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

————一种基于skip-gram的网络嵌入扩展方法

蔡成成　硕士研究生

2020年4月6日

内容提要

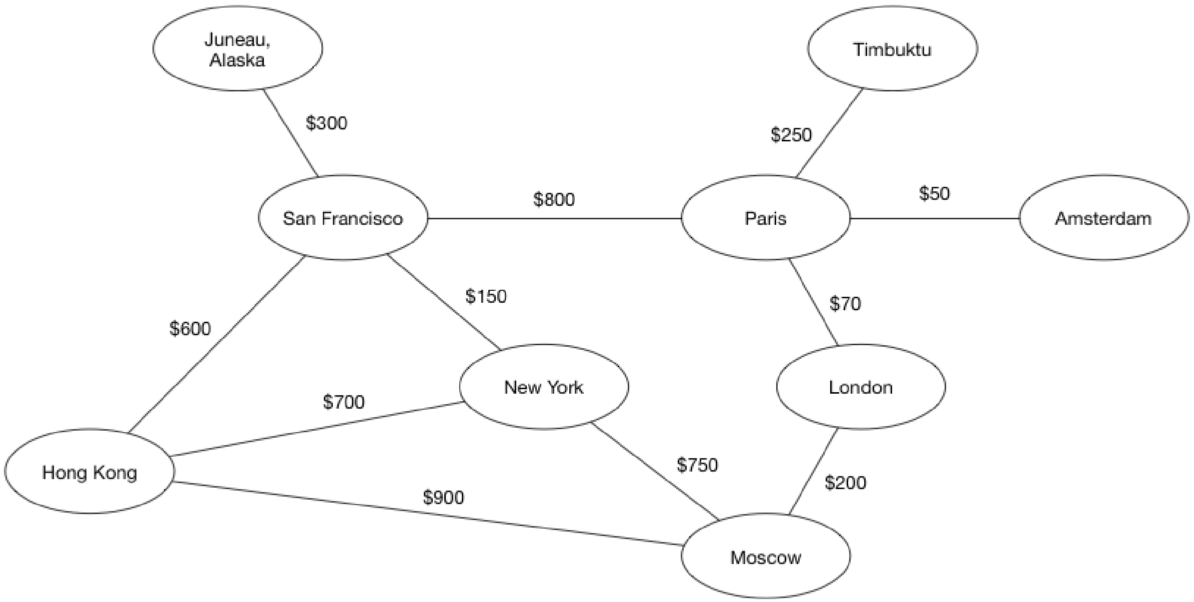
- 背景简介
- 基本概念
- 算法原理
- 优劣分析
- 应用总结
- 参考文献

背景简介

- 预期收获
  - 1. 了解网络嵌入概念及方法
  - 2.巩固skip-gram+负采样的基本思想
  - 3. 理解动态网络嵌入（DNE）的算法原理

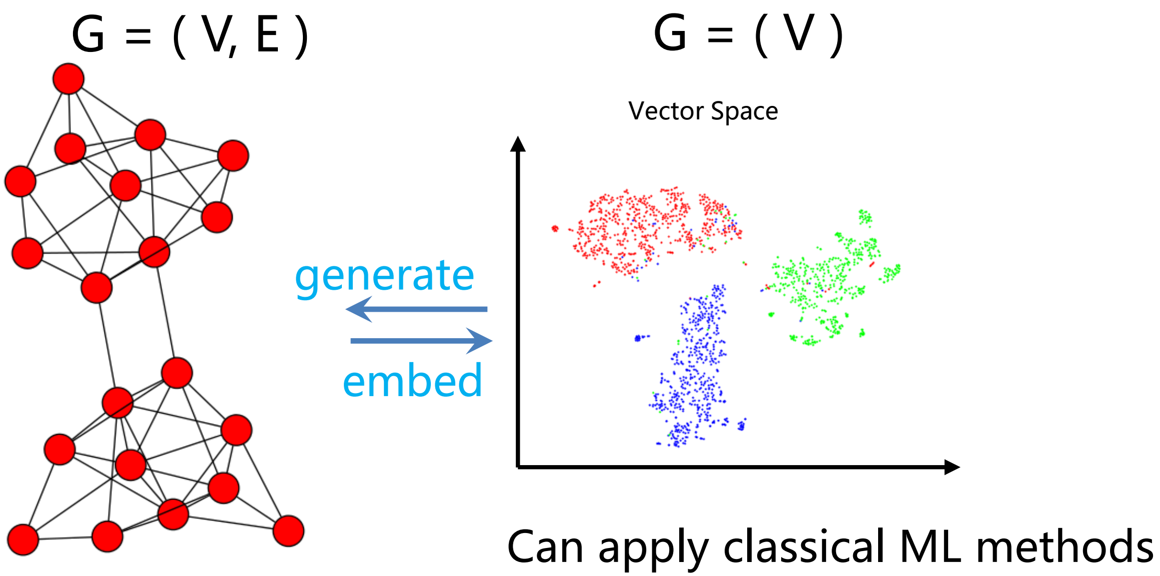
背景简介

- 图(Graph)是一个非常常用的数据结构：
  - Graph广泛存在于真实世界的多种场景中，即节点和边的集合。比如社交网络中人与人之间的联系，生物中蛋白质相互作用以及通信网络中的IP地址之间的通信等等。



背景简介

- 网络表示学习 or 网络嵌入or图嵌入
  - 旨在从网络数据中学习得到网络中每个节点的低维、实值、稠密的向量表示, 之后这些节点表示就可以作为节点的特征应用于后续的网络应用任务中，如节点分类、链接预测、社区发现、可视化任务等。



背景简介

- 为什么要使用网络嵌入or图嵌入？
  - 在图上直接进行机器学习具有一定局限性。图是由节点和边构成，这些向量关系一般只能使用数学，统计或者特定的子集进行表示，但是嵌入之后的向量空间具有更加灵活和丰富的计算方式。
  - 图嵌入能够压缩数据。我们一般用邻接矩阵描述图中节点之间的连接，将邻接矩阵用用大型图的特征空间几乎是不可能的。
  - 向量计算比直接在图上操作更加的简单、快捷。

背景简介

- 已有的网络嵌入研究方法
  - DeepWalk、Node2Vec、SDNE、LINE等。
- 动态网络嵌入概念
  - 没有任何事物是永恒不变的。许多真实世界的网络不是静态的而是处于不断进化的状态，特别是社交网络。比如一个新的用户以一个vertex的形式加入网络结构，或者两个用户建立了link，网络中就会多一个新的edge。
  - 随着网络的不断进化，一方面新节点需要被表示，另一方面，原始节点的嵌入表示就变得陈旧，需要被不断更新。

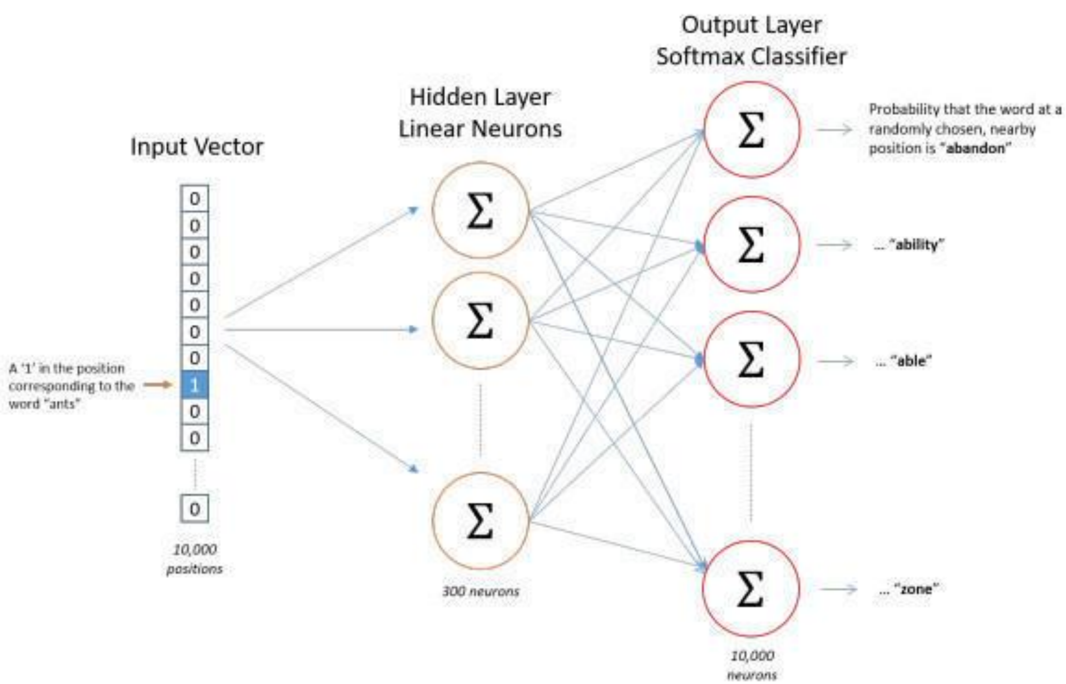
基本概念

基本概念

- Embedding
  - WHAT?
    - Embedding在数学上是一个函数，f : X —> Y，即将一个空间的点映射到另一个空间，通常是从高维抽象的空间映射到低维具象的空间。
    - 一般映射到低维空间的表示具有分布式稠密表示的特性
  - WHY?
    - 抽象的事物应该有一个低维的表示
    - 计算机和神经网络善于处理低维度信息
    - 解决one-hot编码问题
      - one-hot例子：
        - 祖国特征：["中国"，"美国，"法国"]（这里N=3）：
        - 中国 => 100；美国 => 010；法国 => 001

基本概念

- Skip-gram模型
  - 通过中心词预测上下文，该模型给出了给定中心词后，上下文中某个词出现的概率，即图中的$P(w_{t-2}|w_t)$, $P(w_{t-1}|w_t)$, $P(w_{t+1}|w_t)$, $P(w_{t+2}|w_t)$, SkipGram要做的事情就是最大化这些概率。



基本概念

- Skip-gram模型
  - 在跳字模型中，每个词被表示成d维向量，用来计算条件概率。假设这个词在词典中的索引为i，当它为中心词时向量表示为$v_i \epsilon \mathbb{R}^d$,当它为背景词时向量表示为$u_i \epsilon \mathbb{R}^d$，假设中心词$w_o$在词典中索引为o，给定中心词生成背景词的条件概率可以通过对向量内积然后做softmax运算而得到：

$$P(w_o \mid w_c) = \frac{\exp(\boldsymbol{u}_o^\top \boldsymbol{v}_c)}{\sum_{i \in \mathcal{V}} \exp(\boldsymbol{u}_i^\top \boldsymbol{v}_c)}$$

基本概念

- Skip-gram模型
  - 目标函数

$$-\sum_{t=1}^{T}\sum_{-m\leq j\leq m,\ j\neq 0}\log P(w^{(t+j)}\mid w^{(t)}).$$

    - 极小化目标函数，此时已经变为一个数学优化问题，梯度下降法更新参数。即更新中心词向量v_c。

$$v_c := v_c - \alpha * \nabla P(w_o \mid w_c)$$

$$\nabla P(w_o \mid w_c) = \boldsymbol{u}_o - \sum_{j\in\mathcal{V}}\left(\frac{\exp(\boldsymbol{u}_j^\top \boldsymbol{v}_c)}{\sum_{i\in\mathcal{V}}\exp(\boldsymbol{u}_i^\top \boldsymbol{v}_c)}\right)\boldsymbol{u}_j$$

  - 模型权重数量巨大，更新困难
    - 负采样（主要介绍）
    - 层级softmax

基本概念

- Skip-gram 负采样
  - 它是用来提高训练速度并且改善所得到词向量的质量的一种方法。不同于原本每个训练样本更新所有的权重，负采样每次让一个训练样本仅仅更新一小部分的权重，这样就会降低梯度下降过程中的计算量。
  - 使用负采样时，我们将随机选择一小部分的negative words（比如选5个negative words）来更新对应的权重。也会对我们的“positive” word进行权重更新。
  - 一个单词被选作negative sample的概率跟它出现的频次有关，出现频次越高的单词越容易被选作negative words。

$$P(w_i) = \frac{f(w_i)^{3/4}}{\sum_{j=0}^{n}(f(w_j)^{3/4})}$$

基本概念

- Skip-gram 负采样
  - 优化目标函数：

$$log\sigma(v_c \cdot v_w) + \sum_{i=1}^{K} \mathbb{E}_{w_i \sim P_n(w)} [log\sigma(-v_{wi} \cdot v_w)]$$

  - 经过一番求导后：当满足下述条件时，函数l达到最值

$$w \cdot c = log\frac{d(w,c) \cdot |D|}{d(w) \cdot d(c)} - \log(k)$$

  - 其中D是训练样本，c是w的上下文， $d(w,c)$是(w,c)在D中出现的次数， $d(w)$和 $d(c)$依次类推。

- 详情参考：https://www.cnblogs.com/linhao-0204/p/9126037.html

基本概念

- LINE-SP模型
  - 该模型考虑节点的二阶关系影响，适合在有向图中使用，（对于无向图，可以通过把一次无向边复制成两个有向边，进行转换）既然是有向图，一个节点在一条边的关系中就可能作为出度点和入度点这两种角色（分别是u和t），作者就给每一个节点两个词向量，分别对应其两种不同的功能。
  - 比方说对于一条有向边（i，j）（指的是从i指向j）

$$p_1(v_j|v_i) = \frac{exp(t_j.u_i)}{\sum_{k=1}^{|v|} exp(t_k.u_i)}$$

  - 实际上从网络结构本身出发，这两个点之间的亲密程度可以按照该式衡量

$$p_2(v_j|v_i) = \frac{w_{ij}}{d_i}$$

  - d_i是节点i的出度即 $d_i = \sum_{k \in N(i)} w_{ik}$.

基本概念

- LINE-SP模型
  - 优化目标在于使分布p1和p2差异越小越好，即目标函数为：

$$O = \sum_{i \in V} \alpha_i d(p_1, p_2)$$

  - d()函数用来衡量两个分布之间的差异性，一般选用KL散度，其中α_i代表了点i的权重。
  - 运用负采样对模型进行的具体优化，则目标函数就变成如下形式：

$$\max_{\vec{u},\vec{c}} L = \sum_{(i,j)\in E} w_{ij}\left(\log \sigma(\vec{c}_j \cdot \vec{u}_i) + k \cdot \mathbb{E}_{v_n \sim P_n(v)}[\log \sigma(-\vec{c}_n \cdot \vec{u}_i)]\right),$$

  - 其中的参数K即负采样样本的个数。

动态网络嵌入（DNE）

算法原理

算法原理

T	根据网络的动态进化，将网络中每一个节点都映射到一个低维向量空间，保持网络结构。
I	输入网络G(V;E)
P	1.提出可分解的目标函数； 2.学习新节点的嵌入向量表示； 3.更新受影响较大的部分原始节点向量表示。
O	网络节点嵌入向量表示

P	将skip-gram模型思想应用于动态网络嵌入
C	基于skip-gram模型框架
D	表示新节点同时更新原始节点； 重训练困难；
L	IJCAI 2018

算法原理

- 提出一种等同于LINE-SP目标函数的可分解目标函数
  - LINE-SP模型目标函数

$$\max_{\vec{u},\vec{c}} L = \sum_{(i,j)\in E} w_{ij}(\log\sigma(\vec{c}_j\cdot\vec{u}_i) + k\cdot\mathbb{E}_{v_n\sim P_n(v)}[\log\sigma(-\vec{c}_n\cdot\vec{u}_i)]),$$

  - 其中 $\sigma(\cdot)=1/(1+e^{-x})$ 是sigmoid函数，w_{ij}是边(i,j)的权重, $\vec{u}$是节点v_i作为中心节点的向量表示， $\vec{c}$是节点作为环境节点的向量表示， $P_n(v)\propto d_v^{\alpha}$ 是负采样的噪声分布。

算法原理

- 提出一种等同于LINE-SP目标函数的可分解目标函数
  - LINE-SP的目标函数不能同时分解为$\vec{u}$和$\vec{c}$的局部目标函数，因此提出一种可分解的目标函数代替：

$$\sum_{(i,j)\in E} w_{ij}\Big(\log\sigma(\vec{c}_j\cdot\vec{u}_i)+k\big(\mu\cdot\mathbb{E}_{v_n\sim P_{in}(v)}[\log\sigma(-\vec{c}_n\cdot\vec{u}_i)] + (1-\mu)\cdot\mathbb{E}_{v_n\sim P_{out}(v)}[\log\sigma(-\vec{c}_j\cdot\vec{u}_n)]\big)\Big),$$

  - 其中，μ是0-1之间的任意实数，$P_{in}(v)\propto d_v^{(in)}$ | $P_{out}(v)\propto d_v^{(out)}$ 负采样的噪声分布，$d_i^{(in)}$是节点v_i的入度，$d_i^{(out)}$是节点v_i的出度，$d_i^{(in)}=\sum_j w_{ji}$, $d_i^{(out)}=\sum_j w_{ij}$.

算法原理

- 提出一种等同于LINE-SP目标函数的可分解目标函数
  - 当μ为1时，目标函数为：

$$\max_{\vec{u}_i} \sum_{v_j \in N_{out}(v_i)} w_{ij}\left(\log\sigma(\vec{c}_j \cdot \vec{u}_i) + k\mathbb{E}_{v_n \sim P_{in}(v)}[\log\sigma(-\vec{c}_n \vec{u}_i)]\right),$$

  - 当μ为0时，目标函数为：

$$\max_{\vec{c}_i} \sum_{v_j \in N_{in}(v_i)} w_{ji}\left(\log\sigma(\vec{c}_i \cdot \vec{u}_j) + k\mathbb{E}_{v_n \sim P_{out}(v)}[\log\sigma(-\vec{c}_i \vec{u}_n)]\right)$$

- 由带有负采样的Skip-gram的理论最优解：

$$w \cdot c = log\frac{d(w,c)\cdot|D|}{d(w)\cdot d(c)} - \log(k)$$

- 在此基础上，给出了LINE-SP的理论最优解：

$$x_{ij} = \vec{c}_j \cdot \vec{u}_i = \log(\frac{w_{ij}\cdot D}{d_i^{(out)}\cdot d_j^{(in)}}) - \log k.$$

- 结果表明，最优解是针对内积$\vec{c_j}\cdot\vec{u_i}$，而不是每个特定的顶点，从而，通过确定的$\vec{u_i}$（or $\vec{c_j}$），我们可以最优化目标只通过$\vec{c_j}$（or $\vec{u_i}$），来使$\vec{c_j}\cdot\vec{u_i}$趋向于x_{ij}。

算法原理

- 新节点的表示
  - 对任何新节点来说，他们的边可以被分为三类：

$$\Delta E_\tau^{(1)} = \{(i,j)|v_i \in \Delta V_\tau \wedge v_j \in \Delta V_\tau\}$$
$$\Delta E_\tau^{(2)} = \{(i,j)|v_i \in \Delta V_\tau \wedge v_j \notin \Delta V_\tau\}$$
$$\Delta E_\tau^{(3)} = \{(i,j)|v_i \notin \Delta V_\tau \wedge v_j \in \Delta V_\tau\}$$

  - ΔV_τ代表一个动态网络在τ时刻相对其他时刻节点集的变化。ΔE_τ代表在τ时刻相对其他时刻边集的变化。三种情况分别代表着μ取不同的值：1/2，1，0。

算法原理

- 新节点的表示
  - 因此，新节点表示的目标函数为：

$$\min_{\vec{u}^{(\tau)},\vec{c}^{(\tau)}} \ell = \ell_1 + \ell_2 + \ell_3,$$

  - 其中

$$\ell_1 = \sum_{(i,j)\in\Delta E_\tau^{(1)}} -w_{ij}\left(\log\sigma(\vec{c}_j^{(\tau)}\cdot\vec{u}_i^{(\tau)}) + \frac{k}{2}\mathcal{F}_{in}(\vec{u}_i^{(\tau)}) + \frac{k}{2}\mathcal{F}_{out}(\vec{c}_j^{(\tau)})\right),$$

$$\ell_2 = \sum_{(i,j)\in\Delta E_\tau^{(2)}} -w_{ij}\left(\log\sigma(\vec{c}_j^{(\tau-1)}\cdot\vec{u}_i^{(\tau)}) + k\cdot\mathcal{F}_{in}(\vec{u}_i^{(\tau)})\right),$$

$$\ell_3 = \sum_{(i,j)\in\Delta E_\tau^{(3)}} -w_{ij}\left(\log\sigma(\vec{c}_j^{(\tau)}\cdot\vec{u}_i^{(\tau-1)}) + k\cdot\mathcal{F}_{out}(\vec{c}_j^{(\tau)})\right),$$

$$\mathcal{F}_{in}(\vec{u}_i^{(\tau)}) = \mathbb{E}_{v_n\sim P_{in}(v)}[\log\sigma(-\vec{c}_n^{(\tau_n)}\cdot\vec{u}_i^{(\tau)})],$$

$$\mathcal{F}_{out}(\vec{c}_j^{(\tau)}) = \mathbb{E}_{v_n\sim P_{out}(v)}[\log\sigma(-\vec{c}_j^{(\tau)}\cdot\vec{u}_n^{(\tau_n)})],$$

算法原理

- 更新原始节点
  - 一般的，一个网络的动态变化，只会对一部分节点造成影响，因此为了提升效率，只需更新被造成较大影响的原始节点的嵌入表示。
  - 首先计算两个时间切片下最优解的变化（假如一个新节点v_*在时间τ加入了网络）：

$$\Delta x_{ij}^{(\tau)} = x_{ij}^{(\tau)} - x_{ij}^{(\tau-1)}$$

$$= \log \frac{D + d_*^{(out)} + d_*^{(in)}}{D} + \log \frac{d_i^{(out)}}{d_i^{(out)} + w_{i*}} + \log \frac{d_j^{(in)}}{d_j^{(in)} + w_{*j}}.$$

算法原理

- 更新原始节点
  - 然后结合边权重提出了一个标准来判断节点是否应该被更新：

$$\epsilon_i = \frac{1}{Z_i^{(\tau-1)}}\Big(\sum_{j\in N_{out}^{(\tau-1)}(v_i)} w_{ij}\big(x_{ij}^{(\tau)} - \vec{c}_j^{(\tau-1)}\cdot\vec{u}_i^{(\tau-1)}\big) + \sum_{j\in N_{in}^{(\tau-1)}(v_i)} w_{ji}\big(x_{ji}^{(\tau)} - \vec{c}_i^{(\tau-1)}\cdot\vec{u}_j^{(\tau-1)}\big)\Big),$$

  - $x_{ij}^{(\tau)}$可以由前面计算出，$Z_i^{(\tau-1)}$是归一化因子：

$$Z_i^{(\tau-1)} = \sum_{j\in N_{out}^{(\tau-1)}(v_i)} w_{ij} + \sum_{j\in N_{in}^{(\tau-1)}(v_i)} w_{ji}.$$

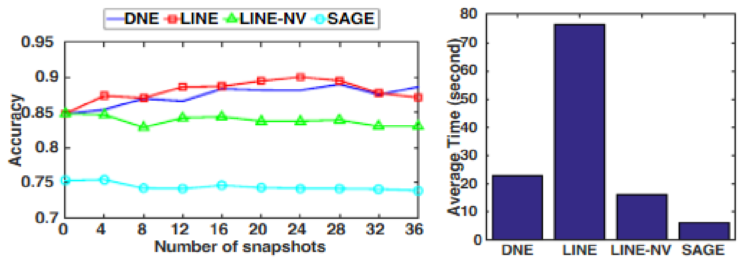
算法原理

- 更新原始节点
  - 调整m个ϵ_i值最大的原始节点进行更新，更新方法与新节点的更新方法相同（见前面公式）。

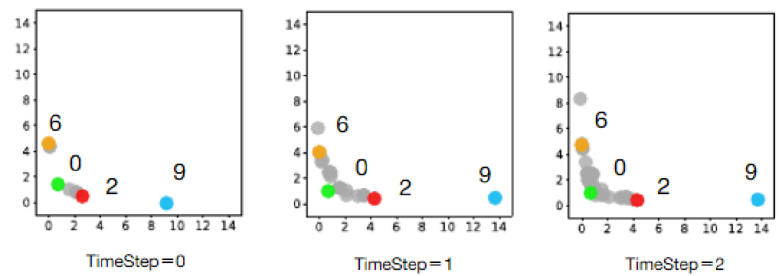
$$\min_{\vec{u}^{(\tau)},\vec{c}^{(\tau)}} \ell = \ell_1 + \ell_2 + \ell_3,$$

算法原理

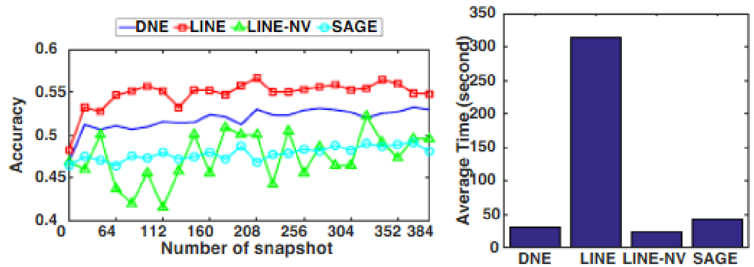
- ## 算法执行结果



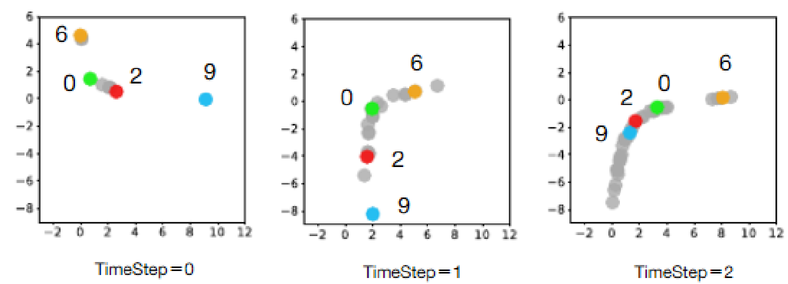
(a) The comparison of vertex classification accuracy and time cost of different models on the Amherst dataset



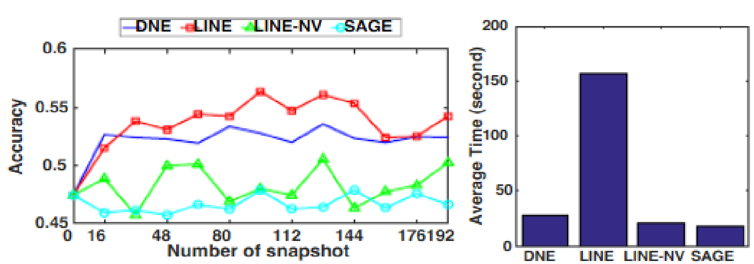
(a) Karate network layout with DNE

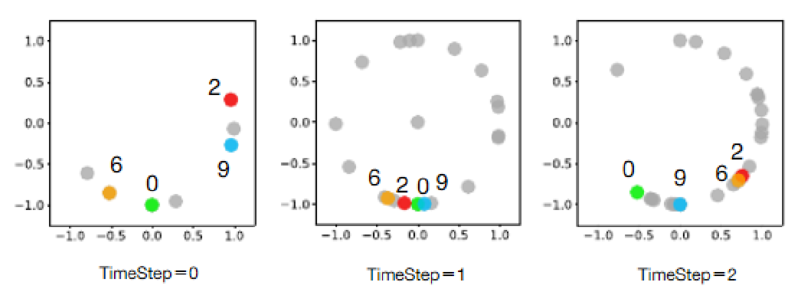


(b) The comparison of vertex classification accuracy and time cost of different models on the Duke dataset



(b) Karate network layout with LINE





(c) Karate network layout with SAGE

)

应用总结

- 框架适用性分析
  - 所有动态网络变化
    - 对于任何形式的网络变化，都会在两个时间切片τ，$\tau-1$存在G_τ，$G_{\tau-1}$两种网络结构，首先计算两个网络的理论最优解；接下来计算每个原始节点的受影响度，最后最优化目标函数，更新m个ϵ_i值最大的原始节点嵌入表示。
  - 多种基于skip-gram的网络嵌入方法
    - 其它基于skip-gram的网络嵌入方法也可以使用此框架，将其扩展到动态设置。这些方法的主要不同在于节点的邻居节点的定义。

参考文献

- [1] Lun Du, Yun Wang. Dynamic Network Embedding :An Extended Approach for Skip-gram based Network Embedding.IJCAI 2018: 2086-2092.
- [2] Omer Levy and Yoav Goldberg . Neural word embedding as implicit matrix factorization. In Advances in neural information processing systems , pages 2177–2185, 2014.

谢谢！

大成若缺，其用不弊。大盈若冲，其用不穷。大直若屈。大巧若拙。大辩若讷。静胜躁，寒胜热。清静为天下正。