

Beijing Forest Studio
北京理工大学信息系统及安全对抗实验中心



动态异质图神经网络

硕士研究生 杨景然

2025年07月06日

- 总结反思
 - 讲解语速较快，时间较短，语气词较多
 - 内容详略不当，算法部分讲解深度不足、浅尝辄止
 - 互动部分较少
- 相关内容
 - 2025.02.16 杨景然 《属性缺失异质图神经网络》
 - 2021.06.14 李新帅 《动态网络嵌入方法研究》

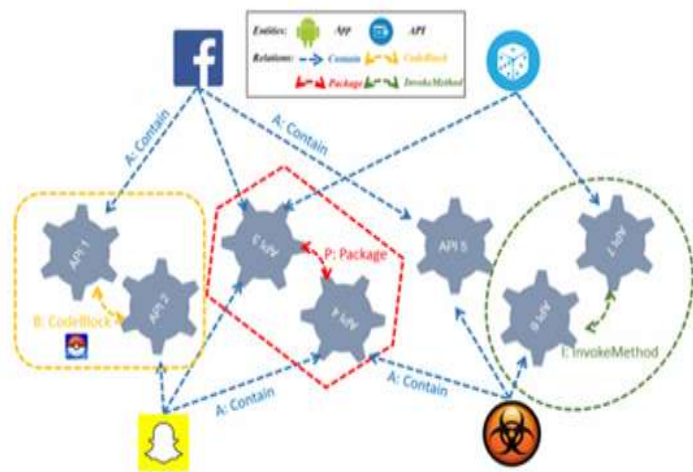
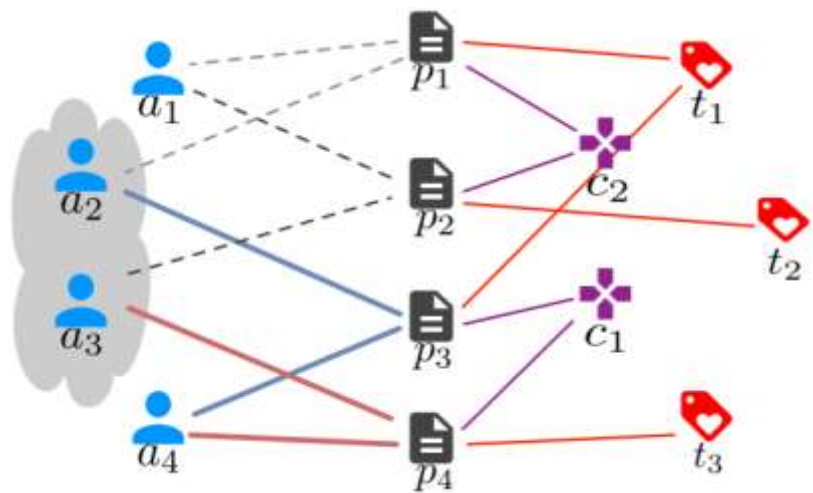
- 预期收获
- 内容引入
- 知识基础
- 内涵解析与研究目标
- 研究背景与研究意义
- 算法原理
 - DyHATR
 - DHINE
- 特点总结与工作展望
- 参考文献



- 预期收获
 - 掌握动态图、动态异质图神经网络的基本概念
 - 了解动态异质图神经网络发展历程
 - 理解动态异质图神经网络的基本原理
 - 明确动态异质图神经网络应用和前沿发展

- 社交网络中有哪些动态关系？
 - 新用户注册、老用户注销
 - 关注和取消关注
 - 用户年龄、位置、内容偏好等属性改变
- 网络攻击中有哪些动态关系？
 - 攻击链路变化
 - 终端防御成功/失败
 - 网络拓扑改变
 - 新漏洞增加、旧情报失效...
- 节点、边、节点属性都**随时间发生变化**

动态图更加贴近实际应用



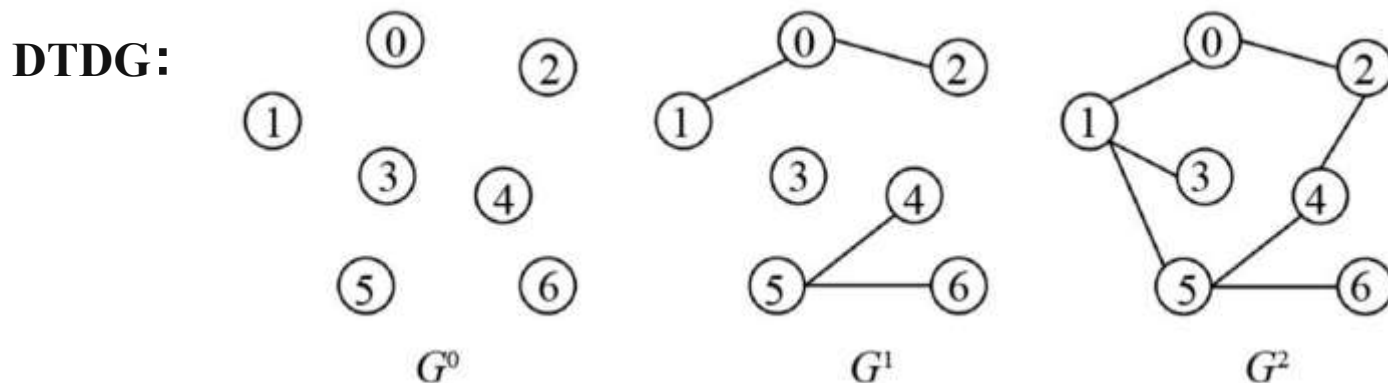


• 基本概念

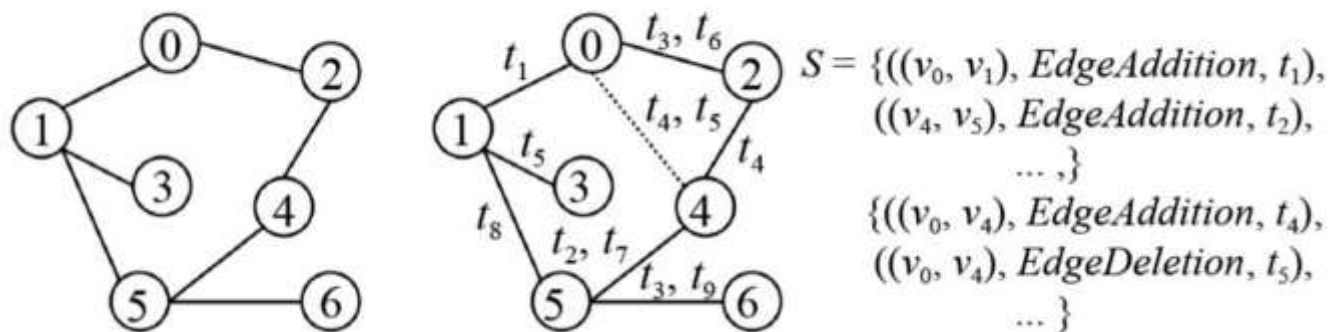
- 动态图：涉及时间演化的复杂图结构，节点、边和属性显示出连续变化的状态
- 分类：离散时间动态图、连续时间动态图
- 离散时间动态图（Discrete-time dynamic graphs, DTDG）
 - 以等间隔对动态图进行快照得到一个离散的静态图演化序列
 - $\mathcal{G} = \{G^0, G^1, \dots, G^T\}$, 其中第 t 个快照： $G^t = (V^t, E^t)$ ($0 \leq t \leq T$)
- 连续时间动态图（Continuous-time dynamic graphs, CTDG）
 - CTDG定义为 $\mathcal{G} = \{G, S\}$, 其中 G 为 t_0 时的初始图结构
 - S 表示图形事件的集合, $S = \{event^1, event^2, \dots, event^T\}$
 - 每个 $event$ 由三元组定义：参与事件的节点、事件类型、时间戳

- 基本概念

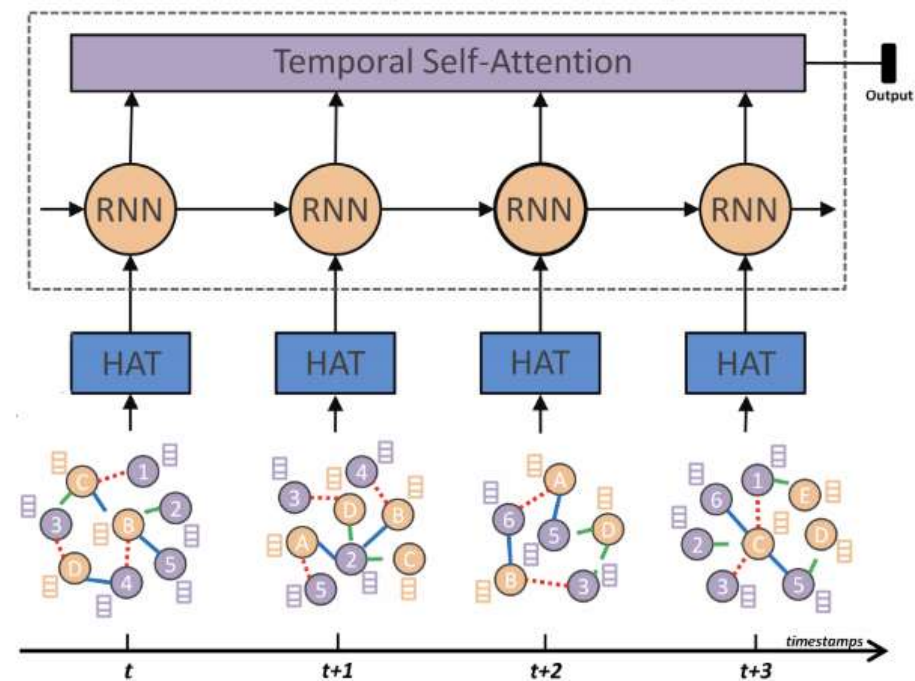
- DTDG: 快照形式节省存储空间，在选择的时间间隔内可能会**丢失**一些复杂的细节
- CTDG: 能够完整记录图上的变化，但存储、计算**消耗大**，**建模困难**



CTDG:



- 内涵解析
 - 动态异质图神经网络：建模图结构（节点/边）和属性随时间的变化：连续事件流 or 离散快照
 - 捕捉突发性、周期性、趋势性等动态模式
- 研究目标
 - 以**动态异质图**数据为研究对象
 - **结合**时序模型（LSTM、TCN、霍克斯过程）
 - 与异质图实现时空联合嵌入
 - 设计同时保留异质语义和时序演化的嵌入方法
 - 满足聚类、分类、预测、分割、生成等下游任务
 - 解决静态HGNN在应用时的局限性



- 研究背景
 - 动态性：现实世界中许多系统都可以建模为**动态异质图**（社交网络、金融交易、网络安全）
 - 传统静态图模型（GAT、GCN）无法建模时序演化，**难以适应**动态数据
- 研究意义
 - 为复杂动态系统提供新的**计算范式**
 - 支持增量学习、解决动态图中的噪声、缺失问题
 - 为金融、社交、安全、医疗等领域提供**分析工具**



研究历史 属性缺失异质图神经网络

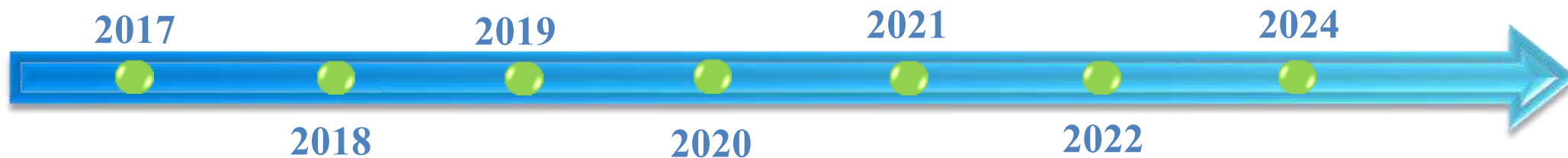


Li等人提出DANE方法采用离线学习的方法学习节点嵌入，并根据网络和属性随时间的变化更新嵌入

Yin等人提出了DNHE方法：基于快照子图构建了综合的历史-当前网络在其上进行随机游走，并使用动态异构跳图模型来学习嵌入

Wang等人提出TEDIC模型，同时在空间和**时间维度**上使用注意机制，将GNN与一维时空卷积结合

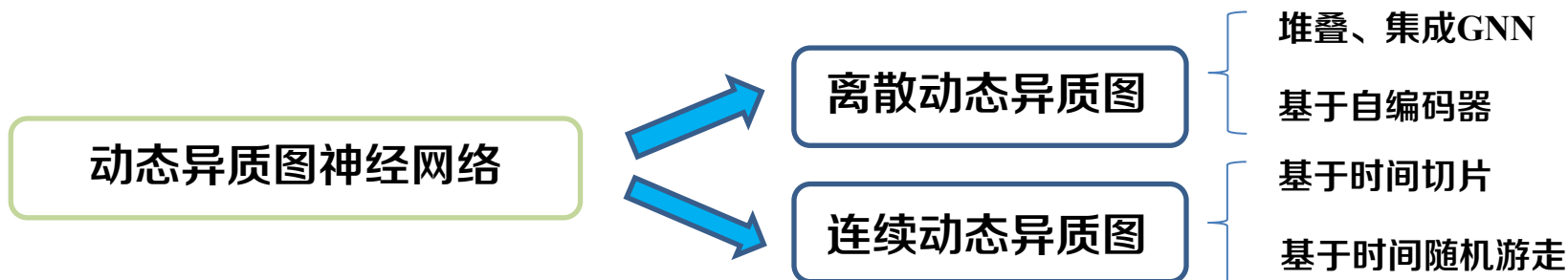
Li等人提出DHANE方法：使用静态捕获更多的拓扑信息，并通过在每个卷积层中**执行基于图元的注意力机制**保持GCN的收敛性



Goyal P等人提出**dyngraph2vec**方法，将动态图的**演化过程编码为低维向量**，同时保留图的结构信息和时序依赖

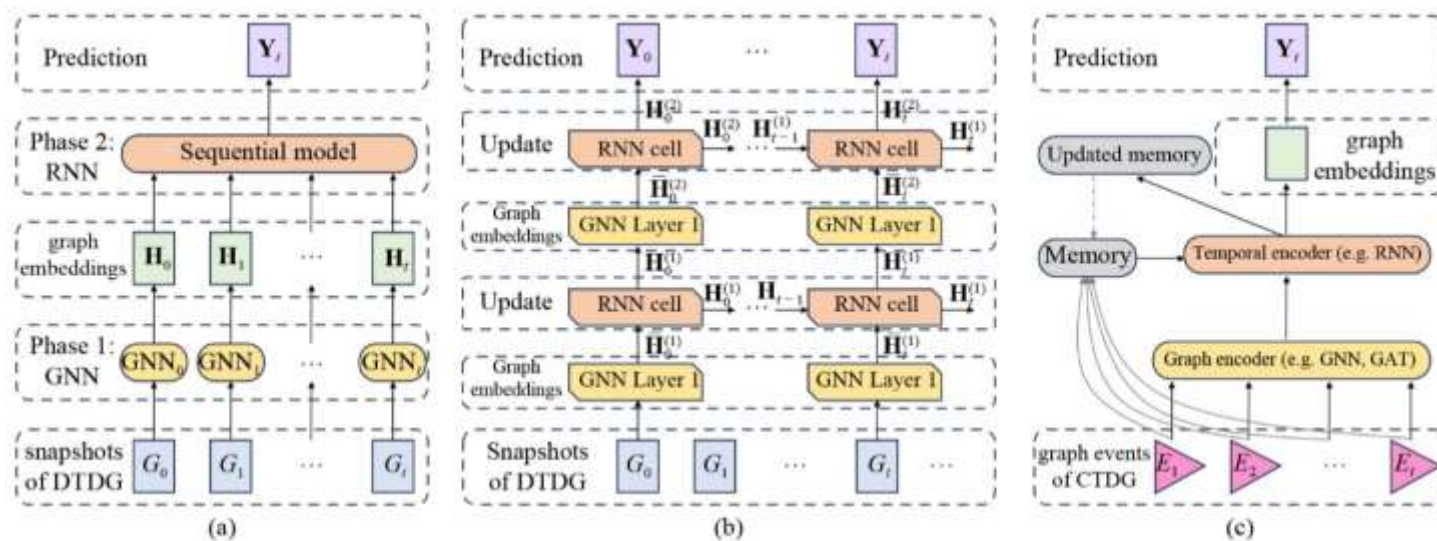
Manessi F等人出了CD-GCN模型：通过 LSTM **更新节点表示**，同时引入Skip 来缓解梯度消失问题，增强长时序依赖的建模能力

Sun H等人提出GHT方法：**Hawkes过程**对节点间的时间依赖性进行建模，分别使用连续时间LSTM和Transformer估计强度函数，学习节点的时间演化表示



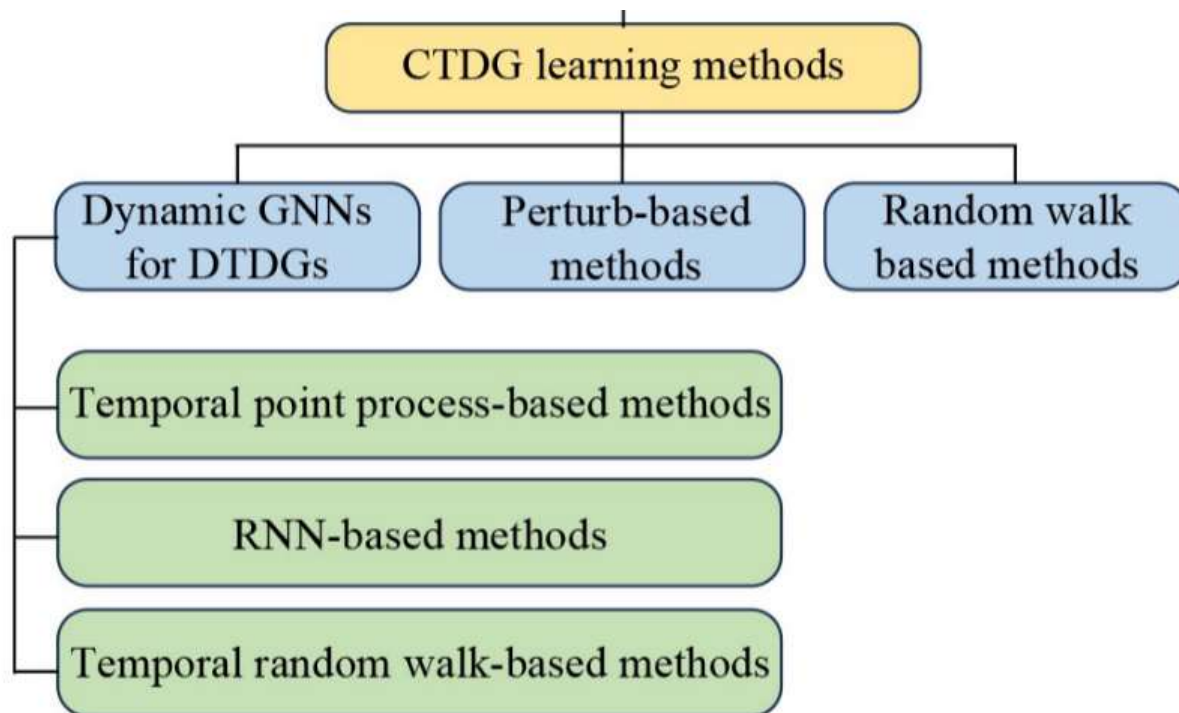
- DTDG

- 堆叠GNN: 将空间GNN和时间RNN以**模块化**的方式结合起来, 对动态图的离散序列进行建模
- 集成GNN: 将GNN和RNN合并在一个**网络层**内, 从而在同一架构层内合并时空建模
- 基于自编码器: 利用动态图嵌入来捕获拓扑及其时间变化



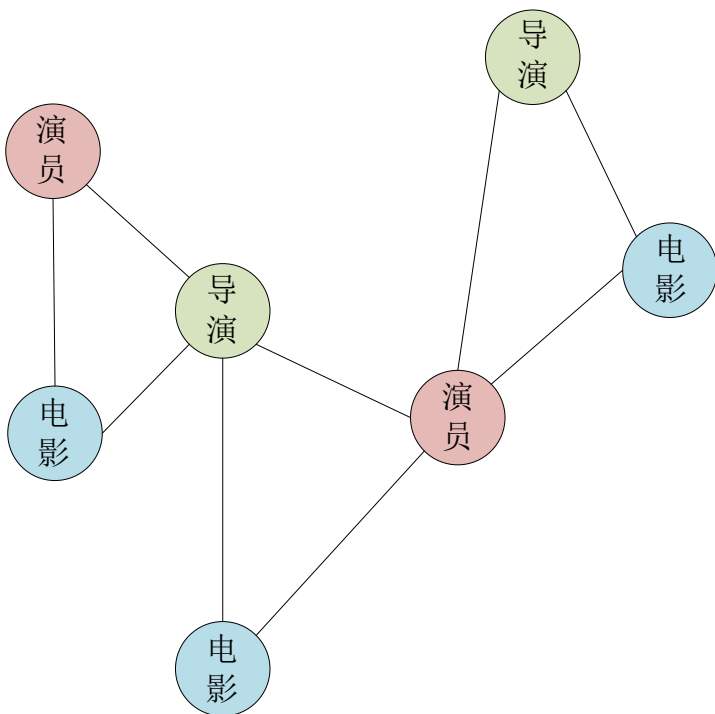
- CTDG

- 基于时间点：将事实的发生建模为一个多变量时间点过程
- 基于RNN：使用RNN以连续的方式不断计算节点嵌入
- 基于时间随机游走：随机漫步由具有非递减时间戳的边序列组成

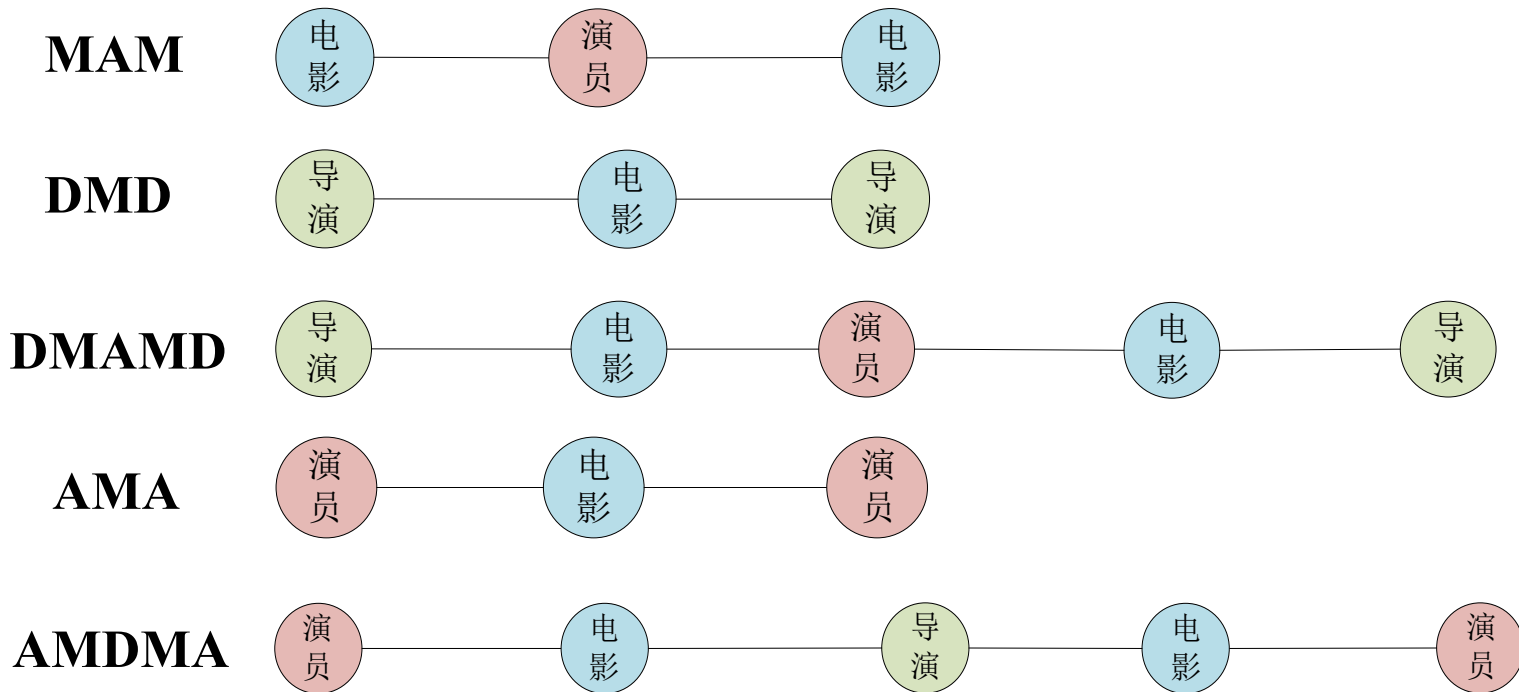


• 元路径

- 连接不同类型节点的一条路径
- 不同的元路径会有不同的路径类型，通常用节点类型路径来表示

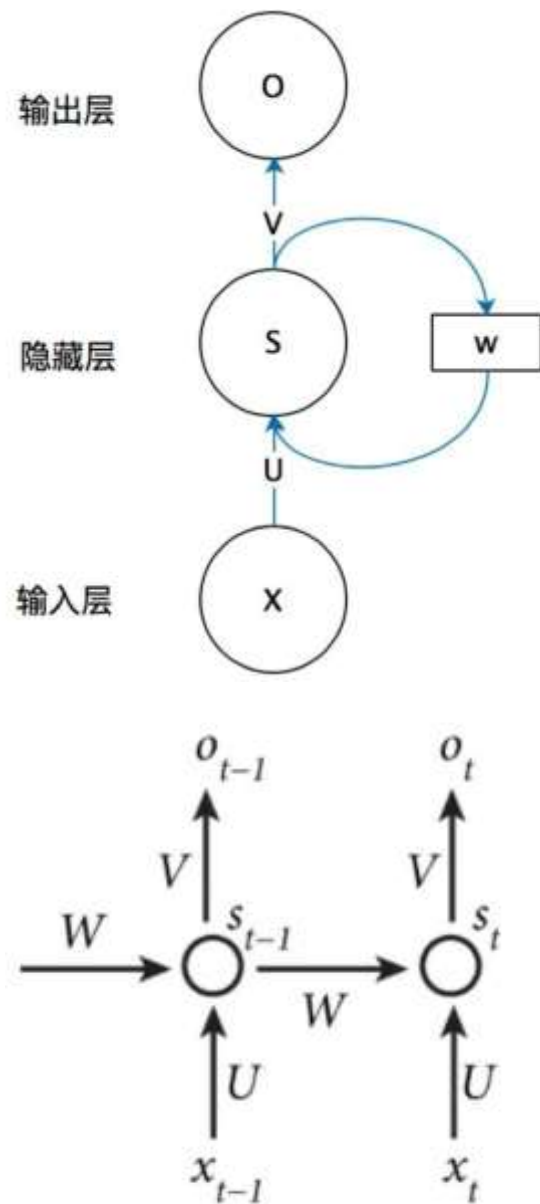


电影 (M) 导演 (D) 演员 (A)



- 递归神经网络（Recurrent Neural Network, RNN）
 - 循环连接：隐藏层神经元之间存在循环连接，使网络能够保留前序步骤的信息
 - 参数共享：同一层的权重在时间步之间共享，减少了参数量，使模型能处理任意长度的序列：
 - $o^t = g(V \cdot S_t) \quad S_t = f(U \cdot X_t + W \cdot S_{t-1})$
 - W 是每个时间步之间的权重矩阵，整个训练过程中不变
 - 每一时刻的隐藏层不仅由该时刻的输入层决定，还由上一时刻的隐藏层决定

挖掘数据中的时序信息



• 长短期记忆网络 (Long short-term memory, LSTM)

– 核心: 选择性地存储信息

– 方法: 门控机制

• Input Gate: 决定此时刻是否有信息输入

• Output Gate: 决定此时刻是否有信息输出

• Forget Gate: 决定是否清除记忆值

– 机理:

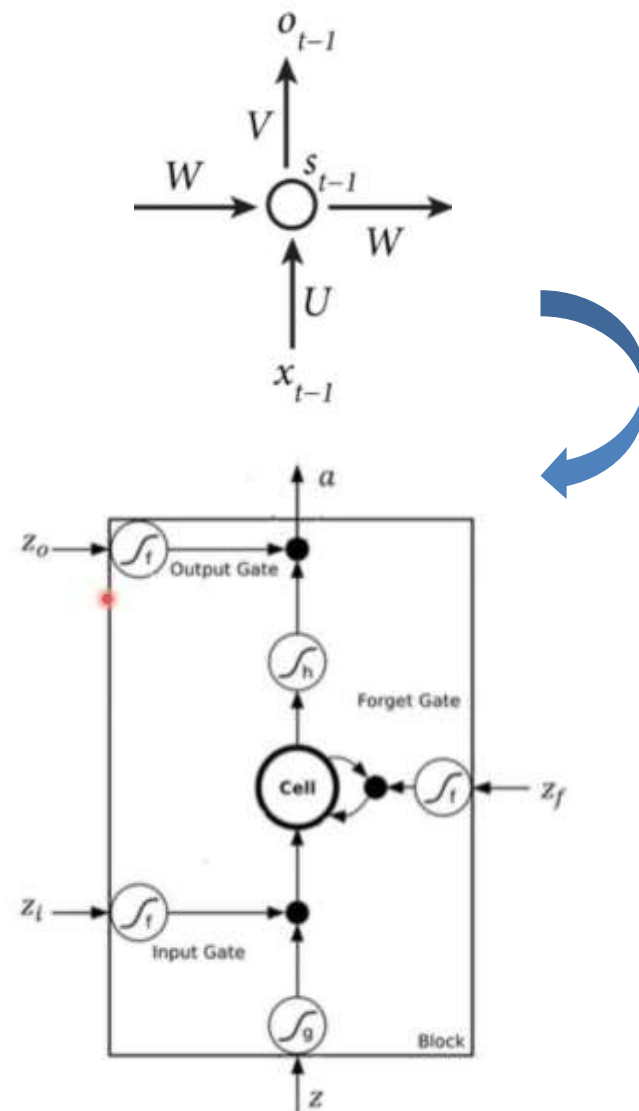
• $Z = \tanh(W[x_t, h_{t-1}])$ 真正的输入

• $Z_i = \sigma(W_i[x_t, h_{t-1}])$

• $Z_f = \sigma(W_f[x_t, h_{t-1}])$

• $Z_o = \sigma(W_o[x_t, h_{t-1}])$

• $Z_i, Z_f, Z_o \in [0, 1]$, 控制门的“开关程度”





- 特征值扰动/摄动理论 (Eigenvalue Perturbation Theory)
 - 研究在矩阵发生微小变化时, 其特征值、特征向量等性质的变化规律
 - 设一个原始矩阵 A , 在它的基础上加入一个小扰动矩阵 E , 得到扰动后的矩阵 $\tilde{A} = A + E$, 研究:
 - 特征值: $\lambda_i(A)$ 与 $\lambda_i(\tilde{A})$
 - 特征向量: $x_i(A)$ 与 $x_i(\tilde{A})$
 - 特征值摄动
 - $(A + E)(x + \Delta x) = (\lambda + \Delta\lambda)(x + \Delta x)$
 - Weyl定理 $A = A^*$, $\tilde{A} = A + E$, 两者的特征值排序为 $\lambda_1 \geq \lambda_2 \geq \cdots \lambda_n$, $\tilde{\lambda}_1 \geq \tilde{\lambda}_2 \geq \cdots \tilde{\lambda}_n$ 则 $|\lambda_i - \tilde{\lambda}_i| \leq \|E\|_2$



Modeling Dynamic Heterogeneous Network for Link Prediction using Hierarchical Attention with Temporal RNN

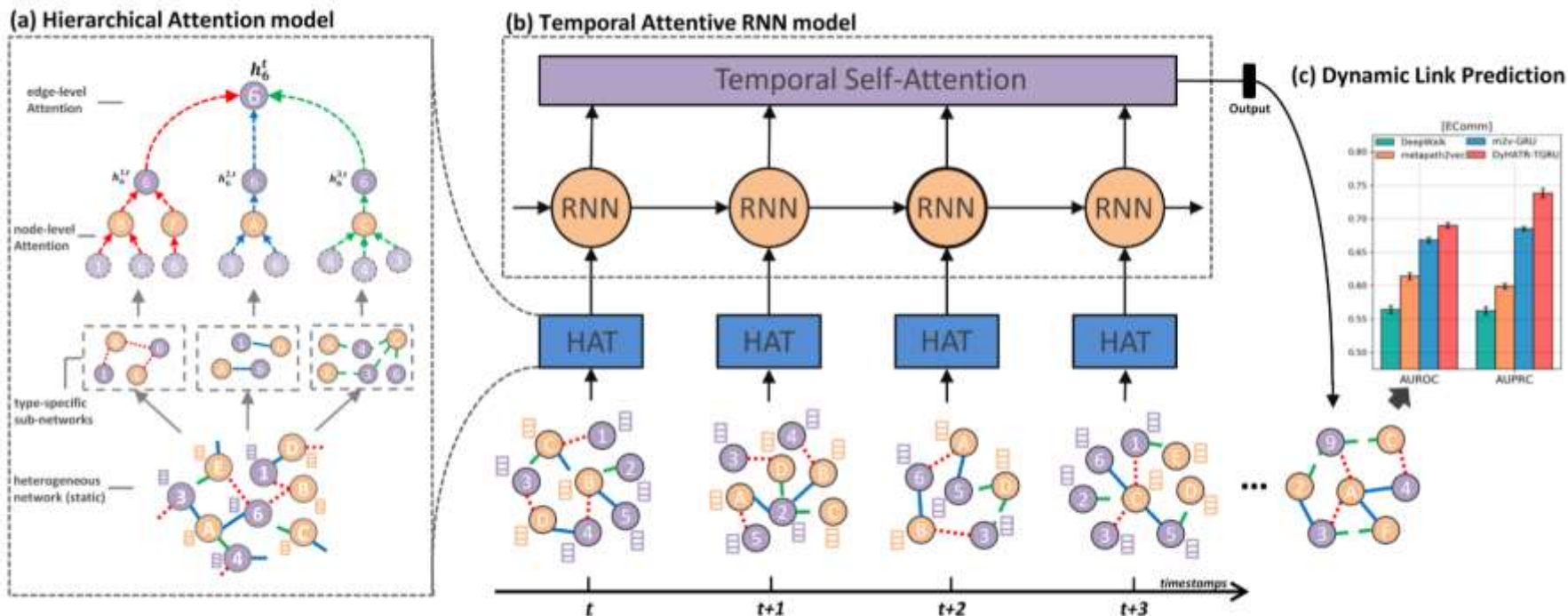


T	目标	生成包含历史信息的节点嵌入
I	输入	一段时间内的图数据快照序列*3 Twitter（2012年7月1日至7日期间的三种用户行为（转发、回复和提及） Math-Overflow（11个时间快照下用户之间的三种不同类型的交互） Ecomm（11个快照内的购物行为）
P	处理	1.静态处理：每个时间切片进行节点级和边级的属性注意力聚合 2.时序处理：用GRU/LSTM捕捉节点 <i>i</i> 的历史时序依赖，并对所有时间步进行注意力加权，得到
O	输出	节点嵌入矩阵*3

P	问题	1.现有方法简单拼接时间步特征，时间信息利用不足
C	条件	1.输入必须是有序的异构网络快照序列 2.每个快照需要明确的节点类型和边类型定义
D	难点	如何有效建模每个时间快照的静态嵌入，并对快照序列进行加权
L	水平	ECML-PKDD 2020（CCF B）

• 算法原理图

- 分层注意力模块：节点级注意力+边级注意力
- 循环神经网络模块：在RNN输出端添加时间注意力层，使用缩放点积注意力机制来学习不同快照的嵌入



化动为静



• 分层注意力模块

– 节点级注意力【捕获单个边类型特定的信息】

- 每个边类型子图上独立运行，计算节点与其邻居的重要性权重
- 加权聚合邻居特征生成节点表示

$$\alpha_{i,j}^{rt} = \frac{\exp(\sigma(\mathbf{a}_r^\top \cdot [\mathbf{W}^r \cdot \mathbf{x}_i || \mathbf{W}^r \cdot \mathbf{x}_j]))}{\sum_{k \in N_i^{rt}} \exp(\sigma(\mathbf{a}_r^\top \cdot [\mathbf{W}^r \cdot \mathbf{x}_i || \mathbf{W}^r \cdot \mathbf{x}_k]))}$$

节点对 (i, j) 快照 t 的权重系数

$$\hat{\mathbf{h}}_i^{rt} = \sigma\left(\sum_{j \in N_i^{rt}} \alpha_{i,j}^{rt} \cdot \mathbf{W}^r \mathbf{x}_j\right)$$

节点 i 在边类型 R 下的嵌入

– 边级注意力机制【整合不同类型的边信息】

- 加权聚合各边类型的节点表示

$$\beta_i^{rt} = \frac{\exp(\mathbf{q}^\top \cdot \sigma(\mathbf{W} \cdot \mathbf{h}_i^{rt} + \mathbf{b}))}{\sum_{r \in R} \exp(\mathbf{q}^\top \cdot \sigma(\mathbf{W} \cdot \mathbf{h}_i^{rt} + \mathbf{b}))}$$

$$\mathbf{h}_i^t = \sum_{r=1}^R \beta_i^{rt} \cdot \mathbf{h}_i^{rt}$$

节点 i 在快照 t 的嵌入



• 循环神经网络模块

– 传统方法

- 假设RNN的输出为 $\{s_1^t, s_2^t, \dots, s_i^t\}$ ，传统方法可能将这些状态向量进行**拼接**或者最后
一个状态 s_i^t 作为节点 i 的最终嵌入
- 信息**丢失**、忽略了不同时间步的重要性差异

– 时序级自注意力模型

- **缩放点积注意力机制**【自注意力的变体】
- 利用输入序列中各个位置的 查询（Query）、键（Key）和 值（Value）来计算注意力权重

$$Z_i = \Gamma_i \cdot V_i = \text{softmax}\left(\frac{(S_i W_q)(S_i W_k)^T}{\sqrt{D'}} + M\right) \cdot (S_i W_v)$$

- S_i ：节点 i 在所有时间步下的RNN状态矩阵
- 可学习参数： W_q, W_k, W_v ：查询（Query）、键（Key）和 值（Value）

- 模型端到端训练

- 从原始网络快照直接学习节点最终嵌入
- 目标是同时捕获动态异质网络中的异质信息和时间演化信息
 - 损失函数：负采样对比损失
 - 目标是最小化相连节点的距离、最大化不相连节点的距离

$$L(\mathbf{z}_u^T) = \sum_{v \in N^T(u)} -\log(\sigma(\langle \mathbf{z}_u^T, \mathbf{z}_v^T \rangle)) - Q \cdot \mathbb{E}_{v_n \sim P_n(v)} \log(\sigma(\langle -\mathbf{z}_u^T, \mathbf{z}_{v_n}^T \rangle)) + \lambda \cdot L_p$$

- 方法

- 固定长度的随机游走构建节点 u 的邻居集 $N^T(u)$
- 从负采样分布中抽取 Q 个负样本计算距离

- 优势：

- 端到端训练避免分阶段训练的信息损失
- 不同快照之间的参数共享，可以处理大批量时间序列



• 数据资源

– 数据集:

数据集	节点数	边数	节点类型数	边类型数	时间快照数
Twitter	10000	63410	1	3	7
Math-Overflow	24818	506550	1	3	11
Ecomm	37724	91033	2	4	11
Alibaba.com	16620	93956	2	3	11

• 对比方法

- 静态同质网络方法: DeepWalk, GraphSAGE, GAT
- 静态异质网络方法: metapath2Vec
- 动态同质网络方法: DynamicTriad、dyngraph2Vec, DySAT
- 动态异质网络方法: MetaDynaMix, change2vec, DHNE, NP-GLM

• 评价指标

- AUROC
- AUPRC
- 节点分类准确率



- 数据集设置

- 随机抽取20%的边作为验证集调参，剩余80%用于链路预测；其中再随机选择25%作为训练集，75%作为测试集。
- 随机抽取相等数量的**无链接节点对**，分别作为训练集和测试集的反例
- 使用两个节点的嵌入特征的内积作为链接的特征

- LSTM与GRU

- LSTM：三重门控结构、有额外记忆细胞专用于长期记忆保存
- **GRU：双重门（更新门、重置门）、无额外记忆细胞**
- LSTM更适用于长序列建模
- GRU**更轻量**：参数少、计算量低、内存占用小

实验结果

- 基于余弦相似度来聚合特征空间中top-k个相似邻居的属性，使Pearson相关性和Heat Kernel得到很大的提高

Methods	Twitter		Math-Overflow		EComm		Alibaba.com	
	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC
DeepWalk	0.520(0.040)	0.686(0.021)	0.714(0.002)	0.761(0.001)	0.564(0.006)	0.562(0.006)	0.546(0.019)	0.570(0.008)
GraphSAGE-mean	0.600(0.024)	0.765(0.016)	0.683(0.005)	0.712(0.003)	0.635(0.007)	0.642(0.008)	0.571(0.019)	0.610(0.014)
GraphSAGE-meanpool	0.562(0.009)	0.711(0.005)	0.671(0.007)	0.697(0.008)	0.600(0.018)	0.590(0.017)	0.552(0.015)	0.570(0.021)
GraphSAGE-maxpool	0.571(0.031)	0.731(0.021)	0.637(0.008)	0.660(0.009)	0.594(0.010)	0.587(0.009)	0.542(0.009)	0.573(0.008)
GraphSAGE-LSTM	0.546(0.016)	0.710(0.010)	0.672(0.005)	0.697(0.008)	0.599(0.013)	0.588(0.015)	0.567(0.006)	0.585(0.007)
GAT	0.587(0.019)	0.744(0.013)	0.737(0.004)	0.768(0.003)	0.648(0.003)	0.642(0.008)	0.572(0.010)	0.610(0.008)
metapath2vec	0.520(0.040)	0.686(0.021)	0.714(0.002)	0.761(0.001)	0.614(0.005)	0.599(0.004)	0.561(0.004)	0.601(0.007)
DynamicTriad	0.626(0.027)	0.785(0.013)	0.704(0.005)	0.759(0.003)	0.614(0.008)	0.688(0.007)	0.543(0.013)	0.598(0.011)
dyngraph2vec-AE	0.549(0.025)	0.724(0.021)	0.517(0.006)	0.565(0.005)	0.503(0.002)	0.509(0.006)	0.506(0.019)	0.534(0.014)
dyngraph2vec-AERNN	0.511(0.034)	0.706(0.031)	0.582(0.003)	0.598(0.003)	0.503(0.004)	0.501(0.004)	0.505(0.008)	0.554(0.045)
DySAT	0.634(0.003)	0.796(0.005)	0.506(0.008)	0.543(0.006)	0.512(0.002)	0.513(0.002)	0.521(0.013)	0.550(0.010)
metapath2vec-GRU	0.539(0.025)	0.744(0.018)	0.730(0.004)	0.770(0.004)	0.668(0.004)	0.685(0.003)	0.544(0.003)	0.576(0.004)
metapath2vec-LSTM	0.554(0.027)	0.754(0.017)	0.727(0.004)	0.771(0.003)	0.666(0.005)	0.683(0.008)	0.547(0.003)	0.578(0.008)
DHNE	0.552(0.005)	0.649(0.010)	0.678(0.001)	0.720(0.001)	0.547(0.003)	0.626(0.004)	0.515(0.001)	0.561(0.003)
DyHATR-TGRU	0.649(0.005)	0.805(0.007)	0.741(0.002)	0.781(0.001)	0.690(0.004)	0.738(0.007)	0.590(0.008)	0.602(0.008)
DyHATR-TLSTM	0.660(0.010)	0.810(0.009)	0.751(0.001)	0.790(0.002)	0.696(0.005)	0.734(0.005)	0.605(0.003)	0.617(0.005)

- 算法贡献

- 首次实现分层注意力与时序RNN的有机融合
- 证明**注意力+RNN**组合在动态图学习的有效性【基于时间快照的经典范式】
- 同步支持GRU和LSTM
- 端到端的模块结构可以灵活使用

- 算法不足

- 没有对快照的时间间隔参数进行验证，鲁棒性未知
- **长时序**依赖处理不足：数据集中最长时间序列仅为11
- 时间**复杂度高**，相较于metapath2vec-GRU等时间增加25%





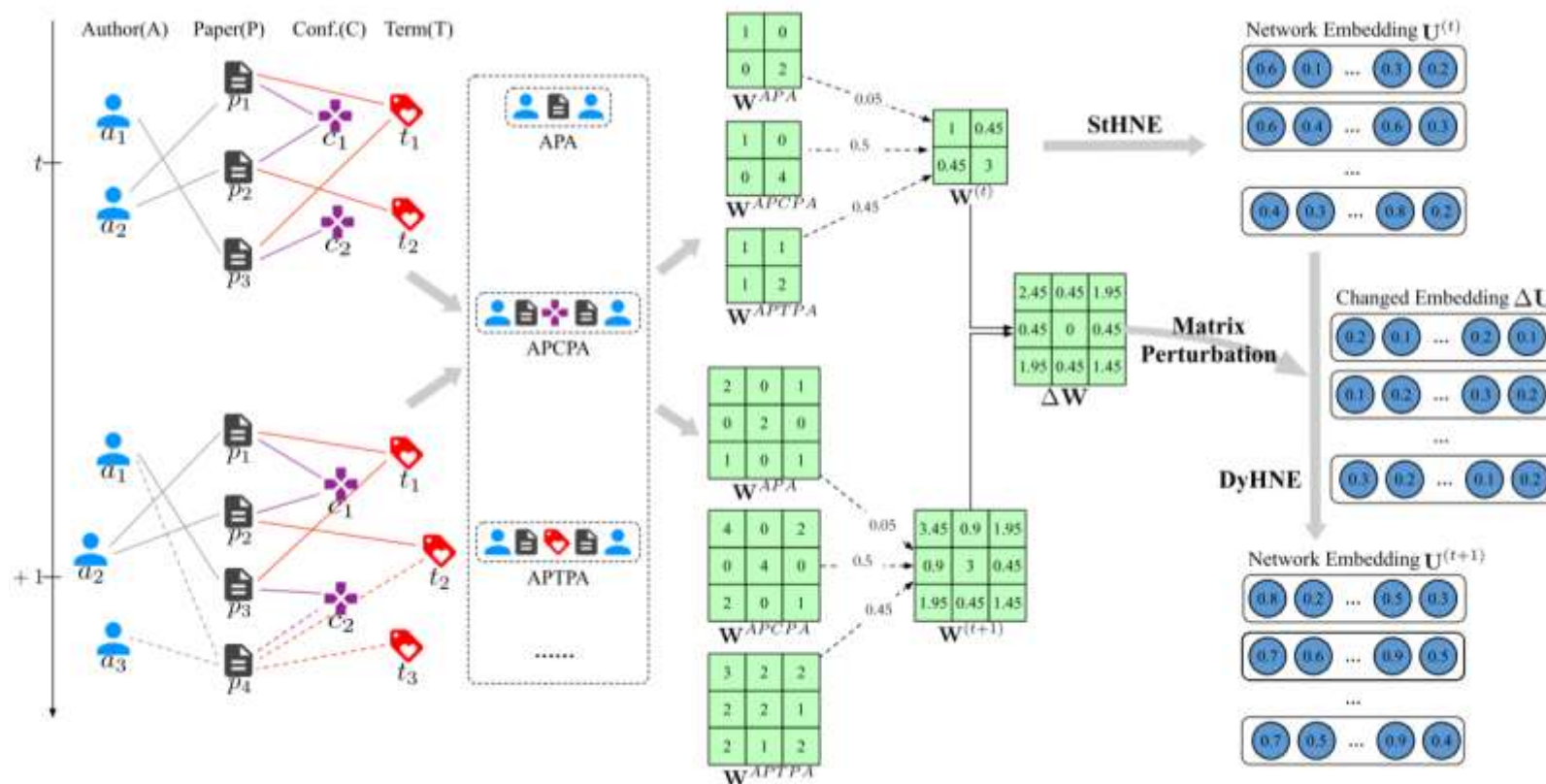
Dynamic Heterogeneous Information Network Embedding With Meta-Path Based Proximity

T	目标	在 不重新训练 的情况下动态更新图数据的节点嵌入
I	输入	10个时间步 的图数据*3 DBLP (14,376篇论文、14475位作者、20个会议/期刊、8833个术语) AMiner (18181篇论文、22942位作者、22个会议/期刊、8811个术语 Yelp (1286名用户、3360个评论、2614个商家、9个星级)
P	处理	1.基于元路径的一阶、二阶邻近构建 静态嵌入 模型 2.使用 元路径增强邻接矩阵的扰动来建模 HIN结构和语义的变化 3.应用 特征值扰动理论 更新网络嵌入
O	输出	节点嵌入矩阵*3

P	问题	1.现有方法重新加入变化就需重新训练整个网络
C	条件	1.图数据需有有意义的元路径定义 2.节点标签明确
D	难点	图局部变化导致高阶邻近性变化
L	水平	IEEE Transactions on Knowledge and Data Engineering2022 (SCI一区)

算法原理图

- 静态HIN嵌入模块：基于元路径的一阶和二阶邻近
- 动态HIN嵌入模块：矩阵扰动建模图变化



融动于静

• 静态HIN嵌入模块

– 一阶邻近性

- 衡量通过元路径连接的节点相似性

$$\mathcal{L}_1^m = \sum_{v_i, v_j \in \mathcal{V}} w_{ij}^m \|\mathbf{u}_i - \mathbf{u}_j\|_2^2.$$

– 二阶邻近性

- 衡量节点邻域结构的相似性

$$\mathcal{L}_2^m = \sum_{v_p \in \mathcal{V}} \|\mathbf{u}_p - \sum_{v_q \in \mathcal{N}(v_p)^m} w_{pq}^m \mathbf{u}_q\|_2^2.$$

– 多路径融合

- 加权整合不同元路径的损失函数

$$\mathcal{L} = \sum_{m \in \mathcal{M}} \theta_m (\mathcal{L}_1^m + \gamma \mathcal{L}_2^m)$$

- γ 是权衡因子,控制两种邻近性的相对重要性

– 数学转换

- 损失函数转为矩阵形式
- 通过特征值求解

$$\mathcal{L}_1^m = \sum_{v_i, v_j \in \mathcal{V}} w_{ij}^m \|\mathbf{u}_i - \mathbf{u}_j\|_2^2 = 2\text{tr}(\mathbf{U}^\top \mathbf{L}^m \mathbf{U}),$$

$$(\mathbf{L} + \gamma \mathbf{H}) \mathbf{U} = \mathbf{D} \mathbf{\Lambda} \mathbf{U},$$

前d个最小非零特征值对应的特征向量

• 动态HIN嵌入模块

– 在 $t = 0$ 时获取初始静态嵌入 $U(0)$

– 变化检测：当时间步 $t \rightarrow t + 1$:

• 邻接矩阵 $W \rightarrow W + \Delta W$

– 特征值求解问题的数学演变

• $(L + \gamma H)U = D\Lambda U \rightarrow (L + \Delta L + \gamma H + \gamma \Delta H)(U + \Delta U) = (D + \Delta D)(\Lambda + \Delta \Lambda)(U + \Delta U)$

• 目标：求出 ΔU ，即嵌入向量的变化量

– 更新推导步骤

• 先更新特征值

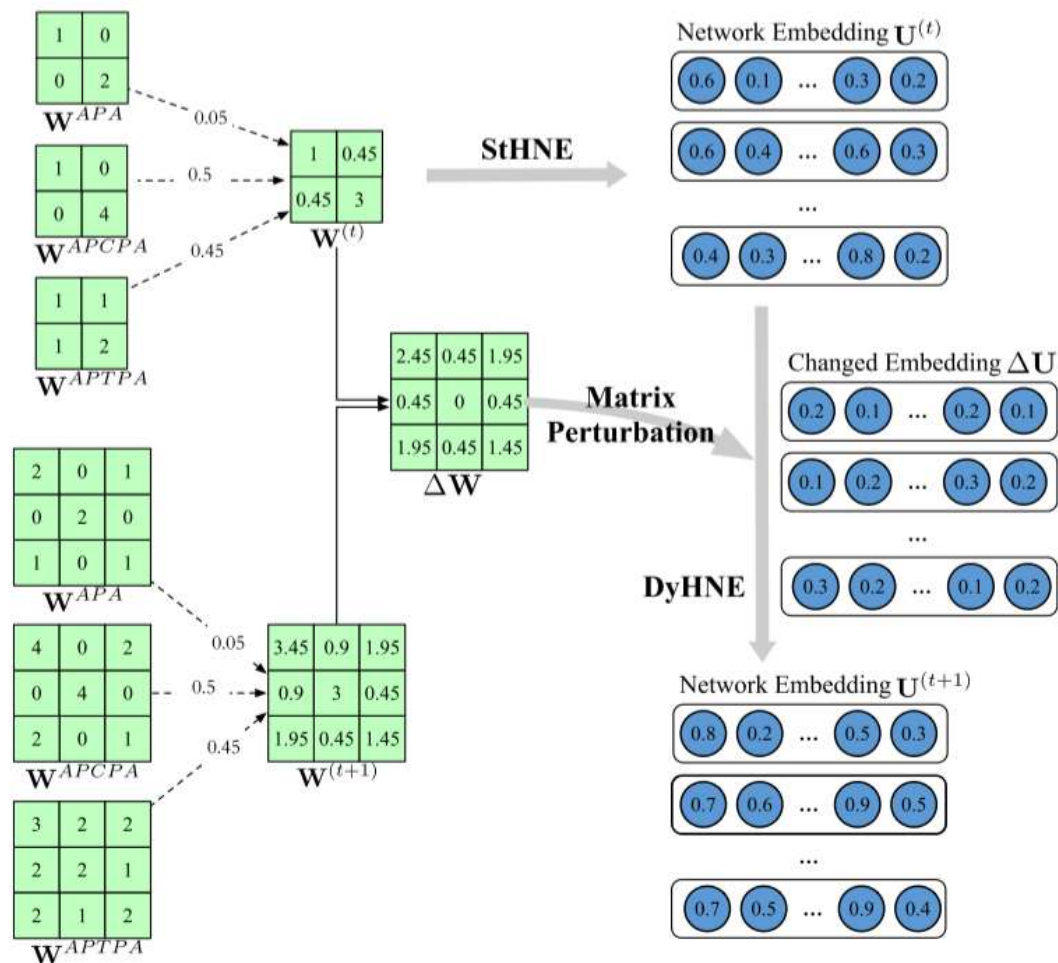
$$\Delta \lambda_i = u_i^\top \Delta L u_i + \gamma u_i^\top \Delta H u_i - \lambda_i u_i^\top \Delta D u_i$$

• 再更新特征向量

$$\Delta u_i = \sum_{j \neq i} a_{ij} u_j \quad a_{ij} = \frac{u_j^\top \Delta L u_i + \gamma u_j^\top \Delta H u_i - \lambda_i u_j^\top \Delta D u_i}{\lambda_i - \lambda_j}$$

动态HIN嵌入模块流程总结

- 在 $t = 0$ 时获取初始静态嵌入 U^0
 - 图发生变化得到 ΔW
 - 根据特征值扰动理论计算 $\Delta L, \Delta H, \Delta D$
 - 更新特征值、更新特征向量
 - 更新嵌入
 - $U^{t+1} = U^t + \Delta U$
- ## 优点
- 不需要重复训练
 - 适用于**真实场景**中大规模异构图的增量学习





• 数据集

数据集	节点类型1	节点类型2	节点类型3	节点类型4	元路径	时间步
DBLP	论文*14376	作者*14475	期刊/会议*20	术语*8833	APA、APCPA、APTPA	10
Aminer	论文*18181	作者*22942	期刊/会议*22	术语*8811	APA、APCPA、APTPA	10
Yelp	用户*1286	商家*2614	评论*3360	星级*9	BSB、BRURB	10

• 对比方法

- 静态同质网络方法：DeepWalk, LINE
- 静态异质网络方法：metapath2Vec、ESim
- 动态同质网络方法：DANE、DHPE
- 动态异质网络方法：DHNE

• 评价指标

- 节点分类：Macro-F1、Micro-F1

- 实验结果【静态嵌入模块对比】
 - 静态嵌入模型显著优于所有基线模型
 - StHNE-1st 与 StHNE-2nd 分别只保留一阶和二阶信息

Datasets	Metric	Tr.Ratio	DeepWalk	LINE-1st	LINE-1st	ESim	metapath2vec	StHNE-1st	StHNE-2nd	StHNE
Yelp	Macro-F1	40%	0.6021	0.5389	0.5438	0.6387	0.5872	0.6193	0.5377	0.6421
		60%	0.5954	0.5865	0.5558	0.6464	0.6081	0.6639	0.5691	0.6644
		80%	0.6101	0.6012	0.6068	0.6793	0.6374	0.6909	0.5783	0.6922
	Micro-F1	40%	0.6520	0.6054	0.6105	0.6896	0.6427	0.6838	0.6118	0.6902
		60%	0.6472	0.6510	0.6233	0.7011	0.6681	0.7103	0.6309	0.7017
		80%	0.6673	0.6615	0.6367	0.7186	0.6875	0.7232	0.6367	0.7326
DBLP	Macro-F1	40%	0.9295	0.9271	0.9172	0.9354	0.9213	0.9392	0.9283	0.9473
		60%	0.9355	0.9298	0.9252	0.9362	0.9311	0.9436	0.9374	0.9503
		80%	0.9368	0.9273	0.9301	0.9451	0.9432	0.9511	0.9443	0.9611
	Micro-F1	40%	0.9331	0.9310	0.9219	0.9394	0.9228	0.9421	0.9312	0.9503
		60%	0.9383	0.9328	0.9291	0.9406	0.9305	0.9487	0.9389	0.9519
		80%	0.9392	0.9323	0.9347	0.9502	0.9484	0.9543	0.9496	0.9643
AMiner	Macro-F1	40%	0.8838	0.8929	0.8972	0.9449	0.9487	0.9389	0.9309	0.9452
		60%	0.8846	0.8909	0.8967	0.9482	0.9490	0.9401	0.9354	0.9499
		80%	0.8853	0.8947	0.8962	0.9491	0.9493	0.9412	0.9381	0.9521
	Micro-F1	40%	0.8879	0.8925	0.8958	0.9465	0.9469	0.9407	0.9412	0.9467
		60%	0.8881	0.8936	0.8960	0.9482	0.9497	0.9423	0.9431	0.9509
		80%	0.8882	0.8960	0.8962	0.9500	0.9511	0.9448	0.9423	0.9529

- 实验结果【动态嵌入模块对比】
 - 动态嵌入模型显著优于所有基线模型
 - StHNE-1st 与 StHNE-2nd 分别只保留一阶和二阶信息

Datasets	Metric	Tr.Ratio	DeepWalk	LINE-1st	LINE-1st	ESim	metapath2vec	StHNE	DANE	DHPE	DHNE	DyHNE
Yelp	Macro-F1	40%	0.5840	0.5623	0.5248	0.6463	0.5765	0.6118	0.6102	0.5412	0.6293	0.6459
		60%	0.5962	0.5863	0.5392	0.6642	0.6192	0.6644	0.6342	0.5546	0.6342	0.6641
		80%	0.6044	0.6001	0.6030	0.6744	0.6285	0.6882	0.6471	0.5616	0.6529	0.6893
	Micro-F1	40%	0.6443	0.6214	0.5901	0.6932	0.6457	0.6826	0.6894	0.5823	0.6689	0.6933
		60%	0.6558	0.6338	0.5435	0.6941	0.6656	0.7074	0.6921	0.5981	0.6794	0.6998
		80%	0.6634	0.6424	0.6297	0.7104	0.6722	0.7281	0.6959	0.6034	0.6931	0.7298
DBLP	Macro-F1	40%	0.9269	0.9266	0.9147	0.9372	0.9162	0.9395	0.8862	0.8893	0.9302	0.9434
		60%	0.9297	0.9283	0.9141	0.9369	0.9253	0.9461	0.8956	0.8946	0.9351	0.9476
		80%	0.9322	0.9291	0.9217	0.9376	0.9302	0.9502	0.9051	0.9087	0.9423	0.9581
	Micro-F1	40%	0.9375	0.9310	0.9198	0.9383	0.9254	0.9438	0.8883	0.8847	0.9352	0.9467
		60%	0.9346	0.9245	0.9192	0.9404	0.9281	0.9496	0.8879	0.8931	0.9404	0.9505
		80%	0.9371	0.9297	0.9261	0.9415	0.9354	0.9543	0.9071	0.9041	0.9489	0.9617
AMiner	Macro-F1	40%	0.8197	0.8219	0.8282	0.8797	0.8673	0.8628	0.7642	0.7694	0.8903	0.9014
		60%	0.8221	0.8218	0.8323	0.8807	0.8734	0.8651	0.7704	0.7735	0.9011	0.9131
		80%	0.8235	0.8238	0.8351	0.8821	0.8754	0.8778	0.7793	0.7851	0.9183	0.9212
	Micro-F1	40%	0.8157	0.8189	0.8323	0.8729	0.8652	0.8563	0.7698	0.7633	0.8992	0.9117
		60%	0.8175	0.8182	0.8361	0.8734	0.8693	0.8574	0.7723	0.7698	0.9045	0.9178
		80%	0.8191	0.8201	0.8298	0.8751	0.8725	0.8728	0.7857	0.7704	0.9132	0.9203

- 算法贡献

- 首次提出**增量学习**的动态异构网络嵌入框架
- 引入二阶邻近性矩阵建模结构与语义信息
- 结合广义特征值分解与矩阵扰动理论，实现高效动态更新

- 算法不足

- 元路径选择仍依赖先验知识或人工经验
- 仅支持边变化的动态建模，节点类型变化处理能力有限：假设节点集合不变，将新增节点**视为度为0的孤立节点**
- 未融合节点属性或外部特征信息





特点总结与未来展望



- 特点总结

算法	DyHATR	DHINE
优势	1. 端到端训练避免分阶段训练的信息损失 2. 不同快照之间的参数共享，可以处理大批量序列	1. 适用于真实场景中大规模异构图的增量学习 2. 不需要重复训练
劣势	1. 对时间快照的间隔鲁棒性未知 2. 长时序依赖处理不足	节点类型变化处理能力有限

- 未来展望

- 融合节点属性的变化
- 高效处理节点与边类型变动
- 针对突发、非平稳的变化的建模



- [1] Xue H, Yang L, Jiang W, et al. Modeling dynamic heterogeneous network for link prediction using hierarchical attention with temporal rnn[C]//Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part I. Springer International Publishing, 2021: 282-298.**
- [2] Wang X, Lu Y, Shi C, et al. Dynamic heterogeneous information network embedding with meta-path based proximity[J]. IEEE Transactions on Knowledge and Data Engineering, 2020, 34(3): 1117-1132.**

知人者智，自知者明。胜人者有力，自胜者强。知足者富。强行者有志。不失其所者久。死而不亡者，寿。

谢谢！

