DeepWalk

Introduction

DeepWalk是**网络表征学习**的比较基本的算法，用于学习网络中顶点的**向量表示**（即学习图的结构特征即属性，并且属性个数为向量的维数），使得能够应用传统机器学习算法解决相关的问题。

Algorithm Theory

- Input: 邻接表
- Output

  第一行为节点个数和向量维数，后面每行为一个节点的向量表示，第一列为NodeID。
- Innovation

  借助语言建模word2vec中的一个模型，skip-gram来学习结点的向量表示。**将网络中的结点模拟为语言模型中的单词，而结点的序列（可由随机游走得到）模拟为语言中的句子，作为skip-gram的输入。**
- Process

  Random + skip-gram model

Algorithm 1 DeepWalk(G, w, d, γ, t)

Input: graph $G(V, E)$
window size w
embedding size d
walks per vertex γ
walk length t
Output: matrix of vertex representations $\Phi \in \mathbb{R}^{|V| \times d}$

Initialization: Sample Φ from $\mathcal{U}^{|V| \times d}$
Build a binary Tree T from V
for $i = 0$ to γ **do**
  $\mathcal{O} = \text{Shuffle}(V)$
  for each $v_i \in \mathcal{O}$ **do**
    $\mathcal{W}_{v_i} = RandomWalk(G, v_i, \text{t})$
    SkipGram$(\Phi, \mathcal{W}_{v_i}, w)$
  end for
end for

- Random-walk

  Random Walk从截断的随机游走序列中得到网络的局部信息，并以此来学习结点的向量表示。

deepwalk中的实现是**完全随机**的，根据**Random Walk**的不同，后面又衍生出了**node2vec**算法，解决了deepwalk定义的结点相似度不能很好反映原网络结构的问题。

所谓随机游走(random walk)，就是在网络上不断重复地随机选择游走路径，最终形成一条贯穿网络的路径。从某个特定的端点开始，游走的每一步都从与当前节点相连的边中随机选择一条，沿着选定的边移动到下一个顶点，不断重复这个过程。

- Skip-gram model

  skip-gram 是使用**单词来预测上下文**的一个模型，通过最大化窗口内单词之间的共现概率来学习向量表示，在这里扩展之后便是**使用结点来预测上下文**，并且不考虑句子中结点出现的顺序，**具有相同上下文的结点的表示相似。（Ps:两个node同时出现在一个序列中的频率越高，两个node的相似度越高。）**

  结点相似性度量： 上下文的相似程度（LINE中的二阶相似度）

  共现概率根据**独立性假设**可以转化为各条件概率之积即

$$\Pr\left(\{v_{i-w},\cdots,v_{i+w}\}\setminus v_i \mid \Phi(v_i)\right)=\prod_{\substack{j=i-w\\ j\neq i}}^{i+w}\Pr(v_j|\Phi(v_i))$$