

图嵌入表示TADW：当DeepWalk加上外部文本信息

原创 kaiyuan NewBeeNLP 1周前

收录于话题

#图网络学习

9个

听说星标这个公众号👉
模型效果越来越好噢👉

大家好，这是Graph Embedding系列文章的第四篇，如果想回顾下之前的几篇，请戳👉

- DeepWalk：图网络与NLP的巧妙融合
- LINE：不得不看的大规模信息网络嵌入
- Node2Vec：万物皆可Embedding
- Graph-Bert：没有我Attention解决不了的

下面看看来自清华的

Network Representation Learning with Rich Text Information

Cheng Yang^{1,2}, Zhiyuan Liu^{1,2*}, Deli Zhao², Maosong Sun¹, Edward Y. Chang²

¹ Department of Computer Science and Technology,
State Key Lab on Intelligent Technology and Systems,
National Lab for Information Science and Technology,
Tsinghua University, Beijing 100084, China

² HTC Beijing Advanced Technology and Research Center, China

- 论文：Network Representation Learning with Rich Text Information^[1]
- 代码：<https://github.com/albertyang33/TADW>

DeepWalk的影响力可太大了，今天这篇包括之前介绍的几篇 Graph Embedding 文章都是在其基础上的优化。在之前，大多数网络表示学习研究仅仅考虑网络结构，而忽略了节点可能包含的丰富的信息，例如节点文本信息。

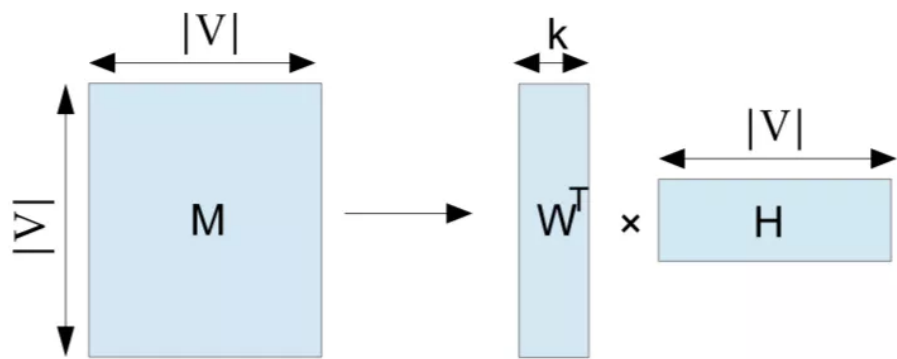
如果想结合的话，一个简单直观的做法是，分别学习 **网络结构的表示** 以及 **文本特征的表示**，然后将两者通过concat操作或者其他操作拼接在一起。但是这样丢失了两者的原始复杂交互，为此论文作者提出一种 『Text-Associated DeepWalk:TADW』 模

型，从矩阵分解的角度优化deepwalk，将丰富的节点文本信息融入进模型。下面来看看具体模型吧~

模型

DeepWalk等价于矩阵分解

论文的一个成果是证明了deepwalk可以被认为是一个矩阵分解的操作，如下图



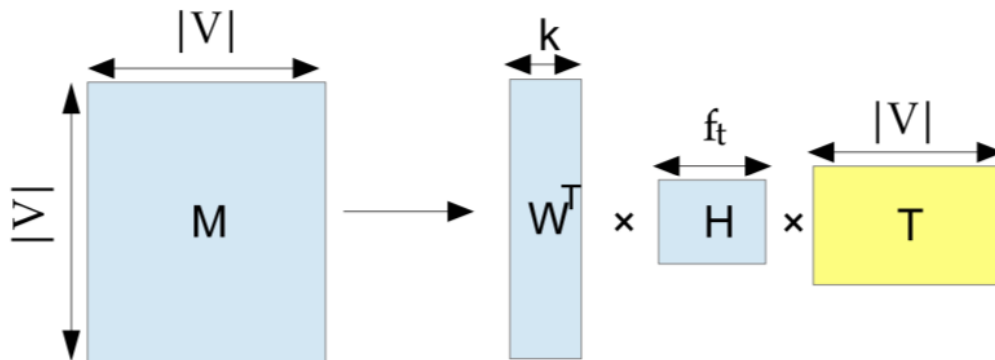
(a) DeepWalk

其中， $M \in R^{|V| \times |V|}$ 是待分解的矩阵，其中每个元素 M_{ij} 代表从节点 v_i 经过固定步长的随机游走到达节点 v_j 的平均概率对数。这里的矩阵分解其实就是和Skip-Gram一样的，感兴趣的可以看看另外一篇关于词向量矩阵分解的论文：Neural Word Embedding as Implicit Matrix Factorization^[2]

ps. 原论文中有关于这个的详细证明，自行查阅。

One more step

那么如何在矩阵分解中加入额外信息呢？那当然是继续加入一个矩阵，如下图，



(b) TADW

黄色的矩阵为新加入的文本特征矩阵。

如果是一般的矩阵分解，损失函数为，

$$\min_{W, H} \sum_{(i, j) \in \Omega} \left(M_{ij} - (W^T H)_{ij} \right)^2 + \frac{\lambda}{2} (\|W\|_F^2 + \|H\|_F^2)$$

对于TADW，可以稍微调整为，

$$\min_{W, H} \|M - W^T H T\|_F^2 + \frac{\lambda}{2} (\|W\|_F^2 + \|H\|_F^2)$$

可以发现，对于 W 和 H 矩阵，在固定其中一个的情况下，整个损失函数对于另外一个来说是凸函数，所以优化过程可以交替固定和更新两个矩阵即可。

实验

实验的话，是在几个常用的数据集上，对不同嵌入模型得出的embedding经过SVM后评判分类效果，

Table 1: Evaluation results on Cora dataset.

Classifier	Transductive SVM				SVM				
% Labeled Nodes	1%	3%	7%	10%	10%	20%	30%	40%	50%
DeepWalk	62.9	68.3	72.2	72.8	76.4	78.0	79.5	80.5	81.0
PLSA	47.7	51.9	55.2	60.7	57.0	63.1	65.1	66.6	67.6
Text Features	33.0	43.0	57.1	62.8	58.3	67.4	71.1	73.3	74.0
Naive Combination	67.4	70.6	75.1	77.4	76.5	80.4	82.3	83.3	84.1
NetPLSA	65.7	67.9	74.5	77.3	80.2	83.0	84.0	84.9	85.4
TADW	72.1	77.0	79.1	81.3	82.4	85.0	85.6	86.0	86.7

Table 2: Evaluation results on Citeseer dataset.

Classifier	Transductive SVM				SVM				
% Labeled Nodes	1%	3%	7%	10%	10%	20%	30%	40%	50%
DeepWalk	-	-	49.0	52.1	52.4	54.7	56.0	56.5	57.3
PLSA	45.2	49.2	53.1	54.6	54.1	58.3	60.9	62.1	62.6
Text Features	36.1	49.8	57.7	62.1	58.3	66.4	69.2	71.2	72.2
Naive Combination	39.0	45.7	58.9	61.0	61.0	66.7	69.1	70.8	72.0
NetPLSA	45.4	49.8	52.9	54.9	58.7	61.6	63.3	64.0	64.7
TADW	63.6	68.4	69.1	71.1	70.6	71.9	73.3	73.7	74.2

尾巴

文章虽然比较老，但是从另一个角度分析了经典算法 DeepWalk，并且应该算是第一批提出将节点文本信息融入network的工作，还是很有新意。

一起交流

想和你一起学习进步！『NewBeeNLP』目前已经建立了多个不同方向交流群（机器学习 / 深度学习 / 自然语言处理 / 搜索推荐 / 图网络 / 面试交流 / 等），名额有限，赶紧添加下方微信加入一起讨论交流吧！（注意一定要备注信息才能通过）



本文参考资料

[1] Network Representation Learning with Rich Text Information: [http](http://mp.weixin.qq.com/s/mZnwVC-PbvX7epkbE3Xdwg)