Pure Mathematics 理论数学, 2025, 15(4), 53-61
Published Online April 2025 in Hans. https://www.hanspub.org/journal/pm
https://doi.org/10.12677/pm.2025.154108

Hans汉斯

基于Node2vec与GCN融合的图嵌入优化方法研究

王　玺，鞠　敏

上海理工大学理学院，上海

收稿日期：2025年2月19日；录用日期：2025年3月17日；发布日期：2025年4月9日

摘　要

本文结合了Node2vec和GCN这两种方法，先利用Node2vec方法得到初步的图嵌入，之后将其作为输入利用GCN进一步更新图嵌入矩阵。本文选择在维基数据集上进行节点分类任务，比较了结合前后方法的表现，验证了其有效性。

关键词

图嵌入，Node2vec，GCN

Research on Graph Embedding Optimization Method Based on the Fusion of Node2vec and GCN

Xi Wang, Min Ju

College of Science, University of Shanghai for Science and Technology, Shanghai

Received: Feb. 19th, 2025; accepted: Mar. 17th, 2025; published: Apr. 9th, 2025

Abstract

In this paper, we integrate the Node2vec and GCN methods. Initially, the Node2vec method is employed to obtain preliminary graph embeddings, which are then used as input to further update the graph embedding matrix through GCN. The study selects the Wikipedia dataset for node classification tasks, comparing the performance of the methods before and after integration to validate their effectiveness.

文章引用: 王玺, 鞠敏. 基于 Node2vec 与 GCN 融合的图嵌入优化方法研究[J]. 理论数学, 2025, 15(4): 53-61.
DOI: 10.12677/pm.2025.154108

Keywords

Graph Embedding, Node2vec, GCN

1. 引言

Node2vec 是 Grover Aditya [1]在 Deepwalk 方法的基础上，提出的一种重要的图表示学习算法。在面向对象是图的问题中，很重要的一个部分是对于节点邻居集的选取，常规的方法是直接利用节点在图上的邻居进行信息聚合，然而有些时候与当前节点联系最为紧密的可能不是与其直接相连的节点，因此作者考虑了一种有策略的随机游走方式，通过改变 p 和 q 的值来控制随机游走的倾向，进而得到与当前节点关系更为紧密的序列。

现有对于 Node2vec 的研究，大多都是直接利用算法得到的节点特征进行诸如节点分类，社区检测以及链路预测等下游任务。王慧雪[2]将 Node2vec 方法运用到带有标签的社区网络之中，对算法得到的节点特征直接运用谱聚类的方式进行社区划分。戴怡清[3]将 11 种基于网络结构相似性的方法和 node2vec 表示学习方法进行链路预测实验研究，探索不同方法间的预测效果规律、不同参数对 node2vec 方法预测效果的影响以及各方法间的相关关系。Ha Jihwan [4]提出了一种基于 Node2vec 的神经协同过滤的新框架，用于基于深度神经网络预测 miRNA 疾病关联(NCMD)。还有工作将 Node2vec 运用到推荐系统中[5]，或者对算法本身进行优化[6]。

图卷积神经网络(GCN)同样是近年来崛起的一种图表示学习算法[7]-[9]，相比较于 Node2vec 方法，GCN 不仅考虑了网络的拓扑结构，同时也考虑了每个节点的属性值，显然这更符合实际情况。GCN 的输入是图的邻接矩阵 A 以及节点的属性矩阵 X，输出的是学习得到的图嵌入(Graph Embedding)矩阵，即更为凝练的节点属性特征。然而在实际问题中我们往往无法获取得到矩阵 X，因此我们需要先想办法获取矩阵 X，常见的方法是直接假设 X 是单位矩阵，但这样可能会延缓算法收敛速度。因此我们的想法是能否先训练得到图的初步特征矩阵，再将其作为 GCN 的输入，进而完成更多下游任务。

本文将这两种方法进行结合，首先利用 Node2vec 方法得到节点的特征表示，然后将其作为节点属性矩阵 X 输入到 GCN 之中得到新的节点表示，为后续节点分类任务做准备。

2. 预备知识

2.1. Node2vec

机器学习算法的核心就是希望计算机能够学习到一个复杂的映射，在 Node2vec 中我们要找的是将顶点 v 映射到图嵌入(Graph embedding)的函数 $f(v)$。现假设 $N_S(v)$是节点 v 在采样策略 S 下得到的邻居集合。我们优化的目标就是通过训练和调整参数，使得给定每个点的情况下，使得其邻近节点出现的概率最大，即对问题(2.1)进行优化：

$$\max_f \sum_{v\in V} \log P\left(N_S(v)\middle| f(v)\right) \tag{2.1}$$

邻近性(proximity)是图分析中的一个核心概念，反映了节点之间的联系紧密程度，如图 1 所示。通

常，邻近的节点具有相似的性质，因此在节点分类等任务中，邻近性是重要的分类依据。然而，人为定义的邻近性存在局限性，尤其是在稀疏图中，邻接矩阵无法充分反映节点之间的复杂关系。为此，Node2vec通过随机游走生成节点序列，利用采样策略 S 灵活捕捉节点的局部和全局结构信息，避免了人为定义邻近性的不足。

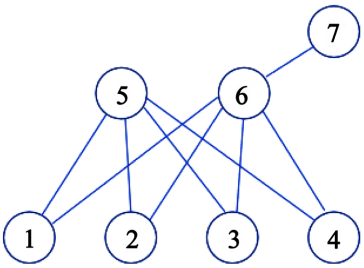

Figure 1. Node Proximity
图 1. 节点邻近性

为了更好的求解问题(2.1)，Grover Aditya 提出了两个假设。

第一个假设是条件独立性假设，即当前节点的邻居集 $N_S(v)$中的节点之间相互独立：

$$P\left(N_S(v)\middle|f(v)\right)=\prod_{v_i\in N_S(v)}P\left(v_i\middle|f(v)\right)$$

另一个假设是特征空间对称性假设，即一个节点作为源节点和邻居节点时公用同一个 embedding。事实上在别的算法中，比如说图表示学习的另一个算法 LINE，同一个节点就有作为源节点和临近节点两个 embedding。在对称性假设下，我们可以得到进一步的条件概率：

$$P\left(v_i\middle|f(v)\right)=\frac{\exp\left(f(v_i)\cdot f(v)\right)}{\sum_{u\in V}\exp\left(f(u)\cdot f(v)\right)}$$

因此问题(2.1)可以转化成以下形式：

$$\max_f\sum_{v\in V}-\log Z_v+\sum_{v_i\in N_S(v)}f(v_i)\cdot f(v).$$

其中 $Z_v=\sum_{u\in V}\exp\left(f(u)\cdot f(v)\right)$。

Node2vec 方法大致可以分为生成序列以及更新节点嵌入这两个步骤。

在生成序列时，我们考虑使用有策略的随机游走去生成节点序列，下面讨论具体的游走策略，如图 2 所示。

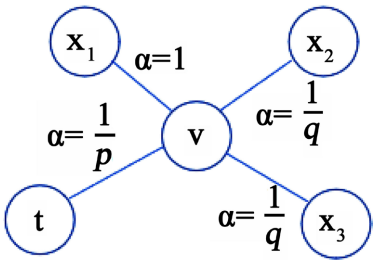

Figure 2. Node2vec Sampling Strategy
图 2. Node2vec 采样策略

一般而言，随机游走的公式都由(2.2)决定：

$$P\left(c_i = x \middle| c_{i-1} = v\right) = \begin{cases} \dfrac{\pi_{vx}}{Z}, (v,x) \in E \\ 0, \text{其它} \end{cases} \tag{2.2}$$

其中 c_i 表示序列中第 i 个节点，v，x 表示图 G 中的点，π_{vx} 表示从点 v 到点 x 的未归一化的转移概率，Z 称之为归一化常数，因此 $P\left(c_i = x \middle| c_{i-1} = v\right)$ 就是归一化后的转移概率。

进一步的，$\pi_{vx} = \alpha_{pq}\left(t,x\right) \cdot w_{vx}$，$w_{vx}$ 是图 G 中边的权重，若 G 为无权图则 $w=1$。$\alpha_{pq}\left(t,x\right)$ 则表示游走时的具体概率，由如下(2.3)式定义。

$$\alpha_{pq}\left(t,x\right) = \begin{cases} \dfrac{1}{p}, d_{tx} = 0 \\ 1, d_{tx} = 1 \\ \dfrac{1}{q}, d_{tx} = 2 \end{cases} \tag{2.3}$$

可以看到算法在随机游走生成序列时，综合考虑了当前节点以及上一步节点的情况。p 值称为返回参数，$d_{tx} = 0$ 表示从当前节点 v 选择下一步返回来时节点 t，p 值越大则返回的概率越小。q 值称为出入参数。

得到随机游走产生的节点序列后，我们就可以使用处理语言模型中常用的 Word2vec 方法，这里我们将节点看作单词，就可以得到对应节点的特征表示。

2.2. GCN

图卷积神经网络(GCN)，是 Kipf [10]提出的专门用于处理图(graph)数据的方法。在这种方法问世之前，人们解决图网络时常常使用卷积神经网络(CNN)，但是 CNN 面向的对象是数据类型更为规整的图像(image)，因此我们需要进行预处理。辛鑫[11]从节点的角度考虑，为每个节点都设计了一个表示图结构的矩阵。然而预处理的方式不同将会直接影响到方法的准确率和效率，并且会增加运算成本，因此 GCN 的出现非常关键。

几乎所有的图表示学习算法的思想都是利用各种方法聚合周围节点信息，以此来更新当前考虑的特征。很自然的一种想法就是利用节点的邻居去进行整合，这也是 GCN 的想法，(2.4)式就是更新公式。

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right) \tag{2.4}$$

其中 l 表示 GCN 的层数，W 是可学习的参数，$\boldsymbol{H}^{(l)}$是第 l 层的属性矩阵，$\boldsymbol{H}^{(0)}$就是输入的属性矩阵 $\boldsymbol{X}$。$\tilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}$ 是邻接矩阵 $\boldsymbol{A}$ 和单位矩阵 $\boldsymbol{I}$ 相加的结果，每一次迭代过程中，我们都直接将当前节点的邻居节点的信息进行相加来更新，而将邻接矩阵 $\boldsymbol{A}$ 和属性矩阵 $\boldsymbol{H}$ 相乘正好可以完成聚合，之所以加入单位矩阵是因为聚合信息的时候需要考虑节点本身。$\tilde{\boldsymbol{D}}$ 是 $\tilde{\boldsymbol{A}}$ 的度矩阵，之所以有是对进行对称归一化，避免节点度大的点聚合过多信息。

3. Node2vec 和 GCN 结合

在上文中我们已经分别对 Node2vec 和 GCN 进行了初步的介绍。接下来我们将说明他们的结合具有显著的优势，尤其是在处理图数据时，能够充分利用两者的互补特性。以下是二者结合的必要性及其理论依据的详细分析：

3.1. Node2vec 的优势与局限性

Node2vec 通过有偏随机游走生成节点序列，并利用 Skip-gram 模型学习节点的嵌入表示。其优势在于：

(1) 捕捉局部和全局结构信息：

通过调整参数 p 和 q，Node2vec 能够灵活地捕捉节点的局部邻居信息(通过深度优先搜索 DFS)和全局结构信息(通过广度优先搜索 BFS)。

(2) 适用于稀疏图：

Node2vec 通过随机游走生成节点序列，能够有效处理稀疏图，尤其是在节点之间直接连接较少的情况下，仍然能够捕捉到高阶邻近性。

尽管如此，Node2vec 仍然具有一些局限性：

(1) 依赖图结构：

Node2vec 生成的节点嵌入仅基于图的结构信息，忽略了节点的属性特征。在实际应用中，节点的属性信息(如文本、图像等)往往对下游任务(如节点分类、链路预测等)至关重要。

(2) 无法动态更新：

Node2vec 生成的节点嵌入是静态的，无法随着图结构的变化而动态更新。

3.2. GCN 的优势与局限性

GCN 通过图卷积操作聚合邻居节点的特征，能够同时利用图的结构信息和节点的属性特征。其优势在于：

(1) 结合结构与属性信息：

GCN 的输入包括图的邻接矩阵 A 和节点特征矩阵 X，能够同时利用图的结构信息和节点的属性信息进行特征学习。

(2) 动态更新：

GCN 通过多层卷积操作，能够逐步聚合更远距离的邻居信息，动态更新节点的表示。

尽管如此，GCN 仍然具有一些局限性：

(1) 依赖初始特征矩阵：

GCN 的性能高度依赖于输入的节点特征矩阵 X，在实际应用中，节点的初始特征矩阵往往难以获取，常见的做法是使用单位矩阵或随机初始化，这可能导致信息不足或收敛速度慢。

(2) 过平滑问题：

当 GCN 的层数过多时，节点的特征会趋于一致，导致“过平滑”现象，尤其是在节点数量较少或图结构较为简单的情况下。

3.3. Node2vec 与 GCN 结合的互补性

Node2vec 和 GCN 的结合能够有效弥补各自的局限性，具体体现在以下几个方面：

(1) 提供高质量的初始特征：

Node2vec 生成的节点嵌入可以作为 GCN 的输入特征矩阵 X，避免了直接使用单位矩阵或随机初始化带来的信息不足问题。Node2vec 通过随机游走生成的节点嵌入已经包含了图的结构信息，能够为 GCN 提供高质量的初始特征。

(2) 增强特征表示能力：

GCN 能够进一步优化 Node2vec 生成的节点嵌入，结合图的结构信息和节点特征进行动态更新。通过多层卷积操作，GCN 能够聚合更远距离的邻居信息，增强节点的特征表示能力。

(3) 提升下游任务性能：

Node2vec 和 GCN 的结合能够充分利用图的结构信息和节点特征，提升下游任务的性能。实验结果表明，结合后的模型在节点分类任务中的准确率显著高于单独使用 Node2vec 或 GCN。

3.4. 结合过程

从数学公式的角度分析，Node2vec 和 GCN 的结合可以表示为以下步骤：

(1) Node2vec 生成初始嵌入：

$$X_{Node2vec} = Node2vec(G) \tag{3.1}$$

其中， $X_{Node2vec}$ 是 Node2vec 生成的节点嵌入矩阵，G 是图的结构信息。

(2) GCN 进一步优化嵌入：

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right), H^{(0)} = X_{Node2vec} \tag{3.2}$$

其中 l 表示 GCN 的层数，W 是可学习的参数，$H^{(l)}$是第 l 层的属性矩阵，$H^{(0)}$就是 Node2vec 生成的初始嵌入属性矩阵。 $\tilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}$ 是邻接矩阵加上自环， $\tilde{\boldsymbol{D}}$ 是 $\tilde{\boldsymbol{A}}$ 的度矩阵， σ 是非线性激活函数。

通过这种方式，Node2vec 和 GCN 的结合能够充分利用图的结构信息和节点特征，提升节点嵌入的质量。

4. 节点分类实验

在实验部分我选择使用维基数据集，数据集中的节点是网页，边表示网页之间的链接情况，一共有 2405 个节点。该数据集给出了网页连接情况以及每个节点所属类别，并未给出每个节点的属性特征信息。因此我们首先使用 Node2vec 方法得到每个节点的 embedding。

4.1. t-SNE 可视化

在第一部分中，我们首先利用 Node2vec 算法求得节点的初步特征表示，因为 p，q 值会影响到节点表示的好坏，因此我们需要先找到合适的这两个值。一般而言，我们都是固定 p 值，调整 q 值，综合测试下来我们最终得到 p = 0.25，q = 4。

t-SNE 是一个较为有效的可视化高维数据的工具，它将数据点之间的相似性转换为联合概率，并试图最小化低维嵌入数据和高维数据联合概率之间的 KL 散度，相比较于主成分分析(PCA)，t-SNE 往往具有更好的效果。

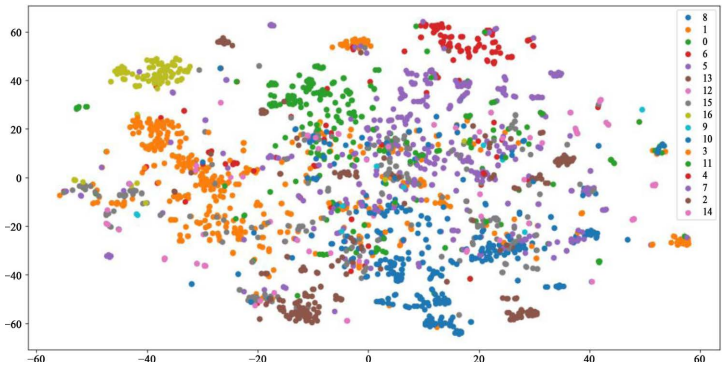

Figure 3. Node2vec Visualization
图 3. Node2vec 可视化

图 3 是我们对 Node2vec 方法得到的节点嵌入进行 t-SNE 可视化，可以看到同种类别的节点被分配在了一起，聚类效果较好。

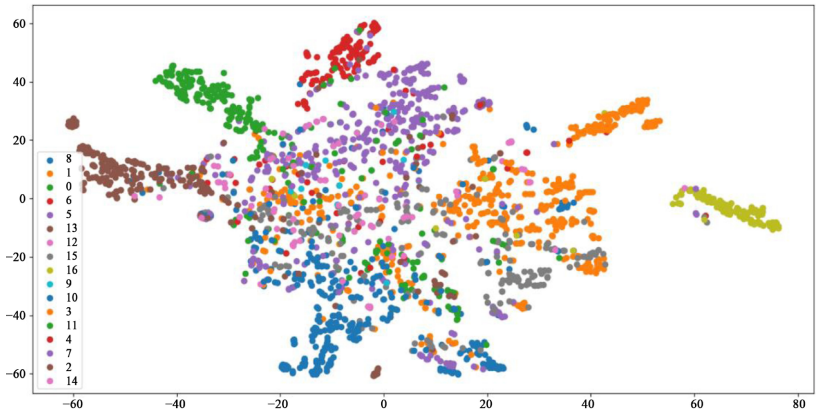

Figure 4. Node2vec-GCN Visualization
图 4. Node2vec-GCN 可视化

第二部分就是将 Node2vec 得到的节点属性矩阵 $\boldsymbol{X}$，利用 GCN 得到节点更好的嵌入，并对节点进行分类。这里我们依然使用 t-SNE 对最终得到的节点嵌入进行可视化，如图 4 所示我们可以看到，相比较于图 3 仅使用 Node2vec 方法，此时同种类型的节点在图中更为紧密，不同类型的节点界限更为明显，由此可知结合两种方法后的节点嵌入具有更好的表现。

4.2. 与其他图嵌入方法的对比

除去可视化，我们还考虑利用不同方法对数据集中节点分类的准确性作为评判标准进行比较。本文考虑与 DeepWalk、GraphSAGE 这些常用的图嵌入方法进行比较。如表 1 所示，如果仅使用 Node2vec 和 GCN 进行节点分类任务准确率分别达到了 68.61%和 67.81%，而将二者进行结合后准确率提升至 72.77%，同时显著高于其他方法。这表明 Node2vec 与 GCN 的结合能够有效提升节点分类的性能。

Table 1. Accuracy of different methods
表 1. 不同方法的准确率

方法	Accuracy
DeepWalk	65.27%
GraphSAGE	70.35%
Node2Vec	68.61%
GCN	67.81%
Node2vec+GCN	72.77%

4.3. 消融实验

为了验证 Node2vec 预训练对最终模型性能的贡献，我们设计了消融实验比较以下两种情况：

(1) 使用 Node2vec 预训练：

首先使用 Node2vec 生成节点嵌入，然后将其作为 GCN 的输入特征矩阵 $\boldsymbol{X}$。

(2) 不使用 Node2vec 预训练：

直接使用单位矩阵作为 GCN 的输入特征矩阵。

从表 2 中可以看出，使用 Node2vec 预训练的模型准确率显著高于直接使用单位矩阵的模型。这表明 Node2vec 预训练能够为 GCN 提供高质量的初始特征，显著提升模型的性能。

除此之外，我们还考虑结合使用 Deepwalk 和 GCN 来进行节点分类任务，这里我们可以看到他的准确率也不如和 Node2vec 结合。

Table 2. Ablation Study
表 2. 消融实验

方法	Accuracy
单位矩阵 + GCN	67.81%
Deepwalk + GCN	70.27%
Node2vec + GCN	72.77%

4.4. GCN 层数的影响

为了进一步分析超参数对实验效果的影响，我们这里选择对网络的层数进行讨论。从表 3 中可以看出，当 GCN 的层数是 1 层时，模型的效果最好，达到了 72.77%。随着 GCN 层数的增加，模型的准确率逐渐下降。这是因为过多的层数会导致“过平滑”问题，无法有效区分不同类别的节点。本文使用数据集仅有 2405 个节点，经过一层 GCN 聚合信息就已经足够，而两层甚至多层 GCN 会使得所有节点的特征趋于一致。

Table 3. Accuracy of different network depths
表 3. 不同网络层数的准确率

方法	Accuracy
Node2vec + GCN (1 层)	72.77%
Node2vec + GCN (2 层)	62.16%
Node2vec + GCN (3 层)	58.25%

4.5. 高/低度节点分类表现

为了进一步分析模型在不同节点上的表现，我们将节点根据其度数分为高度节点(度数高于平均度数的节点)和低度节点(度数低于平均度数的节点)，并分别计算其分类准确率。从表 4 中可以看出，无论是高度节点还是低度节点，Node2vec 与 GCN 结合后的模型准确率均高于单独使用 GCN。然而，低度节点的准确率仍然低于高度节点，这是因为低度节点的邻居信息较少，GCN 在聚合信息时可能无法充分捕捉其结构特征。

Table 4. Accuracy of high/low degree nodes
表 4. 高/低度节点的准确率

节点类型	Accuracy (GCN)	Accuracy (Node2vec + GCN)
高度节点	70.36%	74.67%
低度节点	66.07%	69.32%

5. 结论

节点嵌入的学习一直以来都是图机器学习的重点。然而由于节点初始化特征矩阵往往是随机给定或者用简单的单位矩阵代替，因此可能会影响最终的下游任务结果。本文结合 Node2vec 和 GCN 这两种图

表示学习算法，先利用 Node2vec 方法得到初步的图嵌入，之后将其作为输入利用 GCN 进一步更新图嵌入矩阵，在维基数据集上进行实验，相比于仅使用 Node2vec 在准确性和 t-SNE 降维可视化均有提升。

6. 未来研究方向

尽管本文提出的基于 Node2vec 与 GCN 融合的图嵌入方法在节点分类任务中表现良好，但仍有许多改进和扩展的空间。未来可以探索动态调整 Node2vec 的随机游走策略，结合注意力机制或层次化聚合优化 GCN 的表示能力，并进一步将该方法扩展到图分类、链路预测等复杂任务中。此外，结合 GraphSAGE、GAT 等其他图嵌入方法，或将其应用于社交网络、生物信息学等实际场景，也是值得探索的方向。

参考文献

[1] Grover, A. and Leskovec, J. (2016) node2vec: Scalable Feature Learning for Networks. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, 13-17 August 2016, 855-864. https://doi.org/10.1145/2939672.2939754

[2] 王慧雪. 基于 node2vec 的社区检测方法[J]. 计算机与数字工程, 2020, 48(2): 403-408.

[3] 戴怡清. 基于 node2vec 的科研合作网络链路预测[D]: [硕士学位论文]. 武汉: 武汉大学, 2019.

[4] Ha, J. and Park, S. (2023) NCMD: Node2vec-Based Neural Collaborative Filtering for Predicting miRNA-Disease Association. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, **20**, 1257-1268. https://doi.org/10.1109/tcbb.2022.3191972

[5] 李梦. node2vec 模型在推荐系统中的应用[D]: [硕士学位论文]. 重庆: 西南大学, 2022.

[6] 杜阳阳, 李华康, 李涛. 基于 Node2vec 的改进算法的研究[J]. 计算机技术与发展, 2018, 28(7): 6-10.

[7] 刘向宇, 燕玮, 孟星好, 侯开茂. 一种基于网络表示学习的网络安全用户发现方法[J]. 网络安全与数据治理, 2022, 41(7): 78-82.

[8] 姚锐. 采用 Node2Vec 模型对网络特征表示方法研究[D]: [硕士学位论文]. 南京: 南京大学, 2018.

[9] 杜瑾, 熊回香, 王妞妞. 融合多元网络与网络表示学习的科研合作者推荐研究[J]. 情报资料工作, 2022, 43(4): 27-35.

[10] Thomas, N. (2016) Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks.

[11] Xin, X., Wang, C., Ying, X. and Wang, B. (2017) Deep Community Detection in Topologically Incomplete Networks. *Physica A: Statistical Mechanics and its Applications*, **469**, 342-352. https://doi.org/10.1016/j.physa.2016.11.029