# Boosting Attributed Network Embeddings with Clustering

Lazhar Labiod and Mohamed Nadif

Centre Borelli UMR 9010, Université Paris Cité, France
lastname.firstname@u-paris.fr

**Abstract.** Clustering is a fundamental task in machine learning, widely applied across various domains. This paper focuses on the clustering of attributed networks, which combine structural and attribute information. We introduce a novel model that unifies regularized data embedding and clustering, enhancing the representation and analysis of such networks. Our approach not only improves clustering performance for attributed network data but also demonstrates effectiveness in scenarios where the graph structure is not initially available. Through experimentation on benchmark datasets, we show that our method achieves superior performance in terms of key clustering external metrics. Furthermore, it provides relevant embeddings that simplify the identification of classes.

**Keywords:** attributed networks, embedding, clustering

## 1 Introduction

In data science, data embedding (DE) is commonly used for the purposes of visualizing, but it can also play a significant role in clustering, where the aim is to divide a dataset into homogeneous clusters. Working in a low-dimensional space can be beneficial for data partitioning, and various approaches are documented in the literature; see, for example, [6, 17, 1, 2, 9].

In this paper we focus on *Attributed Networks* (AN) [13] which have been used to model real-world networks, including academic and health care networks. These networks offer node links and attributes for analysis, unlike plain networks with only node links. In AN, each node is linked to a set of features, leading to two matrices. The first is a square matrix $\mathbf{W}$ of size $n \times n$; $\mathbf{W}$ is constructed from a graph represented by an adjacency matrix $\mathbf{A}$. The second is $\mathbf{X}$ of size $n \times d$, the graph feature matrix, where each of the $n$ nodes is described by $d$ features.

Recently, representation learning has become important in fields like social and academic networks, and protein interactions. ANE [3] seeks to create a compact node representation that preserves network topology and node proximity by attributes. While NE [18] has fostered several methods [4], ANE has been less explored. Unlike NE, ANE integrates nodes' proximity and attribute similarity, which distinguishes it from existing NE algorithms. Learned representations are beneficial for tasks like network clustering [16, 10, 11], node visualization [5], node classification [8], and link prediction [12]. Consequently, tackling high-dimensionality, sparsity, and nonlinearity is now

a critical research focus. However, these challenges are particularly pronounced in network clustering techniques. Often, clustering techniques disappoint for two reasons: the continuous embedding solution usually diverges from precise discrete clustering, and information loss occurs between continuous embedding generation and discretization stages.

This paper introduces an objective function integrating embedding and clustering, unlike the separate considerations of $\mathbf{X}$ and $\mathbf{W}$. Our algorithm, based on this function, employs low-rank subspace and clustering to better capture complex relationships between $\mathbf{X}$ and $\mathbf{W}$, enhancing clustering robustness. The paper's key contributions are as follows. First, we introduce a new ANE approach that integrates regularized data embedding through clustering into a unified framework, utilizing information from $\mathbf{X}$ and $\mathbf{W}$. Second, we demonstrate that our contribution also improves clustering in situations where $\mathbf{W}$ is not available.

The remainder of this paper is organized as follows. We first introduce the data into the ANs and then present our proposed method, detail the algorithm, and provide evaluations of our proposal (Section 2). In Section 3, we evaluate our approach in terms of clustering and embedding quality. Finally, we conclude with a summary and discuss future perspectives (Section 4).

## 2 Joint Embedding and clustering

An attributed network $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ consists of the set of nodes $\mathcal{V}$, 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 content information $\mathbf{X} \in \mathbb{R}^{n \times d}$ and 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. From $\mathbf{X}$ and $\mathbf{A}$, we propose to derive two essential matrices, $\mathbf{M}$ based on structural and attribute information, and $\mathbf{S}$ based on attribute information.

### 2.1 Construction of $\mathbf{M}$

From $\mathbf{A}$, we model the proximity of the nodes using 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 $d_{ii} = \sum_{i'=1}^{n} a_{i'i}$. This leads to propose the following construction of $\mathbf{M}$ such as

$$\mathbf{M} = \mathbf{W}\mathbf{X}.$$

Thus, with this type of multiplicative smoothing, the original data is transformed into a set of representative prototypes.

### 2.2 Construction of $\mathbf{S}$

To utilize additional information about node similarity from $\mathbf{X}$, we first preprocess the above dataset $\mathbf{X}$ to produce input from a similarity graph $\mathbf{S}$ of size $(n \times n)$. The

construction of a similarity matrix using a K-Nearest-Neighbor (KNN) graph involves calculating pairwise distances between data points, identifying each point's $K$ nearest neighbors, and forming a graph where edges connect neighboring points. The similarity matrix $\mathbf{S} = (s_{ij})$ is defined such that $s_{ij} = \exp\left(-\frac{d(x_i,x_j)^2}{2\sigma^2}\right)$. Here, $\sigma$ serves as a parameter that regulates the kernel's width. This equation is applied when $\mathbf{x}_j$ is identified as a neighbor of $\mathbf{x}_i$; if not, $s_{ij}$ is set to 0. This matrix is designed to capture local relationships, making it a valuable component in our proposal.

### 2.3 Objective function

In the following, we aim to combine the information from $\mathbf{M}$ and $\mathbf{S}$ to enhance the quality of the embedding described by $\mathbf{B}$. To do this, we will perform two approximations of $\mathbf{M}$ and $\mathbf{S}$, both sharing $\mathbf{B}$ in common. 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 method seeks to obtain the maximally informative embedding with respect to the clustering structure in the attributed network data. Given $\mathbf{Z} \in \mathbb{R}^{k\times k}$, $\mathbf{B} \in \mathbb{R}^{n\times k}$, $\mathbf{Q} \in \mathbb{R}^{d\times k}$ and $\mathbf{G} \in \{0,1\}^{n\times k}$, the proposed objective function $\mathcal{F}(\mathbf{B},\mathbf{Z},\mathbf{Q},\mathbf{G})$ to be optimized is consequently given by

$$\Psi(\mathbf{M}, \mathbf{B}\mathbf{Q}^\top) + \lambda\Psi(\mathbf{S}, \mathbf{G}\mathbf{Z}\mathbf{B}^\top) \tag{1}$$

where $\Psi$ denotes the deviation between $\mathbf{M}$ and $\mathbf{B}\mathbf{Q}^\top$, $\mathbf{B}$ plays the role of the embedding matrix for nodes and $\mathbf{Q}$ the embedding matrix for the attributes, and between $\mathbf{S}$ and $\mathbf{G}\mathbf{Z}\mathbf{B}^\top$, where $\mathbf{G}$ plays the role of memberships in clusters and $\mathbf{Z}$ is an orthonormal rotation matrix which most closely maps $\mathbf{B}$ to $\mathbf{G}$. In other words, we choose to regularize the approximation of $\mathbf{M}$ by a term seeking to approximate $\mathbf{S}$ while taking into account the structure into classes; $\lambda$ is regularized parameter. Note that if $\lambda = 0$, it is possible to consider only the approximation of $\mathbf{M}$ by $\mathbf{B}\mathbf{Q}^\top$. In the following, $\Psi$ denotes the frobenius norm and (1) becomes

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

### 2.4 Optimization

To infer the latent factor matrices $\mathbf{Z}$, $\mathbf{B}$, $\mathbf{Q}$ and $\mathbf{G}$ from $\mathbf{M} = \mathbf{W}\mathbf{X}$ and $\mathbf{S}$, we derive an alternating optimization algorithm. Based on $\mathbf{B}^\top\mathbf{B} = \mathbf{I}$ and $\mathrm{Tr}(AB) = \mathrm{Tr}(BA)$ where Tr denotes the trace of a matrix, it is easy to show that

$$\left\|\mathbf{S} - \mathbf{G}\mathbf{Z}\mathbf{B}^\top\right\|^2 = \left\|\mathbf{S} - \mathbf{S}\mathbf{B}\mathbf{B}^\top\right\|^2 + \|\mathbf{S}\mathbf{B} - \mathbf{G}\mathbf{Z}\|^2. \tag{3}$$

The following outlines the various steps required to infer $\mathbf{Z}$, $\mathbf{Q}$, $\mathbf{B}$ and $\mathbf{G}$, ensuring guaranteed convergence of our algorithm.

**Compute $\mathbf{Z}$.** By fixing $\mathbf{G}$ and $\mathbf{B}$ we reduce the problem that arises in (2) to minimize the second term. Thus, from (3), we have

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

This can be reduced to $\max_{\mathbf{Z}}\ \mathrm{Tr}(\mathbf{G}^\top\mathbf{SBZ})$ s.t. $\mathbf{Z}^\top\mathbf{Z} = \mathbf{I}$. It was shown in [15], with $\mathbf{U}\Sigma\mathbf{V}^\top$ the SVD for $\mathbf{G}^\top\mathbf{SB}$, that

$$\mathbf{Z} = \mathbf{U}\mathbf{V}^\top. \tag{5}$$

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

**Compute 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

$$\mathbf{Q} = \mathbf{M}^\top\mathbf{B}. \tag{6}$$

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

**Compute B**. Given $\mathbf{G}$, $\mathbf{Q}$ and $\mathbf{Z}$, (2) is equivalent to $\max_{\mathbf{B}}\ \mathrm{Tr}((\mathbf{M}^\top\mathbf{Q}+\lambda\mathbf{SGZ})\mathbf{B}^\top)$ 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 $(\mathbf{M}^\top\mathbf{Q}+\lambda\mathbf{SGZ})$, and we get

$$\mathbf{B} = \hat{\mathbf{U}}\hat{\mathbf{V}}^\top. \tag{7}$$

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 G**: Finally, given $\mathbf{B}$, $\mathbf{Q}$ and $\mathbf{Z}$, the problem (2) is equivalent to $\min_{\mathbf{G}} \|\mathbf{SB} - \mathbf{GZ}\|^2$ since from (2) and (3) $\mathbf{G}$ is present only in $\|\mathbf{SB} - \mathbf{GZ}\|$. 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 $\mathbf{Q}$, $\mathbf{Z} = (\mathbf{z}_1^\top, \ldots, \mathbf{z}_k^\top)^\top$ and $\mathbf{B}$. Let $\tilde{\mathbf{B}} = \mathbf{SB} = (\tilde{\mathbf{b}}_1^\top, \ldots, \tilde{\mathbf{b}}_n^\top)^\top$ we then compute

$$g_{is} = \begin{cases} 1 \text{ if } s = \mathtt{arg}\min_{s'} ||\tilde{\mathbf{b}}_i - \mathbf{z}_{k'}||^2 \\ 0 \text{ otherwise.} \end{cases} \tag{8}$$

Thus, at the (t+1)th iteration, this leads to $\|\tilde{\mathbf{B}}^{(t)} - \mathbf{G}^{(t)}\mathbf{Z}^{(t)}\|^2 \geq \|\tilde{\mathbf{B}}^{(t)} - \mathbf{G}^{(t+1)}\mathbf{Z}^{(t)}\|^2$.

The steps of the proposed algorithm are outlined in Algorithm 1 and called . The convergence 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).

### 2.5 Assessing of $\lambda$

To evaluate clustering quality, internal validity criteria are often used to rank solutions, known as relative validity criteria. We propose using the Silhouette Width Criterion (SWC) [14] to estimate hyperparameters $\lambda$. SWC measures how well an object fits its own cluster (cohesion) versus others (separation). It ranges from $-1$ to 1, with higher values indicating better fit within its cluster. For different $\lambda$ values, we run `ANclust` and select the pair maximizing SWC. This version is referred to as `ANclust`$^*$.

**Algorithm 1** `ANclust`

**Input:** $\mathbf{M}$ and $\mathbf{S}$ from structure matrix $\mathbf{W}$ and content matrix $\mathbf{X}$, $k$ and $\lambda$ ;
**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.

**Algorithm 2** `ANclust`$^*$

**Input:** $\mathbf{M}$ and $\mathbf{S}$ from structure matrix $\mathbf{W}$ and content matrix $\mathbf{X}$, $k$
**for** $\lambda \in \{0, 10^{-6}, 10^{-3}, 10^{-1}, 10^{0}, 10^{1}, 10^{3}\}$ **do**
(a) - Run `ANclust`
(b) - Compute SWC
**end for**
$\lambda^* = \max_\lambda SWC$
**Output:** $\lambda^*$

## 3 Evaluation

In our experiments, we initially investigate attributed network datasets in terms of embedding and clustering. Additionally, we explore scenarios where $\mathbf{W}$ is unavailable, showcasing how our approach, developed within the framework of attributed graph frameworks, can significantly improve clustering.

**Clustering of attributed network datasets** We propose evaluating Algorithm 2 on the following commonly used datasets.

- BlogCatalog is a dataset of a blog community social network, which contains 5,196 users as nodes, 171,743 edges indicating the user interactions, and 8,189 attribute categories denoting the keywords of their blogs. Users could register their blogs into six different predefined classes, which are set as labels.
- Flickr is a benchmark attributed social network dataset containing 7,575 nodes. Each node is a Flickr user and each attribute category is a tag related to the photos shared by users. There are 239,738 undirected edges in this network, which indicate the following relationships among users. The nine groups that users have joined are considered as target labels.
- CiteSeer is a co-authorship graph based on 3,312 papers. In this graph, nodes represent authors, and an edge connects two authors if they have co-authored a paper. Node features correspond to the paper keywords from each author's publications, and class labels indicate the most active fields of study for each author.

As Baselines, we only select methods which can simultaneously utilize two types of information as baselines. These baselines have been compared in recent paper [7] where

the authors propose the SSAGCN method via Autoencoder-style self-supervised learning. In table 2 we note that our algorithm achieves remarkable performance in terms of NMI (Normliazed Mutual Information) and ARI (Adjusted Rand Index), two commonly used external metrics for evaluating clustering algorithms. Furthermore, Fig. 1 illustrates the quality of the embedding with $\mathbf{B}$ obtained by our approach compared to that with $\mathbf{X}$ or $\mathbf{M}$.

Table 1: Clustering performances of ANclust$^*$

| Input | **Blogcatalog** | | **Flickr** | | **Citeseer** | |
|---|---|---|---|---|---|---|
| | NMI | ARI | NMI | ARI | NMI | ARI |
| GUCD | 30.2 | 23.1 | 27.1 | 20.9 | 27.4 | 23.3 |
| SDCN | 31.1 | 21.6 | 35.9 | 22.3 | 38.7 | 40.2 |
| DAEGC | 27.7 | 20.2 | 30.8 | 21.5 | 39.7 | 37.8 |
| AdaMRF | 28.9 | 22.3 | 25.3 | 20.6 | 28.8 | 27.3 |
| DGTA | 25.6 | 22.1 | 23.9 | 22.7 | 21.7 | 11.9 |
| SCI | 30.2 | 21.3 | 35.7 | 23.1 | 38.3 | 31.2 |
| ANEM | 30.8 | 21.2 | 31.3 | 21.1 | 28.1 | 26.2 |
| SSAGCN | 39.3 | 25.9 | 40.3 | 27.5 | 39.1 | 37.9 |
| ANclust$^*$ | **68.2** | **70.8** | **63.9** | **49.2** | **40.6** | **41.7** |

**Document clustering without $\mathbf{W}$** In the domain of *natural language processing* (NLP), unsupervised learning is prevalent. When dealing with unlabeled datasets, techniques like clustering and visualization can enhance the usefulness of textual data. The primary obstacle in document clustering is often how to represent the documents, with common methods including Bag-Of-Words (BOW). LLMs process text contextually, mimicking human understanding of nuances. This enables systems to grasp complex concepts and adapt responses. The MiniLM model, using knowledge distillation, creates efficient versions of larger models like BERT.

To evaluate our algorithm we consider three coprus Classic4, NG20 and BBC. These collections are often used as benchmarks for evaluating text classification and clustering techniques. BBC News [1] is a dataset sourced from the BBC News, encompasses a collection of 2225 articles labeled across five categories: business, entertainment, politics, sport, and tech. NG20 Newsgroups[2]: consists of 18,846 newsgroup documents, distributed across 20 different topical newsgroups. Originating from Usenet newsgroups in the late 1990s, it encompasses a broad spectrum of themes such as politics, religion, and science. The Classic4 dataset is a well-known collection used in the field of text mining. It is a collection of 7095 articles and consists of four distinct document sets: CACM, CISI, CRAN, and MED. According the MiniLM model we obtain the three matrices Classic4 ($7095\times384$), NG20 ($18846\times384$) and BBC ($2225\times384$). Without $\mathbf{W}$, we construct a K-Nearest-Neighbor graph (KNN with $K = 15$ and $\sigma = 1$) to create the similarity graph $\mathbf{S}$ of size $(n \times n)$, simplifying the objective function to (2) with $\mathbf{M} = \mathbf{SX}$. The parameter $K$ can, of course, be adjusted according to the data; we made this choice based on experience across all datasets. Table 2 clearly highlights the benefits of introducing $\mathbf{S}$ in this manner.

[1] http://mlg.ucd.ie/datasets/bbc.html

[2] http://qwone.com/ jason/20Newsgroups/

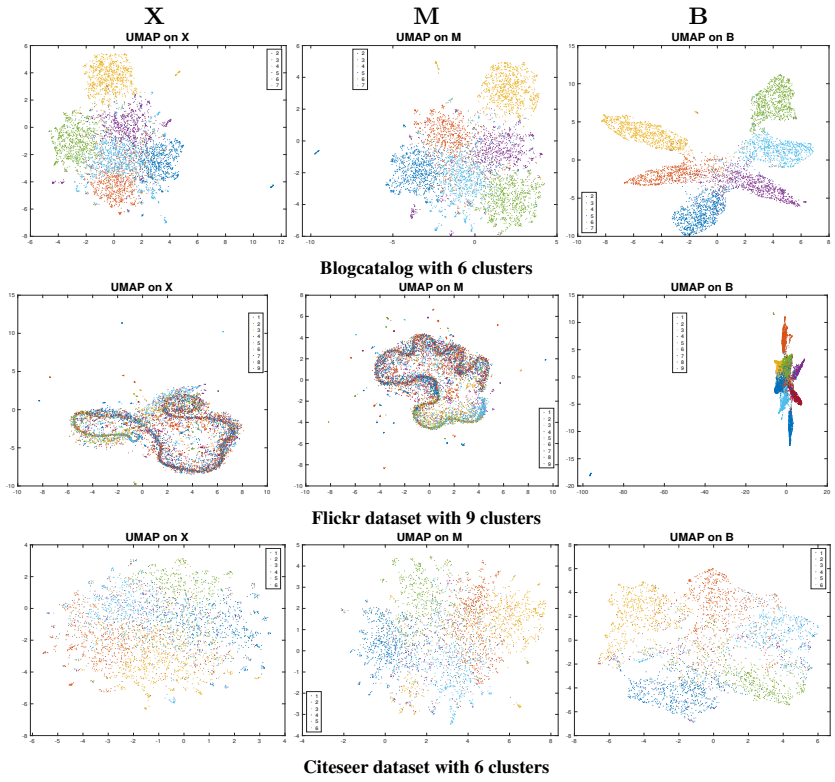


Fig. 1: From top to bottom and from left to right, clusters projection using UMAP applied on $\mathbf{X}$, $\mathbf{M} = \mathbf{SX}$ and $\mathbf{B}$.

Table 2: Clustering performances of `Algorithm 2`.

| Input | Classic4 | | | NG20 | | | BBC | | |
|---|---|---|---|---|---|---|---|---|---|
| | ACC | NMI | ARI | ACC | NMI | ARI | ACC | NMI | ARI |
| $\mathbf{X}$ | 76.25 | 61.07 | 45.10 | 48.40 | 44.42 | 32.63 | 92.63 | 80.20 | 82.97 |
| $\mathbf{M}, \mathbf{S}$ | **94.90** | **85.35** | **85.24** | **56.73** | **54.65** | **37.46** | **95.42** | **86.50** | **89.09** |

## 4 Conclusion

This work presents an innovative approach to enhance the embedding and clustering of attributed networks. By integrating regularized data embedding and clustering into a unified framework, we have demonstrated that our method is beneficial not only for attributed network data but also for scenarios where the graph structure needs to be constructed. Experimental results on various datasets show remarkable performance and quality of embedding, highlighting the effectiveness of our algorithm compared to existing methods. This approach opens new perspectives for processing complex and heterogeneous data, providing robust solutions for diverse applications in network analysis.

## References


[1] Kais Allab, Lazhar Labiod, and Mohamed Nadif. A semi-nmf-pca unified framework for data clustering. *IEEE TKDE*, 29(1):2–16, 2016.

[2] Kais Allab, Lazhar Labiod, and Mohamed Nadif. Simultaneous spectral data embedding and clustering. *IEEE TNNLS*, 29(12):6396–6401, 2018.

[3] HongYun Cai, Vincent W. Zheng, and Kevin Chen-Chuan Chang. A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE TKDE*, 30(9):1616–1637, 2018.

[4] Shiyu Chang, Wei Han, Jiliang Tang, Guo-Jun Qi, Charu C. Aggarwal, and Thomas S. Huang. Heterogeneous network embedding via deep architectures. In *KDD*, pages 119–128, 2015.

[5] Quanyu Dai, Qiang Li, Jian Tang, and Dan Wang. Adversarial network embedding. In *AAAI*, pages 2167–2174, 2018.

[6] S.A. Gattone and R. Rocci. Clustering curves on a reduced subspace. *Journal of Computational and Graphical Statistics*, 21(2):361–379, 2012.

[7] Chaobo He, Junwei Cheng, Guohua Chen, and Yong Tang. Multiple topics community detection in attributed networks. In *SIGIR*, pages 2199–2203, 2023.

[8] Xiao Huang, Jundong Li, and Xia Hu. Label informed attributed network embedding. In *WSDM*, pages 731–739, 2017.

[9] Lazhar Labiod and Mohamed Nadif. Efficient regularized spectral data embedding. *Advances in Data Analysis and Classification*, 15(1):99–119, 2021.

[10] Lazhar Labiod and Mohamed Nadif. Power attributed graph embedding and clustering. *IEEE TNNLS*, 35(1):1439–1444, 2024.

[11] Lazhar Labiod and Mohamed Nadif. Unsupervised learning from attributed networks. *Advances in Data Analysis and Classification*, pages 1–30, 2025.

[12] Shirui Pan, Ruiqi Hu, Guodong Long, Jing Jiang, Lina Yao, and Chengqi Zhang. Adversarially regularized graph autoencoder for graph embedding. In *IJCAI*, pages 2609–2615, 2018.

[13] Guo-Jun Qi, Charu C. Aggarwal, Qi Tian, Heng Ji, and Thomas S. Huang. Exploring context and content links in social media: A latent space method. *IEEE TPAMI*, 34(5):850–862, 2012.

[14] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[15] Jos MF ten Berge. *Least squares optimization in multivariate analysis*. DSWO Press, Leiden University Leiden, 1993.

[16] Chun Wang, Shirui Pan, Guodong Long, Xingquan Zhu, and Jing Jiang. Mgae: Marginalized graph autoencoder for graph clustering. In *CIKM*, pages 889–898, 2017.

[17] Michio Yamamoto and Heungsun Hwang. A general formulation of cluster analysis with dimension reduction and subspace separation. *Behaviormetrika*, 41(1): 115–129, 2014.

[18] Wenchao Yu, Wei Cheng, Charu Aggarwal, Bo Zong, Haifeng Chen, and Wei Wang. Self-attentive attributed network embedding through adversarial learning. In *ICDM*, pages 758–767, 2019.