Self-Supervised Graph Attention Networks for Deep Weighted Multi-View Clustering

Zongmo Huang[1], Yazhou Ren[1,2*], Xiaorong Pu[1,2*], Shudong Huang[3], Zenglin Xu[4], Lifang He[5]

[1]School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China
[2] Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China, Shenzhen, China
[3]College of Computer Science, Sichuan University, Chengdu, China
[4]School of Computer Science and Technology, Harbin Institute of Technology Shenzhen, Shenzhen, China
[5]Department of Computer Science and Engineering, Lehigh Univerisity, Bethlehem, USA
zongmohuang@gmail.com, yazhou.ren@uestc.edu.cn, puxiaor@uestc.edu.cn, huangsd@scu.edu.cn,
zenglin@gmail.com, lih319@lehigh.edu

Abstract

As one of the most important research topics in the unsupervised learning field, Multi-View Clustering (MVC) has been widely studied in the past decade and numerous MVC methods have been developed. Among these methods, the recently emerged graph neural networks (GNNs) shine a light on modeling both topological structure and node attributes in the form of graphs, to guide unified embedding learning and clustering. However, existing GNN-based MVC methods generally do not give sufficient consideration to the use of self-supervised information during the training process, which prevents them from achieving better results. To this end, in this paper we propose Self-Supervised Graph Attention Networks for Deep Weighted Multi-View Clustering (SGDMC), which exploits the self-supervised information to enhance the effectiveness of the graph-based deep MVC model from two aspects. Firstly, a novel attention allocating approach that considers both the similarity of node attributes and the self-supervised information is developed to comprehensively evaluate the relevance among different nodes. Secondly, to alleviate the negative impact caused by noisy samples and the discrepancy of cluster structures, we further design a sample-weighting strategy based on the attention graphs as well as the discrepancy between the global pseudo-labels and the local cluster assignment of each single view. Experimental results on multiple real-world datasets demonstrate the effectiveness of our method over existing approaches.

Introduction

Multi-view clustering aims to promote clustering performance by utilizing the complementary knowledge from multiple views, which has been widely studied in the past decade. Among numerous MVC methods (Yang and Wang 2018), the deep embedded multi-view clustering models (Li et al. 2019; Xu et al. 2022a; Zhou and Shen 2020) perform superior clustering results owing to the representation learning capability of deep networks and become increasingly popular in these years. However, the effectiveness of these methods is limited as they cannot exploit the topological information of samples. To address this issue, a series of studies that introduce the Graph Neural Networks (Wu et al. 2021) (GNN) to the deep embedded MVC framework (Fan et al. 2020; Cheng et al. 2020) are proposed recently. With the GNN structures, these models aggregate the latent features of samples from their neighbors according to the topological structures of nodes. In this way, the GNN-based MVC models explore the clusters based on both attributes and adjacent relationship of samples, thus obtaining much better clustering results.

Although great progresses have been achieved, existing GNN-based MVC methods often give insufficient consideration to the usage of the self-supervised information during the training, hindering their clustering performance. To this end, we introduce Self-Supervised Graph Attention Networks for Deep Weighted Multi-View Clustering (SGDMC), which improves GNN-based MVC models by utilizing self-supervised information in the following two ways.

Firstly, to comprehensively evaluate the relevance of samples and improve the aggregating capability of the graph attention layer, a novel attention allocating approach that considers the similarity of both local node attributes and global pseudo-labels is developed for the learned adjacent graph. Specifically, in each epoch, SGDMC first employs kNN graph algorithm (Fix and Hodges 1989) to construct the adjacent graph on the latent embeddings of nodes in each view. After that, the attention coefficient of each edge in the adjacent graph is determined by the Gaussian similarity of their features and the cosine similarity of their pseudo-labels. In this way, the proposed SGDMC not only improve the representation learning capability of model but also has a much wider application scenario as it does not require explicit graphs as input like most GNN-based MVC models (Fan et al. 2020; Cheng et al. 2020; Xia et al. 2022).

Secondly, based on attention graphs as well as the discrepancy between the global pseudo-label and local cluster assignment, a novel sample-weighting strategy is also proposed in this paper. Since existing GNN-based MVC models generally treat all the samples equally, their clustering performance is extremely sensitive to the existence of the noisy samples and are easily stuck into the suboptimal solutions. Meanwhile, during the self-supervised training pro-

*Corresponding authors.

cess, these models always force all the views to have the consistent latent features or cluster distributions as the global pseudo-labels. However, according to the complementary principle, a single view generally cannot reflect the complete cluster structure, and thus samples belonging to different clusters may be very close to each other in a specific view. Therefore, forcing the samples' cluster assignment aligned across all the views is not conducive to model training and may even result in the loss of view-specific information.

To tackle the above problems, the proposed sample-weighting strategy evaluates the importance of samples in each view based on the attention graph as well as the discrepancy between their local cluster assignments and pseudo-labels. In the attention graph, the received attention of each node indicates its reference value for other nodes. Intuitively, the important nodes generally receive more attention, while the nodes ignored by others are more likely to be the noisy samples. Thus the samples that receive more attention are assigned with higher weights in our model and vice versa. In addition, to alleviate the discrepancy issue, the proposed SGDMC moderately decreases the weights of samples whose local clusters assignments are dissimilar to their pseudo-labels in each view.

With the novel sample-weighting strategy, samples with higher reliability will play more important roles during the training, thus the negative impact from noisy samples and the discrepancy issue is significantly alleviated. Moreover, the obtained sample weight information will be also used in the attention allocation process in the following iterations, so that the nodes will pay more attention to the more important and reliable neighbors during the training and the clustering performance will be further enhanced.

In summary, the contributions of this paper include:

- A novel attention allocating approach that considers both node attributes and the self-supervised information is developed to comprehensively evaluate the relevance of samples and enhance the aggregating capability of the graph attention layer.
- A novel sample-weighting strategy based on attention graphs as well as the discrepancy between self-supervised information and local embedding distributions is proposed to alleviate the negative impact from noisy samples and the discrepancy between the global pseudo-label and the local cluster assignment.
- Experimental results on multiple real-world datasets demonstrate the state-of-the-art clustering effectiveness of our method.

Related Work

Multi-View Clustering

By utilizing the complementary information from multiple views, multi-view clustering achieves much better clustering results than the conventional single-view models (MacQueen 1967; Ng, Jordan, and Weiss 2001; Ester et al. 1996) and become increasingly popular in the past decade.

In Co-train (Kumar and Daumé 2011) and Co-reg (Kumar, Rai, and Daumé 2011), to obtain a consistent cluster assignment, the authors enforce the view-specific eigenvectors in different views to be similar. In (Zhang et al. 2019), a joint framework that simultaneously learns the collaborative binary codes for data and binary cluster structures is proposed to decrease the time and storage cost of MVC methods. SAMVC (Ren et al. 2020) applies the $\ell_{2,1}$ norm and auto-weighting strategy to alleviate the impact caused by noisy samples and corrupted views. To enhance the effectiveness and robustness of MVC model, a novel dual self-paced learning mechanism for both instances and features learning is designed (Huang et al. 2021).

Although considerable progress has been made, the effectiveness of the conventional MVC algorithms are still limited as they are shallow models. To this end, a series of MVC methods that employ deep neural networks to promote the clustering performance has been proposed in the past few years. In EAMC (Zhou and Shen 2020), a novel end-to-end deep MVC model is developed to perform the modality-specific feature learning, feature fusion and cluster assignment in a joint manner. To extract the disentangled features and improve the interpretability of the model, variation auto-encoder (Kingma and Welling 2014) is adopted by (Yin, Huang, and Gao 2020) and (Xu et al. 2021b). A recent work (Xu et al. 2022b) propose an effective multi-view discriminative feature learning framework to alleviate the impact from the views with unclear clustering structures.

With the representation learning capability of deep networks, the deep embedded MVC methods yield considerable clustering results, yet the inability to use topological information impedes them from achieving better clustering performance, until the emergence of graph neural networks.

Graph Neural Networks

Unlike the conventional deep models that can merely process Euclidean data, by aggregating the information from the node neighbors, Graph Neural Networks (GNN) successfully handle both topological information and attributes of samples, thus attracting increasing attention in these years. For instance, Graph Convolutional Networks (GCN) (Kipf and Welling 2017) extends the idea of convolution to the representation aggregating while learning the latent embeddings based on node attributes and graph structures. To decrease the training cost of graph convolutions, an inductive aggregating approach which enables GCN to be trained on the mini-batches is developed in GraphSAGE (Hamilton, Ying, and Leskovec 2017). With the latent self-attention layer, the Graph Attention Networks (GAT) (Veličković et al. 2018) enables the nodes to assign different weights to their neighbors and thus enhance the aggregation capability.

Up to now, GNN has been widely applied in many machine learning fields like recommendation systems (Ying et al. 2018), image denoising (Chen et al. 2020), natural language processing (Gao, Chen, and Ji 2019) and information retrieval (Yu et al. 2018) etc. Recently, there are also some studies that employ the GNN structures to promote the effectiveness of deep embedded MVC models. O2MGC (Fan et al. 2020) makes the first attempt to apply the GNN technique in multi-view clustering. Specifically, this model reconstructs all predefined graphs based on the latent embed-

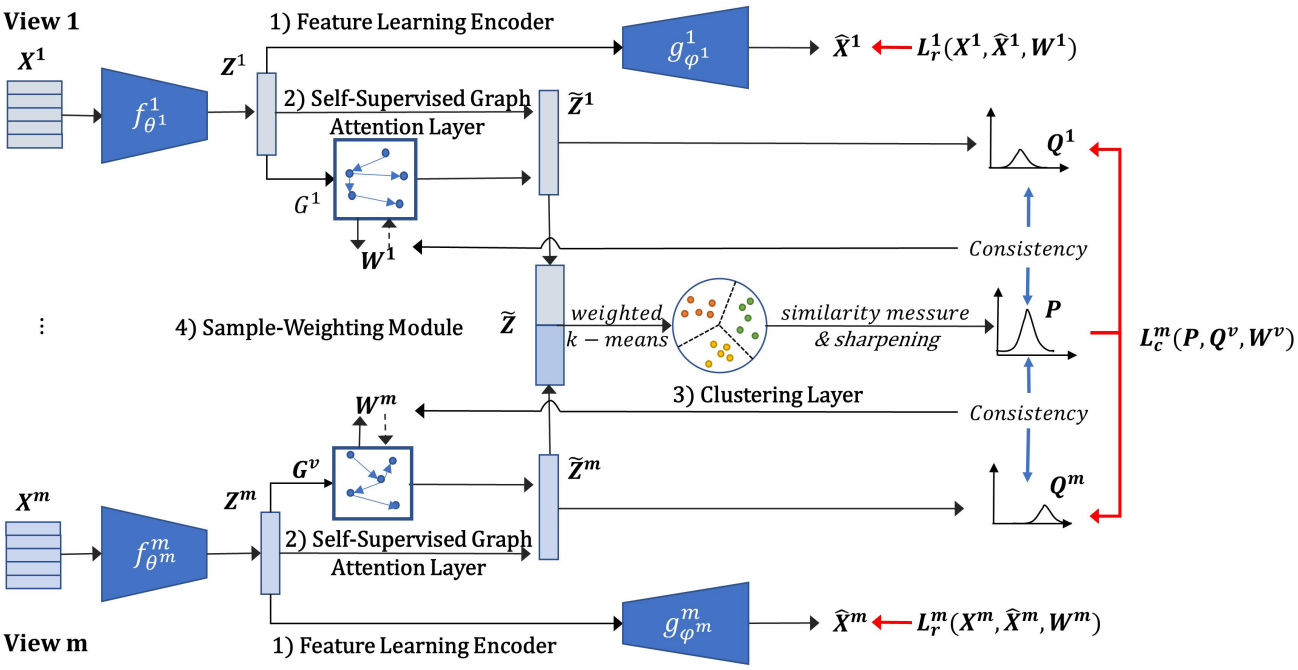


Figure 1: The framework of SGDMC. SGDMC is composed of four kinds of modules: 1) feature learning encoder projects the original data into the low-dimensional latent embeddings; 2) self-supervised graph attention layer aggregates the node features based on both attributes and self-supervised information; 3) clustering layer concatenates the latent representations from all the views and generates the pseudo-labels for the self-supervised training; 4) sample-weighting module assigns different weights to the nodes based on the attention graphs and the discrepancy between their local cluster assignments and pseudo-labels.

ding of samples. In MAGCN (Cheng et al. 2020), the attention mechanism is employed to reduce the noise and redundancy of the multi-view graph data. Another work (Xia et al. 2022) incorporates the graph convolutional network with the deep subspace clustering model, in which the self-supervised information is applied to assist both latent representation learning and coefficient representation learning.

By taking advantage of the GNN structures, these methods achieve state-of-the-art clustering performance. However, since they require explicit graph data as input, their application is extremely limited. Meanwhile, the value of the self-supervised information during the training process is also overlooked in these studies. To this end, the proposed SGDMC not only learns the graph from the sample attributes but also exploits the self-supervised information in the attention allocating and the instance learning process.

Methodology

Problem Definition

Given a dataset $X = \{X^v\}_{v=1}^m$ with n samples in m views, where $X^v = \{x_1^v; x_2^v; \ldots; x_n^v\} \in R^{n \times d^v}$, d^v denotes the dimension of the feature vector in the v-th view. Our target is to partition n instances into k clusters based on the complementary information of different views. Specifically, we aim to improve clustering results by making full use of self-supervised information during training.

Network Architecture

As Figure 1 shows, the network architecture of SGDMC is composed of four different types of modules, *i.e.*, feature learning encoder, self-supervised graph attention layer, clustering layer, and sample-weighting module. The details and functions of each module are introduced as follows.

Feature Learning Encoder Like most existing deep embedded MVC models, the auto-encoder structure is applied in our method to learn the low-dimensional latent features of the samples in each view. Let $f_{\theta^v}^v$ and $g_{\phi^v}^v$ denote the encoder and decoder, where θ^v and ϕ^v are learnable parameters, then the latent features of i-th sample in the v-th view is:

$$z_i^v = f_{\theta^v}^v(x_i^v), \tag{1}$$

After the encoding process, the decoder $g_{\phi^v}^v$ is applied to reconstruct the original data by decoding z_i^v:

$$\hat{x}_i^v = g_{\phi^v}^v(z_i^v), \tag{2}$$

where $\hat{x}_i^v$ represents the reconstructed data.

Then, the feature learning encoder in each view is trained by optimizing the reconstruction loss $l_r^v(i)$ for each node:

$$l_r^v(i) = ||x_i^v - g_{\phi^v}^v(f_{\theta^v}^v(x_i^v))||_2^2. \tag{3}$$

Self-Supervised Graph Attention Layer To enhance latent representation of samples with their neighbors and the self-supervised information, a graph attention layer based on

the novel attention allocating mechanism is applied to aggregate the latent features learned by the auto-encoders. Let $Z^v = \{z_1^v; z_2^v; \ldots; z_n^v\} \in R^{n \times d'_v}$ denotes the latent features learned by the auto-encoder of the v-th view. As our method is designed for the data that do not have explicit graph structures, we first construct the adjacent graph G^v via the kNN graph algorithm for each view. Concretely, the edge $G^v(i,j)$ exists only when z_j^v is one of k-nearest neighbors of z_i^v or $i = j$. After obtaining the adjacent graphs, our method assigns different weights to these edges based on a novel attention allocating approach. Specifically, in a certain iteration, the weight e_{ij}^v assigned to the edge $G^v(i,j)$ is computed by:

$$e_{ij}^v = w_j^v * (exp(-\gamma||z_i^v - z_j^v||^2) + \frac{< p_i, p_j >}{||p_i||||p_j||}), \quad (4)$$

where γ is the control parameter of the Gaussian kernel, and w_j^v, p_i, p_j respectively represent the reliability of the j-th sample, the self-supervised pseudo-labels of the i-th and j-th samples, all of these variables are generated in the earlier iteration and the details of them will be illustrated in the following two sections.

After the normalization via softmax function, the element within the attention matrix of the v-th view A^v has:

$$a_{ij}^v = \begin{cases} \frac{exp(e_{ij}^v)}{\sum_{j \in \mathcal{N}_i^v} exp(e_{ij}^v)} & j \in \mathcal{N}_i^v \\ 0 & j \notin \mathcal{N}_i^v, \end{cases} \quad (5)$$

where $\mathcal{N}_i^v$ denotes the set of index that $G^v(i,j)$ exists.

From Eq. (4), the attention assigned to $G^v(i,j)$ is mainly determined by the Gaussian similarity of latent features and the cosine similarity of pseudo-labels between the end points. Therefore, the relevance of different nodes is evaluated based on information from both local attributes and global cluster assignment, so that the aggregation capability of the self-attention layer is significantly enhanced. In addition, by considering the sample weights of the referenced nodes, the proposed attention allocation approach further reduces the influence of the less reliable samples.

After obtaining the attention matrix, a weighted two-layer plain residual network is applied to refine the latent features:

$$\begin{aligned} \tilde{z}_i^v &= \sum_{h=0}^{2} \alpha_h^v (A^v)^h z_i^v, \\ s.t. \quad & \sum_{h=0}^{2} \alpha_h^v = 1, \end{aligned} \quad (6)$$

where $\alpha^v = [\alpha_0^v, \alpha_1^v, \alpha_2^v]$ are learnable layer weights. Through the residual network, each node aggregates the information of samples in multi-level receptive fields, which brings better generalization and effectiveness to our model.

Clustering Layer At the top of the self-supervised graph attention layer, a clustering layer is constructed to explore the cluster structures of each view. Let $\tilde{Z}^v = \{\tilde{z}_1^v; \tilde{z}_2^v; \ldots; \tilde{z}_n^v\} \in R^{n \times d'_v}$ be the refined latent features in the v-th view. Following most existing deep embedded clustering models, based on the Student's t-distribution (Maaten and Hinton 2008), the probability of the i-th example belonging to the j-th cluster in the v-th view is:

$$q_{ij}^v = c_{\mu^v}^v(\tilde{z}_i^v) = \frac{(1 + ||\tilde{z}_i^v - \mu_j^v||^2)^{-1}}{\sum_j (1 + ||\tilde{z}_i^v - \mu_j^v||^2)^{-1}}, \quad (7)$$

where μ^v denotes the learnable cluster centroids.

Let $Q^v = \{q_1^v; q_2^v; \ldots; q_n^v\} \in R^{n \times k}$ denote the cluster assignments in the v-th view. In this module, we adopt the self-training strategy proposed in (Xu et al. 2022b), which uses self-supervised information to train the whole model by optimizing the discrepancy between the local cluster assignment Q^v and the global pseudo-label P.

Specifically, to obtain P, the refined embeddings in all the views are firstly concatenated as a unified representation $\tilde{Z}$:

$$\tilde{Z} = [\tilde{Z}^1, \tilde{Z}^2, \ldots, \tilde{Z}^m] \in R^{n \times \sum_{v=1}^m d'_v}. \quad (8)$$

Then, to alleviate the noisy issue, each sample is weighted based on the attention η_i it receives from all the view:

$$\eta_i = \sum_{v=1}^{m} \sum_{k \neq i} A_{ki}^v. \quad (9)$$

In Eq. (9), the value of η_i stands for the sum of attention paid to the i-th sample by other samples in all views. Intuitively, samples that receive more attention tend to be more important while the ones generally ignored by others maybe the noises. Therefore the global weight w_i of each sample is:

$$w_i = min(\eta_i / \lambda, 1), \quad (10)$$

where λ is set to the median value of $\eta = [\eta_1, \eta_2, \ldots, \eta_n]$ to ensure at least half of the samples are treated normally.

After that, the weighted k-means is applied to generate the global cluster centroids c_j:

$$\min_{c_1, c_2, \ldots, c_k} \sum_{i=1}^{n} \sum_{j=1}^{k} w_i ||\tilde{z}_i - c_j||^2. \quad (11)$$

Based on Student's t-distribution, the soft assignment s_{ij} of each node and each cluster centroid is:

$$s_{ij} = \frac{(1 + ||\tilde{z}_i - c_j||^2)^{-1}}{\sum_j (1 + ||\tilde{z}_i - c_j||^2)^{-1}}. \quad (12)$$

Finally, by sharpening the soft assignment, the global pseudo-label P has:

$$p_{ij} = \frac{(s_{ij}^2 / \sum_i s_{ij})}{\sum_j (s_{ij}^2 / \sum_i s_{ij})}, \quad (13)$$

where p_{ij} denotes the probability that the i-th example belongs to the j-th cluster.

After obtaining P, we define the Kullback-Leibler divergence between the pseudo-label p_i and q_i^v as the clustering loss $l_c^v(i)$ for each sample, which will be optimized during the self-supervised training process:

$$l_c^v(i) = D_{KL}(p_i || q_i^v) = \sum_{j=1}^{k} p_{ij} log \frac{p_{ij}}{q_{ij}^v}. \quad (14)$$

Sample-Weighting Module To alleviate the impact from noisy samples and the discrepancy between the local and global cluster assignments, we further propose a novel sample-weighting mechanism based on the attention graphs and self-supervised information. Specifically, after obtaining the pseudo-labels in every T iterations, our method updates the sample weights in different views by:

$$w_i^v = min(\eta_i^v/\lambda^v, 1)exp(\frac{< p_i, q_i^v >}{||p_i||||q_i^v||} - \tau^v), \qquad (15)$$

where $\tau^v = max_j exp(\frac{<p_j, q_j^v>}{||p_j||||q_j^v||})$.

Similar to the computation of the global weights, in Eq. (15), η_i^v represents the attention that the i-th sample receives in the v-th view and λ^v is the median value of η^v. Additionally, as we have obtained global cluster assignment P and local cluster assignment Q^v, we can also evaluate the samples from the perspective of assignment consistency. With the complementary principle in multi-view clustering, the data in a single view generally cannot reflect the cluster structures completely, samples in different clusters maybe very close to each other in a certain view. Therefore, enforcing the consistent cluster assignment across all the views is not conducive to model training and may even lead to the loss of view-specific information in the latent features.

To address this issue, in Eq. (15), we moderately decrease the weights of samples whose local clusters assignments are dissimilar to their pseudo-labels in each view. Besides, as Eq. (4) shows, the sample-weights are also utilized to compute the edge weights in the self-supervised graph attention layer. In this way, the nodes aggregate the information from more important and convincing neighbors, thus providing better robustness and effectiveness to the model.

Incorporating the sample weights Eq. (15), reconstruction loss Eq. (3) and clustering loss Eq. (14) into a unified framework, the objective function of the SGDMC is:

$$\begin{aligned} L &= \sum_{v=1}^{m} L_r^v(X^v, \hat{X}^v, W^v) + L_c^v(P^v, Q^v, W^v) \\ &= \sum_{v=1}^{m}\sum_{i=1}^{n} w_i^v l_r^v(i) + w_i^v l_c^v(i) \\ &= \sum_{v=1}^{m}\sum_{i=1}^{n} w_i^v(||x_i^v - g_{\phi^v}^v(f_{\theta^v}^v(x_i^v))||_2^2 + \sum_{j=1}^{k} p_{ij} log\frac{p_{ij}}{q_{ij}^v}). \end{aligned} \qquad (16)$$

Optimization

The optimization of SGDMC is composed by two procedures, *i.e.*, initialization and finetuning. In the initialization stage, for each view, we firstly pretrain auto-encoders $f_{\theta^v}^v$ and $f_{\phi^v}^v$ by optimizing the reconstruction loss in Eq. (3). Setting all the w_j^v to 1 and not considering the pseudo-labels, the attention matrix A^v and the refined representation $\tilde{z}_i^v$ are computed by Eq. (5) and Eq. (6). Based on the unified representation $\tilde{Z}$, the initial pseudo-label P is obtained by Eq. (13) and the global cluster centroids are decomposed to initialize $c_{\mu^v}^v$ for each view. As the last step of the initialization, the initial sample weights w_i^v are generated by Eq. (15).

Algorithm 1: The SGDMC model.

Input: Data set X^v, $v = 1, 2, \ldots, m$; Cluster number k; Align rate threshold δ.
Output: Cluster assignments $Y = \{y_1, y_2, \ldots, y_n\}$.

```
Pretrain the auto-encoder separately in each view by optimizing Eq. (3).
Initialize the pseudo-labels by Eq. (13).
Initialize initialize c_{μ^v}^v by the decomposing the global cluster centroids.
Initialize the sample weights w_i^v by Eq. (15).
while Not reach the maximum iteration T_max do
    repeat
        Finetune all parameters of the entire network by optimizing Eq. (16).
    until The iteration time is divisible by T.
    Update pseudo-labels P and the sample weights w_i^v by Eq. (13) and Eq. (15).
    Compute the aligned rate (AR).
    if AR ≥ δ then
        Stop training.
    end if
end while
Compute the final pseudo-labels P by Eq. (13).
Compute y_i for each sample by Eq. (17).
```

Then, during the finetuning stage, the whole network of SGDMC is trained by optimizing the objective function Eq. (16). At the end of every T iterations during the finetuning stage, the sample weights w_i and pseudo-labels P are updated by Eq. (15) and Eq. (13) respectively.

Besides, following (Xu et al. 2022b), our method terminates when the aligned rate is over a predefined threshold or exceeds the maximum iteration number T_{max}. Specifically, the i-th example is aligned when $y_i^1 = y_i^2 = \cdots = y_i^m$ ($y_i^v = \arg\min_j(q_{ij})$), and the aligned rate is rate of aligned examples in all examples. When the whole training process is completed, we compute the pseudo-label P once again and the final clustering assignment y_i for the i-th sample is:

$$y_i = \arg\max_j(p_{ij}). \qquad (17)$$

The workflow of SGDMC is summarized in Algorithm 1.

Experiments

Experimental Setup

Datasets Three widely used and publicly available multi-view datasets are implemented in our study:

BDGP (Cai et al. 2012) consists of 2500 samples of 5 different kinds of drosophila embryos. Each sample is described by 1750 visual features and 79 textual features.

Handwritten Numerals[1] sources from UCI machine learning repository, which contains 2000 handwritten numeral images over 10 classes (0-9). Each instance has six visual views, including 216 profile correlations, 76 Fourier

[1]https://archive.ics.uci.edu/ml/datasets.php

Dataset	BDGP			Handwritten Numerals			Reuters		
Methods	ACC(%)	NMI(%)	ARI(%)	ACC(%)	NMI(%)	ARI(%)	ACC(%)	NMI(%)	ARI(%)
KM (1967)	57.68(2.93)	47.35(2.43)	19.38(3.90)	75.45(5.00)	78.58(4.14)	66.72(4.32)	29.12(6.33)	13.53(8.87)	6.80(6.33)
SC (2001)	59.98(8.17)	50.98(5.57)	26.20(5.14)	77.69(0.08)	86.91(0.15)	75.26(0.17)	17.89(0.11)	2.71(0.24)	0.05(0.01)
IDEC (2017)	91.28(7.55)	85.64(6.95)	81.98(8.82)	84.13(8.65)	84.61(3.97)	78.37(8.03)	45.98(2.78)	25.17(2.71)	18.02(2.22)
MVKKM (2012)	42.02(3.02)	27.33(1.78)	12.69(1.53)	67.21(3.09)	67.70(0.28)	55.76(0.79)	24.81(5.82)	11.67(6.79)	3.76(3.57)
MLAN (2017)	47.32(0.00)	31.30(0.00)	24.29(0.00)	97.35(0.00)	94.00(0.00)	94.17(0.00)	21.50(0.00)	15.04(0.00)	1.49(0.00)
AMVCD (2020)	56.87(5.41)	43.58(5.72)	22.71(4.29)	79.66(9.32)	84.90(3.58)	75.99(8.69)	27.38(2.73)	10.51(2.58)	4.29(1.65)
GMC (2020)	59.12(0.00)	62.61(0.00)	43.13(0.00)	88.20(0.00)	90.73(0.00)	85.40(0.00)	19.75(0.00)	12.95(0.00)	1.29(0.00)
SAMVC (2020)	51.31(7.48)	45.15(6.49)	19.60(5.96)	76.37(7.36)	84.41(2.50)	73.87(6.25)	18.83(1.92)	4.58(3.40)	0.32(0.62)
DEMVC (2021)	92.78(1.55)	83.31(3.35)	82.64(3.89)	67.69(6.15)	70.61(2.99)	58.86(4.92)	46.71(0.85)	25.31(1.43)	20.41(1.06)
SDMVC (2022)	97.89(0.52)	93.41(1.24)	94.85(1.20)	97.18(0.51)	94.44(0.52)	93.93(0.97)	47.07(0.79)	27.12(1.20)	21.22(0.99)
SGDMC (ours)	**98.78(0.15)**	**96.07(0.41)**	**96.99(0.36)**	**98.24(0.17)**	**95.86(0.39)**	**96.09(0.37)**	**60.08(1.11)**	**36.16(0.99)**	**30.11(1.09)**

Table 1: Clustering results of compared methods on three multi-view datasets, the best result in each column is highlighted in red and the second best result is denoted by underline.

coefficients of the character shapes, 64 Karhunen-Love coefficients, 6 morphological features, 240 pixel averages in 2 $\times$ 3 windows, and 47 Zernike moments.

Reuters[2] is comprised of 1200 articles in 6 categories (C15, CCAT, E21, ECAT, GCAT and M11), each providing 200 articles. For each article, it is written in five different languages (English, French, German, Italian, and Spanish).

Comparing Methods To demonstrate the effectiveness of the proposed SGDMC, we compare it with seven existing state-of-the-art multi-view clustering methods, *i.e.*, MVKKM (Tzortzis and Likas 2012), MALN (Nie, Cai, and Li 2017), AMVCD(Huang, Kang, and Xu 2020), GMC(Wang, Yang, and Liu 2020), SAMVC(Ren et al. 2020), DEMVC(Xu et al. 2021a), SDMVC(Xu et al. 2022b).

To make a comprehensive comparison, we also employ some single-view methods, *i.e.*, KMeans (KM) (MacQueen 1967), Spectral Clustering (SC) (Ng, Jordan, and Weiss 2001), and Improved Deep Embedded Clustering (IDEC) (Guo et al. 2017) by concatenating features from all views.

Implementation Details Following (Guo et al. 2017), we use the same fully connected auto-encoder structure on all three datasets. Specifically, for each view, the structure of encoder is: Input(d^v) - Fc500 - Fc500 - Fc2000 - Fc10(d'_v), and the decoder is symmetric with the encoder. All feature learning encoders are pretrained for 2000 epochs. The aligned rate threshold δ is 0.8 and the number of neighbors β applied to construct the adjacent graph is set to 10. The batch size is set to the instance number n. During the fine-tuning stage, the pseudo-label P and the sample weights W^v are updated for every $T = 1000$ epochs. The training process compulsively terminates when the epoch number exceeds $T_{max} = 10000$. As for the comparing methods, we directly use the open-source codes and follow the parameter settings by the corresponding publications.

[2]http://lig-membres.imag.fr/grimal/data.html

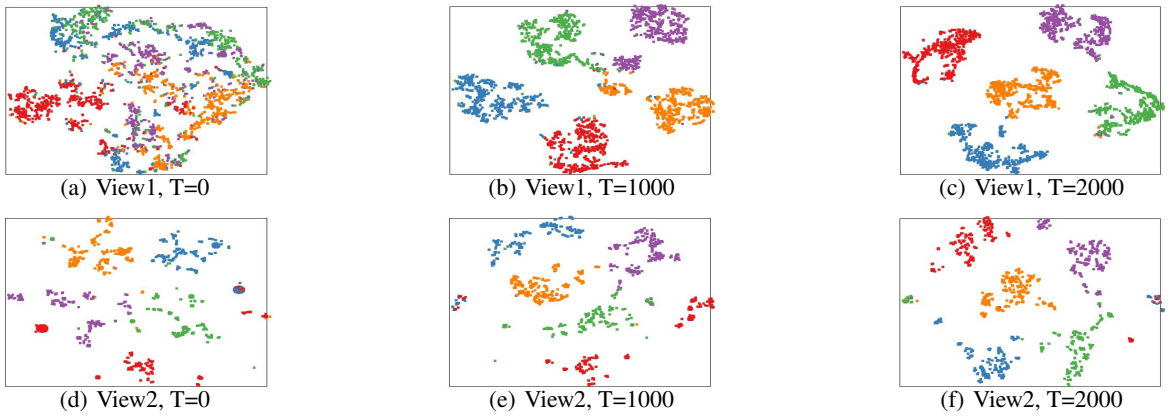
Evaluation Measures Three widely used metrics accuracy (ACC), normalized mutual information (NMI) and adjusted rand index (ARI) are applied to evaluate the clustering performance, higher values of these metrics indicate better clustering performance. The average results and standard deviations of ten independent runs of each method on three datasets are reported.

Clustering Results

In this subsection, we investigate the effectiveness of the proposed SGDMC through the comparison with the baselines, the embedding visualization during training and the analysis of parameter sensitivity .

Comparison with Baselines Table 1 shows the clustering results of the baseline methods compared with SGDMC, where the best result is highlighted in boldface and the second best result is denoted by underline in each column. From the results, we can find that the proposed SGDMC achieves the highest average result on all three metrics over the multi-view datasets with different view compositions. This is because through a novel attention allocation approach, the relevance among different nodes is evaluated in a more comprehensive manner, so that nodes aggregate the information from more important and convincing neighbors in the graph attention layer. Meanwhile, the small variances also illustrate the better robustness of the proposed method, which indicates that the developed sample-weighting strategy can effectively alleviate the influence from noisy samples.

Visualization of Learning Process To visually investigate the effectiveness of the SGDMC, the t-SNE algorithm (Maaten and Hinton 2008) is applied to reduce the dimension of the refined latent representations of different views $\tilde{Z}^v$ to 2D and demonstrate the separability/non-separability of the data at iterations $T = 0, 1000, 2000$ during the fine-tuning stage. From Figure 2, we can observe that even if the discriminative degree of features is very low in the be-



(a) View1, T=0 (b) View1, T=1000 (c) View1, T=2000

(d) View2, T=0 (e) View2, T=1000 (f) View2, T=2000

Figure 2: Visualization of each view's refined latent embeddings during the training on BDGP dataset, different colors denote the labels of corresponding nodes.

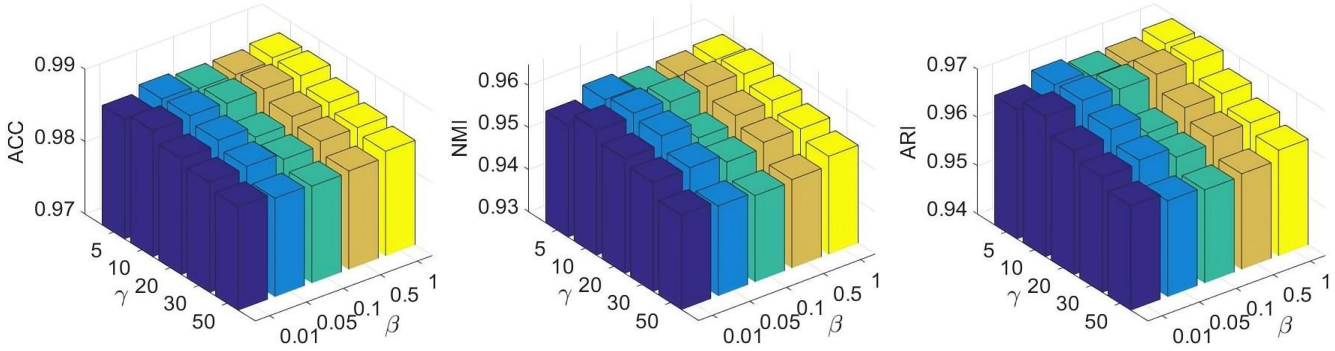


Figure 3: Clustering performance w.r.t. different parameter settings on BDGP dataset.

ginning of the finetuning stage, as the training forwards, the cluster structures become increasingly clear, which demonstrates the effectiveness of the proposed SGDMC.

Parameter Sensitivity We investigate the two main hyper-parameters in constructing attention graphs, *i.e.*, the control parameter γ utilized in the Gaussian kernel and the number of neighbors β applied in kNN graph algorithm. To study the influence of these hyper-parameters on the clustering results, we employ the grid search strategy and test the average clustering performance of ten independent runs on BDGP dataset. From Figure 3, while the proposed method achieves relatively stable results on different values of γ, the clustering performance promotes when the value of β rises from 5 to 10 and declines as β further increases. This is mainly because when the number of neighbors used to aggregate the target node is too small, the nodes cannot make full use of the information from its neighbors, thus limiting the clustering performance. On the other hand, when the value of δ is too large, the nodes will aggregate the information from the less relevant nodes and affect clustering results.

Conclusion

To promote the effectiveness of the GNN-based MVC model with self-supervised information, in this paper we propose Self-Supervised Graph Attention Networks for Deep Weighted Multi-View Clustering (SGDMC). Specifically, a novel attention allocating strategy considering both local and self-supervised information is developed to accurately evaluate the relevance of samples and enhance the aggregating capability of the graph attention layer. Besides, to alleviate the negative impact from noisy samples and discrepancies between global and local cluster structures, a novel sample-weighting mechanism based on attention graphs and the discrepancy between global pseudo-labels and local cluster assignments is also proposed. Experiments on different types of multi-view real-world datasets demonstrate the state-of-the-art performance of the proposed method.

Acknowledgements

This work was supported in part by the National Key Research and Development Program of China (No. 2018AAA0100204), Sichuan Science and Technology Program (Nos. 2021YFS0172, 2022YFS0047, 2022YFS0055), Open Foundation of Nuclear Medicine Laboratory of Mianyang Central Hospital (No. 2021HYX017), Medico-Engineering Cooperation Funds from University of Electronic Science and Technology of China (No. ZYGX2021YGLH022), Lehigh's grants (S00010293 and 001250), and National Science Foundation (MRI 2215789).

References

Cai, X.; Wang, H.; Huang, H.; and Ding, C. H. Q. 2012. Joint stage recognition and anatomical annotation of *drosophila* gene expression patterns. *Bioinform.*, 28(12): 16–24.

Chen, K.; Pu, X.; Ren, Y.; Qiu, H.; Li, H.; and Sun, J. 2020. Low-Dose CT Image Blind Denoising with Graph Convolutional Networks. In *ICONIP*, 423–435.

Cheng, J.; Wang, Q.; Tao, Z.; Xie, D.; and Gao, Q. 2020. Multi-View Attribute Graph Convolution Networks for Clustering. In *IJCAI*, 2973–2979.

Ester, M.; Kriegel, H.-P.; Sander, J.; Xu, X.; et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, 226–231.

Fan, S.; Wang, X.; Shi, C.; Lu, E.; Lin, K.; and Wang, B. 2020. One2Multi Graph Autoencoder for Multi-view Graph Clustering. In *WWW*, 3070–3076.

Fix, E.; and Hodges, J. L. 1989. Discriminatory Analysis - Nonparametric Discrimination: Consistency Properties. *International Statistical Review*, 57: 238–247.

Gao, H.; Chen, Y.; and Ji, S. 2019. Learning Graph Pooling and Hybrid Convolutional Operations for Text Representations. In *WWW*, 2743–2749.

Guo, X.; Gao, L.; Liu, X.; and Yin, J. 2017. Improved Deep Embedded Clustering with Local Structure Preservation. In *IJCAI*, 1753–1759.

Hamilton, W. L.; Ying, Z.; and Leskovec, J. 2017. Inductive Representation Learning on Large Graphs. In *NeurIPS*, 1024–1034.

Huang, S.; Kang, Z.; and Xu, Z. 2020. Auto-weighted multi-view clustering via deep matrix decomposition. *Pattern Recognition*, 97: 107015.

Huang, Z.; Ren, Y.; Pu, X.; Pan, L.; Yao, D.; and Yu, G. 2021. Dual self-paced multi-view clustering. *Neural Networks*, 140: 184–192.

Kingma, D. P.; and Welling, M. 2014. Auto-Encoding Variational Bayes. In *ICLR*.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.

Kumar, A.; and Daumé, H. 2011. A Co-training Approach for Multi-view Spectral Clustering. In *ICML*, 393–400.

Kumar, A.; Rai, P.; and Daumé, H., III. 2011. Co-regularized Multi-view Spectral Clustering. In *NeurIPS*, 1413–1421.

Li, Z.; Wang, Q.; Tao, Z.; Gao, Q.; and Yang, Z. 2019. Deep Adversarial Multi-view Clustering Network. In *IJCAI*, 2952–2958.

Maaten, L. v. d.; and Hinton, G. 2008. Visualizing Data using t-SNE. *JMLR*, 9(86): 2579–2605.

MacQueen, J. 1967. Some Methods for Classification and Analysis of Multivariate Observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, 281–297.

Ng, A. Y.; Jordan, M. I.; and Weiss, Y. 2001. On Spectral Clustering: Analysis and an algorithm. In *NeurIPS*, 849–856.

Nie, F.; Cai, G.; and Li, X. 2017. Multi-View Clustering and Semi-Supervised Classification with Adaptive Neighbours. In *AAAI*, 2408–2414.

Ren, Y.; Huang, S.; Zhao, P.; Han, M.; and Xu, Z. 2020. Self-paced and auto-weighted multi-view clustering. *Neurocomputing*, 383: 248 – 256.

Tzortzis, G.; and Likas, A. 2012. Kernel-Based Weighted Multi-view Clustering. In *ICDM*, 675–684.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *ICLR*.

Wang, H.; Yang, Y.; and Liu, B. 2020. GMC: Graph-Based Multi-View Clustering. *TKDE*, 32(6): 1116–1129.

Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; and Yu, P. S. 2021. A Comprehensive Survey on Graph Neural Networks. *TNNLS*, 32(1): 4–24.

Xia, W.; Wang, Q.; Gao, Q.; Zhang, X.; and Gao, X. 2022. Self-Supervised Graph Convolutional Network for Multi-View Clustering. *TMM*, 24: 3182–3192.

Xu, J.; Li, C.; Ren, Y.; Peng, L.; Mo, Y.; Shi, X.; and Zhu, X. 2022a. Deep Incomplete Multi-View Clustering via Mining Cluster Complementarity. In *AAAI*, 8761–8769.

Xu, J.; Ren, Y.; Li, G.; Pan, L.; Zhu, C.; and Xu, Z. 2021a. Deep embedded multi-view clustering with collaborative training. *Inf. Sci.*, 573: 279–290.

Xu, J.; Ren, Y.; Tang, H.; Pu, X.; Zhu, X.; Zeng, M.; and He, L. 2021b. Multi-VAE: Learning Disentangled View-Common and View-Peculiar Visual Representations for Multi-View Clustering. In *ICCV*, 9234–9243.

Xu, J.; Ren, Y.; Tang, H.; Yang, Z.; Pan, L.; Yang, Y.; Pu, X.; Yu, P. S.; and He, L. 2022b. Self-Supervised Discriminative Feature Learning for Deep Multi-View Clustering. *TKDE*, 1–12.

Yang, Y.; and Wang, H. 2018. Multi-view clustering: A survey. *Big Data Mining and Analytics*, 1(2): 83–107.

Yin, M.; Huang, W.; and Gao, J. 2020. Shared Generative Latent Representation Learning for Multi-View Clustering. In *AAAI*, 6688–6695.

Ying, R.; He, R.; Chen, K.; Eksombatchai, P.; Hamilton, W. L.; and Leskovec, J. 2018. Graph Convolutional Neural Networks for Web-Scale Recommender Systems. In *KDD*, 974–983.

Yu, J.; Lu, Y.; Qin, Z.; Zhang, W.; Liu, Y.; Tan, J.; and Guo, L. 2018. Modeling Text with Graph Convolutional Network for Cross-Modal Information Retrieval. In *PCM*, 223–234.

Zhang, Z.; Liu, L.; Shen, F.; Shen, H. T.; and Shao, L. 2019. Binary Multi-View Clustering. *TPAMI*, 41(7): 1774–1782.

Zhou, R.; and Shen, Y.-D. 2020. End-to-End Adversarial-Attention Network for Multi-Modal Clustering. In *CVPR*, 14607–14616.