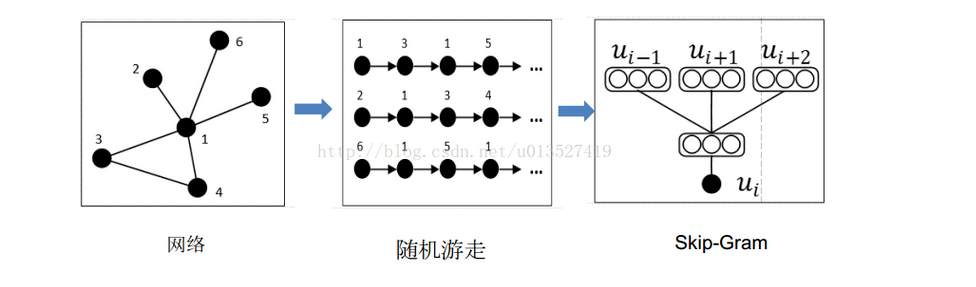
Deep walk

图表示学习baseline

主要思想：不用多说，算是表示学习的起点，借鉴自然语言处理，主要通过

1. random walk 获得网络节点随机游走序列，来描述网络结构
2. word2vec 通过word2vec学习嵌入向量，与NPL一模一样

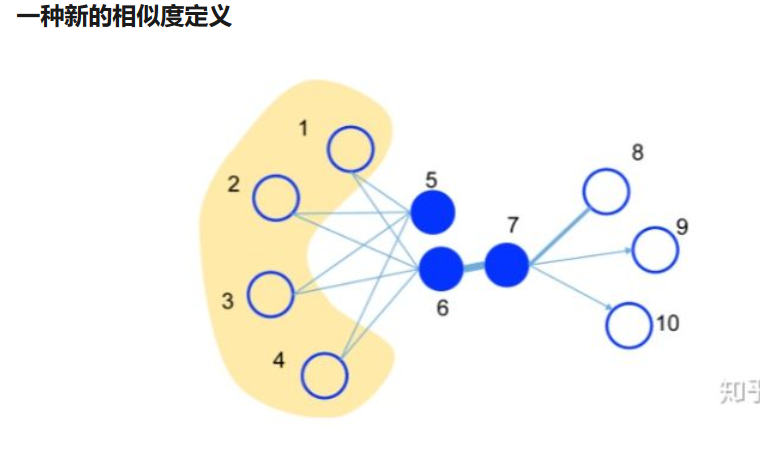
特点：随机游走是无偏的，这种游走生成的节点序列，更加接近图的深度优先搜索，偏向于探寻网络节点高阶的距离较远的空间拓扑结构，个人觉得，这种随机游走方式对于领域和高阶的网络结构探索都存在一定的不确定性，也不够充分。



Line（Large-scale Information Network Embedding）

主要思想：line算法更加关注领域网络结构，更加类似于广度优先搜索

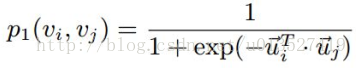
LINE还可以应用在带权图中(DeepWalk仅能用于无权图)。



一阶相似度

1阶相似度用于描述图中成对顶点之间的局部相似度，形式化描述为若 u, v 之间存在直连边，则边权Wuv 即为两个顶点的相似度，若不存在直连边，则1阶相似度为0。 如上图，6和7之间存在直连边，且边权较大，则认为两者相似且1阶相似度较高，而5和6之间不存在直连边，则两者间1阶相似度为0。

通过已经得到的一阶相似度，去学习节点的嵌入向量



v代表节点，u代表节点的embedding。上面式子的意思是两节点越相似，內积越大，sigmoid映射后的值越大，

同时定义经验分布

优化目标为最小化：

d是两个分布的距离，常用的衡量两个概率分布差异的指标为KL散度，K-L散度其实是数据的原始分布p和近似分布q之间的对数差值的期望。

使用KL散度并忽略常数项后有



总结起来大概就是说，用各个节点的嵌入向量的内积sigmoid映射分布去拟合真正的各个边的权重分布，目标函数就是两个分布之间的散度，最小化散度，这样训练得到的嵌入向量学习了边权重的信息，将边的结构和权重转化为个节点的嵌入向量来表示。

二阶相似度

仅有1阶相似度就够了吗？显然不够，如上图，虽然5和6之间不存在直连边，但是他们有很多相同的邻居顶点(1,2,3,4)，这其实也可以表明5和6是相似的，而2阶相似度就是用来描述这种关系的。 形式化定义为，令