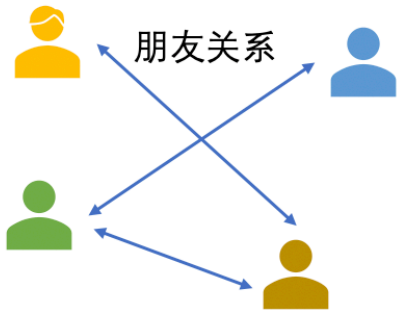
图神经网络七日打卡营

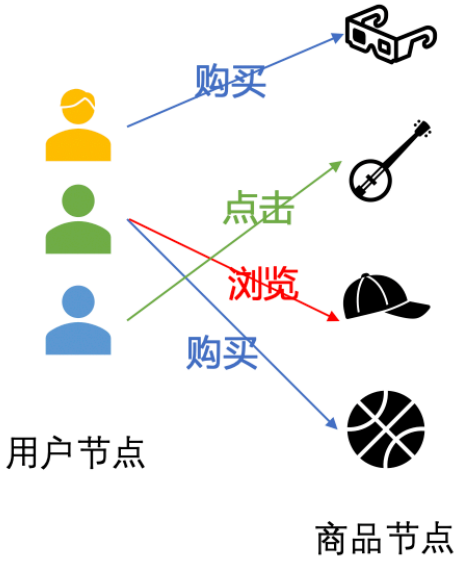
小斯妹　百度PGL团队成员

2020.11.24

昨晚课堂内容的问题



同构图



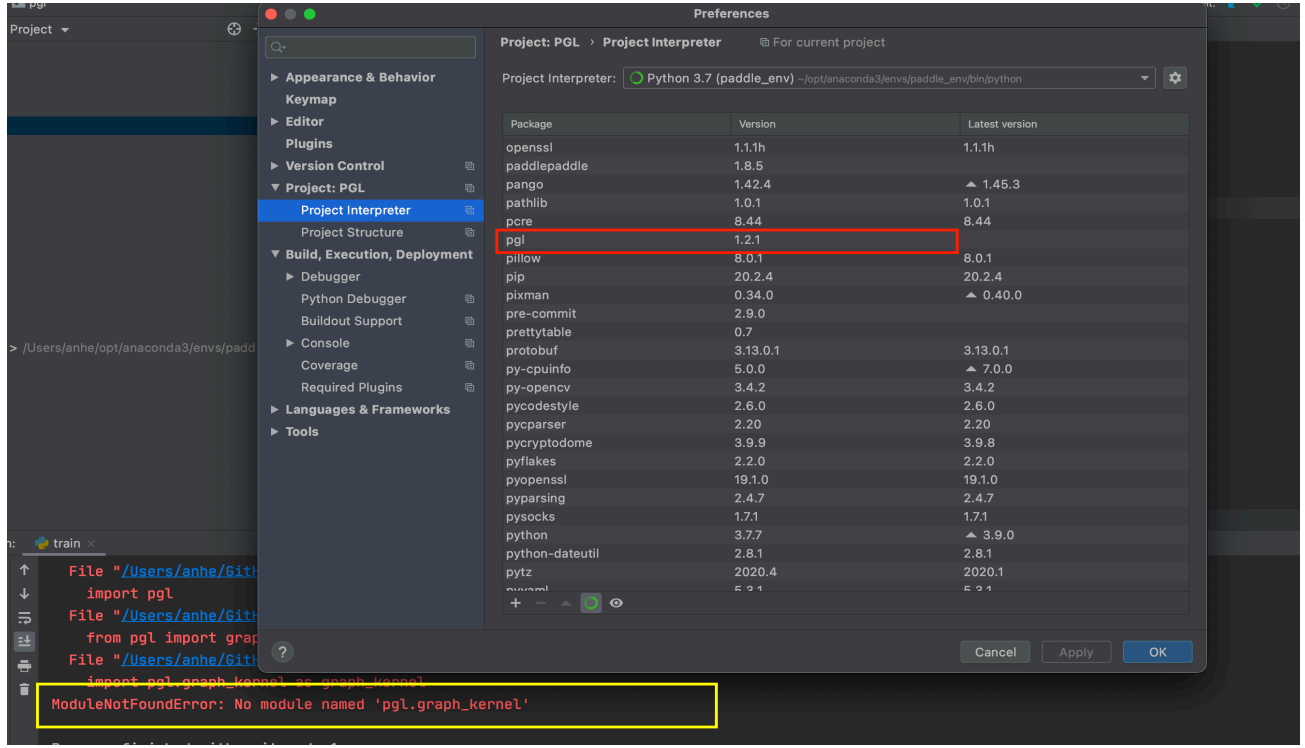
异构图

图论：与图的映射拓扑结构相关

昨晚作业常见的问题

助教总结：https://aistudio.baidu.com/aistudio/projectdetail/1259681

1. PGL 相关： No module named pgl.graph_kernel



切换当前运行目录，使得当前目录无 pgl 目录

昨晚作业常见的问题

2. PGL 相关：Import Error: DLL load failed

```
Traceback (most recent call last):
  File "G:/code/gcn/train.py", line 14, in <module>
    import pgl
  File "C:\Users\Zhou Hang\Anaconda3\envs\paddle_env\lib\site-packages\pgl\__init__.py", line 19, in <module>
    from pgl import graph
  File "C:\Users\Zhou Hang\Anaconda3\envs\paddle_env\lib\site-packages\pgl\graph.py", line 22, in <module>
    import pgl.graph_kernel as graph_kernel
ImportError: DLL load failed: 找不到指定的模块。

Process finished with exit code 1
```

Windows系统 VC 环境问题，需安装 Visual Studio，在安装勾选时一定要选择 使用c++的桌面开发。

昨晚作业常见的问题

3. 在本地运行GCN example时出现epochs错误

```
[INFO] 2020-11-24 00:00:38,175 [    train.py:  135]:    Epoch 171 (0.02935 sec) Train Loss: 0.423836 Train Acc: 0.921429 Val Loss: 0.736453 Val Acc: 0.816667
[INFO] 2020-11-24 00:00:38,225 [    train.py:  135]:    Epoch 172 (0.02935 sec) Train Loss: 0.398153 Train Acc: 0.950000 Val Loss: 0.737138 Val Acc: 0.816667
[INFO] 2020-11-24 00:00:38,272 [    train.py:  135]:    Epoch 173 (0.02935 sec) Train Loss: 0.402271 Train Acc: 0.950000 Val Loss: 0.737251 Val Acc: 0.823333
[INFO] 2020-11-24 00:00:38,319 [    train.py:  135]:    Epoch 174 (0.02936 sec) Train Loss: 0.418642 Train Acc: 0.935714 Val Loss: 0.737554 Val Acc: 0.813333
[INFO] 2020-11-24 00:00:38,365 [    train.py:  135]:    Epoch 175 (0.02936 sec) Train Loss: 0.401634 Train Acc: 0.950000 Val Loss: 0.737445 Val Acc: 0.810000
[INFO] 2020-11-24 00:00:38,412 [    train.py:  135]:    Epoch 176 (0.02936 sec) Train Loss: 0.425680 Train Acc: 0.942857 Val Loss: 0.736473 Val Acc: 0.806667
[INFO] 2020-11-24 00:00:38,459 [    train.py:  135]:    Epoch 177 (0.02936 sec) Train Loss: 0.432720 Train Acc: 0.964286 Val Loss: 0.733174 Val Acc: 0.810000
[INFO] 2020-11-24 00:00:38,506 [    train.py:  135]:    Epoch 178 (0.02936 sec) Train Loss: 0.426675 Train Acc: 0.978571 Val Loss: 0.729171 Val Acc: 0.816667
[INFO] 2020-11-24 00:00:38,552 [    train.py:  135]:    Epoch 179 (0.02936 sec) Train Loss: 0.355110 Train Acc: 0.971429 Val Loss: 0.725868 Val Acc: 0.830000
[INFO] 2020-11-24 00:00:38,598 [    train.py:  135]:    Epoch 180 (0.02936 sec) Train Loss: 0.420999 Train Acc: 0.950000 Val Lo
[INFO] 2020-11-24 00:00:38,645 [    train.py:  135]:    Epoch 181 (0.02935 sec) Train Loss: 0.423321 Train Acc: 0.935714 Val Lo
[INFO] 2020-11-24 00:00:38,690 [    train.py:  135]:    Epoch 182 (0.02935 sec) Train Loss: 0.388520 Train Acc: 0.971429 Val Lo
[INFO] 2020-11-24 00:00:38,737 [    train.py:  135]:    Epoch 183 (0.02935 sec) Train Loss: 0.382683 Train Acc: 0.971429 Val Lo
[INFO] 2020-11-24 00:00:38,784 [    train.py:  135]:    Epoch 184 (0.02935 sec) Train Loss: 0.389729 Train Acc: 0.935714 Val Lo
[INFO] 2020-11-24 00:00:38,830 [    train.py:  135]:    Epoch 185 (0.02935 sec) Train Loss: 0.394108 Train Acc: 0.950000 Val Lo
[INFO] 2020-11-24 00:00:38,877 [    train.py:  135]:    Epoch 186 (0.02934 sec) Train Loss: 0.411241 Train Acc: 0.935714 Val Lo
[INFO] 2020-11-24 00:00:38,924 [    train.py:  135]:    Epoch 187 (0.02935 sec) Train Loss: 0.410364 Train Acc: 0.942857 Val Lo
[INFO] 2020-11-24 00:00:38,975 [    train.py:  135]:    Epoch 188 (0.02937 sec) Train Loss: 0.399058 Train Acc: 0.950000 Val Lo
[INFO] 2020-11-24 00:00:39,022 [    train.py:  135]:    Epoch 189 (0.02937 sec) Train Loss: 0.381832 Train Acc: 0.942857 Val Lo
[INFO] 2020-11-24 00:00:39,067 [    train.py:  135]:    Epoch 190 (0.02937 sec) Train Loss: 0.361222 Train Acc: 0.964286 Val Lo
[INFO] 2020-11-24 00:00:39,113 [    train.py:  135]:    Epoch 191 (0.02936 sec) Train Loss: 0.409456 Train Acc: 0.921429 Val Lo
[INFO] 2020-11-24 00:00:39,159 [    train.py:  135]:    Epoch 192 (0.02936 sec) Train Loss: 0.358753 Train Acc: 0.957143 Val Lo
[INFO] 2020-11-24 00:00:39,206 [    train.py:  135]:    Epoch 193 (0.02936 sec) Train Loss: 0.409283 Train Acc: 0.950000 Val Lo
[INFO] 2020-11-24 00:00:39,253 [    train.py:  135]:    Epoch 194 (0.02936 sec) Train Loss: 0.398915 Train Acc: 0.964286 Val Lo
[INFO] 2020-11-24 00:00:39,299 [    train.py:  135]:    Epoch 195 (0.02936 sec) Train Loss: 0.396609 Train Acc: 0.942857 Val Lo
[INFO] 2020-11-24 00:00:39,348 [    train.py:  135]:    Epoch 196 (0.02936 sec) Train Loss: 0.378188 Train Acc: 0.957143 Val Lo
[INFO] 2020-11-24 00:00:39,394 [    train.py:  135]:    Epoch 197 (0.02935 sec) Train Loss: 0.333872 Train Acc: 0.957143 Val Loss: 0.712905 Val Acc: 0.806667
[INFO] 2020-11-24 00:00:39,441 [    train.py:  135]:    Epoch 198 (0.02935 sec) Train Loss: 0.412541 Train Acc: 0.907143 Val Loss: 0.712430 Val Acc: 0.803333
[INFO] 2020-11-24 00:00:39,487 [    train.py:  135]:    Epoch 199 (0.02935 sec) Train Loss: 0.399026 Train Acc: 0.950000 Val Loss: 0.709993 Val Acc: 0.820000
[INFO] 2020-11-24 00:00:39,508 [    train.py:  148]:    Accuracy: 0.814000

E:\java\PGL-main\examples\gcn>python train.py --epochs 100
usage: train.py [-h] [--dataset DATASET] [--use_cuda]
train.py: error: unrecognized arguments: --epochs 100

E:\java\PGL-main\examples\gcn>python train.py --epoch 100
usage: train.py [-h] [--dataset DATASET] [--use_cuda]
train.py: error: unrecognized arguments: --epoch 100

E:\java\PGL-main\examples\gcn>python train.py -epochs 100
usage: train.py [-h] [--dataset DATASET] [--use_cuda]
train.py: error: unrecognized arguments: -epochs 100

E:\java\PGL-main\examples\gcn>aa
```

周小鱼whoyou

关于预习作业，在本机提示error: unrecognized arguments: --epochs 100的处理经验分享：

本人的机子是没有GPU环境的。

1、操作是按notebook上面的教程做的

4.2 PGL相关信息(上传运行示例-GCN的截图)：

解决方法是删除命令行后面的 --epochs 100 或者直接运行aistudio平台中的作业一。

昨晚作业常见的问题

4. 运行DeepWalk 链接预测任务时出现参数不匹配错误

陈轻歌 6小时前

deepwalk 在运行linkprediction任务时出现参数维度不匹配的错误分析及解决方案

按照教材上给出的样例代码运行deepwalk执行linkprediction任务时，会出现如下错误：

RuntimeError: Variable's shape does not match, the Program requires a parameter with the shape of ((18772, 128)), while the loaded parameter (namely [content]) has a shape of ((18772, 10)).

（图片不知为何上传不了了。）

从报错提示来看，之前输入到模型中的参数维度不匹配，其原因在于之前训练deepwalk中，我们规定了如下参数规格：

```
!cd PGL/examples/deepwalk/; python deepwalk.py --dataset ArXiv --save_path
./tmp/deepwalk_ArXiv --offline_learning --epoch 2 --batch_size 256 --processes 1 --walk_len
10 --hidden_size 10
```

这里将hidden_size规定为10，而后续的任务中，对hidden_size的要求是不小于128。

解决方案：

将之前训练deepwalk的代码修改如下：

```
!cd PGL/examples/deepwalk/; python deepwalk.py --dataset ArXiv --save_path
./tmp/deepwalk_ArXiv --offline_learning --epoch 2 --batch_size 256 --processes 1 --walk_len
10 --hidden_size 128
```

训练完后即可顺利执行linkprediction任务。

1 回复

昨晚作业常见的问题

5. 运行deepwalk.py 时出现整数越界问题。

陈轻歌

关于在Windows系统上运行PGL中DeepWalk.py中遇到的问题及解决方案。

报错截图如下：

```
[INFO] 2020-11-24 10:33:00,850 [ deepwalk.py:  172]:    Start random walk on disk...
Traceback (most recent call last):
  File "deepwalk.py", line 258, in <module>
    main(args)
  File "deepwalk.py", line 179, in main
    for x in range(epoch)]
  File "deepwalk.py", line 179, in <listcomp>
    for x in range(epoch)]
  File "mtrand.pyx", line 744, in numpy.random.mtrand.RandomState.randint
  File "_bounded_integers.pyx", line 1343, in numpy.random._bounded_integers._rand_int32
ValueError: high is out of bounds for int32

(paddle) c:\Codes\python_codes\github_codes\PGL-main\examples\deepwalk>
```

根据报错信息可以定位到是deepwalk.py文件中第178行的代码出错。

```
        args_list = [(x, dataset.graph, walk_save_path, 1,
                      batch_size, walk_len, np.random.randint(2**32, dtype="int64"))
                     for x in range(epoch)]
```

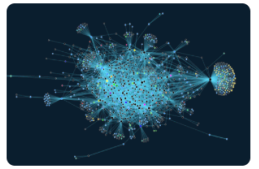
出错原因（猜测）：在Windows平台上，randint（2^32）越界了，解决方案就是在后面指定类型为64位。

解决方案：见上述代码。

助教总结：

https://aistudio.baidu.com/aistudio/projectdetail/1259681

可到课程讨论区查看



图神经网络7日打卡营

飞桨深度学习学院

2020/11/04 - 2021/02/12

分享

课节

作业 2

考试

比赛

讨论区

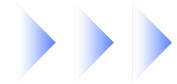
公告栏

如何提交作业


[实战一] 运行GCN和DeepWalk
文件夹
数据集
环境
版本
可视化
生成新版本
暂无历史版本


生成项目版本
说明：
1、ipynb文件已经默认选中。
2、仅支持非data目录下的文件，文件总数小于1000个，总体积不超过1GB。
* 版本名称
请输入版本名称
当前项目文件
PGL
work
已选文件
数量: 0个 体积: 0B
请通过左侧“+”按钮选择文件
生成版本
取 消

课程大纲



第一课：图学习初印象

- 图学习概述、入门路线
- 实践：环境搭建

第二课：图游走类算法

- DeepWalk, node2vec, metapath2vec
- 实践：DeepWalk

第三课：图神经网络算法(一)

- GCN, GAT
- 实践：GCN, GAT

第四课：图神经网络算法(二)

- 图采样、邻居聚合
- 实践：GraphSage

第五课：GNN 进阶

- ERNIESage, UniMP
- 实践：ERNIESage 代码讲解

后续：新冠项目实战，带你助力疫情防控

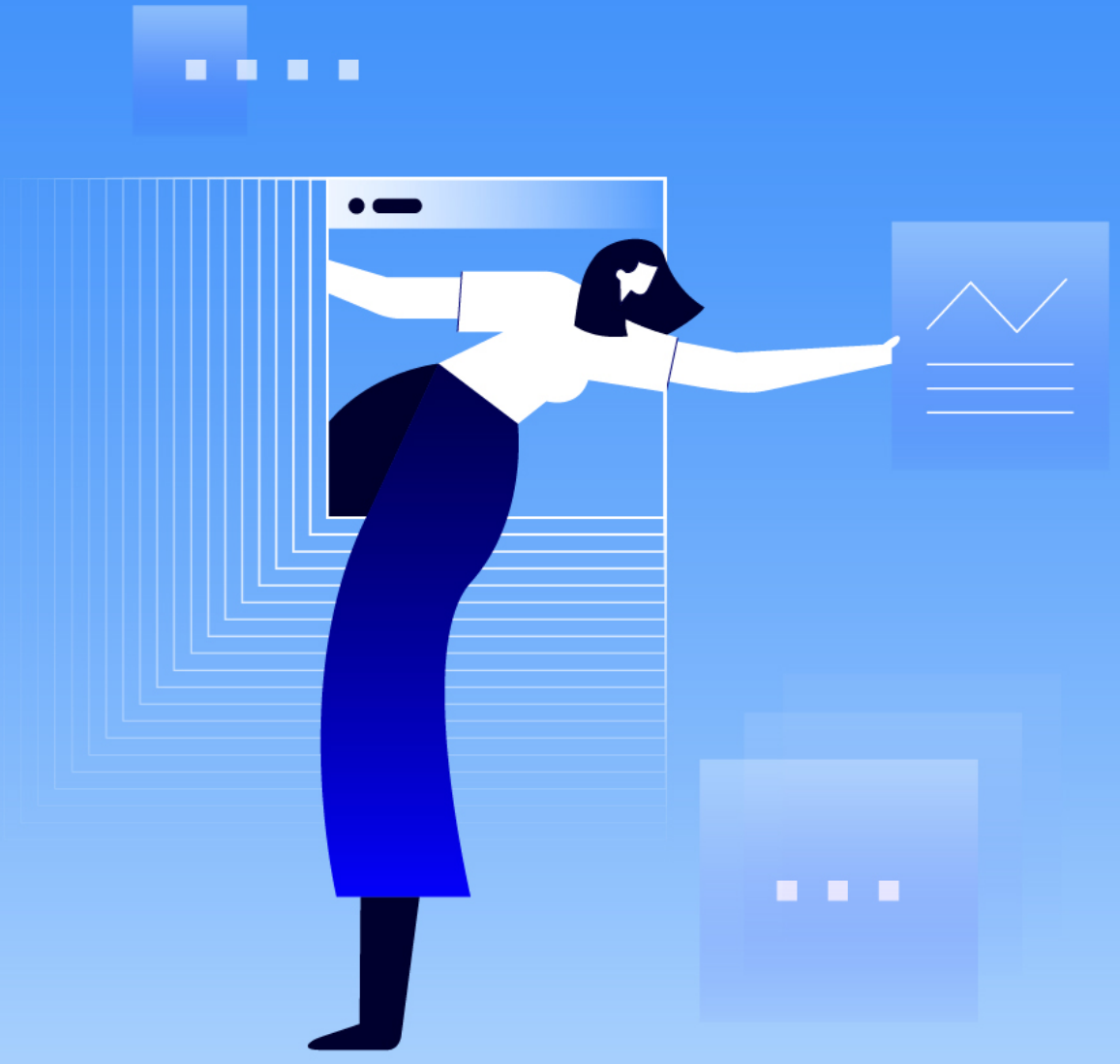
参考材料：

- 斯坦福CS224W课程： http://cs224w.stanford.edu
- 图学习库 PGL：https://github.com/PaddlePaddle/PGL

飞桨

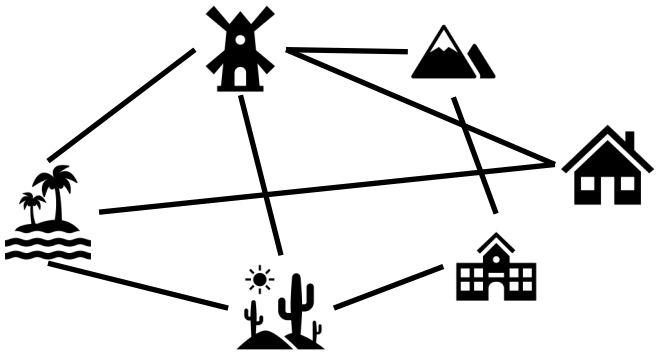
第二课 图游走类算法

小斯妹 百度PGL团队成员



2020.11.24

目标：Node embeddings



Node embeddings → 下游任务

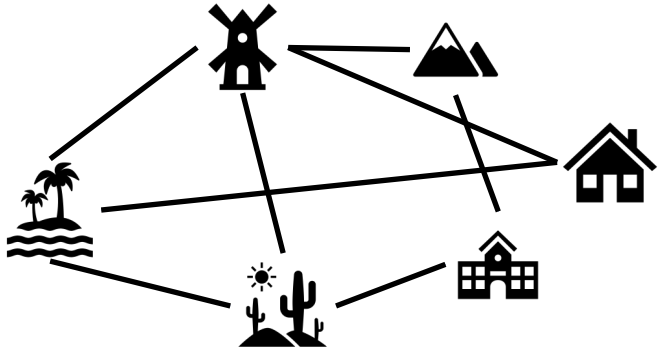
↓

学习到节点与邻居的关系

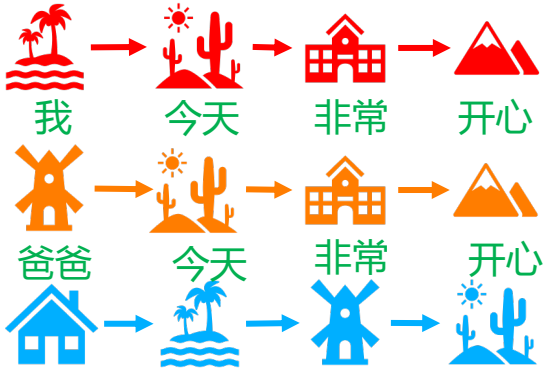
↓

如何得到 Node embeddings ?

图游走类算法



假设序列最大长度为4.



节点 ------ 词 ？
序列 ------ 句子？

图表示学习

得到节点表示，
用于下游任务。

注意：前方即将迎来新内容

Word2vec

NLP领域：Word2vec

图游走类模型最开始参考的就是 NLP 领域中的 Word2vec 模型。

苹果

苹果是蔷薇科苹果亚科苹果属植物，其树为落叶乔木。
苹果营养价值很高，富含矿物质和维生素。
苹果是一种低热量的食物，每100克产生大约60千卡左右的热量。
苹果中营养成分可溶性大，容易被人体吸收。
所有的水果里，我最爱吃苹果了。
……

词的语义由其上下文决定。
A word is characterized by the company it keeps.

Word2vec: Distributed Representations of Words and Phrases and their Compositionality

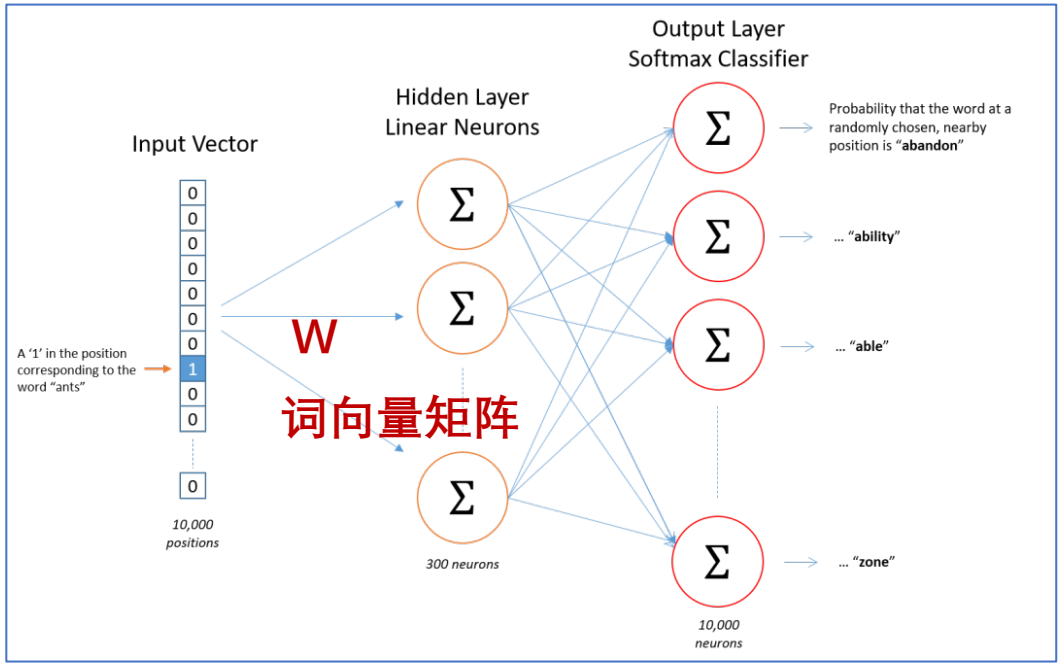
Word2vec: Skip Gram

Skip Gram: 根据中心词预测上下文

window size=3

... uniformly from the neighbors of the last ...

context　　center word　　context



输入层：中心词的one hot 向量

投影层：单词对应的低维向量

输出层：Softmax 层

Word2vec: Distributed Representations of Words and Phrases and their Compositionality

Word2vec: Negative Sampling

假设，给定中心词 **orange**，我们要预测其上下文词中的 **juice**。

计算词表内所有单词的概率 -> **计算量大**

Softmax

0.0003	0.005	0.001	…	0.6	…		…
alien	at	me	happy	juice	out	……	zero

orange

Negative Sampling(负采样)

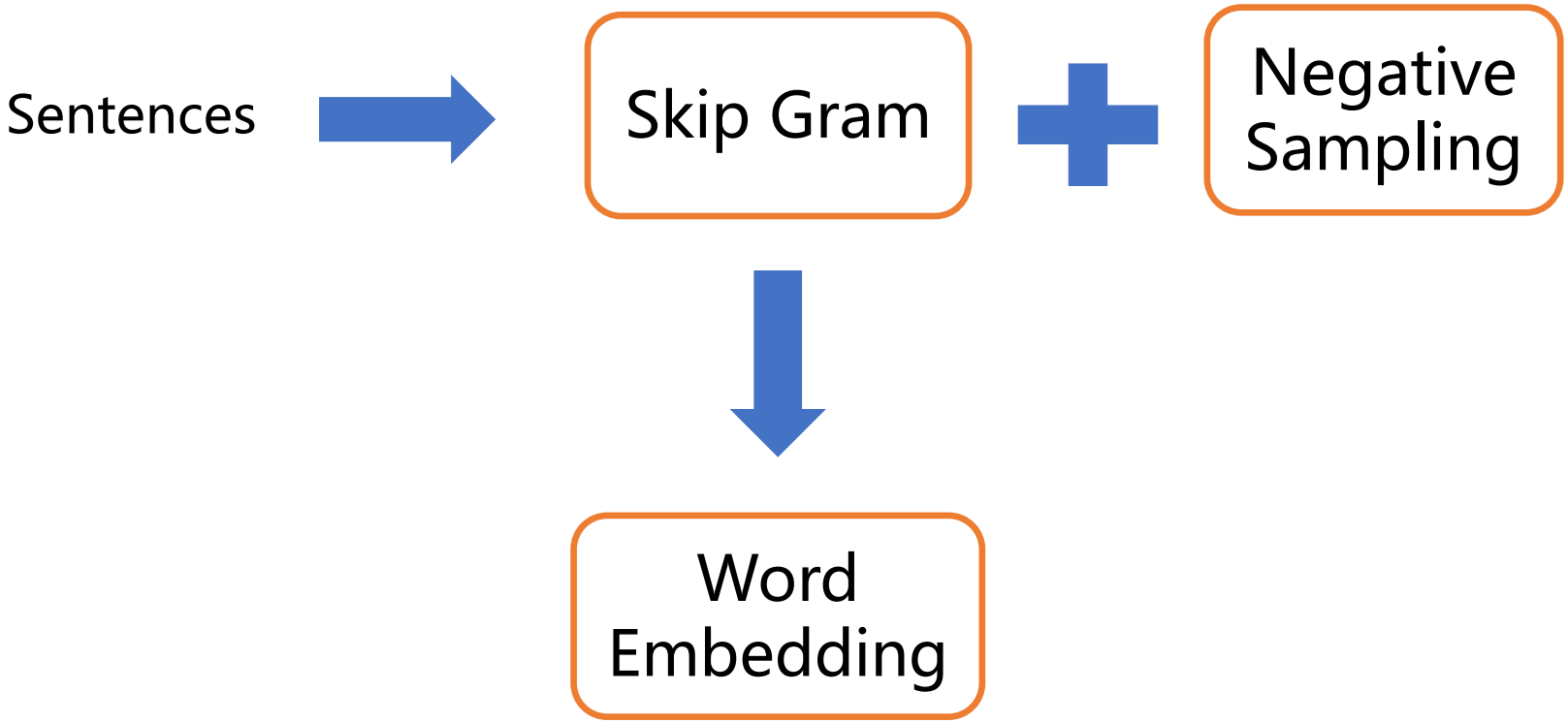
正样本: (orange, juice)　$Y = 1$
负样本: (orange, alien), (orange, happy), (orange, zero)　$Y = 0$

Softmax -> Multiple sigmoid

只对正样本和选取的负样本进行分类，从而减少计算量

Word2vec: Distributed Representations of Words and Phrases and their Compositionality

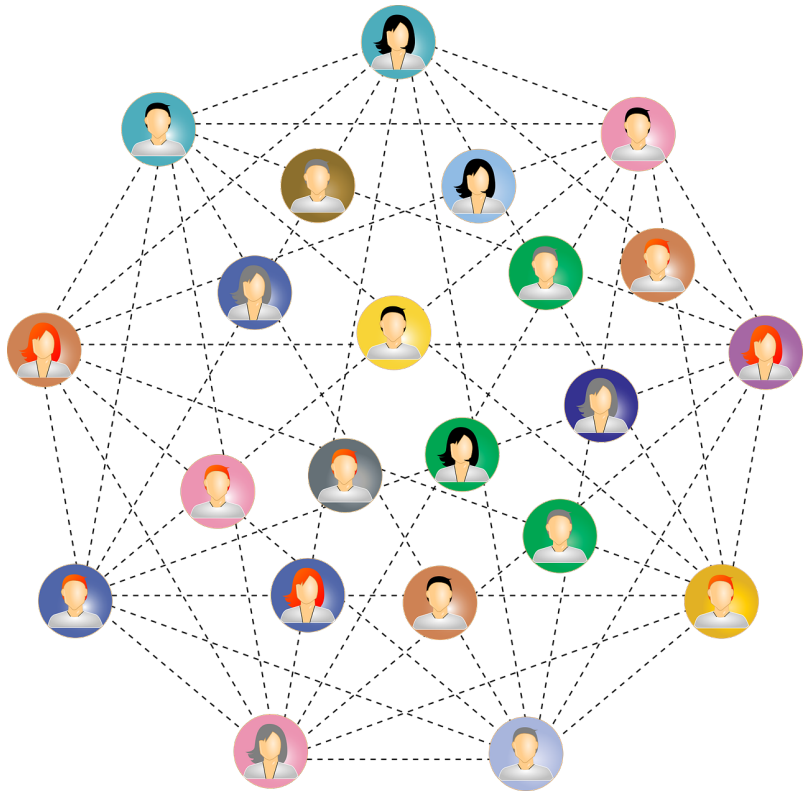
Word2vec: 整体架构



Word2vec: Distributed Representations of Words and Phrases and their Compositionality

Word2vec -> 图嵌入领域

例：社交网络



通常，图中的节点会受其邻居的影响。

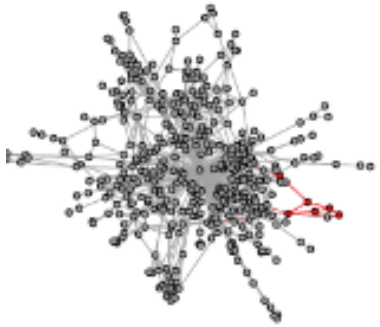
注意：前方即将迎来新内容

DeepWalk

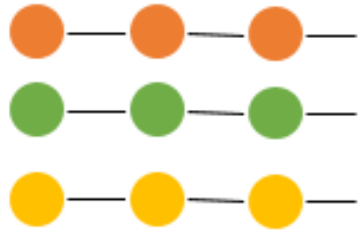
图游走类模型——DeepWalk

将 NLP 领域的思想运用到图(网络)嵌入领域。

DeepWalk



How?



节点 -> 单词
节点序列 -> 句子

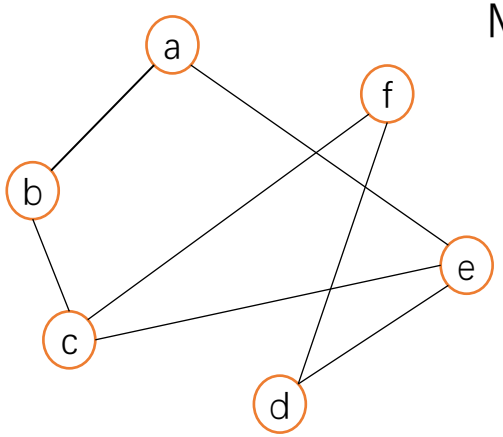
Word2Vec

Sentence1
Sentence2
Sentence3
...

DeepWalk: Online Learning of Social Representations

DeepWalk

游走方式：Random Walk



在无向图上游走

Max length = 3

d e a

c e a

b c e

a b a

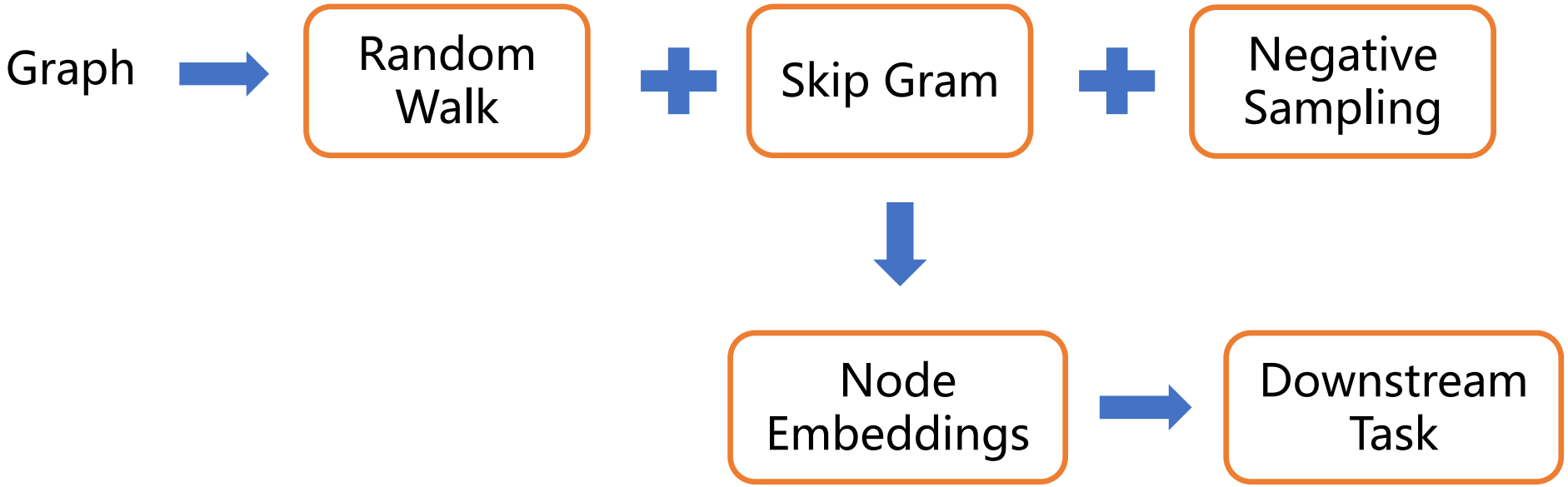
...

本质：可以**回头**的 DFS

$$P(c_i = x | c_{i-1} = v) = \begin{cases} \frac{\pi_{vx}}{Z}, & if\ (v,x) \epsilon E \\ 0, & otherwise \end{cases} = \frac{1}{|N(v)|}$$

下一个选择节点　当前节点

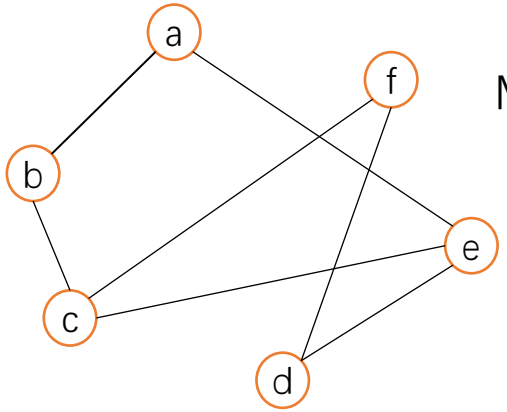
DeepWalk: 整体架构



DeepWalk: 简单却有效

node2vec: 对 DeepWalk 的改进

DeepWalk



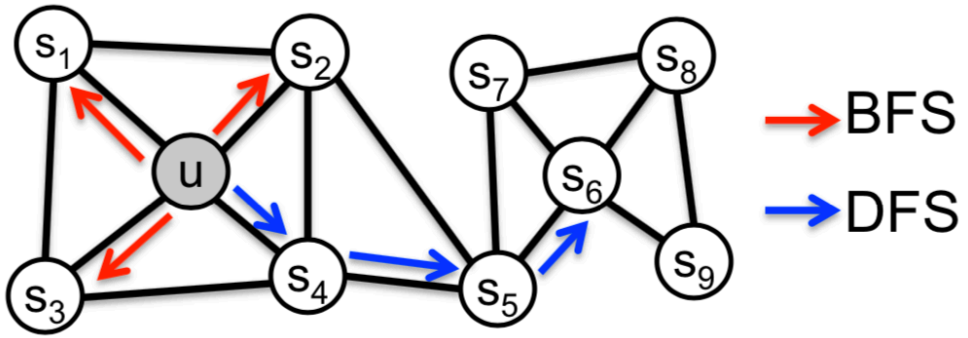
Max length = 3

d e a
c e a
b c e
a b a
...

在无向图上随机游走

- 简单直接：DFS
- 图是一个复杂结构，需要考虑更多因素

node2vec



node2vec: Scalable Feature Learning for Networks

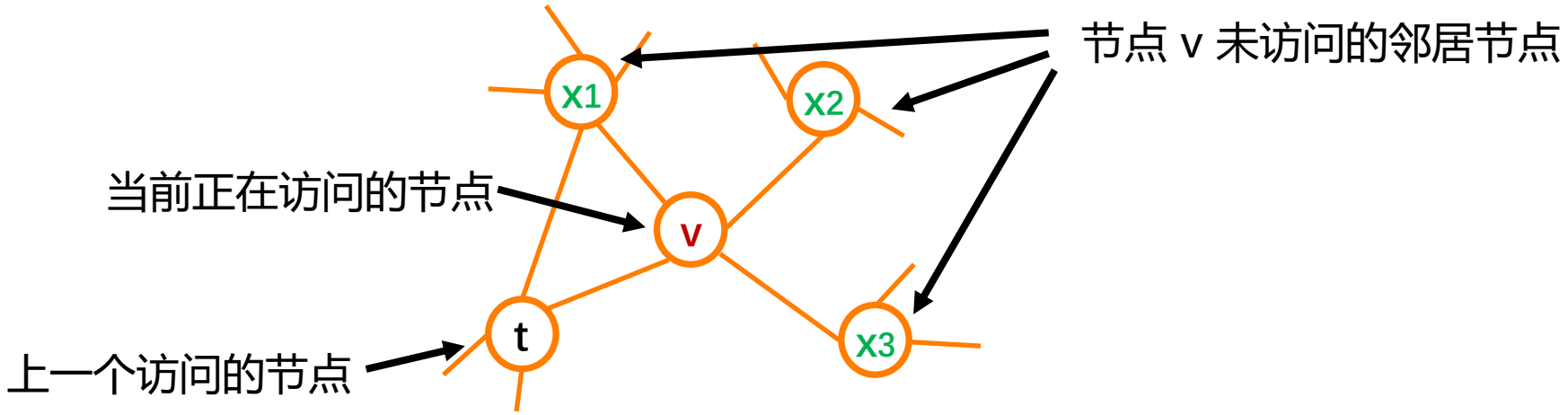
注意：前方即将迎来新内容

node2vec

node2vec: bias random walk

归一化后的转移概率分布

一般的随机游走公式 $$P(c_i = x | c_{i-1} = v) = \begin{cases} \frac{\pi_{vx}}{Z}, & if\ (v,x) \epsilon E \\ 0, & otherwise \end{cases}$$

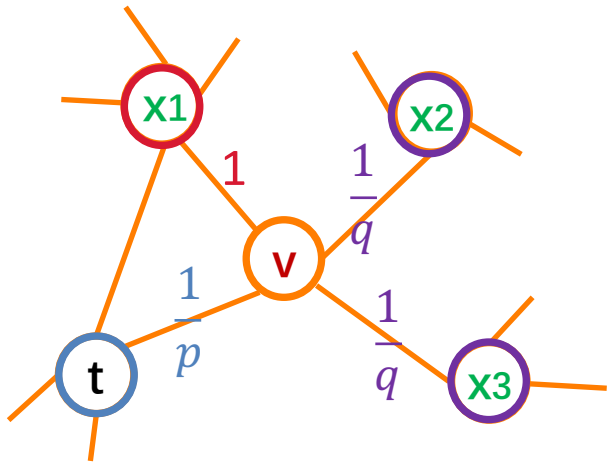


node2vec: Scalable Feature Learning for Networks

node2vec: bias random walk

$$\pi_{vx} = \alpha_{pq}(t,x) \cdot w_{vx}$$ 考虑带权图，边的权值

$$\alpha_{pq}(t,x) = \begin{cases} \frac{1}{p} & if\ d_{tx} = 0 \\ 1 & if\ d_{tx} = 1 \\ \frac{1}{q} & if\ d_{tx} = 2 \end{cases}$$



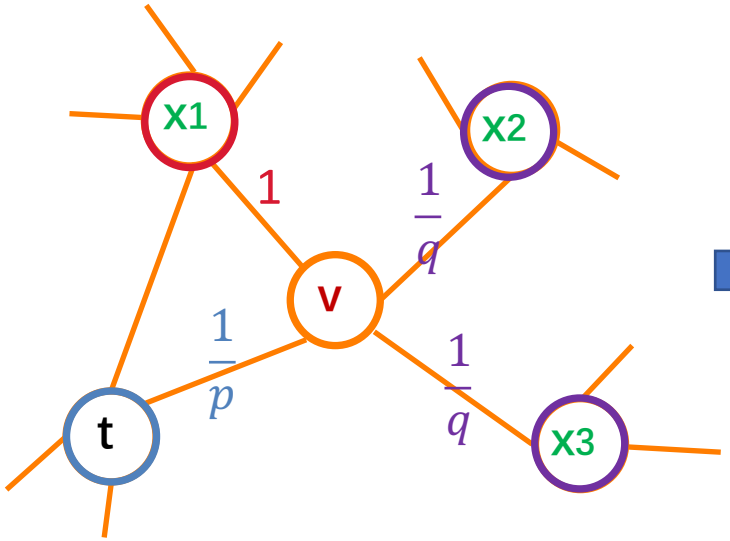
d_{tx}: 当前节点 v 的一阶邻居节点到节点 t 的距离

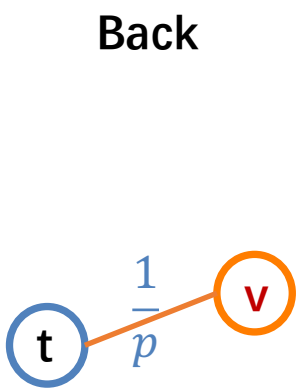
node2vec: Scalable Feature Learning for Networks

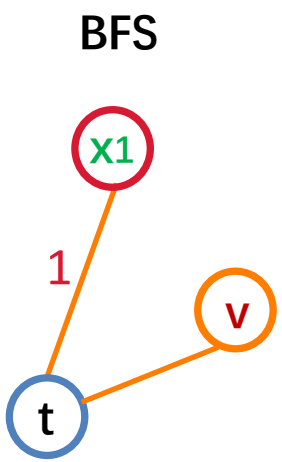
node2vec: bias random walk

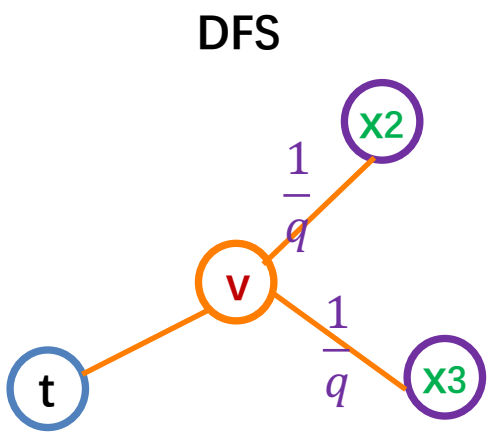
$$\pi_{vx} = \alpha_{pq}(t,x) \cdot w_{vx} \quad \alpha_{\mathrm{pq}}(t,x) = \begin{cases} \frac{1}{p} & if\ d_{tx} = 0 \\ 1 & if\ d_{tx} = 1 \\ \frac{1}{q} & if\ d_{tx} = 2 \end{cases}$$

- **p**: 控制随机游走以多大的概率 “**回头**”
- **参数q**: 控制随机游走偏向DFS 还是 BFS
  - q 较大时(q > 1)，倾向于 BFS；
  - q 较小时(q < 1)，倾向于 DFS；
- 当 **p=q=1**时，$\pi_{vx} = w_{vx}$.



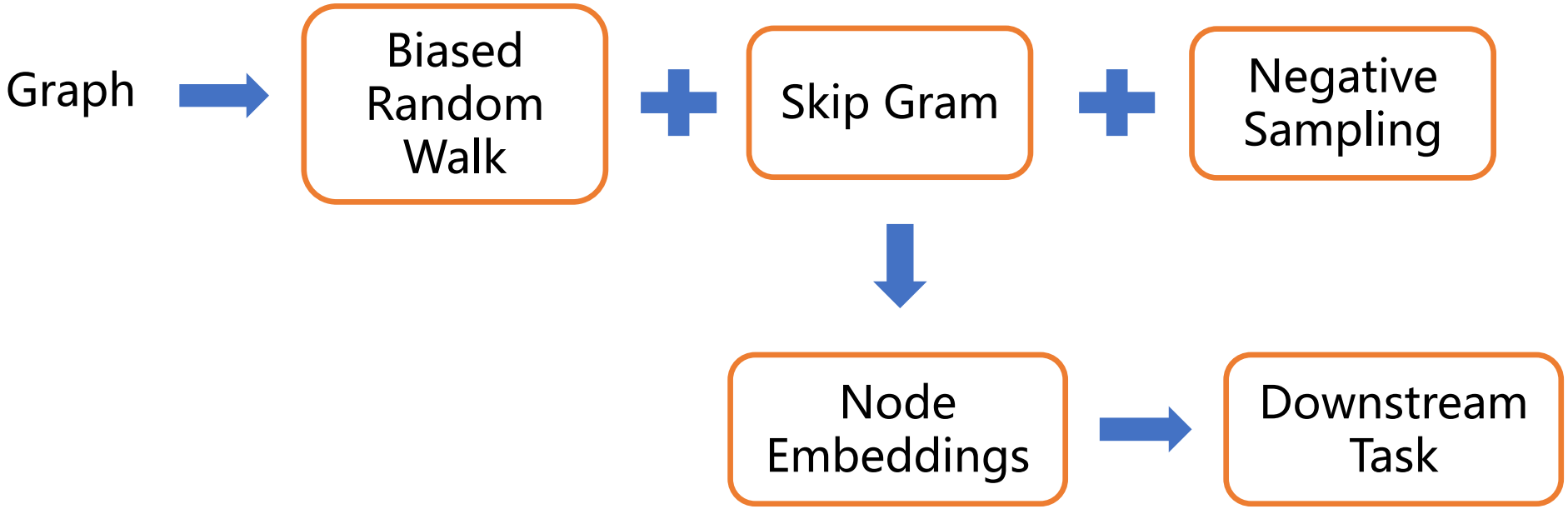






node2vec: Scalable Feature Learning for Networks

node2vec: 整体框架



异构图如何随机游走

不考虑节点类型的异构随机游走，**缺点**：

1. 偏向于出现频率高的节点类型；
2. 偏向于相对集中的节点(即度数高的节点)

DeepWalk
node2vec

metapath2vec

metapath2vec: Scalable Representation Learning for Heterogeneous Networks

注意：前方即将迎来新内容

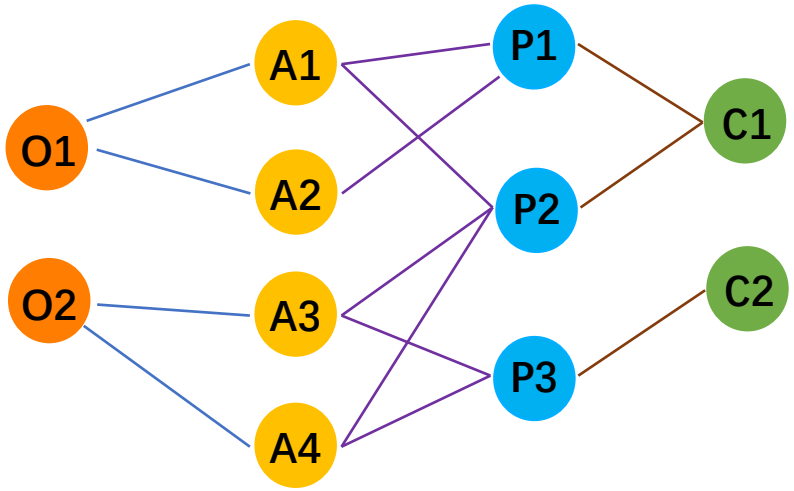
metapath2vec

异构图定义

$$G = (V, E, T)$$

Type

举例：学术网络



Organization 所属机构

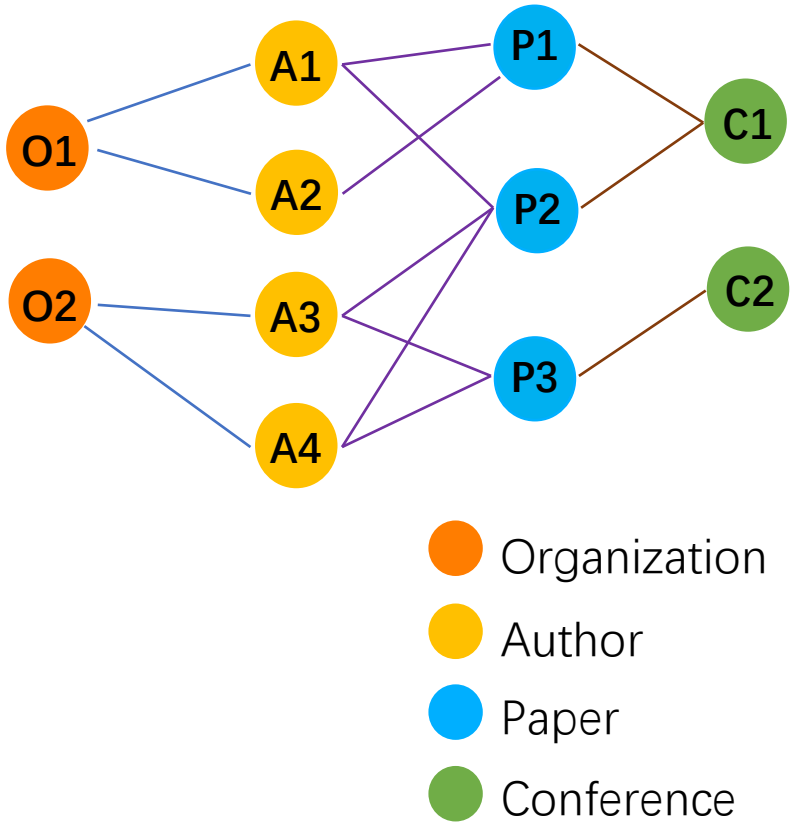
Author 作者

Paper 论文

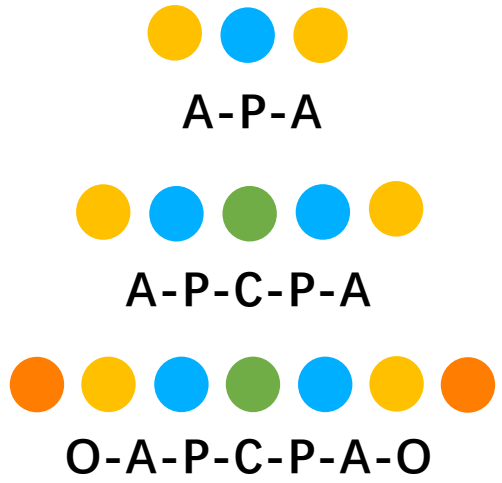
Conference 会议

metapath2vec: Scalable Representation Learning for Heterogeneous Networks

metapath2vec: meta path(元路径)



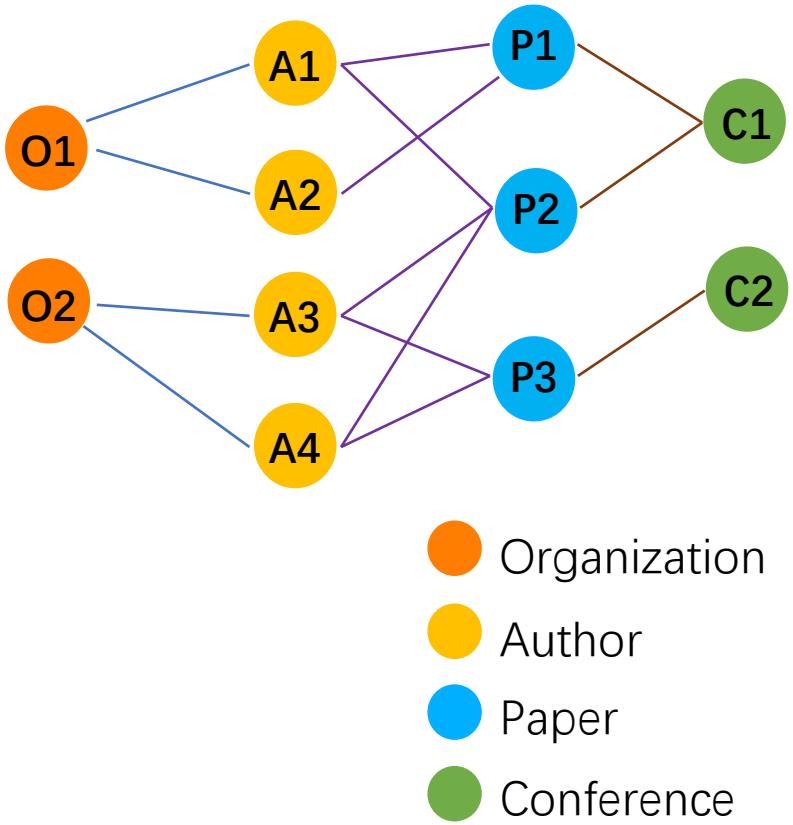
在图中选取的由节点类型构成的组合路径

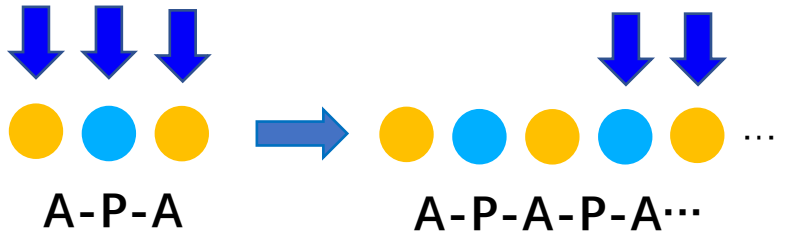


有意义

metapath2vec: Scalable Representation Learning for Heterogeneous Networks

metapath2vec: 基于meta path的随机游走

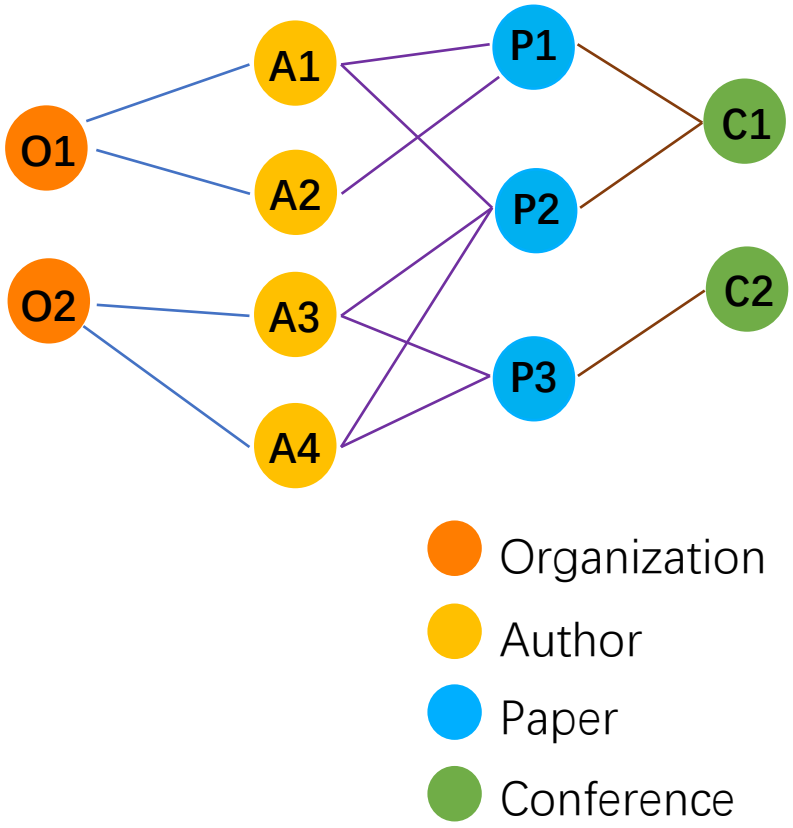




只要首尾节点类型相同，就可以继续游走

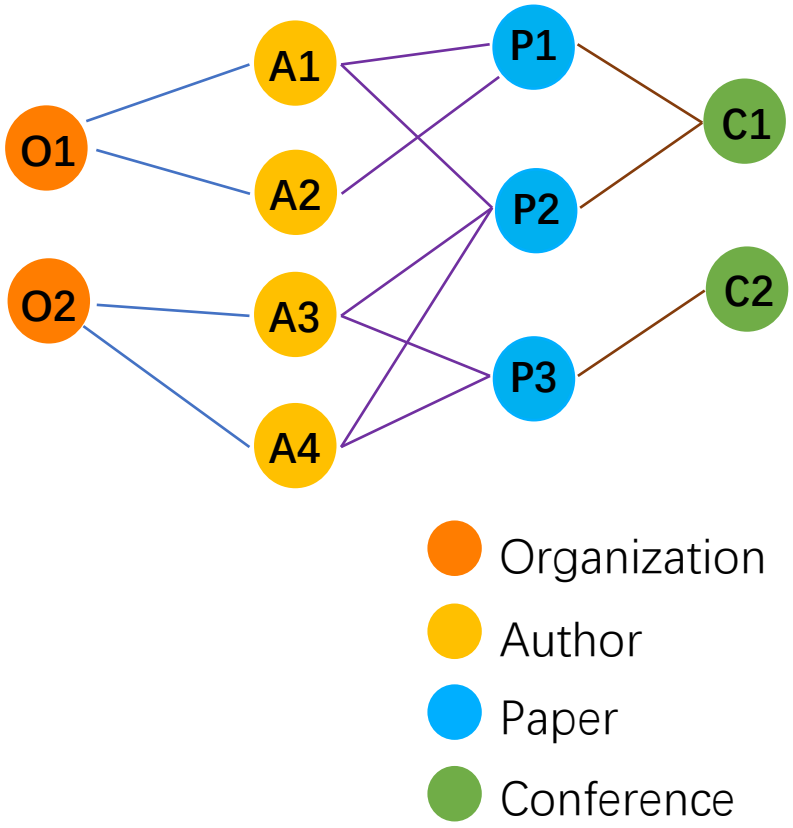
metapath2vec: Scalable Representation Learning for Heterogeneous Networks

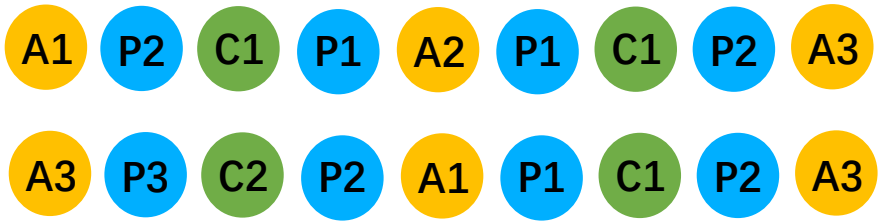
metapath2vec: 基于meta path的随机游走



metapath2vec: Scalable Representation Learning for Heterogeneous Networks

metapath2vec: 基于meta path的随机游走



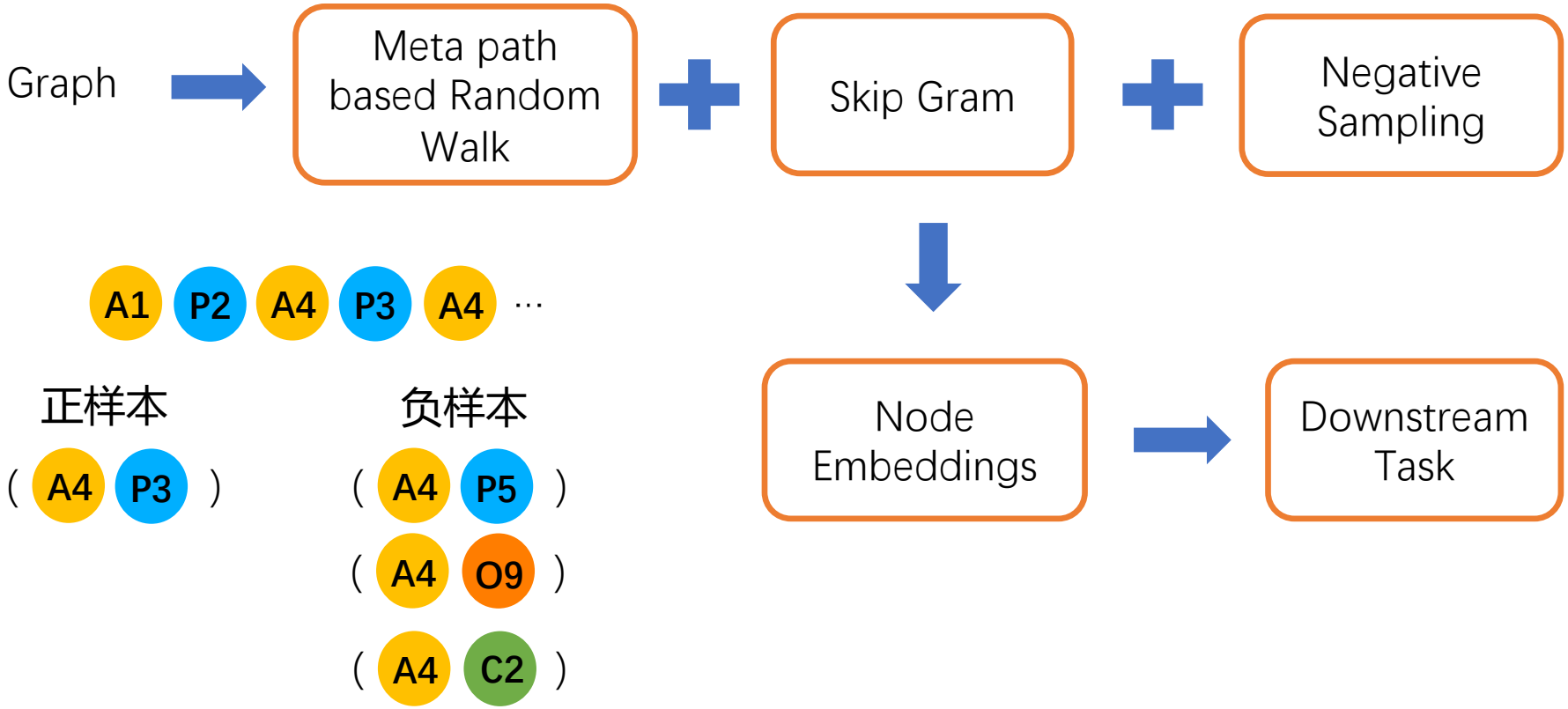


metapath2vec: Scalable Representation Learning for Heterogeneous Networks

metapath2vec: 整体框架

随机游走时考虑了节点类型。

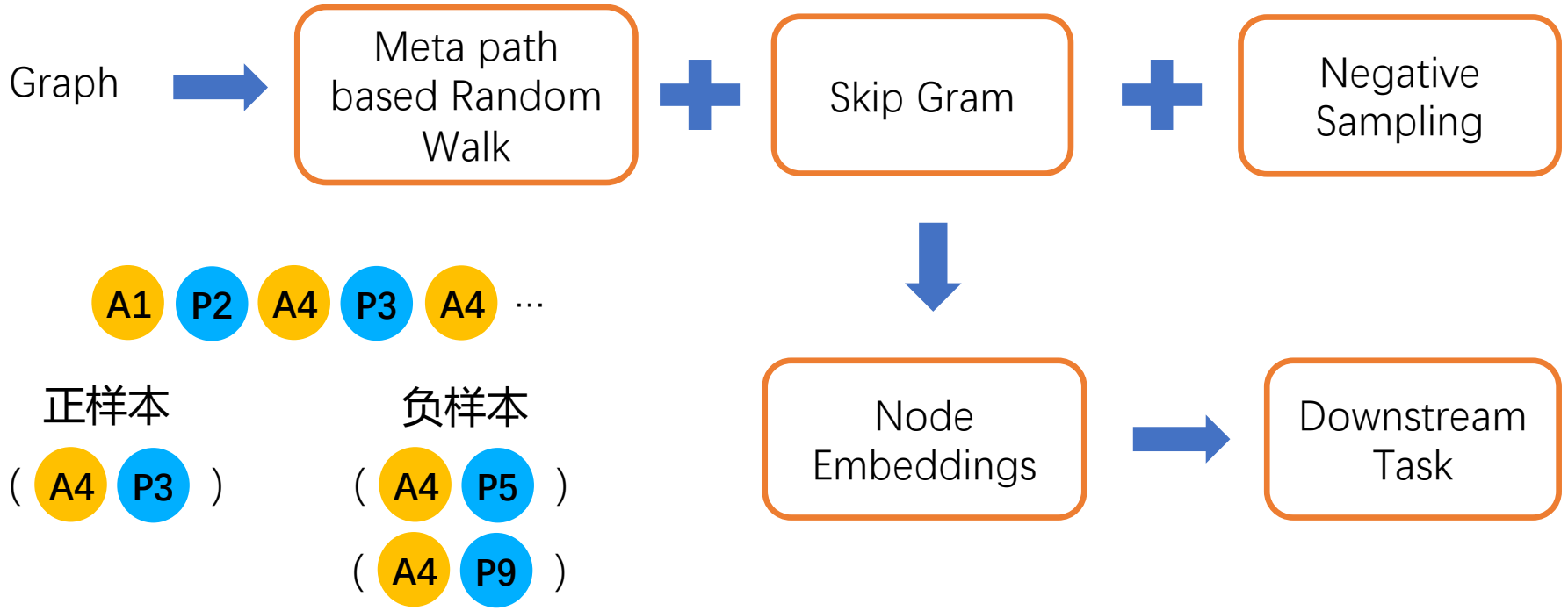
负采样的时候，没有考虑节点类型。



metapath2vec: Scalable Representation Learning for Heterogeneous Networks

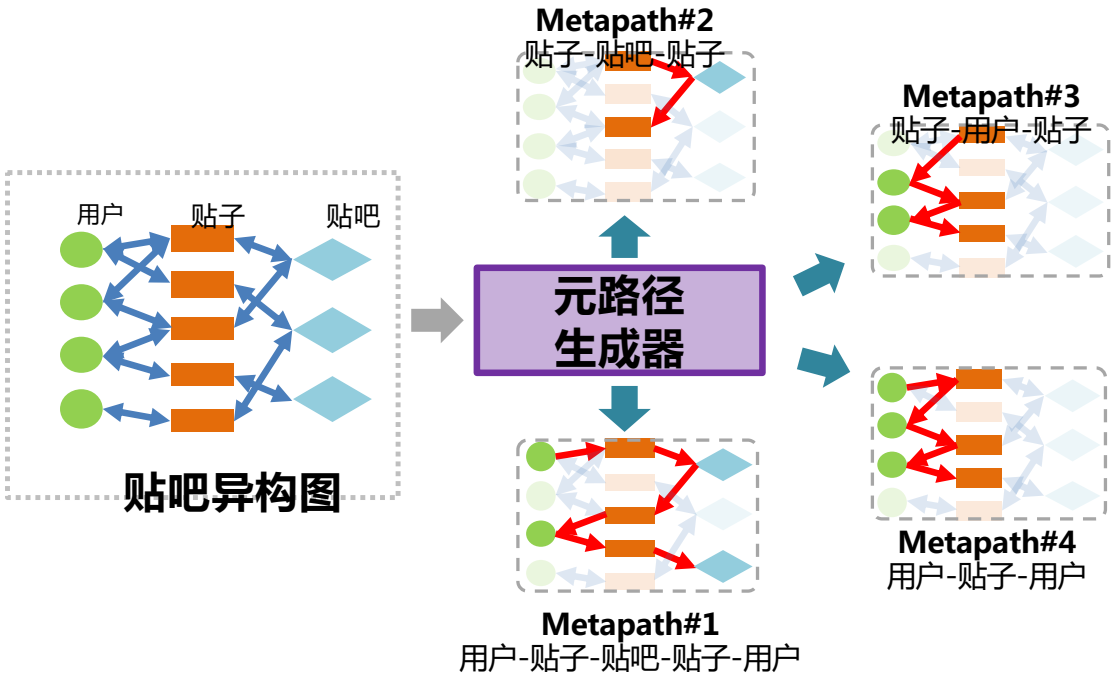
metapath2vec++: 整体框架

负采样的时候，考虑节点类型。

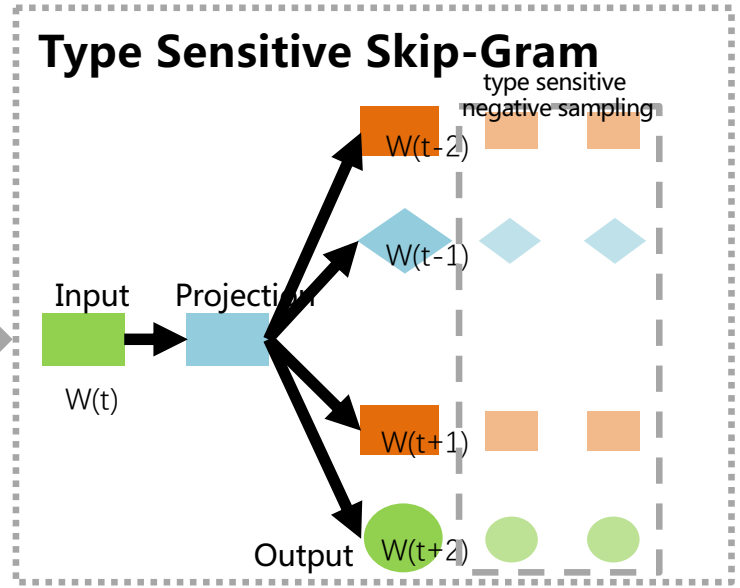


metapath2vec: Scalable Representation Learning for Heterogeneous Networks

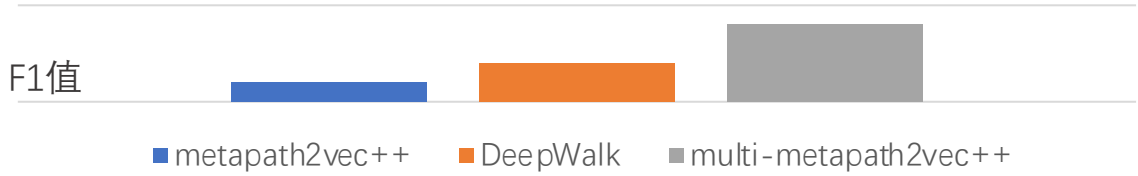
变种：multi-metapath2vec++



模型落地：贴吧场景

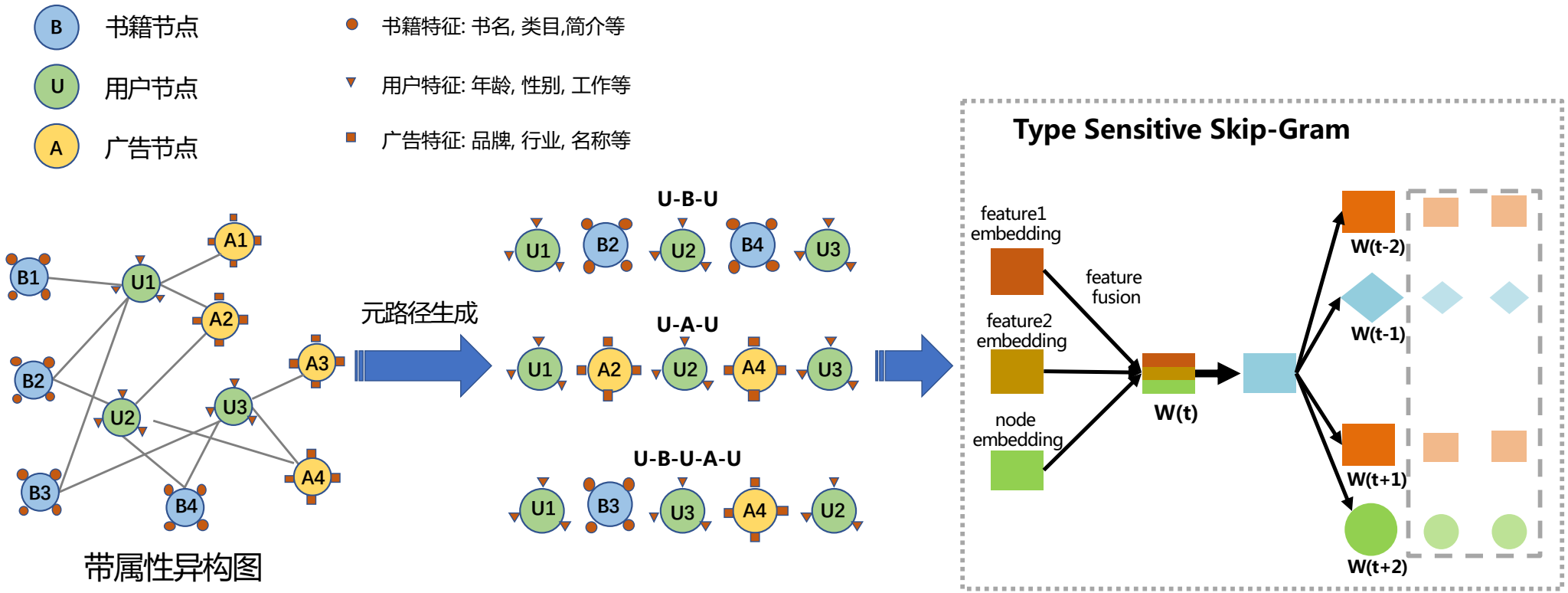


贴吧场景下表现

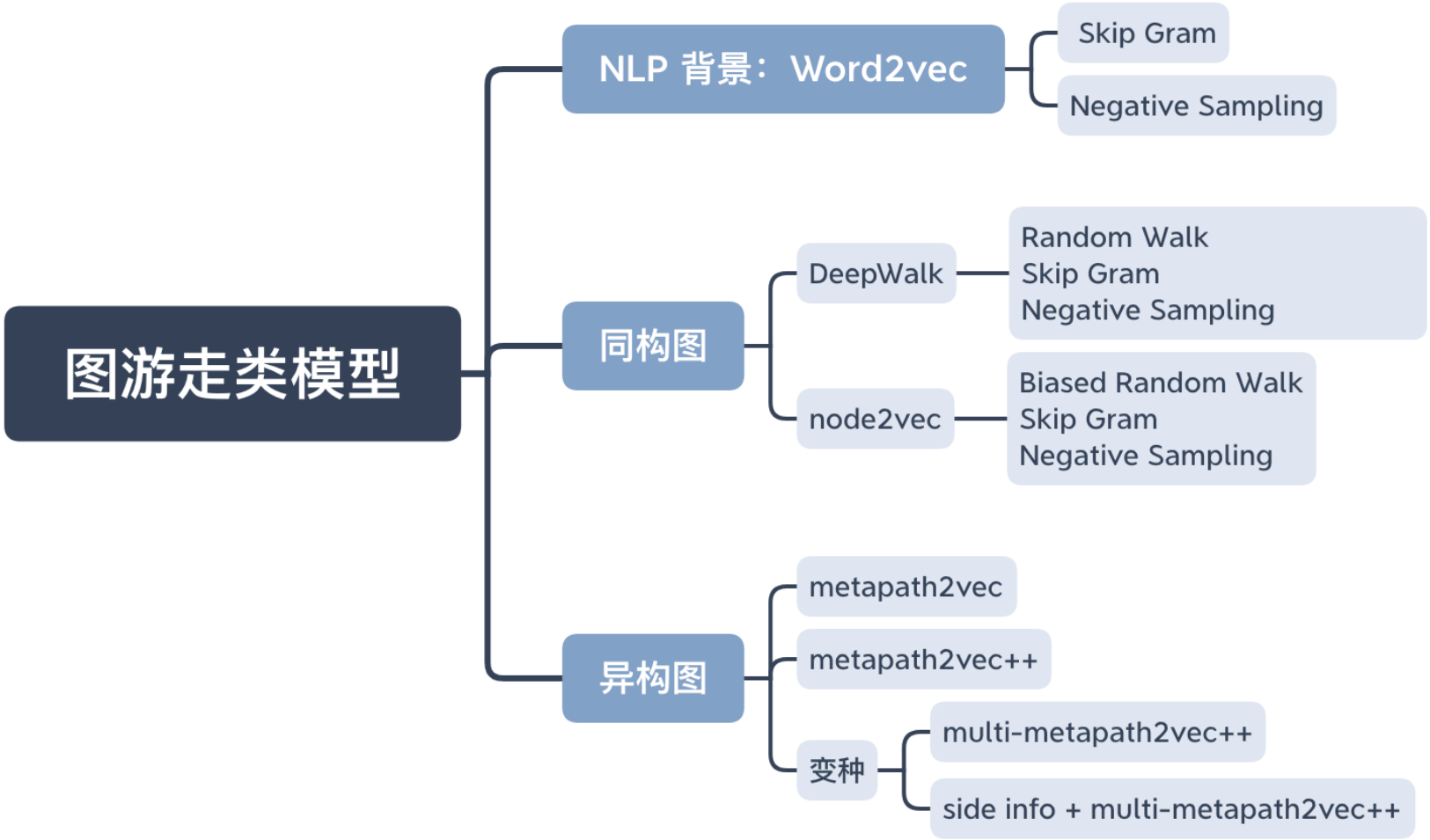


变种：side info+multi-metapath2vec++

模型落地：手百小说场景



总结


图游走类模型
NLP 背景：Word2vec
Skip Gram
Negative Sampling
同构图
DeepWalk
Random Walk
Skip Gram
Negative Sampling
node2vec
Biased Random Walk
Skip Gram
Negative Sampling
异构图
metapath2vec
metapath2vec++
变种
multi-metapath2vec++
side info + multi-metapath2vec++

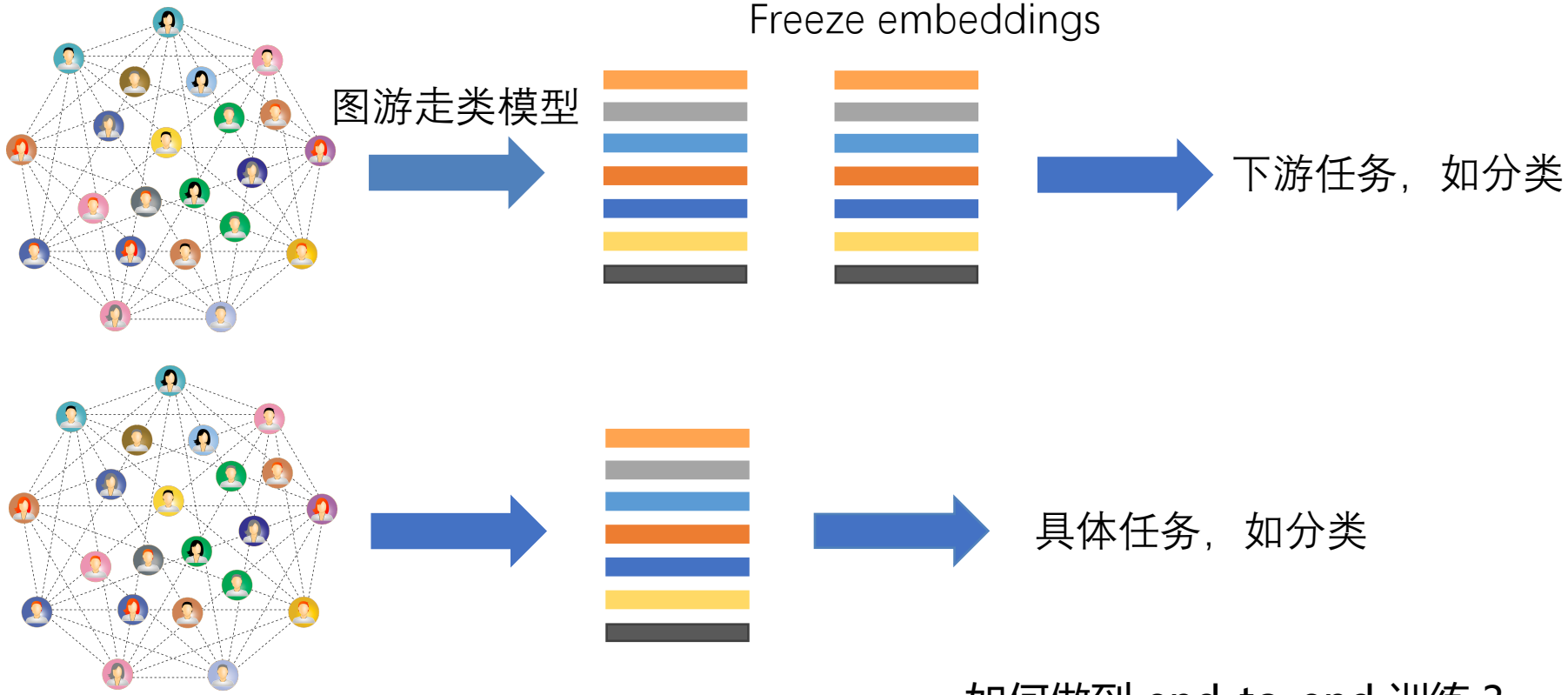
课堂实践

DeepWalk 游走代码讲解

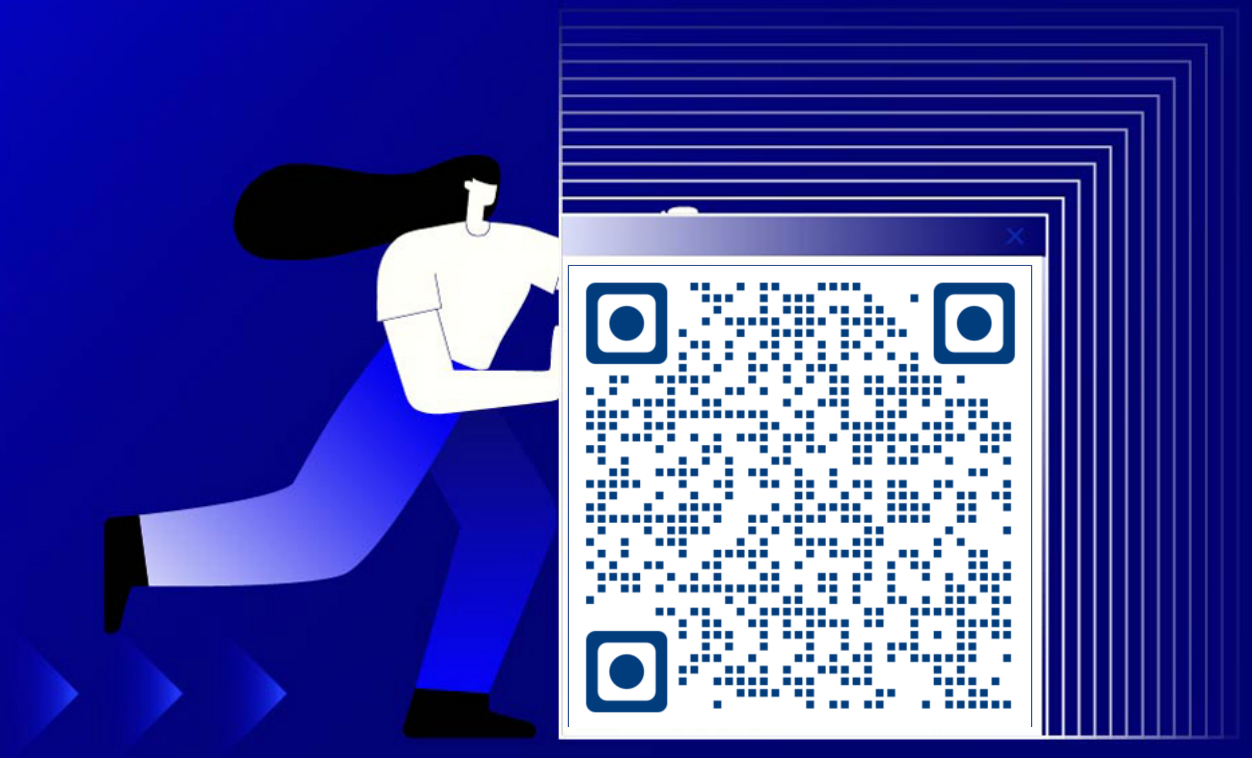
课后作业

动手作业：完成 DeepWalk和 node2vec 的采样算法部分，并且对 Skip Gram 模型进行补充实现。

下节课预告



如何做到 end-to-end 训练？



PGL github

谢谢观看