Power Attributed Graph Embedding and Clustering

Lazhar Labiod, Mohamed Nadif

Power Attributed Graph Embedding and Clustering

Lazhar Labiod, Mohamed Nadif
Centre Borelli UMR 9010, Université Paris Cité, 75006 Paris, France

***Abstract*—Representation learning is a central problem of *Attributed Networks* data analysis in a variety of fields. Given an attributed graph, the objectives are to obtain a representation of nodes and a partition of the set of nodes. Usually these two objectives are pursued separately via two tasks that are performed sequentially, and any benefit that may be obtained by performing them simultaneously is lost. In this paper we propose a *Power Attributed Graph Embedding and clustering* (PAGEC for short) in which the two tasks, embedding and clustering, are considered together. To jointly encode data affinity between node links and attributes, we use a new powered proximity matrix. We formulate a new matrix decomposition model to obtain node representation and node clustering simultaneously. Theoretical analysis shows the close connections between the new proximity matrix and the *random walk* theory on a graph. Experimental results demonstrate that the `PAGEC` algorithm performs better, in terms of clustering and embedding, than state-of-the-art algorithms including deep learning methods designed for similar tasks in relation to attributed network datasets with different characteristics.**

***Index Terms*—Attributed graph, Embedding, Clustering, Spectral rotation.**

I. Introduction

Attributed Networks (AN) [1] have been used to model a wide variety of real-world networks, such as academic and healthcare networks, where both node links and attributes/features are available for analysis. In contrast to plain networks that contain only node links and dependencies, in AN each node is associated with a valuable set of features.

More recently, representation learning has become an important objective in applications including social networks, academic citation networks and protein-protein interaction networks. *Attributed Network Embedding* (ANE) [2] seeks to obtain a continuous low-dimensional matrix representation of the nodes in a network that preserves the topological structure and node attribute proximity of the original network.

Although *Network Embedding* (NE) has given rise to a number of approaches such as [3], research on ANE has so far received little attention [4]. Unlike NE, which learns from plain networks, ANE seeks to utilize information relating both to node proximity and to the affinity of node attributes within the network. Since the two information sources are not the same, it is difficult for existing NE algorithms to be directly applied to ANE.

Learned representations have been shown to be helpful in many learning tasks such as network clustering [5], node visualization [6], node classification [7], and link prediction [8], and as a consequence ANE is becoming a pressing topic for research in which challenging issues of high-dimensionality, sparsity and nonlinearity need to be addressed.

Existing AN clustering methods have been applied widely, but they often perform poorly because of (1) the likelihood that an approximate continuous embedding solution will deviate significantly from a good discrete clustering, and (2) a loss of information between the independent stages, namely continuous embedding generation and embedding discretization.

II. Related Work

Learning a low-dimensional vector representation for each vertex of a network data is a good way to analyze the network. This task attracted the attention of several authors [9] for different purposes such as detection anomalies in dynamic networks [10]. Recently, various models have been proposed for attributed networks showing that jointly learning network representations with network topology information and vertex attributes enhance the performance on various tasks including clustering; in [11] the authors demonstrated the benefits of clustering. In this regard, a number of approaches have been developed, based on matrix decomposition, graph clustering, and deep representation learning. Most of the work that we consider in our comparisons in Section V are the following. Spectral Clustering [12] is a widely used approach for learning social embedding. With the advent of deep learning, several works have tackled the same problem. `Graph Encoder` [13] learns graph embedding for spectral graph clustering, while `DNGR` [14] trains a stacked denoising autoencoder for graph embedding. [15] developed `DeepWalk`, a network representation approach which encodes social relations into a continuous embedding space. [16] proposed `GAE`, using an autoencoder-based unsupervised framework for attributed network data embedding, and `VGAE`, a variational graph autoencoder approach for graph embedding with both node link and node attribute information. [5] presented `MGAE`, a marginalized graph autoencoder for graph clustering. [8] proposed `ARGA`, which is the most recent adversarially regularized autoencoder algorithm using a graph autoencoder to learn the embedding, while the `ARVGA` [8] algorithm uses a variational graph autoencoder to learn the embedding. More recently, [17] developed a new attributed graph clustering algorithm `AGC` based on adaptive graph convolution. In [18] a deep attentional embedding approach `DAEGC` for attributed graph clustering is proposed.

The sequential process in which a learned representation is obtained before clusters are then obtained using a clustering method is a source of problems. The two tasks do not share the same objective and are carried out separately. For this reason, simultaneous embedding and clustering are frequently used to improve representation learning by making use of structure information from the clusters. A number of approaches have been proposed for the representation learning and clustering tasks [19], [20]. However, none of these approaches has

attempted to integrate available information contained within a network, which can be seen as a deficiency. In order to overcome this, we propose a novel simultaneous ANE and clustering scheme which simultaneously (1) learns embedding from information on network topology and on attributes, and (2) learns continuous embedding and discrete clustering labels. Specifically, we explicitly enforce a discrete transformation on the intermediate continuous labels (embedding), which leads to a tractable optimization problem with a discrete solution. The key challenge is knowing how to integrate the information on both node links and attributes in order to carry out node representation learning and discrete node clustering at the same time. To compensate for the information loss when sequential clustering is relaxed, and to obtain a discrete clustering solution, we use a smooth transformation (e.g., rotation) from the relaxed continuous embedding to a discrete solution. In this sense, the continuous embedding only serves as an intermediate product.

To the best of our knowledge, simultaneous attributed network embedding and clustering in a unified learning framework has not so far been adequately investigated. The goal of the present work is to look at this issue by considering matrix decomposition as the embedding framework.

III. Proposed method

In this section we describe the Simultaneous Attributed Network Embedding and Clustering method that we have called PAGEC. We present the formulation of an objective function and an effective algorithm for data embedding and clustering. But we begin by describing the construction of two matrices $\mathbf{S}$ and $\mathbf{M}$ integrating both types of information – content and structure information – that we use to achieve our objective.

A. Content and Structure information

An attributed network $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ consists of the set of nodes V, the set of links $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, and $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$ where $n = |\mathcal{V}|$ and $\mathbf{x}_i \in \mathbb{R}^d$ is the feature/attribute vector of the node v_i. Formally, the graph can be represented by two types of information, namely the content information $\mathbf{X} \in \mathbb{R}^{n \times d}$ and the structure information $\mathbf{A} \in \mathbb{R}^{n \times n}$, where $\mathbf{A}$ is an adjacency matrix of $\mathcal{G}$ and $a_{ij} = 1$ if $e_{ij} \in \mathcal{E}$ otherwise 0; we consider that each node is a neighbor of itself, then we set $a_{ii} = 1$ for all nodes. We therefore model node proximity by an $(n \times n)$ transition matrix $\mathbf{W}$ given by $\mathbf{W} = \mathbf{D}^{-1}\mathbf{A}$, where $\mathbf{D}$ is the degree matrix of $\mathbf{A}$ defined by $d_{ii} = \sum_{i'=1}^{n} a_{i'i}$.

In order to utilize additional information about node similarity from $\mathbf{X}$, we first preprocess the above dataset $\mathbf{X}$ to produce similarity graph input $\mathbf{W}_\mathbf{X}$ of size $(n \times n)$; we then construct a K-Nearest-Neighbor (KNN) graph. To this end we use the heat kernel and L_2 distance, KNN neighborhood mode with a given K and we set the width of the neighborhood $\sigma = 1$. Note that any appropriate distance or dissimilarity measure can be used. Finally we combine node proximity from both content information $\mathbf{X}$ and structure information $\mathbf{W}$ in an $(n \times n)$ matrix $\mathbf{S}$. We thus propose perturbing the similarity $\mathbf{W}$ by adding the similarity from $\mathbf{W}_\mathbf{X}$; we choose to define $\mathbf{S}$ as follows $\mathbf{S} = \mathbf{W} + \mathbf{W}_\mathbf{X}$. In Figure 1 multidimensional scaling is applied to $\mathbf{W}$ and $\mathbf{S}$ in order to illustrate the impact of $\mathbf{W}_\mathbf{X}$. Note that with $\mathbf{S}$, the sparsity is overcome by the presence of $\mathbf{W}_\mathbf{X}$. Later we will see the interest of using $\mathbf{W}_\mathbf{X}$ in $\mathbf{S}$. Since clustering is our main

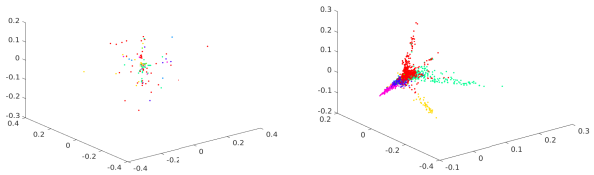

Fig. 1. MDS on $\mathbf{W}$ (left) and $\mathbf{S}$ (right) ($K = 35$): Cora dataset where $\mathbf{W}$ of size (2708×2708) with true 7 clusters.

objective, we propose integrating $\mathbf{S}$ in the formulation of a new data representation by assuming that nodes with the same label tend to similar social relations and similar node attributes. This reflects the fact that labels are strongly influenced by both content and structure information and are inherently correlated to the two information sources, and we are reminded of the idea underlying *Canonical Discriminant Analysis* (CDA), which is a dimension-reduction technique related to principal component analysis (PCA) and canonical correlation [21]. Given groups of observations with measurements on attributes, CDA derives the linear combination of variables that has the highest possible multiple correlation with the groups. It can be seen as a particular kind of PCA in which the observations belonging to a same group are replaced by their centroid. The new data representation $\mathbf{M} = (m_{ij})$ of size $(n \times d)$ can be considered as a multiplicative integration of $\mathbf{W}$ and $\mathbf{X}$ replacing each node by the centroid of its neighborhood (barycenter): i.e, $\mathbf{m}_{ij} = \sum_{k=1}^{n} \mathbf{w}_{ik}\mathbf{x}_{kj}, \forall i, j$ or $\mathbf{M} = \mathbf{WX}$. Since $\mathbf{W}$ is a transition matrix, in order to make better use of the random walk properties we use $\mathbf{W}^p$ to explore the structure of $\mathbf{W}$, where a random walk includes multiple steps instead of only one. $\mathbf{M}$ is then given by

$$\mathbf{M} = \mathbf{W}^p\mathbf{X} \quad \text{where } p \in \mathbb{N}_+. \tag{1}$$

This modification is simple to describe and leads to a refinement of the feature matrix $\mathbf{X}$, creating what can be seen as a smooth version with each row of $\mathbf{M}$ converging to the prototype of its class. $\mathbf{M}$ will therefore be more helpful in the clustering task. In Figure 2 it is interesting to visualize the impact of $\mathbf{W}$ in the formulation of $\mathbf{M}$. To this end we apply CDA on $\mathbf{X}$ and $\mathbf{M}$ and indicate the seven true clusters of the Cora dataset. This reveals clusters separated with $\mathbf{M} = \mathbf{WX}$ and, even better, separated with $\mathbf{M} = \mathbf{W}^8\mathbf{X}$, showing the impact of $\mathbf{W}^p$, which can be seen already to do a good job independently of clustering.

B. Definition of the model and Optimisation

Let k be the number of clusters and also the number of components into which the data is embedded. With $\mathbf{M}$ and $\mathbf{S}$, our PAGEC method seeks to obtain the maximally informative embedding with respect to the clustering structure

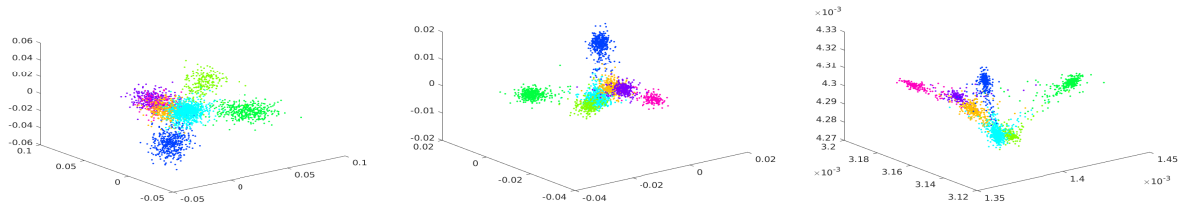

Fig. 2. Cora dataset (2708×1433) with true 7 clusters: CDA on $\mathbf{X}$ (left), $\mathbf{M} = \mathbf{WX}$ (middle) and $\mathbf{M} = \mathbf{W}^8\mathbf{X}$ (right). To evaluate the separability of different classes, we rely on the ratio between-class scatter matrix S_b and total-class scatter S_t defined by R-square $= \frac{\text{Tr}(S_b)}{\text{Tr}(S_t)}$ where Tr(.) denotes Trace(.). We have respectively 2.58%, 5.41% and 10.52%.

in the attributed network data. The proposed objective function $\mathcal{F}(\mathbf{B}, \mathbf{Z}, \mathbf{Q}, \mathbf{G})$ to be minimized is consequently given by

$$\begin{aligned} \left\|\mathbf{M} - \mathbf{B}\mathbf{Q}^\top\right\|^2 \quad &+ \quad \lambda\left\|\mathbf{S} - \mathbf{G}\mathbf{Z}\mathbf{B}^\top\right\|^2 \\ \text{s.t.} \quad &\mathbf{B}^\top\mathbf{B} = \mathbf{I}, \mathbf{Z}^\top\mathbf{Z} = \mathbf{I}, \mathbf{G} \in \{0,1\}^{n\times k} \end{aligned} \quad (2)$$

where $\mathbf{G} = (g_{ij})$ of size $(n \times k)$ is a cluster membership matrix, $\mathbf{B} = (b_{ij})$ of size $(n \times k)$ is the embedding matrix, and $\mathbf{Z} = (z_{ij})$ of size $(k \times k)$ is an orthonormal rotation matrix which most closely maps $\mathbf{B}$ to $\mathbf{G} \in \{0,1\}^{n\times k}$. $\mathbf{Q} \in \mathbb{R}^{d\times k}$ is the features embedding matrix. Finally, the parameter λ is a non-negative value and can be viewed as a regularization parameter.

The idea behind factorizing $\mathbf{M}$ and $\mathbf{S}$ is so that nodes with similar proximity, those with greater similarity in the two matrices, will have closer representations in the latent space given by $\mathbf{B}$. This way, optimizing (2) leads to a clustering of the nodes into k clusters given by $\mathbf{G}$. Note that both tasks – embedding and clustering – are performed simultaneously and supported by $\mathbf{Z}$; this is the key to attaining good embedding while taking the clustering structure into account.

The first term of PAGEC can be seen as a generalization of canonical discriminant analysis in which the partition of nodes is replaced by a more general graph structure defined *a priori* on the set of nodes. To infer the latent factor matrices $\mathbf{Z}$, $\mathbf{B}$, $\mathbf{Q}$ and $\mathbf{G}$ from $\mathbf{M} = \mathbf{W}^p\mathbf{X}$ and $\mathbf{S} = \mathbf{W} + \mathbf{W}_\mathbf{X}$, we derive an alternating optimization algorithm. To this end, we make use of the following proposition.

Proposition 1. *Given* $\mathbf{S} \in \mathbb{R}^{n\times n}$, $\mathbf{G} \in \{0,1\}^{n\times k}$, $\mathbf{Z} \in \mathbb{R}^{k\times k}$, $\mathbf{B} \in \mathbb{R}^{n\times k}$, *we have*

$$\left\|\mathbf{S} - \mathbf{G}\mathbf{Z}\mathbf{B}^\top\right\|^2 = \left\|\mathbf{S} - \mathbf{S}\mathbf{B}\mathbf{B}^\top\right\|^2 + \|\mathbf{S}\mathbf{B} - \mathbf{G}\mathbf{Z}\|^2 \quad (3)$$

Proof. First, since $\mathbf{B}^\top\mathbf{B} = \mathbf{I}$ and $\text{Tr}(AB) = \text{Tr}(BA)$ we have

$$\|\mathbf{G}\mathbf{Z}\mathbf{B}^\top\|^2 = \text{Tr}(\mathbf{B}\mathbf{Z}^\top\mathbf{G}^\top\mathbf{G}\mathbf{Z}\mathbf{B}^\top) = \text{Tr}(\mathbf{Z}^\top\mathbf{G}^\top\mathbf{G}\mathbf{Z}) = \|\mathbf{G}\mathbf{Z}\|^2.$$

Similarly we have $\|\mathbf{S}\mathbf{B}\mathbf{B}^\top\|^2 = \|\mathbf{S}\mathbf{B}\|^2$. This leads to

$$(a)\|\mathbf{S} - \mathbf{G}\mathbf{Z}\mathbf{B}^\top\|^2 = \|\mathbf{S}\|^2 + \|\mathbf{G}\mathbf{Z}\|^2 - 2\text{Tr}(\mathbf{S}\mathbf{G}\mathbf{Z}\mathbf{B}^\top).$$

$$\begin{aligned}(b)\|\mathbf{S} - \mathbf{S}\mathbf{B}\mathbf{B}^\top\|^2 &= \|\mathbf{S}\|^2 + \|\mathbf{S}\mathbf{B}\mathbf{B}^\top\|^2 - 2\text{Tr}(\mathbf{S}\mathbf{B}\mathbf{B}^\top\mathbf{S}^\top) \\ &= ||\mathbf{S}||^2 + ||\mathbf{S}\mathbf{B}||^2 - 2||\mathbf{S}\mathbf{B}||^2 \text{ since } \mathbf{S} = \mathbf{S}^\top \\ &= ||\mathbf{S}||^2 - ||\mathbf{S}\mathbf{B}||^2.\end{aligned}$$

$$(c)\|\mathbf{S}\mathbf{B} - \mathbf{G}\mathbf{Z}\|^2 = \|\mathbf{S}\mathbf{B}\|^2 + \|\mathbf{G}\mathbf{Z}\|^2 - 2\text{Tr}(\mathbf{S}\mathbf{G}\mathbf{Z}\mathbf{B}^\top).$$

Summing (b) and (c) (the right terms of (3)) leads to (a). □

Below we detail the different steps involved in inferring $\mathbf{Z}$, $\mathbf{Q}$, $\mathbf{B}$ and $\mathbf{G}$.

Compute $\mathbf{Z}$. By fixing $\mathbf{G}$ and $\mathbf{B}$ we reduce the problem that arises in (2) to $\min_\mathbf{Z} \left\|\mathbf{S} - \mathbf{G}\mathbf{Z}\mathbf{B}^\top\right\|^2$. From proposition 1, we deduce that

$$\min_\mathbf{Z} \left\|\mathbf{S} - \mathbf{G}\mathbf{Z}\mathbf{B}^\top\right\|^2 \Leftrightarrow \min_\mathbf{Z} \|\mathbf{S}\mathbf{B} - \mathbf{G}\mathbf{Z}\|^2 \quad (4)$$

which can be reduced to $\max_\mathbf{Z}$ $\text{Tr}(\mathbf{G}^\top\mathbf{S}\mathbf{B}\mathbf{Z})$ s.t. $\mathbf{Z}^\top\mathbf{Z} = \mathbf{I}$. It was shown (page 29) in [22], with $\mathbf{U}\Sigma\mathbf{V}^\top$ the SVD for $\mathbf{G}^\top\mathbf{S}\mathbf{B}$, that

$$\mathbf{Z} = \mathbf{U}\mathbf{V}^\top. \quad (5)$$

This problem turns out to be similar to the well-known orthogonal Procrustes problem [23].

Compute $\mathbf{Q}$. Given $\mathbf{G}$, $\mathbf{Z}$ and $\mathbf{B}$, (2) is reduced to $\min_\mathbf{Q} \left\|\mathbf{M} - \mathbf{B}\mathbf{Q}^\top\right\|^2$, and we get

$$\mathbf{Q} = \mathbf{M}^\top\mathbf{B}. \quad (6)$$

It is therefore possible for $\mathbf{Q}$ to be seen as an embedding of attributes.

Compute $\mathbf{B}$. Given $\mathbf{G}$, $\mathbf{Q}$ and $\mathbf{Z}$, (2) is equivalent to $\max_\mathbf{B}$ $\text{Tr}((\mathbf{M}^\top\mathbf{Q} + \lambda\mathbf{S}\mathbf{G}\mathbf{Z})\mathbf{B}^\top)$ s.t. $\mathbf{B}^\top\mathbf{B} = \mathbf{I}$. Similarly to when computing $\mathbf{Z}$, let $\hat{\mathbf{U}}\hat{\Sigma}\hat{\mathbf{V}}^\top$ be the SVD for $(\mathbf{M}^\top\mathbf{Q} + \lambda\mathbf{S}\mathbf{G}\mathbf{Z})$, and we get

$$\mathbf{B} = \hat{\mathbf{U}}\hat{\mathbf{V}}^\top. \quad (7)$$

It is important to emphasize that at each step, $\mathbf{B}$ makes use of the information from the matrices $\mathbf{Q}$, $\mathbf{G}$, and $\mathbf{Z}$. This highlights one of the aspects of a simultaneous embedding and clustering.

Compute $\mathbf{G}$: Finally, given $\mathbf{B}$, $\mathbf{Q}$ and $\mathbf{Z}$, the problem (2) is equivalent to $\min_\mathbf{G} \|\mathbf{S}\mathbf{B} - \mathbf{G}\mathbf{Z}\|^2$ since from (2) and (3) $\mathbf{G}$ is present only in $\|\mathbf{S}\mathbf{B} - \mathbf{G}\mathbf{Z}\|$. Thereby, we are faced with an *assignment step* like that case of the k-means algorithm where $\mathbf{G}$ is a cluster membership matrix. Therefore, it is computed as follows. We first fix $\mathbf{Q}$, $\mathbf{Z}$, $\mathbf{B}$, let $\tilde{\mathbf{B}} = \mathbf{S}\mathbf{B}$ we then compute

$$g_{ik} = \begin{cases} 1 & \text{if } k = \arg\min_{k'} ||\tilde{\mathbf{b}}_i - \mathbf{z}_{k'}||^2 \\ 0 & \text{otherwise.} \end{cases} \quad (8)$$

A the (t+1)th iteartion, this leads to $\|\tilde{B}^{(t)} - \mathbf{G}^{(t)}\mathbf{Z}^{(t)}\|^2 \geq \|\tilde{B}^{(t)} - \mathbf{G}^{(t+1)}\mathbf{Z}^{(t)}\|^2$. The steps of the PAGEC[1] algorithm that uses $\mathbf{S}$, which we will refer to as $\texttt{PAGEC}_\mathbf{S}$, are outlined in Algorithm 1. The convergence of $\texttt{PAGEC}_\mathbf{S}$ is guaranteed due to analytical solutions of $\mathbf{Q}$, $\mathbf{Z}$, $\mathbf{B}$ (*refitting step*) and *assignment step* carried out by $\mathbf{G}$. However, according to the initialization it will reach only a local optimum. We therefore started the algorithm several times and selected the best result minimizing the objective function (2).

Algorithm 1 : $\texttt{PAGEC}_\mathbf{S}$ algorithm

Input: $\mathbf{M}$ and $\mathbf{S}$ from structure matrix $\mathbf{W}$ and content matrix $\mathbf{X}$, k, p and λ ;
Initialize: $\mathbf{B}$, $\mathbf{Q}$ and $\mathbf{Z}$ with arbitrary orthonormal matrix;
repeat
(a) - Compute $\mathbf{G}$ using (8)
(b) - Compute $\mathbf{B}$ using (7)
(c) - Compute $\mathbf{Q}$ using (6)
(d) - Compute $\mathbf{Z}$ using (5)
until convergence
Output: $\mathbf{G}$: clustering matrix, $\mathbf{Z}$: rotation matrix, $\mathbf{B}$: node embedding matrix and $\mathbf{Q}$: attribute embedding matrix.

IV. Powered proximity matrix

Before assessing the Algorithm 1, we shall first present some theoretical reasons to explain why the proposed PAGEC model is able to outperform recent state-of-the-art methods proposed for the same purpose. Let us recall that the key elements when employing this model are, first, designing an affinity (or proximity) matrix that can jointly encode information from the structure $\mathbf{W}$ and attributes $\mathbf{X}$, and, secondly, embedding and clustering AN simultaneously so that there is a mutual reinforcement between $\mathbf{B}$ and $\mathbf{G}$.

With the PAGEC model, we use a powered proximity matrix to harness the benefits of a random walk process. The idea behind using $\mathbf{W}^p$ is to be able to explore the structure of $\mathbf{W}$ when a random walk includes multiple steps instead of just one. We know from the theory of Markov chains that $\mathbf{W}^p$ (where p is any positive integer) is obtained by multiplying $\mathbf{W}$ by itself p times, and consequently, if $\mathbf{W} = \mathbf{V}\Lambda\mathbf{V}^\top$, then we have $\mathbf{W}^p = \mathbf{V}\Lambda^p\mathbf{V}^\top$, where $\mathbf{V}$ is the matrix whose n^{th} column is $\mathbf{v}_n$. The eigensystem of $\mathbf{W}$ is therefore constituted by λ_n, $\mathbf{v}_n$, while the eigensystem of $\mathbf{W}^p$ is constituted by λ^p, $\mathbf{v}_n$.

In a prior work [24], it was noted that for many natural problems, $\mathbf{W}$ is an approximately block stochastic matrix, and hence the first k left eigenvectors of $\mathbf{W}$ are approximate piecewise constant over the k almost invariant subsets of rows. The iterative random walk process converges to the approximated data $\mathbf{W}^p$, where each row and each column moves towards its prototype. In other words, this process converges to an equilibrium (steady) state. The matrix $\mathbf{W}$ is composed of $k << n$ quasi-similar rows, where each row is represented by its prototype; see Figure 3.

[1]From now on, in order to distinguish between a model and its derived algorithm, we will use *typewriter font* for an algorithm. Consequently, PAGEC is the model and PAGEC its derived algorithm.

Let us consider $\mathbf{W}^p$, the p^{th} order transition matrix, as the affinity matrix. The cell $w_{m,n}$ in $\mathbf{W}^p$ gives the total probability that a random walk x_j beginning at m will end up in n after p steps, considering all possible paths between the nodes. We would expect this probability $w_{m,n}$ to be high if there is a good path between m and n, and low otherwise, the intended outcome being a block diagonal matrix suitable for clustering data [24]. However, in practice we usually observe that $\mathbf{W}^p$ behaves differently according to the value of p. If data points i, j are in the same cluster there are values of p for which the i^{th} and j^{th} rows of $\mathbf{W}^p$ become very similar. Consequently, if data points i, j are similar, then after a sufficient number of steps we can expect that particles that begin a random walk in each of them will have the same distribution for their locations after p steps. We also remark that by varying the number of steps p we explicitly explore similarities at different scales in the data, and as p increases we would expect to find a coarser structure. Note that, since $\mathbf{W}^p$ is a stochastic matrix (transition

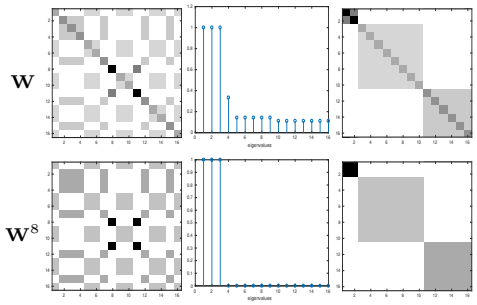

Fig. 3. Random walk process. Evolution of singular values and patterns according $\mathbf{W}$ and $\mathbf{W}^8$.

matrix) the construction of $\mathbf{M}$ as illustrated in Figure can be viewed as an iterative process $\mathbf{W}^p\mathbf{X}$; when $p = 0$ we have $\mathbf{M} = \mathbf{X}$. This process will converge to the approximated data $\mathbf{W}^p\mathbf{X}$ where each row moves towards its prototype. In other words, this process converges to an equilibrium state. With g denoting the number of eigenvalues of $\mathbf{W}^p$ equal to 1, the matrix $\mathbf{W}^p\mathbf{X}$ is composed of $g << n$ quasi-similar rows where each row is represented by its prototype (Figure 4).

At first sight this power iterative process appears to be of little interest, since it eventually leads to a data matrix in which rows coincide for any starting point. However, our practical experience shows that the data quickly give rise to row blocks and that these blocks move towards each others relatively slowly. If we stop the process at this point, the refined affinity matrix $\mathbf{M}$ can be useful for clustering. Thus, this process can be seen as a refining of the feature matrix $\mathbf{X}$ into a matrix structured into blocks, which is beneficial in relation to both the embedding and the clustering tasks.

V. Numerical experiments

Our focus in this work is on different clustering methods. Below we compare the PAGEC algorithm with some competitive methods, including recent deep learning methods.

A. Characteristics of datasets and Compared methods

The performances of clustering methods are evaluated using datasets commonly tested with ANE where the clusters are

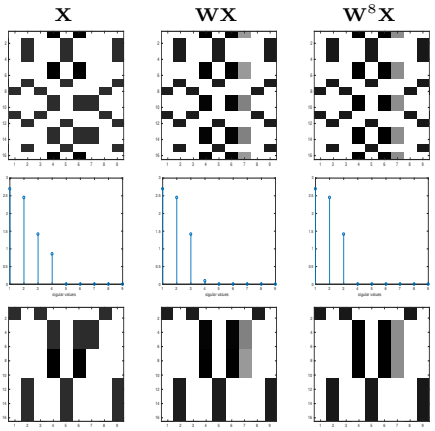

Fig. 4. Random walk process. Evolution of singular values and patterns according $\mathbf{X}$, $\mathbf{WX}$ and $\mathbf{W}^8\mathbf{X}$.

known. The experiments were conducted using four public citation network datasets, namely Citeseer, Cora, Wiki, and Pubmed, which contain a sparse bag-of-words feature vector for each document, together with a list of citation links between documents. Each document has a class label. For our purposes the documents are the nodes and the citation links are the edges. The characteristics of the datasets are summarized in Table I. The balance coefficient is defined as the ratio of the number of documents in the smallest class to the number of documents in the largest class, while *nz* denotes the percentage of sparsity.

TABLE I
CHARACTERISTICS OF DATASETS (#: THE CARDINALITY)

datasets	n	d	# Edges	#Classes	$nz(\%)$	Balance
Cora	2708	1433	5294	7	98.73	0.22
Citeseer	3312	3703	4732	6	99.14	0.35
Wiki	2405	4973	17981	17	86.46	0.02
Pubmed	19717	500	44338	3	89.98	0.52

We compare PAGEC with embedding-based methods and with other methods that are explicitly for graph clustering. In our comparison we include standard methods and also recent deep learning methods; these differ in the way they use available information. Some of them (such as K-means) use only $\mathbf{X}$ as the baseline, while others use more recent algorithms based on $\mathbf{X}$ and $\mathbf{W}$. All the compared methods are also mentioned in Section II above: Graph Encoder [13], DNGR [14], DeepWalk [15], Spectral Clustering [12], while spectral-f denotes a traditional spectral clustering applied on $\mathbf{W}_\mathbf{X}$, spectral-g is applied on $\mathbf{W}$. Using $\mathbf{X}$ and $\mathbf{W}$ we evaluated GAE and VGAE [16], MGAE [5], ARGA and ARVGA [8], AGC [17] and DAEGC [18].

B. Sensitivity analysis of λ and p

With the PAGEC model, the parameter λ controls the role of the second term $||\mathbf{S} - \mathbf{GZB}^\top||^2$ in (2). To measure its impact on the clustering performance of $\text{PAGEC}_\mathbf{S}$, we vary λ in $\{0, 10^{-6}, 10^{-3}, 10^{-1}, 10^0, 10^1, 10^3\}$ and p from 1 to 12. The performances in terms of *accuracy* (ACC), *Normalized Mutual Information* (NMI) and F1 measure are illustrated only on Cora dataset in Figure 5, due to lack of space; a high ACC, NMI or F1 corresponds to a better clustering result. First, we note that with $\lambda = 0$ we are relying only on $\min_{\mathbf{B},\mathbf{Q}} \left\|\mathbf{M} - \mathbf{BQ}^\top\right\|^2$ s.t. $\mathbf{B}^\top\mathbf{B} = \mathbf{I}$. In this case we observed

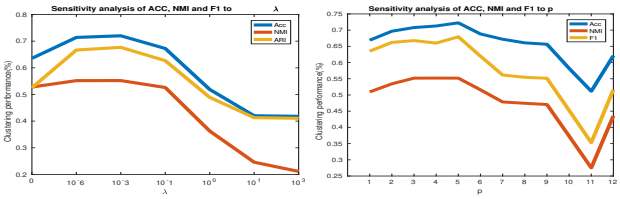

Fig. 5. Cora dataset: Sensitivity analysis according to (left) λ and (right) p.

poor results in terms of quality of clustering, which is an indication of the impact of the second term in (2). Quality increases as λ increases, and with only small values of λ a better performance is obtained on all datasets. We have noted that around 10^{-2} the clustering result becomes stable and less sensitive to λ. With values of λ greater than 10^{-1} the performance of PAGEC degrades sharply. This can be explained by the fact that the initialization of $\mathbf{G}$ is random, and thus can often be a long way from the real solution. As we can observe in Figure 5, the best results are obtained with $\lambda = 10^{-3}$ and $p = 5$.

C. Attributed network clustering

Evaluating clustering results is not a trivial task. Clustering accuracy is not always a reliable measure when clusters are not balanced and the number of clusters is high. As a better indication of the quality of our approach, below we have chosen to retain, in addition to accuracy, two measures that are widely used in assessing the quality of clustering, namely Normalized Mutual Information, and the F1 measure. The F1 measure takes both precision and recall into account in computing the score, F1 being the harmonic mean of the precision and recall. Intuitively, NMI quantifies to what extent the estimated clustering is informative about the true clustering, while F1 is more oriented towards measuring the effectiveness of a clustering algorithm. The higher the ACC/NMI/F1, the better the clustering, and so in our experiments clustering performance in relation to the true available clusters is assessed in terms of ACC, NMI and F1. If one clustering algorithm performs better than other clustering algorithms on a number of these measures, then we can have some confidence that it is truly the best clustering algorithm for the situation being evaluated.

We repeated the experiments 50 times and the averages (mean) and standard-deviations (sd) are reported in Table II; the best performance for each dataset is highlighted in bold. First, we observe the good performances of methods that integrate information from $\mathbf{W}$. The methods that include deep learning algorithms relying on $\mathbf{M}$ and $\mathbf{W}$ are better still. Regarding PAGEC, for the versions based respectively on $\mathbf{W}$ ($\text{PAGEC}_\mathbf{W}$) and on $\mathbf{S}$ ($\text{PAGEC}_\mathbf{S}$), we note good performances for all the datasets. In the case of $\text{PAGEC}_\mathbf{S}$ we remark the impact of $\mathbf{W}_\mathbf{X}$; it learns low-dimensional representations in agreement with the clustering structure.

D. Attributed network embedding

The PAGEC model, through $\mathbf{B}$, offers an embedding into clusters from which a 2d or 3d structure can also be observed.

TABLE II
CLUSTERING PERFORMANCES (ACC % , NMI % AND F1 %) ON CORA, CITESEER, WIKI AND PUBMED DATASETS. (-) REFERS TO THE NON-AVAILABILITY OF ACC IN [18].

Methods	Input	Datasets											
		Cora			Citeseer			Wiki			Pubmed		
		Acc	NMI	F1	Acc	NMI	F1	Acc	NMI	F1	Acc	NMI	F1
K-means	$\mathbf{X}$	34.65	16.73	25.42	38.49	17.02	30.47	33.37	30.20	24.51	57.32	29.12	57.37
Spectral-f	$\mathbf{W_X}$	36.26	15.09	25.64	46.23	21.19	33.70	41.28	43.99	25.20	59.91	32.55	58.61
Spectral-g	$\mathbf{W}$	34.19	19.49	30.17	25.91	11.84	29.48	23.58	19.28	17.21	39.74	3.46	51.97
DeepWalk	$\mathbf{W}$	46.74	31.75	38.06	36.15	09.66	26.70	38.46	32.38	25.74	61.86	16.71	47.06
DNGR	$\mathbf{W}$	49.24	37.29	37.29	32.59	18.02	44.19	37.58	35.85	25.38	45.35	15.38	17.90
GAE	$\mathbf{X}, \mathbf{W}$	53.25	40.69	41.97	41.26	18.34	29.13	17.33	11.93	15.35	64.08	22.97	49.26
VGAE	$\mathbf{X}, \mathbf{W}$	55.95	38.45	41.50	44.38	22.71	31.88	28.67	30.28	20.49	65.48	25.09	50.95
ARGE	$\mathbf{X}, \mathbf{W}$	64.0	44.90	61.90	57.3	35.0	54.60	41.40	39.502	38.27	59.12	23.17	58.41
ARVGE	$\mathbf{X}, \mathbf{W}$	63.8	45.0	62.70	54.4	26.1	52.90	41.55	40.01	37.80	58.22	20.62	23.04
MGAE	$\mathbf{X}, \mathbf{W}$	63.43	45.57	38.01	63.56	39.75	39.49	50.14	47.97	39.20	43.88	8.16	41.98
DAEGC	$\mathbf{X}, \mathbf{W}$	70.04	52.8	68.2	67.2	39.7	63.6	-	-	-	67.1	26.6	65.9
AGC	$\mathbf{X}, \mathbf{W}$	68.92	53.68	65.61	67.00	41.13	62.48	47.65	45.28	40.36	69.78	31.59	68.72
PAGEC$_{\mathbf{W},p=p*}$	$\mathbf{X}, \mathbf{W}$	**70.09**	**53.56**	**66.56**	**66.66**	**40.80**	**62.77**	**50.38**	**43.57**	**41.44**	**69.56**	**30.76**	**68.95**
	sd	.0040	*.0016*	*.0004*	*.0007*	*.00015*	*.0005*	*.0079*	*.0023*	*.0097*	*.00*	*.00*	*.00*
PAGEC$_{\mathbf{S},p=p*}$	$\mathbf{X}, \mathbf{S}$	**72.25**	**55.21**	**67.94**	**68.31**	**43.04**	**63.69**	**55.30**	**51.27**	**45.96**	**72.20**	**33.64**	**71.51**
	sd	.0010	*.0012*	*.0009*	*.0009*	*.0007*	*.0004*	*.0037*	*.0019*	*.0020*	*.00*	*.0028*	*.00*

To illustrate the quality of embedding, we consider the four attributed network datasets above and focus on the R-square= $\frac{\mathrm{Tr}(S_b)}{\mathrm{Tr}(S_t)}$ ratio, where S_b is the between-class scatter matrix and S_t is the total scatter matrix. To evaluate the separability of true classes, we computed this ratio from $\mathbf{X}$, $\mathbf{M}$ and $\mathbf{B}$ respectively (Table III).

TABLE III
EVALUATION OF SEPARABILITY BETWEEN CLASSES GIVEN THE TRUE PARTITION AND DATA REPRESENTATIONS $\mathbf{X}$, $\mathbf{M}$ AND $\mathbf{B}$.

	Rsquare in %		
datasets	$\mathbf{X}$	$\mathbf{M}$	$\mathbf{B}$
Cora	2.58	5.41	44.43
Citeseer	1.73	3.92	40.6
Wiki	5.73	17.69	34.04
Pubmed	1.99	2.02	31.14

VI. CONCLUSION AND PROSPECTS

In unsupervised learning, representation learning and clustering are generally studied by two distinct machine learning communities. In our contribution we argue that bringing the two disciplines together can improve representation learning. Behind the design of representation-learning algorithms of this kind is the all-encompassing quest for *Artificial Intelligence*.

This paper is concerned with both learning representation and clustering. We have proposed a novel matrix decomposition framework for simultaneous attributed network data embedding and clustering. Unlike existing methods that combine the objective function of ANE and the objective function of clustering separately, our proposed method PAGEC$_{\mathbf{S}}$ capitalizes on learning representation and clustering simultaneously.

The proposed framework suggests a number of prospects for further investigation. Following several KNN experiments relating to the choice of the number of neighbors, we have noted that given the sparsity a large number of neighbors (more than 30) is necessary. However, there are other points that warrant in-depth evaluation, such as the choice of λ and p.

In the future we are planning to improve upon our proposed method in several respects. First, we would like to be able to measure the impact of each matrix $\mathbf{W}$ and $\mathbf{W_X}$ in the construction of $\mathbf{S}$ by considering two different weights for $\mathbf{W}$ and $\mathbf{W_X}$ as follows: $\mathbf{S} = \alpha\mathbf{W} + \beta\mathbf{W_X}$. Secondly, we wish to address the problem of assessing the number of clusters, which remains a challenge in unsupervised learning, and specifically in relation to ANE.

REFERENCES

[1] G. Qi, C. C. Aggarwal, Q. Tian, H. Ji, and T. S. Huang, "Exploring context and content links in social media: A latent space method," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 5, pp. 850–862, 2012.

[2] H. Cai, V. W. Zheng, and K. C. Chang, "A comprehensive survey of graph embedding: Problems, techniques, and applications," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 9, pp. 1616–1637, 2018.

[3] W. Yu, W. Cheng, C. Aggarwal, B. Zong, H. Chen, and W. Wang, "Self-attentive attributed network embedding through adversarial learning," in *ICDM*, 2019, pp. 758–767.

[4] S. Chang, W. Han, J. Tang, G.-J. Qi, C. C. Aggarwal, and T. S. Huang, "Heterogeneous network embedding via deep architectures," in *KDD*, 2015, pp. 119–128.

[5] C. Wang, S. Pan, G. Long, X. Zhu, and J. Jiang, "Mgae: Marginalized graph autoencoder for graph clustering," in *CIKM*, 2017, pp. 889–898.

[6] Q. Dai, Q. Li, J. Tang, and D. Wang, "Adversarial network embedding," in *AAAI*, 2018, pp. 2167–2174.

[7] X. Huang, J. Li, and X. Hu, "Label informed attributed network embedding," in *WSDM*, 2017, pp. 731–739.

[8] S. Pan, R. Hu, G. Long, J. Jiang, L. Yao, and C. Zhang, "Adversarially regularized graph autoencoder for graph embedding," in *IJCAI*, 2018, pp. 2609–2615.

[9] W. Yu, C. Zheng, W. Cheng, C. C. Aggarwal, D. Song, B. Zong, H. Chen, and W. Wang, "Learning deep network representations with adversarially regularized autoencoders," in *SIGKDD*, 2018, pp. 2663–2671.

[10] W. Yu, W. Cheng, C. C. Aggarwal, K. Zhang, H. Chen, and W. Wang, "Netwalk: A flexible deep embedding approach for anomaly detection in dynamic networks," in *SIGKDD*, 2018, pp. 2672–2681.

[11] T. Guo, S. Pan, X. Zhu, and C. Zhang, "Cfond: consensus factorization for co-clustering networked data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 4, pp. 706–719, 2018.

[12] L. Tang and H. Liu, "Leveraging social media networks for classification," *Data Mining and Knowledge Discovery*, vol. 23, no. 3, pp. 447–478, 2011.

[13] F. Tian, B. Gao, Q. Cui, E. Chen, and T.-Y. Liu, "Learning deep representations for graph clustering," in *AAAI*, 2014.

[14] S. Cao, W. Lu, and Q. Xu, "Deep neural networks for learning graph representations," in *AAAI*, 2016, pp. 1145–1152.

[15] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *SIGKDD*, 2014, pp. 701–710.

[16] T. N. Kipf and M. Welling, "Variational graph auto-encoders," *NIPS Workshop on Bayesian Deep Learning*, 2016.

[17] X. Zhang, H. Liu, Q. Li, and X.-M. Wu, "Attributed graph clustering via adaptive graph convolution," *arXiv preprint arXiv:1906.01210*, 2019.

[18] C. Wang, S. Pan, R. Hu, G. Long, J. Jiang, and C. Zhang, "Attributed graph clustering: A deep attentional embedding approach," *arXiv preprint arXiv:1906.06532*, 2019.

[19] M. Yamamoto and H. Hwang, "A general formulation of cluster analysis with dimension reduction and subspace separation," *Behaviormetrika*, vol. 41, no. 1, pp. 115–129, 2014.

[20] K. Allab, L. Labiod, and M. Nadif, "Simultaneous spectral data embedding and clustering," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 12, pp. 6396–6401, 2018.
[21] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT Press, 2016, vol. 1.
[22] J. M. ten Berge, *Least squares optimization in multivariate analysis*. DSWO Press, Leiden University Leiden, 1993.
[23] P. H. Schönemann, "A generalized solution of the orthogonal procrustes problem," *Psychometrika*, vol. 31, no. 1, pp. 1–10, 1966.
[24] M. Meila and J. Shi, "Learning segmentation by random walks," in *Advances in neural information processing systems*, 2001, pp. 873–879.