

SNS

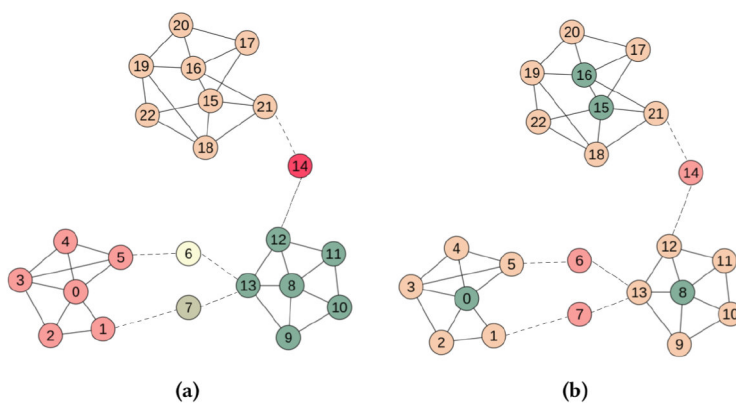
与之前讨论的几篇论文相似，本文的出发点也是考虑网络中结构相似的节点，网络表示学习多用邻居节点作为重要信息，但是仅仅通过邻居节点无法判断一个节点在某个局部的结构特性。传统的随机游走的方法自然是无法捕捉节点的结构等价性的。于是本文提出了SNS用来解决这方面的问题。具体来说，SNS同时利用了邻居信息和局部子图的相似性来学习节点嵌入。

Introduction

非监督学习的网络表示学习方法多基于保存节点邻居信息的方式来得到节点嵌入。

自从DeepWalk提出后，逐渐涌现了一些基于DeepWalk的改进算法，这些算法的改机方向是扩充邻居的定义，利用不同层次的邻居信息，例如定义的一阶相似性、二阶相似性和高阶相似性。

尽管邻居对网络表示学习十分重要，但是同样重要的还有节点在网络中的位置信息。图中展示了两种不同的方法得到的节点表示效果。一种是如果连接的越紧密，则越相似；另一种则是比较节点在网络中所起的作用，结构类似的节点是相似的。



GDV of node 0,2,8

		Orbit													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
Node ID	0	0	5	5	0	0	0	0	0	0	0	6	0	3	1
	8	0	5	5	0	0	0	0	0	0	0	5	0	5	0
	2	4	1	2	0	2	0	0	0	2	2	0	2	1	0

$$E(0,8) = 6^{0.5}, \quad E(0,2) = 98^{0.5}$$

而目前大多数方法都属于前者，作者认为这两种思想并不冲突，反而相辅相成。如果网络中的标签代表兴趣，那么显然前者的表示效果更好；如果网络中的标签代表社会角色或地位，则反之。最重要的是维持两者的平衡。

个人觉得两者是冲突的，作者举的这个例子并不兼容，如果网络中的标签代表兴趣，那么你用基于结构特性的思想得到的嵌入就不满足要求，无法由两个节点结构相似得出这两个节点是同一标签。

显然仅仅依靠随机游走是不足以保持网络节点的结构等价性的，所以本文提出了SNS，它具有以下一些贡献：

- 指出基于随机游走方法的缺陷。
- 提出同时利用邻居信息和局部子图相似性学习节点嵌入。

Related Work

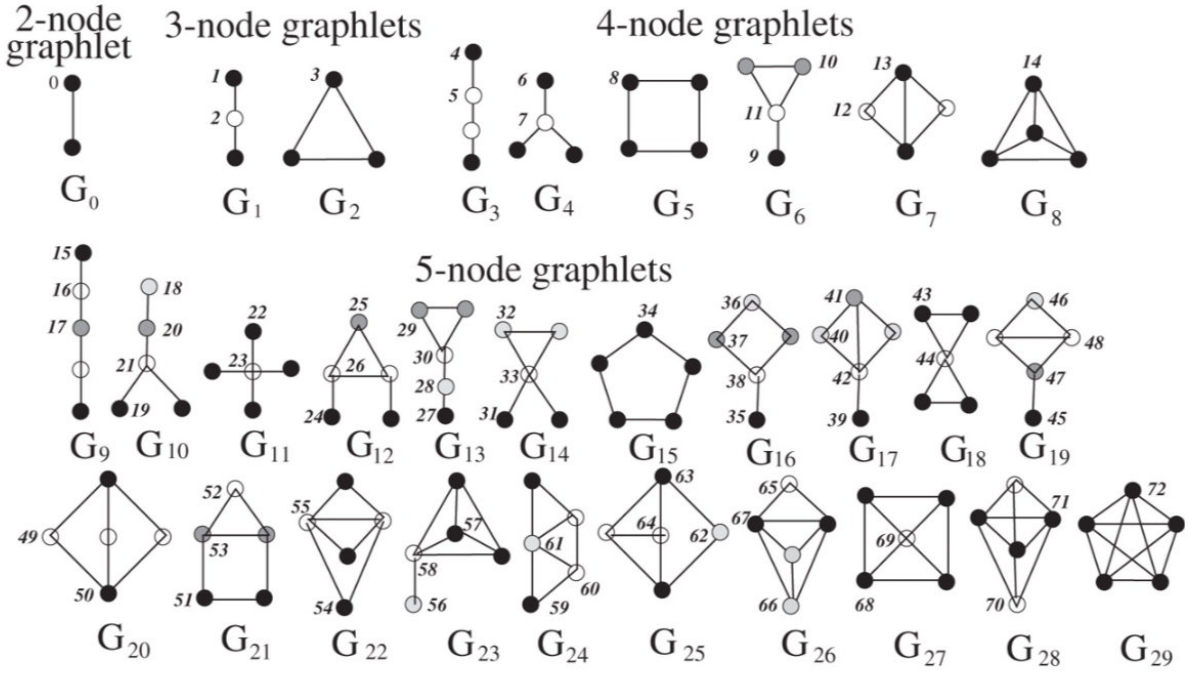
主要介绍了一些传统的方法和DeepWalk后的一系列方法。

而计算结构等价性方面，目前有一些方法包括graphlet、subtree pattern和random walk。因为我们需要一个度量来衡量两个节点之间的结构等价性，本文选择了graphlet。

graphlet的定义：

Graphlets are small, connected, non-isomorphic, induced sub-graphs of a big graph.

如图所示，图中展示了由2-5个节点构成的30个graphlet。每个数字表明了一种graphlet，对称的节点就没有重复编号了，30个graphlet中总共有73个不同的节点位置。现在用一个74维的向量来表示网络中一个节点对应着73个不同位置的情况，这个向量被称为GDV（Graphlet Degree Vector），其中向量中的每个元素表示这个节点对应73个位置的次数，另外再加上节点度数作为额外的一维，总共74维。



针对这样的graphlet，衍生出很多graphlet的计数算法，所以计数这一步的任务暂且不用担心。

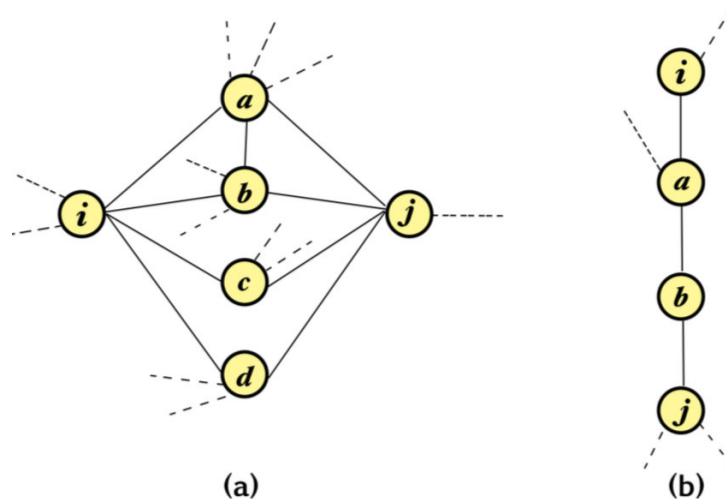
Methods

• Random-Walk Based Sampling

定义一个概率 $P(i, j, r)$ 表示节点 i 经过 r 步后到达 j 的概率。这个概率跟转移矩阵 M 有关，经过推导可得：

$$P(i, j, r) = P(j, r | i) P(i) = \frac{d_i}{2m} [M^r]_{ij} \propto [A M^{r-1}]_{ij}$$

通过上式可以发现，这个概率同两个值有关，一个是从 i 到 j 的路径上节点的度数；另一个是从 i 到 j 的路径条数。路径上节点的度数越小，路径长度越短，条数越多，则说明两个节点在随机游走中共同出现的概率越大。但基于这样的相似性度量，仍然不能反映出两个节点的结构等价性。




在随机游走中刻画两个节点的邻居特征使用的是窗口大小，而这种方法会将不在窗口内的节点认为是不相似的，因此这些节点对即使结构相似也不会被考虑。

• Preprocessing: Obtain the Structurally Similar Nodes

既然基于随机游走的算法并不能满足要求，那么寻求其他的解决方案。之前我们见识了graphlet的定义，一个需要注意的是graphlet中的编号会随着构成graphlet节点数的增加而迅速增加。这么多的特征极易造成过拟合，同时也并非每个特征都有用。于是本文利用随机森林来衡量每个特征在节点分类任务上的重要性。

随机森林是由很多个决策树构成的，在决策树中根据基尼指数或信息增益来计算根据哪个特征做分支。因此训练一棵决策树，对于之前的74个特征，计算不同特征的信息增益来衡量每个特征的重要程度。

以度特征作为基准，其他的特征与基准的倍数如图所示：



orbit	RI%	orbit	RI%	orbit	RI%	orbit	RI%	orbit	RI%
0	0.05	1	21.18	2	1.79	3	0.14	4	84.58
5	62.26	6	95.38	7	10.94	8	21.27	9	81.94
10	63.48	11	8.50	12	41.98	13	6.15	14	3.71
15	92.27	16	76.27	17	85.71	18	86.70	19	93.90
20	82.76	21	62.11	22	86.08	23	20.55	24	86.28
25	83.70	26	65.51	27	88.68	28	81.02	29	82.77
30	63.89	31	100.00	32	76.21	33	20.14	34	75.33
35	95.19	36	77.29	37	82.30	38	44.18	39	94.31
40	77.64	41	65.50	42	18.75	43	80.58	44	11.52
45	87.21	46	80.75	47	58.46	48	65.70	49	83.55
50	24.42	51	76.65	52	80.50	53	43.72	54	85.05
55	15.38	56	91.93	57	66.20	58	14.91	59	77.65
60	55.84	61	13.73	62	74.85	63	31.37	64	41.12
65	83.61	66	59.60	67	16.18	68	39.53	69	7.20
70	49.11	71	14.99	72	12.14				

从图中可以得出两个结论：

- 度这个特征没什么用。
- 度较小的位置标号比度较大的位置编号所代表的特征好很多。

一种解释是度大那些模式可以由一些度小的模型拼凑而成，所以度大的模型可能存在信息冗余。

在找与节点结构相似的节点时，往往会看节点某个邻域内的所有节点，那么具体这个邻域应该取多大，这也是文章探讨的问题之一。在"相连的节点有着相似的标签"的情况下，应该选距离目标节点较近的节点，即在一个相对较小的邻域范围内找与目标节点结构相似的节点；反之，在"结构等价性占主导地位"的网络中，就应该在一个大的邻域内找了。

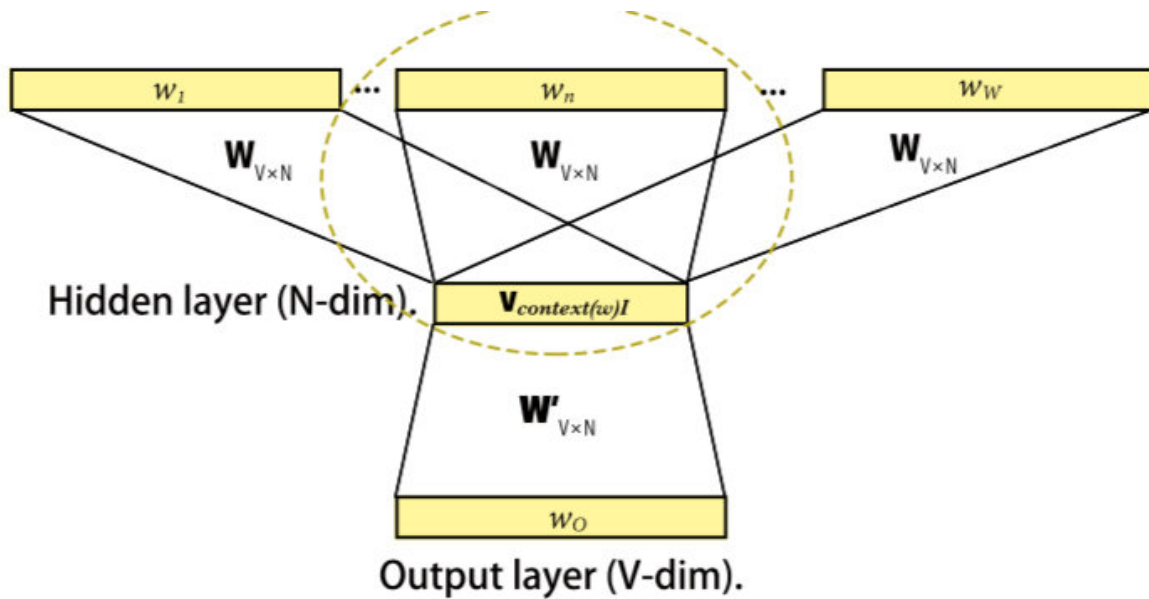
那么具体怎么寻找与目标节点结构相似的节点呢？前面我们介绍了一个向量GDV就派上用场了，对于目标节点的距离 S 邻域内所有节点，比较他们的GDV与目标节点GDV的距离，距离定义为两个向量的余弦相似性，并且在计算距离时，将每个特征对应的重要性程度作为该特征的权重。取前K个距离最大的节点作为与目标节点最相似的节点。对每个节点都这么算一次，得到一个稀疏矩阵 S ，每一项代表两个节点之间的结构相似性。

• Network Embedding Powered by Structural Similarity

这一步使用的是CBOW模型，因为CBOW对于短序列且数据多的情况效果更好。

CBOW加上Negative Sampling就不说了。

下面就是如何利用结构信息加强网络表示使之能够保持结构等价性。



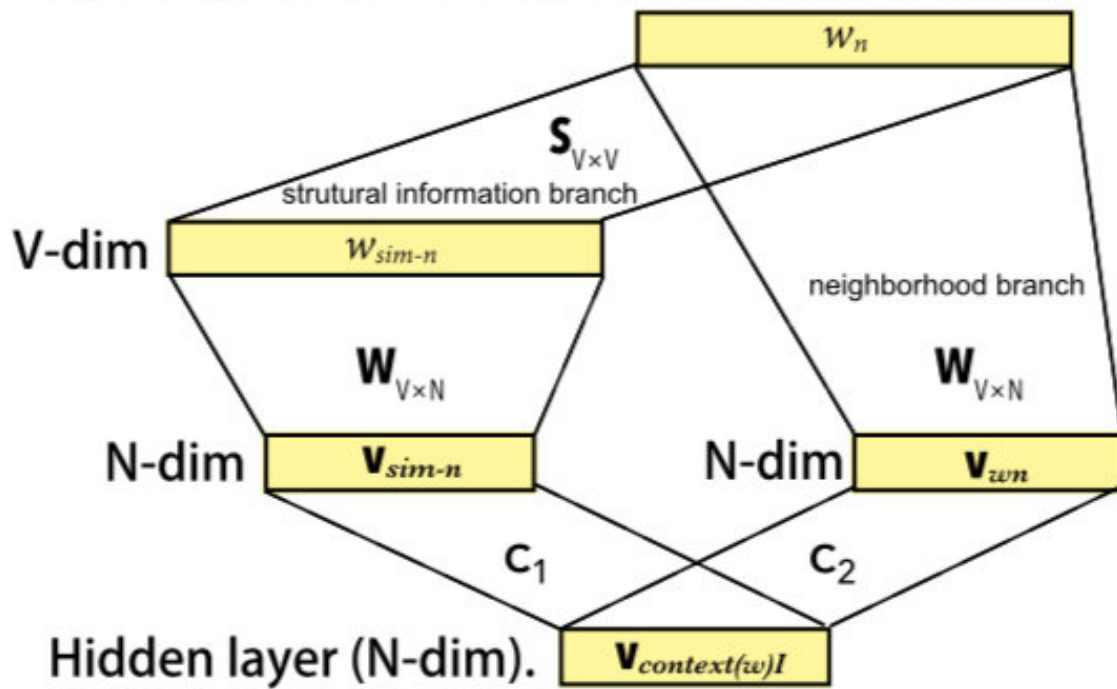
将之前CBOW+Negative Sampling的模型中一部分替换成下图的结构，具体来说，下图右边一部分保持原来的不动，左边部分则表明了如何添加结构信息。

$$\mathbf{v}_{sim-n} = \sum_{m=1}^V s_{nm} \mathbf{v}_{w_m}$$

最后将两部分的信息加权得到新的隐层结果。

$$\mathbf{v}_{w_I} = \frac{1}{W} \sum_{n=1}^W (c_1 (\deg(w_n)) \mathbf{v}_{w_n} + c_2 (\deg(w_n)) \mathbf{v}_{sim-n})$$

Input layer. One-hot encoded vector (V-dim).



Experiment

实验部分只做了可视化的实验和节点分类的实验，由于可视化任务的效果与其他方法各有千秋，从不同的角度看有不同的说法，但是节点分类的任务上效果还行，但是有的数据集上SC的方法居然比LINE还好？