Heterogeneous Graph Neural Network

Chuxu Zhang
University of Notre Dame
czhang11@nd.edu

Dongjin Song
NEC Laboratories America, Inc.
dsong@nec-labs.com

Chao Huang
University of Notre Dame, JD Digits
chuang7@nd.edu

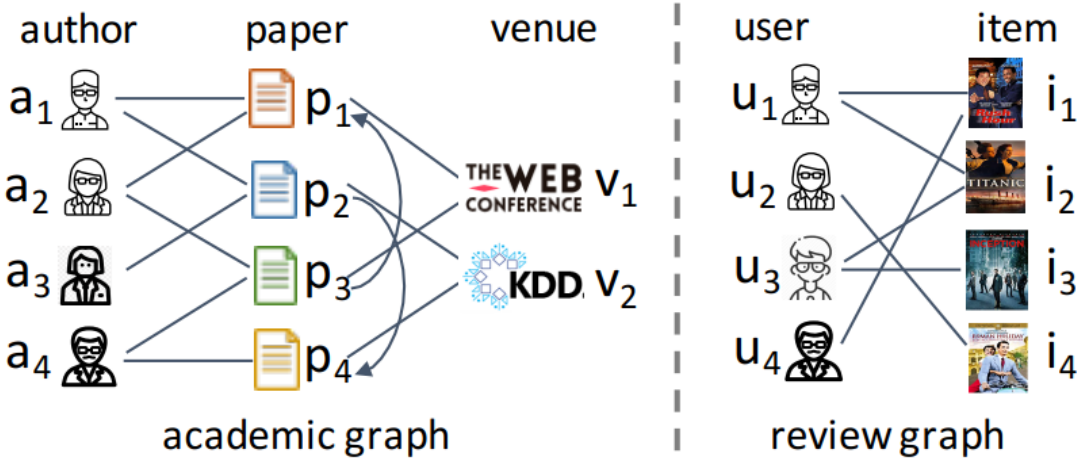
Ananthram Swami
US Army Research Laboratory
ananthram.swami.civ@mail.mil

Nitesh V. Chawla
University of Notre Dame
nchawla@nd.edu

KDD2019

问题背景

异质图内含丰富的信息，包括不同类型的节点，不同类型的边，以及节点的结构信息等。



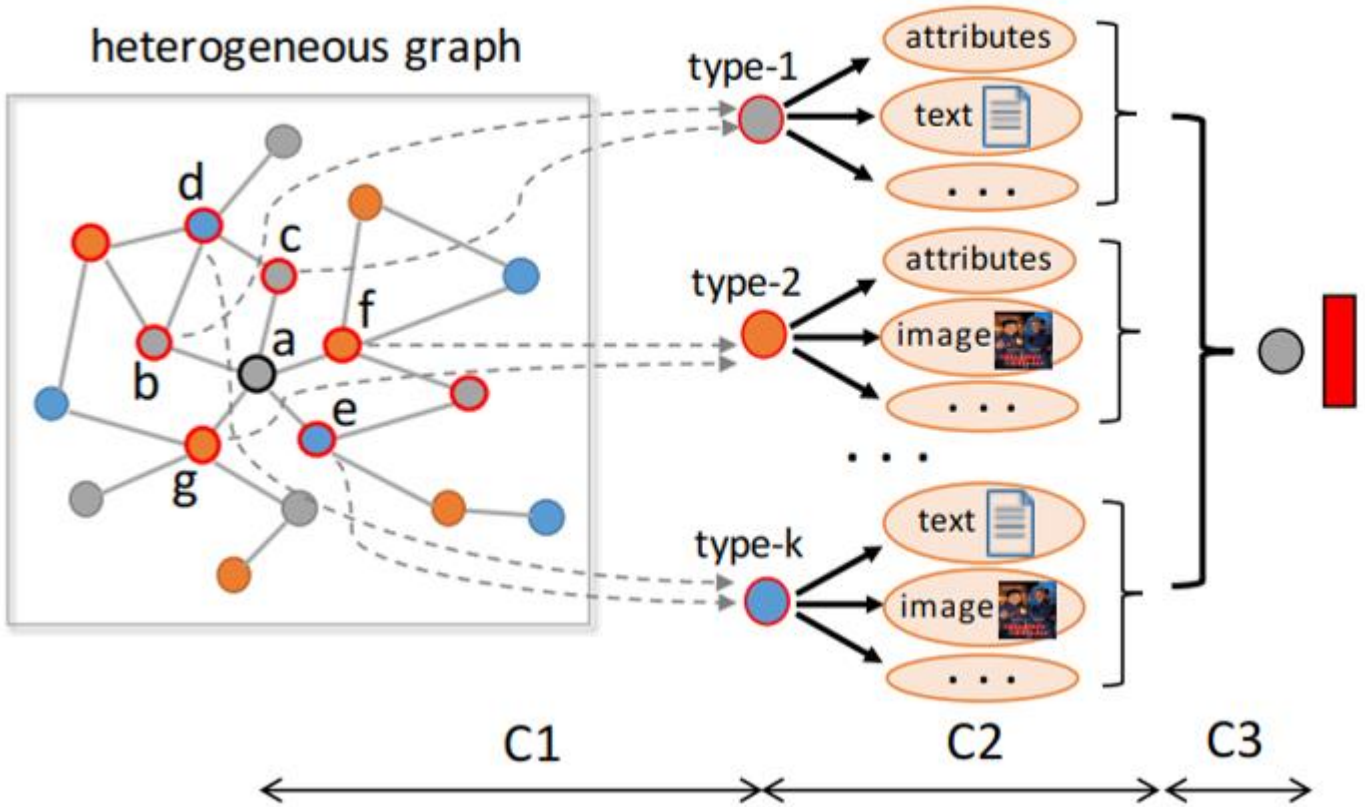
已知的基于异构图的方法有：

- 传统的方法：人为设计特征工程抽取出特征向量。
  缺点：针对具体下游任务，应用受限，不能一般化推广到其他任务。
- 表示学习方法：自动进行特征工程，可适用于多种下游任务。
  - 浅层模型：DeepWalk
  - 语义感知方法(semantic-aware aproaches)：metapath2vec
  - 上下文感知方法(content-aware approaches)：ASNE，使用隐层的特征和属性信息以学习到节点嵌入。
- GNN方法：GCN、GraphSAGE、GAT。

问题与挑战:

在上述方法中，基于GNN的方法效果最好， 但GNN的研究进展和应用主要集中在同构图上。目前最先进的GNN还没有很好地解决异构图面临的以下问题:

C1：如何为异构图中的节点采样到强相关的异质邻居?
C2：如何为异构图中带有异质内容信息的不同节点设计节点内容信息的encoder?
C3：如何在聚合异质邻居特征信息的过程中考虑不同节点类型的影响?



方法与贡献

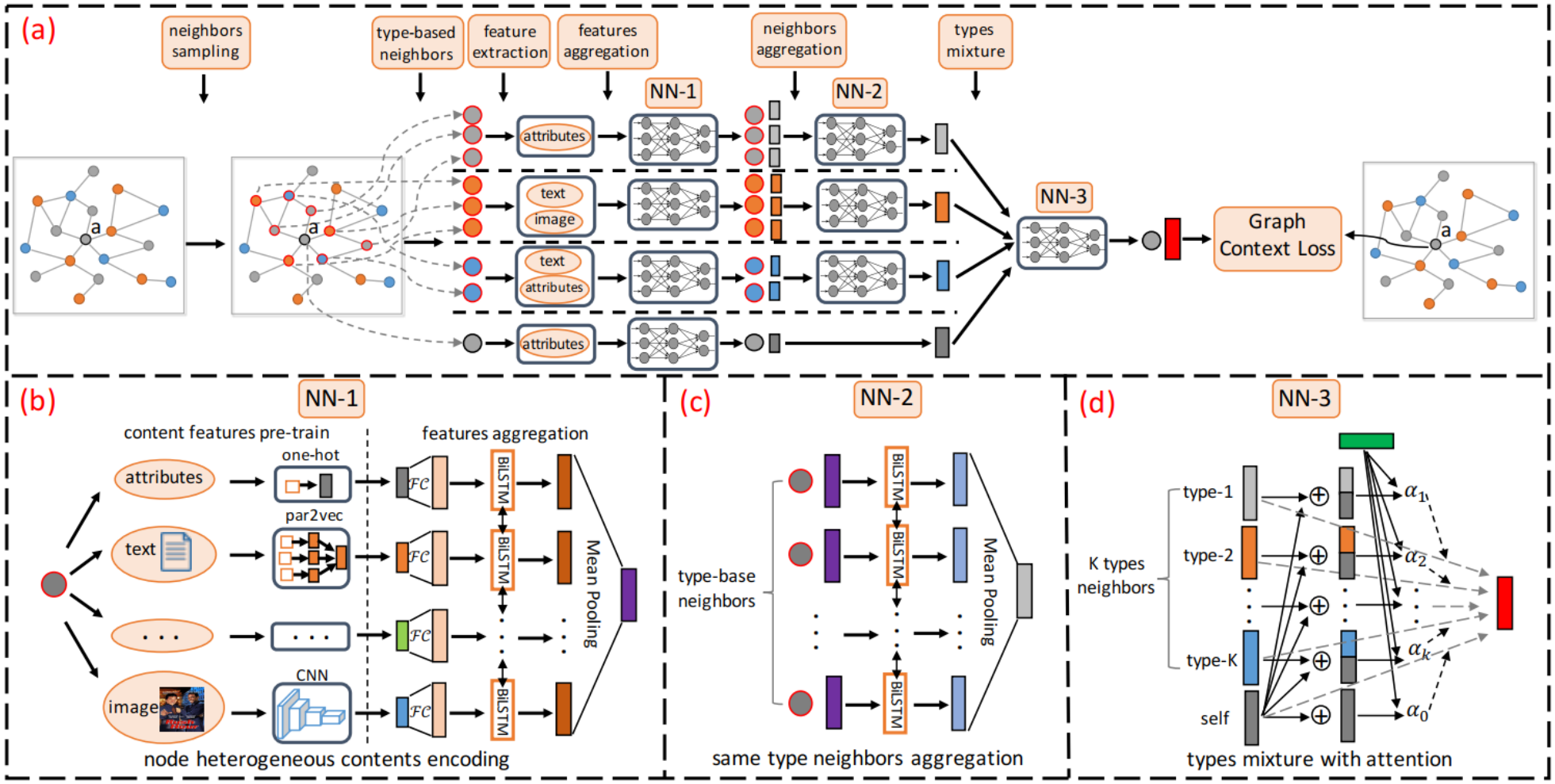
解决方法：

- 设计了一个带重启的随机游走策略，为HetG中的每个节点采样固定数量的强关联的异质邻居。
- 设计了两个模块组成的异质GNN，聚合采样邻居的特征信息。
  - 第一个模块使用了RNN编码节点异质内容信息间深度的特征交互信息，得到每个节点的内容(content)嵌入。
  - 第二个模块使用另一个RNN，聚合不同类别的邻居节点的嵌入，并且运用了注意力机制，为不同类型的异质邻居节点分配不同的注意力，得到最终的节点嵌入。
- 最后使用基于图上下文的loss，运用mini-batch梯度下降法训练模型。

贡献：

- 定义了同时涉及图结构的异质性和节点内容异质性的异质图表示学习问题。
- 提出了HetGNN模型，用于异构图上的表示学习，实现了同时捕获异质的结构和节点内容。
- 多个图数据挖掘任务实现state-of-the-art。

模型：HetGNN



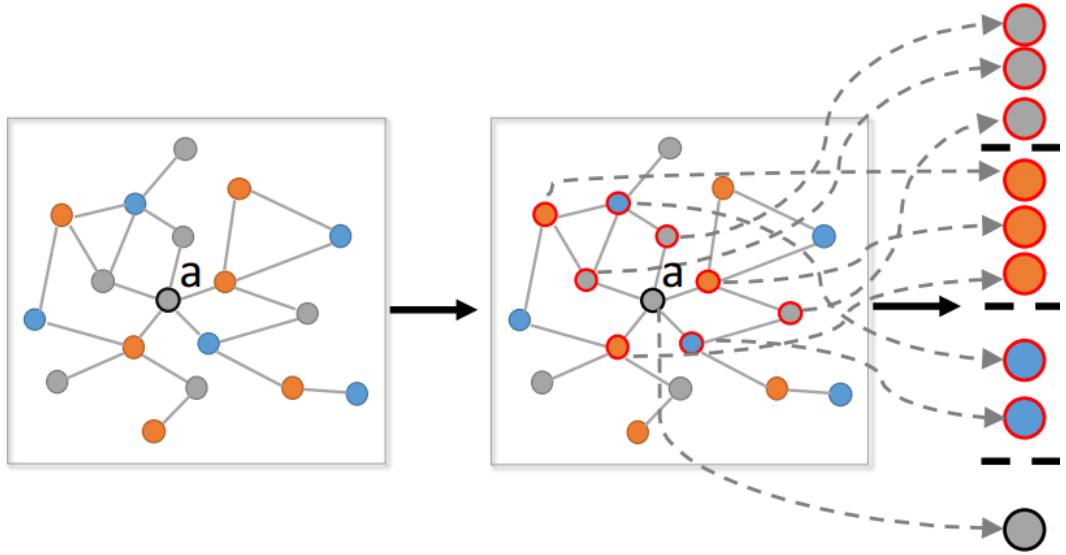
a. HetGNN整体框架
b. 节点异质信息编码
c. 聚合异质邻居节点的信息
d. 类型混合注意力机制

C1：如何为异构图中的节点采样到强相关的异质邻居？

带有重启的随机游走策略（RWR）

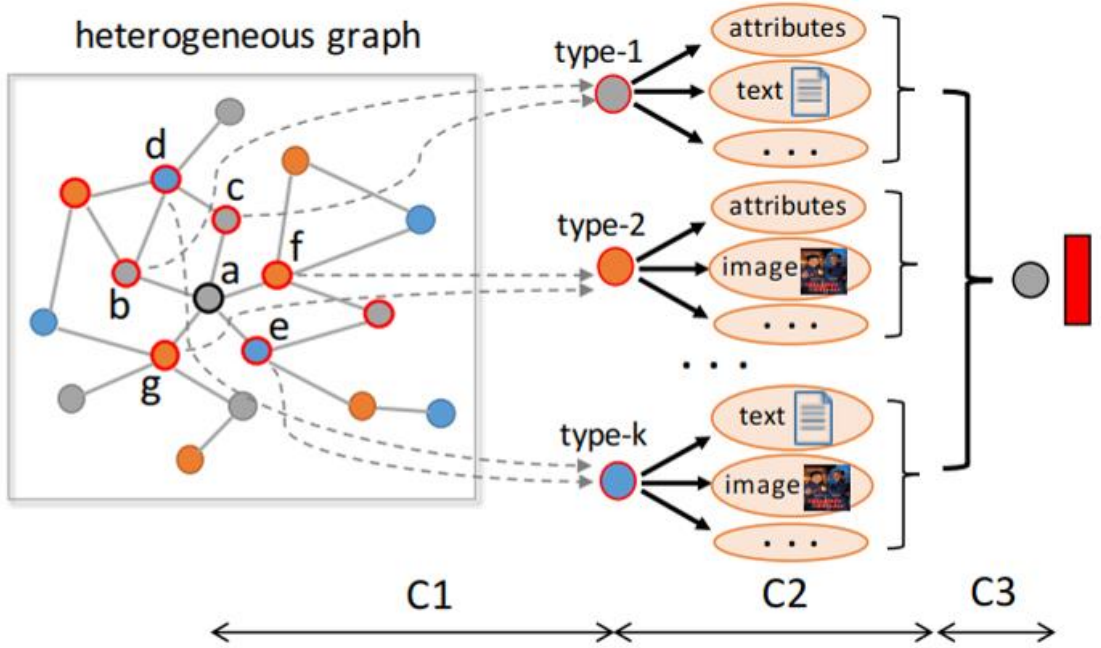
1.使用RWR采样固定长度路径

从节点v出发，以迭代的方式每一步向当前节点的邻居节点，或者以P的概率返回起始节点，迭代直到采样到固定数量的节点为止。同时，确保每种类型节点都被采样到。

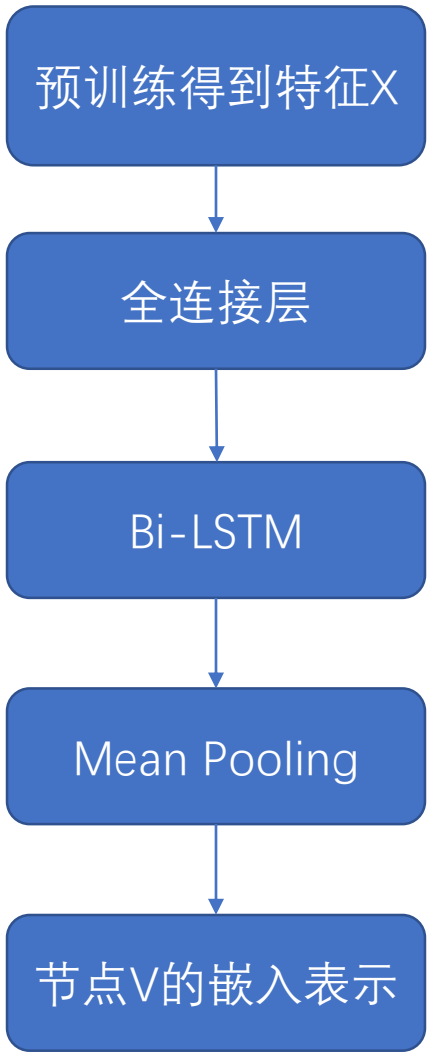


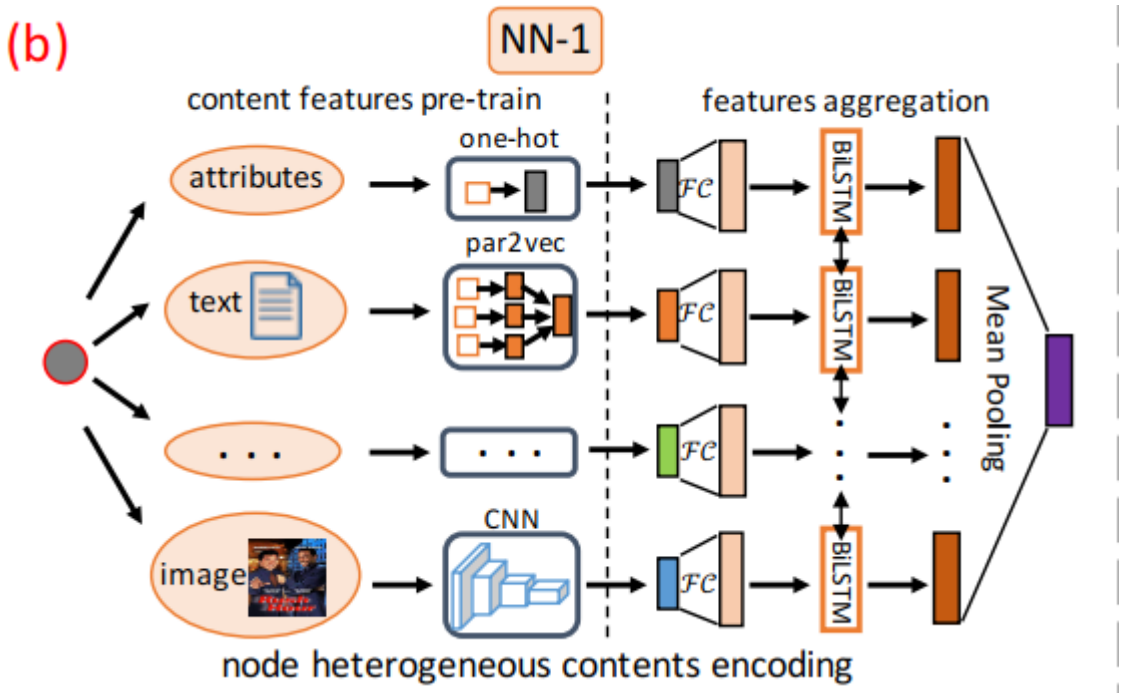
2.将这些采样到的邻居节点按照类型进行分类

针对每种节点类型t，根据其在第一步中出现的频次，选取top k个节点，作为v节点的邻居节点。



C2：如何为异构图中带有异质内容信息的不同节点设计节点内容信息的encoder?





$$f_1(v) = \frac{\sum_{i \in C_v} \left[\overrightarrow{LSTM} \left\{ \mathcal{FC}_{\theta_x}(\mathbf{x}_i) \right\} \oplus \overleftarrow{LSTM} \left\{ \mathcal{FC}_{\theta_x}(\mathbf{x}_i) \right\} \right]}{|C_v|} \tag{1}$$

其中，内容嵌入向量$f_1(V)$是d维的，FC_{θ_x}是参数为θ_x的全连接神经网络，作为特征转换函数；操作符表示拼接操作。

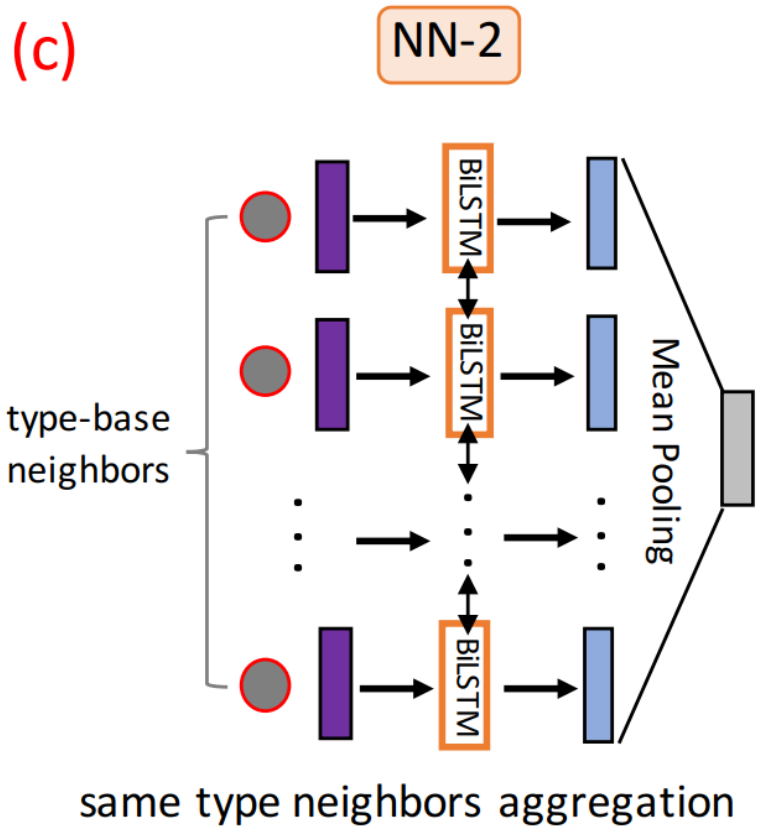
C3：如何在聚合异质邻居特征信息的过程中考虑不同节点类型的影响?

1.同类型邻居聚合

在前面采用RWR策略采样到的top K个邻居节点，使用神经网络f_2来聚合节点v的嵌入表示，这里采用Bi-LSTM

$$f_2^t(v) = \frac{\sum_{v' \in N_t(v)} \left[\overrightarrow{LSTM} \{ f_1(v') \} \oplus \overleftarrow{LSTM} \{ f_1(v') \} \right]}{|N_t(v)|}$$

其中，定义为节点$v \in V$采样到的t类型的邻居节点为：$N_t(v)$。$f_1(v')$是C2中求得的v'的嵌入。



C3：如何在聚合异质邻居特征信息的过程中考虑不同节点类型的影响?

2.不同类型邻居的聚合

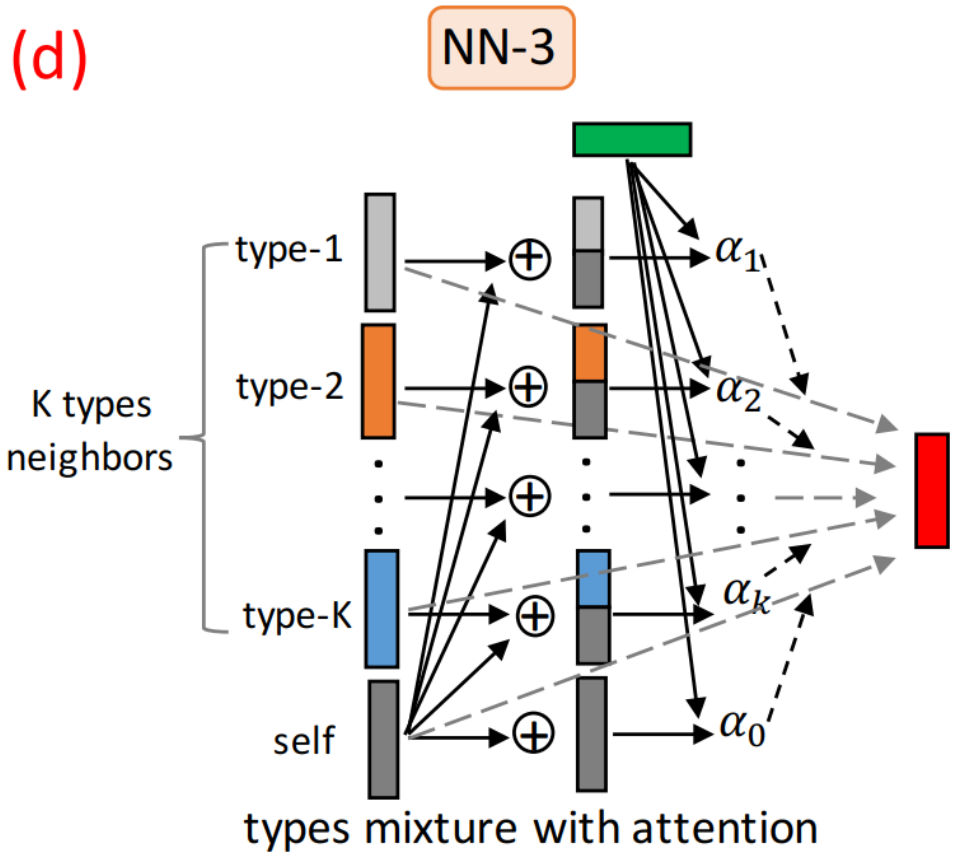
由于不同类型的节点对中心节点的贡献程度不同，这里采用注意力机制，为不同类型节点分配不同的注意力。

输出的嵌入表示为：

$$\mathcal{E}_v = \sum_{f_i \in \mathcal{F}(v)} \alpha^{v,i} f_i$$

$$\alpha^{v,i} = \frac{exp\left\{LeakyReLU(u^T[f_i \bigoplus f_1(v)])\right\}}{\sum_{f_j \in \mathcal{F}(v)} exp\left\{LeakyReLU(u^T[f_j \bigoplus f_1(v)])\right\}}$$

其中，$\alpha^{v,*}$表示不同嵌入的重要程度，$u \in R^{2d\times 1}$是注意力机制的参数



实验

实验回答了4个问题：

RQ1：HetGNN和state-of-the-art baseline相比，在**RQ1-1**链路预测、**RQ1-2**个性化推荐、**RQ1-3**节点分类和聚类任务上效果如何？

RQ2：在归纳式的节点分类和聚类任务上，和HetGNN和state-of-the-art baseline相比效果如何？

RQ3：节点的异质内容信息编码和异质邻居聚合是如何影响模型性能的？

RQ4：不同的超参数，如嵌入向量的维度、采样的邻居数目是如何影响模型性能的？

数据集：

Table 2: Datasets used in this work.

Data	Node	Edge
Academic I (A-I)	# author: 160,713 # paper: 111,409 # venue: 150	# author-paper: 295,103 # paper-paper: 138,464 # paper-venue: 111,409
Academic II (A-II)	# author: 28,646 # paper: 21,044 # venue: 18	# author-paper: 69,311 # paper-paper: 46,931 # paper-venue: 21,044
Movies Review (R-I)	# user: 18,340 # item: 56,361	# user-item: 629,125
CDs Review (R-II)	# user: 16,844 # item: 106,892	# user-item: 555,050

实验

RQ1：HetGNN和state-of-the-art baseline相比，在**RQ1-1**链路预测、**RQ1-2**个性化推荐、**RQ1-3**节点分类和聚类任务上效果如何？

$Data_{split}$	Metric	MP2V [4]	ASNE [15]	SHNE [34]	GSAGE [7]	GAT [31]	HetGNN
A-I_{2003} (type-1)	AUC	0.636	0.683	0.696	0.694	0.701	**0.714**
	F1	0.435	0.584	0.597	0.586	0.606	**0.620**
A-I_{2003} (type-2)	AUC	0.790	0.794	0.781	0.790	0.821	**0.837**
	F1	0.743	0.774	0.755	0.746	0.792	**0.815**
A-I_{2002} (type-1)	AUC	0.626	0.667	0.688	0.681	0.691	**0.710**
	F1	0.412	0.554	0.590	0.567	0.589	**0.615**
A-I_{2002} (type-2)	AUC	0.808	0.782	0.795	0.806	0.837	**0.851**
	F1	0.770	0.753	0.761	0.772	0.816	**0.828**
A-II_{2013} (type-1)	AUC	0.596	0.689	0.683	0.695	0.678	**0.717**
	F1	0.348	0.643	0.639	0.615	0.613	**0.669**
A-II_{2013} (type-2)	AUC	0.712	0.721	0.695	0.714	0.732	**0.767**
	F1	0.647	0.713	0.674	0.664	0.705	**0.754**
A-II_{2012} (type-1)	AUC	0.586	0.671	0.672	0.676	0.655	**0.701**
	F1	0.318	0.615	0.612	0.573	0.560	**0.642**
A-II_{2012} (type-2)	AUC	0.724	0.726	0.706	0.739	0.750	**0.775**
	F1	0.664	0.737	0.692	0.706	0.715	**0.757**
R-$I_{5:5}$	AUC	0.634	0.623	0.651	0.661	0.683	**0.749**
	F1	0.445	0.551	0.586	0.542	0.665	**0.735**
R-$I_{7:3}$	AUC	0.701	0.656	0.695	0.716	0.706	**0.787**
	F1	0.595	0.613	0.660	0.688	0.702	**0.776**
R-$II_{5:5}$	AUC	0.678	0.655	0.685	0.677	0.712	**0.736**
	F1	0.541	0.582	0.593	0.565	0.659	**0.701**
R-$II_{7:3}$	AUC	0.737	0.695	0.728	0.721	0.742	**0.772**
	F1	0.660	0.648	0.685	0.653	0.713	**0.749**

RQ1-1

$Data_{split}$	Metric	MP2V [4]	ASNE [15]	SHNE [34]	GSAGE [7]	GAT [31]	HetGNN
A-I_{2003}	Rec	0.158	0.201	0.298	0.263	0.275	**0.319**
	Pre	0.044	0.060	0.081	0.077	0.079	**0.094**
	F1	0.069	0.092	0.127	0.120	0.123	**0.145**
A-I_{2002}	Rec	0.144	0.152	0.279	0.231	0.274	**0.293**
	Pre	0.046	0.050	0.086	0.073	0.087	**0.093**
	F1	0.070	0.075	0.134	0.112	0.132	**0.141**
A-II_{2013}	Rec	0.516	0.419	0.608	0.540	0.568	**0.625**
	Pre	0.207	0.174	0.241	0.219	0.230	**0.252**
	F1	0.295	0.333	0.345	0.312	0.327	**0.359**
A-II_{2012}	Rec	0.468	0.382	0.552	0.512	0.518	**0.606**
	Pre	0.204	0.171	0.233	0.224	0.227	**0.264**
	F1	0.284	0.236	0.327	0.312	0.316	**0.368**

RQ1-2

Task	Metric	MP2V [4]	ASNE [15]	SHNE [34]	GSAGE [7]	GAT [31]	HetGNN
MC (10%)	Macro-F1	0.972	0.965	0.939	**0.978**	0.962	**0.978**
	Micro-F1	0.973	0.967	0.940	0.978	0.963	**0.979**
MC (30%)	Macro-F1	0.975	0.969	0.939	0.979	0.965	**0.981**
	Micro-F1	0.975	0.970	0.941	0.980	0.965	**0.982**
NC	NMI	0.894	0.854	0.776	**0.914**	0.845	0.901
	ARI	0.933	0.898	0.813	**0.945**	0.882	0.932

RQ1-3

RQ2：在归纳式的节点分类和聚类任务上，和HetGNN和state-of-the-art baseline相比效果如何？

Table 6: Inductive multi-labels classification (IMC) and node clustering (INC) results. Percentage is training data ratio.

Task	Metric	GSAGE [7]	GAT [31]	HetGNN
IMC (10%)	Macro-F1	0.938	0.954	**0.962**
	Micro-F1	0.945	0.958	**0.965**
IMC (30%)	Macro-F1	0.949	0.956	**0.964**
	Micro-F1	0.955	0.960	**0.968**
INC	NMI	0.714	0.765	**0.840**
	ARI	0.764	0.803	**0.894**

RQ3：节点的异质内容信息编码和异质邻居聚合是如何影响模型性能的?

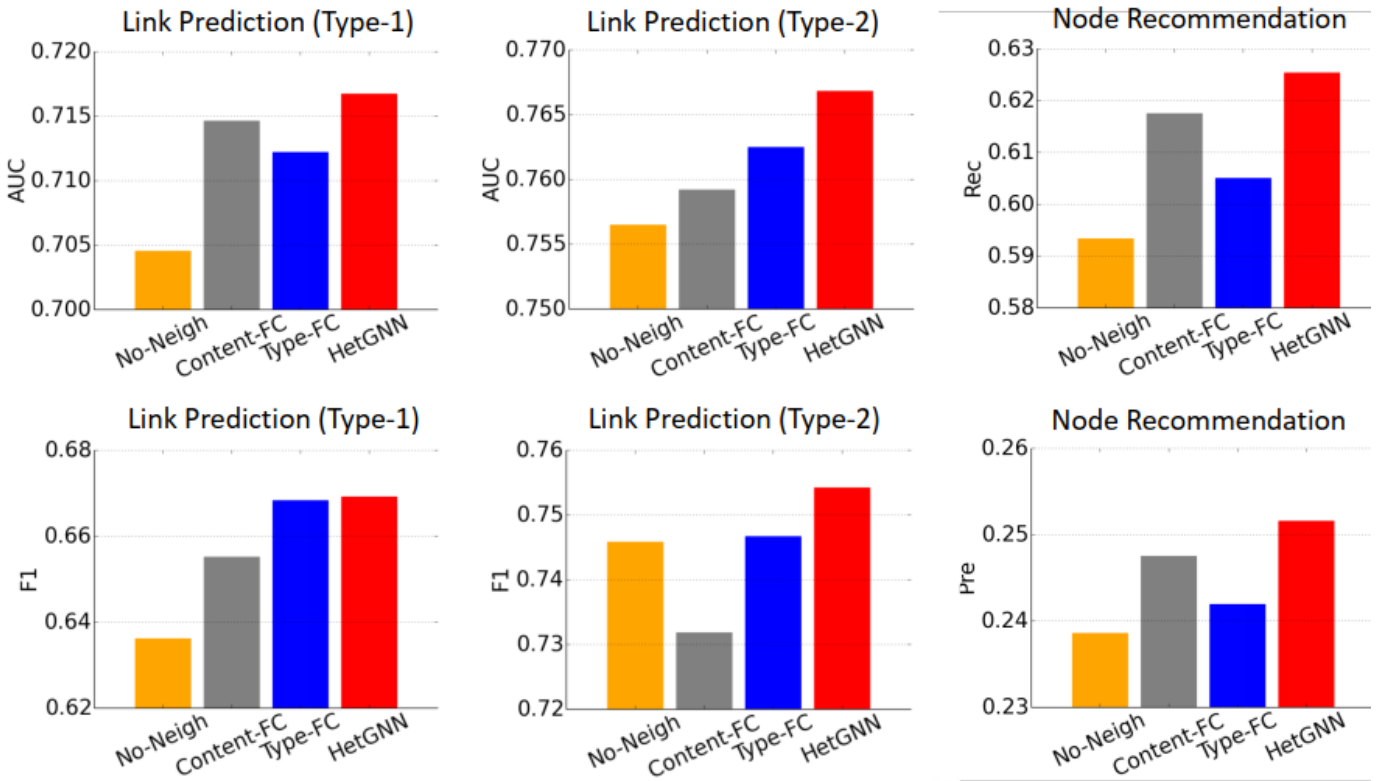


Figure 4: Performances of variant proposed models.

RQ4：不同的超参数，如嵌入向量的维度、采样的邻居数目是如何影响模型性能的？

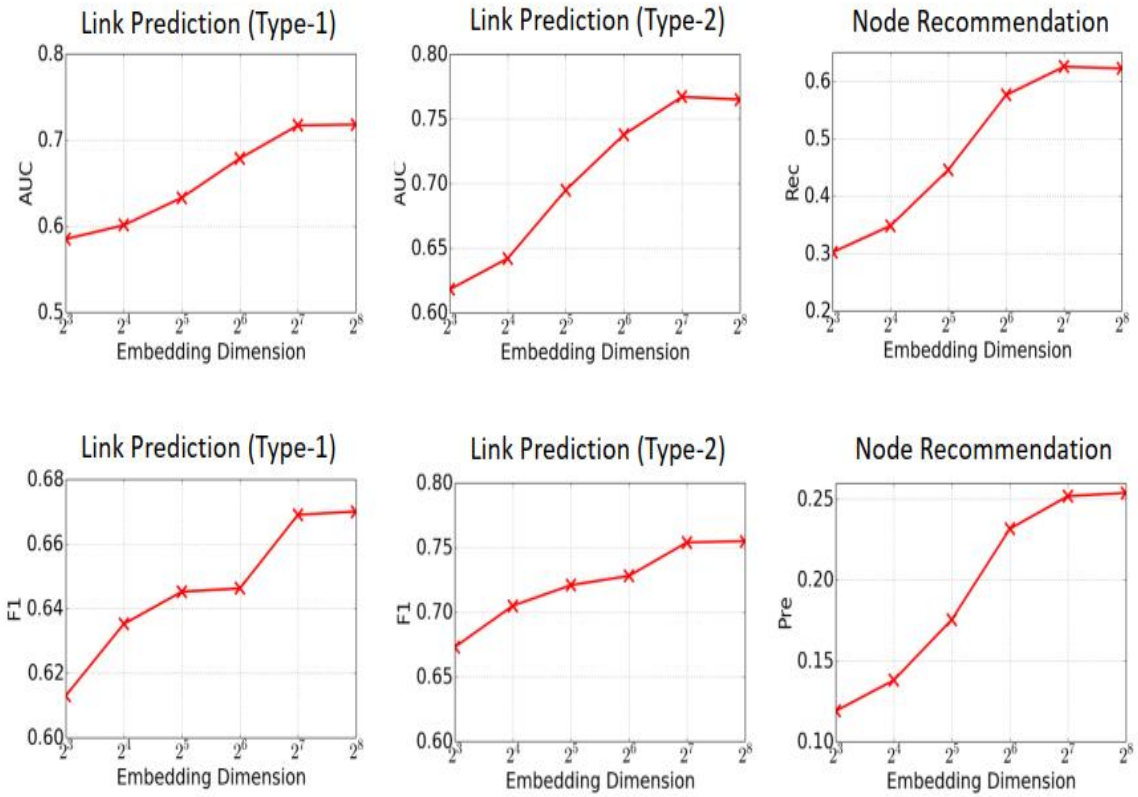


Figure 5: Impact of embedding dimension.

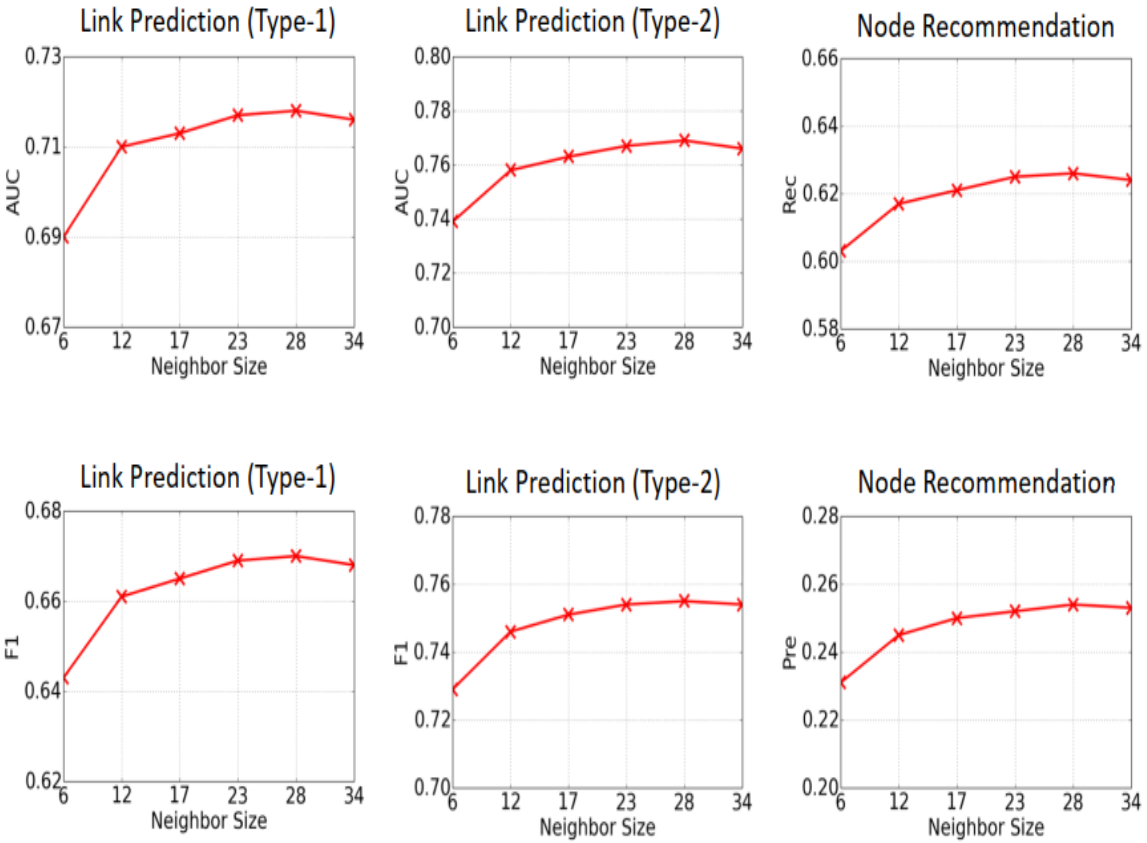


Figure 6: Impact of sampled neighbor size.

总结

- 本文提出了一种用于学习异质图节点嵌入表示的方法：HetGNN。具体来说，模型先编码节点多种多样的(异质)属性信息，将编码后得到的属性嵌入作为节点的表示；然后对于每个节点，根据邻居节点的类型，聚合同一类型的邻居节点的嵌入表示；最后将不同类型的邻居节点信息聚合起来。

- 通过对比实验，HetGNN在链路预测，节点分类、聚类等任务上均取得了最好的实验效果。
- 通过消融实验，分析了不同模块对于实验结果的影响
- 通过消融实验，分析了超参数和邻居采样大小对于性能的影响