计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.210300036

基于融合变分图注意自编码器的深度聚类模型

康 雁 寇勇奇 谢思宇 王 飞 张 兰 吴志伟 李 浩
云南大学软件学院 昆明 650504
(562530855@qq.com)

摘 要 聚类作为数据挖掘和机器学习中最基本的任务之一,在各种现实世界任务中已得到广泛应用。随着深度学习的发展,深度聚类成为一个研究热点。现有的深度聚类算法主要从节点表征学习或者结构表征学习两个方面入手,较少考虑同时将这两种信息进行融合以完成表征学习。提出一种融合变分图注意自编码器的深度聚类模型 FVGTAEDC(Deep Clustering Model Based on Fusion Varitional Graph Attention Self-encoder),此模型通过联合自编码器和变分图注意自编码器进行聚类,模型中自编码器将变分图注意自编码器从网络中学习(低阶和高阶)结构表示进行集成,随后从原始数据中学习特征表示。在两个模块训练的同时,为了适应聚类任务,将自编码器模块融合节点和结构信息的表示特征进行自监督聚类训练。通过综合聚类损失、自编码器重构数据损失、变分图注意自编码器重构邻接矩阵损失、后验概率分布与先验概率分布相对熵损失,该模型可以有效聚合节点的属性和网络的结构,同时优化聚类标签分配和学习适合于聚类的表示特征。综合实验证明,该方法在5个现实数据集上的聚类效果均优于当前先进的深度聚类方法。

关键词:深度聚类;表征学习;自编码器;变分图注意自编码器;自监督聚类
中图法分类号 TP181

Deep Clustering Model Based on Fusion Variational Graph Attention Self-encoder

KANG Yan,KOU Yong-qi,XIE Si-yu,WANG Fei,ZHANG Lan,WU Zhi-wei and LI Hao
School of Software,Yunnan University,Kunming 650504,China

Abstract As one of the most basic tasks in data mining and machine learning,clustering is widely used in various real-world tasks. With the development of deep learning deep clustering has become a research hotspot. Existing deep clustering algorithms are mainly from two aspects of node representation learning or structural representation learning. Less work considers fusing these two kinds of information at the same time to complete representation learning. This paper proposes a deep clustering model FVGTAEDC (Deep Clustering Model Based on Fusion Varitional Graph Attention Self-encoder),this model joints the autoencoder and the variational graph attention autoencoder for clustering. In the model,the autoencoder integrates the variational graph attention autoencoder from the network to learn (low-order and high-order) structural representations,and then the feature representation is learned from the original data. While the two modules are trained,in order to adapt to the clustering task,self-supervised clustering training for the autoencoder module is integrated with the representation features of the node and the structure information. Comprehensive clustering loss,autoencoder reconstruction data loss,and variational graph attention autoencoder reconstruction adjacency matrix loss,the relative entropy loss of the posterior probability distribution and the prior probability distribution. The method can effectively aggregate the attributes of nodes and the structure of the network,while optimizing the assignment of cluster labels and learning the representation features suitable for clustering. Comprehensive experiments prove that the method is better than the current advanced deep clustering method on 5 real data.

Keywords Deep clustering,Representation learning,Self encoder,Variational graph attention self-encoder,Self-supervised clustering

1 引言

我们的世界正处于一个信息和数据的时代,数据的采集量和传输量呈指数增长。数据以不同的形式存在,如图像、文本、语音,或者复杂网络(社交网络、引文网络等)的形式[1]。从这些数据中自动地挖掘有效信息,分析数据之间的相互关系,成为当今的一大需求。聚类作为数据挖掘与机器学习中一项基本任务,主要是根据样本的相似程度,将相似的样本归入同一类簇[2]。这种特点有利于我们挖掘数据之间潜在的信息,从而促进各种现实世界任务,例如图像聚类[3]、文本聚类[4]、社区发现[5]等。

过去几十年,学者们已经从不同角度对聚类进行了广泛

基金项目:国家自然科学基金(61762092);云南省软件工程重点实验室(2020SE303);云南省重大科研计划(202002AB080001)
This work was supported by the National Natural Science Foundation(61762092),Key Laboratory of Software Engineering(2020SE303) and Major Scientific Research Plan of Yunnan Province(202002AB080001).
通信作者:李浩(lihao707@ynu.edu.cn)

研究:聚类的定义是什么?什么是正确的聚类指标?如何有效地将实例分组到集群中?[10]等。K 均值(K-means)[2]和 DBSCAN[7]等传统聚类方法,通常在低维数据上有着不错的表现,但其非常依赖原始特征的好坏。

最近,深度学习在人工智能与机器学习任务中取得了巨大的成功。深度学习的基本思想是通过深层神经网络学习良好的表示特征[8],将深度学习应用于聚类,即深度聚类,成为了一个热门的研究方向。与传统的聚类方法相比,深度聚类能有效将聚类的目标融入深度学习强大的表示能力中。许多深度聚类算法使用到自编码器,例如在自编码器学习的表示中使用 K-means[9]、深度嵌入聚类(DEC)算法[10]、改进的深度嵌入聚类算法(IDEC)[11]、变分深度嵌入(VDE)[13]、稀疏先验深度子空间聚类(PARTY)[12]等。然而,这些方法通常关注从数据本身提取特征,在表征学习时很少考虑数据的结构关系。

近年来图形卷积神经网络(GCN)得到了广泛的研究[14],图聚类取得了显著进展[19]。同样,许多深度图聚类算法也使用到自编码器,例如图自动编码器(GAE)和变分图自动编码器(VGAE)[15]、深度注意力嵌入图聚类(DAEGC)[16]、基于邻接共享的联合聚类嵌入图自动编码器(EGAE-JOCAS)[17]、边缘化图自动编码器(MGAE)[18],上述所有基于 GCN 的聚类方法都依赖于重构邻接矩阵来更新模型,并且这些方法关注从网络结构中进行表征学习(图嵌入)[20],而忽略了数据本身的特征。为了将节点信息和结构信息进行结合,结构深度聚类网络(SDCN)使用简单的 GCN 网络将结构信息集成到深度聚类中[20],但是这种方法存在过度平滑和无法有效融合结构信息的问题。

受到 VGAE 和 IDEC 工作的启发,本文提出一种融合变分图注意自编码器的深度聚类模型,该模型在自编码器和变分图注意自编码器的联合框架下解决表征学习的问题,在该框架下可以有效聚合节点的属性和网络的结构,它们之间的相互作用在学习聚类的表示时起着重要的作用。具体来说,首先根据原始数据相似度构建邻接矩阵(针对非图数据),它能够揭示数据的底层结构信息;随后通过变分图自编码器学习(低阶或者高阶)结构信息;通过权重因子将结构信息传递到自编码器中学习节点信息,同时为了适应于聚类任务,采用了一个自监督聚类模块,该模块将“高置信”的分配作为软标签指导优化聚类过程。

本文主要贡献包括:

(1)首次将变分图自编码器与深度聚类结合,变分图自编码器不仅可以捕获结构非线性相似性,还可以学习数据的分布,因此当自编码器和变分图自编码器被同时优化时,可以学习更有效的表示;

(2)针对图信息,没有使用常规的图卷积网络来融合邻居信息,我们使用图注意卷积来捕捉相邻节点对目标节点的重要程度,并且为了增加图中节点之间的连接性,我们使用图的幂 G^k 来增加图的连通性,该操作在距离最多 k 跳的节点之间建立联系;

(3)在 5 个真实数据集上的大量实验证明了我们的方法可以获得更准确的聚类结果。

2 相关工作

本节介绍聚类的定义,并总结了 4 种传统聚类方法,随后介绍目前主流的 2 种深度学习方法:深度聚类,深度图聚类。

2.1 聚类的定义

聚类是根据数据相似度将模式(通常表示为向量或者多维空间中的点)分类为组(集群)。直观上[22],有效集群中模式彼此之间的相似度高于不同集群中模式彼此之间的相似度,这个“集群”的分类方法事先是未知的,需要算法根据样本的分布和相关性进行切分,聚类的形式描述如下[21]。

令 $U=\{p_1,p_2,\cdots,p_n\}$ 表示一个实体集合,p_i 表示第 i 个模式 $i=\{1,2,\cdots,n\}$;$C_t\subseteq U$,$t=1,2,\cdots,k$,$C_t=\{p_{t_1},p_{t_2},\cdots,p_{t_w}\}$;$proximity(p_{ms},p_{ir})$,其中,第 1 个下标表示模式所属的类,第 2 个下标表示某类中某一模式,函数 $proximity$ 用来刻画模式的相似性距离。若诸类 C_t 为聚类之结果,则诸 C_t 须满足如下条件:

$$(1)\bigcup_{t=1}^{k}C_t=U$$

$$(2)\forall C_m,C_r\subseteq U,C_m\neq C_r,C_m\cap C_r=\varnothing,$$

$$\min_{\forall p_{mu}\in C_m,\forall p_{rv}\in C_r,C_r\subseteq U\&C_m\neq C_r}(proximity(p_{mu},p_{rv}))>\max_{\forall p_{mu}\in C_m,\forall p_{rv}\in C_r,C_r\subseteq U\&C_m\neq C_r}(proximity(p_{mu},p_{rv}))$$

2.2 传统聚类

传统聚类方法主要包含以下几类。(1)基于划分的方法:例如 K 均值[2]、高斯混合模型(GMM)[23],这类方法通过迭代将每个样本划分给特定的类别,然后根据当前的划分更新中心参数。(2)基于层次的方法:例如 AGNES 算法[24],采用自底向上的策略,先将每个样本作为一个簇,然后合并这些原子簇为越来越大的簇,直到满足某个终止条件。DIANA 算法[24]则相反,采用自顶向下的策略,先将所有样本置于一个簇中,然后逐渐细分为越来越小的簇,直到达到某个终止条件。(3)基于密度的方法:例如 DBSCAN[7],使用密度替代距离来描述样本点之间的相似度,通过连接密度较大的样本,得到聚类结果。(4)基于图的方法:例如谱聚类[25],该方法需要构造一个相似矩阵,利用相似矩阵的拉普拉斯矩阵进行特征分解,取 k 个最小特征值对应的特征向量构成低维矩阵,再对其使用其他的聚类算法进行聚类。我们可以发现传统聚类方法大多集中在原始数据空间上进行建模,而忽略了学习适合聚类任务的良好表示。

2.3 深度聚类

深度聚类方法旨在将深度表示学习与聚类目标相结合。深度聚类算法大致可以分为两类:(1)在学习表示之后应用聚类的两阶段工作,例如利用自编码器来学习原始数据的低维特征,然后运行 k-means 算法得到聚类结果[9]。利用稀疏先验的自编码器学习非线性潜在空间中同时适应局部和全局子空间结构的表示,然后采用传统的聚类算法进行标签分配[12]。这类方法将聚类与训练自动编码器分开,这可能导致学习的表示不是最适合后续的聚类任务。(2)联合优化特征学习和聚类的方法,例如深度嵌入聚类算法使用堆栈式自编码对数据进行预训练,然后移除解码器。剩余编码器通过定义的 KL-散度聚类损失微调,从而提高了聚类的内聚性[10]。改进的深度嵌入聚类算法(IDEC)认为定义的聚类损失会破坏特征空间,导致特征不具有代表性,因此它们重新加入解码器,并与聚类损失一起优化重构误差[11]。而这些方法关注从数据本身学习表示特征,很少关注于数据之间的联系。

2.4 深度图聚类

卷积神经网络在处理频谱或图像问题时被广泛用于深度

学习。最近已有图卷积网络被用于解决基于图的学习问题[14]，如复杂网络聚类（图聚类），复杂网络聚类的研究对分析复杂网络的拓扑结构具有十分重要的理论意义[1]。当涉及到用于图聚类的深度方法时，许多方法涉及自动编码器，例如Kipf提出了图自编码器（GAE）和图变分自编码器（VGAE），其使用GCN作为编码器来将图结构集成到节点特征中以学习节点嵌入，之后利用传统的聚类算法得到聚类结果[15]。边缘化图自动编码器（MGAE）利用一种边缘化的图卷积网络来破坏网络节点内容，允许节点内容与网络特征交互，并在图自动编码器上下文中边缘化被破坏的特征来学习图特征表示。学习到的特征被输入到图聚类的谱聚类算法中[18]。很明显，这类方法和深度聚类一样存在着表示学习和聚类分离的问题。另一类算法试图将表示学习和聚类统一在一个框架下，深度注意嵌入式图聚类（DAEGC）算法通过使用图注意力网络来捕捉相邻节点对目标节点的重要性，将图中邻接结构和节点特征编码成紧凑的表示，用该表示重构邻接矩阵。此外，来自图嵌入本身的软标签被生成以监督自训练图聚类过程，该过程迭代地细化聚类结果[16]。基于邻接共享的联合聚类嵌入图自编（EGAE-JOCAS）将松弛的k均值和谱聚类统一在一个框架内；此外，该方法提出的联合聚类在图卷积层具有相同的邻接性。通过执行SGD和交替采用闭式解来同时优化两个部分，以确保快速收敛[17]。但是这些方法在利用图卷积网络不断融合结构信息（低阶和高阶）的过程中，可能弱化了数据本身的特征。

基于以上工作，我们可以了解到如何有效利用数据特征信息和网络结构信息进行聚类仍然是一个开放的问题。

3　融合变分图注意力自编码器的深度聚类模型

本文提出的融合变分图注意自编码器的深度聚类模型的总体框架如图1所示。其主要包含4个部分：(1)基于原始数据相似度构建邻接矩阵（针对非图数据）；(2)变分图注意自编码模块学习结构表示；(3)将结构表示传递到自编码模块中学习特征表示；(4)自监督聚类模块指导聚类提高性能。

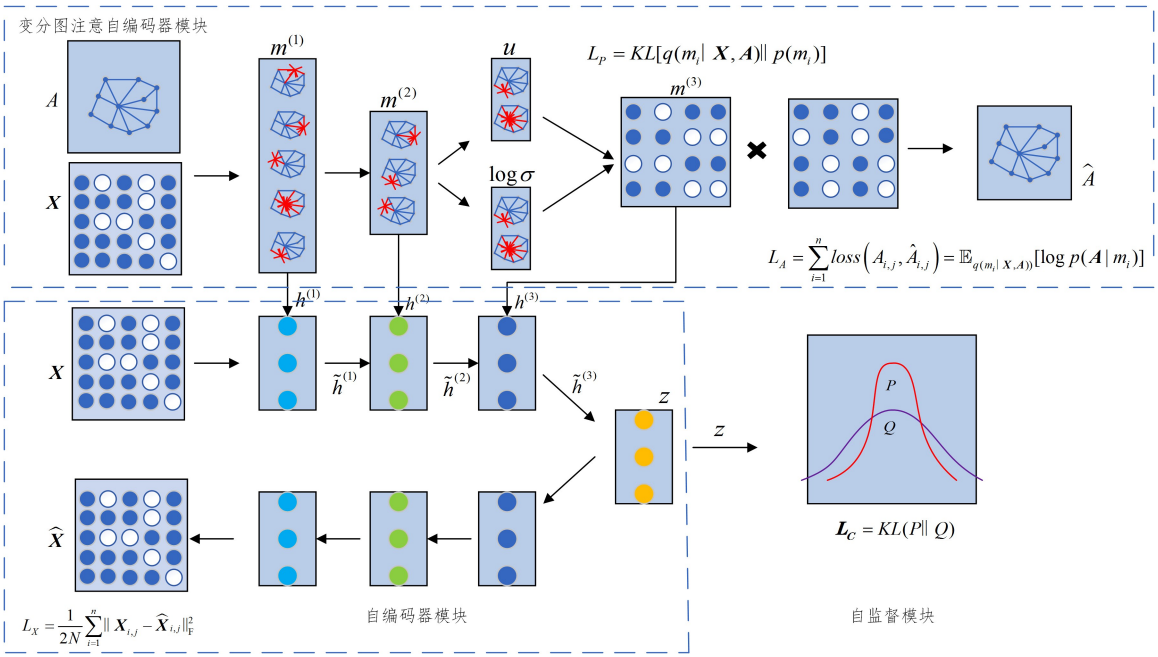


图1　FVGTAEDC模型结构

Fig. 1　Architecture of FVGTAEDC

图1中，X和$\hat{X}$分别是原始数据和重构数据，$\boldsymbol{A}$和$\hat{\boldsymbol{A}}$分别是原始邻接矩阵和重构邻接矩阵。$h^{(L)}$和$m^{(L)}$分别是自编码器模块和变分图注意自编码器模块中的第L层表示。注意：z为自编码器嵌入表示，z在重构数据的同时用来生成分布Q，目标分布P由分布Q计算。$m^{(3)}$为变分图注意自编码器嵌入表示，$m^{(3)}$用来重构邻接矩阵。整个框架由4部分损失负责优化训练，即L_X，L_P，L_A，L_C。

3.1　基于原始数据相似度构建邻接矩阵

当给定原始数据$X\in\mathbb{R}^{N\times d}$，其中$N$是数据个数，$d$是数据维度，我们基于数据间的相似度构造相似矩阵，随后找出每个数据的前k个相似邻居，设置边将其与邻居连接起来。基于以上操作，我们就可以从非图数据中获得邻接矩阵。

计算数据间相似性的方法主要有以下两种。

(1)热核函数

$$similar(x_i,x_j,t) = \exp\frac{-\|x_i - x_j\|}{t} \tag{1}$$

其中，t是热传导方程中的时间参数，设置$t=2$。

(2)余弦函数

$$similar(x_i,x_j) = \frac{x_i \cdot x_j}{\|x_i\|\,\|x_j\|} \tag{2}$$

其中，$\|\cdot\|$是欧几里得L2范数。

3.2　变分图注意自编码器模块

如前所述，学习有效的网络结构信息对深度聚类非常重要，为此我们设计了变分图注意自编码器模块。变分图注意自编码器由变分图注意编码器和内积解码器组成，其结构如图2所示。

(1)变分图注意编码器。为了度量各种邻居的重要性，使用了图注意卷积[29]，即：

$$m_i^{l+1} = \sigma\Big(\sum_{j\in N_i}\alpha_{ij}\boldsymbol{W}^l m_j^l\Big) \tag{3}$$

其中，m_i^{l+1} 为第 $l+1$ 层节点 i 的输出表示；N_i 为节点 i 的邻居；α_{ij} 是注意系数，表示邻居节点 j 对节点 i 重要性；σ 是非线性函数。为了计算注意系数 α_{ij}，我们从属性值和拓扑距离两个方面来度量邻居节点 j 的重要性。

(1)从属性值的角度来看，注意系数 α_{ij} 可以表示为 x_i 和 x_j 的拼接与 $\vec{a}\in R^{2m'}$ 构成的前馈神经网络，即：

$$e_{ij}=\vec{a}^{\mathrm{T}}(\boldsymbol{W}x_i,\boldsymbol{W}x_j) \tag{4}$$

其中，$x_i\in R^m$ 代表节点 i 的属性，$x_j\in R^m$ 代表节点 j 的属性，权重矩阵 $\boldsymbol{W}\in R^{m'\times m}$。

(2)在拓扑结构上，常规方法[29]只考虑 1 阶相邻节点，为了增加图中节点之间的连接性，我们使用图(邻接矩阵)的幂 G^k 来增加图的连通性，该操作在距离最多 k 跳的节点之间建立联系[30]。注意系数矩阵与邻接矩阵有着相似的特征，都可以反映节点之间的连接关系，我们在注意系数矩阵掩码后进行幂运算，新的注意系数可以表示为：

$$\alpha_{ij}=(\alpha_{ij})^k_{\mathrm{mask}} \tag{5}$$

其中，设置 $k=2$。

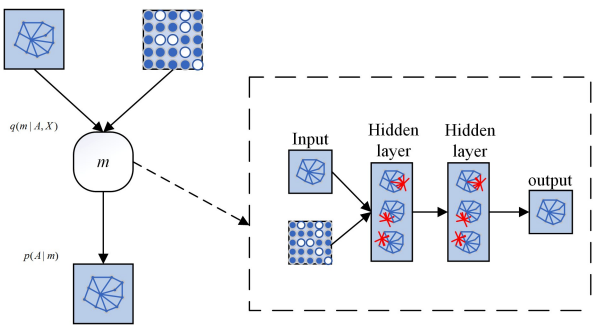


图 2 变分图注意自编码器

Fig. 2 Varitional graph attention self-encoder

另外，变分图注意编码器学习有效结构表示的同时，还可以学习数据的分布。利用后验概率得到隐变量 m_i：

$$q(m_i|\boldsymbol{X},\boldsymbol{A})=N(m_i|\boldsymbol{\mu}_i,\mathrm{diag}(\boldsymbol{\sigma}_i^2)) \tag{6}$$

其中，$\boldsymbol{\mu}_i$ 是由图注意卷积层得到的均值向量，$\boldsymbol{\sigma}_i$ 是由图注意卷积层得到的标准差向量，为了便于计算，采用 $\log\boldsymbol{\sigma}_i$。

(2)内积解码器。利用编码器部分学习到的 m_i 进行内积运算，以重构邻接矩阵：

$$p(\boldsymbol{A}|m_i)=\prod_{i=1}^{N}\prod_{j=1}^{N}p(A_{ij}|m_i,m_j) \tag{7}$$

其中，$\hat{A}_{ij}=p(A_{ij}=1|m_i,m_j)=\mathrm{sigmoid}(m_i^{\mathrm{T}}m_j)$。该模块的优化目函数包括重构邻接损失和后验概率分布与标准正态分布的相对熵：

$$L_{A+P}=E_{q(m_i|\boldsymbol{X},\boldsymbol{A})}[\log p(\boldsymbol{A}|m_i)]-\mathrm{KL}[q(m_i|\boldsymbol{X},\boldsymbol{A})\parallel p(m_i)] \tag{8}$$

其中，$p(m_i)=\Pi_i N(m_i|0,\boldsymbol{I})$表示标准正态分布。

3.3 自编码器模块

变分图注意自编码器模块能够从网络中学习有用的结构信息，例如 $m^{(1)}$，$m^{(2)}$，$m^{(3)}$，然而这可能忽略了数据自身的特征信息。本节将介绍如何使用自编码器模块将结构信息进行集成，那么自编码器可学习的表示将能够适应两种不同的信息，即数据本身和数据之间的关系。本文采用栈式自编码器，以无监督的方式学习原始数据的表示。栈式自动编码器由编码器和解码器组成[26]。

编码器：用于从输入数据中提取特征信息。为了有效利用变分图自编码器学习到的结构表示，我们通过一个权重因子 λ 将结构表示融合到自编码器的特征表示学习中，组合成更完整、更强大的表示。

$$\tilde{h}^{(l)}=(1-\lambda)m^{(l)}+\lambda h^{(l)} \tag{9}$$

其中，$\tilde{h}^{(l)}$ 代表第 l 层新的表示；$h^{(l)}$ 代表自编码器模块第 l 层学习的特征表示；$m^{(l)}$ 代表变分自编码器模块第 l 层学习的结构表示；λ 是一个权重因子，设置 $\lambda=0.5$。通过这种方式，我们按层连接图注意卷积层和全连接层。然后使用 $\tilde{h}^{(l)}$ 作为编码器第 $l+1$ 层的输入，以产生表示 $h^{(l+1)}$。

$$h^{(l+1)}=\sigma(\boldsymbol{W}_{\mathrm{enc}}{}^{(l+1)}\tilde{h}^{(l)}+b_{\mathrm{enc}}{}^{(l+1)}) \tag{10}$$

其中，$\sigma(\cdot)$表示激活函数，例如 $ReLU(\cdot)=\max(0,\cdot)$；$Tanh(\cdot)=\max(-1,1)$[28]；$l$ 表示层数；$\boldsymbol{W}_{\mathrm{enc}}^{(l+1)}$ 和 $b_{\mathrm{enc}}^{(l+1)}$ 分别是编码器第 $l+1$ 层的权重矩阵和偏差，编码器部分输出为 $\boldsymbol{z}$，$h^{(0)}=\boldsymbol{X}$。

注意，第一层全连接层输入是原始特征 $\boldsymbol{X}$：

$$h^{(1)}=\sigma(\boldsymbol{W}_{\mathrm{enc}}{}^{(1)}\boldsymbol{X}+b_{\mathrm{enc}}{}^{(1)}) \tag{11}$$

解码器：将嵌入特征 $\boldsymbol{z}$ 重构输入数据，定义为：

$$h^{(l+1)}=\sigma(\boldsymbol{W}_{\mathrm{dec}}{}^{(l+1)}\boldsymbol{z}+b_{\mathrm{dec}}{}^{(l+1)}) \tag{12}$$

其中，$\boldsymbol{W}_{\mathrm{dec}}^{(l+1)}$ 和 $b_{\mathrm{dec}}^{(l+1)}$ 分别是解码器第 $l+1$ 层的权重矩阵和偏差，解码器部分的输出为重构输入 $\tilde{\boldsymbol{X}}$。自编码器最小化原始数据和重构输入之间的误差，该模块的优化函数：

$$L_r=\frac{1}{N}\parallel\boldsymbol{X}-\tilde{\boldsymbol{X}}\parallel_2^2 \tag{13}$$

3.4 自监督聚类模块

至此，我们已经将自编码器和变分图注意自编码器连接起来。然而，聚类是一个无监督过程，不存在标签指导，因此在训练期间不能获得关于所学习的表示是否被很好地优化的反馈。为了解决这个问题，我们尝试使用自监督机制优化自编码器学习到的联合特征，并使用温和的 KL 目标函数优化[10]：

$$L_c=\mathrm{KL}(P\parallel Q)=\sum_i\sum_j p_{ij}\log\frac{p_{ij}}{q_{ij}} \tag{14}$$

其中，q_{ij} 可以被视为每个节点的软标签分配分布，通过学生 t 分布测量的嵌入点 z_i 和聚类中心 u_j 之间的相似性，设置 $\nu=1$。

$$q_{ij}=\frac{(1+\parallel z_i-\mu_j\parallel^2/\nu)^{\frac{-\nu+1}{2}}}{\sum_k(1+\parallel z_i-\mu_k\parallel^2)^{\frac{-\nu+1}{2}}} \tag{15}$$

而 p_{ij} 的目标分布定义为：

$$p_{ij}=\frac{q_{ij}^2/\sum_i q_{ij}}{\sum_k(q_{ik}^2/\sum_i q_{ik})} \tag{16}$$

其中，高概率的软分配(靠近簇中心的节点)在 Q 中被认为是可信的。因此目标分布 P 将提升 Q 到二次幂，以强调那些“高置信分配”的作用。最小化聚类损失迫使当前分布 Q 逼近目标分布 P，从而将这些“高置信分配”设置为软标签来监督的表征学习。

我们联合优化 FVGTAEDC 表征学习和聚类，将总目标函数定义为：

$$L_{\mathrm{total}}=\alpha L_c+\beta L_r+(1-\beta)L_{A+P} \tag{17}$$

其中，$\alpha>0$ 是控制聚类损失超参数，$\beta>0$ 是平衡重构数据损失、重构邻接损失、后验概率分布与标准正态分布的相对熵的超参数。

3.5 优化

我们首先预训练变分图注意编码器，以获得经过良好训练的表征 $m^{(l)}$，如 3.2 节所述。然后将训练好的 $m^{(l)}$ 通过传递因子 λ 集成到自动编码器，获得更加完整和强大的特征 z。通过重构函数和自训练聚类目标函数改善此特征。为了初始化聚类中心，我们在特征 z 上执行标准 k 均值聚类，以获得 k 个初始质心 $\{\mu_j\}_{j=1}^{k}$。具体来说，有 5 种参数需要更新：变分图注意编码器的权重 $\boldsymbol{W}^{(l)}$、聚类中心 u、自动编码器的权重 $\boldsymbol{W}_{\text{enc}}$，$\boldsymbol{W}_{\text{dec}}$ 和目标分布 P。

给定学习率为 φ，变分图注意编码器 $\boldsymbol{W}^{(l)}$ 更新如下：

$$\boldsymbol{W}^{(l)}=\boldsymbol{W}^{(l)}-\varphi(1-\beta)\frac{\partial L_{A+P}}{\partial \boldsymbol{W}^{(l)}} \tag{18}$$

在固定目标分布 P 和给定样本 N 的情况下，聚类中心 u_j 相对于 L_c 的梯度可以计算为：

$$\frac{\partial L_c}{\partial u_j}=2\sum_{i=1}^{N}(1+\|z_i-u_j\|^2)^{-1}(q_{ij}-p_{ij})(z_i-u_j) \tag{19}$$

给定学习率为 φ，u_j 更新如下：

$$u_j=u_j-\varphi\frac{\partial L_c}{\partial u_j} \tag{20}$$

自动编码器的权重 $\boldsymbol{W}_{\text{enc}}$ 和 $\boldsymbol{W}_{\text{dec}}$ 的更新如下：

$$\boldsymbol{W}_{\text{enc}}=\boldsymbol{W}_{\text{enc}}-\lambda\left(\beta\frac{\partial L_r}{\partial \boldsymbol{W}_{\text{enc}}}+\alpha\frac{\partial L_c}{\partial \boldsymbol{W}_{\text{enc}}}\right) \tag{21}$$

$$\boldsymbol{W}_{\text{dec}}=\boldsymbol{W}_{\text{dec}}-\lambda\left(\beta\frac{\partial L_r}{\partial \boldsymbol{W}_{\text{dec}}}\right) \tag{22}$$

根据式(15)和式(16)更新目标分布 P，经过迭代优化以后，我们从最后一个优化的 Q 中获得我们的聚类结果，并且分配给样本 i 的标签为：

$$l_i=\arg\max_u q_{iu} \tag{23}$$

4 实验

4.1 实验数据

我们提出的模型是在 5 个数据集上评估的。这些实验数据集如表 1 所列。

(1)USPS[32]：USPS 数据集包含 9298 幅 16×16 像素大小的灰度手写数字图像。这些特征是图像中像素点的灰度值，所有特征都归一化为[0,1]。

(2)HHAR[33]：异质性人类活动识别(HHAR)数据集包含来自智能手机和智能手表的 10299 条传感器记录。所有的样本被划分为 6 类人类活动，包括骑躺、坐、站、走、上楼梯和下楼梯。

(3)ACM[20]：来自 ACM 数据集的论文网络。如果两篇论文是由同一作者撰写，它们之间存在边。论文特征是关键词的词袋。我们选择了在 KDD，SIGMOD，SIGCOMM 和 MobiCOMM 发表的论文，并按研究领域将论文分为 3 类(数据库、无线通信、数据挖掘)。

(4)DBLP[20]：来自 DBLP 数据集的作者网络。如果两位作者是合著者，他们之间就有边连接。作者分为 4 个领域：数据库、数据挖掘、机器学习和信息检索。我们根据每个作者提交的会议来标记他们的研究领域。作者特征是由关键词组成的单词包的要素。

(5)Citeseer[20]：引用网络，包含每个文档的稀疏单词包特征向量和文档之间的引用链接列表。标签包含 6 个区域：代理、人工智能、数据库、信息检索、机器语言和人机交互。

表 1　实验数据集

Table 1　Experimental datasets

Dataset	Type	Samples	Classes	Dimension
USPS	Image	9298	10	256
HHAR	Record	10299	6	561
ACM	Text	3025	3	1870
DBLP	Graph	4058	4	334
Citeseer	Graph	3327	6	3703

4.2 比较方法

将本文提出的模型方法与 3 种类型的方法进行了比较，包括传统聚类方法、深度聚类方法和深度图聚类方法。

(1)K-means[6]：基于原始数据的经典聚类方法。

(2)DEC[10]：一种深度聚类方法，它设计了一个聚类目标来指导数据表示的学习。

(3)IDEC[11]：这种方法给 DEC 增加了一个重构损失，以便学习更好的表示。

(4)GAE，VGAE[15]：这是一种使用 GCN 学习数据表示的无监督图形嵌入方法。

(5)DAEGC[16]：使用一个注意力网络来学习节点表示，并使用一个聚类损失来监督图聚类的过程。

(6)SDCN[20]：通过传递算子将自编码器学习的数据表示与图卷积网络学习的结构表示进行融合，并设计了一种双重自监督的方法来监督聚类过程。

(7)FVGTAEDC：本文提出的方法。

4.3 评估指标与参数设置

对于聚类结果，我们使用常用的评估方式来衡量算法的性能：准确率(ACC)，正则互信息量(NMI)，兰德指数(ARI)。

(1)准确率(ACC)

准确率用于表示被正确分类的样本。

$$ACC=\frac{\sum_{i=1}^{N}\delta(map(l_i)=y_i)}{N} \tag{24}$$

其中，δ 是一个指示函数；l_i 是算法得到的 x_i 的聚类标签；y_i 是 x_i 的真实标签；map 是一个转化函数，将每种聚类标签 l_i 整体映射到一个类别，使得预测标签整体与真实标签更接近，该过程可以通过 Kuhn-Munkres 算法得到。

(2)正则互信息量(NMI)

正则互信息量是另一个常用的聚类效果度量方式，用于表示在知道预测的聚类结果后，能够获得的真实标签信息量正则化量。

$$NMI(Y,L)=\frac{I(Y,L)}{\sqrt{H(Y)H(L)}} \tag{25}$$

其中，Y 和 L 对应真实标签和预测标签。H 是信息熵函数，$\sqrt{H(Y)H(L)}$ 是正则化 l 项。

(3)兰德指数(ARI)

兰德指数用来衡量两个数据分布的吻合程度。

$$ARI=\frac{RI-E[RI]}{\max(RI)-E[RI]} \tag{26}$$

$$RI=\frac{a+b}{C_n^2} \tag{27}$$

其中，n 表示样本总数，C 表示正确类别划分，K 表示预测聚类结果。定义 a 为在 C 中被划分为同一类，在 K 中被划分为同一簇的实例对数量。定义 b 为在 C 中被划分为不同类别，

在 K 中被划分为不同簇的实例对数量。ARI 的取值范围为[−1,1],值越大聚类效果越好。

我们将自编码器的尺寸设置为 d-500-500-2000-10,其中 d 是输入数据的尺寸,图注意卷积层的尺寸设置为 500-500-2000。对于非图数据(USPS,HHAR),由于需要基于数据相似度构建邻接矩阵,底层结构信息并不明确,我们用 200 次迭代训练;对于图数据(ACM,DBLP,Citeseer),由于本身含有邻接矩阵,比非图包含更多的信息,我们用 50 次迭代训练。对自编码器预训练时,批处理大小设置为 256,学习率统一设置为10^{-3},超参数统一设置为 $\alpha=0.2$,$\beta=0.5$,$k=5$。对于 K 均值算法生成聚类分配时,我们初始化 20 次;最后运行方法 10 次,并报告平均结果,以防止极端情况。

4.4 评估指标与参数设置

表 2 列出了 5 个数据集上的聚类结果。基线方法的结果取自文献[20]。SDCN* 代表基线的最佳结果。请注意,对于非图数据 USPS 和 HHAR,我们使用基于原始数据构建的图作为模型模块的输入;而对于图数据 ACM,DBLP 和 Citeseer,我们使用原始图作为模型模块的输入。

表 2 5 个数据集的聚类结果(均值)

Table 2 Clustering results (mean) of five data sets

Dataset	Metric	K-means	DEC	IDEC	GAE	VGAE	DAEGC	SDCN*	FVGTAEDC
USPS	*ACC*	0.6682	0.7331	0.7622	0.6310	0.5619	0.7355	**0.7808**	0.7772
	NMI	0.6263	0.7058	0.7556	0.6069	0.5108	0.7112	0.7951	**0.7955**
	ARI	0.5455	0.6370	0.6786	0.5030	0.4096	0.6333	0.7184	**0.7185**
HHAR	*ACC*	0.5998	0.6939	0.7105	0.6233	0.7130	0.7651	0.8426	**0.8679**
	NMI	0.5886	0.7291	0.7419	0.5506	0.6295	0.6910	0.7990	**0.8152**
	ARI	0.4609	0.6125	0.6283	0.4263	0.5147	0.6038	0.7284	**0.7570**
ACM	*ACC*	0.6731	0.8433	0.8512	0.8452	0.8413	0.8694	**0.9045**	0.8958
	NMI	0.3244	0.5454	0.5661	0.5538	0.5320	0.5618	**0.6831**	0.6572
	ARI	0.3060	0.6064	0.6216	0.5946	0.5772	0.5935	**0.7391**	0.7167
DBLP	*ACC*	0.3865	0.5816	0.6031	0.6121	0.5859	0.6205	0.6805	**0.7736**
	NMI	0.1145	0.2951	0.3117	0.3080	0.2692	0.3249	0.3950	**0.4610**
	ARI	0.0697	0.2392	0.2537	0.2202	0.1792	0.2103	0.3915	**0.5055**
Citeseer	*ACC*	0.3932	0.5589	0.6049	0.6135	0.6097	0.6454	0.6596	**0.6759**
	NMI	0.1694	0.2834	0.2717	0.3463	0.3269	0.3641	0.3871	**0.4042**
	ARI	0.1343	0.2812	0.2570	0.3355	0.3313	0.3778	0.4017	**0.4317**

注:粗体数字代表最佳结果

可以看到,本文方法明显优于大多数基线方法的评估指标。我们可以从这些结果中观察到:

(1)基于深度聚类的方法往往好过基于传统聚类的方法,例如 DEC 和 IDEC 的结果优于 K-means,这说明将原始数据降维到低维特征空间能够学到良好的表示。

(2)在基于原始数据构建图的数据上,基于深度聚类的方法的聚类结果通常优于基于深度图聚类的方法,例如 HHAR 和 USPS 数据集上,DEC 和 IDEC 的结果优于 GAE 和 VGAE,这说明在图结构信息不明确的情况下,节点信息往往比结构信息对聚类的影响更大。

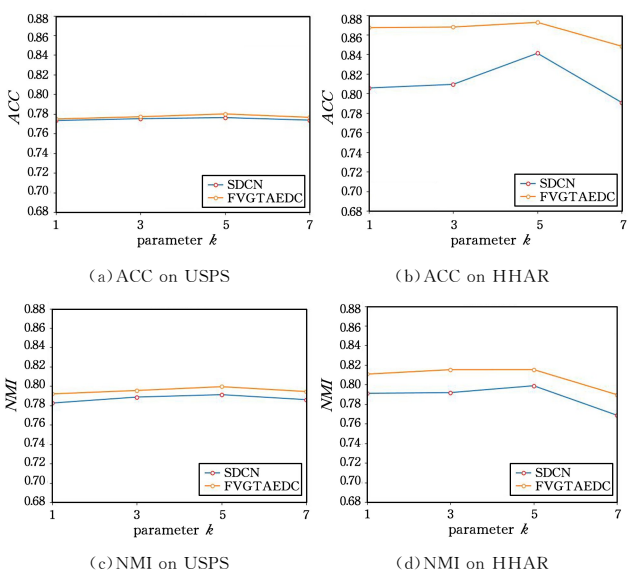
(3)在图结构信息明确的情况下,基于深度图聚类的方法往往优于基于深度聚类的方法,例如 ACM,DBLP,Citesee 数据集上,GAE,VGAE,DAEGC 的结果优于 DEC,IDEC,这说明基于深度图聚类的方法可以有效利用结构信息来进行表征学习。

(4)同时使用结构信息和节点信息的方法通常比只使用信息的一面的方法表现更好,在 5 个数据集上,SDCN 和 FVGTAEDC 优于基线中使用信息的一面的方法。这表明,图形结构和节点内容都包含对聚类有用的信息,并说明了捕捉双方信息之间的相互作用的重要性。

(5)值得一提的是,FVGTAEDC 明显优于最佳基线方法。例如,在 HHAR 数据集上,本文方法相对于最佳基线方法的 ACC,NMI 和 ARI 分别提高了 3.00%,2.02% 和 3.92%,在 DBLP 和 Citeseer 数据集上的提高更大,ACC,NMI 和 ARI 平均提高了 8.08%,10.57% 和 18.30%。其原因是 FVGTAEDC 成功将结构信息集成到深度聚类中,自监督模块指导变分图自编码器和自编码器的更新,使其互相增强。

4.5 K 值敏感性分析

针对非图数据构造邻接矩阵时,K 值是一个非常重要的参数。为了寻找我们的方法在非图数据上的最佳 k 值,同时也为了证明我们的模型具有良好的鲁棒性将 FVGTAEDC 与最佳基线方法 SDCN 进行比较,结果如图 3 所示。



(a)ACC on USPS (b)ACC on HHAR

(c)NMI on USPS (d)NMI on HHAR

图 3 不同 k 值对聚类的影响

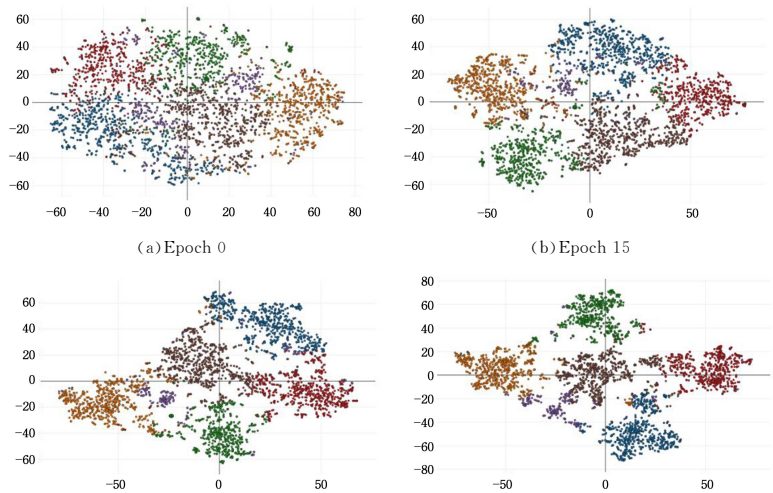
Fig. 3 Influence of different k values on clustering

从图 3 中,我们可以发现,FVGTAEDC 在 $K=5$ 的情况下可以达到最好的性能,并且在 $K=7$ 的情况下,性能会明显下降。原因是当 $K=7$ 时,根据数据相似性构建图中的结构出现明显重叠。另外,在 $K=\{1,3,5,7\}$ 的情况下,FVGTAEDC 比 SDCN 都有提高,FVGTAEDC 可以在具有不同最

近邻数的图上获得稳定的结果，这证明了我们的模型具有良好的鲁棒性。

4.6　聚类可视化分析

我们通过在训练期间对学习的嵌入应用 t-SNE 算法[31]来可视化二维空间中的 Citeseer。图 4 中的结果表明，预训练后变分图注意自编码器的嵌入已经是有意义的了。随后，通过应用自编码器和自监督聚类，随着训练的进行，嵌入变得更加明显，重叠更少，并且每组节点逐渐聚集在一起。



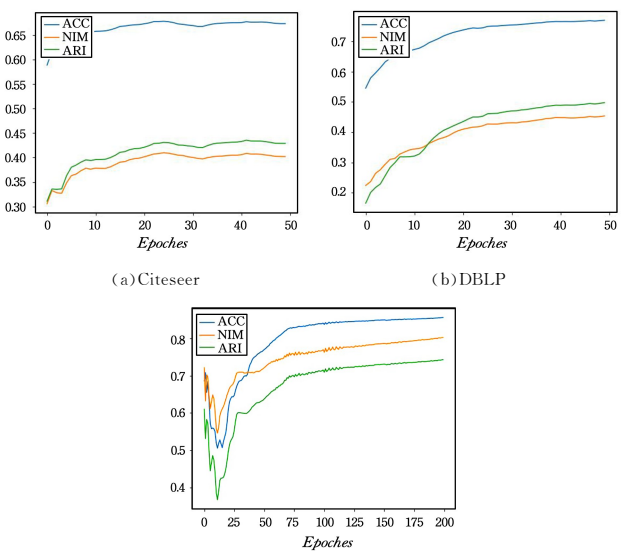
(a) Epoch 0　(b) Epoch 15　(c) Epoch 30　(d) Epoch 50

图 4　Citeseer 数据集上 FVGTAEDC 算法的 2D 可视化

Fig. 4　2D visualization of the FVGTAEDC algorithm on the Citeseer dataset

4.7　训练过程中聚类指标的变化

本节中分析不同数据集的训练过程，具体来说，我们想探索聚类评估指标如何随着迭代次数而变化。在图 5 中，蓝线、黄线、绿线分别代表 ACC，NMI，ARI。大多数情况下，我们可以看到每个评估指标都有明显的上升趋势。该结果表明模型结构和优化算法都朝着期望的方向工作。然而在 HHAR，ACM 数据集上，我们发现在开始阶段时 3 种指标在不同的范围内都有所下降。这可能是由于自编码器和变分图注意自编码器学习的信息不同，从而会在两个模块的结果之间产生冲突，使得聚类结果下降，之后随着迭代次数的增加，FVGTAEDC 聚类结果趋于稳定，没有明显的波动，这也表明我们提出的模型具有良好的鲁棒性。



(a) Citeseer　(b) DBLP　(c) ACM

图 5　训练期间，聚类指标随迭代次数的变化(电子版为彩色)

Fig. 5　During training, the clustering index changes with the number of iterations

结束语　本文尝试将深度图聚类方法与深度聚类方法进行结合。我们提出了一种融合变分图注意自编码器的深度聚类模型，由自编码器模块、变分图注意自编码器模块和自监督聚类模块组成。我们的模型能够学习集成结构和内容信息的表示特征，并利用了一个自监督聚类组件，它从“高置信”分配中生成软标签来监督表示特征的更新。在多个开放数据集上，我们提出的方法优于现有的深度聚类方法。

参 考 文 献

[1] YANG B, LIU D Y, LIU J M, et al. Complex network clustering method[J]. Journal of Software, 2009, 20(1): 54-66.

[2] AHARTIGAN J, WONG M A. Algorithm AS 136: A k-means clustering algorithm. Journal of the Royal Statistical Society[J]. Series C (Applied Statistics), 1979, 28(1): 100-108.

[3] CHANG J L, WANG L F, MENG G F, et al. Deep Adaptive Image Clustering[C]// IEEE International Conference on Computer Vision. 2017: 5880-5888.

[4] CAGGARWAL C, ZHAI C X. A survey of text clustering algorithms[C]// Mining Text Data. Springer. 2012: 77-128.

[5] SHANG J W, WANG C K, XIN X, et al. Community discovery algorithm based on deep sparse autoencoder[J]. Journal of Software, 2017, 28(3): 648-662.

[6] ARTHUR D, ASSILVITSKII S V. k-means++: The advantages of careful seeding[C]// SODA. 2007: 1027-1035.

[7] ESTER M, KRIEGEL H P, SANDER J, et al. A density-based algorithm for discovering clusters in large spatial databases with noise[C]// KDD. 1996: 226-231.

[8] POUYANFAR S, SADIQ S, YAN Y L, et al. A Survey on Deep Learning: Algorithms, Techniques, and Applications[C]// ACM Computing Surveys. 2019: 1-36.

[9] TIAN F, GAO B, CUI Q, et al. Learning Deep Representations for Graph Clustering[C]// AAAI. 2014: 1293-1299.

(下转第 116 页)

ting Technology and Automation, 2019(4): 144-150.
[13] WANG H, LE Z C, GONG X, et al. Link Prediction of Complex Networks is Analyzed from the Perspective of Informatics[J]. Journal of Chinese Computer Systems, 2020, 41(2): 316-326.
[14] BAI H, MA Y L, BI Y, et al. A Complicated Network Link Prediction Algorithm Based on Local Similarity of Nodes[J]. Computer Applications and Software, 2020, 37(5): 298-301.
[15] LIU S X, LI X, CHEN H C, et al. Link prediction method based on matching degree of resource transmission for complex network[J]. Journal on Communications, 2020, 41(6): 70-79.
[16] QI F P, WANG T, FU Z Q. Link prediction in complex networks based on mutual information[J]. Journal of University of Science and Technology of China, 2020, 50(1): 57-63.
[17] REVELLE M, DOMENICONI C, SWEENEY M, et al. Finding Community Topics and Membership in Graphs[C]// Joint European Conference on Machine Learning and Knowledge Discovery in Databases. 2015: 625-640.

HUANG Shou-meng, born in 1975, master, associate professor. His main research interests inclue information technology and information security.

(上接第 87 页)

[10] XIE J Y, GIRSHICK R, FARHADI A. Unsupervised deep embedding for clustering analysis[C]// ICML. 2016: 478-487.
[11] GUO X F, GAO L, LIU X W, et al. Improved deep embedded clustering with local structure preservation[C]// IJCAI. 2017: 1753-1759.
[12] PENG X, XIAO S J, FENG J S, et al. Deep subspace clustering with sparsity prior[C]// IJCAI. 2016: 101-115.
[13] JIANG Z X, ZHENG Y, TAN H C, et al. Variational deep embedding: An unsupervised and generative approach to clustering [C]// IJCAI. 2017: 4305-4324.
[14] NKIPF T, WELLING M. Semi-supervised classification with graph convolutional networks[C]// ICLR. 2017: 1-14.
[15] NKIPF T, WELLING M. Variational graph auto-encoders[J]. NIPS, 2016, 21(11): 1-3.
[16] WANG C, PAN S R, HU R Q, et al. Attributed Graph Clustering: A Deep Attentional Embedding Approach[C]// IJCAI. Marina del Rey CA USA: Association for the Advancement of Artificial Intelligence (AAAI), 2019: 3670-3676.
[17] LI X L, ZHANG H Y, ZHANG R. Embedding Graph Auto-Encoder with Joint Clustering via Adjacency Sharing[C]// WWW. 2020: 1-11.
[18] WANG C, PAN S R, LONG G D, et al. MGAE: Marginalized Graph Autoencoder for Graph Clustering[C]// ACM on Conference on Information and Knowledge Management. 2017: 889-898.
[19] ZHANG X T, LIU H, LI Q M, et al. Attributed Graph Clustering via Adaptive Graph Convolution[C]// IJCAI. 2019: 4327-4333.
[20] BO D Y, WANG X, SHI C, et al. Structural Deep Clustering Network[C]// WWW. 2020: 1-11.
[21] SUN J G, LIU J, ZHAO L Y. Research on clustering algorithm [J]. Journal of Software, 2008, 19(1): 48-61.
[22] JAIN A K, DUBES R C. Algorithms for clustering data [J]. Technometrics, 1988, 32(2): 227-229.
[23] REYNOLDS D A. Gaussian mixture models[C]// Encyclopedia of Biometrics. 2015: 1-23.
[24] JOHNSON S C. Hierarchical clustering schemes[J]. Psychometrika, 1967, 32(3): 241-254.
[25] NG A Y, JORDAN M I, WEISS Y. On spectral clustering: Analysis and an algorithm[C]// Advances in Neural Information Processing Systems. 2002: 849-856.
[26] HINTON G E, SALAKHUTDINOV R R. Reducing the dimensional ity of data with neural networks [J]. Science, 2006, 7(28): 504-507.
[27] HINTON G E. Learning multiple layers of representation[J]. Science, 2007, 7(4): 428-434.
[28] RAMACHANDRAN P, ZOPH B, LE Q V. Searching For Activation Functions[C]// ICLR. 2018: 1-13.
[29] CASANOVA A, ROMERO A, LIO P, et al. Graph Attention Networks[C]// IJCAI. 2018: 1-12.
[30] CHEPURI S P, LEUS G. Subsampling For Graph Power Spectrum Estimation[C]// IEEE SAM. 2016: 1250-1263.
[31] VAN DER MAATEN L, HINTON G. Visualizing data using t-sne[J]. Journal of Machine Learning Research, 2008, 9(Nov): 2579-2605.
[32] DENKER J, GARDNER W R, GRAF H, et al. Neural Network Recognizer for Hand-Written Zip Code Digits[C]// NIPS. 1988: 323-331.
[33] STISEN A, BLUNCK H, BHATTACHARYA S, et al. Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition [C] // SenSys. ACM, 2015: 127-140.

KANG Yan, born in 1972, Ph.D, associate professor. Her main research interests include transfer learning, deep learning and integrated learning.

LI Hao, born in 1970, Ph.D, professor. His main research interests include distributed computing, grid and cloud computing.