 | 0.22 | 0.22 |
| news-2cl-3 | 0.18 | 0.23 | 0.25 | 0.25 | 0.24 | 0.25 | 0.25 | 0.26 | 0.26 | 0.26 |
| news-3cl-1 | 0.18 | 0.22 | 0.22 | 0.22 | 0.23 | 0.23 | 0.23 | 0.23 | 0.23 | 0.23 |
| news-3cl-2 | 0.14 | 0.17 | 0.19 | 0.19 | 0.20 | 0.20 | 0.20 | 0.20 | 0.20 | 0.21 |
| news-3cl-3 | 0.15 | 0.18 | 0.20 | 0.20 | 0.21 | 0.21 | 0.20 | 0.21 | 0.21 | 0.21 |
| news-5cl-1 | 0.13 | 0.17 | 0.18 | 0.18 | 0.19 | 0.19 | 0.19 | 0.19 | 0.19 | 0.19 |
| news-5cl-2 | 0.13 | 0.16 | 0.17 | 0.18 | 0.18 | 0.18 | 0.18 | 0.18 | 0.18 | 0.18 |
| news-5cl-3 | 0.10 | 0.14 | 0.15 | 0.15 | 0.15 | 0.15 | 0.15 | 0.15 | 0.15 | 0.16 |

**Tab. A.17.:** AUC provided by DWg using five social dimensions.

| *t* value: Dataset: | 5 | 15 | 30 | 40 | 80 | 120 | 160 | 300 | 500 | 1000 |
|---|---|---|---|---|---|---|---|---|---|---|
| WebKB-texas | 0.11 | 0.12 | 0.12 | 0.12 | 0.12 | 0.12 | 0.12 | 0.14 | 0.14 | 0.14 |
| WebKB-washington | 0.08 | 0.10 | 0.10 | 0.10 | 0.10 | 0.11 | 0.11 | 0.13 | 0.13 | 0.13 |
| WebKB-wisconsin | 0.10 | 0.13 | 0.11 | 0.11 | 0.11 | 0.11 | 0.12 | 0.12 | 0.13 | 0.13 |
| WebKB-cornell | 0.06 | 0.07 | 0.08 | 0.08 | 0.08 | 0.09 | 0.09 | 0.10 | 0.10 | 0.11 |
| Imdb | 0.18 | 0.19 | 0.19 | 0.19 | 0.20 | 0.20 | 0.21 | 0.21 | 0.22 | 0.21 |
| news-2cl-1 | 0.17 | 0.20 | 0.22 | 0.22 | 0.23 | 0.23 | 0.23 | 0.23 | 0.23 | 0.23 |
| news-2cl-2 | 0.17 | 0.19 | 0.21 | 0.21 | 0.22 | 0.22 | 0.21 | 0.22 | 0.22 | 0.22 |
| news-2cl-3 | 0.20 | 0.23 | 0.25 | 0.24 | 0.25 | 0.25 | 0.25 | 0.25 | 0.25 | 0.25 |
| news-3cl-1 | 0.20 | 0.22 | 0.23 | 0.22 | 0.22 | 0.23 | 0.23 | 0.23 | 0.23 | 0.23 |
| news-3cl-2 | 0.15 | 0.19 | 0.19 | 0.19 | 0.20 | 0.20 | 0.20 | 0.20 | 0.20 | 0.20 |
| news-3cl-3 | 0.17 | 0.19 | 0.20 | 0.20 | 0.20 | 0.21 | 0.21 | 0.21 | 0.21 | 0.21 |
| news-5cl-1 | 0.15 | 0.18 | 0.18 | 0.18 | 0.19 | 0.19 | 0.19 | 0.19 | 0.19 | 0.19 |
| news-5cl-2 | 0.14 | 0.17 | 0.17 | 0.18 | 0.18 | 0.18 | 0.18 | 0.18 | 0.18 | 0.18 |
| news-5cl-3 | 0.12 | 0.14 | 0.14 | 0.14 | 0.15 | 0.15 | 0.15 | 0.15 | 0.15 | 0.15 |

**Tab. A.18.:** AUC provided by DWt using five social dimensions.

| *k* order: Dataset: | 2 | 3 | 5 | 10 | 20 | 40 | 50 |
|---|---|---|---|---|---|---|---|
| WebKB-texas | 0.13 | 0.13 | 0.13 | 0.12 | 0.11 | 0.09 | 0.09 |
| WebKB-washington | 0.07 | 0.07 | 0.07 | 0.06 | 0.06 | 0.05 | 0.05 |
| WebKB-wisconsin | 0.11 | 0.10 | 0.08 | 0.07 | 0.07 | 0.09 | 0.06 |
| WebKB-cornell | 0.11 | 0.11 | 0.09 | 0.07 | 0.06 | 0.06 | 0.07 |
| Imdb | 0.25 | 0.25 | 0.24 | 0.22 | 0.20 | 0.20 | 0.19 |
| news-2cl-1 | 0.17 | 0.16 | 0.15 | 0.12 | 0.10 | 0.10 | 0.10 |
| news-2cl-2 | 0.17 | 0.17 | 0.15 | 0.11 | 0.09 | 0.09 | 0.09 |
| news-2cl-3 | 0.22 | 0.22 | 0.20 | 0.17 | 0.16 | 0.15 | 0.14 |
| news-3cl-1 | 0.20 | 0.19 | 0.17 | 0.12 | 0.10 | 0.10 | 0.10 |
| news-3cl-2 | 0.16 | 0.15 | 0.13 | 0.08 | 0.08 | 0.08 | 0.08 |
| news-3cl-3 | 0.17 | 0.16 | 0.15 | 0.13 | 0.11 | 0.08 | 0.07 |
| news-5cl-1 | 0.14 | 0.14 | 0.13 | 0.12 | 0.09 | 0.08 | 0.08 |
| news-5cl-2 | 0.14 | 0.14 | 0.13 | 0.11 | 0.10 | 0.09 | 0.09 |
| news-5cl-3 | 0.12 | 0.12 | 0.11 | 0.10 | 0.09 | 0.06 | 0.05 |

**Tab. A.19.:** AUC provided by MFDW using five social dimensions.

| $\theta$ value: Dataset: | $10^{-6}$ | $10^{-5}$ | $10^{-4}$ | $10^{-3}$ | $10^{-2}$ | $10^{-1}$ | 1 | 10 |
|---|---|---|---|---|---|---|---|---|
| WebKB-texas | 0.08 | 0.08 | 0.08 | 0.08 | 0.13 | 0.16 | 0.13 | 0.02 |
| WebKB-washington | 0.07 | 0.07 | 0.07 | 0.05 | 0.06 | 0.10 | 0.09 | -0.02 |
| WebKB-wisconsin | 0.09 | 0.09 | 0.09 | 0.09 | 0.11 | 0.15 | 0.09 | 0.02 |
| WebKB-cornell | 0.07 | 0.07 | 0.07 | 0.07 | 0.08 | 0.12 | 0.10 | 0.00 |
| Imdb | 0.16 | 0.17 | 0.18 | 0.24 | 0.27 | 0.30 | 0.21 | 0.06 |
| news-2cl-1 | 0.12 | 0.12 | 0.15 | 0.18 | 0.21 | 0.20 | 0.06 | 0.07 |
| news-2cl-2 | 0.10 | 0.11 | 0.13 | 0.18 | 0.20 | 0.21 | 0.07 | 0.09 |
| news-2cl-3 | 0.15 | 0.16 | 0.20 | 0.21 | 0.25 | 0.25 | 0.09 | 0.07 |
| news-3cl-1 | 0.13 | 0.14 | 0.18 | 0.20 | 0.23 | 0.19 | 0.05 | 0.07 |
| news-3cl-2 | 0.11 | 0.11 | 0.15 | 0.16 | 0.19 | 0.20 | 0.05 | 0.07 |
| news-3cl-3 | 0.11 | 0.11 | 0.15 | 0.17 | 0.21 | 0.17 | 0.07 | 0.07 |
| news-5cl-1 | 0.09 | 0.10 | 0.15 | 0.15 | 0.19 | 0.14 | 0.03 | 0.05 |
| news-5cl-2 | 0.08 | 0.09 | 0.14 | 0.14 | 0.17 | 0.17 | 0.04 | 0.06 |
| news-5cl-3 | 0.07 | 0.08 | 0.12 | 0.12 | 0.15 | 0.16 | 0.04 | 0.05 |

**Tab. A.20.:** AUC provided by CovH using five social dimensions.

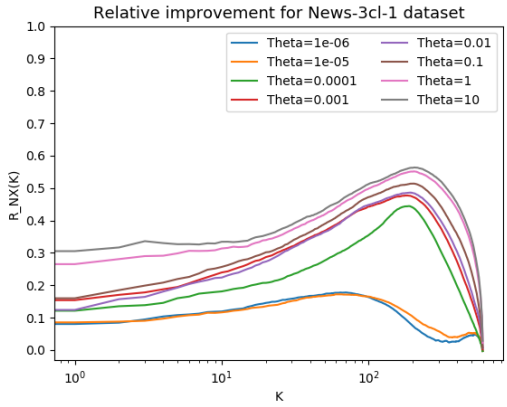
| $\theta$ value: Dataset: | $10^{-6}$ | $10^{-5}$ | $10^{-4}$ | $10^{-3}$ | $10^{-2}$ | $10^{-1}$ | 1 | 10 |
|---|---|---|---|---|---|---|---|---|
| WebKB-texas | 0.09 | 0.09 | 0.10 | 0.15 | 0.15 | 0.15 | 0.14 | 0.05 |
| WebKB-washington | 0.12 | 0.12 | 0.13 | 0.12 | 0.13 | 0.15 | 0.09 | 0.00 |
| WebKB-wisconsin | 0.10 | 0.10 | 0.12 | 0.15 | 0.15 | 0.17 | 0.19 | 0.08 |
| WebKB-cornell | 0.09 | 0.09 | 0.11 | 0.13 | 0.15 | 0.15 | 0.12 | 0.04 |
| Imdb | 0.26 | 0.27 | 0.26 | 0.28 | 0.29 | 0.31 | 0.28 | 0.07 |
| news-2cl-1 | 0.18 | 0.19 | 0.20 | 0.19 | 0.22 | 0.27 | 0.08 | 0.08 |
| news-2cl-2 | 0.19 | 0.19 | 0.19 | 0.19 | 0.23 | 0.27 | 0.10 | 0.10 |
| news-2cl-3 | 0.23 | 0.22 | 0.24 | 0.24 | 0.27 | 0.29 | 0.11 | 0.08 |
| news-3cl-1 | 0.20 | 0.21 | 0.21 | 0.22 | 0.25 | 0.25 | 0.07 | 0.07 |
| news-3cl-2 | 0.15 | 0.16 | 0.18 | 0.19 | 0.22 | 0.26 | 0.07 | 0.07 |
| news-3cl-3 | 0.17 | 0.18 | 0.19 | 0.19 | 0.23 | 0.23 | 0.08 | 0.08 |
| news-5cl-1 | 0.15 | 0.16 | 0.17 | 0.18 | 0.21 | 0.20 | 0.04 | 0.05 |
| news-5cl-2 | 0.15 | 0.17 | 0.16 | 0.16 | 0.20 | 0.23 | 0.05 | 0.06 |
| news-5cl-3 | 0.13 | 0.14 | 0.13 | 0.14 | 0.16 | 0.20 | 0.06 | 0.06 |

**Tab. A.21.:** AUC provided by NCorH using five social dimensions.

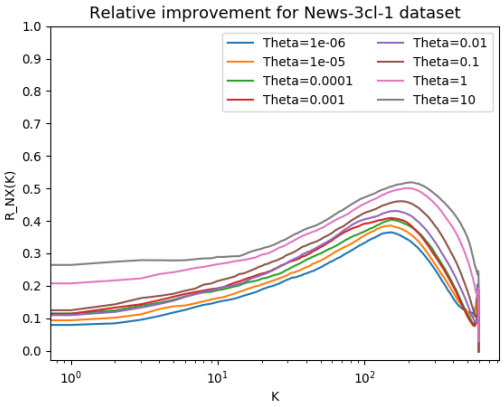
| $\theta$ value: Dataset: | $10^{-6}$ | $10^{-5}$ | $10^{-4}$ | $10^{-3}$ | $10^{-2}$ | $10^{-1}$ | 1 | 10 |
|---|---|---|---|---|---|---|---|---|
| WebKB-texas | 0.12 | 0.10 | 0.12 | 0.11 | 0.12 | 0.16 | 0.18 | 0.19 |
| WebKB-washington | 0.07 | 0.07 | 0.07 | 0.08 | 0.10 | 0.13 | 0.16 | 0.18 |
| WebKB-wisconsin | 0.16 | 0.17 | 0.17 | 0.17 | 0.20 | 0.24 | 0.27 | 0.30 |
| WebKB-cornell | 0.08 | 0.08 | 0.07 | 0.09 | 0.11 | 0.15 | 0.15 | 0.16 |
| imdb | 0.18 | 0.18 | 0.19 | 0.23 | 0.28 | 0.31 | 0.31 | 0.33 |
| news-2cl-1 | 0.18 | 0.21 | 0.28 | 0.38 | 0.38 | 0.37 | 0.38 | 0.40 |
| news-2cl-2 | 0.22 | 0.22 | 0.26 | 0.34 | 0.36 | 0.35 | 0.37 | 0.38 |
| news-2cl-3 | 0.22 | 0.23 | 0.27 | 0.37 | 0.41 | 0.38 | 0.40 | 0.41 |
| news-3cl-1 | 0.17 | 0.19 | 0.26 | 0.38 | 0.38 | 0.36 | 0.38 | 0.40 |
| news-3cl-2 | 0.17 | 0.17 | 0.25 | 0.35 | 0.36 | 0.34 | 0.38 | 0.38 |
| news-3cl-3 | 0.13 | 0.14 | 0.26 | 0.37 | 0.37 | 0.37 | 0.38 | 0.40 |
| news-5cl-1 | 0.13 | 0.13 | 0.20 | 0.36 | 0.37 | 0.34 | 0.36 | 0.38 |
| news-5cl-2 | 0.08 | 0.12 | 0.23 | 0.36 | 0.37 | 0.34 | 0.37 | 0.36 |
| news-5cl-3 | 0.09 | 0.11 | 0.17 | 0.34 | 0.31 | 0.31 | 0.32 | 0.34 |

**Tab. A.22.:** AUC provided by tSNE using five social dimensions.

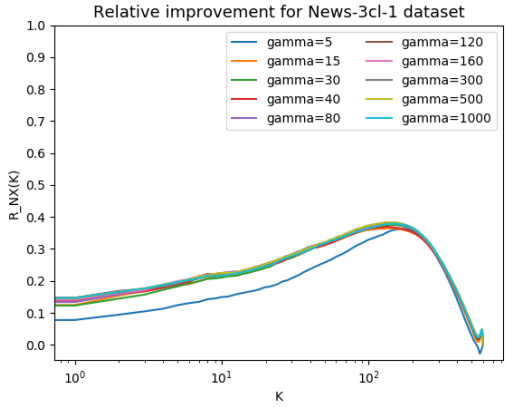
**Fig. A.1.:** $R_{NX}(K)$ criterion on News-3cl-1 dataset using five social dimensions. X-axis in log scale



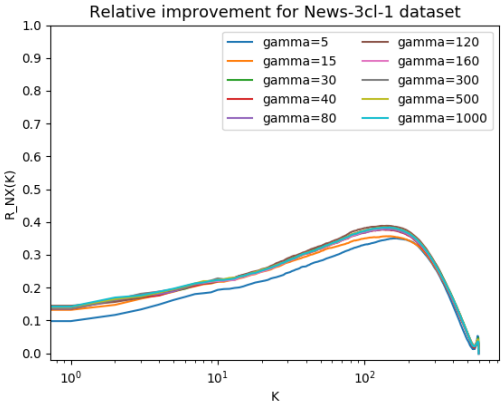

(a) BoPP-m



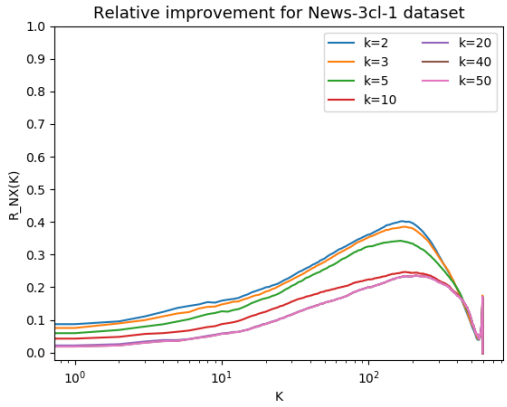

(b) BoPP-g



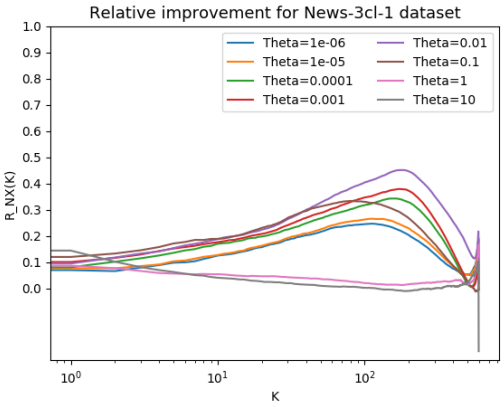

(c) DWg



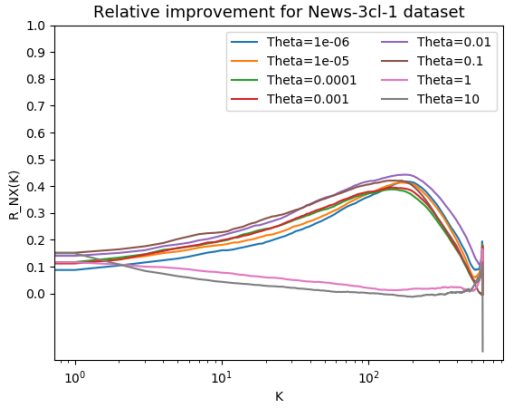

(d) DWt



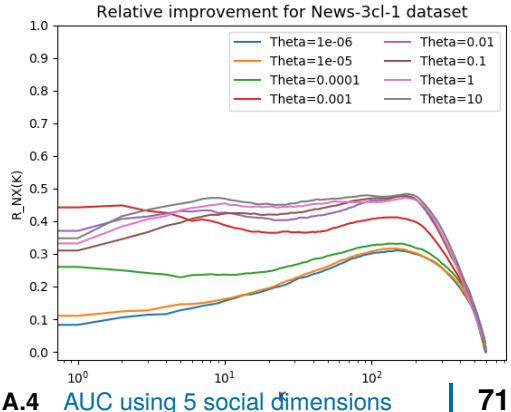

(e) MFDW




(f)CovH

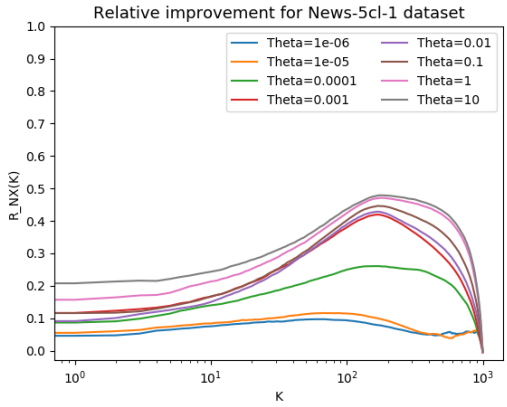



(g) NCorH

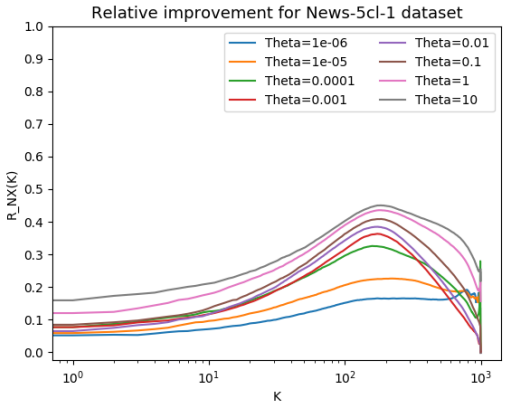



(h) tSNE

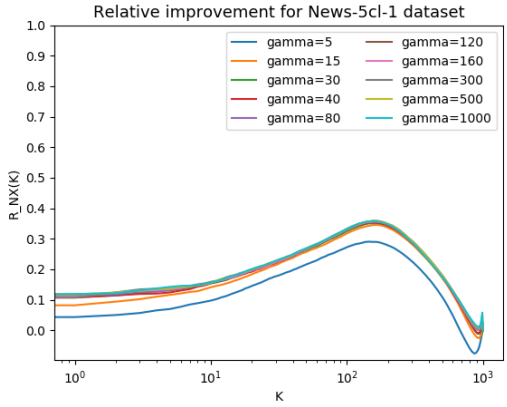
**Fig. A.2.:** $R_{NX}(K)$ criterion on News-5cl-1 dataset using five social dimensions. X-axis in log scale



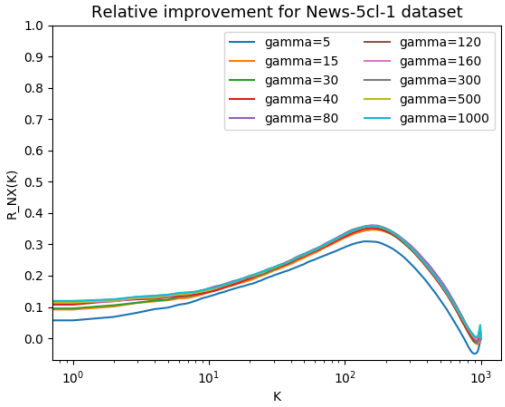

(a) BoPP-m



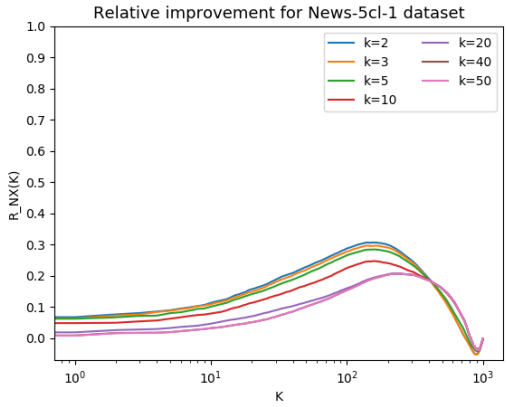

(b) BoPP-g



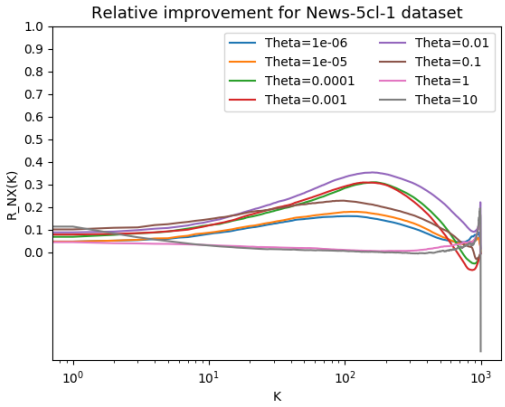

(c) DWg



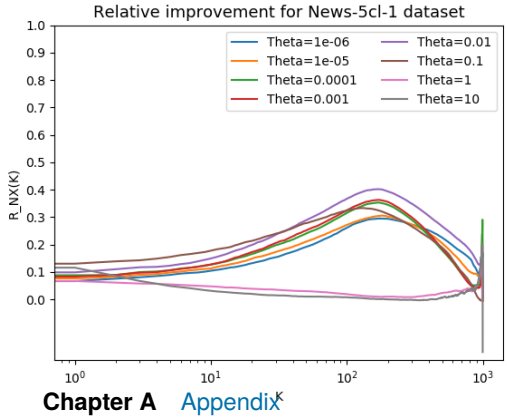

(d) DWt



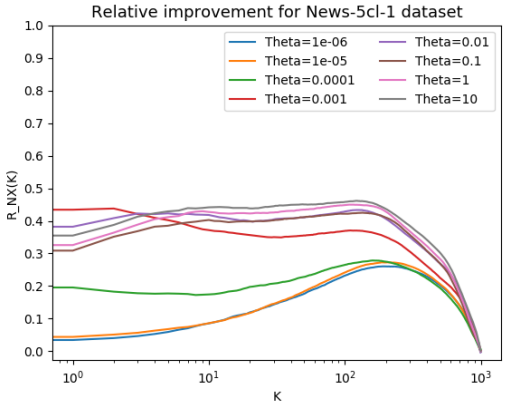

(e) MFDW




(f)CovH

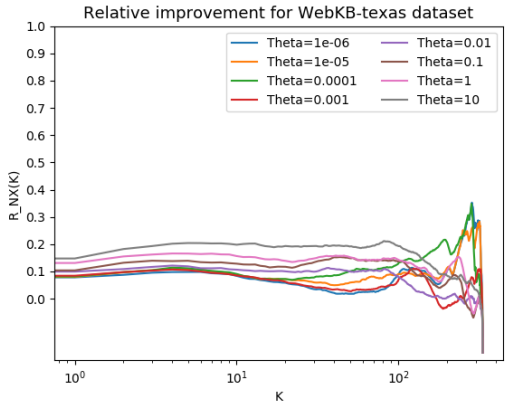



(g) NCorH

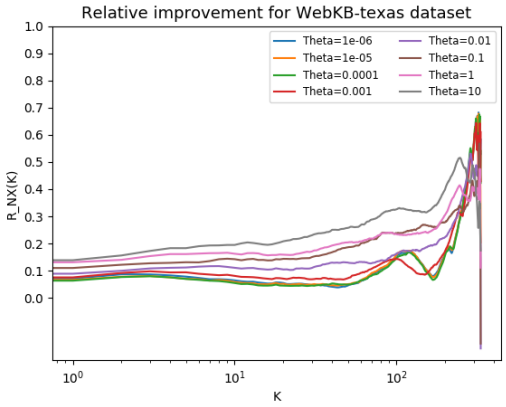



(h) tSNE

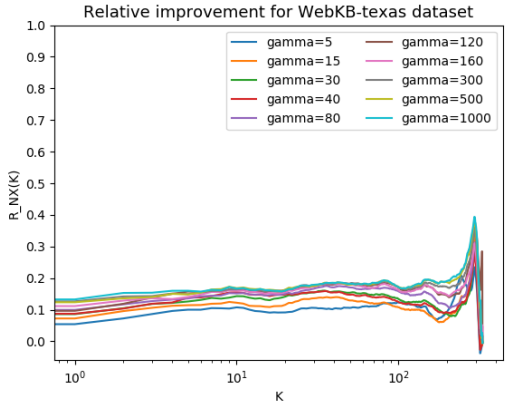
**Fig. A.3.:** $R_{NX}(K)$ criterion on WebKB-texas dataset using five social dimensions. X-axis in log scale



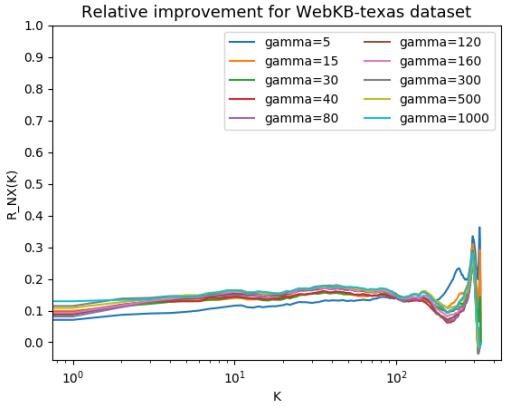

(a) BoPP-m



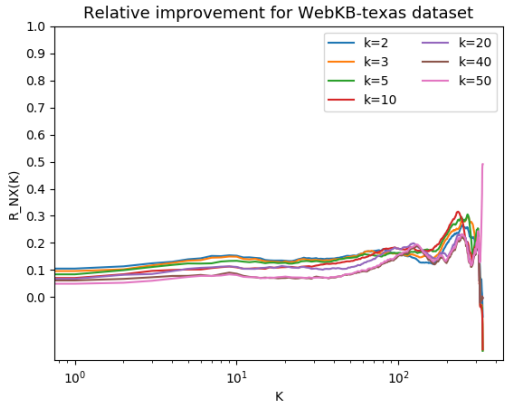

(b) BoPP-g



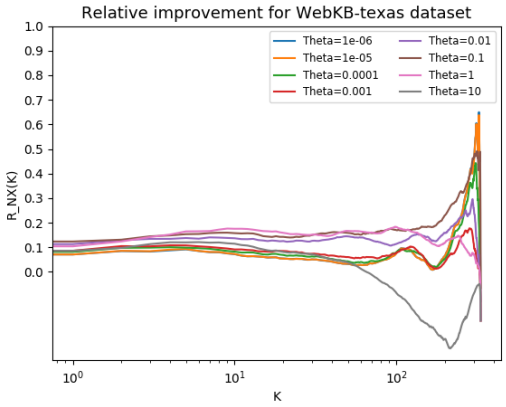

(c) DWg



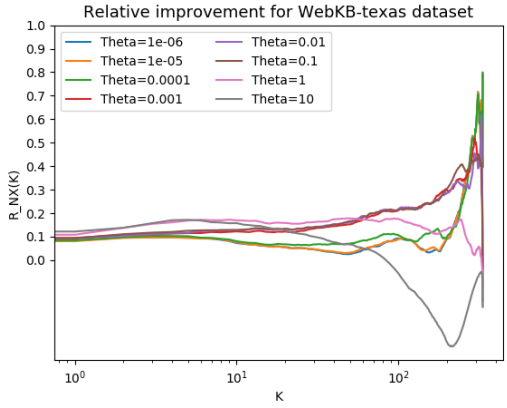

(d) DWt



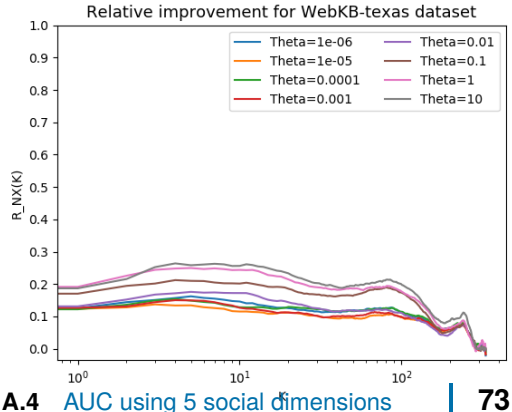

(e) MFDW




(f)CovH




(g) NCorH




(h) tSNE

## A.5 AUC using 2 social dimensions

| $\theta$ value: Dataset: | $10^{-6}$ | $10^{-5}$ | $10^{-4}$ | $10^{-3}$ | $10^{-2}$ | $10^{-1}$ | 1 | 10 |
|---|---|---|---|---|---|---|---|---|
| WebKB-texas | 0.06 | 0.06 | 0.05 | 0.06 | 0.04 | 0.08 | 0.08 | 0.08 |
| WebKB-washington | 0.09 | 0.09 | 0.06 | 0.05 | 0.06 | 0.08 | 0.07 | 0.06 |
| WebKB-wisconsin | 0.06 | 0.06 | 0.08 | 0.11 | 0.06 | 0.11 | 0.11 | 0.12 |
| WebKB-cornell | 0.07 | 0.07 | 0.05 | 0.04 | 0.04 | 0.06 | 0.06 | 0.06 |
| imdb | 0.13 | 0.13 | 0.13 | 0.13 | 0.16 | 0.19 | 0.18 | 0.15 |
| news-2cl-1 | 0.05 | 0.05 | 0.05 | 0.14 | 0.14 | 0.14 | 0.17 | 0.18 |
| news-2cl-2 | 0.05 | 0.05 | 0.10 | 0.13 | 0.13 | 0.13 | 0.15 | 0.15 |
| news-2cl-3 | 0.08 | 0.08 | 0.12 | 0.17 | 0.18 | 0.19 | 0.19 | 0.21 |
| news-3cl-1 | 0.02 | 0.05 | 0.10 | 0.15 | 0.15 | 0.16 | 0.17 | 0.17 |
| news-3cl-2 | 0.03 | 0.03 | 0.06 | 0.11 | 0.11 | 0.11 | 0.13 | 0.14 |
| news-3cl-3 | 0.03 | 0.04 | 0.05 | 0.14 | 0.14 | 0.14 | 0.15 | 0.16 |
| news-5cl-1 | 0.04 | 0.04 | 0.03 | 0.13 | 0.13 | 0.13 | 0.14 | 0.15 |
| news-5cl-2 | 0.03 | 0.03 | 0.10 | 0.11 | 0.12 | 0.13 | 0.13 | 0.14 |
| news-5cl-3 | 0.02 | 0.03 | 0.06 | 0.09 | 0.09 | 0.09 | 0.10 | 0.11 |

**Tab. A.23.:** AUC provided by BoPP-m using two social dimensions.

| $\theta$ value: Dataset: | $10^{-6}$ | $10^{-5}$ | $10^{-4}$ | $10^{-3}$ | $10^{-2}$ | $10^{-1}$ | 1 | 10 |
|---|---|---|---|---|---|---|---|---|
| WebKB-texas | 0.07 | 0.07 | 0.07 | 0.07 | 0.09 | 0.08 | 0.08 | 0.09 |
| WebKB-washington | 0.09 | 0.09 | 0.10 | 0.09 | 0.09 | 0.08 | 0.06 | 0.06 |
| WebKB-wisconsin | 0.07 | 0.07 | 0.07 | 0.09 | 0.08 | 0.10 | 0.09 | 0.09 |
| WebKB-cornell | 0.07 | 0.07 | 0.06 | 0.06 | 0.08 | 0.06 | 0.05 | 0.06 |
| imdb | 0.10 | 0.10 | 0.10 | 0.18 | 0.17 | 0.17 | 0.15 | 0.15 |
| news-2cl-1 | 0.09 | 0.10 | 0.11 | 0.11 | 0.11 | 0.11 | 0.12 | 0.12 |
| news-2cl-2 | 0.10 | 0.09 | 0.09 | 0.09 | 0.09 | 0.10 | 0.12 | 0.11 |
| news-2cl-3 | 0.12 | 0.12 | 0.12 | 0.12 | 0.13 | 0.12 | 0.14 | 0.15 |
| news-3cl-1 | 0.08 | 0.09 | 0.12 | 0.13 | 0.13 | 0.12 | 0.13 | 0.13 |
| news-3cl-2 | 0.08 | 0.08 | 0.08 | 0.08 | 0.08 | 0.09 | 0.10 | 0.10 |
| news-3cl-3 | 0.06 | 0.09 | 0.09 | 0.09 | 0.09 | 0.11 | 0.11 | 0.12 |
| news-5cl-1 | 0.02 | 0.07 | 0.08 | 0.08 | 0.08 | 0.08 | 0.09 | 0.10 |
| news-5cl-2 | 0.02 | 0.07 | 0.08 | 0.09 | 0.09 | 0.09 | 0.10 | 0.10 |
| news-5cl-3 | 0.02 | 0.06 | 0.06 | 0.06 | 0.06 | 0.07 | 0.08 | 0.09 |

**Tab. A.24.:** AUC provided by BoPP-g using two social dimensions.

| $k$ order: Dataset: | 2 | 3 | 5 | 10 | 20 | 40 | 50 |
|---|---|---|---|---|---|---|---|
| WebKB-texas | 0.08 | 0.08 | 0.08 | 0.08 | 0.08 | 0.08 | 0.08 |
| WebKB-washington | 0.04 | 0.04 | 0.04 | 0.04 | 0.04 | 0.05 | 0.05 |
| WebKB-wisconsin | 0.04 | 0.05 | 0.06 | 0.06 | 0.06 | 0.11 | 0.05 |
| WebKB-cornell | 0.06 | 0.06 | 0.06 | 0.06 | 0.06 | 0.06 | 0.07 |
| imdb | 0.16 | 0.16 | 0.17 | 0.17 | 0.17 | 0.17 | 0.17 |
| news-2cl-1 | 0.10 | 0.10 | 0.10 | 0.09 | 0.09 | 0.09 | 0.09 |
| news-2cl-2 | 0.09 | 0.09 | 0.09 | 0.09 | 0.09 | 0.09 | 0.09 |
| news-2cl-3 | 0.15 | 0.14 | 0.14 | 0.14 | 0.14 | 0.14 | 0.14 |
| news-3cl-1 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 |
| news-3cl-2 | 0.08 | 0.08 | 0.08 | 0.08 | 0.08 | 0.08 | 0.08 |
| news-3cl-3 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 |
| news-5cl-1 | 0.08 | 0.08 | 0.07 | 0.07 | 0.07 | 0.08 | 0.08 |
| news-5cl-2 | 0.09 | 0.09 | 0.09 | 0.09 | 0.09 | 0.09 | 0.09 |
| news-5cl-3 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 |

**Tab. A.25.:** AUC provided by MFDW using two social dimensions.

| $\theta$ value: Dataset: | $10^{-6}$ | $10^{-5}$ | $10^{-4}$ | $10^{-3}$ | $10^{-2}$ | $10^{-1}$ | 1 | 10 |
|---|---|---|---|---|---|---|---|---|
| WebKB-texas | 0.05 | 0.05 | 0.05 | 0.04 | 0.09 | 0.09 | 0.06 | 0.02 |
| WebKB-washington | 0.05 | 0.05 | 0.05 | 0.03 | 0.04 | 0.05 | 0.06 | -0.01 |
| WebKB-wisconsin | 0.04 | 0.04 | 0.04 | 0.04 | 0.03 | 0.06 | 0.06 | 0.00 |
| WebKB-cornell | 0.04 | 0.04 | 0.04 | 0.03 | 0.05 | 0.06 | 0.06 | 0.01 |
| imdb | 0.07 | 0.07 | 0.07 | 0.14 | 0.17 | 0.18 | 0.14 | 0.04 |
| news-2cl-1 | 0.01 | 0.01 | 0.10 | 0.10 | 0.11 | 0.08 | 0.03 | 0.06 |
| news-2cl-2 | 0.01 | 0.01 | 0.02 | 0.09 | 0.10 | 0.07 | 0.04 | 0.07 |
| news-2cl-3 | 0.03 | 0.04 | 0.11 | 0.13 | 0.12 | 0.16 | 0.04 | 0.05 |
| news-3cl-1 | 0.02 | 0.02 | 0.08 | 0.13 | 0.12 | 0.12 | 0.02 | 0.06 |
| news-3cl-2 | 0.02 | 0.01 | 0.08 | 0.08 | 0.09 | 0.09 | 0.03 | 0.07 |
| news-3cl-3 | 0.02 | 0.02 | 0.09 | 0.10 | 0.06 | 0.07 | 0.03 | 0.06 |
| news-5cl-1 | 0.02 | 0.02 | 0.08 | 0.08 | 0.09 | 0.06 | 0.02 | 0.05 |
| news-5cl-2 | 0.01 | 0.01 | 0.09 | 0.10 | 0.11 | 0.10 | 0.02 | 0.05 |
| news-5cl-3 | 0.01 | 0.01 | 0.05 | 0.05 | 0.05 | 0.04 | 0.03 | 0.05 |

**Tab. A.26.:** AUC provided by CovH using two social dimensions.

| $\theta$ value: Dataset: | $10^{-6}$ | $10^{-5}$ | $10^{-4}$ | $10^{-3}$ | $10^{-2}$ | $10^{-1}$ | 1 | 10 |
|---|---|---|---|---|---|---|---|---|
| WebKB-texas | 0.08 | 0.08 | 0.07 | 0.10 | 0.08 | 0.10 | 0.06 | 0.03 |
| WebKB-washington | 0.08 | 0.08 | 0.08 | 0.09 | 0.08 | 0.09 | 0.06 | -0.01 |
| WebKB-wisconsin | 0.06 | 0.06 | 0.06 | 0.09 | 0.05 | 0.06 | 0.08 | 0.02 |
| WebKB-cornell | 0.08 | 0.08 | 0.10 | 0.07 | 0.06 | 0.06 | 0.08 | 0.03 |
| imdb | 0.15 | 0.15 | 0.19 | 0.17 | 0.17 | 0.18 | 0.16 | 0.04 |
| news-2cl-1 | 0.06 | 0.11 | 0.11 | 0.11 | 0.12 | 0.10 | 0.04 | 0.06 |
| news-2cl-2 | 0.10 | 0.10 | 0.09 | 0.09 | 0.10 | 0.10 | 0.05 | 0.07 |
| news-2cl-3 | 0.12 | 0.12 | 0.12 | 0.12 | 0.13 | 0.18 | 0.05 | 0.05 |
| news-3cl-1 | 0.12 | 0.12 | 0.13 | 0.13 | 0.13 | 0.14 | 0.03 | 0.07 |
| news-3cl-2 | 0.08 | 0.08 | 0.08 | 0.08 | 0.10 | 0.12 | 0.05 | 0.07 |
| news-3cl-3 | 0.09 | 0.10 | 0.09 | 0.09 | 0.11 | 0.10 | 0.04 | 0.06 |
| news-5cl-1 | 0.05 | 0.06 | 0.07 | 0.08 | 0.10 | 0.08 | 0.03 | 0.05 |
| news-5cl-2 | 0.09 | 0.09 | 0.09 | 0.10 | 0.11 | 0.12 | 0.03 | 0.05 |
| news-5cl-3 | 0.06 | 0.06 | 0.06 | 0.06 | 0.06 | 0.07 | 0.03 | 0.05 |

**Tab. A.27.:** AUC provided by NCorH using two social dimensions.

| $\theta$ value: Dataset: | $10^{-6}$ | $10^{-5}$ | $10^{-4}$ | $10^{-3}$ | $10^{-2}$ | $10^{-1}$ | 1 | 10 |
|---|---|---|---|---|---|---|---|---|
| WebKB-texas | 0.08 | 0.08 | 0.08 | 0.08 | 0.11 | 0.13 | 0.13 | 0.14 |
| WebKB-washington | 0.06 | 0.05 | 0.05 | 0.06 | 0.08 | 0.10 | 0.11 | 0.11 |
| WebKB-wisconsin | 0.13 | 0.12 | 0.12 | 0.14 | 0.15 | 0.17 | 0.20 | 0.21 |
| WebKB-cornell | 0.05 | 0.06 | 0.05 | 0.07 | 0.08 | 0.11 | 0.11 | 0.11 |
| imdb | 0.17 | 0.19 | 0.18 | 0.18 | 0.22 | 0.24 | 0.23 | 0.23 |
| news-2cl-1 | 0.13 | 0.14 | 0.18 | 0.22 | 0.24 | 0.24 | 0.26 | 0.27 |
| news-2cl-2 | 0.11 | 0.05 | 0.18 | 0.23 | 0.20 | 0.23 | 0.23 | 0.24 |
| news-2cl-3 | 0.16 | 0.16 | 0.20 | 0.25 | 0.27 | 0.27 | 0.30 | 0.31 |
| news-3cl-1 | 0.12 | 0.10 | 0.18 | 0.23 | 0.25 | 0.26 | 0.26 | 0.27 |
| news-3cl-2 | 0.09 | 0.11 | 0.17 | 0.20 | 0.23 | 0.22 | 0.23 | 0.24 |
| news-3cl-3 | 0.09 | 0.10 | 0.17 | 0.22 | 0.23 | 0.23 | 0.24 | 0.26 |
| news-5cl-1 | 0.05 | 0.05 | 0.15 | 0.20 | 0.23 | 0.23 | 0.23 | 0.23 |
| news-5cl-2 | 0.03 | 0.06 | 0.14 | 0.21 | 0.20 | 0.21 | 0.22 | 0.23 |
| news-5cl-3 | 0.05 | 0.06 | 0.15 | 0.18 | 0.17 | 0.18 | 0.18 | 0.19 |

**Tab. A.28.:** AUC provided by tSNE using two social dimensions.

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