

GraRep- Learning Graph Representations with Global Structural Information

Shaosheng Cao

Xidian University

简介:

工作:

- 1、提出了 skip-gram 模型的显式损失函数，将不同的 K 步关系保存在不同的子空间中，揭示了与图形有关的全局结构信息
- 2、使用矩阵分解优化每个模型，通过组合从不同模型学习的不同表示，为每个顶点构建全局表示。
- 3、整合了不同非线性的组合的损失函数

GraRep: 一种用于学习加权图的顶点表示的新模型。

Deep Walk: 不清楚他们的学习过程中所涉及的图上定义的确切损失函数是什么

LINE: 定义了 1-step 和 2-step 关系信息的损失函数，但是扩展性差

GraRep 模型:

定义 1: 邻接矩阵 S , 对角矩阵 D , 假设在邻接矩阵 S 中, 两个顶点之间的转移概率成比例。定义 1-step 的概率转移矩阵 A 为: $A = D^{-1}S$ 。可以看出矩阵 A 可以被看做是其归一化的重新缩放的 S 矩阵

定义 2: 矩阵 W , W 的第 i 行是一个 d 维的向量, 图 G 可以在这样的矢量中捕获图的全局信息

全局结构信息服务服务于两个功能:

- 1、捕获两个不同顶点之间的长距离关系;
- 2、根据不同的过渡步骤考虑不同连接的差异

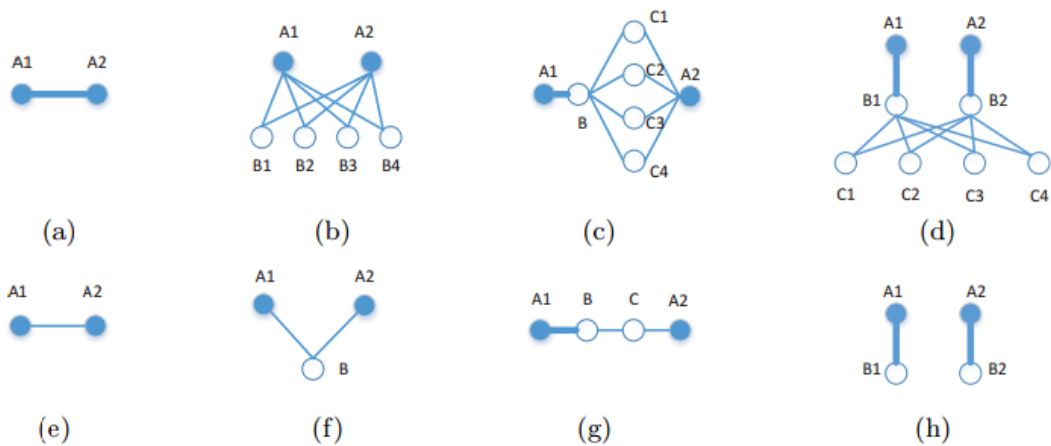
● 区分强关系和弱关系:

1-step: 强、弱

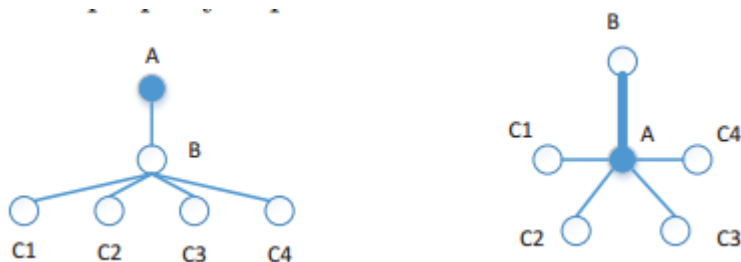
2-step: 两个顶点之间共享越多的相同邻居, 他们之间的关系越强

3-step: 两个点之间只有一个点连接和有多个点连接, 强弱关系不同

4-step: 两个点之间连接和不连接, 连接的关系要比不连接的关系强



- 区分不同 step 之间的重要性（构建一个新形式）：



- 图中的损失函数：

损失函数：两个顶点之间连接的紧密关系

1、两个顶点之间是否存在链接

2、如果两个顶点之间有链接，那么从顶点 w 到顶点 c 的可能性有多大，是否在固定数量的步骤内呢

定义：

k -step 概率转移矩阵

$$A^k = \underbrace{A \cdots A}_k$$

在 k -step 中，从顶点 i 到顶点 j 的可能性 $p_k(c|w) = A_{w,c}^k$

$A_{i,j}^k$ 表示在矩阵 A 中， w 行 c 列的值

那么首先要考虑 k ，针对一个图 G ，顶点 w 称为当前顶点， c 称为上下文顶点，目的是最大化：1、来自图中顶点对的概率 2、不来自该图中顶点对的概率

由 skip-gram(噪声对比估计)定义目标函数：

$$L_k = \sum_{w \in V} L_k(w)$$

$$L_k(w) = \left(\sum_{c \in V} p_k(c|w) \log \sigma(\vec{w} \cdot \vec{c}) \right) + \lambda \mathbb{E}_{c' \sim p_k(V)} [\log \sigma(-\vec{w} \cdot \vec{c}')]]$$

$\sigma(\cdot)$ 是 sigmoid 函数: $\sigma(x) = (1 + e^{-x})^{-1}$, λ 是指示负样本数量的超参数, $p_k(V)$ 是图中顶点分布, $\mathbb{E}_{c' \sim p_k(V)}[\cdot]$ 是指当 c 满足这个分布时的期望, c' 是负抽样获得的实例,

$$\begin{aligned} & \mathbb{E}_{c' \sim p_k(V)} [\log \sigma(-\vec{w} \cdot \vec{c}')] \\ &= p_k(c) \cdot \log \sigma(-\vec{w} \cdot \vec{c}) + \sum_{c' \in V \setminus \{c\}} p_k(c') \cdot \log \sigma(-\vec{w} \cdot \vec{c}') \end{aligned}$$

局部损失定义在特定的 (w, c)

$$L_k(w, c) = p_k(c|w) \cdot \log \sigma(\vec{w} \cdot \vec{c}) + \lambda \cdot p_k(c) \cdot \log \sigma(-\vec{w} \cdot \vec{c})$$

在这篇论文中, 设置一个最长路径, 即定义: $1 \leq k \leq K$ 事实上, 当 k 足够大时, 转移概率收敛到某些固定值, 分布 $p_k(c)$ 定义为:

$$p_k(c) = \sum_{w'} q(w') p_k(c|w') = \frac{1}{N} \sum_{w'} A_{w', c}^k$$

N 为图中顶点的个数, $q(w')$ 为选中 w' 为路径第一个顶点的可能性, 我们假设

遵循均匀分布, 则 $q(w')$ 为 $1/N$

因此:

$$L_k(w, c) = A_{w, c}^k \cdot \log \sigma(\vec{w} \cdot \vec{c}) + \frac{\lambda}{N} \sum_{w'} A_{w', c}^k \cdot \log \sigma(-\vec{w} \cdot \vec{c})$$

根据【16】, 我们定义: $e = \vec{w} \cdot \vec{c}$, 求导取 0, 则为

$$\vec{w} \cdot \vec{c} = \log \left(\frac{A_{w, c}^k}{\sum_{w'} A_{w', c}^k} \right) - \log(\beta)$$

[16] O. Levy and Y. Goldberg. Neural word embedding as implicit matrix factorization. In NIPS, pages 2177{2185, 2014.

因式分解，将 Y 分解为 W 和 C 两个矩阵，

$$Y_{i,j}^k = W_i^k \cdot C_j^k = \log \left(\frac{A_{i,j}^k}{\sum_t A_{t,j}^k} \right) - \log(\beta)$$

上式即为新定义的损失函数，同时指出损失函数的本质是矩阵分解

- 使用矩阵分解优化

用 0 代替 $Y_{i,j}^k$ 中的负样例，则变为 $X_{i,j}^k = \max(Y_{i,j}^k, 0)$

使用奇异值分解（简单、是作为降维的重要方法，且控制变量），

$$X^k = U^k \Sigma^k (V^k)^T$$

U 和 V 是正交矩阵， Σ 为包含有序奇异值的对角矩阵
对矩阵中取前 d 个奇异值，则原式变为

$$X^k \approx X_d^k = U_d^k \Sigma_d^k (V_d^k)^T$$

$$X^k \approx X_d^k = W^k C^k$$

where

$$W^k = U_d^k (\Sigma_d^k)^{\frac{1}{2}}, \quad C^k = (\Sigma_d^k)^{\frac{1}{2}} V_d^{kT}$$

W 作为图中捕获 k 步全局结构信息的顶点的 d 低维表示

算法

Table 1: Overall Algorithm

GraRep Algorithm
Input Adjacency matrix S on graph Maximum transition step K Log shifted factor β Dimension of representation vector d
1. Get k-step transition probability matrix A^k Compute $A = D^{-1}S$ Calculate A^1, A^2, \dots, A^K , respectively
2. Get each k-step representations For $k = 1$ to K <div> 2.1 Get positive log probability matrix calculate $\Gamma_1^k, \Gamma_2^k, \dots, \Gamma_N^k$ ($\Gamma_j^k = \sum_p A_{p,j}^k$) respectively calculate $\{X_{i,j}^k\}$ $X_{i,j}^k = \log \left(\frac{A_{i,j}^k}{\Gamma_j^k} \right) - \log(\beta)$ assign negative entries of X^k to 0 </div> <div> 2.2 Construct the representation vector W^k $[U^k, \Sigma^k, (V^k)^T] = SVD(X^k)$ $W^k = U_d^k (\Sigma_d^k)^{\frac{1}{2}}$ </div>

Skip-Gram model

扩展 SGNS 方法得到 E-SGNS

令偏导数为 0

$$Y_{i,j}^{E-SGNS} = \log \left(\frac{M_{i,j}}{\sum_t M_{t,j}} \right) - \log(\beta)$$

M_{ij} 是转换概率，区别 E-SGNS 和 GraRep，E-SGNS 是 K-step 损失的线性组合，每一个损失具有相同的权重；而 GraRep 具有不同的权重。

实验设计

● 数据集

1、社会化网络：Blogcatalog

每个顶点代表一个博客作者，每个边对应作者之间的关系，数据集提供了 39 中不同类型的主题类别作为标签

构建了一个不加权图。

2、语言网络： 20-Newsgroup

2 万份新闻组文件，20 个不同的组织分区，每个文档由一个带有每个单词的 tf-idf 分数的向量表示，余弦相似度计算文档相似度，根据相似度得分建立语言网络，实验中构建了 3, 6, 9 个不同新闻组的图

3-NG: *comp.graphics, comp.graphics and talk.politics.guns;*
6-NG: *alt.atheism, comp.sys.mac.hardware, rec.motorcycles,*
rec.sport.hockey, soc.religion.christian and talk.religion.misc;
9-NG: *talk.politics.mideast, talk.politics.misc, comp.os.ms-*
windows.misc, sci.crypt, sci.med, sci.space, sci.electronics,
misc.forsale, and comp.sys.ibm.pc.hardware

实验抽取了 200 个文档，构建了一个全连接，加权图，演示了使用图表示作为特征的聚类结果

3、引用网络：DBLP Network

从 DBLP 中提取作者引用网络，其中每个顶点表示一个作者，并且这两个作者之间的边缘权重记录了从一个作者到另一个作者的引用数量。

● 对比算法

1、LINE：只定义了 1-step 和 2-step 两个关系信息，通过扩展邻居使图更加密集从而改善小角度顶点性能

2、DeepWalk：是一种学习社交网络表示的方法，原始模型仅适用于未加权图，对于每个顶点，使用截断随机游走将图结构转换为线性序列。

3、E-SGNS：

4、Spectral Clustering：谱聚类是一种合理的基本算法，目标是最小归一化切割，与 Gragep 方法类似，谱聚类也将矩阵分解，但集中在图的不同矩阵，本质上，谱聚类和 E-SGNS 之间的区别在于他们不同的损失函数

● 实验结果

1、20-Newsgroup Network

归一化互信息 (NMI)：常用在聚类中，度量 2 个聚类结果的相近程度

Table 3: Results on 20-NewsGroup

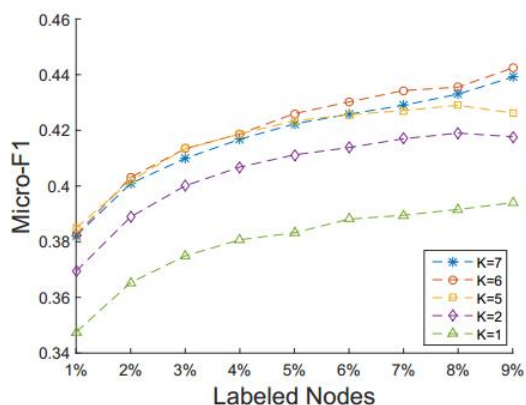
Algorithm	200 samples			all data		
	3NG(200)	6NG(200)	9NG(200)	3NG(all)	6NG(all)	9NG(all)
GraRep	81.12	67.53	59.43	81.44	71.54	60.38
LINE ($k\text{-max}=0$)	80.36	64.88	51.58	80.58	68.35	52.30
LINE ($k\text{-max}=200$)	78.69	66.06	54.14	80.68	68.83	53.53
DeepWalk	65.58	63.66	48.86	65.67	68.38	49.19
DeepWalk (192dim)	60.89	59.89	47.16	59.93	65.68	48.61
E-SGNS	69.98	65.06	48.47	69.04	67.65	50.59
E-SGNS (192dim)	63.55	64.85	48.65	66.64	66.57	49.78
Spectral Clustering	49.04	51.02	46.92	62.41	59.32	51.91
Spectral Clustering (192dim)	28.44	27.80	36.05	44.47	36.98	47.36
Spectral Clustering (16dim)	69.91	60.54	47.39	78.12	68.78	57.87

Table 4: Results on Blogcatalog

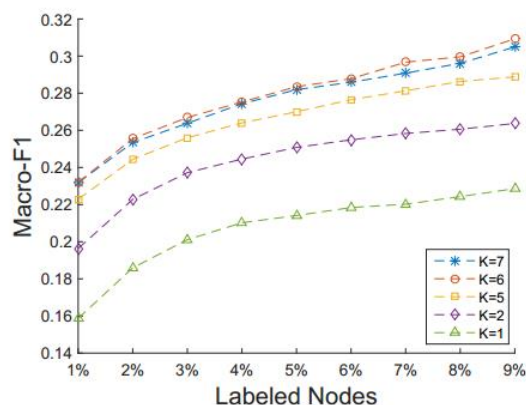
Metric	Algorithm	10%	20%	30%	40%	50%	60%	70%	80%	90%
Micro-F1	GraRep	38.24	40.31	41.34	41.87	42.60	43.02	43.43	43.55	44.24
	LINE	37.19	39.82	40.88	41.47	42.19	42.72	43.15	43.36	43.88
	DeepWalk	35.93	38.38	39.50	40.39	40.79	41.28	41.60	41.93	42.17
	E-SGNS	35.71	38.34	39.64	40.39	41.23	41.66	42.01	42.16	42.25
	Spectral Clustering	37.16	39.45	40.22	40.87	41.27	41.50	41.48	41.62	42.12
Macro-F1	GraRep	23.20	25.55	26.69	27.53	28.35	28.78	29.67	29.96	30.93
	LINE	19.63	23.04	24.52	25.70	26.65	27.26	27.94	28.68	29.38
	DeepWalk	21.02	23.81	25.39	26.27	26.85	27.36	27.67	27.96	28.41
	E-SGNS	21.01	24.09	25.61	26.59	27.64	28.08	28.33	28.34	29.26
	Spectral Clustering	19.26	22.24	23.51	24.33	24.83	25.19	25.36	25.52	26.21

增加维度并没有提高效率，因为较高的维度不能提高与表示不同的补充信息

2、K 值和维度的影响（实验结果、运行时间）

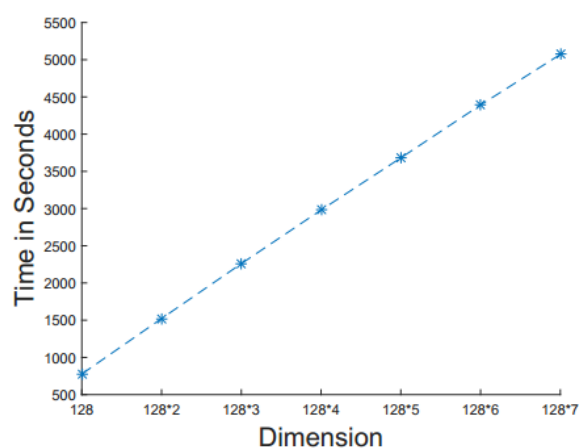


(a) Blogcatalog, Micro-F1

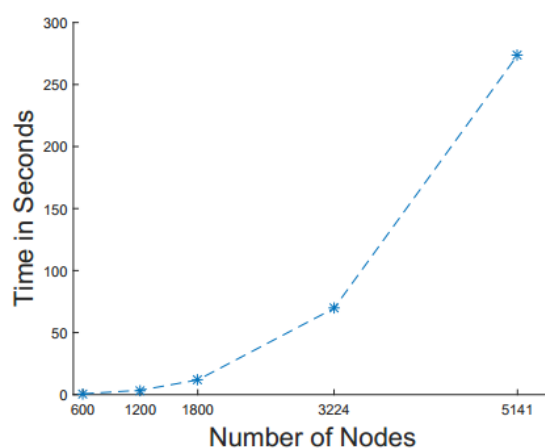


(b) Blogcatalog, Macro-F1

对实验结果的影响



(a) Blogcatalog, running time



(b) 20NG, running time

对运行时间的影响

总结

这个模型带有一个局限性：在学习过程中需要耗费较多的时间用于的矩阵和奇异值分解的计算。未来的工作将包括调查有效和在线的方法来近似矩阵代数操作，以及通过采用深层架构学习低维表示代替 SVD 来研究替代方法。