PARFAITE: PageRank-Matrix Factorization for Interpretable Graph Embeddings*

Gabriel Damay[0009−0008−9337−6387] and Mauro Sozio[0000−0002−2031−3974]

Institut Polytechnique de Paris, Télécom Paris, Palaiseau, France
{gabriel.damay,sozio}@telecom-paris.fr

Abstract. There have been increasing efforts in recent years to develop so-called graph embeddings, which among other things allow to employ standard machine learning techniques to solve urgent real-world problems. However, developing *interpretable* graph embeddings has received much less attention. In our work, we develop PARFAITE, an algorithm for finding an interpretable and effective graph embedding, based on the factorization of the PageRank matrix of the input graph. We evaluate the interpretability of our method against popular graph embedding techniques, such as node2vec, showing that PARFAITE boasts significantly higher interpretability scores. Another contribution of our work is the release of a novel dataset constructed from all pages of the French version of Wikipedia, which we release for reproducibility and benchmarking.

Keywords: network embedding · interpretability · personalized PageRank · graph mining

1 Introduction

Graph embeddings consist of low-dimensional vector representations reflecting the relationships between vertices as well as some other properties of the underlying graph. They allow to employ standard machine learning techniques to perform vertex classification, link prediction and many other tasks. To compute such embeddings, one often aims at optimising some objective functions reflecting proximity relationships between the vertices or edges of the input graph.

There have been significant efforts in recent years to compute effective graph embeddings. They can be classified in three main categories [5]. The first category regroups the approaches based on matrix factorization, such as [1] and the more recent approach HOPE [14]. The second category regroups the approaches based on sampling random walks from the input graph, with the most prominent work being perhaps node2vec [6]. Finally, the last category regroups Graph Neural Network-based approaches, with GCN [10] being one of the best-known approaches.

In our work, we focus on matrix-factorization based approaches which typically consist of factorizing a matrix representing some proximity relationships

* This work was partially supported by the French National Agency (ANR) under project APY (ANR-20-CE38-0011)

between the vertices of the input graph. In particular, approaches based on the Personalized PageRank (PPR) matrix, which we focus on in our work, have been shown to provide very good results in a wide range of tasks, such as graph reconstruction and nodes classification [20, 23], link prediction [20, 21, 23], as well as nodes recommendation [21]. There are other methods based on the factorization of the PPR matrix, such as APP [23], STRAP [21], and NRP [20]. HOPE [14] is considered to be one of the state-of-the-art approaches based on matrix factorization. One of its variants also focuses on the SVD of the PPR matrix. However, most of the analysis and the experiments in [14] focus on the Katz proximity matrix.

The interest for explainable algorithms is growing recently among the broad research community and in the general public alike. Many data-processing algorithms are fed with embeddings of complex data. If the embedding methods are not interpretable, there is little hope to provide satisfying explanations of the result of the algorithm. As a result, many recent papers focus on developing embedding methods that are interpretable, such as [13] and [18], for image and video processing, respectively. In the field of graph embeddings, most of the efforts have focused on defining measures to assess the interpretability of a graph embedding algorithm, with community-based metrics emerging as one of the most popular metrics. In particular, [9] and [4] develop three different community-based interpretability metrics and evaluate node2vec and HOPE in terms of those metrics. Those works suggest that a satisfactory solution for an interpretable graph embedding is still missing. Our work aims at filling this gap, focusing on developing an effective and interpretable graph embedding.

We develop Parfaite, a novel approach based on the SVD of the PPR matrix. Our experimental evaluation against the state-of-the-art shows that our method provides higher interpretability scores, while boasting similar results in link prediction. To improve the interpretability of our method, we depart from the related work in a number of ways. In particular, in contrast with previous work, our method focuses on the *centered* PPR matrix. It seems natural to center the PPR matrix, in that, the uncentered matrix is positive and therefore its first singular vectors are also positive. Moreover, an embedding based on such a positive matrix would use only half of the available space. Observe that centering is also performed in principal component analysis (PCA). As we show in Section 3.2, centering helps us both reducing biases in the PPR matrix and to interpret the left part of the decomposition of the SVD. We also depart from the related work in the way our embedding is derived from the singular values of the SVD. In particular, virtually all methods based on the SVD of the PPR matrix, use the square root of the singular values to build the vertex embeddings $\boldsymbol{U}\boldsymbol{\Sigma}^{1/2}$ and $\boldsymbol{V}\boldsymbol{\Sigma}^{1/2}$. However, $\boldsymbol{U}\boldsymbol{\Sigma}$ and $\boldsymbol{V}\boldsymbol{\Sigma}$ contain some relevant information because they represent the lines and columns of the original matrix projected onto the eigenspaces. Such information is not leveraged in previous work, to the best of our knowledge. Another contribution of our work is a new metric measuring the interpretability of a graph embedding addressing some of the limitations of previous metrics. Finally, we release a novel dataset constructed from all pages

Table 1. Notation

Symbol	Meaning	Definition
$\boldsymbol{A}$	Adjacency matrix of the graph	
$\boldsymbol{D}$	Diagonal matrix of out-degrees	
$\boldsymbol{P}$	Stochastic matrix of random walk in the graph	$D^{-1}A$
$\boldsymbol{U}$, $\boldsymbol{V}$, $\boldsymbol{\Sigma}$	Result matrices of the SVD, $\boldsymbol{U}$ and $\boldsymbol{V}$ are unit matrices, and $\boldsymbol{\Sigma}$ is diagonal	
$\boldsymbol{\Pi}$	PPR matrix, each line is the PPR vector of a node	$\alpha \sum_{i=0}^{+\infty} (1-\alpha)^i \boldsymbol{P}^i$
$\boldsymbol{\Pi_l}$	PPR matrix, approximated using only $l+1$ iterations	$\alpha \sum_{i=0}^{l} (1-\alpha)^i \boldsymbol{P}^i$
G	Number of known ground-truth communities	
K	Number of dimensions of an embedding	
D	Number of dimension of the SVD	

of the French version of Wikipedia, which we release for reproducibility and benchmarking.

The rest of our work is organized as follows. In section 2 we present the main preliminary concepts needed for our work. In section 3, we present our approach, first through an overview and an explanation of its interpretation and then through the technical details of its use and implementation. In section 5 we conduct an experimental evaluation showing that our method boasts higher interpretability scores than node2vec. Finally, section 6 summarizes our work and discussed interesting directions for future work.

2 Preliminaries

2.1 Notations

Vectors and matrices are represented respectively by bold lower- and uppercase letters. Indices and exponents after a vector or matrix name are part of the name if they are bold, but are respectively entries in the object or power operator if they are not bold. Each time the cardinality of a set is noted with an uppercase letter (e.g. K is the number of dimensions of an embedding), we use the corresponding lowercase letter as a variable for the index of an element of the set (e.g. k is the index of a dimension of the embedding).

2.2 Personalized PageRank (PPR)

The Personalized PageRank (PPR) score is an extension of the PageRank score introduced in [15] in which the starting vertices are in a defined subset. We call *Personalized PageRank of a node* u the PPR score in which the starting subset contains only the node u, and we note $\boldsymbol{\pi_u}$ the vector containing the scores.

Let $\boldsymbol{A}$ be the adjacency matrix of a graph (directed or not) and $\boldsymbol{D}$ the diagonal matrix of its out-degrees. The matrix $\boldsymbol{P} = \boldsymbol{D}^{-1}\boldsymbol{A}$ is the stochastic matrix of a random walk in the graph.

Let α be the probability at each step that our random walk terminates given that it didn't terminate before, and X_0 and X_f the vertex on which the walker is respectively before the first step and after the last step.

$$\boldsymbol{\pi}_{\boldsymbol{u}v} = \mathbb{P}\left(X_f = v | X_0 = u\right) \tag{1}$$

$$\boldsymbol{\pi}_{\boldsymbol{u}}{}^{\top} = \alpha \sum_{i=0}^{\infty} (1-\alpha)^i \boldsymbol{e}_{\boldsymbol{u}}{}^{\top} \boldsymbol{P}^i \tag{2}$$

where $\boldsymbol{e}_{\boldsymbol{u}}$ is the u^{th} vector from the canonical basis of $\mathbb{R}^n$, i.e a vector of which u^{th} entry is 1 and all other entries are 0.

We call *PPR matrix of the graph*, and we note $\boldsymbol{\Pi}$ the matrix of which each line is the PPR vector of the related node. We can see from (2) that

$$\boldsymbol{\Pi} = \alpha \sum_{i=0}^{\infty} (1-\alpha)^i \boldsymbol{P}^i \tag{3}$$

We call *Reversed PPR vector of a node* the vector $\boldsymbol{\gamma}_{\boldsymbol{u}}$ that contains, for each dimension v, the PPR score when the node v is the source node and the node u is the target node, i.e. $\boldsymbol{\gamma}_{\boldsymbol{u}v} = \boldsymbol{\pi}_{\boldsymbol{v}u}$.

2.3 SVD

The Singular Values Decomposition (SVD) is a well-known method of matrix factorization. Each matrix $\boldsymbol{M} \in \mathbb{R}^{m \times n}$ is factorized into two unit matrices $\boldsymbol{U} \in \mathbb{R}^{m \times m}$ and $\boldsymbol{V} \in \mathbb{R}^{n \times n}$, and a diagonal matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{m \times n}$ so that $\boldsymbol{M} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\top}$. The *truncated SVD* of $\boldsymbol{M}$, approximates $\boldsymbol{M}$ with the matrices $\boldsymbol{\Sigma}_{\boldsymbol{d}} \in \mathbb{R}^{d \times d}$, $\boldsymbol{U}_{\boldsymbol{d}} \in \mathbb{R}^{m \times d}$ and $\boldsymbol{V}_{\boldsymbol{d}} \in \mathbb{R}^{n \times d}$ that are the matrices made respectively of the d highest values of $\boldsymbol{\Sigma}$ and the related columns of $\boldsymbol{U}$ and $\boldsymbol{V}$. The acronym SVD is usually used as a metonymy for the Truncated SVD and we will do so in the rest of this paper.

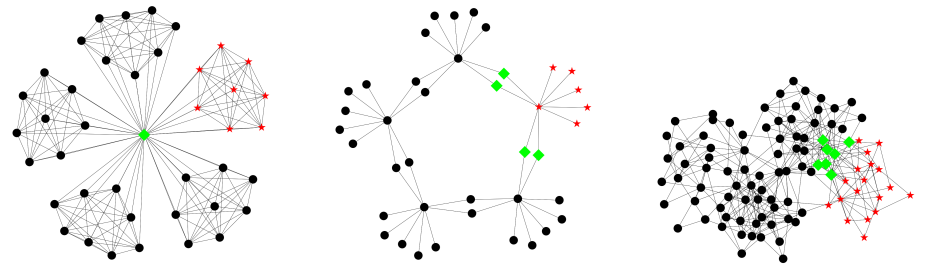
One of the main advantages of the SVD, that make it especially attractive for embedding, is that it is known to filter out the noise in the data. More specifically, the SVD removes the small variations between data so that only the main patterns in the matrix are kept in the final result [12].

3 Our approaches

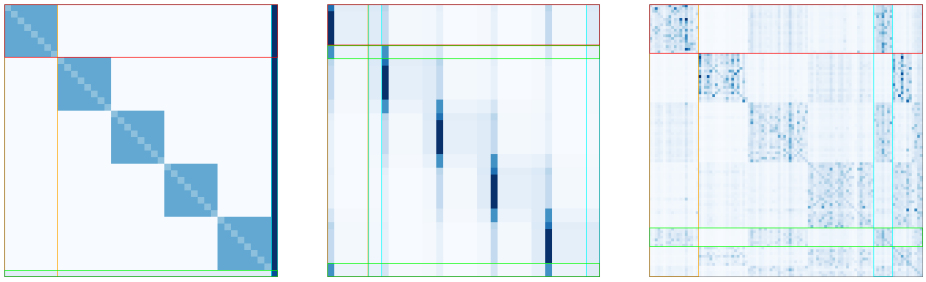
3.1 Overview

We introduce the new PAgeRank FActorization-based InTerpretable Embedding (PARFAITE) method. It produces two embeddings, PARFAITE_L and PARFAITE_R .

Our method consists of three main parts. First, a truncated Singular Values Decomposition is performed. To overcome the issues of representing the PPR matrix exactly for very large graphs, we employ a function m that, given any



(a) Star of cliques (b) Ring of stars (c) Toy Graph from SBM



(d) $\boldsymbol{\Pi} - \alpha \boldsymbol{I}$ matrix of the star of cliques (e) $\boldsymbol{\Pi} - \alpha \boldsymbol{I}$ matrix of the ring of stars (f) $\boldsymbol{\Pi} - \alpha \boldsymbol{I}$ matrix of the toy graph from SBM

Fig. 1. Toy graphs and their respective $\boldsymbol{\Pi} - \alpha \boldsymbol{I}$ matrices. On each graph, a community is highlighted, the red stars vertices belonging exclusively to the community, and the green diamonds ones belonging both to the highlighted community and to at least another one. On each matrix, the rows and columns relative to the red star vertices are highlighted through red and orange boxes, and the rows and columns relative to the green diamond vertices are highlighted through green and cyan boxes. We use $\boldsymbol{\Pi} - \alpha \boldsymbol{I}$ instead of $\boldsymbol{\Pi}$ for better readability.

vector $\boldsymbol{v}$, approximates the multiplication of the matrix with the vector. This results in two representations of the vertices as $\boldsymbol{U\Sigma}$ and $\boldsymbol{V\Sigma}$. Then, vertices are clustered, with each vertex being represented by a concatenation of its left and right representation. This provides the central points of the communities in the space of these representations, stored in the matrix $\boldsymbol{C}$. Finally, our left and right embeddings, respectively PARFAITE_L and PARFAITE_R are computed by projecting back these central points onto the original spaces. A pseudocode of our algorithm is shown in Algorithm 1.

3.2 Interpretation of the steps

The PARFAITE method relies on the well-established fact that the PPR vector of each vertex often contains large scores at dimensions corresponding to vertices it shares at least one community with, while it contains small scores at other dimensions [8, 11].

Algorithm 1 PARFAITE

procedure PARFAITE($\boldsymbol{P}$: stochastic random walk matrix, m_P: function that approximate $\bar{\boldsymbol{\Pi}}\boldsymbol{v}$ for any $\boldsymbol{v}$)
  $\boldsymbol{U}$, $\boldsymbol{\Sigma}$, $\boldsymbol{V} \leftarrow$ SVD(m_P)
  clustering $\leftarrow$ kmeans(concat(normalize($\boldsymbol{U\Sigma}$), normalize($\boldsymbol{V\Sigma}$)))
  $\boldsymbol{C} \leftarrow$ clustering.clusters_centers
  PARFAITE_L $\leftarrow \boldsymbol{U}\boldsymbol{C}^{\top}_{:,d:}$
  PARFAITE_R $\leftarrow \boldsymbol{V}\boldsymbol{C}^{\top}_{:,:d}$
  return PARFAITE_L , PARFAITE_R
end procedure

This property strengthens the one stated by [22] that two nodes are likely to share a community not only if the PPR score from one node to the other is high, but especially if their PPR vectors are similar.

We illustrate this property with three toy graphs represented in Figure 1. The first two graphs provide ideal cases where we can find communities according to two natural community definitions: a) a clique and b) a star where all nodes but one are connected to a single node (e.g. they are connected to a same influencer in social media). In particular, Figure 1a represents a star of cliques, while Figure 1b represents a ring of stars. Observe that the communities in the graphs of Figure 1 do *overlap*, with red star vertices belonging exclusively to one community, while green diamond vertices belonging to multiple communities. The graph in Figure 1c is built using the Stochastic Block Model (SBM) with 100 vertices, 300 edges, 4 communities, with each pair of vertices of a same community being 100 times more likely to be connected than two vertices from different communities. As a result, in Figure 1c, there are 19 vertices belonging to exactly two communities, 7 of which are depicted in green.

Figures 1d, 1e and 1f show the corresponding PPR matrices. Recall that each row corresponds to some vertex v and it represents the PPR vector of v, that is when v is the source vertex of the random walk. Indeed, we can see that PPR vectors contain larger scores to vertices of a same community as the source node, generating "square" patterns in the matrices.

Similarly, we observe that the reversed PPR vectors (columns in the matrix) exhibit the same pattern of large scores inside the community and low scores outside.

However, there might be relatively few vertices that have very large PPR scores even if they do not share communities with other nodes. This is apparent in Figure 1e, where we observe that the central nodes in the neighboring cliques have higher scores in the PPR vector of the studied community than other nodes in the same community. The reversed PPR is not affected by that issue.

To avoid this biais, we define the $\bar{\boldsymbol{\Pi}}$ matrix obtained by centering the columns of $\boldsymbol{\Pi}$.

Property 1.

$$\bar{\boldsymbol{\Pi}}_{i,j} = \frac{1}{n}\mathbb{P}\left(X_f = j\right)\left(\mathbb{P}\left(X_0 = i | X_f = j\right) - \mathbb{P}\left(X_0 = i\right)\right)$$

Proof.

$$\begin{aligned}\bar{\boldsymbol{\Pi}}_{i,j} &= \mathbb{P}\left(X_f = j | X_0 = i\right) - \mathbb{P}\left(X_f = j\right) \\ &= \frac{\mathbb{P}\left(X_0 = i | X_f = j\right)\mathbb{P}\left(X_f = j\right)}{\mathbb{P}\left(X_0 = i\right)} - \mathbb{P}\left(X_f = j\right)\end{aligned}$$

If we look at the rows of this new matrix, each entry is then the excess of probability to go to each vertex from the reference vertex, compared to the agnostic probability. If we look at the columns and because $\mathbb{P}\left(X_f = j\right)$ is a constant along a column, each entry is proportional to the excess of probability to come from each vertex given that the walk arrived at the reference vertex.

Because the SVD only keeps the main patterns of the matrix it decomposes, we expect the result of the SVD to exhibit the typical rows and columns for each community. We could then interpret these vectors as respectively the belonging of each node to the community and its relevance wrt. the community.

Our last problem is that, although the truncated SVD should make the community patterns in the matrix apparent, each dimension of the decomposition does usually not match a community, hindering the interpretation. To tackle this issue we perform a clustering on the vertices represented by the SVD, and we use the central points to obtain the desired representative vectors for each community.

3.3 Decomposition of the PPR matrix

The PPR matrix is a dense matrix belonging to $\mathbb{R}^{n\times n}$. In most cases of big graphs, this matrix is too big to be explicitly represented. As we saw in section 3.1, most modern SVD algorithms don't require an explicit representation of the matrix $\boldsymbol{M}$ but only a function $m(\boldsymbol{v}) = \boldsymbol{M}\boldsymbol{v}$. We use a reduced form of (3) to approximate the PPR matrix.

$$m(\boldsymbol{v}) = \bar{\boldsymbol{\Pi}}_{\boldsymbol{l}}\boldsymbol{v} = (\alpha\sum_{i=0}^{l}(1-\alpha)^i\boldsymbol{P}^i\boldsymbol{v}) - (\boldsymbol{\pi}\cdot\boldsymbol{v})\mathbf{1} \tag{4}$$

where $\mathbf{1}$ is the vector of which all entries equal 1 and $\boldsymbol{\pi}$ is the (not-personalized) PageRank vector of the matrix.

We know from [11] that a few steps are usually enough to compute an approximation of the PageRank vector that outlines the community of the node. We fix $l = 10$ for the rest of this paper.

3.4 Finding the communities

SVD provides representations that should, if used to reconstruct the matrix, contain only the main patterns of the matrix which are the communities. There is

no reason however to think that the dimensions of the representations correspond themselves to communities. That is why we perform a clustering of the vertices using their SVD representations to find the communities. Since both matrices of the embedding provide relevant and distinct information, there is no reason to exclude one and therefore we use a concatenation of both sides as the vectors for the clustering. Choosing the best clustering algorithm for this step is out of the scope of this paper, we simply use the well-known k-means++ algorithm.

As we saw before, the PPR and reversed PPR vectors for vertices of a same community are expected to have similar patterns of high and low entries, which correspond to similar direction of the vectors. They are however not supposed to have similar norms, especially for the reversed PPR for which the norm typically follows the importance of the vertex. To make the clustering algorithm work on directions and not on euclidean distance both embeddings are normalized before concatenation and then the cosine similarity is used.

3.5 Reconstructing the communities rows and columns

The $\boldsymbol{U\Sigma}$ and $\boldsymbol{V\Sigma}$ embeddings are the projection of the columns and rows of the centered PPR matrix onto the singular spaces, e.g. for a vertex of the graph w, we have $(\boldsymbol{U\Sigma})_{w,\cdot} = \bar{\boldsymbol{\pi}}_{\boldsymbol{w}}{}^{\top}\boldsymbol{V}$. Therefore the clusters centers are the central columns and rows of each community, projected on the singular spaces. To reconstruct the true central columns and rows of the communities, which are the typical centered reversed PPR and centered PPR vectors for the community, we multiply by the transposed of the projection matrices, which are $\boldsymbol{U}$ and $\boldsymbol{V}$. Note that this reconstruction is not perfect because the dimensions of the singular spaces we use are smaller than the dimension of the original space.

4 Metrics

Several metrics have been proposed to evaluate the interpretability of a graph embedding, such as the Interpretability Score (IS) [4], the Betweenness Centrality Importance (BCI) and Closeness Centrality Importance (CCI) [9]. Those metrics all consider the vector representation of a given node interpretable if it encodes somehow whether that node belongs to some real-world communities.

We evaluate the results of our algorithms in terms of IS. We also study the BCI and CCI, however, due to the strong limitations of BCI and CCI, discussed in Section 4.2 and because of space limitations, we do not include them in our experimental evaluation. Finally, we propose the new Complete Interpretability Score Integrating Priority (CISIP) metric to tackle some weaknesses of the previous metrics.

4.1 Interpretability score (IS)

Let k be the index of one of the embedding dimensions, and C_g one of the ground-truth communities. The IS for a (k, g) pair is decomposed into a top part

$IS_{top(k,g)}$ that equals the recall@$|C_g|$ of the top-scores vertices of the embedding and a bottom $IS_{bottom(k,g)}$ part defined similarly on the lowest scores. The article then proposes to aggregate the scores by dimension or ground-truth group using either average or maximum function. In order for us to obtain a single result score for the entire embedding, we will aggregate first by taking the maximum IS per embedding dimension over the ground-truth groups and then the mean over all embedding dimensions. The use of the mean allows us to obtain scores between 0 and 1. In contrast with [4], we do not multiply the result by 100, which only changes the results by this factor without any other impact.

$$IS = \frac{1}{K}\sum_{k=1}^{K} \max_{g\in[0,G]}(\max(IS_{top(k,g)}, IS_{bottom(k,g)}) \tag{5}$$

4.2 Betweenness Centrality Importance (BCI) and Closeness Centrality Importance (CCI)

These scores are defined based on the well-known Betweenness Centrality and Closeness Centrality scores, introduced in [2]. For a (k, g) pair, the top BCI score is defined as the normalized Betweenness Centrality in relation with the community, of the top nodes of the dimension that belong to the community. The CCI top score is defined similarly but using the Closeness Centrality instead of the Betweenness Centrality. However the normalization of these scores make them harder to use and interprete. For a (k, g) pair, the scores are normalized by the number of nodes with non-null contribution. Let's consider a community C_g and two embedding dimensions k_1 and k_2 so that the first one contains only one very central vertex among its top vertices, and the second one contains this same vertex plus another slightly less central vertex. Then, although arguably much more interpretable w.r.t the g^{th} ground-truth community, the k_2 embedding dimension will have a lower score than the k_1 one.

4.3 CISIP metric

A weak point of the metrics already proposed in the literature is that, although they evaluate the fitness of an embedding dimension to a ground-truth group, they don't evaluate how well the embedding separates the data and the redundancy between the dimensions. Let's take the IS as an example: if 50% of the $|C_1|$ highest scores for the first dimension belong to C_1, then $IS_{top(1,1)} = 0.5$. But then it is possible that 50% of the highest scores for the second dimension belong either to the exact same part of C_1, denoting strong redundancy for the interpretations of these dimensions, or the other points of C_1, denoting that the first group is halved between these dimensions hence reducing the interpretability of both dimensions. In both cases, $IS_{top(2,1)} = 0.5$.

On the top of that all these metrics only consider the nodes that receive top- or bottom-$|C_g|$ scores from the embedding, and all these vertices are consider with equal weight. It seems however natural to consider that the importance

should be decreasing before reaching the $|C_g|$ threshold, e.g. the vertex with the highest score should have higher importance than the vertex with the second highest score. Similarly, it seems that the importance should not drop to 0 after the $|C_g|^{\text{th}}$ score: if two embedding dimensions have the exact same top-$|C_g|$ vertices, but one has its $(|C_g|+1)^{\text{th}}$ vertex belonging to C_g while the other has not, it seems natural to consider that the first one is more interpretable wrt. C_g.

To tackle these weaknesses, we propose the new Complete Interpretability Score Integrating Priority (CISIP). First of all, we use a function f to smooth the binary belonging feature, providing a score of belonging or importance of each node in each community. The simplest of these functions is simply to use the belonging feature (identity function), but we can also use the Mean Belonging of the Neighbors, the PPR of the community or other functions.

A score is then attributed to each (k, g) pair. Using the Hungarian algorithm for optimal assignment, an exclusive mapping $S = \{(k_i, g_i), i \in [1, \min(K, G)]\}$ of the embedding dimensions to the ground-truth groups is performed. To achieve good performances, the scoring at this step for a couple (k, g) is computed by summing the embedding scores for the dimension k of the vertices that belong to C_g. To account for the possibility that the bottom part of the embedding is the part that matches the community, we score in the same way the opposite of the dimension and we take the max score of the two. If the score with the opposite is the max, we will then use the opposite of the dimension for the Weighted Kendall Tau scoring.

Then, for each pair $(k_i, g_i) \in S$, a score is computed using Vigna's weighted Kendall Tau score WKT [16]. This method provides a score between 0 and 1 of how much the two vectors (embedding score and smoothed ground-truth belonging) are ranked in the same way, with more importance given to the top-scores of each vector. Finally, our metric $CISIP$ is computed as the mean score over the dimensions and groups

$$CISIP = \frac{1}{\min(K, G)} \sum_{i=1}^{\min(K,G)} WKT(v_d, f(w_g)) \tag{6}$$

5 Experiments

Our main goal is to show that our method provides better interpretability scores than state-of-the-art approaches, while boasting similar results for a popular machine learning task in graph analysis, such as link prediction.

Datasets. We use two datasets that are available on the SNAP website [19]. The first one named *Wikispeedia* contains 4592 vertices, and 120 000 edges. 105 overlapping ground-truth community are given. The second one named *Facebook* contains 10 graphs. We exclude 2 of them, numbered 698 and 3980, because they contain less than 128 nodes and we could therefore not compute the SVD for them using the same parameters used for the others. The remaining 8 graphs contain between 155 and 1035 vertices and between 3312 and 60050 edges. Between 7 and 46 overlapping ground-truth communities are given for each graph.

We also release a new dataset with ground-truth communities[1] (Wikipedia fr), constructed from all the pages of the French version of Wikipedia.[2] In such a graph, nodes represent Wikipedia pages while directed edges represent links between the corresponding Wikipedia pages. In the French version of Wikipedia, it is common to add links to so called "portals" at the end of the page which serve as reference pages for given topics and can be seen as ground-truth communities. Such novel graph contains 2.52 millions vertices, 102 million edges and 2 700 ground-truth communities. To keep the dimension of the embeddings manageable however we only study the 117 communities that contain more than 10 000 vertices.

Methods. We evaluate our method against two widely used algorithms for graph embedding:

- HOPE, a state-of-the-art approach based on SVD;
- node2vec, which is a state-of-the-art approach based on random walks.

We evaluate all methods in terms of the IS and CISIP metrics, discussed in the previous section. We include in our experiments embeddings produced by the state-of-the-art node2vec and HOPE algorithms and our 2 new embeddings PARFAITE_L and PARFAITE_R .

For node2vec, we use the most-used python3 implementation [3]. We keep the default parameters, i.e. the number of walks is 200 per vertex, the length of the walks is 30 and the window size is 10.

For HOPE, we use the official implementation in Matlab[3] provided by the authors of [14] that we reimplement in python3 (while using numpy and scipy), so as to provide a fair comparison with the other approaches.

Our algorithm is implemented in python3 using mainly the numpy [7] and scipy [17] packages. The jump parameter α for the PPR algorithm is set to $\alpha = 0.1$. The number l of iterations to approximate the PPR matrix is set to $l = 10$ and the dimension D of the intermediate SVD is set to $D = 128$.

For all embeddings the dimension K of the embedding is set to be the number G of known ground-truth communities.

Metrics. We measure the interpretability of the methods using each of the metrics mentioned in section 4. The Interpretability Scores (IS) are aggregated along the ground-truth groups using the max function, and then along the embedding dimensions using the sum function. For CISIP, three smoothing functions f are considered. The first one, that we call "identity" is the direct use of the binary belonging feature, the second one is the Mean Neighbors Belonging (MNB), that is the mean of the belonging features of the neighbors, and the last one is the PPR of the community.

[1] https://gitlab.telecom-paris.fr/gabriel.damay/WikipediaFRNetwork

[2] All the pages from the main space of Wikipedia, that is all the pages usually accessed by public, excluding discussions, user pages etc.

[3] https://github.com/ZW-ZHANG/HOPE

5.1 Results

The results are given in Table 2 and 3. Because of space limitation, only Facebook 107 and 1912, as parts with highest degree and order of Facebook, are presented, as well as Facebook 3437 because of the good performances of HOPE on this dataset with the CISIP-identity metric. The results for the other versions of Facebook are similar to what is presented.

As we see in these results, our algorithm's interpretability is much higher than node2vec's when evaluated using the Interpretability Score or any of the variants of CISIP. HOPE's interpretability is closer but still generally below PARFAITE's.

Our left embedding greatly outperforms the right one when using CISIP with the PPR smoothing, and this result was expected, as the left embedding is an approximation of the typical PPR vector of each clusters detected. The results are however equivalent between our two embeddings when compared using either the IS or CISIP with the identity smoothing, which both evaluate directly the matching between the embedding and the belonging features, or with the MNB smoothing. This is consistent with our interpretation that the left embedding represents which communities a vertex belongs to, while the right embedding measures somehow the "importance" of a vertex in a community.

5.2 Is the clustering needed ?

To check if the last step of our algorithm of clustering the data and computing the final embeddings is really needed, we compare our results to what we would have without the clustering step. To achieve this, we take the $\boldsymbol{U\Sigma}$ and $\boldsymbol{V\Sigma}$ results from the SVD, and we keep only the G first dimensions to match the dimension of the PARFAITE embeddings.

The results for IS and for CISIP with the three smoothing functions already used are presented in Table 4 and 5. We see that in most cases the results of our PARFAITE_L and PARFAITE_R outperform those before clustering, sometimes significantly (e.g. more than 0.1 points of difference for the IS). This is especially true for the IS and CISIP with PPR smoothing. We note however that the SVD provides better results in several occurrences when compared to our embeddings using CISIP with the identity or the MNB smoothing.

Overall we conclude that PARFAITE_L and PARFAITE_R do generally perform better than single SVD, i.e. without the clustering and reconstruction steps.

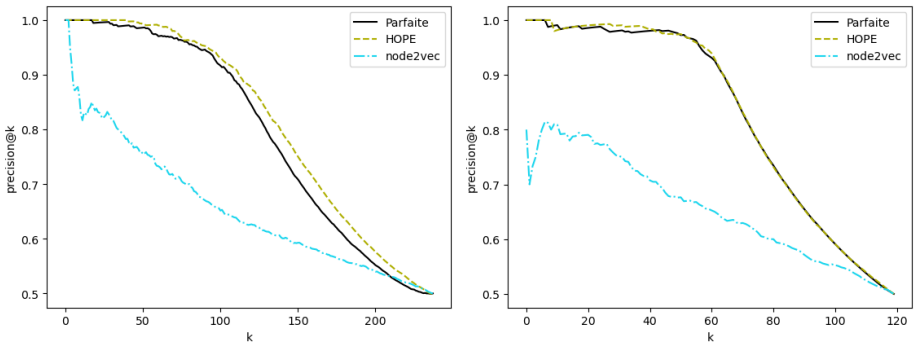
5.3 Is our embedding efficient ?

We check the efficiency of PARFAITE against HOPE and node2vec at the task of Link Prediction. We build test graphs by removing 0.1% of the edges of a real-world graph. We store the pairs of vertices of these edges as "positive" pairs, and build a set of "negative" pairs by drawing the same number of pairs of vertices that are not connected by an edge.

The embedding of the test graph is computed and a score is attributed to each positive or negative pair of vertices using this embedding.

For HOPE, the score is the dot product of the left embedding of the first vertex in the pair and the right embedding of the second vertex. The score for PARFAITE is similar but the left embedding of the first vertex is normalized to account for the entire use of the singular values on each side of the embedding. For node2vec, following [6], a Logistic Regression is trained on the test graph by representing the pairs with a concatenation of their vertice's embeddings.

We run this experiment on 10 test graph for both the Wikispeedia dataset and on the 1912 part of the Facebook dataset, as the part with the highest order. The mean results are given in Figure 2. As we can see, node2vec is outperformed by both HOPE and PARFAITE. HOPE achieves similar results as PARFAITE on Facebook 1912 and slightly better on Wikispeedia but overall we can say that the greater interpretability of PARFAITE does not come at the cost of lower efficiency on the task of link prediction.



(a) Wikispeedia (b) Facebook 1912

Fig. 2. Mean Precision@k of the Link Prediction on 20 test graphs built from the Wikispeedia and Facebook 1912 datasets

6 Conclusion and future work

We have presented PARFAITE, a novel graph embedding algorithm and we have evaluated its interpretability against the state-of-the-art node2vec and HOPE algorithms. We also presented the new interpretability score CISIP to assure that interpretations are well-separated and non-redundant. Our approach relies on the clustering of the vertex embeddings for which we use the k-means algorithm. An interesting direction for future work would be to study whether other clustering methods provide better results. We have also seen that, although the clustering phase increases the interpretability of our embedding, in some rare

Table 2. Results for the Interpretability Score

Dataset	PARFAITE		HOPE		node2vec
	left	right	left	right	
Wikispeedia	**0.389**	0.375	0.200	0.266	0.139
Facebook 107	**0.626**	0.601	0.550	0.550	0.323
Facebook 1912	**0.766**	0.764	0.603	0.603	0.348
Facebook 3437	0.430	**0.759**	0.429	0.429	0.188
WikipediaFr	0.040	**0.083**	0.041	**0.083**	*

* The embedding for this graph with node2vec as been stopped after 36h of computation.

cases clustering might have a negative impact on the interpretability. It would be interesting to have a better understanding of this phenomenon which could pave the way for a more effective approach. Finally, the CISIP metric contains a smoothing function as a parameter and it would be interesting to study the properties of various functions for this task and what they imply in terms of interpretability of the embedding at hand.

References

1. Belkin, M., Niyogi, P.: Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. Neural Computation **15**(6), 1373–1396 (2003). https://doi.org/10.1162/089976603321780317
2. Bloch, F., Jackson, M.O., Tebaldi, P.: Centrality measures in networks. Social Choice and Welfare **61**(2), 413–453 (2023)
3. Elior Cohen: node2vec 0.4.6, https://pypi.org/project/node2vec/
4. Gogoglou, A., Bruss, C.B., Hines, K.E.: On the Interpretability and Evaluation of Graph Representation Learning. In: NeurIPS workshop on Graph Representation Learning (2019)
5. Goyal, P., Ferrara, E.: Graph embedding techniques, applications, and performance: A survey. Knowledge-Based Systems **151**, 78–94 (2018). https://doi.org/10.1016/j.knosys.2018.03.022
6. Grover, A., Leskovec, J.: node2vec: Scalable Feature Learning for Networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 855–864 (2016). https://doi.org/10.1145/2939672.2939754
7. Harris, C.R., Millman, K.J., van der Walt, S.J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N.J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M.H., Brett, M., Haldane, A., del Río, J.F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., Oliphant, T.E.: Array programming with NumPy. Nature **585**(7825), 357–362 (2020). https://doi.org/10.1038/s41586-020-2649-2
8. Hollocou, A., Bonald, T., Lelarge, M.: Multiple local community detection. SIGMETRICS Perform. Eval. Rev. **45**(3), 76–83 (2018). https://doi.org/10.1145/3199524.3199537

Table 3. Results for CISIP

Dataset	PARFAITE		HOPE		node2vec
	left	right	left	right	
No smoothing (identity function)					
Wikispeedia	**0.429**	0.414	0.339	0.318	0.255
Facebook 107	0.403	**0.408**	0.316	0.316	0.234
Facebook 1912	0.401	**0.425**	0.352	0.353	0.266
Facebook 3437	0.300	0.310	**0.331**	**0.331**	0.256
WikipediaFr	0.350	**0.376**	0.339	0.353	*
Smoothing with MNB					
Wikispeedia	0.431	**0.508**	0.282	0.369	0.105
Facebook 107	**0.545**	0.515	0.384	0.384	0.161
Facebook 1912	0.504	**0.511**	0.269	0.267	0.029
Facebook 3437	0.513	**0.524**	0.373	0.373	0.067
WikipediaFr	0.272	**0.499**	0.400	0.493	*
Smoothing with PPR					
Wikispeedia	0.449	**0.516**	0.189	0.174	0.012
Facebook 107	**0.585**	0.576	0.304	0.304	0.093
Facebook 1912	0.557	**0.633**	0.278	0.279	0.101
Facebook 3437	0.568	**0.688**	0.402	0.402	0.073
WikipediaFr	**0.123**	-0.067	0.049	-0.015	*

Table 4. SVD Results for the Interpretability Score

Dataset	SVD (left)	SVD (right)
Wikispeedia	0.234	0.204
Facebook 107	0.538	0.516
Facebook 1912	0.589	0.604
Facebook 3437	0.270	0.463
WikipediaFr	**0.043**	0.061

9. Khoshraftar, S., Mahdavi, S., An, A.: Centrality-based Interpretability Measures for Graph Embeddings. In: 2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA). pp. 1–10 (2021). https://doi.org/10.1109/DSAA53316.2021.9564221
10. Kipf, T.N., Welling, M.: Semi-Supervised Classification with Graph Convolutional Networks. In: International Conference on Learning Representations (2022)
11. Kloumann, I.M., Kleinberg, J.M.: Community membership identification from small seed sets. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p. 1366–1375 (2014). https://doi.org/10.1145/2623330.2623621
12. Konstantinides, K., Natarajan, B., Yovanof, G.: Noise estimation and filtering using block-based singular value decomposition. IEEE Transactions on Image Processing **6**(3), 479–483 (1997). https://doi.org/10.1109/83.557359
13. Lee, S., Song, B.C.: Interpretable Embedding Procedure Knowledge Transfer via Stacked Principal Component Analysis and Graph Neural Network. Proceed-

Table 5. SVD Results for CISIP

	No smoothing		MNB smoothing		PPR smoothing	
Dataset	**SVD_L**	**SVD_R**	**SVD_L**	**SVD_R**	**SVD_L**	**SVD_R**
Wikispeedia	0.363	0.362	0.280	0.369	0.244	0.202
Facebook 107	**0.493**	**0.464**	**0.596**	**0.571**	**0.609**	**0.604**
Facebook 1912	0.377	0.392	0.299	0.320	0.319	0.320
Facebook 3437	**0.304**	**0.337**	0.429	0.443	0.419	0.421
WikipediaFr	**0.350**	**0.381**	0.259	**0.502**	**0.150**	**-0.011**

ings of the AAAI Conference on Artificial Intelligence **35**(9), 8297–8305 (2021). https://doi.org/10.1609/aaai.v35i9.17009
14. Ou, M., Cui, P., Pei, J., Zhang, Z., Zhu, W.: Asymmetric transitivity preserving graph embedding. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p. 1105–1114 (2016). https://doi.org/10.1145/2939672.2939751
15. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Tech. rep., Stanford InfoLab (1999)
16. Vigna, S.: A Weighted Correlation Index for Rankings with Ties. In: Proceedings of the 24th International Conference on World Wide Web. pp. 1166–1176 (2015). https://doi.org/10.1145/2736277.2741088
17. Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S.J., Brett, M., Wilson, J., Millman, K.J., Mayorov, N., Nelson, A.R.J., Jones, E., Kern, R., Larson, E., Carey, C.J., Polat, İ., Feng, Y., Moore, E.W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E.A., Harris, C.R., Archibald, A.M., Ribeiro, A.H., Pedregosa, F., van Mulbregt, P., SciPy 1.0 Contributors: SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods **17**, 261–272 (2020). https://doi.org/10.1038/s41592-019-0686-2
18. Wu, J., Ngo, C.W.: Interpretable Embedding for Ad-Hoc Video Search. In: Proceedings of the 28th ACM International Conference on Multimedia. pp. 3357–3366 (2020). https://doi.org/10.1145/3394171.3413916
19. Yang, J., Leskovec, J.: Defining and evaluating network communities based on ground-truth. In: Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics (2012). https://doi.org/10.1145/2350190.2350193
20. Yang, R., Shi, J., Xiao, X., Yang, Y., Bhowmick, S.S.: Homogeneous network embedding for massive graphs via reweighted personalized pagerank. Proc. VLDB Endow. **13**(5), 670–683 (2020). https://doi.org/10.14778/3377369.3377376
21. Yin, Y., Wei, Z.: Scalable graph embeddings via sparse transpose proximities. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. p. 1429–1437 (2019). https://doi.org/10.1145/3292500.3330860
22. Zhang, Y., Xia, X., Xu, X., Yu, F., Wu, H., Yu, Y., Wei, B.: Robust Hierarchical Overlapping Community Detection With Personalized PageRank. IEEE Access **8**, 102867–102882 (2020). https://doi.org/10.1109/ACCESS.2020.2998860
23. Zhou, C., Liu, Y., Liu, X., Liu, Z., Gao, J.: Scalable graph embedding for asymmetric proximity. Proceedings of the AAAI Conference on Artificial Intelligence **31**(1) (2017). https://doi.org/10.1609/aaai.v31i1.10878