

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



动态网络嵌入方法研究

硕士研究生 李新帅

导师：潘丽敏

2021年06月14日

- 背景简介
- 基本概念
- 算法原理
- 应用总结
- 参考文献

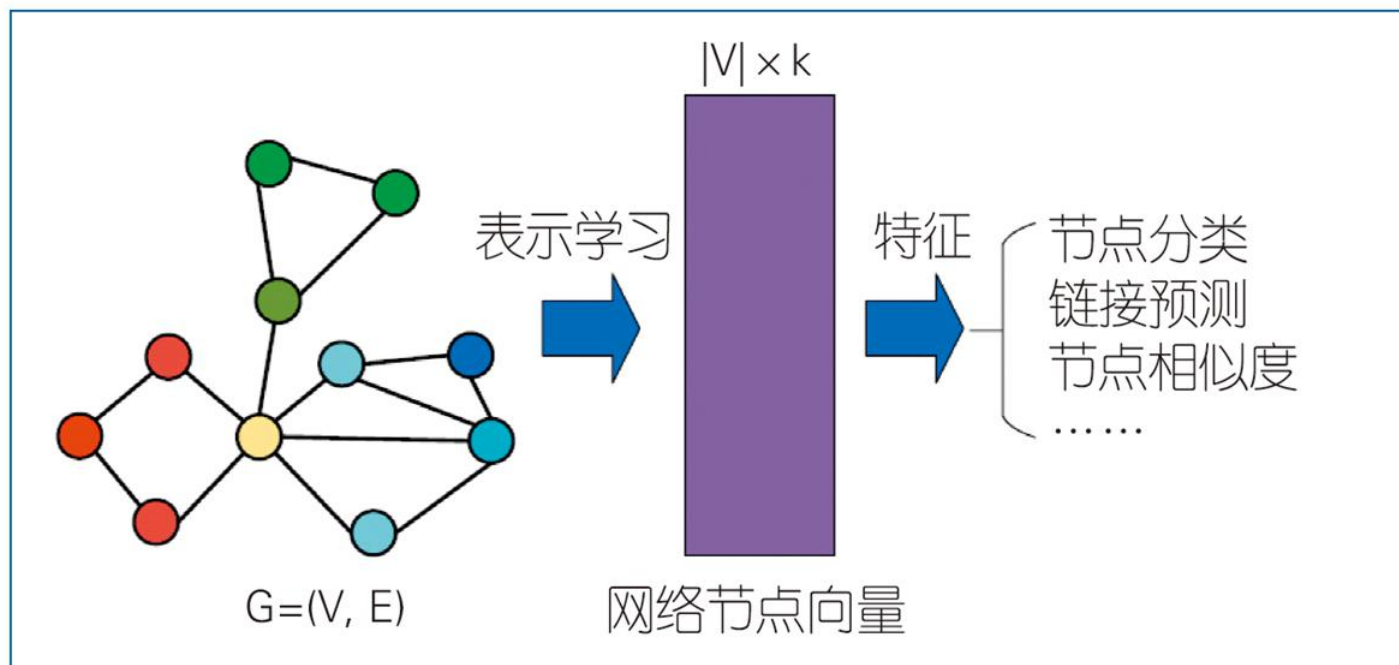
- 预期收获
 - 1.了解网络嵌入基本思想
 - 2.了解网络嵌入的研究历史及现状
 - 3.了解两种动态网络嵌入的算法及应用

背景介绍

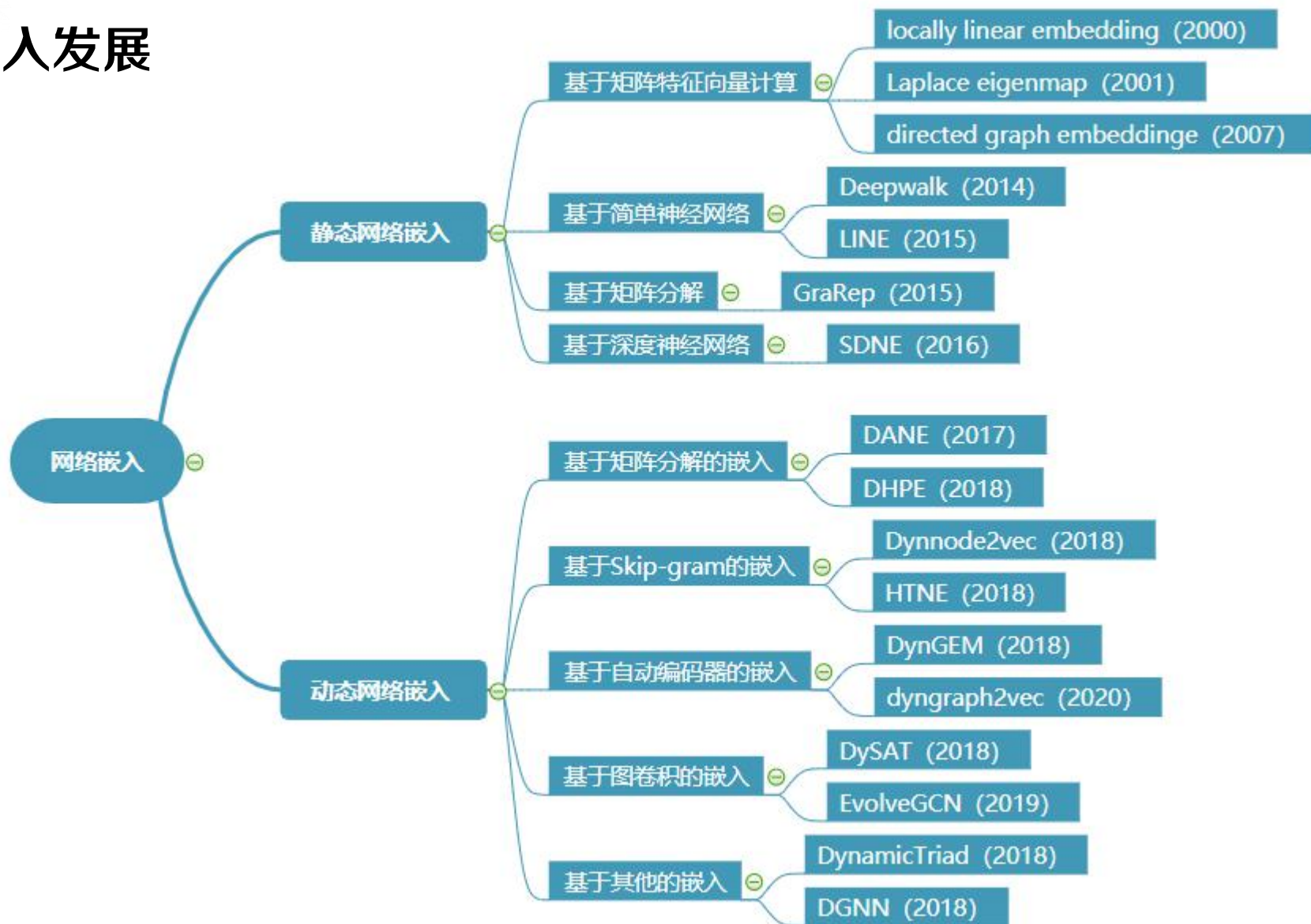


背景介绍

- 网络嵌入 (Network Embedding)
 - 网络表示学习 (Network Representation Learning)
 - 图嵌入 (Graph Embedding)



• 网络嵌入发展



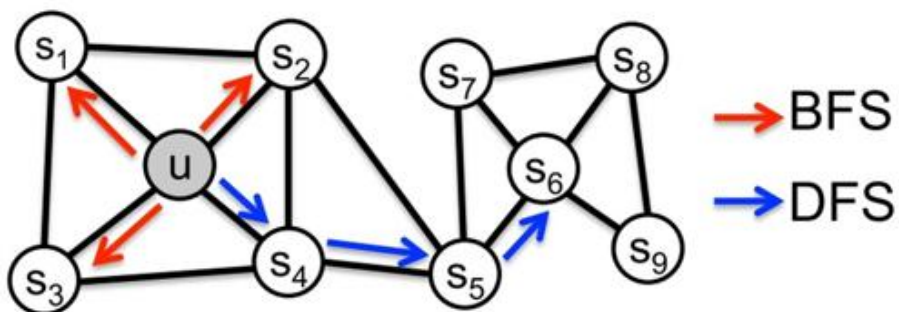
- 静态网络嵌入

- 解决问题:

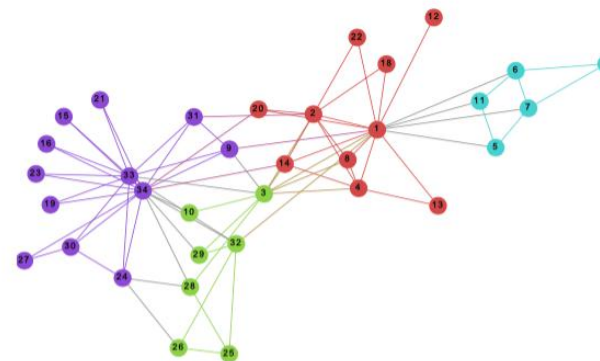
- 节省运行时间、计算空间
 - 学习网络拓扑结构

- 存在问题:

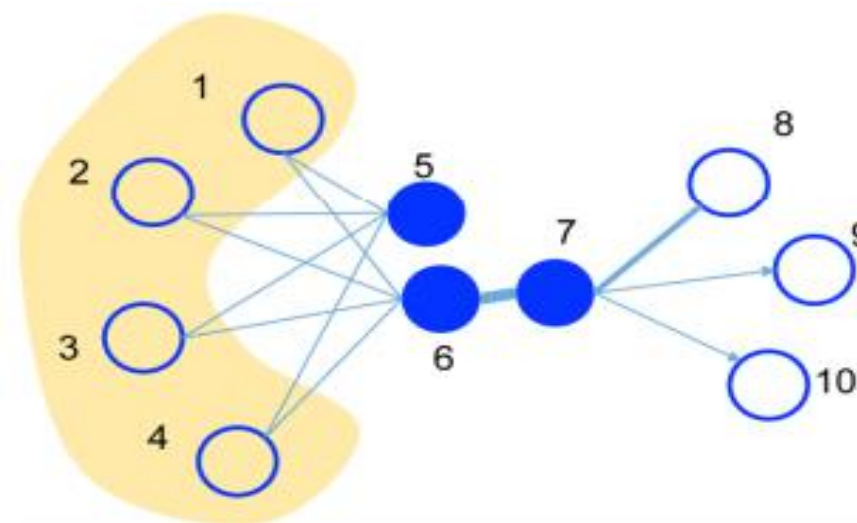
- 依赖于静态的网络拓扑结构信息
 - 忽略网络结构的动态性



Node2vec (2016)

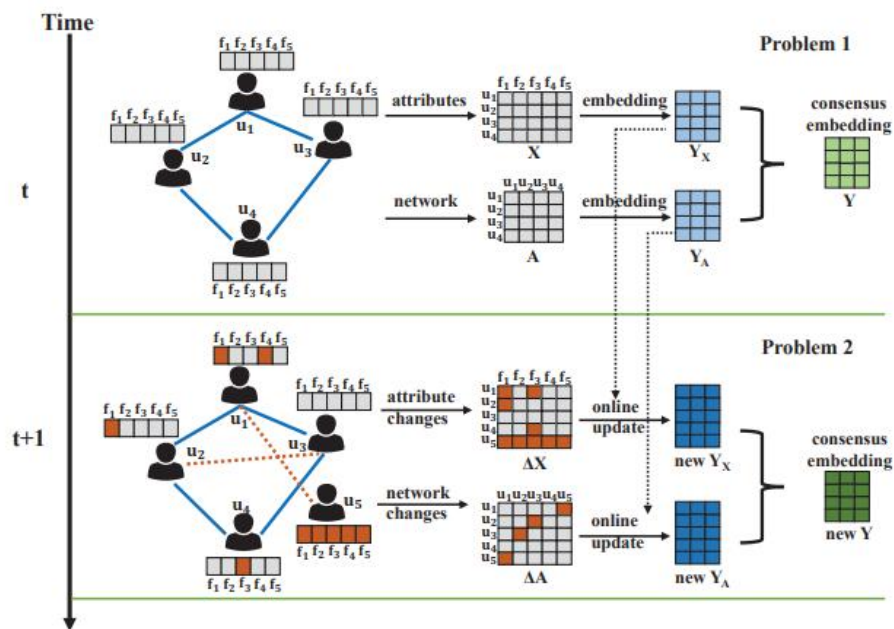


DeepWalk (2014)

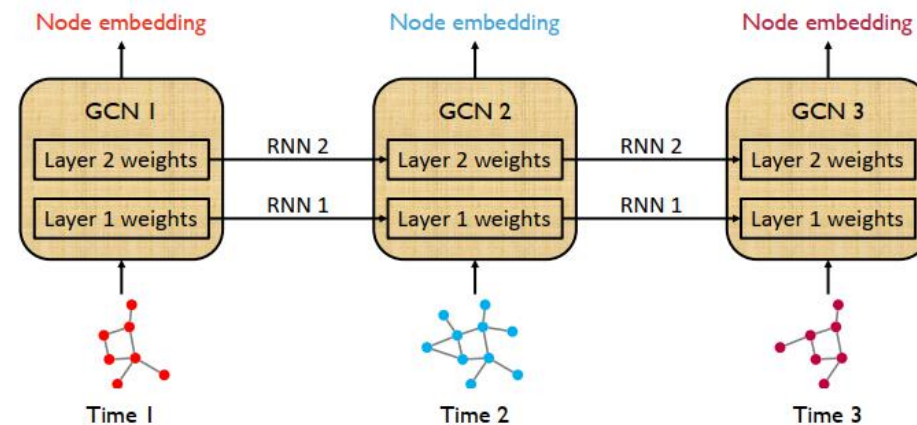
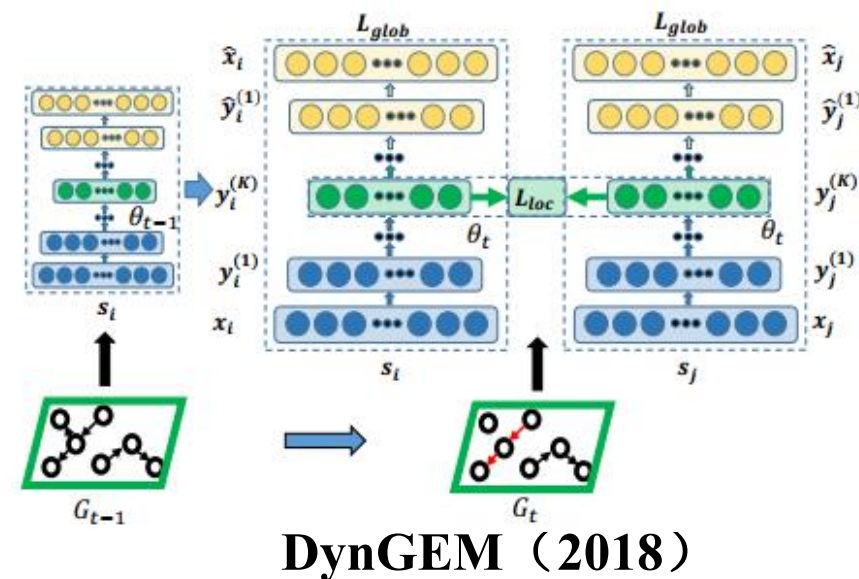


LINE (2015)

- 动态网络嵌入
 - 解决问题：
 - 捕获网络演化的时序信息



DANE (2017)



EvolveGCN (2019)

设计概念



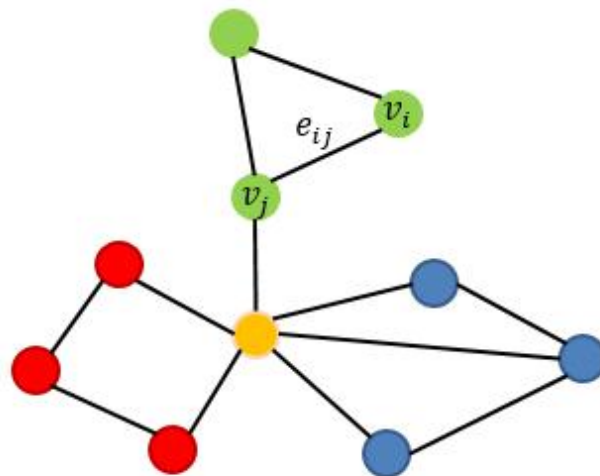
基本概念

- 网络

- 将网络记为图 $G(V, E)$ ，其中 $V = \{v_1, v_2, \dots, v_n\}$ 是节点集合， $E = \{e_{ij}\}_{i,j=1}^n$ 是边的集合， e_{ij} 表示节点 v_i 和 v_j 之间的边

- 邻接矩阵

- 图 G 的邻接矩阵定义为 $S \in \mathbb{R}^{n \times n}$ ，包含与每个边相关的非负权重： $s_{i,j} \geq 0$ 。如果 v_i 和 v_j 没有相互连接，则 $s_{i,j} = 0$



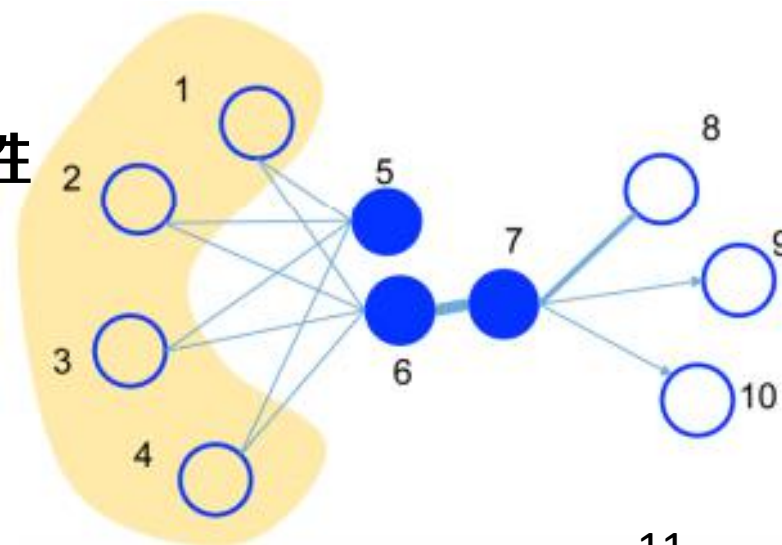
$G = (V, E)$

- 一阶相似度

- 邻接矩阵 S 中的一行 $S_i = \{s_{i,1}, s_{i,2}, \dots, s_{i,|V|}\}$ 表示 v_i 与其他顶点之间的一阶相似度
- 如果权重 $s_{i,j} > 0$ ，则 v_i 和 v_j 之间存在正的一阶相似度，权重越高，两个节点越相似
- 如果节点之间没有连接， $s_{i,j} = 0$ ，一阶相似性为0

- 二阶相似度

- 二阶相似度描述了一对节点的邻域结构的接近程度
- 节点 v_i 和 v_j 之间的二阶相似度定义为： S_i 和 S_j 之间的相似性

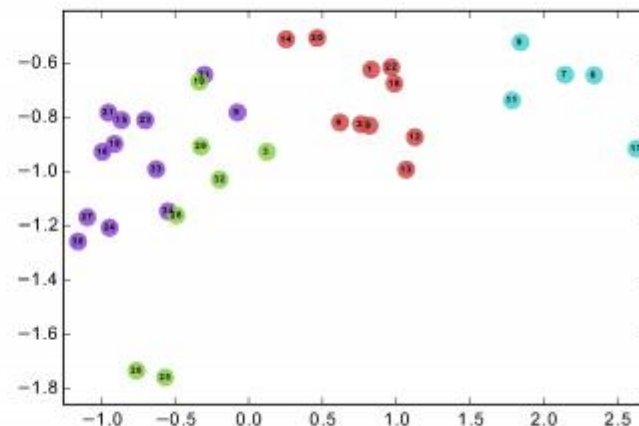


- 网络嵌入

- 给定一个表示为 $G = (V, E)$ 的图，网络嵌入旨在学习映射函数 $f: v_i \mapsto y_i \in \mathbb{R}^d$ ，其中 $d \ll |V|$ 。函数 f 的目的是使 y_i 和 y_j 之间的相似性保留 v_i 和 v_j 之间的相似度量
- 这些低维的向量表示使得快速高效的算法设计成为可能，而不必再去考虑原本的网络结构



Input



Output

- 静态网络

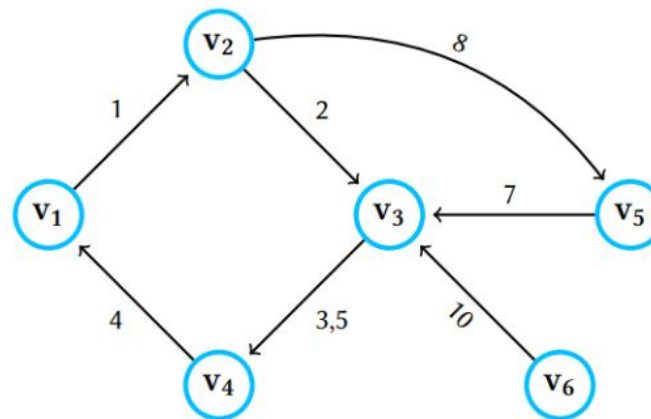
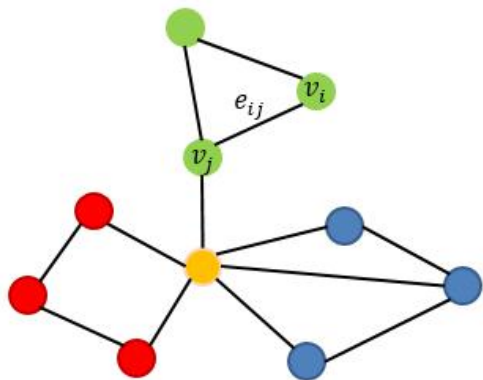
- 所有节点和边，保持一个时间常数

- 将静态网络记为图 $G(V, E)$ ，其中集合 $V = \{v_1, v_2, \dots, v_n\}$ ，集合 $E = \{e_{ij}\}_{i,j=1}^n$

- 动态网络

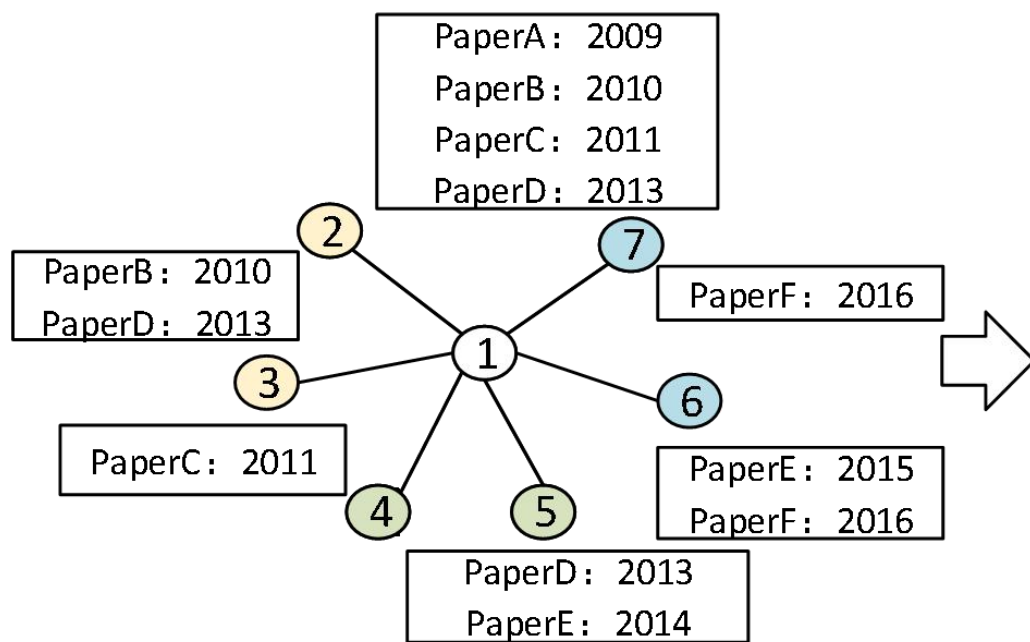
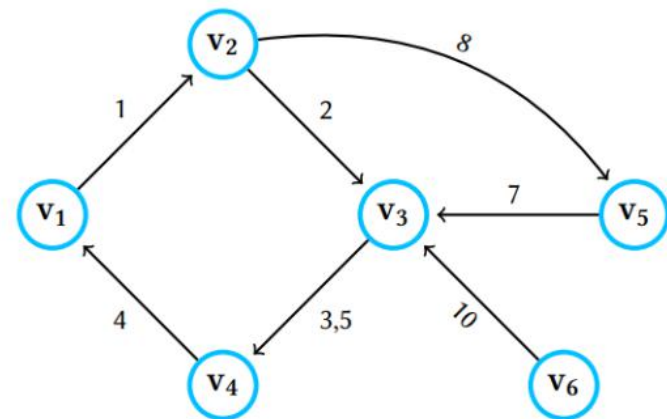
- 所有节点和边，随着时间流逝出现和消失

- 将动态网络记为图 $G(V, E)$ ，其中集合 $V = \{(v, t_s, t_e)\}$ ，集合 $E = \{(u, v, t_s, t_e)\}$

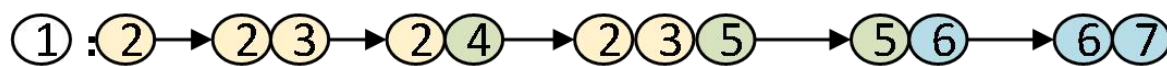


- 动态网络——连续时间

- 能够记录动态网络**完整**的时序信息
- 所有节点的集合，记录了节点出现的时间
- 每一条边都被赋予了一个特定的**时间戳**



(a) 共同作者时序网络

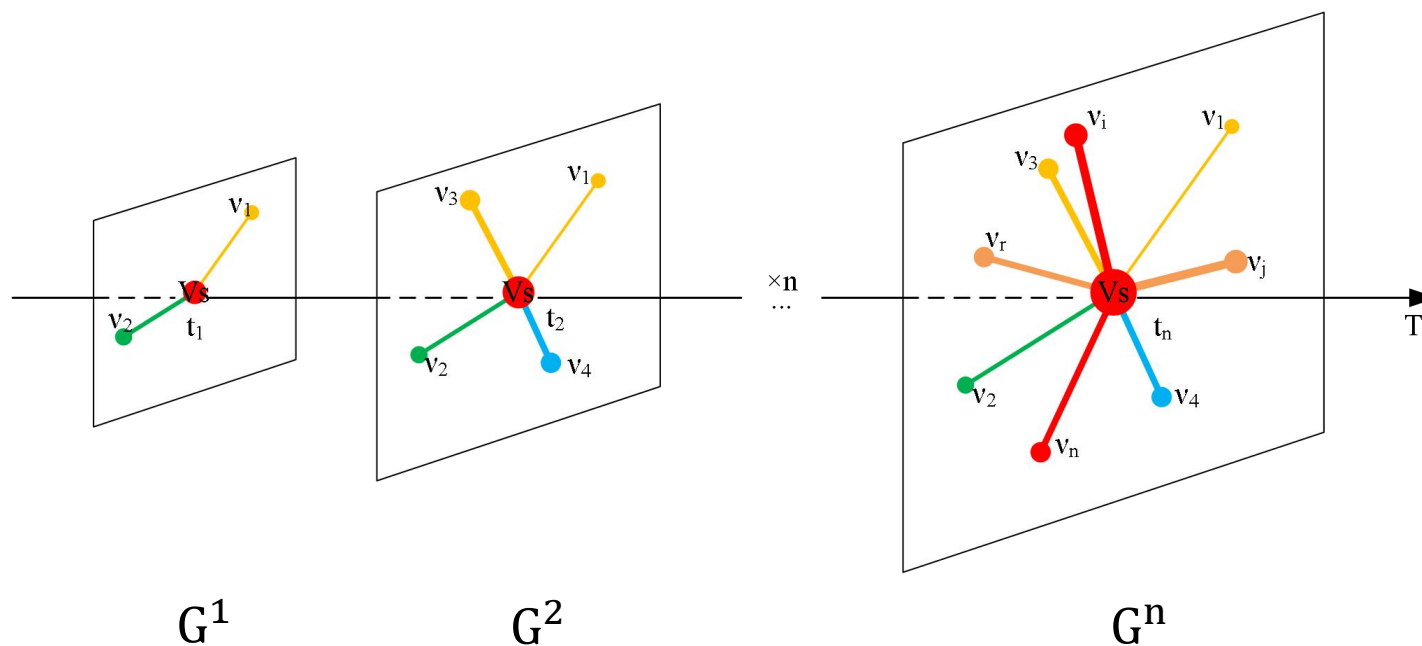
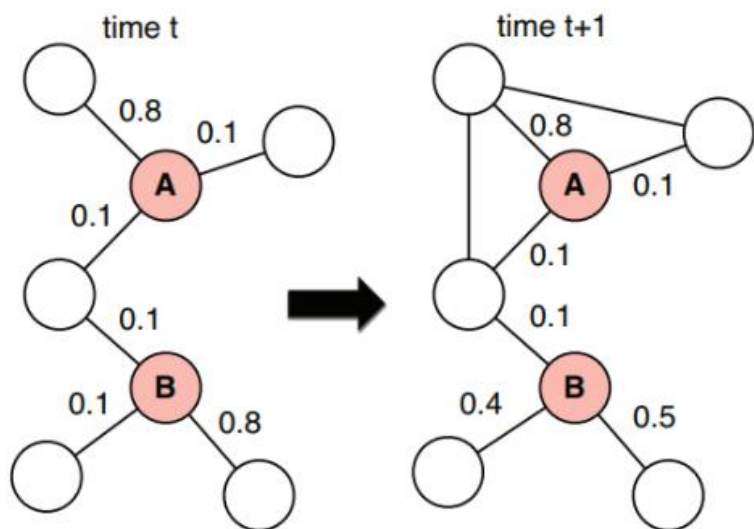


(b) 邻居生成序列

HTNE (2018)

- 动态网络——快照网络（Snapshot）
 - 按时间序列等间隔划分为快照，将动态网络分解为静态网络序列，得到网络演化的离散序列

$$\mathcal{G} = \{G^1, G^2, G^3, \dots, G^T\}$$



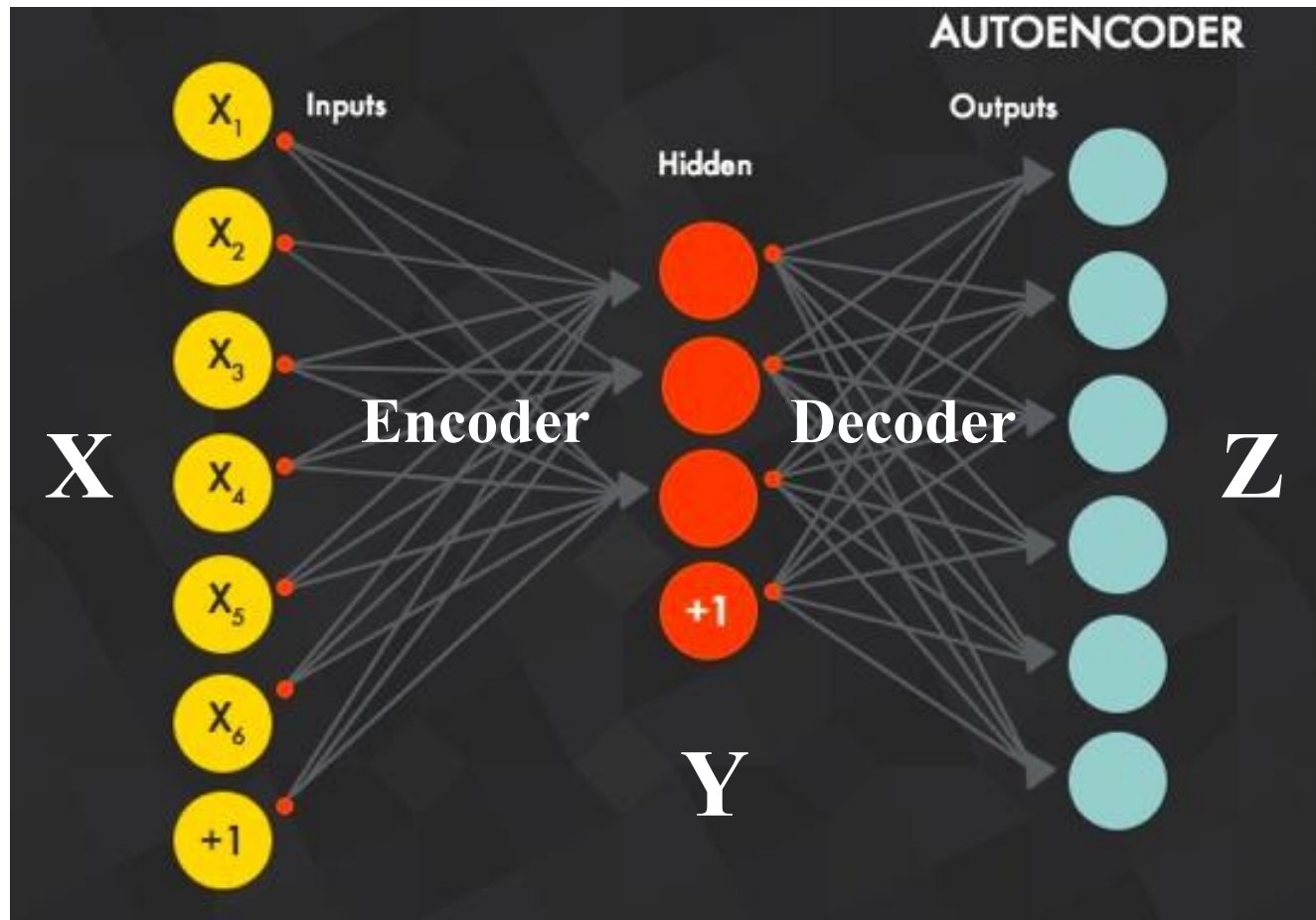
- 自编码器 (Autoencoder)
 - 无监督机器学习算法
 - 用于数据降维

Encoder:

$$y = h(x)$$

Decoder:

$$z = f(y) = f(h(x))$$



算法原理



算法原理

图神经网络

T	对网络完整的时序形成过程进行学习建模，将网络中的节点表示成低维、实值、稠密的向量形式
I	网络拓扑结构
P	1.深度自编码 2.启发式的算法-PropSize 3.参数优化
O	节点向量表示

P	尽可能利用图的拓扑结构信息、时序信息，研究网络的形成过程
C	具备网络拓扑结构、时序信息
D	1.稳定性 2.灵活性
L	IJCAI2018

- 提出问题

- 稳定性：由静态网络生成的Embedding是不稳定的
- 图增长：随时间增长，图中会有新的节点、新的关系
- 可扩展性：时间复杂度与snapshot的个数呈线性关系，模型训练效率低

- 解决办法

- 提出了一种快速有效的动态图表示学习方法——DynGEM
- 提出了一种可动态调节神经网络中神经元个数、隐层层数的方法——PropSize

- 问题定义

- 拆分为 T 个Snapshot

$$\mathcal{G} = (G_1, G_2, \dots, G_T)$$

- t 时刻的网络图

$$G_t = (V_t, E_t) \in \mathcal{G}$$

- S 邻居矩阵, s_i 邻接矩阵的第 i 行

$$s_i = (s_{i,1}, s_{i,2}, \dots, s_{i,|V|})$$

if $(u, v) \in E,$

$$s_{u,v} > 0$$

else

$$s_{u,v} = 0$$

- 稳定性计算

- 绝对稳定性

$$\mathcal{S}_{abs}(\mathcal{F}; t) = \frac{\|F_{t+1}(V_t) - F_t(V_t)\|_F}{\|S_{t+1}(V_t) - S_t(V_t)\|_F}$$

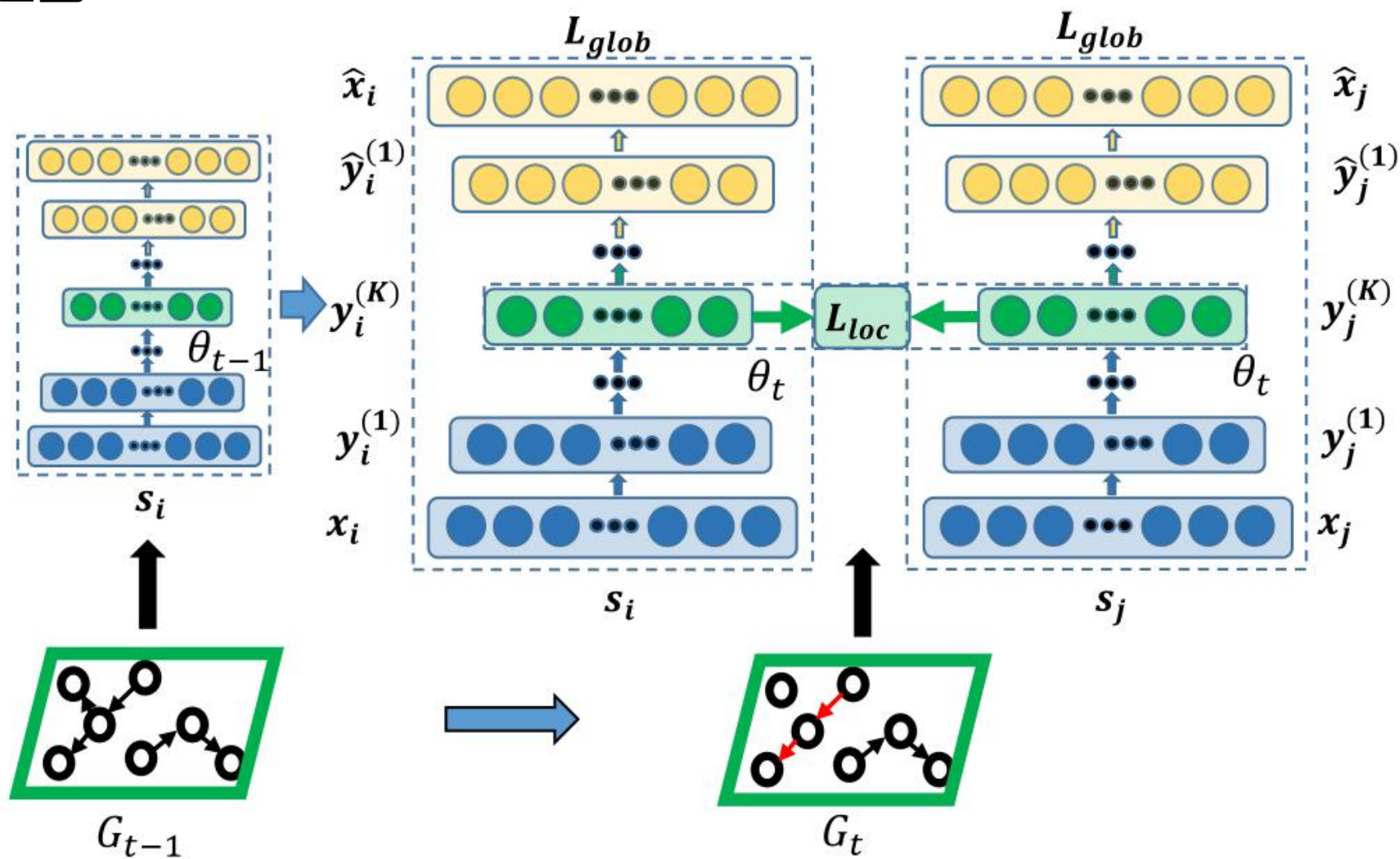
- 相对稳定性

$$\mathcal{S}_{rel}(\mathcal{F}; t) = \frac{\|F_{t+1}(V_t) - F_t(V_t)\|_F}{\|F_t(V_t)\|_F} / \frac{\|S_{t+1}(V_t) - S_t(V_t)\|_F}{\|S_t(V_t)\|_F}$$

- 稳定常量

$$K_s(\mathcal{F}) = \max_{\mathcal{T}, \mathcal{T}'} |\mathcal{S}_{rel}(\mathcal{F}; \mathcal{T}) - \mathcal{S}_{rel}(\mathcal{F}; \mathcal{T}')|$$

• 算法原理图



- 损失函数

$$L_{net} = L_{glob} + \alpha L_{loc} + v_1 L_1 + v_2 L_2$$

$$L_{glob} = \sum_{i=1}^n \|(\hat{x}_i - x_i) \odot b_i\|_2^2 = \|(\hat{X} - X) \odot B\|_F^2$$

$$\alpha L_{loc} = \alpha \sum_{i,j=1}^n s_{i,j} \|y_i - y_j\|_2^2$$

$$L_1 = \sum_{k=1}^K (\|W^{(k)}\|_1 - \|\widehat{W}^{(k)}\|_1)$$

$$L_2 = \sum_{k=1}^K (\|W^{(k)}\|_F^2 - \|\widehat{W}^{(k)}\|_F^2)$$

- 图增长

- 提出了一种动态调整神经网络模型中网络隐层层数和神经元个数的启发式方法

PropSize

$$size(l_{k+1}) \geq \rho \times size(l_k)$$

$$\rho \in (0,1)$$

- 可扩展性

- 激活函数使用ReLU
- 采用基于动量的参数优化方法Nesterov Momentum
- 使用L1和L2正则相结合

- 链接预测——MAP

	SYN	HEP-TH	AS	ENRON
GF_{align}	0.027	0.04	0.09	0.021
GF_{init}	0.024	0.042	0.08	0.017
$SDNE_{align}$	0.031	0.17	0.1	0.06
SDNE	0.034	0.1	0.09	0.081
DynGEM	0.194	0.26	0.21	0.084

- 稳定性—— $K_s(\mathcal{F})$

	SYN	HEP-TH	AS	ENRON
SDNE	0.18	14.715	6.25	19.722
$SDNE_{align}$	0.11	8.516	2.269	18.941
DynGEM	0.008	1.469	0.125	1.279

图神经网络

T	对网络完整的时序形成过程进行学习建模，将网络中的节点表示成低维、实值、稠密的向量形式
I	网络拓扑结构
P	1.深度自编码 2.LSTM算法 3.重构误差
O	节点向量表示

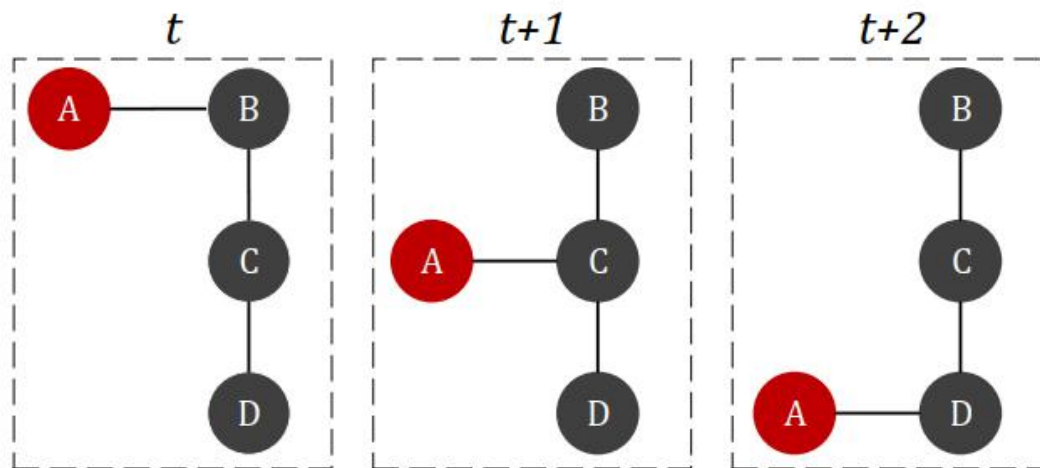
P	尽可能利用图的拓扑结构信息、时序信息，研究网络的形成过程
C	具备网络拓扑结构、时序信息
D	1.利用历史信息 2.预测不可见链路
L	2020（SCI，2区）

- 提出问题

- 时间模式可以存在于变化的周期长度上
- 不同的顶点可能具有不同的网络图

- 解决办法

- 提出了dyngraph2vec, 可以捕获时间动态
- 在网络中使用多个非线性层来学习结构模式
- 使用循环层来学习时序转变



- 损失函数

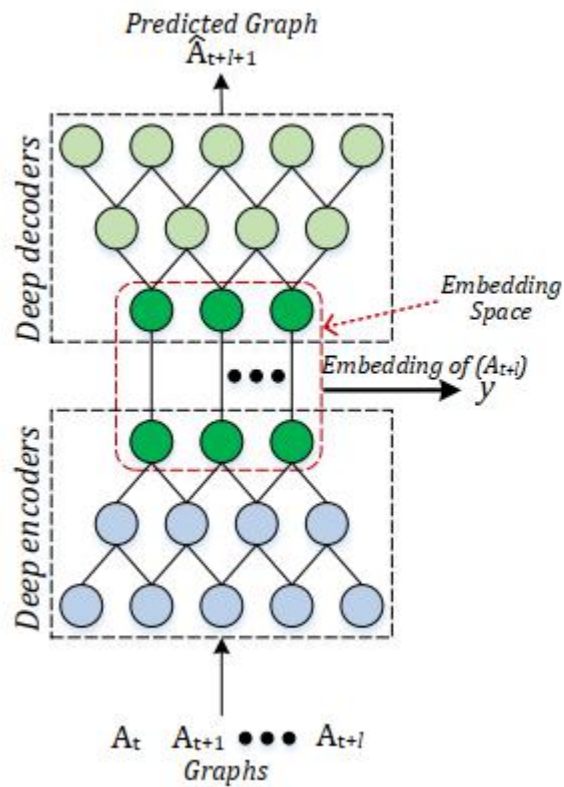
DynGEM:

$$\begin{aligned} L_{glob} &= \sum_{i=1}^n \|(\hat{x}_i - x_i) \odot b_i\|_2^2 \\ &= \|(\hat{X} - X) \odot B\|_F^2 \end{aligned}$$

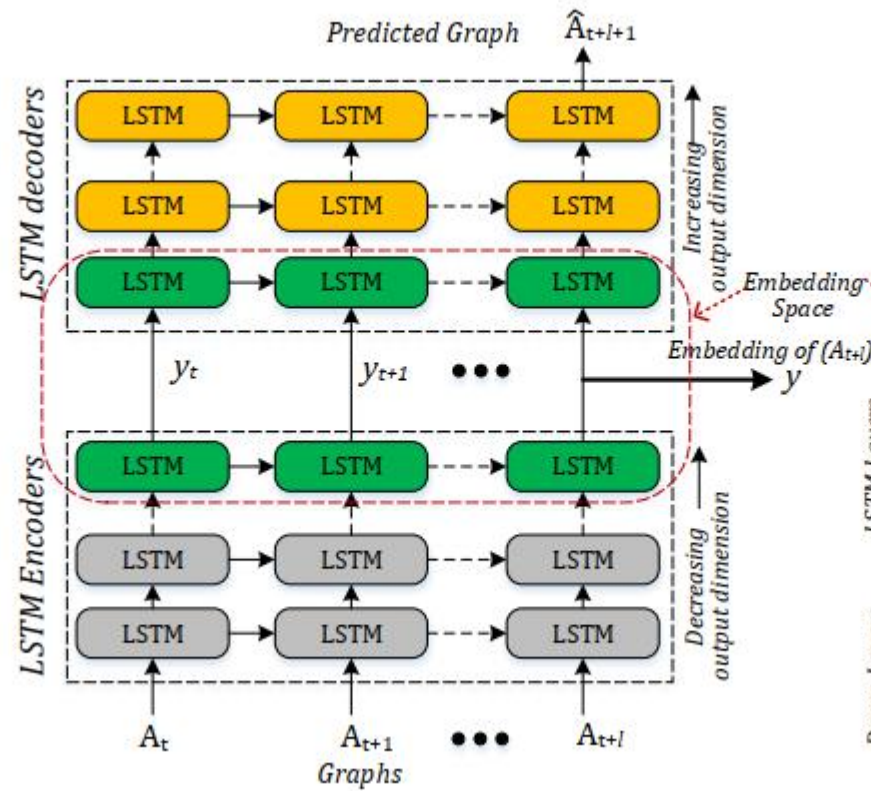
Dyngraph2vec:

$$\begin{aligned} L_{t+1} &= \sum_{i=1}^n \|(\hat{A}_{t+l+1} - A_{t+l+1}) \odot B\|_F^2 \\ &= \|(f(A_t, \dots, A_{t+l}) - A_{t+l+1}) \odot B\|_F^2 \end{aligned}$$

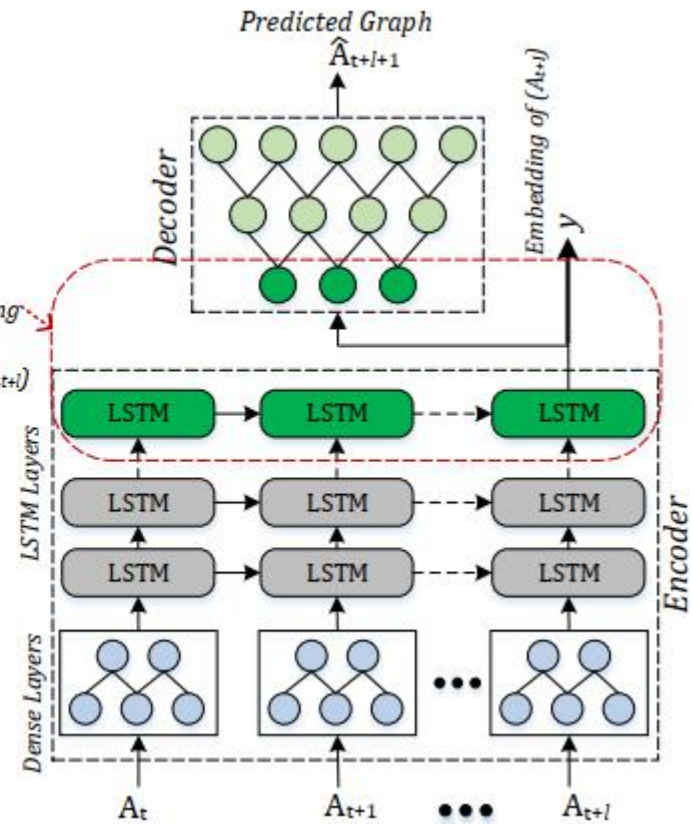
• 算法原理图



dyngraph2vecAE

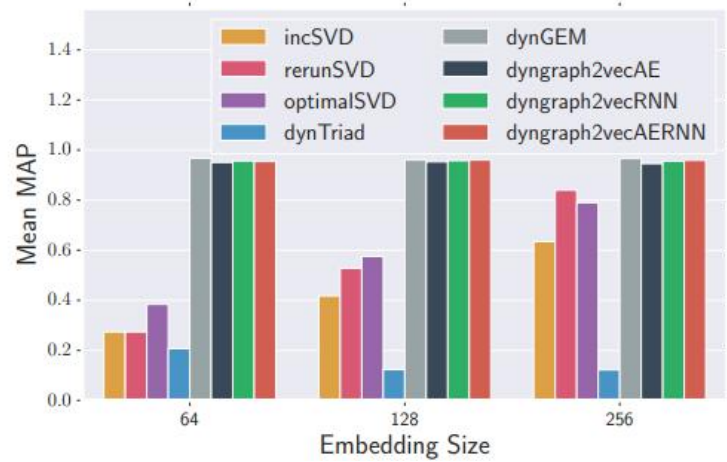


dyngraph2vecRNN

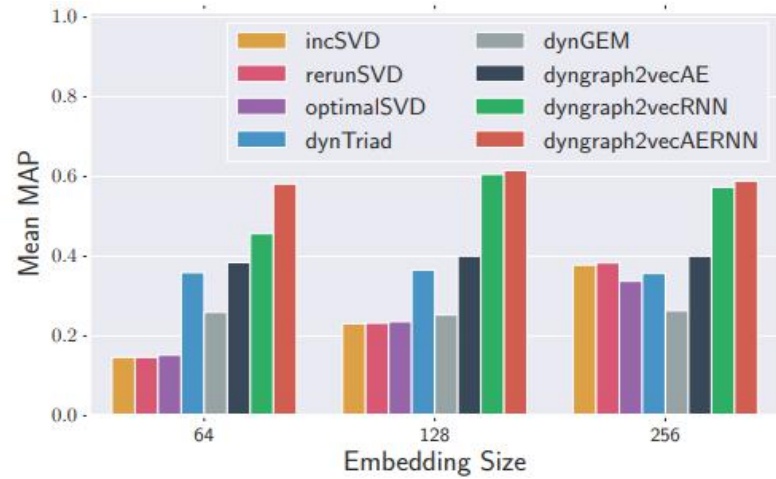


dyngraph2vecAERNN

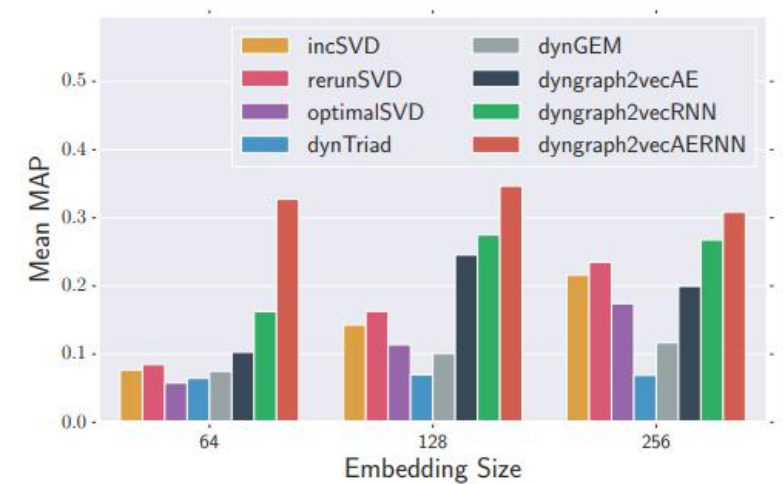
实验结果



SBM



Hep-th



AS



总结

- 应用
 - 节点分类
 - 链接预测
 - 可视化任务
 - 社区发现
- 未来发展
 - 大规模网络的网络嵌入
 - 异构网络的网络嵌入，元路径
 - 多属性网络的网络嵌入

- [1] 涂存超, 杨成, 刘知远,等. 网络表示学习综述[J]. 中国科学:信息科学, 2017(8).
- [2] Yu Xie, Chunyi Li , Bin Yu , Chen Zhang , Zhouhua Tang. A Survey on Dynamic Network Embedding[C]// CVPR 2020.
- [3] Goyal P , Kamra N , He X , et al. DynGEM: Deep Embedding Method for Dynamic Graphs[C].2018.IJCAI 2018.
- [4]Goyal, Palash, Sujit Rokka Chhetri, and Arquimedes Canedo. "dyngraph2vec: Capturing network dynamics using dynamic graph representation learning." Knowledge-Based Systems 187 (2020): 104816.

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！

