Node2vec理论与实践

# 背景

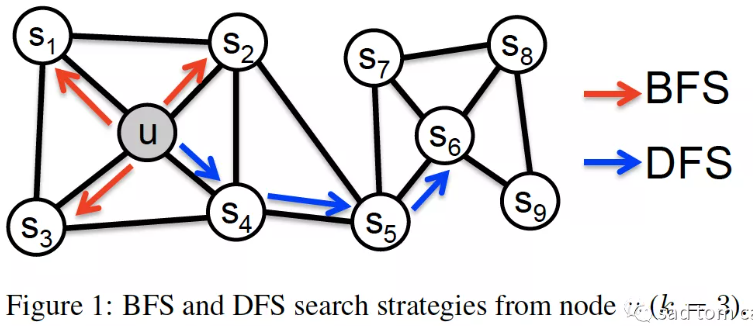
# 简介

Node2Vec是Stanford在2016年发表的论文。

DeepWalk僵硬、死板的地方在于，它的训练结果是依靠随机游走产生的序列的质量，没有解释结点间的关系。Node2Vec在此基础上提出结点间的同义性（homophily）和同构性（structural equivalence）。

# Node2vec

根据一张原文里的图，详细解释结点的关系：



图中的结点大致可以分为两个community：一个是u,s1,s2,s3,s4组成的，以u为中心；另一个是由s5,s6,s7,s8,s9组成的，以s6为中心。

## 同义性

同义性强调一个community里结点间的高度相互连接。例如图中的结点u和结点s1。通过广度优先搜索（BFS）获取的随机游走序列，可以突出结点间的同义性关系（简单理解，使劲转悠还是困在一个community里）。在经过skip-gram之后，同义性高的结点在特征空间里，距离要更近。

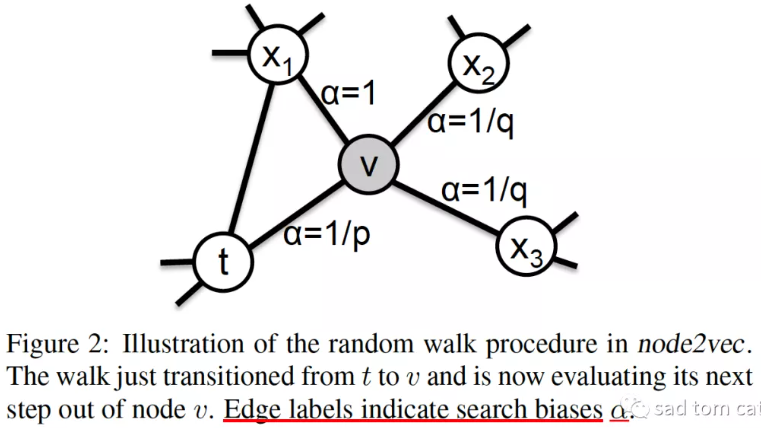
## 同构性

同构性强调结点在网络结构中表现出的结构性的作用。例如图中的结点u和结点s6。通过深度优先搜索（DFS）获取的随机游走序列，可以突出结点间的同构性关系（简单理解，游走的方向更容易跨越community）。在经过skip-gram之后，同构性高的结点在特征空间里，距离也应该更近。

## Node2vec算法细节

Node2Vec通过对同义性和同构性的定义，区分出结点间的关系。那么在随机游走的过程中，样本侧重不同的结构特点，最终就能获得不同的embedding表达。所以接下来，要约定侧重的“力度”。

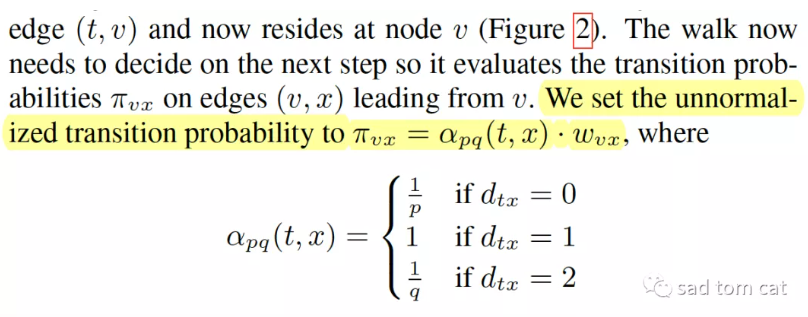
BFS和DFS产生的随机游走序列按照一定比例混合在一起，作为训练样本，确实可以影响模型的训练方向。但需要注意的是，在实际情况中，产生的序列并不是这么完美的，每个序列可能都会是BFS和DFS的混合结果。Node2Vec采用了名叫“搜索偏差”（Search bias）的量影响游走方向，记作。



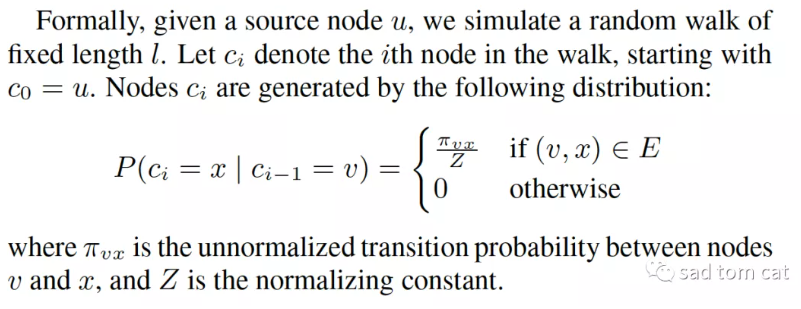
上图是，游走的序已经列从点t移动至点v，现在点面临的是四个选择，{t,x1,x2,x3}。接下来是分类讨论。

* 点t。若从点v重新返回至点t，这是一种“重返”行为，由重返参数（Return Parameter）p控制，此时的搜索偏差；点v的前后两个点都是点t，它们的最远路径长为0。如果p越小，游走的序列里BFS的成分就越浓。
* 点x1。若从点v移动至点x1，点x1与点t也有边连接，它们的最短路径长为1。此时的搜索偏差。
* 点x2和点x3。若从点v移动至点x2或者点x3，点x2、点x3与点t都没有边连接，它们的最短路径长为2。此时的搜索偏差。q被定义为出入参数(In-Out Parameter)。如果q越小，游走的序列里DFS的成分就越浓。

上述过程在给定当前结点时，要根据前一个结点和后一个结点的最短距离给出搜索偏差；搜索偏差和边权重w相乘给出转移概率。



将转移概率归一化，得到最终的转移概率。

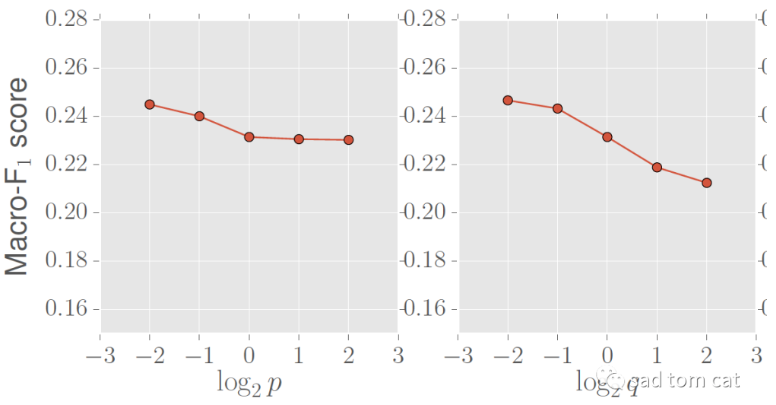


综上来看，这个转移过程是一个2阶马尔科夫过程。经过若干轮的循环遍历，Node2Vec就产生若干个序列，输入给skip-gram模型，得出结点的Embedding表示。

# 实验

## 参数敏感度

笔者只截取了p和q的参数敏感度比较，如下图所示：

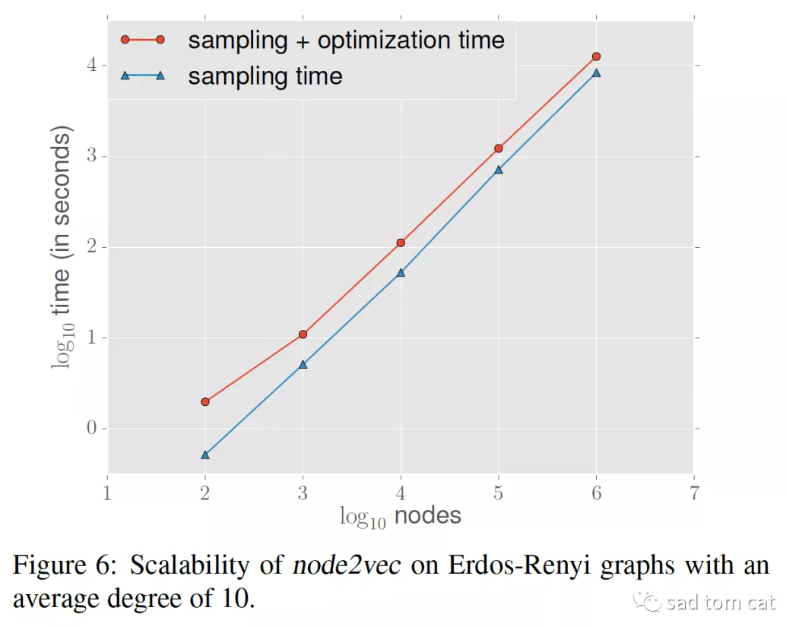


在公开数据集BlogCatalog上，以(多标签分类里有用到)为指标，使用不同的参数，讨论对模型效果的影响。显而易见，不管是增大p，还是增大q，模型的效果都变得更差了。“具体问题具体分析”——针对这个数据集，p小（高重返，偏向BFS）或者q小（高跳出，偏向DFS）对模型分类的结果更友好。

那为什么没有一个折中的参数组合呢？可惜的是原文也没有给出和组合的网格搜索结果。感兴趣的读者朋友可以试试。

## 可扩展性

此处讨论的是结点个数对模型运行时间的影响。下图是在Erdos-Renyi图上依次采取100到1000000个结点（按10的倍数取），对模型运行时间的影响。



这里只分析红色的折线，在这个图上，算上采样和skip-gram训练模型的时间，时间对数和结点数对数基本上是线性的关系，给有意使用Node2Vec的人心里有个底。

# 实践

# 总结

Node2Vec的精彩之处，在于给随机游走的过程提供了一个可控的方向，并且为它的控制做出了合理的解释。原文也给出了Node2Vec与其他方法的对比结果，感兴趣的读者朋友可以看看原文。

参考