

日期： /

Random Walk

Notation :

- Z_u : u 的节点嵌入

- $P(v|Z_u)$: 在由节点 u 开始的一次随机游中访问

$Z_u^T Z_v \approx u$ 与 v 在一次随机 V 的概率 (预测)

游走中出现的
概率 — 随机游走：在 K 时间步内，由某一节点开始，序贯地
从当前节点以均等概率移动至其邻居节
点的过程

Why Random Walks? 具有表达力、灵活，考虑高阶邻居信息
高效，无需考虑所有节点对

Intuition: (自监督式地) 找到节点嵌入，使得图中接
近的节点在嵌入空间中也接近
如何定义接近？

$N_r(u)$ 即 u 的邻域，定义为由 u 开始的一系列
随机游走经过的节点。

Goal: Given $G = (V, E)$, find $f: u \rightarrow \mathbb{R}^d$: $f(u) = Z_u$

$$\max_f \sum_{u \in V} \log P(N_r(u) | Z_u)$$

随机游走策略

日期： /

- 定义随机游走策略 R
- 由起始点 u 开始采集 $N_R(u)$, 即定长随机游走中访问的节点集合
- 极大对数似然 $\max_f \sum_{u \in V} \log P(N_R(u) | Z_u)$
$$L = \sum_{u \in V} \sum_{v \in N_R(u)} -\log(P(v | Z_u))$$
- 对 $P(v | Z_u)$ 参数化：

$$P(v | Z_u) = \frac{\exp(Z_u^T Z_v)}{\sum_{n \in V} \exp(Z_u^T Z_n)}$$

极其昂贵 ($O(|V|)$)!

$$L = \sum_{u \in V} \sum_{v \in N_R(u)} -\log \left(\frac{\exp(Z_u^T Z_v)}{\sum_{n \in V} \exp(Z_u^T Z_n)} \right)$$

\rightarrow k ↑, 更精确, 对负样本有更大概率, 通常 $5 \sim 20$

$$\approx \log(\sigma(Z_u^T Z_v)) - \sum_{i=1}^k \log(\sigma(Z_u^T Z_{n_i})), n_i \sim p_u$$

采集 k 个负样本 每个节点被采集的概率与其度数成正比

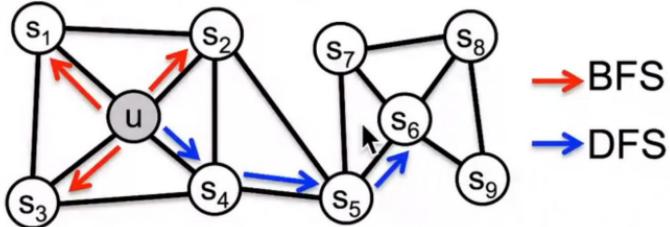
一 通过梯度下降优化损失函数

所以不同方法的区别在于策略 R 的选择：

[deep walk] 从每个节点开始运行固定长度的无偏均匀随机游走 缺乏表现力

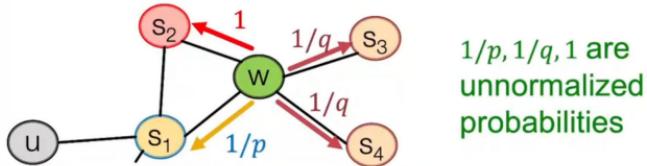
[node2vec] 使用有偏的、灵活的随机游走，在图的全局和局部中平衡
BFS 2^{nd} -Order random walk DFS

日期: /



此策略下的随机游走 $N_r(u)$ 有 2 个参数:

- Return parameter p : 返回前一结点的概率
- In-out parameter q : BFS / DFS



基于 random walk 的方法有线性时间复杂度, 可并行.
但需为每个节点分配一个嵌入向量