Multi-View Attribute Graph Convolution Networks for Clustering

Jiafeng Cheng[1] , **Qianqian Wang**[1*] , **Zhiqiang Tao**[2] , **Deyan Xie**[1] and **Quanxue Gao**[1,3†]
[1]State Key Laboratory of Integrated Services Networks, Xidian University
[2]Northeastern University
[3]Unmanned System Research Institue, Northwestern Polytechnical university
xd.jiafengcheng@gmail.com, qianqian174@foxmail.com, {zqtaomail, weisong1220}@gmail.com,
qxgao@xidian.edu.cn

Abstract

Graph neural networks (GNNs) have made considerable achievements in processing graph-structured data. However, existing methods cannot allocate learnable weights to different nodes in the neighborhood and lack of robustness on account of neglecting both node attributes and graph reconstruction. Moreover, most of multi-view GNNs mainly focus on the case of multiple graphs, while designing GNNs for solving graph-structured data of multi-view attributes is still under-explored. In this paper, we propose a novel Multi-View Attribute Graph Convolution Networks (MAGCN) model for the clustering task. MAGCN is designed with two-pathway encoders that map graph embedding features and learn view-consistency information. Specifically, the first pathway develops multi-view attribute graph attention networks to reduce the noise/redundancy and learn the graph embedding features of multi-view graph data. The second pathway develops consistent embedding encoders to capture the geometric relationship and the consistency of probability distribution among different views, which adaptively finds a consistent clustering embedding space for multi-view attributes. Experiments on three benchmark graph databases show the effectiveness of our method compared with several state-of-the-art algorithms.

1 Introduction

Multi-view clustering is a fundamental task in machine learning. It aims to integrate multiple features and discover consistent information among different views [Xie *et al.*, 2019; Li *et al.*, 2019; Zhang *et al.*, 2018a]. Existing multi-view clustering methods have achieved considerable results in the Euclidean domains [Andrew *et al.*, 2013; Gao *et al.*, 2020]. However, those algorithms are no longer suitable for processing intensively studied data, which often occurs in the non-Euclidean domains such as graphs in social network connections, article citations, *etc*. In light of this, graph embedding methods, which could effectively explore the graph structured data, have received much attention recently.

Graph embedding converts graph data into a low-dimensional, compact, and continuous feature space, which is usually implemented by matrix-factorization [Belkin and Niyogi, 2002], random-walk [Perozzi *et al.*, 2014], or graph neural networks (GNN) [Salehi and Davulcu, 2019; Kipf and Welling, 2017]. Among existing methods, GNNs, largely owing to their efficiency and inductive learning capability [Hamilton *et al.*, 2017], have become one of the most popular paragons. Generally, GNNs calculate the embedding of a graph node by applying multiple graph convolutional layers to gather the information of node neighbors through nonlinear transformations and aggregation functions. Hence, they can preserve the topological structure, vertex content information of the graph-structured data. Although the above GNNs can effectively process single-view graph data, they are not applicable to multi-view graph data.

To tackle this challenge, some research attempts are made in using GNNs for multi-view data of multiple graph structures. For example, multi-view graphs are used for prediction and classification of drug similarity and medicine in the medical filed [Zhang *et al.*, 2018b; Ma *et al.*, 2018], and are also leveraged for ridehailing demand forecasting and global poverty [Geng *et al.*, 2019; Khan and Blumenstock, 2019]. However, those multi-view graph neural networks have the following limitations: *1)* They cannot allocate learnable specifying different weights to different nodes in neighborhood; *2)* They may neglect to proceed the reconstruction of both node attributes and graph structure to improve the robustness; *3)* The similarity distance measure is not explicitly considered for the consistency relationship among different views. Moreover, existing multi-view GNN methods mainly focus on the case of multiple graphs, while ignoring equally important attribute diversity, *i.e.* multi-attribute. Nonetheless, in the real world, it is common that we have multiple characteristic attributes with the same connection relationship of a graph. For instance, people could have multiple attributes, such as, job, hobby, *etc.*, in the connection graph of social network.

Motivated by the above observations, in this paper, we propose a novel Multi-view Attribute Graph Convolution Networks for clustering (MAGCN) the graph-structured data of multi-view attributes (see Fig.2). Specifically, 1) to allocate learnable weights to different nodes, MAGCN devel-

[*] Corresponding Author

[†] Corresponding Author

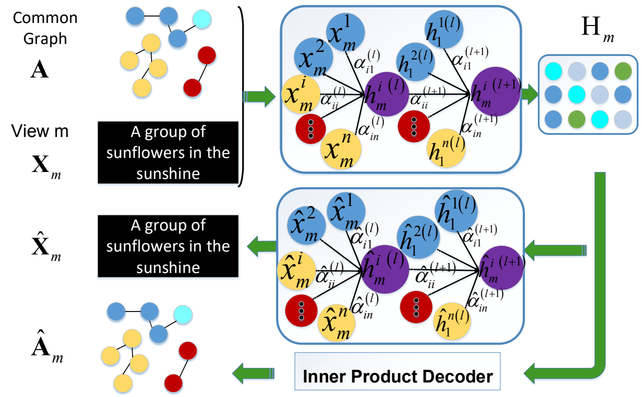

Figure 1: Part of Multi-view Attribute Graph Convolution Encoder for view m.

ops multi-view attribute graph convolution encoders with attention mechanism for learning graph embedding from multi-view graph data. 2) Attribute and graph reconstruction are both computed by the graph convolution decoders of MAGCN. 3) The geometric relationship and the probability distribution consistency among multi-view graph data are incorporated into the consistent embedding encoders of MAGCN to further facilitate the clustering task. The key contributions of our work are summarized as follows.

- We propose a novel Multi-View Attribute Graph Convolution Networks for clustering on the graph-structured data of multi-view attributes.
- We develop multi-view attribute graph convolution encoders with attention mechanism to reduce the noise/redundancy of the multi-view graph data. In addition, reconstruction of both node attributes and graph structure are considered to improve the robustness.
- Consistent embedding encoders are designed to extract the consistency information among multiple views, by exploring the geometric relationship and the probability distribution consistency of different views.

2 Related Work

Learning graph node embedding within broader graph structure is crucial for many tasks on graphs. Existing GNNs models in processing graph-structured data belong to a set of graph message-passing architectures that use different aggregation schemes for a node to aggregate feature messages from its neighbors in the graph. Graph Convolutional Networks [Kipf and Welling, 2017] scale linearly in the number of graph edges and learn hidden layer representations that encode both local graph structure and features of nodes. By stacking self-attention layers in which nodes are able to attend over their neighborhoods' features, Graph Attention Networks (GAT) [Velickovic *et al.*, 2018] enable specifying different weights to different nodes in neighborhood. GraphSAGE [Hamilton *et al.*, 2017] concatenates the node's feature with diversified pooled neighborhood information and effectively trades off performance and runtime by sampling node neighborhoods. Message Passing Neural Networks [Gilmer *et al.*, 2017] further incorporate edge information when doing the aggregation.

To handle the problem of multi-view graph node embedding, some researchers have made some attempts. In the medical field, [Zhang *et al.*, 2018b] propose a method based on GCNs for fusing multiple modalities of brain images in relationship prediction, which is useful for distinguishing Parkinson's Disease cases (a prevalent neurodegenerative disease) from controls. Another novel model [Geng *et al.*, 2019], called spatiotemporal multi-graph convolution network, encodes the non-Euclidean correlations among regions using multiple graphs and explicitly captures them using multi-graph convolution encoder. In application accounting for social networks, Multi-GCN [Khan and Blumenstock, 2019] incorporates non-redundant information from multiple views into the learning process. [Ma *et al.*, 2018] utilize multi-view graph auto-encoder, which integrates heterogeneous, noisy, nonlinear-related information to learn accurate similarity measures especially when labels are scarce.

However, those multi-view GNNs cannot allocate learnable specifying different weights to different nodes in neighborhood, where we can learn from the excellent ideas of GAT. And the clustering performance of them is limited as they do not consider the structure and distribution consistency for clustering embedding. What's more, GNNs for solving graph-structured data of multi-view attributes is still underexplored. Existing multi-view GNNs mainly focus on the case of multiple graphs and neglect the equally important attribute diversity. Thus, in this paper, we propose MAGCN on multi-view attribute graph data.

3 Proposed Methodology

3.1 Notation

A graph can be represented as $\mathbf{G} = (\mathbf{V}, \mathbf{E})(\mathbf{G} \in \mathbb{R}^{n \times n})$, where $\mathbf{V} = \{v_1, v_2, ..., v_n\}$ is the node set and $\mathbf{E}$ is the edge set, n denotes the number of nodes, and v_i represents the i-th node. In this study, we augment graph $\mathbf{G}$ with the node m-th view attribute feature $\mathbf{X}_m = \{x_m^1, ..., x_m^i, ..., x_m^n\}(\mathbf{X}_m \in \mathbb{R}^{n \times d_m})$, $m = 1, 2, ..., M$, where x_m^i refers to the feature vector associated with node v_i and M is the number of views.

3.2 The Framework of MAGCN

As shown in Fig.2, our model contains two principle modules: multi-view attribute graph convolution Encoders and consistent embedding Encoders. We first encode multi-view graph data $\mathbf{X}_m$ into graph embedding $\mathbf{H}_m = \{h_m^1, ..., h_m^i, ..., h_m^n\}(\mathbf{H}_m \in \mathbb{R}^{n \times d})$ by multi-view attribute graph convolution encoders. Then, fed $\mathbf{H}_m$ into consistent embedding encoders and obtain a consistent clustering embedding $\mathbf{Z}$. The clustering process is eventually conducted on the ideal embedding intrinsic description space which is computed by $\mathbf{Z}$.

Multi-view Attribute Graph Convolution Encoder

In multi-view attribute graph convolution encoders, the first pathway encoders map multi-view node attribute matrix and graph structure into graph embedding space. Specifically, for the m-th view, the function of a graph embedding model is

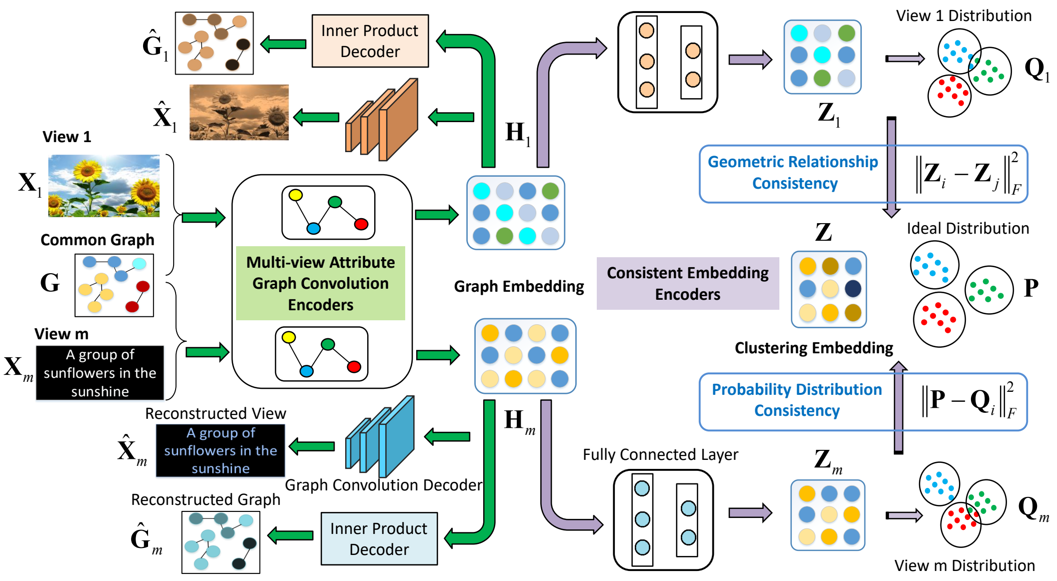

Figure 2: The framework of Multi-View Attribute Graph Convolution Networks for Clustering (MAGCN). It contains two key components: *1)* Multi-view attribute graph convolution encoders with attention mechanism: they are used to learn graph embedding from node attribute and graph data. Attribute and graph reconstruction are executed for end-to-end learning. *2)* Consistent embedding encoders further obtain a consistent clustering embedding among multiple views through geometric relationship and probability distribution consistency.

$f_m(\mathbf{G}, \mathbf{X}_m; \theta) \rightarrow \mathbf{H}_m$. It maps graph $\mathbf{G}$ and m-th view attributes $\mathbf{X}_m$ to d-dimensional graph embedding features $\mathbf{H}_m$, where θ represents multi-view graph auto-encoder parameter. As shown in Fig.1, the output of l-th multi-view attribute graph convolution encoders is

$$\mathbf{H}_m^{(l)} = \sigma\left(\mathbf{D}^{-\frac{1}{2}}\mathbf{G}'\mathbf{D}^{-\frac{1}{2}}\mathbf{H}_m^{(l-1)}\mathbf{W}^{(l)}\right), \tag{1}$$

where $\mathbf{G}' = \mathbf{G} + \mathbf{I}_N$ is the relevance coefficient matrix with added self-connection. $\mathbf{I}_N$ is the identity matrix, $\mathbf{D}_{ii} = \sum_j \mathbf{G}'_{ij}$ and $\mathbf{W}^{(l)}$ is the trainable parameter of the l-th multi-view auto-encoder layer and σ denotes the activation function. As for $\mathbf{H}_m^{(l)}$, when $l = 0$, $\mathbf{H}_m^{(0)}$ is the initial m-th view attribute matrix $\mathbf{X}_m$ and when $l = L$, $\mathbf{H}_m^{(L)}$ is the final graph embedding feature representation $\mathbf{H}_m$.

To determine the relevance between nodes and their neighbors, we use a attention mechanism with shared parameters among nodes. In the l-th multi-view encoder layer, the learnable relevance matrix $\mathbf{S}$ is defined as

$$\mathbf{S} = \varphi\left(\mathbf{G} \odot t_s^{(l)}\mathbf{H}_m^{(l)}\mathbf{W}^{(l)} + \mathbf{G} \odot t_r^{(l)}\mathbf{H}_m^{(l)}\mathbf{W}^{(l)}\right), \tag{2}$$

where ${t_s}^{(l)}$ and ${t_n}^{(l)} \in \mathbb{R}^{1\times d_l}$ represent the trainable parameters related to their own nodes and neighbor nodes, respectively. $\odot$ refers to the element-wise multiplication with broadcasting capability, and φ denotes the activation function which is generally set as the sigmoid activation function (*i.e.*, $sigmoid\,(x) = 1/\left(1 + \exp^{-x}\right)$). We normalize $\mathbf{S}$ to get the final relevance coefficient $\mathbf{G}$, so $\mathbf{G}_{ij}$ is computed by

$$\mathbf{G}_{ij} = \frac{\exp\left(\mathbf{S}_{ij}\right)}{\sum\limits_{k\in\mathbf{N}_i}\exp\left(\mathbf{S}_{ik}\right)}. \tag{3}$$

where $\mathbf{N}_i$ is the set of all nodes adjacent to node i.

After applying L multi-view encoder layers, we get graph embedding $\mathbf{H}_m$ which preserves basically all information about multi-view node attribute matrix $\mathbf{X}$ and graph structure $\mathbf{G}$. Then, we consider the feature $\mathbf{H}_m$ that contains almost all the information to decode as different views. In the decoding process, we use the same number of layers as encoders for decoders, and each decoder layer tries to reverse its corresponding encoder layer. In other words, the decoding process is the inverse of the encoding process. The final decoded output is reconstructed node attribute matrix $\hat{\mathbf{X}}_m$ and the reconstructed graph structure $\hat{\mathbf{G}}_m$, $m = 1, 2, ..., V$. Specifically, the output of the $(l-1)$-th multi-view attribute graph convolution decoders is

$$\hat{\mathbf{H}}_m^{(l-1)} = \sigma\left(\hat{\mathbf{D}}^{-\frac{1}{2}}\hat{\mathbf{G}}'\hat{\mathbf{D}}^{-\frac{1}{2}}\hat{\mathbf{H}}_m^{(l)}\hat{\mathbf{W}}^{(l)}\right). \tag{4}$$

After applying L decoder layers, reconstructed multi-view node attribute matrix $\hat{\mathbf{X}}_m = \hat{\mathbf{H}}_m^{(0)}$ is obtained. As for the reconstructed graph structure $\hat{\mathbf{G}}_m$, $\hat{\mathbf{G}}_m^{ij}$ is implemented by an inner product decoder of h_m^i and h_m^j, where h_m^i and h_m^j are the node i and j of graph embedding features $\mathbf{H}_m$. Specifically, $\hat{\mathbf{G}}_m^{ij}$ is computed by

$$\hat{\mathbf{G}}_m^{ij} = \phi(-{h_m^i}^{\top} h_m^j), \tag{5}$$

where $\phi(\cdot)$ is the inner product operator.

Finally, the reconstruction loss $\mathbf{L}_{re}$ of reconstructed multi-view node attribute matrix $\hat{\mathbf{X}}$ and reconstructed graph structure $\hat{\mathbf{G}}$ can be computed by following:

$$\mathcal{L}_{re} = \min_{\theta}\sum_{i=1}^{M}\left\|\mathbf{X}_i - \hat{\mathbf{X}}_i\right\|_F^2 + \lambda_1\sum_{i=1}^{M}\left\|\mathbf{G} - \hat{\mathbf{G}}_i\right\|_F^2, \tag{6}$$

where θ is the network parameter of multi-view attribute graph convolution encoders.

Consistent Embedding Encoders

In consistent embedding encoders, for view m, we first adopt nonlinear feature extraction mapping for graph embedding $\mathbf{H}_m$. $\mathbf{H}_m$ is mapped into low-dimensional space $\mathbf{Z}_m$. The mapping function is $g_m(\mathbf{H}_m;\eta) \to \mathbf{Z}_m$, where η represents encoder parameter. $\mathbf{Z}_m$ contains almost all the original information so that it is not suitable for multi-view integrating directly. Then, we use consistent clustering layer to learn a common clustering embedding $\mathbf{Z}$ which is adaptively integrated by all the $\mathbf{Z}_m$.

Assume $\mathbf{Z}_m$ and $\mathbf{Z}_b$ are the low-dimensional space feature matrices of view m and b obtained from consistent embedding encoders. Then we can use them to compute a geometric relationship similarity score as $si(\mathbf{Z}_m, \mathbf{Z}_b)$, where $si(\cdot)$ is a similarity function. $si(\cdot)$ can be measured by the Manhattan Distance, Euclidean distance, cosine similarity, *etc*. If the simplest difference similarity function is taken, the loss function of *geometric relationship consistency* $\mathcal{L}_{geo}$ is

$$\mathcal{L}_{geo} = \min_{\eta} \sum_{i \neq j}^{M} \|\mathbf{Z}_i - \mathbf{Z}_j\|_F^2. \tag{7}$$

where η is the network parameter of encoder.

Besides the geometric relationship, we also consider the consistency of the probability distribution between common representation $\mathbf{Z}$ and latent representation of each view $\mathbf{Z}_m$, where $\mathbf{Z}{=}\sum_{m=1}^{M} \beta_i \mathbf{Z}_i$ is an adpative fusion on a low-dimensional feature space, $\mathbf{Z}_m = \{z_1, ..., z_i, ..., z_n\} (\mathbf{Z}_m \in \mathbb{R}^{n \times d})$. We show the details of computing the probability distribution of $\mathbf{Z}$ and $\mathbf{Z}_m$ in the following.

Following [Tao *et al.*, 2019; Wang *et al.*, 2018], we use the Student's t-distribution, , as a kernel to measure the similarity between integrated node representation z_i and centroid μ_j. Thus, the original probability distribution $\mathbf{Q}$ of $\mathbf{Z}$ is

$$q_{ij} = \frac{(1 + ||z_i - \mu_j||^2/\alpha)^{-\frac{\alpha+1}{2}}}{\sum_{j'} (1 + ||z_i - \mu_{j'}||^2/\alpha)^{-\frac{\alpha+1}{2}}}, \tag{8}$$

where $\{\mu_j\}_{j=1}^{k}$ is the k initial cluster centroids, α is the degree of freedom of the Student's t-distribution, q_{ij} is the probability of assigning node i to cluster j. In our experiments, we compute target probability distribution $\mathbf{P}$ of $\mathbf{Z}$. p_{ij} is the elements of $\mathbf{P}$, and $0 \leq p_{ij} \leq 1$. By raising q_i to its second power and normalizing it with the frequency per cluster as follows, we obtain

$$p_{ij} = \frac{q_{ij}^2 / f_i}{\sum_{j'} q_{ij'}^2 / f_{j'}}, \tag{9}$$

where $f_j = \sum_i q_{ij}$ are soft cluster frequencies. To this end, we define our objective as a *probability distribution consistency* loss $\mathcal{L}_{pro}$ between the soft assignment $\mathbf{Q}_m$ of $\mathbf{Z}_m$ and the auxiliary distribution $\mathbf{P}$ of $\mathbf{Z}$ with trade-off parameters ρ as follows

$$\mathcal{L}_{pro} = \min_{\eta} \sum_{m=1}^{M} \rho_m \|\mathbf{Q}_m - \mathbf{P}\|_F^2. \tag{10}$$

Database	Attribute1	Attribute2	Classes	Nodes	Edges
Cora	$1,433$	$2,708$	7	$2,708$	$5,429$
Citeseer	$3,703$	$3,327$	6	$3,327$	$4,732$
Pubmed	500	$19,717$	3	$19,717$	$44,438$

Table 1: The details for experimental databases.

In this way, we could concentrate on the same class data by sharping the data distribution and get a more effective and common representation for multi-view clustering.

3.3 Task for Clustering

By combining Eq. (6), Eq. (7) and Eq. (8), the total loss function of the proposed MAGCN is eventully formulated as

$$\mathcal{L} = \min_{g,c,\mathbf{P}} \mathcal{L}_{re} + \lambda_2 \mathcal{L}_{geo} + \lambda_3 \mathcal{L}_{pro}. \tag{11}$$

Optimizing the overall loss $\mathcal{L}$, we learn the auxiliary distribution $\mathbf{P}$ from the clustering embedding feature $\mathbf{Z}$. Then we predict the cluster of each node from auxiliary distribution $\mathbf{P}$. For node i, its cluster can be calculated by p_i, in which the index with the highest probability value is the i's cluster. Hence we could obtain the cluster label of node i as

$$y_i = \arg\max_k (p_{ik}). \tag{12}$$

4 Experimental Analysis

4.1 Experimental Setting

Metrics and Databases

In order to evaluate the effectiveness of our proposed approach, we conduct extensive experiments on three citation network databases (Cora, Citeseer and Pubmed) [Sen *et al.*, 2008] with three evaluation metrics: clustering accuracy (ACC), normalized mutual information (NMI) and average rand index (ARI), and the higher these indicators, the better the clustering effect. The general graph-structured database contains one graph and one attribute and there is no such real graph-structured data with multi-view attributes at present. Due to the databases used in the experiments, the attributes of graph-structured data are $0, 1$, which are discrete structured and also one-sided described. So we make attributes continuous by changing the operation of that, for the purpose of describing the graph structure more abundantly. Inspired by multi-graph, which constructs another graph by themselves, we construct additional attribute views by original attributes. Specifically, we use Fast Fourier Transform (FFT), Gabor transform, Euler transform and Cartesian product to construct view 2 based on view 1. In Sec. 4.2, the results of each view 2 are analyzed. Brief statistics of the three databases are shown in Table 1, where view 2 is constructed by Cartesian product. For other experiments, we construct from the first view based on Cartesian product.

Implementation Details

In our experiments, we used two layers of multi-view attribute graph convolution encoders for all three databases. For Cora database, node representation dimensions of the two layer are set as $[512, 512]$. As to the Citeseer database, node representation dimensions of the two layer are set as

Database	*Info.*	Cora			Citeseer			Pubmed		
Metric		ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
K-means [MacQueen, 1967]	$\mathbf{X}_1$	0.500	0.317	0.239	0.544	0.312	0.285	0.580	0.278	0.246
Graph Encoder [Tian *et al.*, 2014]	$\mathbf{G}$	0.301	0.059	0.046	0.293	0.057	0.043	0.531	0.210	0.184
Deep Walk [Perozzi *et al.*, 2014]	$\mathbf{G}$	0.529	0.384	0.291	0.390	0.131	0.137	0.647	0.238	0.255
DNGR [Cao *et al.*, 2016]	$\mathbf{G}$	0.419	0.318	0.142	0.326	0.180	0.043	0.468	0.153	0.059
M-NMF [Wang *et al.*, 2017b]	$\mathbf{G}$	0.423	0.256	0.161	0.336	0.099	0.070	0.470	0.084	0.058
DCCA [Andrew *et al.*, 2013]	$\mathbf{X}_1\&\mathbf{X}_2$	0.436	0.214	0.160	0.450	0.221	0.204	0.443	0.048	0.040
DCCA [Andrew *et al.*, 2013]	$\mathbf{G}\&\mathbf{X}_1\&\mathbf{X}_2$	0.583	0.416	0.342	0.513	0.283	0.238	0.508	0.097	0.097
DCCAE [Wang *et al.*, 2015]	$\mathbf{X}_1\&\mathbf{X}_2$	0.472	0.289	0.221	0.503	0.240	0.211	0.537	0.122	0.092
DCCAE [Wang *et al.*, 2015]	$\mathbf{G}\&\mathbf{X}_1\&\mathbf{X}_2$	0.564	0.311	0.241	0.534	0.310	0.230	0.581	0.239	0.186
GAE [Kipf and Welling, 2016]	$\mathbf{G}\&best\mathbf{X}$	0.530	0.397	0.293	0.380	0.174	0.141	0.632	0.249	0.246
VGAE [Kipf and Welling, 2016]	$\mathbf{G}\&best\mathbf{X}$	0.592	0.408	0.347	0.392	0.163	0.101	0.619	0.216	0.201
MGAE [Wang *et al.*, 2017a]	$\mathbf{G}\&best\mathbf{X}$	0.684	0.511	0.448	0.661	0.412	0.414	0.593	0.282	0.248
ARGAE [Pan *et al.*, 2018]	$\mathbf{G}\&best\mathbf{X}$	0.640	0.449	0.352	0.573	0350	0.341	0.681	0.276	0.291
ARVGAE [Pan *et al.*, 2018]	$\mathbf{G}\&best\mathbf{X}$	0.638	0.450	0.374	0.544	0.261	0.245	0.513	0.117	0.078
DAEGC [Wang *et al.*, 2019]	$\mathbf{G}\&best\mathbf{X}$	0.704	0.528	0.496	0.672	0.397	0.410	0.671	0.266	0.278
GATE [Salehi and Davulcu, 2019]	$\mathbf{G}\&best\mathbf{X}$	0.658	0.527	0.451	0.616	0.401	0.381	0.673	0.322	0.299
MAGCN-view 1	$\mathbf{G}\&\mathbf{X}_1$	0.710	0.553	0.476	0.698	0.418	0.403	0.683	0.321	0.310
MAGCN-view 2	$\mathbf{G}\&\mathbf{X}_2$	0.594	0.409	0.327	0.621	0.363	0.366	0.539	0.261	0.227
MAGCN	$\mathbf{G}\&\mathbf{X}_1\&\mathbf{X}_2$	0.751	0.598	0.532	0.711	0.458	0.462	0.691	0.331	0.321

Table 2: Clustering results of various methods on three databases. Best results are highlighted in red and the suboptimal results are marked in blue. *Info.* means the input of different methods: $\mathbf{G}$ donates the graph structure, $\mathbf{X}_1$ and $\mathbf{X}_2$ represent the node feature of view 1 and view 2.

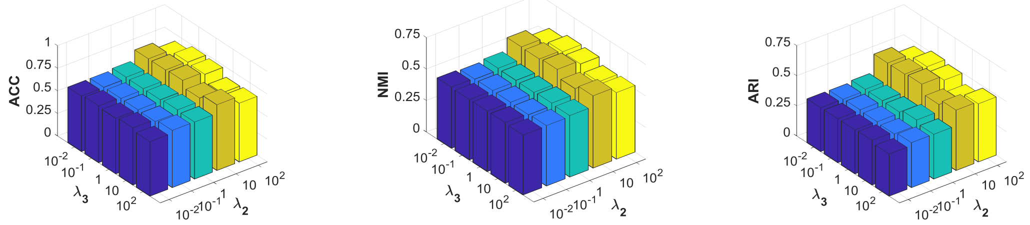

Figure 3: Visualization the change of parameters λ_2 and λ_3 of geometric relationship consistency and probability distribution consistency.

$[2000, 512]$. For Pubmed, the dimension of two-layer multi-view attribute graph convolution auto-encoder is $[128, 64]$. In integrate-encoder, we use a fully connected layer in all three databases. We use non-linear activation function σ as Relu function in the multi-view graph convolution auto-encoder. As for regular term coefficient λ_1, λ_2 and λ_3, we set λ_1 as 1. λ_2 and λ_3 are set range from 10^{-2} to 10^{2}, and analyze the influence of parameters later in Sec. 4.2: Impact of Loss Coefficient.

Comparison Algorithms

We choose several state-of-the-art clustering compared algorithms as follows.

- **node attribute**: K-Means;
- **graph structure**: Graph Encoder, DeepWalk, denoising autoencoder for graph embedding (DNGR) and modularized nonnegative matrix factorization (M-NMF);
- **graph structure & node attribute**: graph auto-encoders (GAE) and variational graph auto-encoders (VGAE), marginalized graph autoencoder (MGAE), adversarial regularized graph autoencoder (ARGAE) and adversarial variational regularized graph autoencoder (ARVGAE), deep attentional embedding graph clustering (DAEGA) and graph attention auto-encoders (GATE);
- **deep multi-view clustering**: deep canonical correlation analysis (DCCA) and deep typical correlated auto-encoder (DCCAE).

4.2 Experimental Results

Evaluation Metrics with Comparison Algorithms

Table 2 summarizes the comparative evaluation results on three databases of Cora, CIteseer, and Pubmed. The proposed MAGCN clearly outperforms all the compared methods. Specifically, MAGCN improves on GCNs of single view by a margin of more than 5% on Cora and Citeseer and more than 1% on Pubmed, which indicates that it is effective to integrate different views with the consistency of geometric relationship and the probability distribution. From the suboptimal results, we can see that our method also has a substantial im-

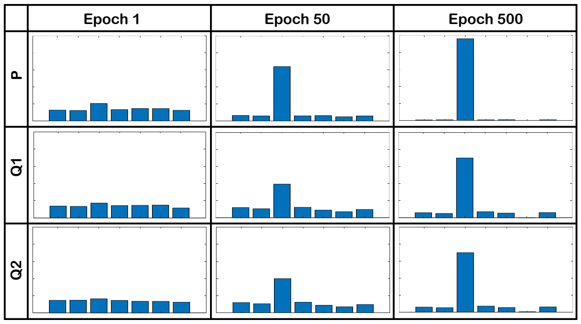

Figure 4: Visualization of the distribution $\mathbf{Q}_1$, $\mathbf{Q}_2$ and $\mathbf{P}$. The x-axis carries the clusters, and y-axis informs Cluster Probability.

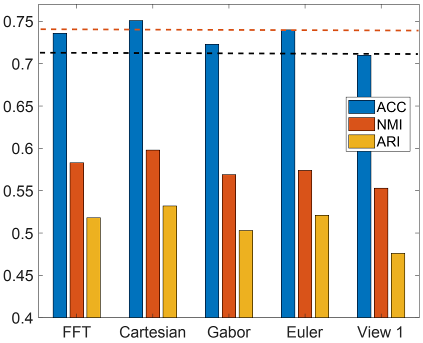

Figure 5: Metrics vs. different view 2 on Cora database.

provement in single-view graph clustering. Meanwhile, the single view graph convolution clustering methods: DAEGC and GATE, have relatively better clustering performance, which shows that attention mechanism aggregating neighborhood information according to trainable attention weights helps improving clustering performance. In addition, deep multi-view clustering methods achieve better performance by using the graph structure information. This shows that the graph structure information can make beneficial contribution to clustering. The performance of deep multi-view clustering methods is even worse than many single-view GCN clustering methods, which also indicates the excellence of GCN in processing graph structured data.

Analysis of Probability Distribution Consistency

To illustrate the advantages of probability distribution consistency in our model, we conduct the following visual experiments on Cora database. We randomly select a sample from the third class and compute original probability distribution $\mathbf{Q}_m$ for each view's low dimensional representation $\mathbf{Z}_m$ and the target probability distribution $\mathbf{P}$ for the common feature representation $\mathbf{Z}$. Our goal is to get the ideal description space of multi-view. In terms of low-dimensional features $\mathbf{Z}_m$, we use the consistency of probability distribution as constraint to reduce the differences between different views. As shown in Fig.4, in the initial iteration, the random initialization makes the probability of each class basically similar, and it is impossible to find out which class the sample belongs to. But the probability in the third class increases with the number of iterations, and the probability distributions on $\mathbf{Z}$, $\mathbf{Z}_1$ and $\mathbf{Z}_2$ tend to be consistent, which indicates that the ideal multi-view description feature $\mathbf{Z}$ is learned gradually.

Impact of Parameters

There are three regularization parameters in our model: λ_1, λ_2 and λ_3, and we use controlling variables method to analyze their impact. We keep the regular parameters λ_1 of reconstruction loss unchanged and change the regular parameters λ_2 and λ_3 of geometric relationship consistency and probability distribution consistency in the model. As shown in Fig.3, when λ_2 is between 10^{-2} and 10^{2}, the corresponding evaluation metrics ACC, NMI and ARI will largely remain the same. That explains that the model has a relatively good clustering robustness on geometric relationship consistency. As for the regular term λ_3 of probability distribution consistency, when λ_3 was set as 10, the multi-view clustering has the best performance. When the value of λ_2 is too large, the model clustering performance is not good. Therefore, we set λ_3 of probability distribution consistency around 10 in the experiment.

Analyzing Different View 2

Analyze the performance of our model when the second view are constructed in different ways (FFT, Cartesian, Garbor, and Euler) on Cora database. To facilitate the comparison, we use the view 1 as the baseline. We keep all parameters consistent to ensure the fairness of the experiments. As shown in Fig.5, we can see for all kind of view 2, the clustering results with two views is better than the case of single view 1 (marked by black line). In addition, the Cartesian product way works the better than other constructing ways for view 2 (marked by red line). This is why we construct the second view using Cartesian products in all the other experiments.

5 Conclusions

In this paper, we propose a novel Multi-View Attribute Graph Convolution Networks for Clustering (MAGCN), a generally method to multi-view graph neural network. MAGCN is designed with dual encoders that reconstruct the extracted features in high dimension and integrate the low dimension consistent information. Multi-view attribute graph auto-encoder and consistent embedding encoder network successively reduce the noise and the difference among different views, and finally get the ideal description space of multi-view attribute graph for clustering. Experimental results on the multi-view graph structure databases demonstrate the validity of our method and perform superior advantages over several state-of-the-art algorithms.

Acknowledgments

This work is supported by National Natural Science Foundation of China (Grant 61773302, 61906142, 61906141), Initiative Postdocs Supporting Program (Grant BX20190262), China Postdoctoral Science Foundation (Grant 2019M653564, 2019M663642), National Natural Science Foundation of Shaanxi Province (Grant 2020JZ-19, 2020JQ-317, 2020JQ-327).

References

[Andrew *et al.*, 2013] Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. Deep canonical correlation analysis. In *ICML*, pages 1247–1255, 2013.

[Belkin and Niyogi, 2002] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *NeurIPS*, pages 585–591, 2002.

[Cao *et al.*, 2016] Shaosheng Cao, Wei Lu, and Qiongkai Xu. Deep neural networks for learning graph representations. In *AAAI*, pages 1145–1152, 2016.

[Gao *et al.*, 2020] Quanxue Gao, Zhizhen Wan, Ying Liang, Qianqian Wang, Yang Liu, and Ling Shao. Multi-view projected clustering with graph learning. *Neural Networks*, pages 335–346, 2020.

[Geng *et al.*, 2019] Xu Geng, Yaguang Li, Leye Wang, Lingyu Zhang, Qiang Yang, Jieping Ye, and Yan Liu. Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting. In *AAAI*, pages 3656–3663, 2019.

[Gilmer *et al.*, 2017] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *ICML*, pages 1263–1272, 2017.

[Hamilton *et al.*, 2017] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NeurIPS*, pages 1024–1034, 2017.

[Khan and Blumenstock, 2019] Muhammad Raza Khan and Joshua E Blumenstock. Multi-gcn: Graph convolutional networks for multi-view networks, with applications to global poverty. In *AAAI*, volume 33, pages 606–613, 2019.

[Kipf and Welling, 2016] Thomas N. Kipf and Max Welling. Variational graph auto-encoders. In *arXiv preprint arXiv:1611.07308*, 2016.

[Kipf and Welling, 2017] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[Li *et al.*, 2019] Zhaoyang Li, Qianqian Wang, Zhiqiang Tao, Quanxue Gao, and Zhaohua Yang. Deep adversarial multi-view clustering network. In *IJCAI*, pages 2952–2958, 2019.

[Ma *et al.*, 2018] Tengfei Ma, Cao Xiao, Jiayu Zhou, and Fei Wang. Drug similarity integration through attentive multi-view graph auto-encoders. In *IJCAI*, pages 3477–3483, 2018.

[MacQueen, 1967] James MacQueen. Some methods for classification and analysis of multivariate observations. In *Mathematical Statistics and Probability*, pages 281–297, 1967.

[Pan *et al.*, 2018] 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.

[Perozzi *et al.*, 2014] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: online learning of social representations. In *ACM SIGKDD*, pages 701–710, 2014.

[Salehi and Davulcu, 2019] Amin Salehi and Hasan Davulcu. Graph attention auto-encoders. *CoRR*, abs/1905.10715, 2019.

[Sen *et al.*, 2008] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, pages 93–93, 2008.

[Tao *et al.*, 2019] Zhiqiang Tao, Hongfu Liu, Jun Li, Zhaowen Wang, and Yun Fu. Adversarial graph embedding for ensemble clustering. In *IJCAI*, pages 3562–3568, 2019.

[Tian *et al.*, 2014] Fei Tian, Bin Gao, Qing Cui, Enhong Chen, and Tie-Yan Liu. Learning deep representations for graph clustering. In *AAAI*, pages 1293–1299, 2014.

[Velickovic *et al.*, 2018] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018.

[Wang *et al.*, 2015] Weiran Wang, Raman Arora, Karen Livescu, and Jeff Bilmes. On deep multi-view representation learning. In *ICML*, pages 1083–1092, 2015.

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

[Wang *et al.*, 2017b] Xiao Wang, Peng Cui, Jing Wang, Jian Pei, Wenwu Zhu, and Shiqiang Yang. Community preserving network embedding. In *AAAI*, pages 203–209, 2017.

[Wang *et al.*, 2018] Qianqian Wang, Zhengming Ding, Zhiqiang Tao, Quanxue Gao, and Yun Fu. Partial multi-view clustering via consistent gan. In *IEEE ICDM*, pages 1290–1295, 2018.

[Wang *et al.*, 2019] Chun Wang, Shirui Pan, Ruiqi Hu, Guodong Long, Jing Jiang, and Chengqi Zhang. Attributed graph clustering: A deep attentional embedding approach. In *IJCAI*, pages 3670–3676, 2019.

[Xie *et al.*, 2019] Deyan Xie, Xiangdong Zhang, Quanxue Gao, Jiale Han, Song Xiao, and Xinbo Gao. Multiview clustering by joint latent representation and similarity learning. *IEEE Trans. Cybernetics*, DOI: 10.1109/TCYB.2019.2922042, 2019.

[Zhang *et al.*, 2018a] Changqing Zhang, Huazhu Fu, Qinghua Hu, Xiaochun Cao, Yuan Xie, Dacheng Tao, and Dong Xu. Generalized latent multi-view subspace clustering. *IEEE TPAMI*, 42(1):86–99, 2018.

[Zhang *et al.*, 2018b] Xi Zhang, Lifang He, Kun Chen, Yuan Luo, Jiayu Zhou, and Fei Wang. Multi-view graph convolutional network and its applications on neuroimage analysis for parkinson's disease. In *AMIA*, page 1147, 2018.